2nd Edition

CHAMBERGON BATTLE LOGIC

An INTRODUCTION *to*
SYMBOLIC LOGIC
(*only, more gamey*)

Micah D. Tillman, Ph.D.

© Micah Tillman, 2015. All rights reserved. No duplication or distribution of any portion of this document in any form (e.g., physical or electronic), nor any sale of any portion of this document, is permitted, without express, written permission from the author.

The author can be contacted via http://micahtillman.com/

TABLE OF CONTENTS

Introduction **1**
Part 0: What We're Up To **1**
Intro.0.0: Two Books in One 1
Intro.0.1: Logic as a Game 1
Intro.0.2: How Will This Help? 2
Intro.0.3: What Is Logic? 3
Intro.0.4: What Are Propositions? 3
Intro.0.5: Propositions in Arguments 4
Intro.0.6: How to Connect Propositions to Each Other 4
Intro.0.7: The Logical Connectives 5
Intro.0.8: The Essence of Connectives 6
Intro.0.9: The Truth of Arguments 7
Intro.0.10: Our Mission Is to Prove Validity (not Soundness) 9
Intro.0.11: The Rules of Logic 9
Intro.0.12: Why Bother? 10
Intro.0.13: Conclusion 10

Part 1: Where All of This Comes From **11**
Intro.1.0: Reflecting and Foreshadowing 11
Intro.1.1: Special Terminology for Connectives 11
Intro.1.2: Conjunction Symbols 12
Intro.1.3: The Conjunction Icon 12
Intro.1.4: Disjunction Symbols and Icon 13
Intro.1.5: (Material) Conditional Symbols 13
Intro.1.6: The Conditional Icon 14
Intro.1.7: A Defense of My Choices regarding Fundamentality 15
Intro.1.8: Biconditional Symbols 16
Intro.1.9: The Biconditional Icon 17
Intro.1.9: Negation Symbols 17
Intro.1.10: The Negation Icon 18
Intro.1.11: "Operators," instead of "Connectives" 19
Intro.1.12: Summary of the Operators, Their Symbols, and Icons 19
Intro.1.13: Translating Conjunctive Sentences into Symbols and Icons 20
Intro.1.14: Translating Disjunctive Sentences into Symbols and Icons 20
Intro.1.15: When the Letters (or Colors) Matter 21
Intro.1.16: Translating Conditional Statements into Symbols 22
Intro.1.17: Translating Conditionals into Icons (plus Sufficient Conditions) 23
Intro.1.18: Conditionals and Necessary Conditions 24
Intro.1.19: Making Necessary Conditions Explicit ("Only If" and "Unless") 25
Intro.1.20: Translating Biconditional Statements into Symbols and Icons 26
Intro.1.21: Translating English Negations into Symbols and Icons 27
Intro.1.22: How to Tell Characters Apart 28

Chapter 0:
Level 0 Powers ... 31
0.0: Starting the Game ... 31
0.1: Double Negation ... 34
0.1.0: Negation as Denial ... 34
0.1.1: The Motivation for Clubs ... 35
0.1.2: The Power of Double Negation (A Recycling/Equivalence Rule) ... 36
0.2: The Power of Tautology (A Recycling/Equivalence Rule) ... 38
0.3: Recycling Rules vs. Landfill Rules ... 45
0.4: The Power of Conjunction Elimination (A Landfill/Inference Rule) ... 45
0.5: The Power of Conjunction Introduction (A Landfill/Inference Rule) ... 48
0.6: Conclusion ... 50

Chapter 1:
Level 1 Powers ... 51
1.0: Only Two Rules This Time ... 51
1.1: De Morgan's Theorem ... 51
1.1.0: Who Was De Morgan? ... 51
1.1.1: Why Is De Morgan's Theorem Important? ... 52
1.1.2: The Power of De Morgan's Theorem (A Recycling/Equivalence Rule) ... 52
1.2: The Power of Disjunction Elimination (or "Disjunctive Syllogism") Power (A Landfill/Inference Rule) ... 58
1.3 Conclusion ... 63

Chapter 2:
Level 2 Powers and Activation Games ... 64
2.0: Recap ... 64
2.1: Implication ... 65
2.1.0 Fear of Heights ... 65
2.1.1: The Power of Implication (A Recycling/Equivalence Rule) ... 67
2.2: The Power of Disjunction Introduction (A Landfill/Inference Rule) ... 75
2.3: Activation Games ... 81
2.3.0: Introduction ... 81
2.3.1: The Basic Truth Tables ... 81
2.3.2: Filling in More Complex Truth Tables ... 83
2.3.3: Activation Rules for Chambergon Battle Logic ... 102
2.3.4: More Complex Examples ... 108
2.3.5: Activation Games Strategy ... 110
2.3.6: The Point of Activation Games ... 113

Chapter 3:
Level 3 Powers ... 114
3.0: A Logician's Three Fundamental Moves ... 114
3.1: The Power of Modus Ponens (A Landfill/Inference Rule) ... 115
3.2: The Power of Hypothetical Syllogism (A Landfill/Inference Rule) ... 121
3.3: The Power of Constructive Dilemma (A Landfill/Inference Rule) ... 129

Chapter 4:
Level 4 Powers and Encoding Games 135
4.0: Chapter Overview 135
4.1: The Power of Modus Tollens (A Landfill/Inference Rule) 135
4.2: The Power of Contraposition (A Recycling/Equivalence Rule) 141
4.3: The Power of Commutation (A Recycling/Equivalence Rule) 151
4.4: A Logician's Fourth Fundamental Move 163
4.5: Encoding Games 164
4.5.0: The Background Story 164
4.5.1: Basic Encodings 165
4.5.2: The Procedure for More Complex Encodings 167
4.5.3: Encoding Entire Games 172
4.5.4: More Coding Details 173

Chapter 5:
Level 5 Powers 176
5.0: Detour Rules 176
5.1: The Power of Assumption (A Landfill/Inference Rule?) 176
5.3: The Power of Conditional Proof (A Landfill/Inference Rule) 179
5.3: The Power of Indirect Proof (or "Reductio ad Absurdum")
(A Landfill/Inference Rule) 187

Chapter 6:
Level 6 Powers 203
6.0: Introduction 203
6.1: The Power of Exportation (A Recycling/Equivalence Rule) 204
6.2: The Power of Association (A Recycling/Equivalence Rule) 215
6.3: The Power of Distribution (A Recycling/Equivalence Rule) 226

Chapter 7:
The Level 7 Power(s) and Propositional Powers Summary 235
7.0: One Last Rule (and All the Others Too) 235
7.1: The Power of Biconditional Equivalence (A Recycling/Equivalence Rule) 235
7.2: Summary of the All Propositional Logic Rules 244
7.3: Overview of the Main Rules 254
7.4: Rules Abbreviations Complete 256

Chapter 8:
Predicate Logic and Level 8 Powers 257
8.0: New Pieces 257
8.0.0: Entertainment for Characters 257
8.0.1: Square Lenses/Monocles/Glasses 258
8.0.2: Circular Lenses/Monocles/Glasses 261
8.0.3: Coins 262
8.0.4: Domes and Bowls 263
8.0.5: The Logic of Lenses, Domes, and Bowls 265

8.1: The Power of Quantifier Negation (A Recycling/Equivalence Rule) 268
8.2: The Power of Universal Elimination (A Landfill/Inference Rule)..... 278
8.2.0: Free and Bound Lenses..... 278
8.2.1: Domes with a Character Inside 279
8.2: Domes with a Chambergon Inside..... 285
8.3: The Power of Existential Elimination (A Landfill/Inference Rule) 292
8.3.0: Characters inside Bowls..... 292
8.3.1: Chambergons inside Bowls 298
8.4: Why Universal Elimination and Existential Elimination Work Like They Do..... 304
8.5: Encoding the New Pieces 307
8.6: A Note about Coins 312

Chapter 9:
Level 9 Powers..... 315
9.0: The Last Three Powers..... 315
9.1: The Power of Existential Introduction (A Landfill/Inference Rule) 316
9.1.0: Putting Characters into Bowls 316
9.1.1: Putting Chambergons inside Bowls..... 318
9.2: The Power of Universal Introduction (A Landfill/Inference Rule)..... 320
9.2.0: Putting Characters inside Domes..... 320
9.2.1: Putting Chambergons inside Domes..... 327
9.3: The Power of Identity Elimination (A Landfill/Inference Rule)..... 327
9.4: A Note about Lens Colors and Shapes for UI and IE..... 329
9.5: Decoding Games (The Other Kind of Encoding Games)..... 331
9.5.0: Decoding Propositional Logic Games 332
9.5.1: Predicate Logic Games 346
9.5.2: More Complex Predicate Logic Games..... 352

Introduction
Part 0: What We're Up To

Intro.0.0: Two Books in One

The book you are holding in your hands . . . I mean, the book you are reading on this screen . . . is actually two books in one. It is a textbook, if you treat it like one. It will teach you modern symbolic logic, which is good for you if you want to learn symbolic logic (i.e., if you want to program computers or design database searches). But you probably don't want to learn symbolic logic.

Fortunately, therefore, this book is also the player's manual for a game, if you treat it like one. It will teach you to play Chambergon Battle Logic, which is a puzzle-solving card game that borrows some of the "leveling" mechanics of role-playing games.

The reason one book can play both roles is that symbolic logic is itself a puzzle-solving game. (No, it *is*. Seriously. Just . . . look, just trust me for a minute.) The problem is that symbolic logic is normally presented through symbols that make it look like algebra. There's nothing necessarily wrong with this; it *is* called "symbolic" logic, after all. But most people find algebra frustrating and frightening, and that means they see symbolic logic as a chore, not a game.

Intro.0.1: Logic as a Game

What the world needs, then, is a different way of presenting symbolic logic. We need symbolic logic *without symbols*. And that is where Chambergon Battle Logic ("CBL") comes in. CBL makes two simple substitutions — and suddenly something that used to look like math begins to look like a game you might actually enjoy.

Here's the reason symbolic logic usually looks like math to students. Normal math uses two different types of symbols. There are numerals ("0, 1, 2, 3, 4, 5, 6, 7, 8, 9") and operators ("+, –, ×, ÷") that connect those numerals together. In symbolic logic, there is a similar distinction. There are the letters of the alphabet ("A, B, C, . . . X, Y, Z") and operators ("∧,∨ ,¬,⊃" or "&,∨, ~, →," or "&&, ||, !, if(){}").

To make it obvious that symbolic logic is a game, therefore, we need to translate those two types of symbols into two types of . . . something else. And we need those types of "something else" to look more gamey. (I know. That's not what "gamey" actually means. I know.) What Chambergon Battle Logic does is to translate the letters of the alphabet into "Characters," and the operators into "Chambergons." Here's what the Characters look like:

And here's what the Chambergons look like:

Intro.0.2: How Will This Help?

"That's adorable," you might be tempted to respond, "but how is it going to help me? Just because you translated the symbols of symbolic logic into shapes and colors doesn't mean it's going to be easier. *Bambi* and *Dumbo* are cartoons about adorable animals, but that doesn't mean they aren't traumatizing movies. And Philosophy classes use books translated into English from ancient Greek and modern German. But they're still impossible to understand."

That is an excellent objection, my friend. (We are friends now.) So allow me to explain my theory.

We use different parts of our brain to do different things. We hear with the auditory cortex. When we're seeing, however, we use our visual cortex. And something similar is true of thinking. When we're thinking about words, we use one part of our brain, but when we're thinking about numerals or doing math, we use another part of our brain.[1]

Now, if symbolic logic normally seems like math to people, and people don't enjoy doing math, what we need to do is help them use the other, "non-math" parts of their brains to do symbolic logic. We need to find a way for them to do symbolic logic, but to use the parts of their brains that they find easier and more enjoyable to employ.

When Chambergon Battle Logic translates symbols into shapes and colors, that is exactly what it is doing. Instead of making your brain use its abilities to think about symbols and numbers, it is inviting your brain to use its abilities to think about how you deal with things that have particular shapes, and how you can transform one shape into another. Furthermore, by using little quasi-anthropomorphic "Characters," Chambergon Battle Logic allows you to use the parts of your brain that reason about emotions and motivations.

"That's great and all," you might respond, "but won't there still be all kinds of weird rules I'll have to follow, just like all the weird rules in algebra?"

Yes, there will be. All the rules of symbolic logic are still in effect. However, the rules are translated too. Instead of being rules about what to do with symbols, they are rules about how to tear a shape apart, or morph one shape into another. (If you know how to open a birthday present, or make a paper airplane, you won't have any problem.) Or they are rules about how a Character will react emotionally if you suddenly drop him from a great height. (Hint: He'll get

[1] See, e.g., Brian Butterworth, "Numbers in the Brain," in *What Counts: How Every Brain Is Hardwired for Math* (New York: Free Press, 1999); "Foundational numerical capacities and the origins of dyscalculia," *Trends in Cognitive Sciences* 20, Special Issue: "Space, Time and Number" (2010), 1–8; Rochel Gelman and Brian Butterworth, "Number and language: How are they related?," *Trends in Cognitive Sciences* 9, no. 1 (January 2005), 6–10.

scared. Obviously.)

"But those aren't logical rules. Those are like, I don't know, *practical* rules or emotional rules."

True enough. But they have exactly the same structure as the logical rules. You can translate back and forth between them, whenever you want. What you do with the Characters and Chambergons is exactly mirrored by what a normal, miserable symbolic logic student would be doing with symbols (and vice versa). It's just you'll be having fun while doing it.

Intro.0.3: What Is Logic?

But before I go any further, however, I need to explain what logic even is.

Logic is the study of arguments. But logic doesn't study the kind of arguments where people are yelling at each other. It studies the kind of arguments where a person calmly tries to convince someone of something.

To convince someone of something, you have to offer the person at least one reason to believe that "something." And for your attempt-to-convince to count as an *argument*, both the "reason" and the "something" have to be propositions. You can't just threaten to punch the person if she or he doesn't agree.

So, when you make an argument, you give someone a reason—in the form of a proposition—to believe some conclusion—which is also in the form of a proposition. But that won't make any sense unless you know what a proposition is.

Intro.0.4: What Are Propositions?

Take a look at the following two sentences.

1. "Girls don't like boys; girls like cars and money."[2]
2. "Boys only want love if it's torture."[3]

I think we can agree that they are different sentences, right? Now, take a look at the next two sentences.

3. "We've been spending most our lives living in a gansta's paradise."[4]
4. "We've been spending most our lives living in a gansta's paradise."[5]

I think we can agree that sentences 3 and 4 are identical. They are the same sentence. They are *one* sentence, repeated *twice*.

But what about the following three sentences?

---

[2] Good Charlotte, "Girls & Boys," *The Young and the Hopeless* (Epic/Daylight, 2003).
[3] Taylor Swift, "Blank Space," *1989* (Big Machine/Republic, 2014).
[4] Coolio, "Gangsta's Paradise," *Gangsta's Paradise* (Tommy Boy/Warner Bros., 1995).
[5] Ibid.

5. "I'm gonna swing from the chandelier."[6]
6. "I am going to hang from that fancy light fixture while rocking to and fro."
7. "I am about to do something wild."

I think we can agree that those are three different sentences. However, I think we also might agree that all three sentences are saying the same thing. What does Sia mean when she sings the first sentence? To be very literal about it, she means the same thing as I would mean if I said the second sentence. But if her night of binge-drinking concluded without her ever swinging from a chandelier, you wouldn't necessarily call her a liar. So long as she had done *something* crazy, that would count as "swinging from a chandelier."

Sentences 5, 6, and 7 all make the same basic claim, in other words. But when you make a claim, you are proposing something; you are saying that it is true, and inviting other people to take it as true. So, sentences 5, 6, and 7 "all express the same proposition." They are three different sentences that mean or say the same thing, that make the same claim.

Intro.0.5: Propositions in Arguments

Logic is the study of how propositions can be connected together to make good arguments. Take the following two arguments, for example.

Argument 1

1. It has been really cold outside all day, and I don't like going out in the cold.
2. I do not do things that I do not like.
3. Therefore, I did not go out in the cold today, no matter what Bill told you.

Argument 2

1. The temperature out of doors has been very low today, and I find being exposed to such low temperatures uncomfortable.
2. I refuse to engage in activities that I find uncomfortable.
3. While William may have claimed that I left the house today, therefore, he must be lying, since you can see from points 1 and 2 that I would not have left the house today.

The sentences in Argument 1 are not the same sentences as in Argument 2. However, you can tell that Arguments 1 and 2 are the same argument. While they are not made of the same sentences, they are made of the same *propositions*. (The first sentences in both arguments are making the same claim. The second sentences in both arguments are saying the same thing. The third sentences in both arguments are proposing that the same thing is true.)

A logician, therefore, will treat Arguments 1 and 2 as the same argument. Since each is made of the same propositions—and those propositions are connected to each other in the same ways—the arguments are identical as far as a logician is concerned.

Intro.0.6: How to Connect Propositions to Each Other

In the 300s BC, Aristotle explained to his fellow philosophers that what makes an argument good

[6] Sia, "Chandelier," *1000 Forms of Fear* (Monkey Puzzle/RCA, 2014).

or bad is its structure. That is, an argument is good or bad depending on how the propositions it contains are connected to each other.

Take this argument, for example.

Argument 3
1. It is raining outside.
2. There is cheesecake in the refrigerator.
3. Therefore, Sam is going to the store.

Those three propositions seem to be connected by the word "therefore." Or, more precisely, they seem to be connected by the idea that the first two propositions are evidence for the third proposition. The only problem is, they aren't. The three propositions have nothing to do with each other. The "therefore" connection between them is an illusion.

However, we can add in some connections that will make the argument work.

Argument 3, revised
1. It is raining outside [and whenever it rains, I make a cheesecake].
2. [Whenever I make a cheesecake, I have to put it in the refrigerator.]
3. [Therefore,] There is cheesecake in the refrigerator.
4. [Whenever Sam finds cheesecake in the refrigerator, he goes to the store to by chocolate icing for the cake.]
5. Therefore, Sam is going to the store.

In that example, I connected the original propositions to each other primarily by adding new propositions. However, in the 1800s AD, a mathematician named George Boole (and his followers) worked out another way of connecting propositions with each other, using the words "and," "or," and "if-then."

Intro.0.7: The Logical Connectives

Take the following two propositions for example.

1. "You can get with this."
2. "You can get with that."[7]

I can connect those two propositions in any of the following ways.

1. "You can get with this *and* you can get with that."
2. "You can get with this *or* you can get with that."[8]
3. "*If* you can get with this *then* you can get with that."

We encounter propositions all the time that are connected like in these three ways. There are

[7] Black Sheep, "The Choice Is Yours," *A Wolf in Sheep's Clothing* (Mercury, 1991).
[8] This is the way it is put in the song (ibid.).

and-connected propositions like:

- "I am a champion, and you're gonna hear me roar."[9]
- "You're my end and [you're] my beginning."[10]
- "You can call me 'Queen Bee', and baby, I'll rule."[11]

There are *or*-connected propositions like:

- "You can be the outcast or [you can] be the backlash of somebody's lack of love."[12]
- "You better hurry up . . . , or I'll be gone, gone, gone."[13]
- "It doesn't matter if you love him or [if you love] capital 'H I M.'"[14]

And there are *if-then*-connected propositions, like:

- "Clap along if you feel like happiness is the truth."[15]
- "If I was your boyfriend, I'd never let you go."[16]
- "If you do not hear me then you will be history, Kendrick."[17]

Intro.0.8: The Essence of Connectives

Take a look at the third "or-connected" proposition above. Logically-speaking, Ms. Gaga is saying, "You love him, or you love HIM. Either way, it doesn't matter." The logical connection in the line is between the two propositions, "You love him," and "You love capital 'H I M'." So, the word "or" is doing the connecting, while "if" is just a synonym for "whether." ("Whether you love him, or you love HIM, is irrelevant.")

What this shows us is that we can't just look for "and," "or," and "if," if we want to find the "seams" between propositions that have been sown together. Sometimes a word that usually signals a logical connection turns out not to have any logical significance. Sometimes "if" just means "whether." And sometimes, words other than "and," "or," and "if" turn out to be one of the normal connectives, wearing a disguise.

"And-connected" propositions, for example, aren't always connected by the word "and." Sometimes, they're connected by the word "but." If you say, "You are kind *and* I'll be going," you're saying that both propositions ("you are kind," and "I'll be going") are true. And if you say, "You are kind *but* I'll be going," you're still saying that both propositions are true. The reason you use "but" instead of "and" is that you know people will be surprised that you're going, even though you told the person she or he was kind.

---

[9] Katy Perry, "Roar," *Prism* (Capitol, 2013).
[10] John Legend, "All of Me," *Love in the Future* (GOOD/Columbia, 2013).
[11] Lorde, "Royals," *Pure Heroine* (Lava/Republic, 2013).
[12] Sara Bareilles, "Brave," *The Blessed Unrest* (Epic, 2013)
[13] Everly Brothers, "Gone Gone Gone," *Gone Gone Gone* (Warner Bros., 1964)
[14] Lady Gaga, "Born This Way," *Born This Way* (Streamline/Interscope, 2011).
[15] Pharrell Williams, "Happy," *Girl* (i Am Other/Columbia, 2014).
[16] Justin Bieber, "Boyfriend," *Believe* (Island/RBMG, 2012).
[17] Kendrick Lamar, "Swimming Pools (Drank)," *good kid, m.A.A.d city* (Aftermath/Interscope, 2012).

What matters with the three proposition-connectors ("and," "or," and "if"), in fact, is how they link the truth values of the propositions they connect to the truth values of the larger propositions they help to form. Take these two propositions: "I am reading a book," and "It is raining outside." The first proposition is true. Logicians say it's truth value is "true" (as opposed to "false"). The second proposition may be true or false. I don't know. It depends on if it is actually raining outside. Is it? If it is, the truth value of the second proposition is "true." If it isn't, the truth value of the second proposition is "false."

Now, let's link those two propositions into one larger proposition, using the word "and": "I am reading a book and it is raining outside." Is that whole proposition true? I know the first part of it is true, because you are reading this book right now. If the second part is also true (if it is actually raining) then the whole sentence is true. It says both that you are reading and that it is raining, and both of those things are actually true. However, if it isn't raining, the whole proposition is a lie. It says both that you are reading and that it is raining, but it isn't raining. You are reading while rain falls, you are reading while no rain false. The first half of the sentence is true, the second half is false, and that makes the whole thing false.

Contrast this with the connective, "or." Imagine that a friend asks you what is going on, and you respond, "Either I am reading, or it's raining" because you honestly can't remember which. Imagine furthermore that it turns out that you aren't reading, but it is raining. Did you lie to your friend? You said, "Either I'm reading or it's raining," and while you aren't actually reading, it is actually raining? So, did you lie? Was the thing you said false?

The first part of your proposition, "I am reading," is false. The second part, "It is raining," is true. But when you said, "Either I am reading or it is raining," you weren't saying, "It is true *both* that I am reading *and* that it is raining." You were just saying, "One of the following two things is true: I am reading, or it is raining." So long as one of those two things is actually true, the whole proposition is true.

What this means is that the connective logicians call "conjunction" (i.e., the words "and," "but," "although," "even though," etc.) joins two propositions together creating a larger proposition that is true if and only if both the small propositions are true. The connective that logicians call "disjunction" (i.e., the word "or," in English), on the other hand, joins two propositions together, creating a larger proposition that is true so long as at least one of the smaller propositions are true.

Intro.0.9: The Truth of Arguments

Once logicians had discovered all the rules for the "connectives" (both-and, either-or, if-then, etc.), they were able to work out the rules for making good arguments. You can join two smaller propositions into a single, larger proposition. The truth value of the larger proposition is a result of the truth values of the smaller propositions, and depends on which connective you used. Similarly, you can join propositions together into arguments, and the truth of the argument as a whole depends on the truth of the propositions you used to put it together.

Logicians, however, don't talk about whether an argument is true or not. Instead, they talk about (1) whether it is "valid" or not, and (2) whether it is "sound" or not. Let's talk about validity first.

Recall that an argument has two parts. The first part is the conclusion you are trying to convince someone to believe. The second part is the reason or reasons you are giving the person in support of your conclusion. If you have all the right propositions in your argument, and you have them linked together using the correct connectives, then an interesting thing happens. If the reasons you give in the argument turn out to be true, then the conclusion will have to be true too.

Now, logicians normally don't use the word "reasons" when they are talking about arguments. Instead, they use the word "premise" for "a single reason," and "premises" for "multiple reasons." So, we can rephrase what we said in the previous paragraph like this: if (a) you argument has the right premises and the right conclusion, then (b) if it turns out that the premises are all true, that means the conclusions will have to be true too. That is what it means for an argument to be "valid."
Let me give you an example.

Argument 4

1. If a musical artist has the most number one singles in a year, then that musical artist is the greatest musical artist of all time.
2. The Beatles have most number one singles in a year.[18]
3. Therefore, the Beatles are the greatest musical artist of all time.

You will notice that the first proposition (the first premise) is actually two propositions linked together by the words "if-then." The second proposition (the second premise) is just a single proposition. It contains no connecting words. Instead, it kind of repeats the first half of the first proposition: "The Beatles have the most number one singles in a year" is just a more specific version of, "a musical artist has the most number one singles in a year." Similarly, the third proposition (the conclusion), is just a single proposition, connected to everything that went before it by the word "therefore." That single proposition, "the Beatles are the greatest musical artist of all time" is just a more specific version of the second half of the first proposition, "that musical artist is the greatest musical artist of all time."

This argument, therefore, has the following structure. It starts with an "if-then" proposition as its first premise. It repeats the first half of that "if-then" proposition, in a more specific form, as its second premise. It then repeats a more specific version of the second half of its "if-then" proposition as its conclusion.

Now, we ask ourselves: "If the two premises of Argument 4 were true, would that force the conclusion to be true as well?" And the answer is, "Yes." *If* the first two lines were true, the third line would have to be true too. We can get into a debate about whether the first line is *actually* true, but that is irrelevant at the moment. For an argument to be *valid*, all it has to do is to have the right structure. And you know an argument has the right structure if its conclusion would have to be true if all its premises were true.

---

[18] "List of Billboard Hot 100 chart achievements and milestones," Wikipedia, http://en.wikipedia.org/wiki/List_of_Billboard_Hot_100_chart_achievements_and_milestones#Most_number-one_singles_in_a_calendar_year.

Intro.0.10: Our Mission Is to Prove Validity (not Soundness)

You might wonder, however, what we call an argument that is valid, when its premises all turn out to be true. The answer is, we say that the argument is "sound." A "sound" argument is a valid argument, all of whose premises are true. (And since the conclusion of a valid argument has to be true if all its premises are true, the conclusion of a sound argument is true *by definition*.)

The problem is that, as logicians, we have no idea whether the premises of our arguments are true. The arguments we study don't have words in them. They just have symbols. Or, if you're lucky, they have adorable little Characters hiding in boxes and triangles and stuff. And there's no way of knowing whether the propositions they stand for are true, because we don't even know what propositions they stand for.

Our job, therefore, is not to prove that arguments are sound. We would have to be able to show that their premises are true to do that, and we can't do that. Our job, instead, is to prove that arguments are valid. Our job is to take arguments and show that if their premises turned out to all be true, then their conclusion would necessarily turn out to be true too.

We do our job as logicians by showing that, if you start with the premises of an argument, and follow the rules of logic, you'll eventually come to the conclusion of the argument, even if the argument hasn't spelled out all the intermediate steps. Doing logic, in other words, is like working out a maze. You know where you start and where you're supposed to end up; the fun is figuring out how to get from one to the other.

Intro.0.11: The Rules of Logic

In a maze, there are only four things you can do (besides sitting down and giving up). You can go forward. Or you can turn around. Or you can turn left. Or you can turn right. To get through the maze, you have to do those four things the right number of times, in the right order. The same is true when you're proving that an argument is valid in logic. There are a certain number of "moves" you're allowed to make, and you have to make them the right number of times, in the right order.

The moves you are allowed to make in logic are called "rules," but that makes them sound too boring. It's better to think of them as "powers," like magical powers in a game. The more you play, the more points you earn, they higher your level is, and the more powers you can use. And with more powers available to you, you can solve more puzzles—you can successfully navigate more logic mazes.

Depending on how you count, there are 27 or 28 rules in symbolic logic. That means there are 27 or 28 powers for you to unlock in Chambergon Battle Logic, and one of your central goals in the game is to unlock all of them.

You may be wondering, however, why we spent so much time talking about connectives before, when now it sounds like logic is all about rules. Well, as you will see in the pages below, the rules all have to do with the things you are allowed to do to or with a proposition, given the connectives ("and," "or," "if") that it contains. If a proposition consists of two smaller propositions connected by an "and," you can take it apart, like Lego blocks. And then you can

use those small propositions (those smaller Lego blocks) to build new propositions that will get you to where you need to go. If, however, a proposition is made of two smaller propositions connected by an "or," you can just pry them apart. You have to bring in another proposition of a very particular kind to smash one of the smaller propositions and free the other.

Intro.0.12: Why Bother?

The actual practice of logic boils down to proving that arguments are valid, using a specific set of rules. But why bother? What's the point of studying it?

There are four basic answers to this question. The first answer is that you have to know this stuff if you are going to program computers. Computers function—at a fundamental level—completely in terms of symbolic logic, so you have to understand symbolic logic in order to understand computers.

The second reason for studying logic is closely related to the first. If you want to work with databases (in a library, for example, or at a company that stores customer data, or at a search engine like Google or Bing, or as a scientist collecting data from experiments), you have to know symbolic logic in order to craft searches. A search is, in effect, the premises of an argument, while the results are its conclusion. You have to know how the rules of logic work to know how to get the right results.

The third reason for studying logic is that it is excellent brain exercise. If you want to keep your mind sharp, working out symbolic logic proofs is a good way to do it. In this sense it is like Sudoku or crossword puzzles.

And fourth, just like Sudoku and crossword puzzles, symbolic logic is a game. It's a game with practical uses, yes. But it's still a game. And that means it's fun.

Intro.0.13: Conclusion

Things don't really get going until Chapter 0. That's where we'll actually introduce the game of Chambergon Battle Logic, and start exploring how logic is actually done. For the moment, however, let's look back briefly at what we've discussed in Part 0 of this Introduction.

I started by trying to explain why I have taken the unusual approach of presenting logic as a game in this book. I didn't mention it there, but Lewis Carroll (yes, *that* Lewis Carroll) did something similar back in 1886. Carroll was a logician, whose real name was Charles Lutwidge Dodgson, and he wrote a book called *The Game of Logic*, which explained his own method for gamifying our subject.[19]

My justification for gamifying symbolic logic, however, is not that Carroll/Dodgson did it, but that symbolic logic is a game in itself, and that treating it as a game should help more of us not only be able to do it, but also to enjoy it. Specifically, I proposed that the way I have translated the symbols of symbolic logic into "Characters" and "Chambergons" will help us use different parts of our brains than we would otherwise be forced to use.

---

[19] C. L. Dodgson/Lewis Carroll, *The Game of Logic* (London: McMillan & Co., 1886). http://www.gutenberg.org/files/4763/4763-h/4763-h.htm.

I then provided you with a tedious discussion of what logicians do. They analyze arguments, which consist of propositions that have been joined together in particular ways. We discussed what propositions are, and as well as devoting some time to their connections. We then talked about the logicians tasks of proving that arguments are "valid," using the rules of logic. In the next chapter, we will learn four of those rules, and begin to play the game.

Part 1: Where All of This Comes From

Intro.1.0: Reflecting and Foreshadowing

But first!

Whew! That Part 0 was something else, am I right? A mighty good read. Relaxing. And look! The Introduction keeps going!

In chapter 0, we are going to talk about being a Level 0 player. At Level 0, there are four powers available to you, which correspond to four of the most basic rules of symbolic logic. We'll talk about how to use those powers, and why they are logically legitimate.

Before we get to that, however, we need to talk about how we get from normal English sentences to the symbols of symbolic logic, and the icons of Chambergon Battle Logic. Unfortunately, that is going to be a long, boring process. By the end of Part 1, you'll hate me and everything I stand for. Then, Chapter 0 will make you think I might be okay, maybe. And you'll try to forgive me.

I bet you've been wondering, however, why this is "Part 1" of the Introduction, even though it's the second part. Shouldn't it be "Part 2"? And, for that matter, why was the first part called "Part 0"? ("Wait just a second!" you think to yourself, as realization begins to dawn. "It's as if Tillman starts counting at '0' instead of '1'! What is wrong with this guy?")

What is wrong with me is that my bachelor's degree is in computer science, and whenever a computer scientist wants to count, she or he starts at "0" instead of "1."

"Why?" you might ask. "Isn't it silly to make '0' the first numeral, when '1' pretty much means 'first'?"

"Yes," I respond. "Yes, it is silly. But it also makes us programmers feel different and special. And now you too can feel different and special with us. It's like having a secret handshake."

Intro.1.1: Special Terminology for Connectives

In Part 0, we talked about the three "connectives" that logicians focus on: "and," "or," and "if" (or "both-and," "either-or," and "if-then"). But we also noted that the three connectives can be represented by different words in different contexts. And we also noted that sometimes the normal connective words don't actually represent connectives.

(In other news, "connective" is a silly word. "Connector" would be more sensible. But it doesn't sound as fancy. And we logicians are very fancy.)

In any case, to avoid confusion, logicians talk about "conjunctions," "disjunctions," and "conditionals" instead of "ands," "ors," and "ifs." It's especially helpful to do so since the primary word for conjunction is "and" in English, but "et" in French, "y" in Spanish, and "и" in Russian.

It would be even more helpful if we didn't have to use a word for "and" at all. We don't have to write "plus" every time we add something in math, after all. Wouldn't it be nice if there were a simple symbol for "and," like there is a simple symbol for "plus"? Well, as it turns out, there is.
In the next few sections, I'm going to discuss the symbols for each of the main logical connectives. It's going to be tedious. If you find yourself falling asleep, just skip ahead to the summary in section Intro.1.12.

Intro.1.2: Conjunction Symbols

The proposition-connector that we usually use "and" to stand for is called "conjunction." Two propositions are "conjoined" together by "and." But there is also a *symbol* for conjunction. Actually, there are two different symbols that you will see, depending on which logician you're talking to.

"and" = "conjunction" = ∧
"and" = "conjunction" = &

The first symbol is not on your keyboard. If you want to insert it, you can use the Cambria Math font . . . but it still won't be on your keyboard. In Word, you'll find "∧" in the "Symbol" dialog box (look for the button that has an "Ω" on it, and says "Symbol"). Just make sure that you set your font to "Cambria Math" in the Symbol dialog box.

If you want to use "∧" often, find it in the Symbol dialog box and assign a "Shortcut key" combination to it. I, for instance, use "alt+6" for it, since it looks like the caret on the 7 key. In fact, if you're pressed for time, you can probably get away with using a caret instead. However, I think it's best to think about "∧"as a capital "A" (for "And"), without the horizontal bar.

The second version of the logical "and symbol" is much easier to access. It's the ampersand, and is on the "7" key on your keyboard. It means "and," which makes it easy to remember. In fact, I'm going to be using "&&" instead of "∧" in this book, because that's what you (usually) do in programming languages like Java, C++, Perl, Ruby, Swift, and PHP. ("Why '&&' instead of '&'?" I'll explain when we get to truth tables.)

Intro.1.3: The Conjunction Icon

In Chambergon Battle Logic, however, we prefer not to use symbols at all. We use icons instead. Here, therefore, is the Chambergon Battle Logic icon for conjunction:

It's a rectangle divided into two chambers. (And it has eyes on top, because that makes it look funny. And funny is better than boring. Also, if it has eyes, it's easier to anthropomorphize it, which will help us to engage the parts of our brain that we use to reason about emotions and motivations. And if we are using those parts of our brains, we'll be able to deal with the rules of logic better than if we just had to use the "math" parts of our brains.)

This kind of Chambergon is called a "Parcel" or "Package," because—if you ignore the eyes—it looks like a box with a piece of string tied around the middle. An Parcels are "Chambergons" because they rectangles are a kind of polygon, and a polygon divided into chambers is a "Chambergon." Get it? Good.

Anyway, for the moment, what you need to remember is that "and" usually stands for the logical connective called "conjunction," that "&&" is the symbol we'll be using for conjunctions in this book, and the icon we'll be using to stand for conjunctions is the Parcel (or "Package," if you prefer).

Intro.1.4: Disjunction Symbols and Icon

If two propositions are connected by the word "or," the connective in question is "disjunction." They are "conjoined" but "disjoined." There are, once again, two normal symbols for disjunction

"or" = "disjunction" = ∨
"or" = "disjunction" = |

Once again, our first symbol is not available on our keyboard, but can be accessed using the Cambria Math font and the Symbol dialog box. The second symbol is a "vertical bar" (or "pipe") and is on the "\" key (which is right above the Enter key . . . on my keyboard, anyway). In this book, I'll be using "||," for the same reason I'm using "&&."

The Chambergon Battle Logic icon for disjunction, however, looks nothing like "||." Here it is:

That icon is called a "Paragon," since it is the parallelogram-shaped Chambergon. You might recognize it as a Parcel that has been "squashed" slightly, and is now tilting to the left. The "string" down the middle—or the partition between chambers—has also gone crooked, so it's not a Parcel anymore. And, given that we use it to stand for disjunctions (rather than conjunctions), it's best to give it its own name. So, it's a "Paragon," not a "Squashed Parcel."

Intro.1.5: (Material) Conditional Symbols

When two propositions are connected by "if" (or by "if-then"), they form a "conditional" proposition. The "if" part of the larger proposition specifies the "conditions" for the "then" part. The proposition, "If you go to the store, then I will come with you," says that I will go to the store with you on the condition that ("if it turns out that") you are yourself going.

To make it possible to deal with conditional statements in logic, we treat them as "material" conditionals. We call them "material" conditionals in order to say we're going to be following specific rules for analyzing their "truth values" that are stricter than the rules we follow in normal conversations.[20] But for the most part we won't care that the rules we are following are stricter than normal (we haven't even talked about what they are, yet), so we'll often just use the term "conditional" without the word "material."

As with conjunctions and disjunctions, there are two normal symbols for conditionals.

"if" = "material conditional" = ⊃
"if" = "material conditional" = →

Unfortunately, however, neither of those are available on your keyboard. Times New Roman contains the arrow as a special symbol, but you'll need Cambria Math for the horseshoe. And in both cases, you'll either need to use the Symbol dialog box, or you'll need to set up a shortcut/hotkey.

If you're in a hurry, of course, you can always type a hyphen and then a right angle bracket (or a "dash" and a "greater-than" symbol), like this: "->." Unfortunately, that looks more like an arrow in some fonts than in others. In Times New Roman it doesn't look very good (see above), but in Courier New, it looks okay: "–>."

"But what symbol do they use in programming languages?" you ask. The answer, unfortunately, is there really isn't one. In most "normal" programming languages, you have to write out the word "if" (and in Visual Basic you even have to write out the word "then"). To write the proposition, "If you go to the store, then I'll come along," in Java, for instance, you would have to write "if(you.getGoingToStore()){me.setGoingToStore(true);}." In other words, the basic "symbol" for conditionals would be something like, "if(){}."[21]

In this book, therefore, I'm going to do what you would do if you were having a conversation online with someone. You would type "–>" and hope it looked good enough.

Intro.1.6: The Conditional Icon

Just as with conjunctions and disjunctions, conditionals get their own icon in Chambergon Battle Logic. It is called a "Pyramid":

---

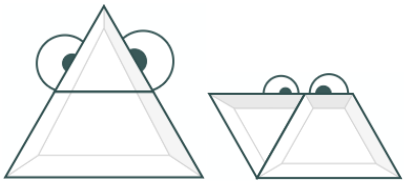
[20] See the "Philosophical treatments" section of, "Counterfactual conditional," *Wikipedia* (http://en.wikipedia.org/wiki/Counterfactual_conditional#Philosophical_treatments), the "Distinctions between the material conditional and the indicative conditional" section of, "Indicative conditional," *Wikipedia* (http://en.wikipedia.org/wiki/Indicative_conditional#Distinctions_between_the_material_conditional_and_the_indicative_conditional), and the more philosophical "Indicative Conditionals," by Dorothy Edgington, *Stanford Encyclopedia of Philosophy* (http://plato.stanford.edu/entries/conditionals/).

[21] In some languages, you can use the symbol "?" with the symbol ":" to do conditionals, like this: "itIsRaining ? bringUmbrella : leaveUmbrella." But that makes it look like conditionals tie one thing to two other things, which isn't really what's going on.

The Pyramid is different from the other two Chambergons we've seen, in that its two chambers are located one-on-top-of-the-other, instead of side-by-side. However, imagine what would happen if you "popped open" a Pyramid by tilting its top chamber up and to the right. Imagine that you kept the lower left-hand corner of the upper chamber where it is, but rotated the rest of the upper chamber around till it was upside down. What shape would you have then? (Ignore the eyes.)

Do you see it?

Here, let me put to two icons beside each other:



Do you see how you can make a Paragon by flipping the top of a Pyramid to the left, and can make a Pyramid by flipping the left chamber up on top of the right chamber? (You'd have to stretch the right chamber out a bit, but still.)

In fact, since a Paragon is just a squashed Parcel, and a Pyramid is just a Parcel with its left chamber flipped up on top, you can think of all three icons as being the same shape, mashed and stretched into different forms.

Intro.1.7: A Defense of My Choices regarding Fundamentality

You now know the symbols we will be using for the three main connectives (&&, ||, ->), and the icons we will be using for those symbols (the Parcel, the Paragon, and the Pyramid). If you've studied logic before, however, you've been tearing out your hair every time I said that conjunctions, disjunctions, and conditionals were the main connectives in logic. First, you were enraged that I left out negation. Second, you realized that conditionals are often treated as less fundamental than conjunctions, disjunctions, and negations. And, third, you thought that if I was

going to treat conditionals as fundamental, why not go all the way and include biconditionals too? Because these questions have been so emotionally draining on you for so long, I now owe you some answers.

First, I do not include negation among the *connectives* of symbolic logic because negations *do not connect* two propositions together to form a larger proposition. Since they don't connect, negations cannot be connectives. It is just plain silly to call them that (and yet many people do).

Second, I include conditionals as one of the fundamental connectives of symbolic logic because I'm still enamored of the (strictly false) idea that the conditionals we are analyzing in symbolic logic are the same as the conditionals we use in actual language. I know they aren't exactly the same, but I like to pretend. And while we will see that you can duplicate the functionality of *material* conditionals by using the right combination of negations and disjunctions (or negations and conjunctions), we couldn't do that in normal language. So, I treat conditionals as being just as fundamental as conjunctions and disjunctions.

Third, I do not include biconditionals among the fundamental connectives of symbolic logic because we express them by using the word "if" in normal language, and they can easily be reduced to a combination of conjunctions and material conditionals in symbolic logic (as we will see). That is, I treat them as non-fundamental for the same kind of reason others treat conditionals as non-fundamental. (Does this make me a hypocrite? I shall have to ponder that.)

However, just because I do not believe that negations and biconditionals are fundamental connectives doesn't mean we aren't going to use them. In fact, let's talk about biconditionals now, and negations after that.

Intro.1.8: Biconditional Symbols

While the (material) conditional connective is usually expressed by the word "if," or the words "if" and "then," the (material) biconditional connective is usually expressed by the words "if, and only if." For example, "I'll go to the store if, and only if, you go too." This is a stronger statement than "I'll go to the store if you go too." "I'll go to the store if you go to" says that if you go to the store, I'll tag along. However, it leaves open the question of whether I would go to the store under any other circumstances. I might also be willing to go if Sam went, but you didn't. If I say, "I'll go to the store if, and only if, you go too," in contrast, I'm being very clear that I will go to the store if you go, and if you do not go, I will not go either.

If you're around people who know logic, you will find them often writing "iff" instead of "if and only if." Writing the word "if" with an extra "f" is just shorter. However, you will also often find them using one of the following symbols.

"if, and only if" = "iff" = material biconditional = $\equiv$
"if, and only if" = "iff" = material biconditional = $\leftrightarrow$

The symbol "≡" is also used to mean, "is defined as," and I think it looks too much like the equals sign to properly express biconditionality. Thus, I think the double-headed arrow is clearer.

Neither symbol, however, is available on your keyboard, though both are included in Times New Roman and can be accessed through the Symbol dialog box.

In this book, I will imitate my choice for conditionals, and write "<->" for biconditionals.

Intro.1.9: The Biconditional Icon

An "if, and only if" statement says that A will be true only if B is true, and B will be true only if A is true. It is a combination of two "if" statements then. In fact, it's a conjunction of two if statements.

Since we are using the Pyramid to represent conditionals, and biconditionals are a combination of two conditionals (they are *bi*conditionals), therefore, we use the Pulsar as our icon for biconditionals:

Do you see how the Pulsar is a Pyramid facing up, "crashed into" a Pyramid facing down? Good. It's a single shape made of two conditionals (two Pyramids). It's a *bi*conditional.

Notice, however, that you cannot make a Pulsar out of a Paragon. If you flipped a Paragon's left chamber up on top, you'd get a Pyramid. But you need *two* Pyramids to make a Pulsar. That is why biconditionals (<->, Pulars) are not fundamental, while conjunctions (&&, Parcels), disjunctions (||, Paragons), and conditionals (->, Pyramids) are. They are not fundamental because they are not the same basic shape as the others, "just squashed or stretched into a new form." They are two shapes, crushed together.

Intro.1.9: Negation Symbols

It is important in logic to recognize not only that people connect propositions together into larger propositions, but that people also negate propositions. We can say, "I am going to the store," but we can also say, "I am *not* going to the store." We can say, "You claim that Bob is a cool guy," but we can also say, "You are *wrong* that Bob is a cool guy." We can not only propose that something is true, but we can propose that something is false.

This means that we logicians need to have a way of dealing with negations. We need a symbol for the word "not," or the words "that is false." Lucky for us, there are a *lot* of negation symbols:

"not" = "negation" = ¬ = ~ = – = !

The first symbol ("¬") is a negation symbol that is available in the Symbol dialog box. The second ("~") is the tilde that is on the "`" key (which is beside the "1" key on my keyboard). The third sign ("–") is technically an en-dash, which is not available on your keyboard. However, it is

available in the Symbol dialog box, and the hyphen key on your keyboard provides a perfectly-acceptable substitute. And the fourth sign ("!") is just an exclamation mark, which is on the "1" key.

Of the four signs for negation, the exclamation mark is the one most often used in computer programming, so it is the one I will use. (Note: you may hear "!" read as "bang," instead of "exclamation mark" or even "not," in settings were programmers are wandering around. For instance, they might read "1 != 2" as "one bang equals two" instead of "one does not equal two.")

Intro.1.10: The Negation Icon

I did not get very creative when trying to come up with the Chambergon Battle Logic icon for negation. I thought to myself, "I'm using '!' to represent negation, like a good computer programmer. And you know what exclamation marks look like? They look like clubs. And we computer programmers pronounce them 'bang!', like you got hit on the head with a club. So, let's just use exclamation marks for negation, and call them 'Clubs'."

And that's just what I did. Here is a Club:

Except, you can't have Clubs just standing there by themselves. They'd fall over. Someone has to hold them up. So, you're more likely to see something like this:

Or like this:

Yes, the Parcel is "holding" the Club, even though it has no hands. It's magic, okay?

Intro.1.11: "Operators," instead of "Connectives"

Before turning to our summary, I should say a word about the special status of negation. I've already argued that it is not a "connective" since it doesn't connect two propositions to each other. Instead, it negates a single proposition (even if that single proposition is a combination of other propositions). Nevertheless, negation belongs in the same class as logical connectives like conjunction and disjunction, rather than in the class of propositions.

Since negation, like conjunction and disjunction, is something you *do to* propositions, I'm going call it an "operator." In fact, I'm going to call negations, conjunctions, disjunctions, conditionals, and biconditionals all "operators" from now on. This will help us to remember that in doing symbolic logic, we are doing the same kind of thing that mathematicians and computer programmers do (since both mathematicians and programmers talk about "operations" and "operators").

I don't want you to get the impression that calling negation and the connectives "operators" was my idea, however. It's very common. I'm just letting you know that I've made a choice between options, and will follow that choice for the rest of the book, but other people choose differently.

Intro.1.12: Summary of the Operators, Their Symbols, and Icons

If you read all of that rubbish I just wrote, I want to thank you. If you didn't, well, I guess you can just look at the following chart. It gives you a handy summary of what you need to know. Feel free to check it again later to refresh your memory!

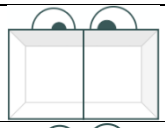
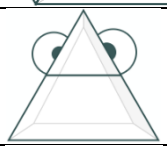
Logical term for the operator	**Common English phrase for the operator**	**Symbol we will use in this book**	**Other equivalent symbols**	**Chambergon Battle Logic Icon:**
Negation	Not	!	¬, ~, –	
Conjunction	And (or "Both-And")	&&	∧, &	
Disjunction	Or (or "Either-Or")	||	∨, |	
(Material) Conditional	If (or "If-Then")	->	⊃, →	
(Material) Biconditional	If, and only if (or "iff")	<->	≡, ↔	

Chart 1.

Intro.1.13: Translating Conjunctive Sentences into Symbols and Icons

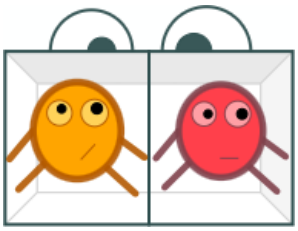
In symbolic logic, we say that the following propositions all have the same structure:

- "I am going to the store and you are going to the store."
- "It is raining outside and the groundhogs have grown restless."
- "The yoke of oppression is present and we must stop eating so much cotton candy."

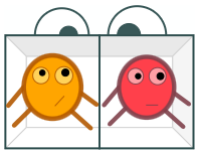
Each of the above propositions is actually a conjunction of two smaller propositions. Each fits the following form: "_____ and _____." However, in symbolic logic, we don't use blanks and words. We use symbols. So, we would write, "A && B," or "B && C," or "Q && R," or something like that.

In "A && B," symbol "A" stands for the first proposition in some conjunction, while the symbol "B" stands for the second proposition. For example, in, "I'm on top of the world and I will rule with an iron fist," "A" = "I'm on top of the world," while "B" = "I will rule with an iron fist." However, it might be better if we used "O" for "I'm on top of the world," since "on top of" starts with "o." And it might be better to use "R" for "I will rule with an iron fist," since "rule" starts with "r." It doesn't matter logically, but it might help us remember what the symbols stand for.

If we do that—if we symbolize "I'm on top of the world and I will rule with an iron fist" as "O && R," then we would "iconize" the proposition like this:



In that icon, the orange Character stands for "O," the red Character stands for "R," and the Parcel stands for "&&." So:

"I'm on top of the world and I will rule with an iron fist." = O && R = 

Intro.1.14: Translating Disjunctive Sentences into Symbols and Icons

We can do the same kind of thing with disjunctions. The following propositions all have the same "disjunctive" structure:

- "I will pass this class or I will end up being sad"
- "You either say 'hello' or you say 'howdy'."
- "The ocean is very wide or ice cream melts quickly in the sun."

Each of the above propositions is actually a disjunction of two smaller propositions. Each fits the following form: "_____ or _____." However, as I said before, we don't use blanks and words in symbolic logic. We use symbols. So, we would write, "D || E," or "P || Z," or "M || N," or something like that.

Instead of, "I will feel better tomorrow or I will go to the library," a logician would write, "B || G," where "B" stands for the proposition, "I will feel better tomorrow," and "G" stands for the proposition, "I will got to the library." Actually, a logician could just as easily write, "A || B." It doesn't actually matter what letters we use unless we care about remembering what the letters stand for. ("B" might help us to remember "better" and "G" might help us to remember "go.")

If we use "B" and "G," then we would iconize the proposition like this:

In that icon, the blue Character stands for "B" (which stands for "I will feel better tomorrow"), and the green Character stands for "G" (which stands for "I will go to the library"). So:

"I will feel better tomorrow or I will go to the library." = B || G =

Intro.1.15: When the Letters (or Colors) Matter

The only time it matters what letters we use is when the same proposition comes up in multiple places, and we want to acknowledge that fact. Take the following propositions, for example:

1. I am bored and should yell my head off.
2. I should either yell my head off or play soccer.
3. If I should play soccer, then I should run a lot.

The first proposition is a conjunction of two smaller propositions, while the second is a disjunction of two smaller propositions, and the third is a conditional consisting of two propositions. We could, therefore, symbolize the three as follows:

1. A && B
2. P || Q
3. Y -> Z

However, B and P are the same proposition ("I should yell my head off"), and it would be smart to symbolize this. Similarly, Q and Y are the same proposition ("I should play soccer"). So, we should symbolize . . .

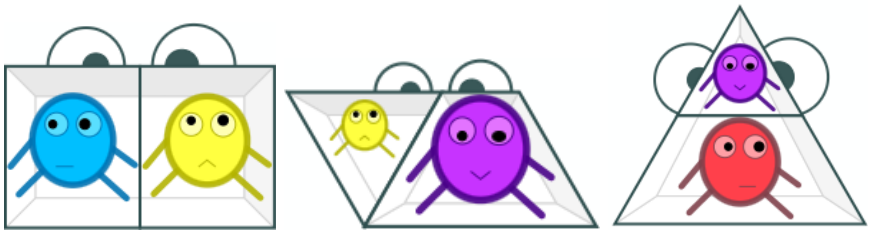
1. I am bored and should get up.
2. I should either go to bed or play soccer.

3. If I should play soccer, then I should run a lot.

. . . like this:

1. B && Y
2. Y || P
3. P -> R

Which would look like this in Chambergon Battle Logic icons:



Getting into the habit of automatically using the same symbols—or same colors—to translate the same propositions, and different symbols—or different colors—for different propositions, is a key part of becoming a mature logician.

Intro.1.16: Translating Conditional Statements into Symbols

If you want more practice with all this translating stuff (and who wouldn't?!), I would heartily suggest you take a course in Critical Thinking. At McDaniel College—a wonderful school you may have heard of—the course is PHI1101, and some guy named "Tillman" happens to be teaching a section of it this semester.

However, while *translating* from English into symbols can't be the focus of a symbolic logic course (*working with* the symbols should be the focus), we do need to at least get familiar with how all five of operators listed in Chart 1 can be found in normal English sentences.

The most tricky operators to deal with are the conditionals. Take the following sentence, for instance: "If you loved me, you would stay." This is a conditional proposition that consists of two smaller propositions: "You love me," and "you stay." We know that—in this book—the conditional operator is "->," and that it stands for the word "if." However, the word "if" in, "If you really loved me, you would stay," doesn't appear *between* the two constituent propositions. It comes *before* them.

You might be tempted, therefore, to symbolize, "If you loved me, you would stay," as "->LS," where "L" stands for the proposition, "You love me," and "S" stands for the proposition, "You stay." But the "->" operator is one of the connectives, and connectives go between. So, in the symbolization, you should put it where the "then" would go, if the sentence had a "then" in it.

"If you loved me, [then] you would stay" = "If _____ then _____" = "L -> S"

I know, it's weird to think of the "->" symbol as representing "if," but to put it where "then" belongs, but there's a method to our madness. Remember that "and" is "both-and," and "or" is "either-or." Remember, furthermore, that the "&&" symbol goes where the "and" is in the sentence, and the "||" symbol goes where the "or" is. But "and" is the second word in "both-and," just like "or" is the second word in "either-or." So, it would make sense to put the "->" symbol where the second word in "if-then" goes.

Sometimes, of course, we say conditional propositions backwards: "I'll go if you say 'Please'." This time, the "if" really does seem to connect the propositions, "I'll go" and "You say 'Please'," since we said it in between them. But as you know if you've ever sat between two people who "aren't talking to each other" before, just because you're between two things doesn't mean you connect them.

So, to properly translate conditional statements into symbolic form, we have to rephrase them in "if-then" format.

"I'll go if you say please" = "If you say please, then I'll go."

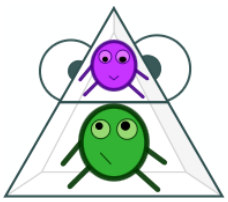
And then we put the "->" where the "then" is.

"I'll go if you say please" = "If you say please, then I'll go." = "P -> G"

In symbolic form, "P" represents the proposition, "If you say please," and "G" represents the proposition, "I'll go." We could have used "A" and "B," or any other two letters, but I find it helpful to use letters that remind me of the predicates (the verb parts) of the propositions being symbolized.

Intro.1.17: Translating Conditionals into Icons (plus Sufficient Conditions)

If we have a proposition symbolized as "P -> G," then we can easily translate it into icons. The "->" symbol is represented by the Pyramid icon. "P" is the antecedent, and thus we should put a purple Character in the "attic" of the Pyramid. "G" is the consequent, and thus we should put a green Character in the "basement" of the Pyramid.



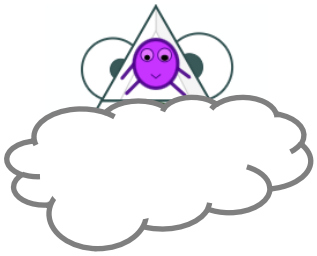
If it helps, think of the arrow in "P -> G" as pointing in the direction of gravity, toward the center of the earth. Thinking of the arrow in this way will help us to remember how conditional propositions are related to the ideas of "necessary and sufficient conditions."

Whenever you have a proposition like "P -> G," the part before the arrow is called the "antecedent," while the part after the arrow is called the "consequent." In a conditional statement, the antecedent ("P") states a sufficient condition for the consequent ("G"). That means

that if the antecedent proposition is true, that is sufficient to make the consequent proposition true. “If you say please, then I’ll go,” means that your saying please is enough to get me to go (it is “sufficient for” my going).

Actually, that’s not *quite* right—for a logician, anyway. When we logicians analyze conditional statements, we treat them as material conditionals. And that means we are working with the truth values of propositions, not actual facts out in the world. So, when a logician reads, “If you say ‘please’, then I’ll go,” the logician will hear, “If (it is true that) you say ‘please’, then (it is true that) I’ll go.” The logician doesn’t care whether your saying please will *make* me go. The logician only cares about the truth or falsity of the propositions, “You say please,” and “I will go.”

Therefore, as far as logicians are concerned, if the antecedent of a (material) conditional is true, that is a sufficient condition for the consequent of the (material) conditional being true. This is reflected in the Pyramid icon. If there is a heavy fog, but you see the attic of a Pyramid poking out of it, that is sufficient condition for concluding that there must be a basement holding it up.



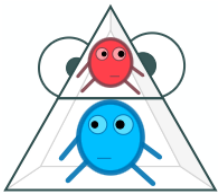
If the fog then clears, and you see that there is a green Character in the basement, you know that it was there, supporting the purple Character in the attic all along.

Intro.1.18: Conditionals and Necessary Conditions

In a (material) conditional proposition, the consequent states a necessary condition for the antecedent. (There *has to be* someone in the basement of the Pyramid holding up the attic.) For a logician, this means that the consequent is a proposition that absolutely must be true whenever and wherever the antecedent is true. This doesn’t mean that the consequent has to describe something that comes *before* the antecedent, however. The order in time doesn’t matter. All that matters is that the consequent must necessarily be true if the and when the antecedent is true.

Take this conditional proposition, for example: “If you run very fast, then you will be out of breath.” For a logician, this says, “If ‘You run very fast’ is true, then ‘You will be out of breath’ must be true too.” If we write this in symbols, we would have, “R –> B,” which says that the truth of “R” (i.e., “You run very fast”) is a sufficient condition for the truth of “B” (“You will be out of breath”) while the truth of “B” is a *necessary* condition for the truth of “R.” Wherever “R” is true, “B” will have to be true too.

Let’s use a red Character for “R” and a blue Character for “B.” If we do, we end up with this:



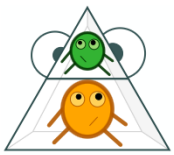
This says that the Character in the attic "rests on," or "depends on" the Character in the basement. If the red Character is going to be there, the blue Character has to be there too, holding up the red Character. The blue Character's being there is a "necessary condition" for the red Character's being there.

At the moment, we are discussing a case where the red Character represents "R," which represents, "You run very fast," and the blue Character represents "B," which represents, "You will be out of breath." The claim that "if you run very fast, then you'll be out of breath" doesn't say you can't become winded in other ways. You might get out of breath by having to climb stairs, for instance. So, "R -> B," in other words, doesn't say that "R" is a necessary condition for "B." It says that R's being true is sufficient for B's being true, and that B's being true is necessary for R's being true. The presence of the red Character in the attic is sufficient for you to conclude there must be a blue Character in the basement, holding up the attic, while the blue Character's being in the basement is necessary for the red Character to be in the attic (without falling out).

Intro.1.19: Making Necessary Conditions Explicit ("Only If" and "Unless")

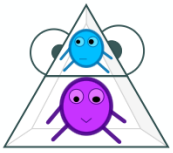
Conditional statements usually emphasize the antecedent, by putting an "if" in front of it, and often drop the "then" altogether. However, we sometimes emphasize the consequent, by using the phrases "only if" and "unless." For example, imagine that your friend Bob says, "I will go to the movie, but only if you get at least one other person to come too." What Bob is saying is that your getting at least one other person to go to the movie is a necessary condition for his agreeing to go.

Let's use "G" for "I will go to the movie," and "O" for "You get at least one other person to go." This is a conditional statement, so we will use an "->" symbol. The question is whether we should write "G -> O" or "O -> G." Our answer is that the sufficient condition goes in the antecedent (in the attic), and the necessary condition goes in the consequent (in the basement). Since we already figured out that your getting at least one other person to go to the movie was the necessary condition, that means "O" has to be the consequent: "G -> O."



Something similar happens when we use the word "unless." Take, for example, the conditional proposition: "Tina won't believe Bob unless Sally punches George." This proposition says that Sally's punching George is a necessary condition for Tina's believing Bob. Let's use "B" for

"Tina believes Bob," and "P" for "Sally punches George." Since the necessary condition always goes in the consequent (in the basement), we put "P" *after* the "->" symbol: "B -> P."



What this means then is that wherever we see the words "only if" or "unless," we know that what comes next is the *necessary* condition, and should go in the consequent or basement. What's tricky about "unless" statements is that they are often stated negatively ("I will *not* do *x* unless you do *y*"), and getting them translated into symbols and icons often requires us to first rephrase them positively (like I did in the previous paragraph).

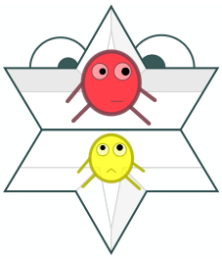
Furthermore, we have to remember that while "only if" signals a consequent/basement/necessary condition, the word "if" by itself signals an antecedent/attic/sufficient condition. So, you know. Watch out for that. It's a Hobbit-filled world out there, full of sentences that are tricksy and false and want to steal your Precious.

Intro.1.20: Translating Biconditional Statements into Symbols and Icons

English biconditional statements are "if, and only if" propositions. Notice that "if, and only if," adds an extra "if" onto the front of "only if." Why is that something to notice? Well, we were just saying that the word "if" (by itself) signals that what comes next is an antecedent/attic/sufficient condition, while the words "only if" signal that what comes next is a consequent/basement/necessary condition. That means that what comes after the phrase "if, and only if," has to be both an antecedent and a consequent, both an attic and a basement, both a sufficient condition and a necessary condition.

If, therefore, I have a statement like, "I will run this business, if, and only if, you help me," the proposition, "You help me" (1) is both a necessary and sufficient condition for "I will run this business," (2) is both an antecedent and a consequent to "I will run this business," and (3) is both in the attic above, and in the basement below, "I will run this business. To symbolize, "I will run this business, if, and only if, you help me," therefore, we need an arrow that points both ways: "R <-> Y."

To iconize that proposition, furthermore, we need a Pyramid that "points both ways." But that means we need two Pyramids at once. Luckily for us, we have the Pulsar:



Notice that in this Pulsar, the red Character is in the attic of the Pyramid that points up, but in the basement of the Pyramid that points down, while the yellow Character is in the basement of the Pyramid that points up, but in the attic of the Pyramid that points down. Each is an attic for the other, and thus a sufficient condition for the other, while being a basement for the other, and thus being a necessary condition for the other. In other words, in "if, and only if" propositions, each of the two constituent propositions is a necessary and sufficient condition for the other.

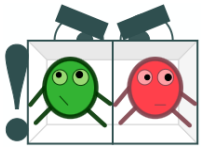
Intro.1.21: Translating English Negations into Symbols and Icons

Now, here's where things get fun. We have to translate the word "not," in addition to phrases like, "it is false that," "it is not the case that," "is a lie," and so on, into symbolic logic as well. If I say, "You can be good," I can symbolize that as "G." But if I say, "You can*not* be good," or, "It is not true that you can be good," or, "It is false that you can be good," I have to symbolize the proposition as "!G." In icons, that would be:

Look how grumpy he is! He's got a club, and he is not happy.

If, however, someone said, "It is not true that you can be both good and rich," we would have to symbolize the proposition like this: "!(G && R)." Notice that the "!" is outside the parentheses, not inside with the "G" and "R." The person isn't saying you can't be good. The person is saying you can't be both good *and* rich. That means the negation applies to the conjunction of goodness and richness, not to goodness all by itself. So, the "!" goes outside the parentheses, not beside the "G."

If we were to iconize "!(G && R)," we would have:



Notice that in this icon, the green Character is perfectly happy. It doesn't have a Club. The Parcel is the angry one with the Club. This signifies that the Parcel is the one being negated, not the two Characters inside.

It is perfectly possible, however, to conjoin two negated propositions. Take, for instance, "I am not old and I am not bored." That would be: "!O && !B" or "(!O && !B)." (The outside parentheses don't really make any difference, though I will usually include them from now on.) In icons, it would look like this:

And then you could add another negation to the whole thing! For instance, you might say, “It’s not the case that I am not old and not bored,” or, “I said I wasn’t old and wasn’t bored, but that was false.” Then you would have “!(!O && !B),” or:

Now everybody’s grumpy, even the Parcel. And it’s not only Parcels that can have Clubs. Paragons, Pyramids, and Pulsars can have them too. Sometimes, in fact, a Character or Chambergon will have *two* clubs. We call propositions like that “double negations.”

We don’t use double negations much in English, but sometimes we do. Take, for instance, the following three claims.

1. I am ready.
2. I’m not ready.
3. Bob said I wasn’t ready. He was lying. (In other words, “I am *not* ‘not ready’.”)

We would symbolize those as:

1. R
2. !R
3. !!R or !(!R)

In Chambergon Battle Logic, those three would be (1) , (2) , and (3) . Notice that in symbols, both exclamation marks in the third case “go in front” of the “R,” while in icons, the Character gets to hold the extra Club in its other hand. The “second Club,” which is the furthest to the left in the symbol, “!!R” (or “!(!R)”) is the furthest to the right in the icon.

Intro.1.22: How to Tell Characters Apart

We’ve talked about the different Chambergons, but haven’t really discussed the different Characters. There are eight: red, orange, yellow, green, blue, brown, purple, and pink. You may have noticed, furthermore that I like to use each Character to stand for a specific letter.

= B

= A

= P

= Z

= R

= O

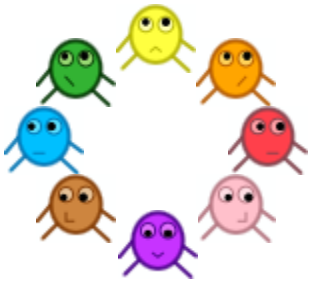
= Y

= G

Chart 2.

As you can see from Chart 2, the red Character stands for “R,” the orange Character for “O,” the yellow Character for “Y,” and the green Character for “G.” Since the blue Character stands for “B,” we use the brown Character for “A,” and since the purple Character stands for “P,” we use the pink Character for “Z.” That’s not twenty-six letters, of course, but it’s enough. We never need more than eight letters at a time when doing symbolic logic.

It is important to me, however, that we not exclude colorblind players. My father is colorblind, and it would be rather mean of me translate symbolic logic from a form he can do (using symbols on a page) to a form he can’t do (using icons with various colors). So, to make sure everyone can play, I have made each Character have both a color and a “direction.” You know two Characters are the same color—they stand for the same letter—if they are looking in the same direction. (Their mouths, furthermore, point in the direction they are looking.)



Here’s the key: red and blue are on opposite sides, with red on the right (and looking to the right) and blue on the left (and looking to the left). To remember this arrangement, simply recall one of the following three facts.

1. “Red” and “right” both start with “r,” while blue is the opposite of red (like left is the opposite of right).
2. The red wavelengths of light are longer, and thus end up on the right side of color/wavelength graphs, while blue wavelengths are shorter, and thus end up on the left side of the same graphs.
3. Republicans are “on the Right” (and Republican states are “Red States”) while Democrats are “on the Left” (and Democrat states are “Blue States”).

Rising like the sun at the top of the color wheel is yellow. And since it is rising like the sun, the yellow Character looks straight up. Halfway between yellow and red—looking diagonally up and to the right—is orange, since orange paint is a mixture of yellow and red. Halfway between yellow and blue—looking diagonally up and to the left—is green, since green paint is a mixture of yellow and blue.

If you have a hard time remembering whether green looks up and to the left or up and to the right, just remember:

1. “The Greens” tend to side with Democrats “on the Left.”
2. You type “g” with your left hand, and “o” with your right hand.

Since yellow gets the top of the color wheel as the brightest color, and since purple is the darkest

color, purple gets the bottom. This puts purple halfway between red and blue, which is appropriate since purple paint is a mixture of red and blue. Since yellow is at the top, and looks straight up, purple—being at the bottom—looks straight down.

With purple at the bottom, however, two spaces are left open in the circle. We fill those with brown and pink, since those are the only other colors that The Typical American Male knows. We put pink on the right next to red, since pink is just light red, and that means brown goes on the left next to blue.

Using brown and pink, however, creates a problem. Brown can't stand for "B," since that's already taken care of by blue. And pink can't stand in for "P," since that's already taken care of by purple. So, we need to use brown and pink to stand for some other pair of letters, and we need that pair to be obvious so they will be easy to remember. I think "A" and "Z" are the best choice, with brown standing for "A," looking down and to the left, and pink for "Z," looking down and to the right. Furthermore, putting brown = A on the left, and pink = Z on the right matches the placement of A and Z in the alphabet.

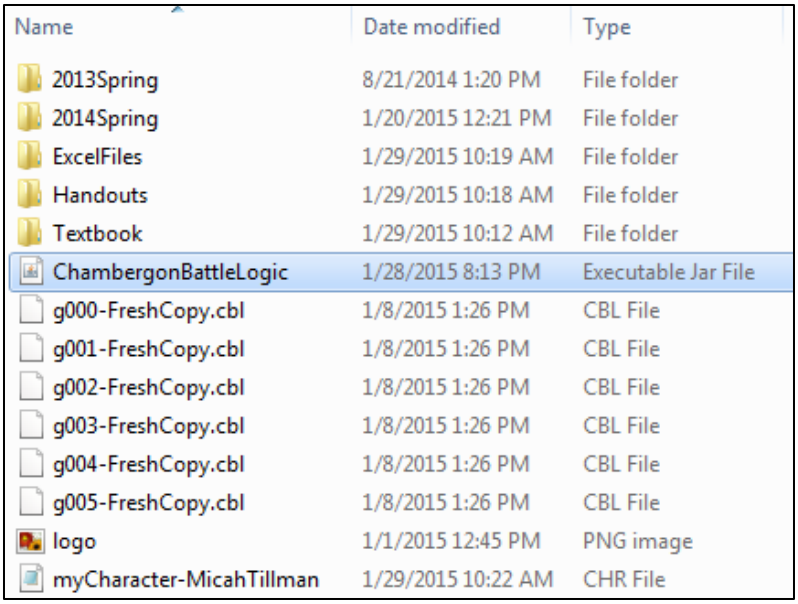
Chapter 0:
Level 0 Powers

0.0: Starting the Game

That was painful. Ugh. So long. I kept typing and typing and typing and it just never ended. Until now. Finally. Let's get to the game.

You'll be playing Chambergon Battle Logic on your computer, although you can play it on paper if you wish. If you play it on paper, it's more like playing Hangman, or whatever that game is where you fold a paper into thirds, and one person draws in the top third, then folds it so the next person can't see, and the next person draws the middle part, and so on. Then you unfold the paper and look at the bizarre thing you've drawn together. Except only you will be doing the drawing, and you'll get to see every step in the game.

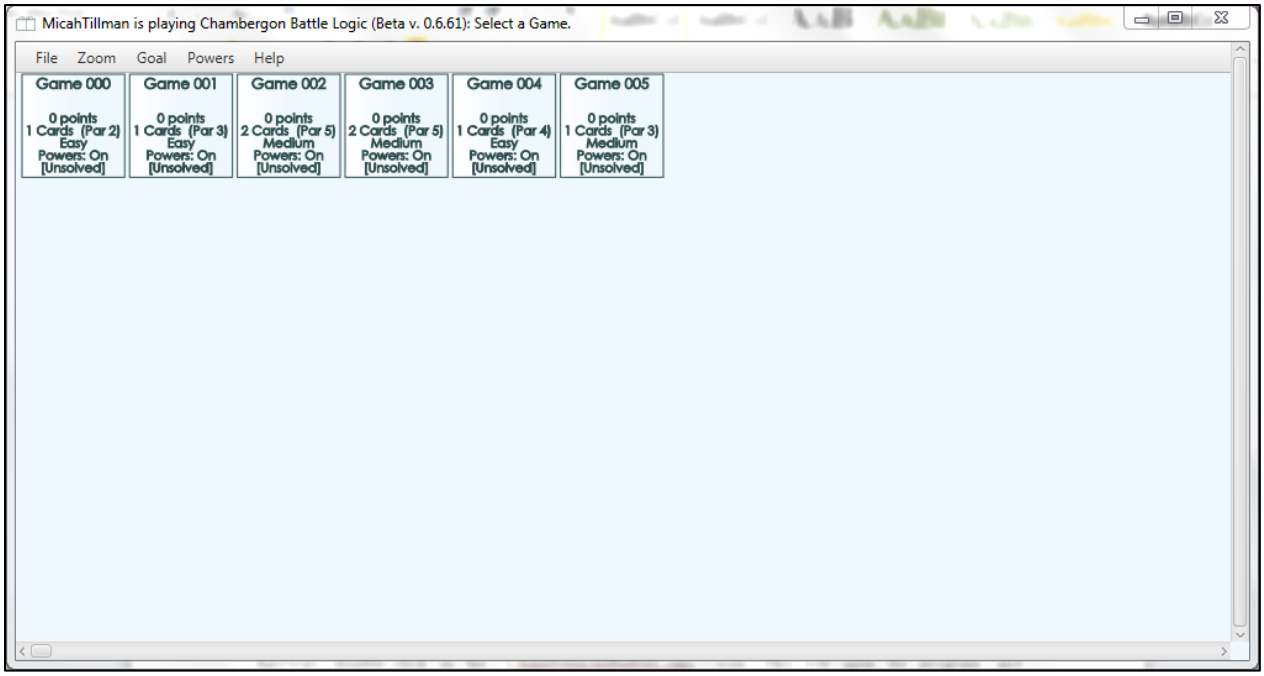
But anyway, let's assume you're playing the game on your computer. For specific instructions about how the program works, see the *Chambergon Battle Logic User's Manual*. In general what you will do is double-click on the "ChambergonBattleLogic.jar" icon to start the program.



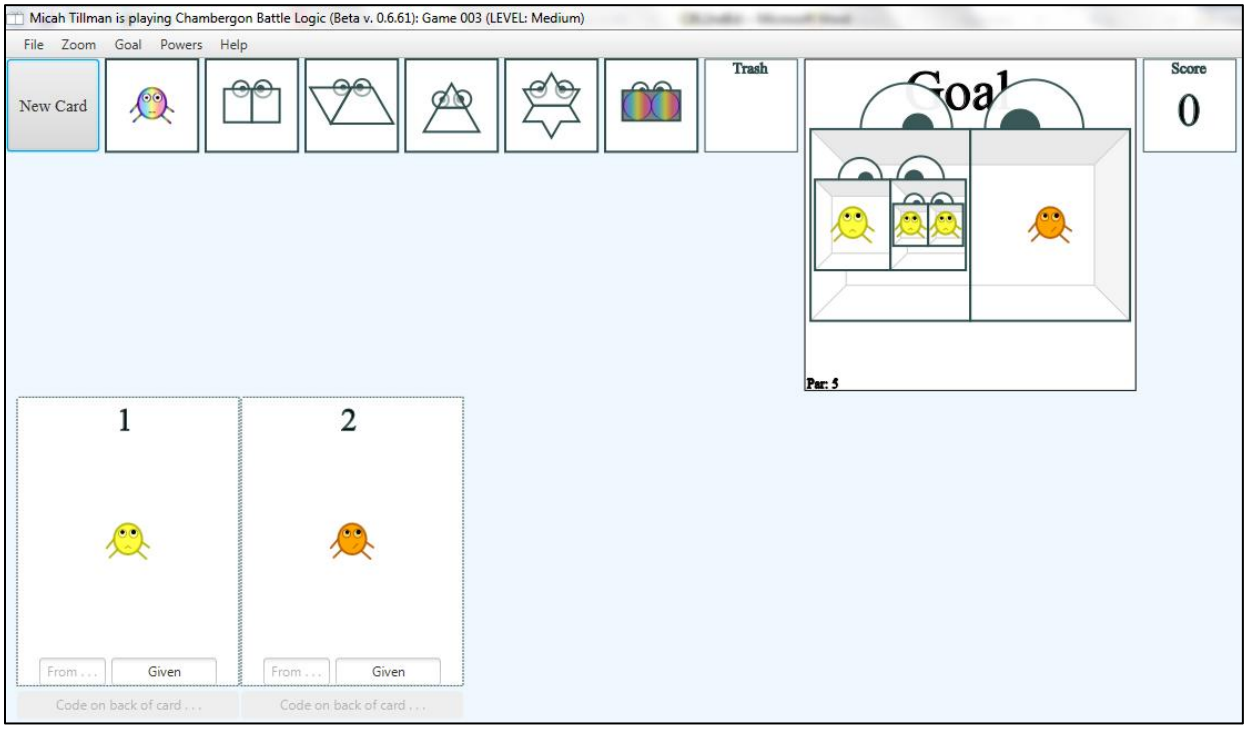
Name	Date modified	Type
2013Spring	8/21/2014 1:20 PM	File folder
2014Spring	1/20/2015 12:21 PM	File folder
ExcelFiles	1/29/2015 10:19 AM	File folder
Handouts	1/29/2015 10:18 AM	File folder
Textbook	1/29/2015 10:12 AM	File folder
ChambergonBattleLogic	1/28/2015 8:13 PM	Executable Jar File
g000-FreshCopy.cbl	1/8/2015 1:26 PM	CBL File
g001-FreshCopy.cbl	1/8/2015 1:26 PM	CBL File
g002-FreshCopy.cbl	1/8/2015 1:26 PM	CBL File
g003-FreshCopy.cbl	1/8/2015 1:26 PM	CBL File
g004-FreshCopy.cbl	1/8/2015 1:26 PM	CBL File
g005-FreshCopy.cbl	1/8/2015 1:26 PM	CBL File
logo	1/1/2015 12:45 PM	PNG image
myCharacter-MicahTillman	1/29/2015 10:22 AM	CHR File

Notice in that picture that the "ChambergonBattleLogic" file is in the same folder as five game files, a file called "logo," and a file called "myCharacter-MicahTillman." You will get your own copies of all these files from Blackboard, and will need to make sure that you download them all into the same folder. It doesn't matter what folder. It just has to be the same folder.

Anyway, double-click on the "ChambergonBattleLogic" icon. This will open the program, and you should see something like this:



If you double-click on one of the game icons, it will open the game for you, and you will see something like the picture on the next page:

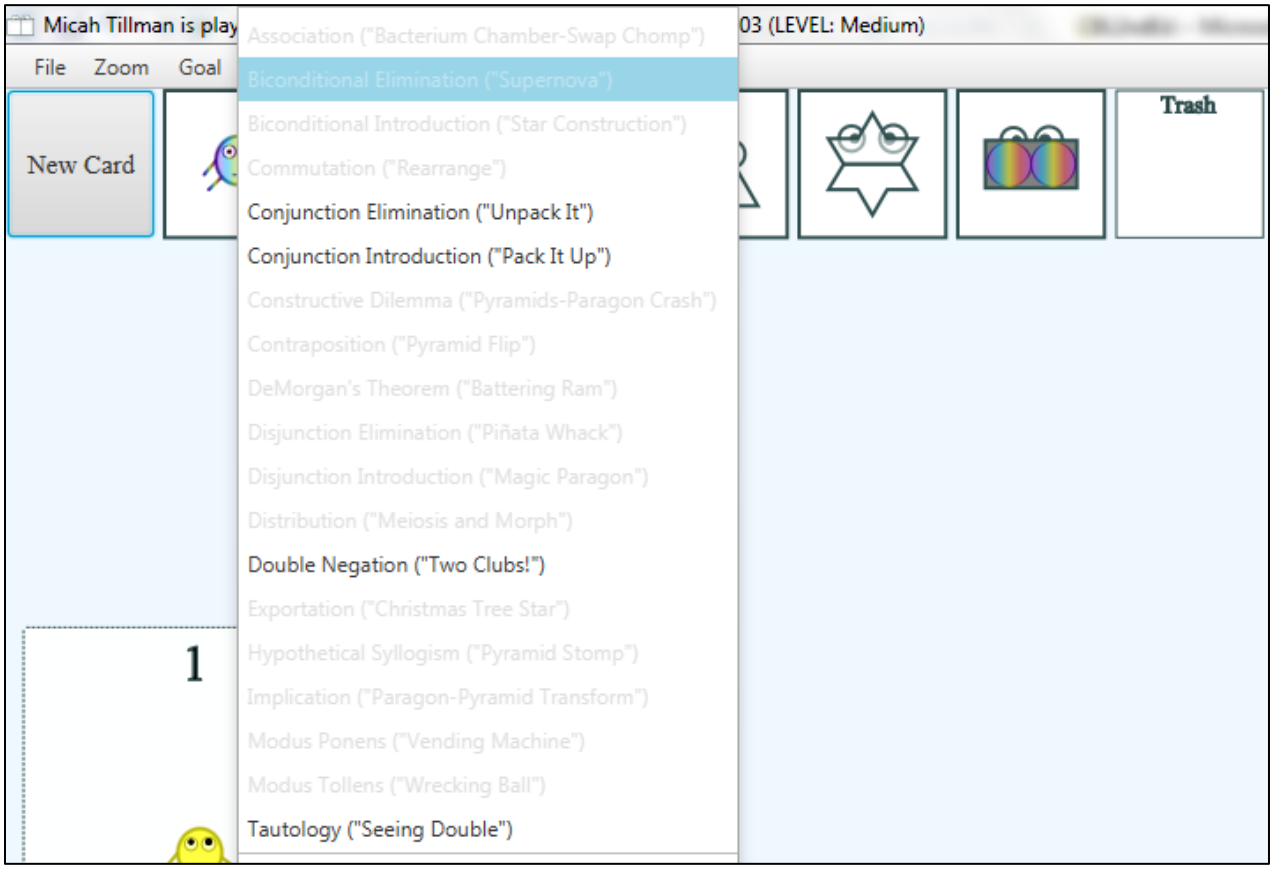


In this game, you have been given two Cards to start. Each Card contains on Character. Your goal is to use the contents of Cards 1 and 2 to produce the complex Chambergon-Character combination in the Goal box. If you look in the bottom left-hand corner of the Goal box, you will see, “Par: 5.” That means it should only take you five Cards total to get from the contents of

Cards 1 and 2 to the complex icon in the Goal box. Since you've already been given two Cards, that means you have to reach the Goal by adding only three more cards.

There are two ways to go from here. You can either click on the "New Card" button, and then drag-and-drop icons into the new card. (The boxes across the top of the screen will give you and infinite supply of new icons.) Or you can use the "Powers" menu.

If you click on the "menu," you'll see something like the picture on the following page:



If you are at Level 0, only four powers are available to you. The rest are "greyed-out," and you have to unlock them by leveling up. If you click on one of the available powers, a little instruction window will pop up, telling you how to use the power on the Cards that you already have. If you can actually use the power on the Cards you already have—you'll find out!—the power will produce a new Card, with a new icon in it. You can then use the same power again—or a different power—on the new Card, to produce another Card, and so on, till you reach the Goal.

As I said above, when you are at Level 0, you only have four powers. You can see them in the image above: "Conjunction Elimination," "Conjunction Introduction," "Double Negation," and "Tautology." If you read the horrible Part 0 of this chapter, you know that "Conjunction Elimination" and "Conjunction Introduction" must have something to do with Parcels, since Parcels stand for the "&&" symbol, which stands for conjunction. Likewise, if you read the

terrible Part 0 of this chapter, you know that "Double Negation" must have something to do with Clubs, since Clubs stand for the "!" symbol, which stands for negation. (In fact, the menu option "Double Negation" includes the hint: "Two Clubs!") The "Tautology" option, however, shouldn't mean anything to you yet.

What I want to do in the rest of this chapter is to talk about the four Level 0 powers. I want to discuss how they work, and *why* they work.

0.1: Double Negation

0.1.0: Negation as Denial

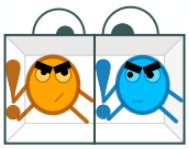
Clubs stand for the "!" symbol, which stands for the logical operation of negation. In negation, you take a proposition and deny it, or claim that it is false. "It is raining" becomes, "It is *not* raining," after you apply the operation of negation. If "R" stands for "It is raining," then "!R" stands for "It is not raining." In icons, that would be:

Now, it is worth asking why the "!R" Character is angry. And it is worth asking *who* the !R Character is angry at. Well, since "R" stands for, "It is raining," and "!R" stands for, "It is not raining," the symbols "R" and "!R" are opposites. They are opposed to each other. "!R" denies that "R" is true. "!R" says that "R" is wrong.

So, who is the red Character with the Club angry at? He's angry at the red Character who has no Club. He couldn't care less about blue Characters, orange Characters, or pink Characters, but he is absolutely *against* red Characters.

Now, take a look at one of the angry Parcels we saw above:

Notice that all three icons in that picture are angry. The orange Character is angry at all orange Characters. After all, an angry orange Character represents "!O," which is the denial "O." The blue Character is angry at all blue Characters. After all, an angry blue Character represents "!B," which is the denial of "B." And the Parcel is angry at all Parcels that contain an angry orange Character and an angry blue Character. After all, an angry Parcel that contains an angry orange Character and an angry blue Character represents "!(!O && !B," which is the denial of "(!O && !B)." In other words, the Parcel above would really like to beat the Parcel below.



That second Parcel—the not-angry one—however, couldn't care less about the angry Parcel above. It's just chillin', doing it's own thing. The angry Parcel *denies* (negates) the not-angry Parcel, but the not-angry Parcel simply asserts itself. (This is because an angry icon represents a negated proposition, which is the denial of a proposition, while a not-angry icon represents a positive proposition, which simply asserts something as true.)

0.1.1: The Motivation for Clubs

It would behoove you to ask at this point, why anger and Clubs go together. Why is it that all the icons that have a Club are angry, and all the ones that don't, aren't? The answer is that Clubs are made of fear, anger, and rage. They are a visible, physical manifestation of those emotions. (If you were drawing a comic book, after all, and wanted to show that someone was making a wordless exclamation of surprise, shock, fear, or anger, you could just draw a word bubble with an "!" in it.)

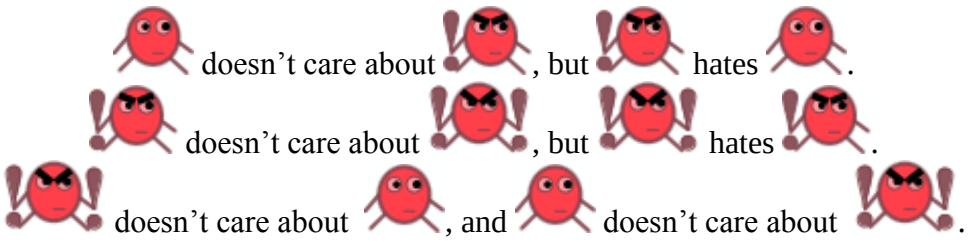
When you have a Club, therefore, the Club fills you with fear, anger, and rage. You might think you are in control of the Club—you might think *you* are holding *it*—but really, it is in control of you.

"But why does that mean you are fearful of, or angry at, other things of your own kind?" you might ask. Why is a blue Character with a Club angry at all blue Characters? Why not be angry at pink Characters or purple Characters?

The answer is that when you are angry, you want to lash out at whatever is nearest to hand. But the Semantic Realm (in the world of meanings, where propositions exist), transcends space. So, things are near or far based on what they are, not where they are. And there is nothing closer to you than something that is identical with you.

When a Chambergon or Character gets a Club, therefore, it is filled with fear and rage and wants to lash out at whatever is closest to it. And what is closest to it are the other Chambergons or Characters that are identical to it, but have one fewer Club. Since they have one fewer Club, they are similar enough to be "close," but are weaker and thus safe to attack.

If a Chambergon or Character has two Clubs, it will only attack another Chambergon or Character that matches it, but has only one Club. If it meets a matching Chambergon or Character that has no Clubs, it will leave it alone.



A doubly-negated proposition is logically equivalent to that same proposition without any negations. So a Chambergon or Character with two Clubs doesn't even notice a matching

Chambergon or Character with no Clubs. It will only go after a matching Chambergon or Character with one Club.

0.1.2: The Power of Double Negation (A Recycling/Equivalence Rule)

The motto for this rule is, "Clubs can be given and taken in pairs." It's like two Clubs were the same as none—like they cancel each other out—or something. This is useful because it means you can give a pair of Clubs to, or take a pair of Clubs from, a Character or Chambergon, whenever you want.

Restrictions: None. You can give a pair of Clubs to any Chambergon or Character anywhere, whether or not it is inside a Chambergon or already has one or more Clubs. Also, you can take a pair of Clubs away from any Chambergon or Character anywhere, so long as it has two or more Clubs to take.

The reason this is logically-valid is that two Clubs represent negating the negation of a proposition. But if you negate the negation of a proposition, the second negation cancels the first negation, leaving the proposition alone. For example, if I write, "It is false that I am not going to the store," the "it is false" part negates the word "not." That leaves us with just, "I am going to the store." So, "It is false that I am not going to the store" is logically equivalent to "I am going to the store." "!!G" = "G."

But if it works in one direction, it works in the other as well. "G" = "!!G." You can take any proposition and negate it twice, without actually changing it. The two negations cancel out, like a negative times a negative making a positive.

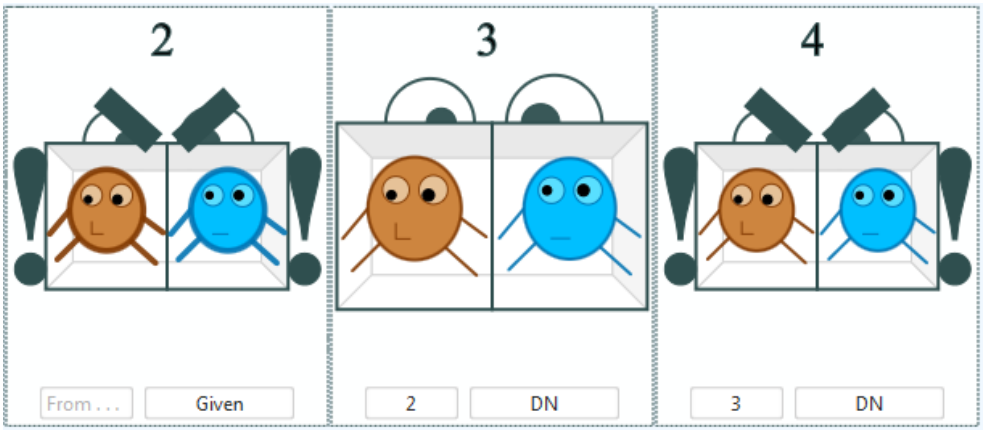
Example 1

In this example, we have three Cards. The Card 2 contains an orange Character, and was simply given. (That is, it was already there when we started the Game.) The Card 3 contains an orange Character with two Clubs. The boxes at the bottom of Card 2 tell us that this Character with two Clubs came from Card 2, using the "DN" rule. ("DN" is an abbreviation for "Double Negation.") This means that we can take the orange Character in Card 2, give it two Clubs, and stick it in

Card 3, if we use the rule of Double Negation ("Clubs can be given and taken in pairs").

Card 4 then looks identical to Card 2, except that it is justified by taking the orange Character from Card 2 and removing its Clubs. This is just the "reverse" application of Double Negation ("Clubs can be given and taken in pairs"). Why would we do this? Who knows? "Who knows?," indeed.

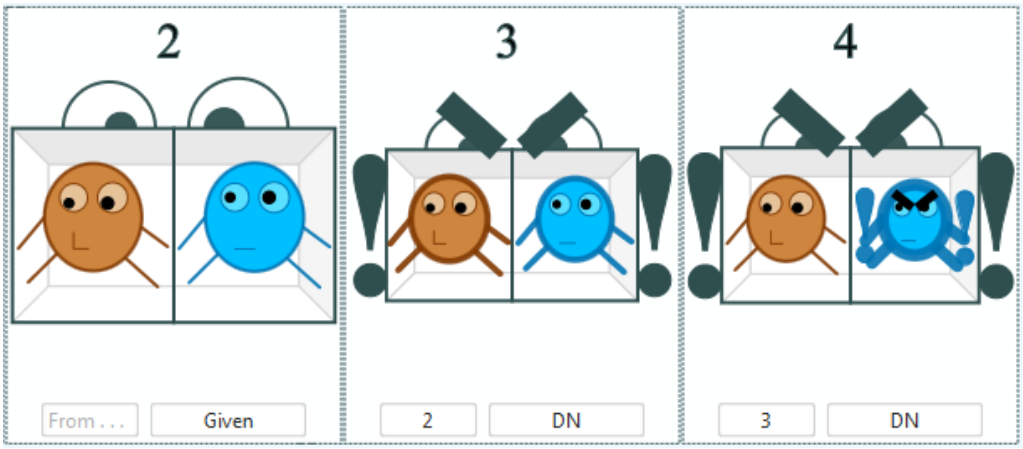
Example 2



In Example 2, we have the same sequence of justifications (look at the boxes at the bottoms of the Cards). However, the contents of the Cards are different. In this example, we started with a Parcel. The Parcel had two Clubs, and contained two Characters. We were then able to copy the Parcel into Card 3 and remove its Clubs, using the "DN" rule (since Clubs can be given and taken in pairs). We were then able to copy the Parcel from Card 3 into Card 4, and give it back its two Clubs, using the same rule.

As with Example 1, there wasn't much point in doing this. We just ended up back where we started. However, it serves to illustrate the fact that the Double Negation rule works "in both directions" (both for giving *and* taking pairs of Clubs).

Example 3



In Example 3, we actually make progress. We start in Card 2 with a Parcel containing two

Characters. This is simply given to us. Then, we copy that Parcel to Card 3, and give it two Clubs, using the rule of Double Negation. Then we copy it from Card 3 to Card 4, and given one of its Characters two Clubs, using the rule of Double Negation again.

The nice thing about the Double Negation rule is not only that it works both for giving and taking pairs of Clubs, but that it works both on the "outermost" Piece in any Card (like the Parcel in Card 3) as well as for any "inner" Pieces (like the blue Character above).

Example 4

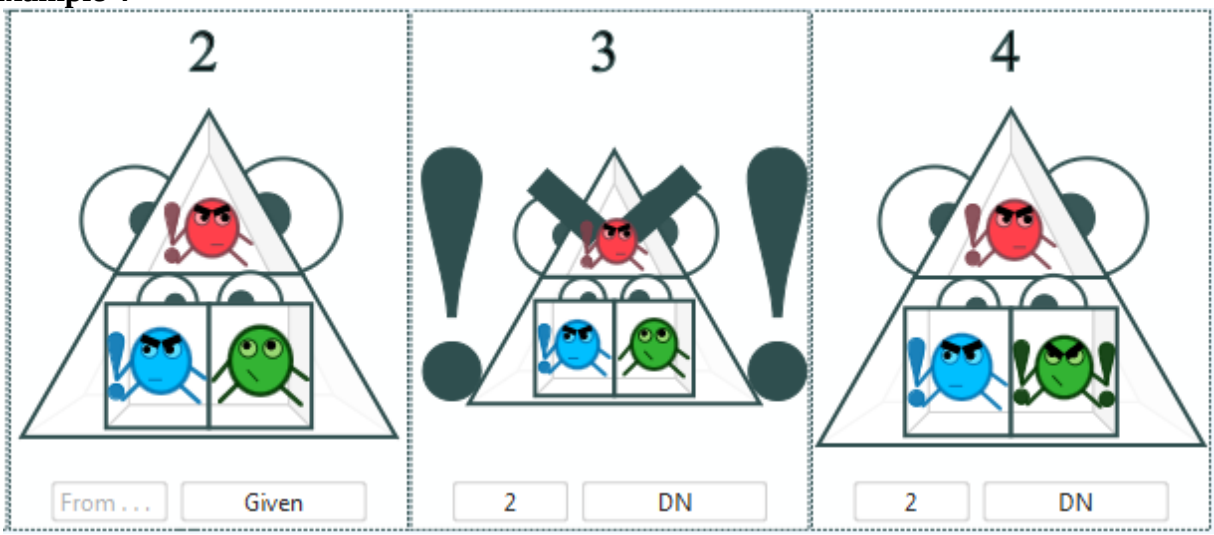


Figure 32

The thing to notice in Example 4 is that Card 4 is obtained not by applying the rule of Double Negation to Card 3, but by going back and applying the rule of Double Negation to Card 2. Cards 3 and 4, in other words, show that you can apply the same rule to the same Card (in this case, applying Double Negation to Card 1) and yet obtain different results. This works because both Cards 3 and 4 were obtained by taking the contents of Card 2 and giving two Clubs. In Card 3, those two Clubs were given to the outermost Pyramid. In Card 4, those two Clubs were given to the "C" Character in the Parcel in the Pyramid's basement.

0.2: The Power of Tautology (A Recycling/Equivalence Rule)

This is a rule for both Parcels and Paragons. The motto for this rule is, "You can go back and forth between seeing double and seeing single." (All you have to do is unfocus and refocus your eyes! Try it now!) What it says is that if you have any Character or Chambergon, you can make two of it and put them in a Parcel, or in a Paragon, together. Likewise, if you have Parcel or Paragon containing two identical Characters or Chambergons—one in each chamber—you can replace the Parcel or Paragon with the contents of one of its chambers.

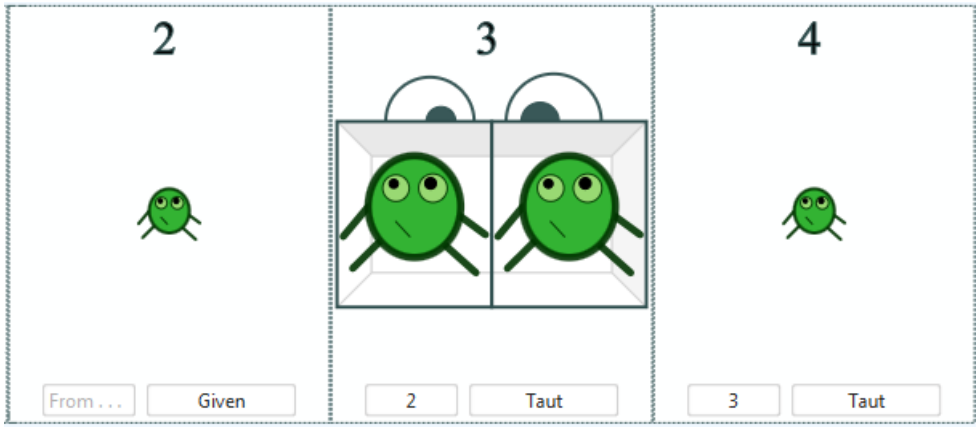
<u>Restrictions</u>: None. You can use this rule on any Character or Chambergon anywhere (even if it is a Character or Chambergon inside a Chambergon).

The reason Tautology is justified is that saying a proposition twice cannot possibly produce a false statement if the original, single proposition was true. "George Washington was America's

1st President" is true. And there's no way to produce a false statement by simply conjoining or disjoining that proposition with itself. "George Washington was America's 1st President and George Washington was America's 1st President" is just as true as "George Washington was America's 1st President." And the disjunction, "George Washington was America's 1st President or George Washington was America's 1st President," must also be true, since "George Washington was America's 1st President" is true.

Likewise, if a proposition is true when you say it twice, saying it only once won't change that. So, if you ever find yourself needing to assert a true proposition twice in a row, you can go ahead and do that (if you join the repeated proposition to itself through conjunction or disjunction). And if you find yourself having a conjunction or disjunction that connects two identical propositions, you can scrap the complicated proposition in favor of its simpler, short version.

Example 1

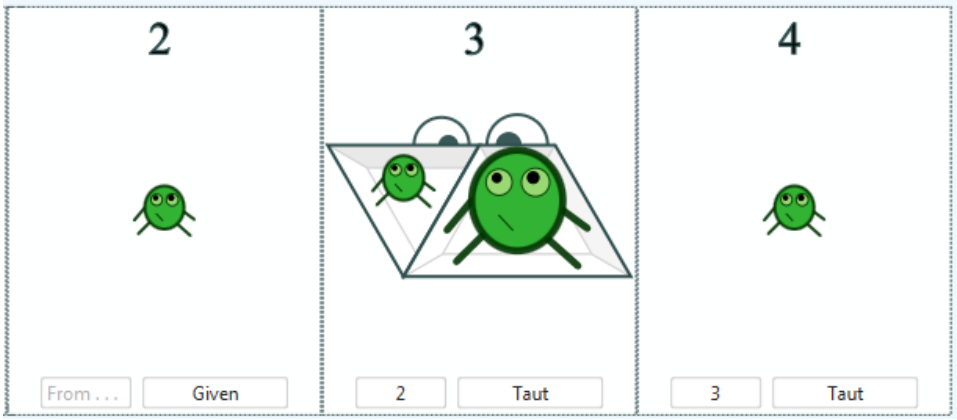


Since we're given a Character in Card 2, we can replace it with a doubled version of itself in Card 3, citing the Tautology rule. And, given that we have a Character in Card 2, we can duplicate it and put it in a Parcel in Card 3. However, if we had been given the Parcel in Card 3 to begin with, we could have replaced it with just a single green Character, as we do when moving from Card 3 to Card 4.

In symbols, Example 1 would look like this:

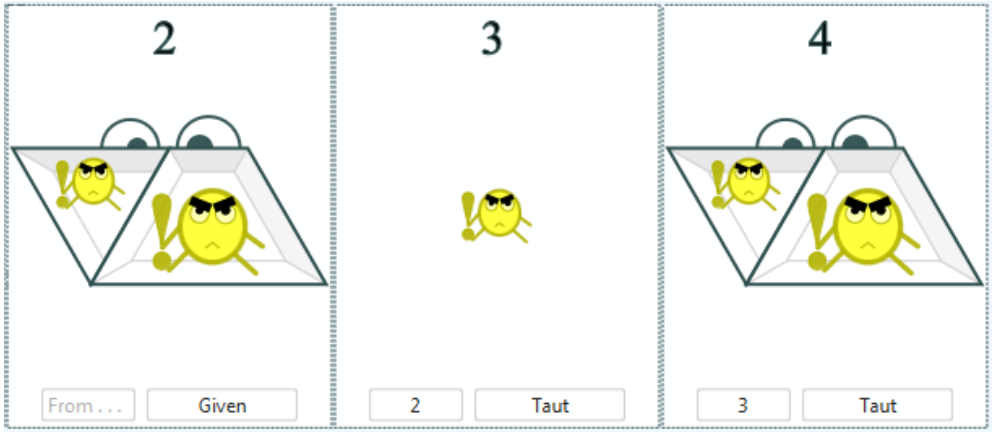
2.	G	Given
3.	(G && G)	2 Taut
4.	G	3 Taut

Note, however, that we didn't have to use Tautology to produce a Parcel ("&&"). We could have produced a Paragon instead.



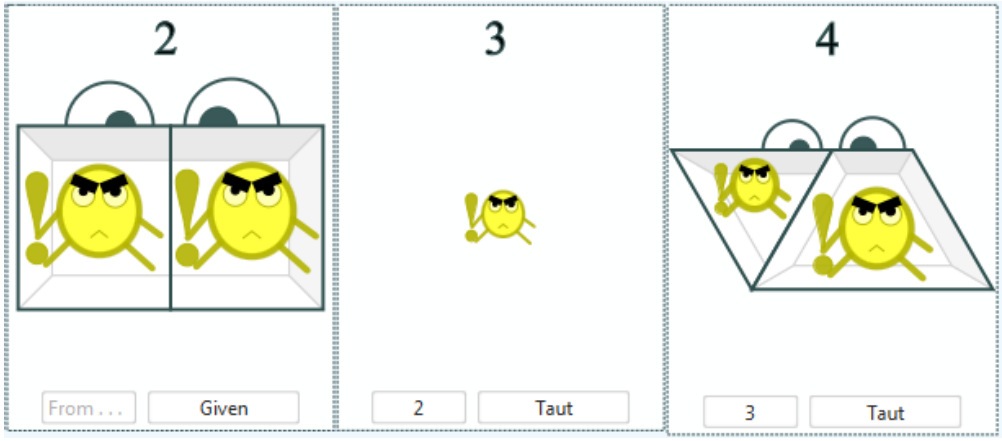
What this means is that from a single Character, you could produce both a Parcel and a Paragon (so long as it contained two identical copies of the Character), and from either a Parcel or a Paragon, you could produce a single Character (so long as that Character was identical to the contents of both chambers of the Parcel or Paragon).

Example 2



Instead of using a Paragon, we could have used a Parcel, like this:

In fact, we could mix the two together:

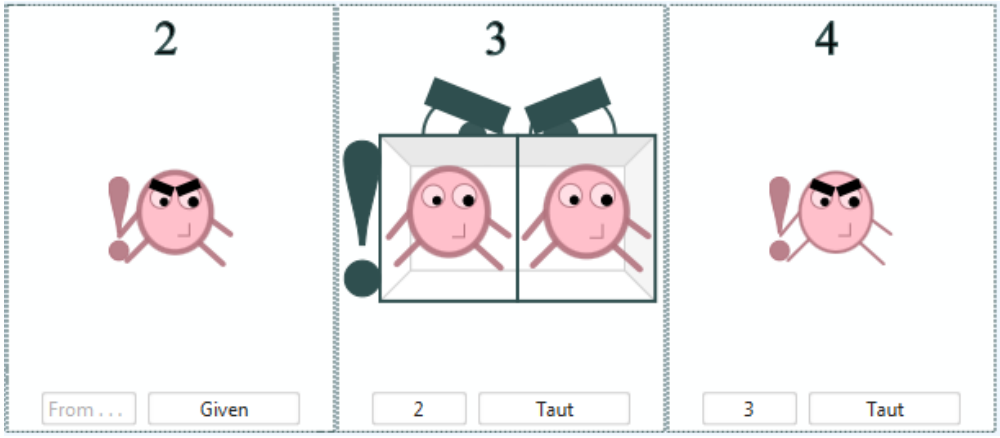


That last version of the example would look like this in symbols:

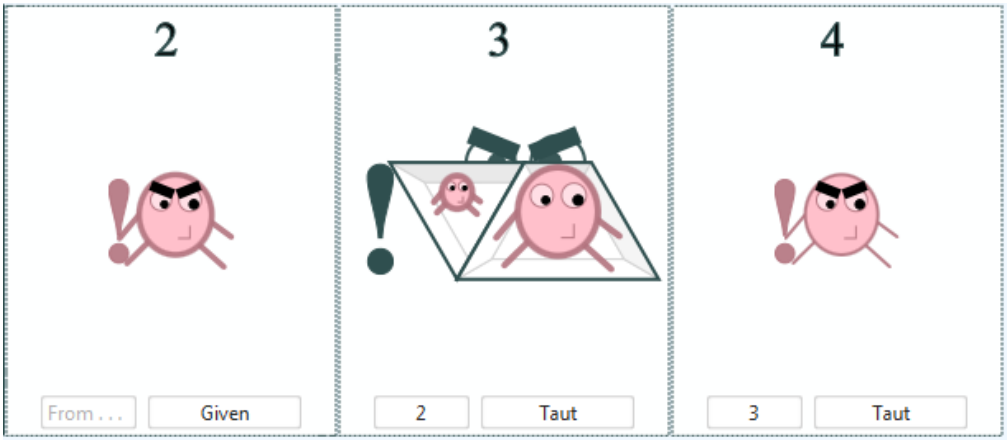
2. (!Y && !Y) Given
3. !Y 2 Taut
4. (!Y || !Y) 3 Taut

The principle here is the same as before. The yellow Character(s) has (have) a Club ("!Y"), and we start with the Character "doubled" in Card 2. But both of those facts are irrelevant. All that is relevant in Card 3 is that the yellow Character with a Club is identical to the contents of both chambers of the Paragon or Parcel in Card 2. They're all yellow Characters with exactly one Club each. And the same is true of the move from Card 3 to Card 4. However, the introduction of a Club does make things potentially tricky. (See the Example 3.)

Example 3



Example 3 is rather odd, in that it duplicates the pink Character from Card 2, but not its Club. Instead, it leaves the Club outside the resultant Parcel. We could do the same thing with a Paragon, however, as follows:



The version with the Parcel would look like this in symbols:

2.	!Z	Given
3.	!(Z && Z)	2 Taut
4.	!Z	3 Taut

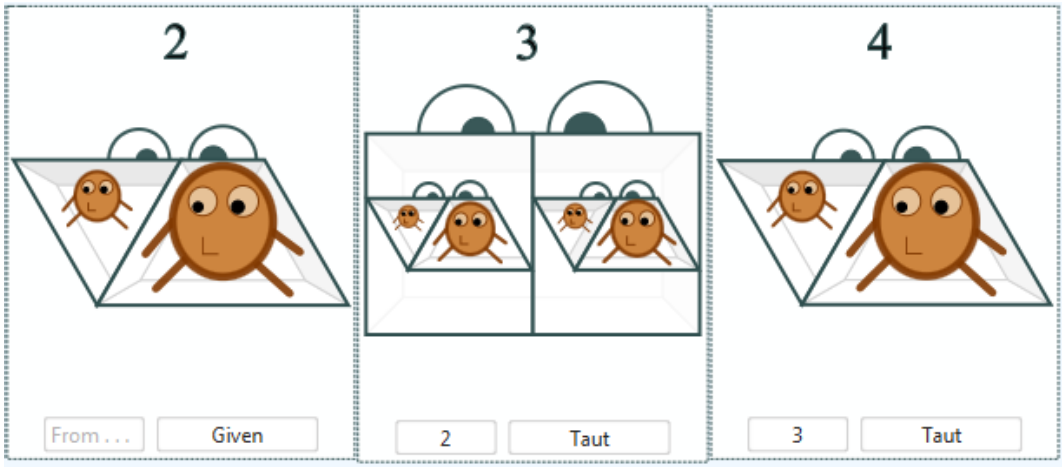
The version with the Paragon would look like this:

2.	!Z	Given
3.	!(Z || Z)	2 Taut
4.	!Z	3 Taut

This is justified by the Tautology rule. After all, you start with a negated proposition, and that negation is still there in Card 3, even though you're now "saying" the proposition twice. It is as if you went from "!(Z)" to "!(Z && Z)." The change is happening "inside the parentheses."

Furthermore, when you replace the Parcel or Paragon in Card 3 with the pink Character in Card 4, the Club is not replaced. It remains, and now attaches itself to the Character. It's as if it had been patiently waiting, outside the Parcel or Paragon, for the Character to return.

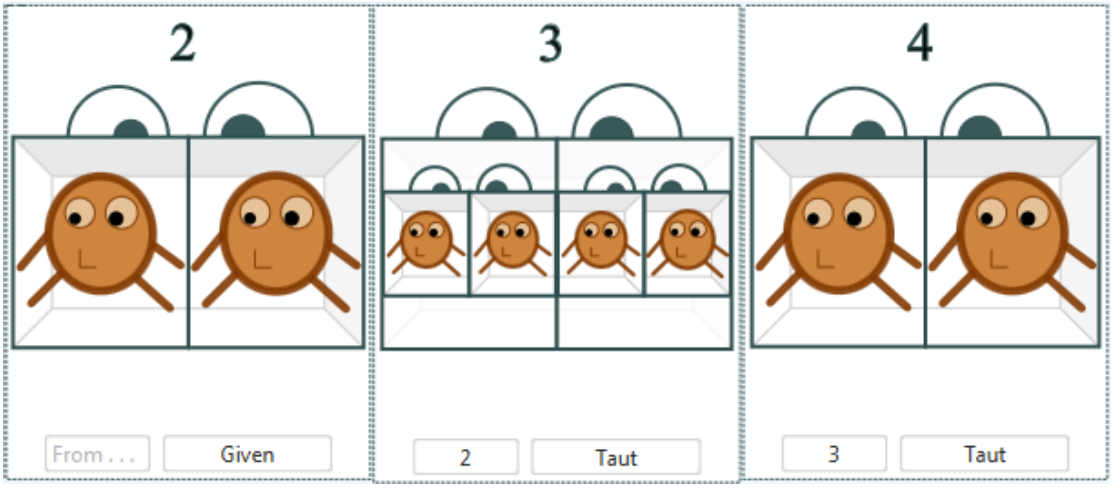
Example 4



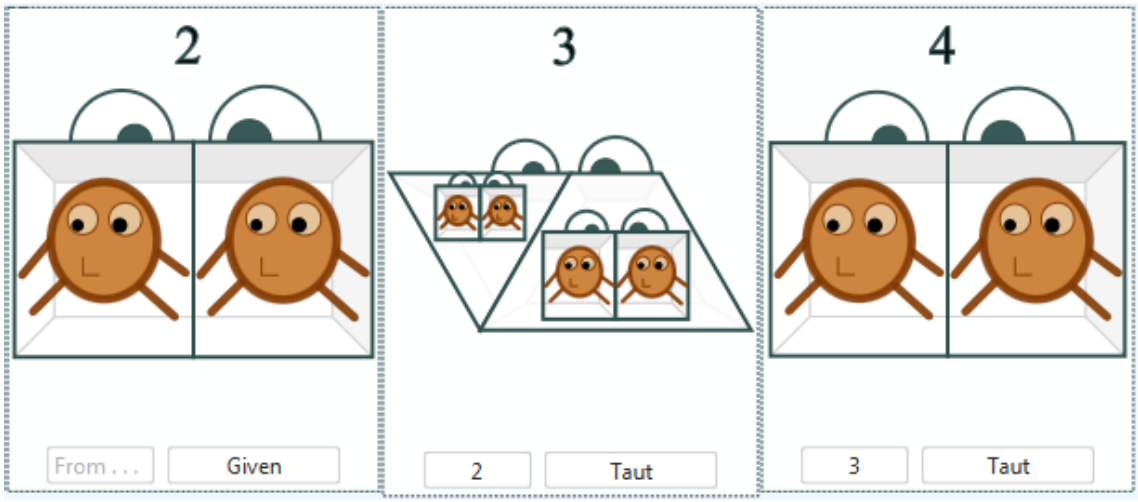
Example 4 is the same as Examples 1 and 2, believe it or not. We took a Paragon in Card 1, duplicated it in Card 2, and packaged both Paragons in a single Parcel. This is justified by the "Tautology" rule, as is the collapsing of both Paragons back into a single Paragon in Card 3. In symbols, it would look like this:

2.	(A || A)	Given
3.	((A || A) && (A || A))	2 Taut
4.	(A || A)	3 Taut

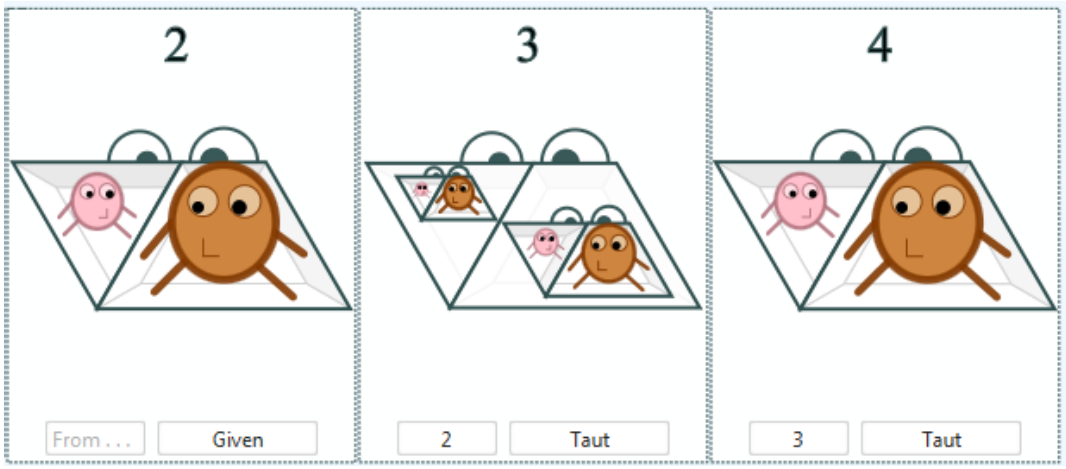
And if we wanted to, we could've done any number of different combinations:



Or:

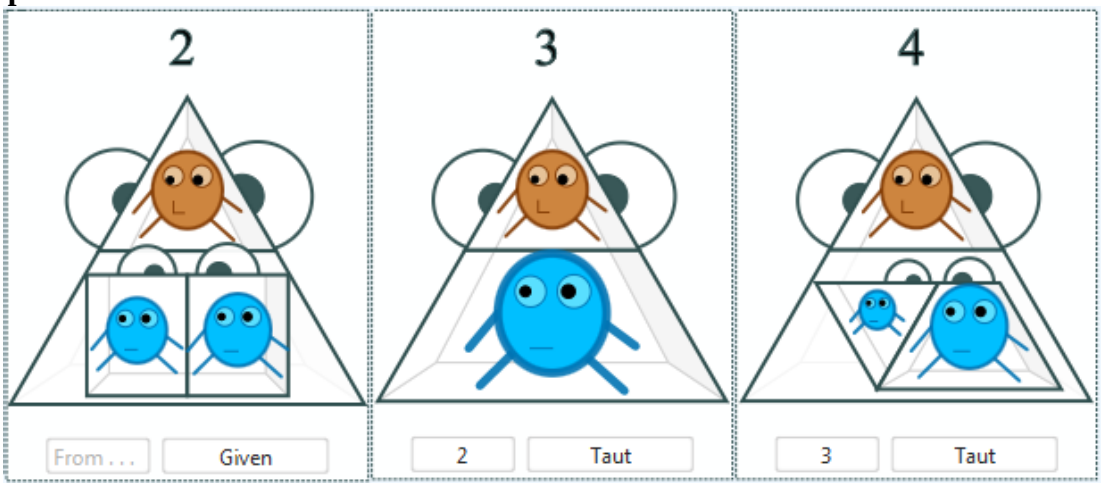


And the colors don't have to all be the same, either:



The reason these examples are justified by Tautology is that (1) a single thing is being duplicated and placed inside the chambers of a new Paragon or Parcel, or (2) a Paragon/Parcel whose contents exactly match each other is being replaced by a single copy of whatever it contains. It just so happens that in these instances, the "single thing" is itself a Chambergon containing multiple things.

Example 5



Here, in Card 2, we noticed that there was a Parcel, both of whose chambers had identical contents. We then replaced that Parcel in Card 3 with one copy of the duplicate contents. This is justified by Tautology, since we've just gone from saying something twice (using a conjunction), to saying it once. Likewise, the move from Card 3 to Card 4 is justified by the Tautology rule, since we are simply going from saying something once to saying it twice (using a disjunction). We could've duplicated the brown Character in the attic, and replaced it with a Parcel or Paragon containing two brown Characters. But we didn't. Because we didn't have to. We can do what we want. You can't tell us what to do.

Example 5 would look like this, in symbols, if you were wondering:

2. (A -> (B && B)) Given
3. (A -> B) 2 Taut
4. (A -> (B || B)) 3 Taut

0.3: Recycling Rules vs. Landfill Rules

The Double Negation and Tautology rules (or "powers") are "recycling" rules. You can undo their results simply by applying them again. And since "recycling" rules are "equivalence" rules, you can apply them to any Character or Chambergon *anywhere*. You don't have to apply them to the outermost Chambergon in the Card. You can, but you can also apply them to things that are buried deep inside a Chambergon.

Not all rules are like this, however. Some, like the Conjunction Elimination and Conjunction Introduction rules (which we will do next), are "Landfill" rules. They only work in one direction, so once you use one of these powers, there's no going back. You have to throw it in the landfill. (Well, that's actually a lie; in some cases you can go back, but to do so requires using a *different* power. You can't just use the same power all over again.)

Landfill rules (or "inference rules"), furthermore, can only be applied to whatever is the outermost icon in a Card. They can't be applied to icons that are inside of other icons. The reason for this is that when you use a landfill power, you are not just saying the equivalent of what you said previously. You are actually making an inference, and thus are saying something new. But you can't take a true proposition, changing one of its parts to something new, and still expect the resulting proposition to still be true. It might be, but there are no guarantees. So, landfill rules—rules that effectively introduce something new instead of recycling something old—can only be applied to "the proposition as a whole" (i.e., to the outermost icon/piece in a Card).

To see what I mean, take a look at the next two rules.

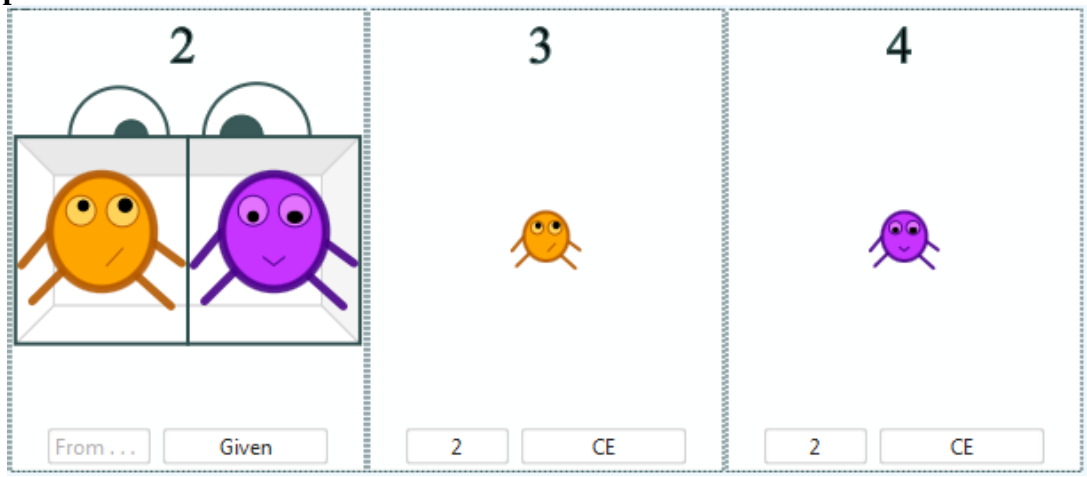
0.4: The Power of Conjunction Elimination (A Landfill/Inference Rule)

The motto for this rule is, "It's Always Christmas!," or "You can open a Package whenever you want." When you open a Parcel, you get to take out one thing. If you want to take out both, you have to use two Cards (one for each chamber you empty).

<u>Restrictions:</u> This only works for Parcels that are not inside other Chambergons, and that do not have a Club. If the Parcel is inside another Chambergon, the other Chambergon holds it shut. And if the Parcel has a Club, it will fight off any attempt to open the Parcel. Furthermore, you can only take one thing out of a Parcel per Card.

The logical justification for this rule is as follows. If you make a conjunctive statement, you are asserting that both one thing and another thing are true. But if both are true, it's perfectly acceptable for you to assert that either one is true individually. For example, if I say, "I'm happy and I'm going to have a snack," I've claimed that both the proposition, "I'm happy," and the proposition, "I'm going to have a snack" are true. But if they're both true, I am justified in saying that either one is true, all by itself. I could just say, "I'm going to have a snack." And since the more complex conjunction, "I'm happy and I'm going to have a snack," is true, the simpler, "I'm going to have a snack" must be true too.

Example 1

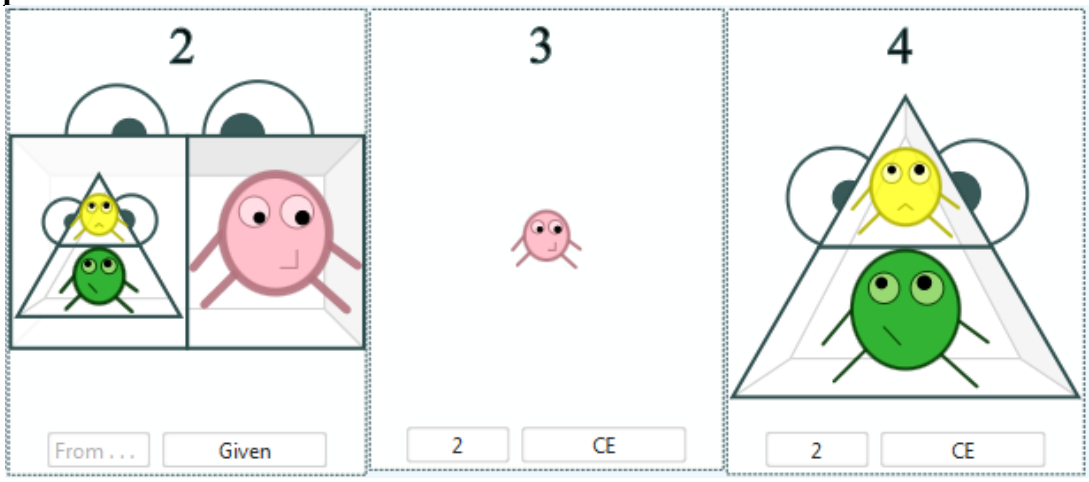


In Example 1, we are given a Parcel in Card 2. Since the "Conjunction Elimination" (abbreviated "CE") rule says that it's always Christmas, we can open the Parcel whenever we want. We empty its first chamber into Card 1, and its second chamber into Card 2. We didn't have to empty both chambers if we didn't want to, and we could have emptied chamber 2 first. But that's how we did it this time. (Notice that the justification for Card 4 is "2 CE," since the purple Character comes from the Parcel in Card 2, rather than coming from Card 3.)

In symbols, Example 1 would look like this:

2.	(O && P)	Given
3.	O	2 CE
4.	P	2 CE

Example 2

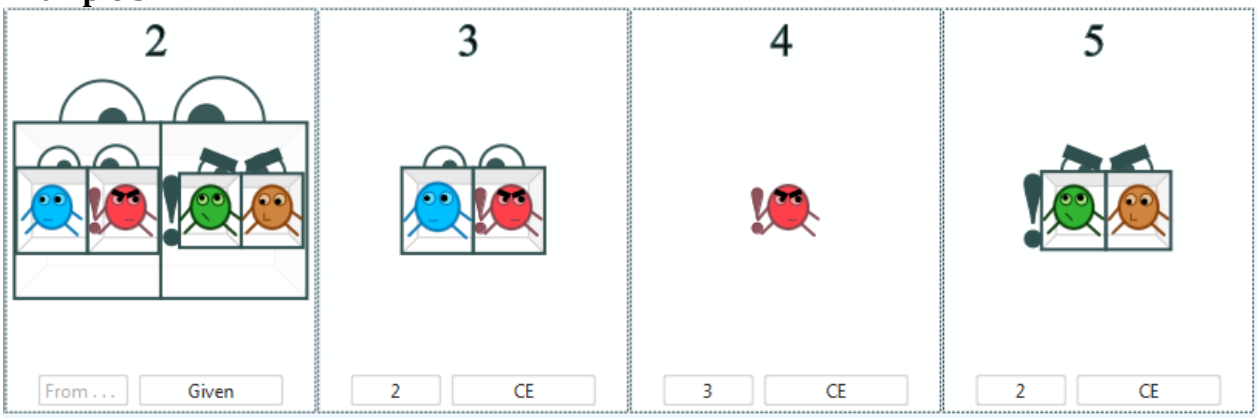


In Example 2, we use the "Conjunction Elimination" (or "Simplification") rule to unpack a more complex Parcel. We empty its second chamber first, producing Card 3, and then its first chamber, producing Card 4.

In symbols, Example 2 would look like this:

2.	((Y -> G) && Z)	Given
3.	Z	2 CE
4.	(Y -> G)	2 CE

Example 3



We have three Parcels in Card 2. Conjunction Elimination allows us to open Parcels and take out their contents. However, Conjunction Elimination is a landfill rule, which means it can only be applied to the outermost icon in a card. If we want to take out the red Character, therefore, we have to apply "CE" twice. First, we apply it to open the Parcel in Card 2 and take out the contents of the left chamber. We place those contents in Card 3. This makes the Parcel containing the red Character the outermost icon in a Card, and thus we can apply CE to it to extract the red Character.

However, while we can take the Parcel with the Club out of the Parcel in Card 2, and put it in another Card (as we do in Card 5), that's as far as we can go with the contents of the right chamber of the Parcel in Card 2. The Parcel containing the green and brown Characters has a Club. If we tried to open it and take out its contents, it would smack our hands with its Club.

Here's what Example 3 would look like in code (i.e., in symbols):

2.	((B && !R) && !(G && A))	Given
3.	(B && !R)	2 CE
4.	!R	3 CE
5.	!(G && A)	2 CE

The reason you can't open a Parcel that has a Club is that a Parcel with a Club is a negated conjunction. It says that a conjunction (e.g., "G && A") is false. But there are three ways in which a conjunction could end up being false. The first of its constituent propositions ("G") might be false, while the second ("A") is true. The second of its propositions ("A") might be false, while the first ("G") is true. Or both might be false. We just don't know. So, we can't just

take the contents out of a Parcel that has a Club, and then "assert" those contents to be true by themselves.

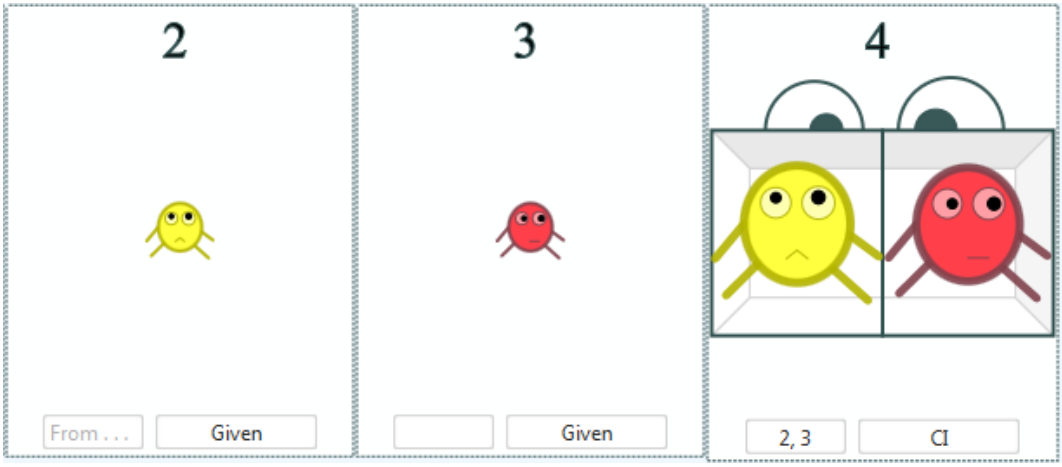
0.5: The Power of Conjunction Introduction (A Landfill/Inference Rule)

The motto of this rule is, "You can pack things up whenever you want." It's the opposite of Conjunction Elimination/Simplification. If you take the contents of any two Cards (and you must take *all* the contents of both Cards)—whether they be Characters or Chambergons—you can put them in a Parcel. And what's especially nice is, you can put them in, in any order. Just because one is from an earlier Card than the other doesn't mean you have to put it in the "first" (that is, the left) chamber. You can put it in the right chamber if you want.

Restrictions: The two things you're packing into the Parcel must actually be available. That is, they must be in a Card, but not in any Chambergon inside their Cards.

The logical justification for this rule is that if two propositions are true independently, then the assertion that they are both true must also be true. For example, if it is true that it is raining, and it is true that chocolate is tasty, then it must be true that "It is raining *and* chocolate is tasty."

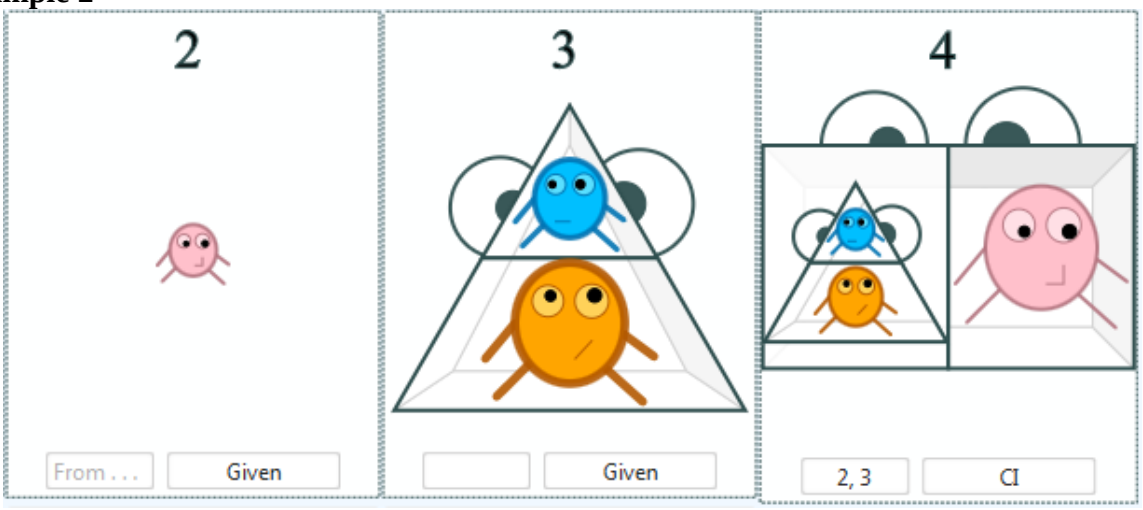
Example 1



There are two things to pay attention to in Example 1. The first is the visual similarity between Cards and the chambers of a Parcel. In a way, two Cards placed side-by-side, resemble a Parcel containing the contents of the two Cards. It is natural, therefore, to simply make this implicit image explicit, by packing the contents of Cards 2 and 3 into a Parcel in Card 4. (The Parcel in Card 4, you might say, is just a picture of Cards 1 and 2, "zoomed out.")

The second thing to pay attention to in Example 1 is the fact that Card 4 is justified by "2, 3 CI." That is, the justification for Card 3 cites two Cards, rather than one, since the contents of its Parcel come from two different Cards. Whereas you could get two separate Cards worth of content by applying Conjunction Elimination, it takes two separate Cards worth of content to be able to use Conjunction Introduction.

Example 2



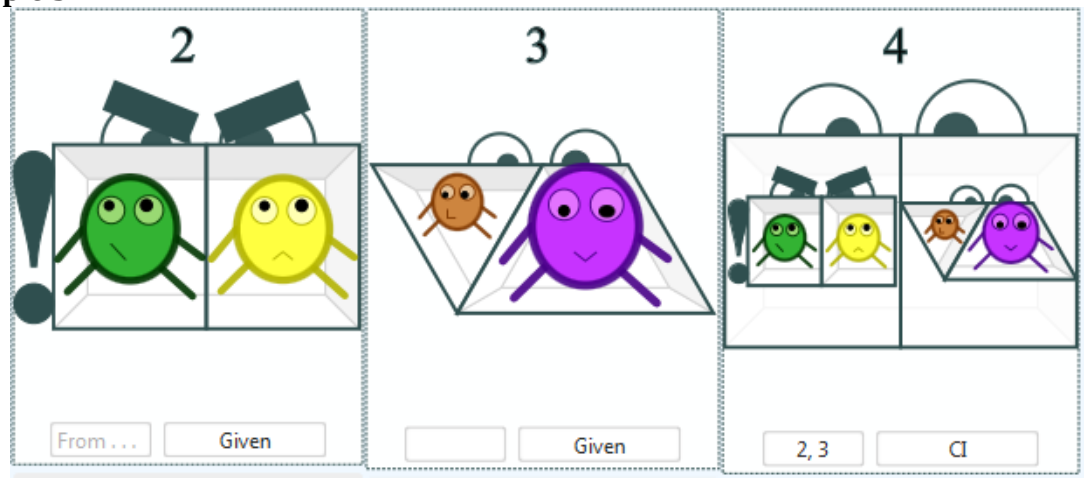
The message of Example 2 is that (a) not only can Characters be packed away using the Conjunction Introduction rule, but Chambergons can as well, and (b) the Conjunction Introduction rule allows us to pack things away in reverse order. Thus, though the Pyramid comes in Card 3, and the pink Character in Card 2, the Pyramid is packed into chamber 1 of the Parcel. We could have put it in chamber 2, but we decided to switch things up.

In symbols, Example 2 would look like this:

2.	Z	Given
3.	(B –> O)	Given
4.	(Z && (B –> O))	2, 3 CI

Notice, once again, that the justification for Card 4 (line 4) cites both Cards 2 and 3 (lines 2 and 3).

Example 3



The first purpose of Example 3 is to remind us that if the contents of a Card have Clubs, those Clubs must be packed away along with the contents, if we are using the Conjunction Introduction rule. The second purpose of Example 3 is to remind us that anything can be packed up into a Parcel using CI. It doesn't matter if the things being packed up are themselves Parcels, if they have Clubs, or if they are some other Chambergon or Character. Anything can be put in a box. This is especially important to remember, since Conjunction *Elimination* can*not* be applied to anything but Parcels (i.e., it will only allow you to unpack a Parcel), and it can't be applied to Parcels that have Clubs (i.e., if a Parcel has a Club, you can't open it to take out its contents).

0.6: Conclusion

Done! We finally finished the chapter! Did you think it was never going to end? I sure thought it wasn't. I believed the proposition, "It is false that the chapter will end," which I will symbolize as "!E." (Actually, I didn't believe that it is false that the chapter would end. So, "!!E.")

In this chapter, we have seen examples of how the four rules available to all Level 0 players work. Now, I think, it is time to actually play some games.

Chapter 1: Level 1 Powers

1.0: Only Two Rules This Time

In Chapter 0 we covered four rules. Two of them—Double Negation ("DN") and Tautology ("Taut")—were recycling rules (or, if we're being technical, "equivalence rules"). They can be applied to any piece, anywhere, in any game at any time—and if you don't like the result all you have to do is apply the same rule again to get back to where you started.

The other two rules in Chapter 0—Conjunction Elimination ("CE") and Conjunction Introduction ("CI")—where landfill rules (or, if we're being technical, "inference rules"). They can only be applied to icons that are the "outermost" in their Cards. You can't apply them to Chambergon that's inside another Chambergon. And you can't get back to where you started just by applying them again. To "get back to where you started," you'd have to apply the other rule (CI if you used CE; CE if you used CI)

So, Chapter 0 was split half-and-half between recycling rules and landfill rules. In this chapter, the same thing will happen. The difference is that in this chapter, we only have to cover two rules. DeMorgan's Theorem is a recycling rule—when you apply it, you don't change anything. You just say the same thing in a slightly different form. Disjunction Elimination, however, is a landfill rule. By applying it, you get something new, and have, as it were, thrown the old away. (Though you can always go back and dig it out of the trash if you really want it.)

1.1: De Morgan's Theorem

1.1.0: Who Was De Morgan?

Some guy.

Okay, what? You want me to go look him up on Wikipedia?

Fine. Wait a second.

I'm back. First, I can tell you De Morgan's full name was "Augustus De Morgan," and that—despite having such an un-British name—he was British. He was born in the early 1800s and died in the late 1800s. This made him a contemporary of George Boole, one of the other important founders of symbolic logic. In fact, it would appear from their Wikipedia entries, that De Morgan and Boole were friends.[22] Encyclopedia Britannica, in fact, says this:

> The renaissance of logic studies, which began in the first half of the 19th century, came about almost entirely because of the writings of De Morgan and another British mathematician, George Boole.[23]

---

[22] "Augustus De Morgan," *Wikipedia*, http://en.wikipedia.org/wiki/Augustus_De_Morgan; "George Boole," *Wikipedia*, http://en.wikipedia.org/wiki/George_Boole.

[23] "Augustus De Morgan," *Encyclopedia Britannica*, http://www.britannica.com/EBchecked/topic/153815/Augustus-De-Morgan.

In other words, De Morgan is one of the two people who started modern symbolic logic, and thus is one of the two people who are responsible for the fact that you have to take this course.

But De Morgan's impact on your life hasn't been all bad. You know how you can write a fraction like this—"1/2"—instead of like this—"$\frac{1}{2}$"? That was De Morgan's idea.[24]

1.1.1: Why Is De Morgan's Theorem Important?

De Morgan's Theorem (or "De Morgan's Law") shows that conjunction and disjunction are the same operation, if you also have access to negation. If you've got a conjunction, you can turn it into a disjunction with negation. If you've got a disjunction, you can turn it into conjunction with negation.

In other words, while conjunction and disjunction look like different ways of combining two propositions into a larger proposition, they are actually two different expressions of a more fundamental operation. This operation can appear as either conjunction (and negation), or as disjunction (and negation), but it is, strictly speaking, neither.

The idea that you can get one operation from another if you just mix in a little negation also occurs in math, if you think about it. Subtraction, after all, is equivalent to adding a negative number. And addition is equivalent to subtracting a negative number. You can get subtraction from addition if you have access to negation (I mean, "to negatives"), and you can get addition from subtraction if you have access to negatives.

Perhaps more interestingly for us (and by "us," I mean, "me"), is this: the fact that conjunction is, in some sense, disjunction plus negation—and that disjunction is, in some sense, conjunction plus negation—shows once again that negation is a different kind of thing from conjunction and disjunction. While conjunction and negation can make disjunction, and disjunction and negation can make conjunction, disjunction and conjunction cannot make negation. Negation is not derivable or derivative, while conjunction and disjunction both are.

1.1.2: The Power of De Morgan's Theorem (A Recycling/Equivalence Rule)

The motto of this rule is, "Clubs are like battering rams." It's a rule that works for both Parcels and Paragons. If the Parcel or Paragon has a Club, you can move it through the wall of the Chambergon. But when you do this, two things happen. First, the Club shatters into two pieces, one of which goes to the left chamber, and the other of which goes to the right chamber. Second, the walls of the Chambergon are shifted; if it's a Parcel, it tilts into a Paragon, and if it's a Paragon, it is straightened back up into a Parcel.

Now, the process also works in reverse. If the contents of the two chambers in either a Parcel or a Paragon have at least one Club, you can combine one Club from each chamber into a larger Club for the whole Parcel. This reduces the number of Clubs for each of the chambers by one, but increases the number of Clubs the Parcel or Paragon has by one. Furthermore, as this new Club crashes through the walls and out into the open, the walls are shifted, and the Parcel becomes a Paragon, or the Paragon becomes a Parcel.

[24] Ibid.

In De Morgan's Theorem, what happens is that the "fear" or "anger" that Clubs represent is transferred from a Chambergon to its contents, or from a Chambergon's contents to the Chambergon. In essence, De Morgan's Theorem allows Pieces in the game to let other pieces worry for them. And, after all, there's not much difference between the whole Chambergon being worried or angry, and everything in it being worried or angry (just like it wouldn't make any sense to say that a group is worried or angry if everyone in the group weren't worried or angry, and it wouldn't make sense to say that everyone in a group was worried or angry, but the group wasn't worried or angry).[25]

So, De Morgan's Theorem allows us to "comfort" by shifting their distress to other Pieces, and to worry or enrage Pieces by asking them to take over the emotions of other Pieces.

<u>Restrictions:</u> This rule works for Parcels and Paragons that are in other Chambergons, and for Parcels and Paragons that are not in other Chambergons. However, either the Parcel or Paragon itself must have at least one Club (in which case that Club can be split and given to both chambers inside), or else the two things in its chambers must both have at least one Club (in which case those two Clubs can be joined into one, and given to the Chambergon as a whole).

The logical justification for De Morgan's Theorem would go something like this. . . .

[**WARNING**: The following contains a lot of symbolism, and will remind you of why I decided to do icons instead of symbols in this book. If you can't handle it, skip the paragraphs with the "Symbolism Warning" in the margin.]

!!!SYMBOLISM WARNING!!!

Take two propositions—like, "Logic is the funnest," and "College courses are so easy"—and conjoin them into the single proposition: "Logic is the funnest and college courses are so easy."

Let's use "F" to stand for "Logic is the funnest," and "E" to stand for "College courses are so easy." If we do that, then "Logic is the funnest and college courses are so easy" is "(F && E)."

!!!SYMBOLISM WARNING!!!

Now, imagine that I assert that proposition, and you tell me that I'm wrong. You say, "'Logic is the funnest and college courses are so easy' *is false*," or "*It is not the case that* logic is the funnest and college courses are so easy." In symbols, that would be "!(F && E)."

But that just means you disagree with the whole conjunction (or, "the conjunction as a whole"). I don't know, based on your denial of the whole, how you feel about the individual parts. Did you negate whole proposition because you disagree that logic is the funnest? Or did you negate it because you disagree that college courses are so easy? Do you believe "!F" or "!E"? (Or maybe both?)

The answer is that, until you tell me more, I don't know. Either you think "Logic is the funnest" is wrong, or you think "College courses are so easy" is wrong. So, from the claim, "It is not the case that logic is the funnest and college courses are so easy," I can infer that, "Either 'logic is

[25] The difference, of course, is that a group is not a whole; it is not a single thing. It is a bunch of different things.

!!!SYMBOLISM WARNING!!!

the funnest' is false, or 'college courses are so easy' is false." From a negated conjunction, I conclude a disjunction of negations. From "!(F && E)," I can infer "(!F || !E)." (Do you see how the "!" moves from outside the parentheses, to inside, and the "&&" changes to "||"?)

Now, imagine that you come up to me and say, "We need both Francine and Eddie to come through for us tonight. The problem is getting Francine to fly. Eddie *will* fly—he always does—but you know how forgetful he is. He'll forget to eat before flying, and will pass out from hunger mid-trip. I've just got a bad feeling about the whole thing. Either Francine won't fly tonight or Eddie won't eat. This is going to be a disaster." If we use "F" for "Francine will fly tonight," then "Francine *won't* fly tonight" would be "!F." And if we use "E" for "Eddie will eat," then "Eddie *won't* eat" is "!E." Therefore, "Either Francine won't fly tonight or Eddie won't eat," is "(!F || !E)."

From what you told me, you think tonight is going to be a disaster because we need "(F && E)" to be true (we need Francine to fly and Eddie to eat), but we know that "(!F || !E)" (either Francine won't fly or Eddie won't eat). So, that means "(F && E)" isn't going to happen. At least one of the two—Francine or Eddie—is going to fail us, which means "(F && E)" isn't going to be true. In other words, from "(!F || !E)," we can conclude "!(F && E)." (Notice how the exclamation marks inside the parentheses move to the outside, and the "||" changes to "&&.")

What we have just seen is two examples of DeMorgan's Theorem. The first showed us that "!(F && E)" entails "(!F || !E)." The second showed us that "(!F || !E)" entails "!(F && E)." And that means:

!(F && E) -> (!F || !E)

and

(!F || !E) -> !(F && E)

It goes both ways, which means we can just used a double-headed arrow, like this:

!(F && E) <-> (!F || !E)

But that's just one half of De Morgan's Theorem. The rule also works if we start or end with a negated disjunction. Imagine, for instance, that I say it will either be frigid or epic tomorrow, but you disagree. I say, "(F || E)," while you say, "!(F || E)." Perhaps you think it is going to be mediocre or be boring, instead of being either frigid or epic. But, whatever you think is actually going to happen, you hold that "it is not the case either that it will be frigid or that it will be epic." Neither is going to occur, you claim.

So, it goes like this. "It is not the case either that it will be frigid or that it will be epic"—"!(F || E)." But that means, "Neither will it be frigid, nor will it be epic" (or, "It will neither be frigid nor be epic"). But that means, "It is not going to be frigid, and it is not going to be epic"—"(!F && !E)." In other words, "!(F || E)" means the same thing as "(!F && !E)."

!!!SYMBOLISM WARNING!!!

Now, imagine that your parents ask you what phys ed stuff you are doing this semester. They suggest you join the fishing or equestrian teams, but you refuse. You say, “I am not going to fish, and I am not going to do equestrian. You can’t make me.” If “F” stands for “I am going to fish,” then “!F” stands for “I am not going to fish. And if “E” stands for “I am going to do equestrian,” then “!E” stands for “I am not going to do equestrian.” What you have told your parents (“I am not going to fish and I am not going to do equestrian”) is “(!F && !E).”

But if you say you are not going to fish and you are not going to do equestrian, you are saying you will do neither of them. You will not do either of them. It is false that you will do either one or the other—“!(F || E).” If you are refusing to do both, then you won’t do either. So, “(!F && !E)” (“I am not going to fish, and I am not going to do equestrian”) means the same thing as “!(F || E)” (“I am not either going to fish or do equestrian.”).

So, we have seen that !(F || E) -> (!F && !E). And we have seen that (!F && !E) -> !(F || E). But if both are true, then “it goes both ways.”

!(F || E) <-> (!F && !E)

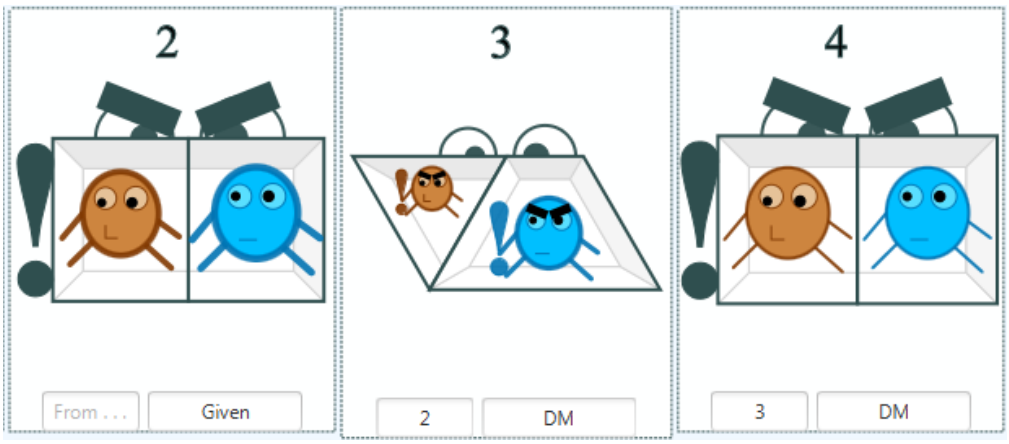
!!!SYMBOLISM WARNING!!!

You will recall, furthermore, that !(F && E) <-> (!F || !E). So, De Morgan’s Theorem says:

!(F && E) <-> (!F || !E)
and
!(F || E) <-> (!F && !E)

Got it? Are you ready for the symbolism to be over? Good. Let’s take a look at some examples using icons instead.

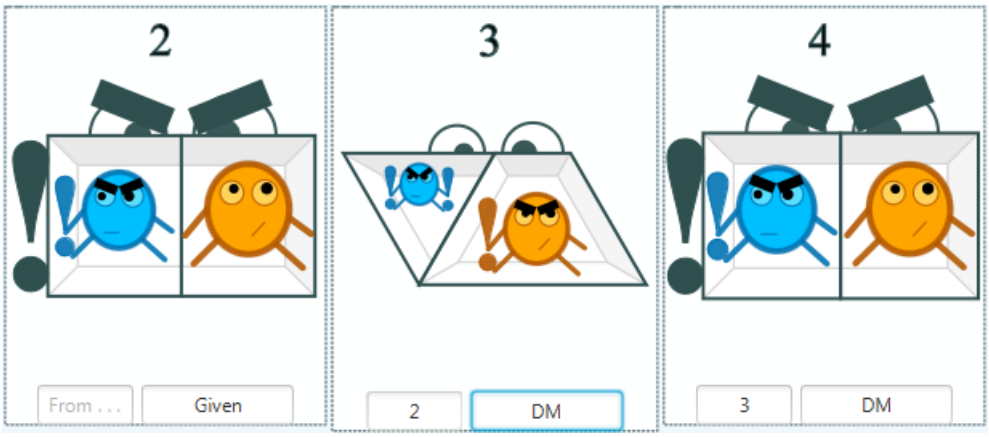
Example 1



In Card 2, we have a Parcel with a Club. That’s all we need to apply DeMorgan’s Theorem. We push the Club through the wall, causing it to splinter into two smaller Clubs, and causing the walls to shift. One of the new smaller Clubs goes to the brown Character in the left chamber, while the other goes to the blue Character in the right chamber. Meanwhile, because of the shifting of the walls, the Parcel has transformed into a Paragon.

Then, in Card 3, we see that we have a Paragon, and the things in both its chambers have at least one Club. Thus, we realize we can apply DeMorgan's Theorem again. We take the Clubs away from the brown Character and the blue Character, and pull them through the wall, back into the open. This passage through the wall reunites them into one larger Club, but also shifts the walls again. The wall-shift transforms the Paragon back into a Parcel.

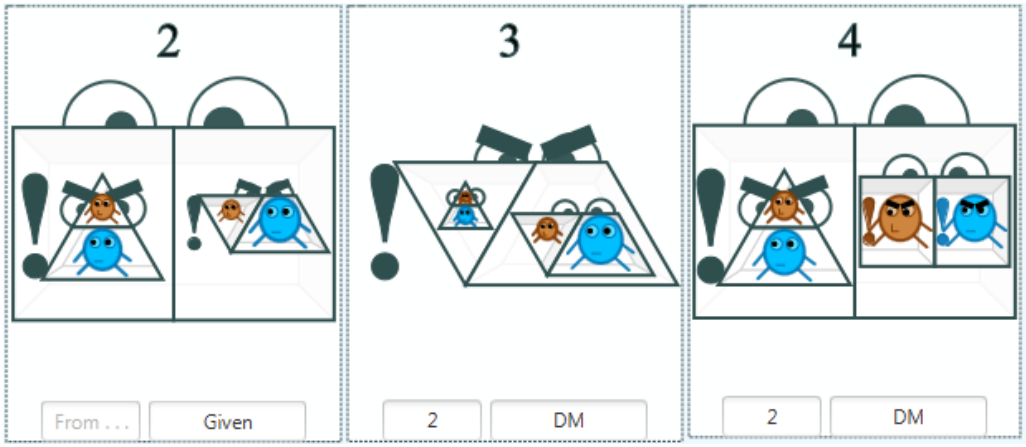
Example 2



In this example, we have a situation much like that in the first. We have a Parcel with a Club. We shove the Club through the wall of the Parcel, whereupon it shatters into two smaller Clubs. One Club goes to the left chamber, and one goes to the right. However, the blue Character in the left chamber already had a Club, so now it has two. Furthermore, the impact of the Club crashing into the wall of the Parcel tilts all the walls, transforming the Parcel into a Paragon.

Then, in Card 3, we see that we have a Paragon, and that the contents of both its chambers have at least one Club. In fact, the blue Character in its left chamber has *two* clubs. So, we take one Club away from the contents of both chambers, and mold them into one larger club, which we give to the outermost Chambergon. However, the impact of this new Club, passing through the wall, shifts the walls' alignment, transforming the Chambergon into a Parcel. Thus, we have a Parcel with a Club, which contains a blue Character that now only has one Club, and an orange Character that has no Clubs.

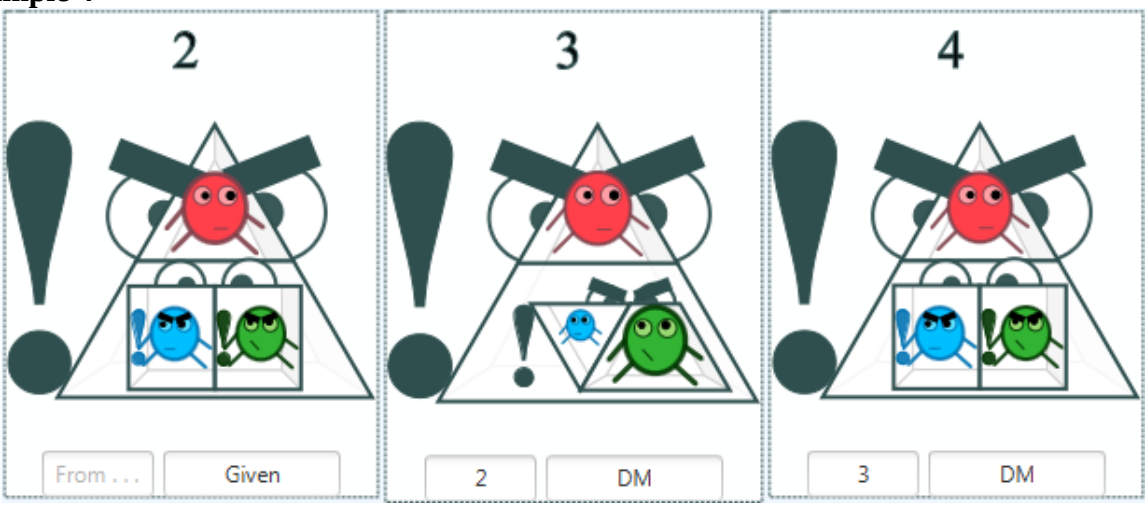
Example 3



In this example, we begin with a Parcel. But this time, the Parcel has no Club. Rather, the contents of its chambers both have one Club. This means we can apply DeMorgan's Theorem. We take the Club from the Pyramid in the left chamber, and the Club from the Paragon in the right chamber, and mold them into one larger Club. As this Club crashes through the wall of the Parcel, however, the walls are destabilized and the Parcel becomes a Paragon.

However, we could've done something different. So, in Card 4, we take our other option. De Morgan's Theorem is a recycling rule, so it can be applied not only to the outermost piece in a Card, but also to pieces that are inside other pieces. In Card 4, we have performed De Morgan on the contents of the right chamber of the Parcel in Card 2.

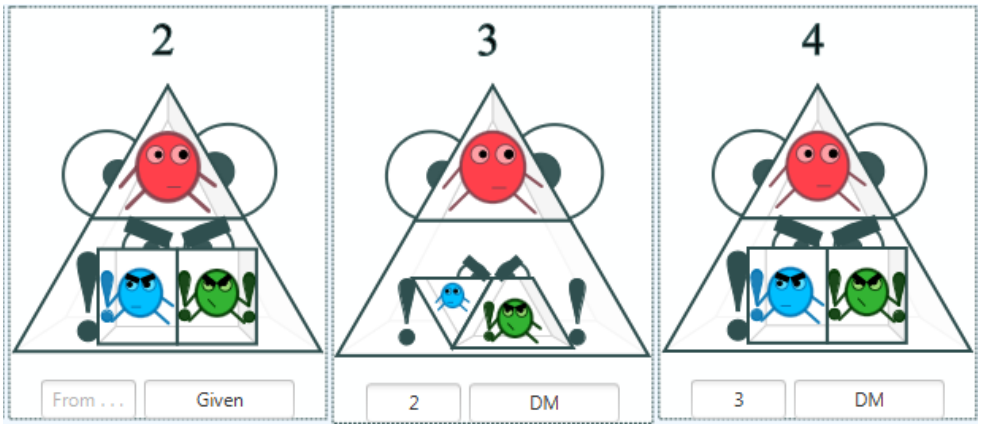
Example 4



Here, we have a Pyramid, with a Club. De Morgan's Theorem, however, only applies to Parcels and Paragons. Looking at the basement of the Pyramid, however, we see a Parcel, the contents of whose chambers both have a Club. In Card 3, therefore, we have combined the two Clubs into one, and move it outside the Parcel. This destabilized the Parcel, shifting it into a Paragon.

Since De Morgan's Theorem is recycling rule, furthermore, we can apply it again, and end up back where we started.

Example 5



This example is very similar to the previous one. The main difference is that we seem to have a choice about how to apply DeMorgan's Theorem when moving from Card 2 to Card 3. We could take one Club away from both the blue and green Characters, and mold the two into a single larger Club, transforming the Parcel into a Paragon, and giving the new Club to the Paragon. Alternatively, it seems like we could move the Parcel's Club inside, giving another Club to both the blue and green Characters, and changing the Parcel into a Paragon. However, if we did this, the green Character would have three Clubs, which is not allowed in Chambergon Battle Logic. (You can triple negate propositions in symbolic logic, but that's overkill; so, in Chambergon Battle Logic, you're limited to two Clubs.)

In moving from Card 3 to Card 4, however, we do not have a choice about how to apply DeMorgan's Theorem. This is because the blue Character in the Paragon's left chamber does not have a Club, and so we could not take a Club away from it. Thus, our only *immediate* option is to take one of the two Clubs away from the Paragon, and split it up, giving one new Club to the blue Character, and another new Club to the green Character.

That is our only immediate option. However, there is another option that would require an extra Card. We *could* apply the Double Negation rule to Card 3, and give the blue Character a pair of Clubs. *Then*, we could apply DeMorgan's Theorem "in either direction." Not only would the Paragon have two Clubs, but the Characters in both its chambers would each have *at least* one Club. (The problem would, once again, however, be that moving Clubs in one of those directions would force one Piece to end up with three Clubs, which is not allowed in Chambergon Battle Logic.)

The lesson to be learned from this is that we can use Double Negation to enable us to apply DeMorgan's Theorem in situations where we otherwise wouldn't be able to. For example, if a Parcel has no Clubs, we can use Double Negation to give it *two* Clubs, and then move one of those Clubs inside, transforming the Parcel into a Paragon. Or, if neither of the contents of a Paragon's two chambers has a Club, we can give a pair of Clubs to each, applying Double Negation over two Cards. Then, we can take one Club away from each, change the Paragon to a Parcel, and fashion those two Clubs into a single larger Club for the Parcel.

Oh the possibilities!

1.2: The Power of Disjunction Elimination (or "Disjunctive Syllogism") Power (A Landfill/Inference Rule)

We covered "Conjunction Elimination" earlier (0.4). That rule says you're allowed to open a Parcel and take out its contents whenever you want (so long as you take them out one at a time). This is the corresponding rule for disjunctions. It says you can open a Paragon and take out one of its contents *if* you have help.

Here's the deal: Parcels have nice, safe, box-like corners. But Paragons have sharp corners. If you try to grab a Paragon to open it, you'll get stabbed by the pointy corners. So, you can't open a Paragon to extract its contents if you won't have something that will beat its sharp edges into a more smooth and graspable shape.

The problem, then, is to find just the right tool for the job. What you need, specifically, is something that will blunt the sharp corner *opposite* to the contents you want to extract from the Paragon. After all, that's the corner you'll need to hold onto as you pry open the lid with your other hand.

So, to extract the contents of the left chamber of a Paragon, you need help from something that exactly matches the contents of the *right* chamber, except that it has one more club than those contents. (After all, one piece will only attack another if they are identical to each other, except that it has one more club than the thing it is going to attack.) And to extract the contents of the right chamber of a Paragon, you need help from something that exactly matches the contents of the *left* chamber, except that it has one more club than those contents.

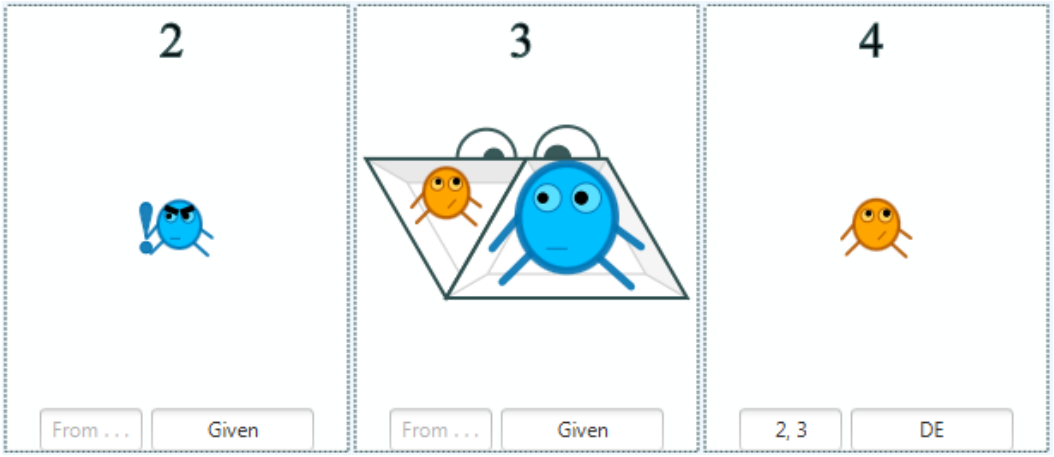
Restrictions: This rule will not work on a Paragon that has a Club. If it has a Club, it will fend off the attack, and nothing will come of it. Likewise, this rule will not work if the Paragon in question, and the attacker you turn loose on it, are not both the main pieces in two separate Cards. (Neither can be inside a Chambergon, in other words.)

The logical justification for this rule is much more straightforward than the justification for De Morgan's Theorem. The idea is this: If you know that either Sally or Billy stole the last cookie, but then you discover that Sally didn't steal the cookie, you can conclude that Billy must have. After all, it was one of the two of them, and it wasn't Sally.

In other words, if you know that either A or B is true (you're just not sure which), but then you happen to discover that A isn't true, you can rightfully conclude that B must be true. After all, the whole statement, "(A || B)," was true, and that means one of its parts (either A or B) must be true. So, if A turned out not to be true ("!A"), the only option left over is that B is true.

This kind of reasoning—where you start with one claim, and then add a second that allows you to draw a conclusion—is called a "syllogism." That's just an old, fancy Greek word for "argument," but it's helpful to know. And since this kind of argument centers on a "disjunction," the rule is often called "Disjunctive Syllogism" instead of "Disjunction Elimination."

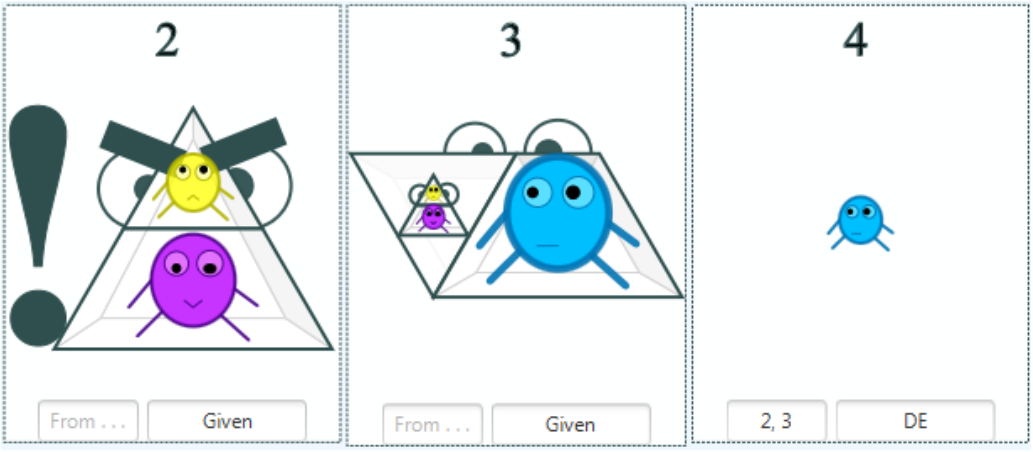
Example 1



Here we have a lone blue Character with a Club. On the horizon, it spies a Paragon. And in one chamber of that Paragon, it sees a blue Character . . . *with no Club*. Its anger begins to rise, like the Incredible Hulk. It charges the Paragon, heedless of the fact that it is trying to singlehandedly take on a tank. But it knows the Paragon's weakness: those acute corners are unstable, and thus vulnerable to a Club attack. And *attack* is just what the blue Character with a Club does. It will not stand for a simple blue Character to go on peacefully riding around in a tank, believing things that are totally wrong. No. That blue Character must be destroyed.

And when the blue Character is destroyed—when its chamber of the Paragon lies in ruins—the orange Character in the other chamber suddenly realizes it is free. No longer will it be tortured by the annoying way the blue Character breathes through its nose when it is listening to music too loudly through its headphones. It is free. Thus, in Card 4, the orange Character moves out into the open.

Example 2



Imagine what we have above are three scenes from a movie. The two Chambergons in Cards 2 and 3 are "Given." But then we get to Card 4. In this scene, there's a blue Character running around free, not in any Chambergon whatsoever. Where did it come from? What happened between the third and fourth scenes of the movie to allow the blue Character to be in this position?

Looking back over Cards 2 and 3, we see a blue Character in the right chamber of the Paragon. So, we conclude that the blue Character must have come there. Now, the justification for Card 4 says that the blue Character is "from" both Cards 2 and 3, using "DE" ("Disjunction Elimination"). That is, somehow the blue Character got free because of the application of the Disjunction Elimination rule to both Cards 2 and 3. But what does this mean?

Here is the sordid tale. You will recall that anything with a Club loathes anything it matches, but which has one fewer Clubs than it does. So, the yellow/purple Pyramid in Card 2 hates all yellow/purple Pyramids who have no Clubs, since it has exactly one Club. What does this mean? How can a Pyramid have feelings? And what is it that this yellow/purple Pyramid hates about all yellow/purple Pyramids who have no Clubs?

It could be that when we say the yellow/purple Pyramid with one Club hates all yellow/purple Pyramids with no Clubs, what we mean is that the yellow Character and blue Character in the Pyramid strongly disagree with any yellow Character and purple Character being together in a Pyramid *like that*. After all, all yellow Characters belong to a single class, as do all purple Characters, all orange Characters, and so on. So, maybe the yellow Character and purple Character in the Pyramid with a Club are "classists." Maybe they think that purple Characters belong to the upper class, and yellow Characters belong to the working class, and so it is just not right for a yellow Character to be in the attic of a Pyramid while the purple Character is in the basement.

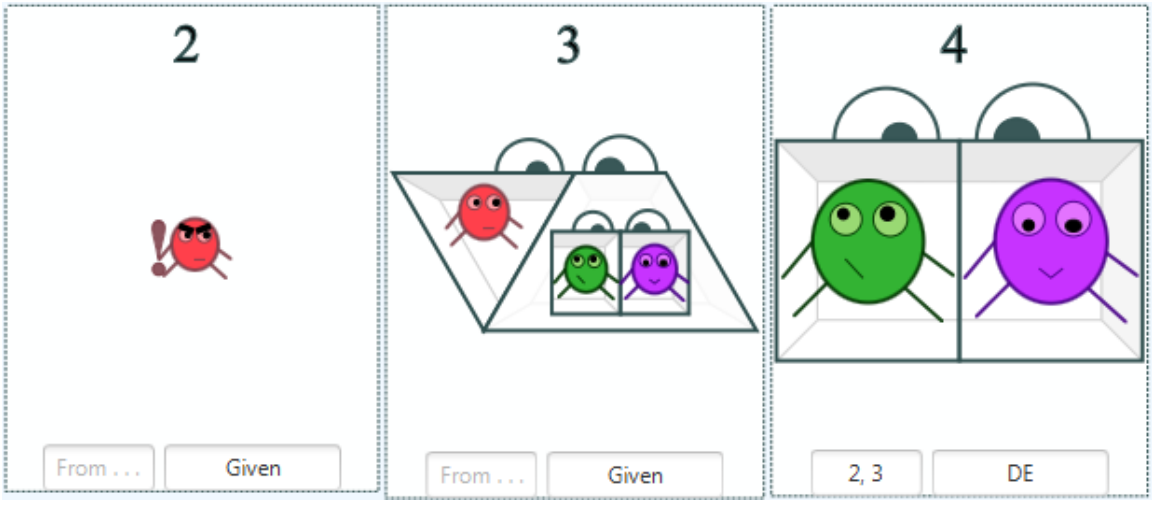
Or, maybe they are made uncomfortable by the implication that yellow Characters rest on, depend on, or are supported by, purple Characters. That is, after all, what a normal yellow/purple Pyramid with no Club would tell you (since the yellow Character is in its attic, which rests on, or depends on, the basement, which contains the purple Character). After all, nobody wants to believe they intrinsically depend on other people, and nobody wants to be told that they exist only as support for someone else.

But why should this all be so personal for the yellow/purple Pyramid with the Club? Why the *hate* that would lead it to attack any yellow/purple Pyramid who had no Club? It's only because the yellow/purple Pyramid with a Club recognizes itself in the yellow/purple Character with no Club. After all, they're both Pyramids, and both have an yellow Character in the attic and a purple Character in the basement. And sometimes the things that you dislike most about other people are the things you are actually most disgusted with, or ashamed of, about yourself. And once an yellow/purple Pyramid gets a Club, the negative powers that emanate from the Club elicit and intensify this natural self-disapproval, magnify it into self-loathing, and turn the fire of this fury against everything the yellow/purple Pyramid sees in others that it is opposed to in itself.

Or maybe, since you can think of a Pyramid as something like a tank that has been transformed into a mobile antiaircraft platform, the fact that an yellow/purple Pyramid with a Club will seek out and destroy all other yellow/purple Pyramids that have no Club is a consequence of faulty programing. Maybe the tank/antiaircraft platform's automated search for enemy troops is interfered with by the presence of the Club, and this sends the Pyramid on an unending seek-and-destroy mission against all Pyramids that match it, but have one fewer Club. (After all, why go after another Pyramid that has an equal number of Clubs, or a greater number? You'd just be asking for trouble.)

Whatever the case, once the yellow/purple Pyramid with a Club in Card 2, above, realizes there's an yellow/purple Pyramid *with no club* in the left chamber of the Paragon in Card 3, it will go berserk and fall upon it in a rage. And since the Paragon in Card 3 has no Club of its own, it cannot properly defend itself. The yellow/purple Pyramid with a Club will destroy the left chamber of the Paragon, and the Clubless yellow/purple Pyramid it contains. And this is how the blue Character ends up escaping into Card 4.

Example 3

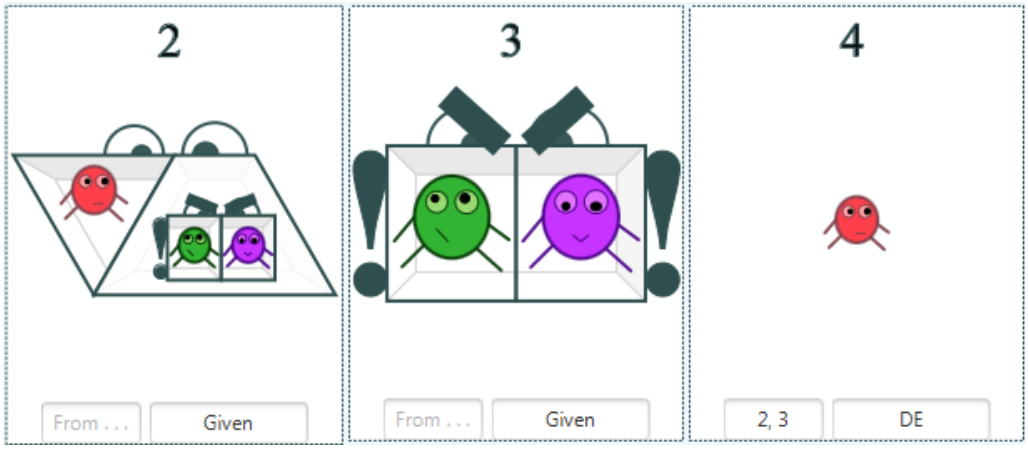


We could tell much the same story here as we did for the previous example. We are given a Paragon in Card 3, and something that matches the contents of one of its chambers, except that the matching thing has one more Club that the contents of that chamber. The red Character with a Club in Card 2, specifically, hates the Clubless red Character in the left chamber of the Paragon in Card 3.

So it attacks.

Now, if that red Character in Card 3 had been in a Parcel, rather than a Paragon, it would have been safe. The compacted shape of a Parcel could have withstood the attack. But the acute angles of the Paragon, while sharp and dangerous, are also weak points in the structure. And the red Character with a Club in Card 2 takes advantage of this weakness to wreak havoc. This results in the freedom of the green/purple Parcel in Card 4, but you have to ask yourself: was it really worth it?

Example 4



The green/purple Parcel in Card 1 hates all green/purple Parcels who have no Clubs, because it has exactly one Club. But that's irrelevant, because it's locked away inside a Paragon, and can't do anything to anyone. But the green/purple Parcel with *two* Clubs can. And what it hates are not

green/purple Parcels with no Clubs. After all, we could easily turn it into such a Parcel just by using the Double Negation rule on it. No. What it hates are green/purple Parcels with *one* Club. Those are the enemies it roams the battlefield seeking to take down.

When turned loose on the Paragon in Card 2, therefore, the green/purple Parcel with two Clubs will attack the right chamber, freeing the red Character from the left chamber. And the red Character is grateful, of course. Look at it, sitting there, being grateful in Card 4.

The important point to notice here is that the order of the Paragon and the Club cards doesn't matter. In previous examples, the Piece with the Club that did the attacking was the first Card in the sequence, while the Paragon Card was second.. Here, however, it the piece that does the attacking with the Club is the second card, with the Paragon-to-be-attacked being the first. This is irrelevant. All that matters is that you have one Card whose outermost Piece is a Paragon (and that that Paragon has no Clubs), and that you have another Card whose outermost Piece exactly matches the icon in one of the chambers of the Paragon, except that it has one more Club than that icon.

1.3 Conclusion

Remember that all the powers from Chapter 0 are still in effect. Sometimes, to use De Morgan you first have to apply Double Negation, so that you have enough Clubs. Other times, you have to use Conjunction Introduction to get a Parcel that you can use De Morgan on. Still other times, you have to use Conjunction Elimination to extract a Piece with a Club from a Parcel, so you can use it to attack a Paragon in another Card. So, as you level up, you will not only be unlocking new powers, but will be unlocking new power sequences and combinations.

Do not forget, furthermore, that there is a difference between recycling ("equivalence") rules and landfill ("inference") rules. Recycling rules allow us to swap pieces out for their equivalents, and thus can be applied to any icon anywhere. Landfill rules, however, can only be applied to the outermost Pieces in the Cards they use. Icons inside those pieces can't be touched unless you have a rule that specifically allows you to extract them and put them in a Card all by themselves.

Chapter 2:
Level 2 Powers and Activation Games

2.0: Recap

The Level 0 Powers were Double Negation, Tautology, Conjunction Elimination, and Conjunction Introduction. Double Negation and Tautology are recycling rules, which means that they can undo themselves. If you use them, but find you don't like the results, you can always apply them again to get back to where you started. The recycling rules don't actually change anything, logically-speaking—they just change the way things look. And, since they don't change anything you can apply them to any piece/icon anywhere in any card.

What was cool about Double Negation and Tautology, furthermore, is that they both doubled things. Double Negation doubled Clubs. Tautology doubled everything else. (And it even doubled Clubs so long as you were doubling whatever the Clubs were attached to in the process.) Conjunction Elimination and Introduction, however, were not cool. They were boring. One let you open boxes and take out the stuff inside. Boring. The other let you take the stuff you took out and put it back in. Boring. Whereas the recycling powers ("equivalence rules") could undo themselves, the landfill rules ("inference rules," like Conjunction Introduction and Conjunction Eliminatio) require a counterpart power in order to undo their damage. Wimps. Take some responsibility for the messes you make.

Furthermore, landfill rules could only be applied to the outermost piece in each Card. You couldn't unpack a box with Conjunction Elimination if it was stuck inside another box. (After all, it's inside the other box! You have to open the outisde box before you can open the inside box.) And you couldn't pack things into a box if that box was inside another box. You can't just magically teleport things into things that inside other things. This is not Star Trek.

"But how come you can Double Negate or Tautology things even if they're inside other things?" you wisely ask. Because recycling rules don't actually change anything. The logical meaning of the overall structure remains the same. With landfill rules, however, you're actually adding something new, or leaving out something old. You don't just have the same structure all over again, in a slightly different form.

Level 1 then added another power to each set, so that we now know three recycling rules and three landfill rules. The new recycling rule was De Morgan, which showed us how to change Parcels into Paragons by moving Clubs from outside to inside, or from inside to outside. The new landfill rule was Disjunction Elimination, which showed us how to open a Paragon without cutting our hands on its sharp corners. Disjunction Elimination also involved Clubs, you will recall. It's just that the Clubs in question had to be brought in from another card. Using Conjunction Elimination, you can unpack a Parcel by yourself. If you want to unpack a Paragon, however, you need assistance from the main inhabitant of another Card, who just happens to match one of the things inside the Paragon, while having one more Club than that thing. Both Conjunction Elimination and Disjunction Elimination, therefore, are ways of getting rid of a

Chambergon and extracting what's inside it. But Disjunction Elimination is a more complex, surgical procedure.

Just like Conjunction Elimination could be undone by Conjunction Introduction, Disjunction elimination also has a partner and opposite power: Disjunction Introduction. It may be the strangest power in this entire game, though, so we did the more relatively-normal DE first. In fact, while we will be learning Disjunction Introduction in this chapter, we're going to do another power first: Implication. In this chapter, therefore, we will continue the tradition of doing half recycling rules, and half landfill rules. And we'll continue the tradition of doing the recycling rules first.

2.1: Implication

2.1.0 Fear of Heights

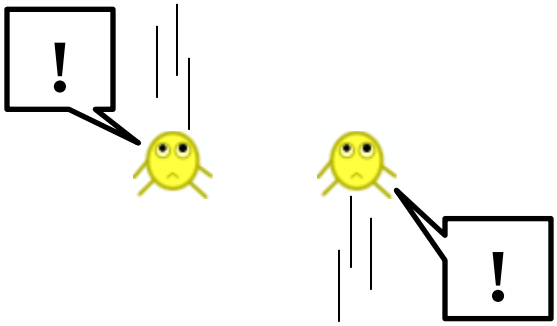
The motto for this rule is, "You can turn any Paragon anywhere into a Pyramid, and vice versa." This is one of the most important rules in the entire game, but it is not without precursors. It essentially extends the principle introduced by De Morgan's Theorem: the main connectives of logic are not actually distinct operations. They are a single operation manifesting in different guises depending on how negation is applied.

Like De Morgan's Theorem, furthermore, Implication allows you to give or take a single Club, rather than having to give and take them in pairs (using Double Negation). De Morgan does this by—to speak metaphorically—transferring responsibility for the group's fear and anger to the individuals, or by transferring responsibility for the individuals' fear and anger to the group. The Club moves from belonging to the Parcel or Paragon, to belonging to the contents of the Parcel or Paragon (or vice versa).

Implication, in contrast—and to speak metaphorically, again—has to do with the response of individuals within the group to changes in the group. It is a way of manipulating the feelings of the parts by changing the whole.

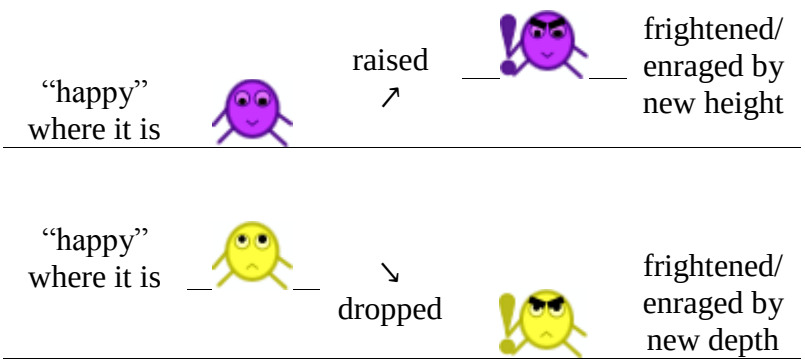
And it is based on the principle that people don't like sudden changes in height.

Here's how it works. When someone is frightened suddenly by falling, or by being flung unexpectedly upwards, the surprised and frightened person will cry out. That is, she or he will exclaim—emit a wordless exclamation. And how would you draw that if you an artist?



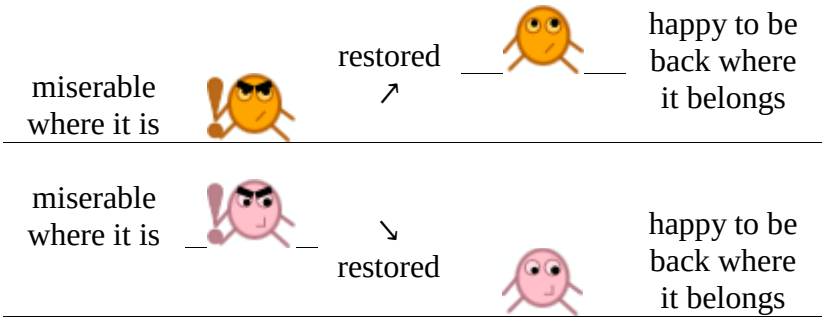
Exactly.

So, if a Character or Chambergon is happy where it is—if it has no Clubs, and thus no fear or anger—all we have to do is lower it or raise it within whatever Chambergon contains it.



In addition, however, to learning that we can give a Character or Chambergon one more Club than it already has by quickly dropping or raising it in a frightening manner, we can also learn from this how to take away a single Club from a Character or Chambergon. If a Character or Chambergon picks up an extra Club by being suddenly dropped from where it was, or suddenly lifted from where it was, then to get rid of that extra Club all we have to do is to restore it to its original level.

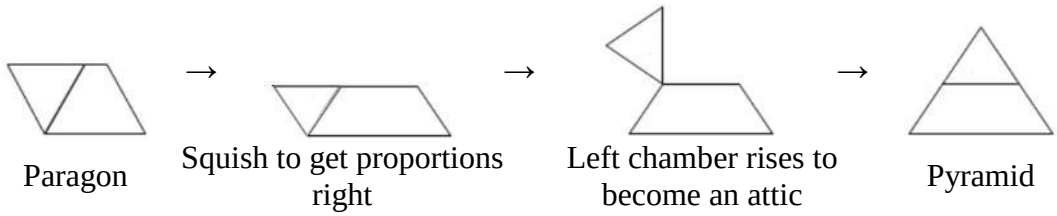
If a Character or Chambergon is on ground level and has a Club, this must mean it was dropped from a higher level, and we can soothe its fright by returning it to that higher level. Being back in its home environment, it will relax, and its fear and rage will disappear—as will one of its Clubs. Likewise, if it is on a higher level and has a Club, this must mean it was suddenly raised from ground level. All we have to do to soothe its fright, therefore, is to return it to ground level. Being back in its home environment once more, it will feel safe, and its fear and rage will evaporate—along with one of its Clubs.



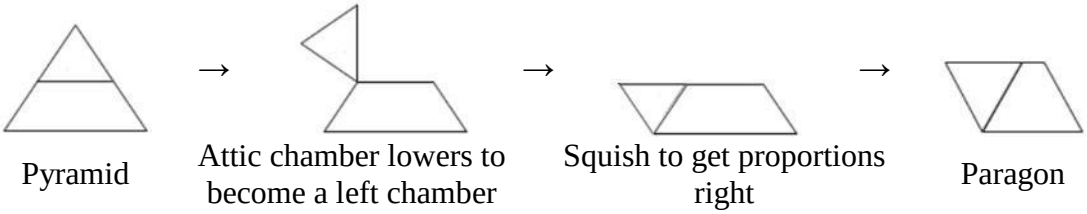
Now, here's the deal: there is only one place in the whole of Chambergon Battle Logic that is higher than ground level, and that is in the attic ("chamber 1") of a Pyramid. Every other place in the game—whether inside or outside a Chambergon—is technically at ground level. So, if you find a Character or Chambergon with a Club, and that Character or Chambergon isn't in the attic of a Pyramid, you know it fell from the attic of some Pyramid. If you can just get it back into the attic of a Pyramid, one of its Clubs will disappear.

Likewise, if you find a Character or Chambergon with a Club, and that Character or Chambergon *is* in the attic of a Pyramid, you know it used to be somewhere else. *Anywhere* else. And if you can just find a way to get it out of the attic of that Pyramid and into anywhere else, one of its Clubs will disappear.

So, what we're ultimately talking about is raising things into, or lowering things from, the attic of a Pyramid. And this exactly what Implication allows you to do. You take the left chamber of a Paragon, and tip it up on top, creating a Pyramid, and thus raising whatever was in the chamber into the attic of a Pyramid.



Likewise, you can use Implication to tip the attic of a Pyramid to the left, creating a Paragon, and lowering whatever was in the attic to ground level.



2.1.1: The Power of Implication (A Recycling/Equivalence Rule)

Implication allows us to change Paragons into Pyramids, and Pyramids into Paragons, by moving the left/top chamber as seen above. In addition to changing the outer shape of the Chambergon, however, we must also change the number of Clubs of whatever is in the chamber being flipped. If the Character or Chambergon in the lowering/raising chamber has no Club, it gains exactly one Club. It is being frightened. If it has one or more Clubs, it loses exactly one Club. It is being returned to its original position, and thus is relaxing.

Restrictions: There are no restrictions on this rule. You can use it on any Paragon or Pyramid anywhere, even if it is inside some other Chambergon, and even if it has a Club.

The logical justification for this rule goes as follows. Imagine that you want to visit your friend Bobby, but you're 9 years old and can't drive. You ask your mom for a ride, and she tells you, "If you want to visit Bobby, you'll have to come when I go grocery shopping. I ain't going out twice today." This, in effect, gives you a choice. Either you don't go see Bobby, or you go grocery shopping with your mom. In other words, the conditional claim, "If you want to see Bobby, you have to come grocery shopping with your mo," is logically equivalent to the disjunctive claim, "Either you don't go see Bobby or you go grocery shopping with your mom."

Any conditional statement can be changed into a disjunctive statement, in other words, if we are willing to negate the antecedent. "(A -> B)" is logically equivalent to "(!A || B)."

But there's more. If you don't buy it that "(A -> B)" says the same thing as, "(!A || B)," look at the following.

Imagine that we write our two complex propositions "(A -> B)" and "(!A || B)" side-by-side, divided by a line. Let's underline them, while we're at it.

(A -> B)	**(!A || B)**

Now, imagine that we notice that both propositions contain the simpler propositions "A" and "B." So, let's give "A" and "B" their own spot in our lineup of propositions.

A	**B**	**(A -> B)**	**(!A || B)**

Now, each of our four propositions ("A," "B," "(A -> B)," and "(!A || B)") is either true or false. Since we don't know what these propositions actually are (who knows what claims "A" and "B" stand for?) we don't know whether they're true or not. All we know are the possible combinations of truth and falsity. Those possibilities are as follows:

A	**B**	**(A -> B)**	**(!A || B)**
T	T		
T	F		
F	T		
F	F		

In other words, either "A" and "B" are both true (the first line), or "A" is true while "B" is false (the second line), or "A" is false while "B" is true (the third line), or both are false (the fourth line. So, let's fill in all the values for "A" and "B" in the two more complex propositions.

A	**B**	**(A -> B)**	**(!A || B)**
T	T	T	T
T	F	T	F
F	T	F	T
F	F	F	F

On the first line, the leftmost "A" is true, so all the other A's[26] on that line have to be true too. The leftmost "B" is also true on the first line, so all the other B's on that line have to be true as well. On the second line, however, the leftmost "B" is false, so all the other B's have to be false. Etc.

[26] Yes, I use apostrophes to pluralize "A" and "B". It actually is appropriate to use apostrophes in this case. Look it up.

!!!WARNING: TECHNICAL STUFF!!! !!!WARNING: TECHNICAL STUFF!!! !!!WARNING: TECHNICAL STUFF!!!

!!!WARNING: TECHNICAL STUFF!!! !!!WARNING: TECHNICAL STUFF!!! !!!WARNING: TECHNICAL STUFF!!!

In conditional propositions (like "(A –> B)") the proposition as a whole is false if the antecedent is true and the consequent is false. So, if "A" is true, but "B" is false, then the whole statement is false. Otherwise, conditional statements are always true. Underneath the arrow for the conditional statement, therefore, we can write the truth value for the statement given the truth values of its antecedent and consequent for that line.

A	**B**	**(A**	**–>**	**B)**	**(!A**	**||**	**B)**
T	T	T	T	T	T		T
T	F	T	F	F	T		F
F	T	F	T	T	F		T
F	F	F	T	F	F		F

Now, we want to do the same thing for "(!A || B)." First, however, we have to negate all of A's truth values, since the "A" has an exclamation mark in front of it. So, we cross out the first value we wrote for "A," and write its opposite under the exclamation mark.

A	**B**	**(A**	**–>**	**B)**	**(!A**	**||**	**B)**
T	T	T	T	T	F~~T~~		T
T	F	T	F	F	F~~T~~		F
F	T	F	T	T	T~~F~~		T
F	F	F	T	F	T~~F~~		F

Disjunctive statements (like "(!A || B)") are true whenever either one of their disjuncts (the two simpler propositions they unite) are true. The only time a disjunction is false is when both its component propositions are false. But that means "(!A || B)" is only false when "!A" is false *and* "B" is false *at the same time*. On all the other lines of the table, "(!A || B)" will be true.

A	**B**	**(A**	**–>**	**B)**	**(!A**	**||**	**B)**
T	T	T	T	T	F~~T~~	T	T
T	F	T	F	F	F~~T~~	F	F
F	T	F	T	T	T~~F~~	T	T
F	F	F	T	F	T~~F~~	T	F

What we just did is called "filling in a truth table." We show "the answer" we have found by circling the columns for the main connectives in the two complex propositions we were analyzing.

A	**B**	**(A**	**–>**	**B)**	**(!A**	**||**	**B)**
T	T	~~T~~	T	~~T~~	~~F~~~~T~~	T	~~T~~
T	F	~~T~~	F	~~F~~	~~F~~~~T~~	F	~~F~~
F	T	~~F~~	T	~~T~~	~~T~~~~F~~	T	~~T~~
F	F	~~F~~	T	~~F~~	~~T~~~~F~~	T	~~F~~

But notice this: both the circled columns are identical. On every line where "(A –> B)" is true, "(!A || B)" is also true, and on every line where "(A –> B)" is false, so is "(!A || B)." No matter

!!!WARNING: TECHNICAL STUFF!!!

how we arrange the truth values of their component propositions (i.e., no matter whether we make "A" true or false, and no matter whether we make "B" true or false), "(A –> B)" always ends up with the same truth value as "(!A || B)." And that means the two propositions are "logically equivalent." They say the same thing, or have the same meaning, as far as we are concerned as logicians.

Now, remember why we were doing all this. We were just trying to show that the logical rule called "Implication" (or "Material Implication") is legitimate. Implication says that whenever you have a conditional statement, you can change it to a disjunctive statement if you negate its antecedent. The truth table we just filled out has shown us why this is true: a conditional is logically equivalent to a disjunction of the same component propositions if the first component proposition is negated.

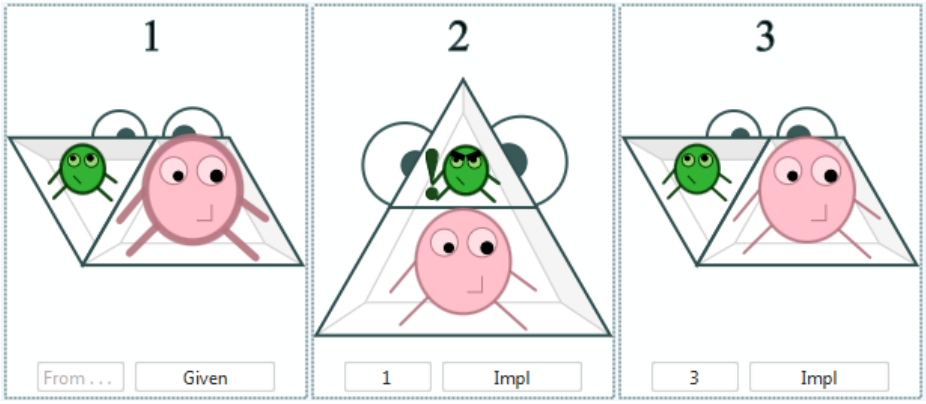
We must not forget, however, that we can use Implication to change conditionals into disjunctions *and* disjunctions into conditionals. After all, if "(A –> B)" equals "(!A || B)," then it must also be true that "(!A || B)" equals "(A –> B)." Furthermore, according to the Implication rule, "(A || B)" equals "(!A –> B)." Take a look at their truth table.

A	**B**	**(A**	**||**	**B)**	**(!A**	**–>**	**B)**
T	T	T	T	T	F~~T~~	T	T
T	F	T	T	F	F~~T~~	T	F
F	T	F	T	T	T~~F~~	T	T
F	F	F	F	F	T~~F~~	F	F

No matter how you arrange the T's and F's for "A" and "B," the more complex statements "(A || B)" and "(!A –> B)" always end up with the same truth vale.

Summary: If you have a conditional statement, you can change it to a disjunction if you "flip the sign" of its antecedent. (If the antecedent had one Club, give it zero clubs. If it had zero, give it one. If it had two Clubs, give it one club.) Similarly, if you have a disjunction, you can change it to a conditional if you "flip the signs" of the left disjunct. (If the component proposition on the left had one Club, give it zero clubs. If it had zero, give it one. If it had two Clubs, give it one club.)

Example 1

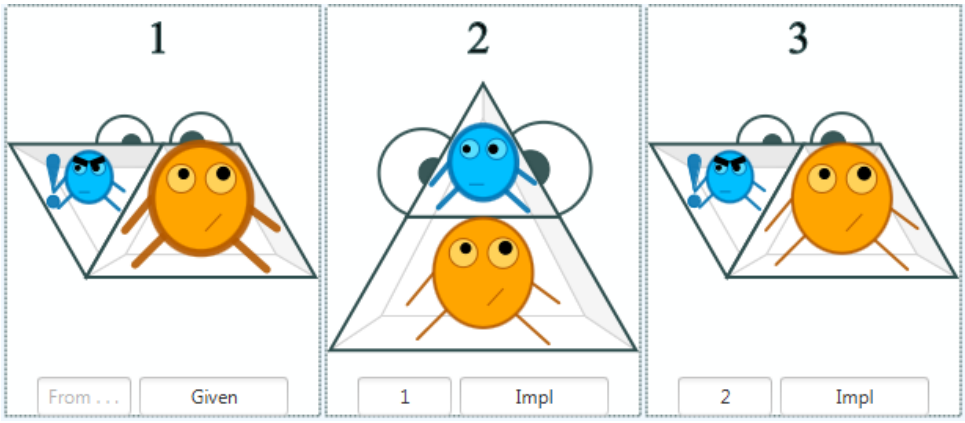


Here, we use Implication first to tip the left chamber of the Paragon up on top, forming a Pyramid, and then to tip it back to the left, reforming the Paragon. When we tip the chamber up, we are raising the green Character with in it. This frightens the green Character, giving it a Club. When moving from Card 2 to Card 3, however, the Character we are dropping already has a Club. This tells us that it is not comfortable being at the height of a Pyramid's attic, and originally comes from ground level. So, we tip the Pyramid's attic over to the left, returning the green Character to ground level. It relaxes, and its Club disappears.

Notice that we are always dealing with moving from the left chamber of a Paragon to a Pyramid's attic, and vice versa. The right chamber of the Paragon simply becomes the basement of the Pyramid, and vice versa. Whatever is in it, therefore, remains at ground level the entire time, and thus does not gain or lose a Club. Remember, in other words, not to use Implication to change whatever is in the basement of a Pyramid or in the right chamber of a Paragon, and make sure that when you tip the attic of a Pyramid over, in order to form a Paragon, that you tip it *to the left*.

Now, I have to admit something: some, perhaps many, logicians will not let you use Implication unless the occupant of the Paragon's left chamber has at least one Club. This doesn't mean you can't use Implication in the example above; it just means you'd need to stick an extra Card between Cards 1 and 2, and between Cards 2 and 3. This would be a Double Negation Card, in which you give the green Character two Clubs. Then everyone would be happy. But all you would really end up doing is what we did above, and given the way similar rules like Modus Tolens and Contraposition work (which we'll deal with later), I think this requirement is arbitrary. Thus, while you should be aware that other people will be more restrictive in their use of Implication, you should also ignore that restriction when playing Chambergon Battle Logic. Because we're rebels.

Example 2



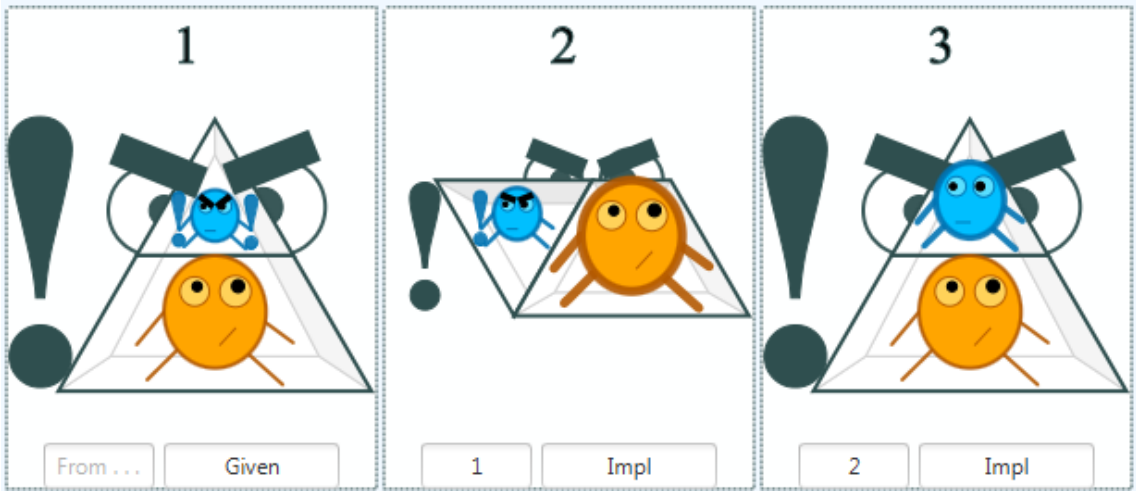
When we encounter the blue Character with a Club in Card 1, we know it is not comfortable where it is. The Club tells us that it has been frightened by being moved here. But where is it? Is it at ground level, or is it higher than ground level? To answer this question, we have to ask whether its chamber is at ground level, or higher than ground level. The answer is that it is no higher than the right chamber, which is at ground level, and it is resting on its tip, which is on the ground. So, even the left chamber of a Paragon is at ground level.

Now, since the blue Character with a Club in Card 1 is at ground level, and since a Club is an indicator that the blue Character is not happy where it is—that it was shocked and frightened by being moved there suddenly—we know that the blue Character originally came from some higher plane. And we also know that it is to that higher plane that it longs to return. We can comfort it, therefore, if we only return it to a higher level. And we can do just that with Implication.

So, in Card 2, we apply Implication to Card 1, flipping the left chamber of the Paragon up on top of the right chamber to create a Pyramid. This raises the blue Character back to its home state, and relaxes it. Its Club disappears.

And then! For some reason we decide to be mean to it. Seeing that the blue Character is happy in its current position in the attic of the Pyramid, we decide to scare it by dropping it suddenly. Implication allows us to do this, and so we do, producing Card 3, and a very frightened and angry blue Character.

Example 3



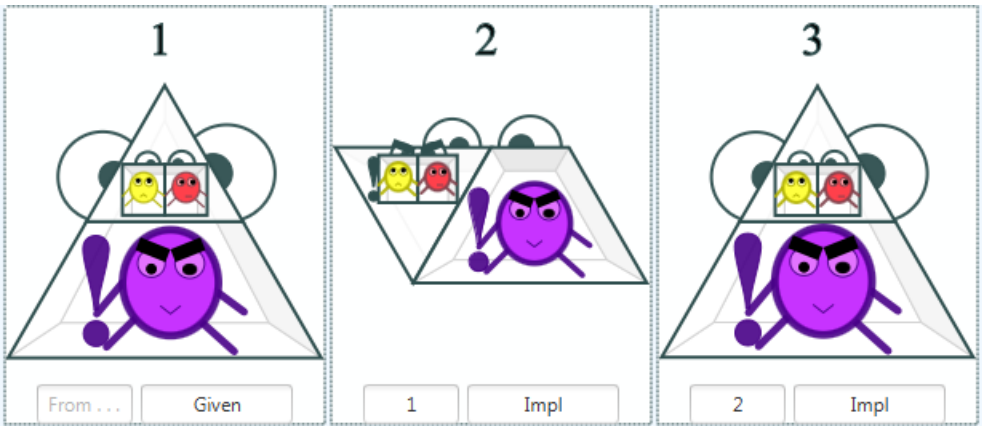
Now, this is an odd example for a couple reasons. First, there are Clubs everywhere. The reason I included the Pyramid's Club, in addition to the two Clubs belonging to the blue Character, is to show that the fact that the Pyramid has a Club doesn't keep you from applying Implication. After all, you're not trying to open up the Pyramid and take anything out of it. You're just rearranging it into a Paragon.

Second, the blue Character starts out with two Clubs, and loses one Club each Card. After all, the rule is that a Character who already has a Club will lose a Club when it is raised or lowered. Normally, when you apply a rule like Implication twice, you get back to where you started. But here, you start with an blue Character with two Clubs, and end with an blue Character with no Clubs.

If you think about it, however, you'll see that we *have* gotten back to where we started. Cards 1 and 3 are essentially identical; the only difference is how many Clubs the blue Character has, and the blue Character in both Cards has an even number of Clubs (since zero counts as even in

Chambergon Battle Logic). That is, the blue Character with two Clubs in Card 1, and the blue Character with no Clubs in Card 2, are on the same team. After all, you could take the two Clubs from the first, using Double Negation, or give two Clubs to the second, using Double Negation. They are, as we say, logically equivalent to each other.

Example 4

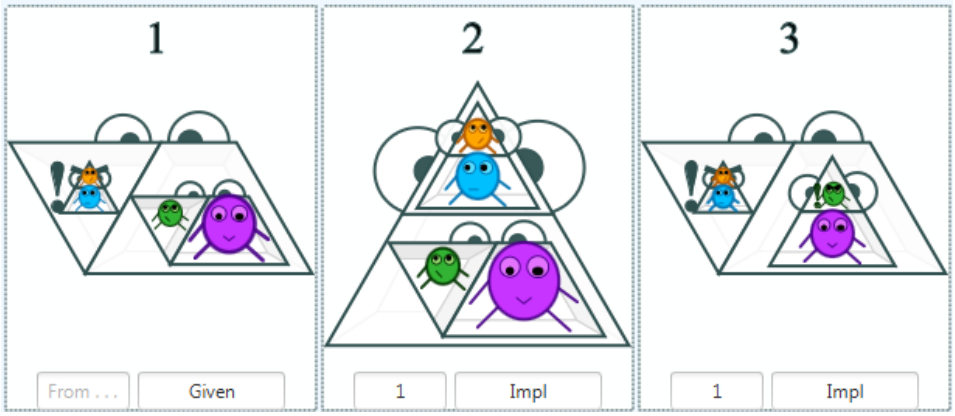


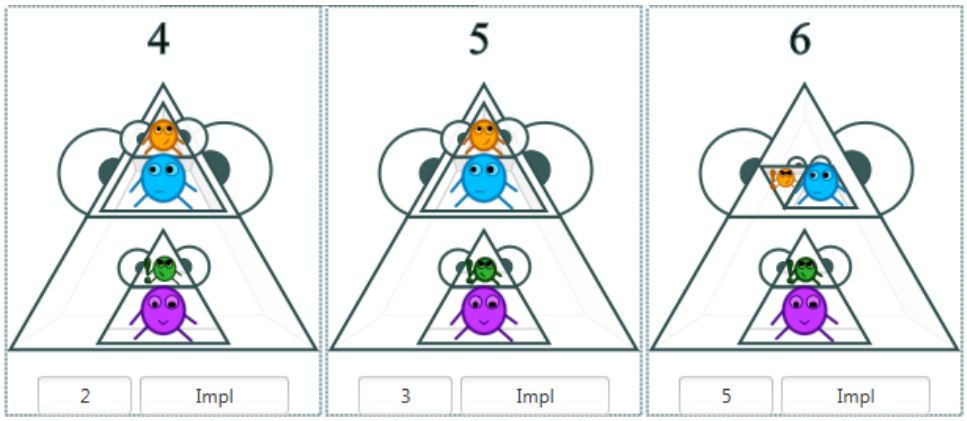
In this example, we do not frighten a Character by suddenly dropping it from the sky. Rather, we frighten a Parcel. As far as the Characters inside the Parcel are concerned, nothing has changed. They're inside a Parcel, and can't see what's going on outside. But the Parcel itself is freaked out. It, as we can see from Card 2, is afraid of depths.

Don't ask me how a Parcel can be frightened. It's something to do with artificial intelligence, probably. What I do know is that the Parcel is much happier once it's been returned to attic level in Card 3. Okay, it's happier when the *chamber* it's in has been returned to ground level. That just feels much more dominant and safe to it.

Contrast all of this with the purple Character. It's at ground level the whole time, and is frightened and angry the whole time. It wants to be in the attic of a Pyramid. That's where it fell from. But did you give it what it wanted? No. You gave what it wanted to the Parcel.

Example 5





In this example, I deliberately included a Pyramid and a Paragon, both inside a Paragon, to show that Implication works no matter what is inside the Pyramid or Paragon we are dealing with. In Card 1, we see that the Pyramid is unhappy (it has a Club). So we make it happy in Card 2 by raising it—using Implication—back to the level it must have fallen from. This leaves the Paragon in the basement untouched.

In Card 3, *we go back to Card 1*, and do something different to it. This time, we use Implication on the Paragon inside the Paragon. We transform this inside Paragon into a Pyramid using Implication. Now, since the green Character in the left chamber of the inside Paragon was perfectly happy where it was (it had no Club), being suddenly raised frightens it, and it ends up with a Club.

And, to answer your question, yes. You are allowed to use the same Card twice, and to do different things to it both times. However, notice that while Cards 2 and 3 are different from each other, we get them both by applying Implication to Card 1. It's just we apply Implication to different parts of Card 1. In Card 2, we apply Implication to the outside Paragon in Card 1, transforming it into a Pyramid, and removing the Club that belonged to the inside Pyramid. In Card 2, we apply Implication to the inside Paragon, transforming it into a Pyramid and giving a Club to its green Character.

In Card 4, we then go back to Card 2, and use Implication to turn the Paragon in the basement of the Pyramid into another Pyramid. This frightens the green Character in the new Pyramid's attic, and gives that green Character a Club. In Card 5, in contrast, *we go back to Card 3*, and change the outside Paragon to a Pyramid, using Implication once again. This comforts the inside Pyramid in the Paragon's left chamber, since this returns that Pyramid to the level from which it must have originally fallen. Thus, it loses its Club.

You will notice, however, that Cards 4 and 5 look identical. We applied the same rule to different Cards, and ended up "in the same place" (in contrast with Cards 2 and 3, in which we applied the same rule to the same Card, and ended up in two different places). The lesson of all this? Implication is a lot of fun.

Finally, in Card 6, we apply Implication to Card 5's inner attic Pyramid, changing it into a Paragon. This frightens the orange Character, whose chamber has dropped from being an attic in

a Pyramid to being simply the left chamber in a Paragon. (Notice that we could have achieved the same result by applying Implication to Card 4, instead of Card 5, since Cards 4 and 5 are identical.)

2.2: The Power of Disjunction Introduction (A Landfill/Inference Rule)

The motto for this rule is, "You can have a magic Paragon whenever you want!" If you have any Character or Chambergon that "owns" a Card (i.e., that is the outermost Piece in the Card), you can put it in the left chamber of a Paragon and stick absolutely whatever you want in the right chamber.

Restriction: This rule does not work for anything that is inside another Chambergon. It can only be applied to Characters and Chambergons that are not inside other Chambergons. Also, it *can* be applied to Characters and Chambergons that have Clubs, but the Club has to stay with the thing that has the Club. It can't be transferred to the new, magic Chambergon (without using De Morgan's Theorem, that is).

The logical justification for this rule is as follows. The outermost piece in a Card represents an asserted proposition. That is, it stands for a proposition you are claiming is true. However, if you have a true proposition, and you join it to another proposition using disjunction, the resulting (more complex) proposition will also automatically be true. Take the following example, for instance.

It is currently snowing outside my apartment. That is a true proposition (at least at the moment). Now, imagine that I add to that proposition the following additional proposition, "A yeti just walked past my window." That proposition happens, unfortunately, to be false, and if I united it with, "It is currently snowing outside my apartment," using a conjunction, that would make the entire proposition ("It is currently snowing outside my apartment *and* a yeti just walked past my window") false. That's how conjunctions work. Both the conjuncts (both the component propositions) have to be true for the whole thing to be true.

But disjunction is different. If I take the true statement, ("It is currently snowing outside my apartment," and unite it with, "A yet just walked past my window," using disjunction, the whole resulting proposition will be true. That's how disjunction works. So long as one of the component propositions of a disjunction is true, the whole thing is true. (After all, I would have basically said, "It's either snowing outside my apartment, or a yeti just walked past my window; take your pick; at least one of those things is true." And at least one of those things *is* true—just not the interesting one.)

So, you can take any true statement and unite it to another statement using a disjunction, and the whole resulting statement will end up being true too *even if the proposition you added is false*. And that means we can use Disjunction Introduction to introduce brand new pieces into our games—pieces that are as simple or as complex as we like. Give us something in a Card by itself, and in the next Card we'll give you a Paragon containing the thing from the first Card in its left chamber and *whatever we want* in the right chamber. Then we'll put "Disjunction Introduction" in the Card's "Using" box, and look very smug.

"But doesn't that mean you can just make stuff up, and call it 'logical'?" you wisely ask. And the answer is, "Yes." From a true statement like, "Healthy grass is usually green," we can draw the "conclusion": "Healthy grass is usually green *or* Martians not only invented the internet but have perfected time travel to the extent that we are all our own grandparents, just living in an alternate dimension where healthy grass is usually not green, but orange, and the Beatles are joint Presidents of the World." The fact that the second "half" of that proposition (from "Martians" to "World") is completely made up and almost certainly false is irrelevant. The first "half" is true, and that's all it takes to make the whole disjunctive statement true.

"But that means the second half of the disjunction is just completely irrelevant," you complain. And that's true, *so long as the first half is true*. If the first component proposition inside a larger conjunctive statement is true, then the whole thing is automatically true, no matter what the truth vale of the second component proposition is. Computer programming languages like Java take advantage of this fact to make things go faster. If you write, "A || B" and the computer discovers that "A" is true, then it won't even check to see if "B" is true since what matters is whether the whole thing is true, and it already knows the whole thing is true since the first part was. This allows the computer to skip half the work of figuring out what to do next, saving itself—and the user—time.

You can be mean, however, and write "A | B" instead of "A || B." Then the computer will check both "A" and "B" for truth, even if it doesn't have to. But that is almost always a stupid idea, and everyone writes "A || B" to give the computer permission to save everyone time. (NOTE: Something similar is true of "A && B." If it turns out that "A" is false, you don't even have to check "B." The whole thing is automatically false. If you write "A & B," the computer will go ahead and check "B" anyway. But if you write "A && B," this tells the computer it has your permission not to worry about "B" if "A" is false. After all, even if "B" turned out to be true, "A && B" as a whole would be false because one of its parts is false.)

So, the basic idea behind Disjunction Introduction is built into (some) the programming languages that run our computers. But if you still don't believe that Disjunction Introduction is a legitimate logical move, take a look at the truth table on the next page.

!!!WARNING: TECHNICAL STUFF!!!

!!!WARNING: TECHNICAL STUFF!!! !!!WARNING: TECHNICAL STUFF!!! !!!WARNING: TECHNICAL STUFF!!!

A	B	C	D	(A	||	B)	(A	||	C)	(A	||	D)	(A	||	(!B	&&	(C	->	!D)))
T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	FT	F	T	F	FT
T	T	T	F	T	T	T	T	T	T	T	T	F	T	T	FT	F	T	T	TF
T	T	F	T	T	T	T	T	T	F	T	T	T	T	T	FT	F	F	T	FT
T	T	F	F	T	T	T	T	T	F	T	T	F	T	T	FT	F	F	T	TF
T	F	T	T	T	T	F	T	T	T	T	T	T	T	T	TF	F	T	F	FT
T	F	T	F	T	T	F	T	T	T	T	T	F	T	T	TF	T	T	T	TF
T	F	F	T	T	T	F	T	T	F	T	T	T	T	T	TF	T	F	T	FT
T	F	F	F	T	T	F	T	T	F	T	T	F	T	T	TF	T	F	T	TF
F	T	T	T	F	T	T	F	T	T	F	T	T	F	F	FT	F	T	F	FT
F	T	T	F	F	T	T	F	T	T	F	F	F	F	F	FT	F	T	T	TF
F	T	F	T	F	T	T	F	F	F	F	T	T	F	F	FT	F	F	T	FT
F	T	F	F	F	T	T	F	F	F	F	F	F	F	F	FT	F	F	T	TF
F	F	T	T	F	F	F	F	T	T	F	T	T	F	F	TF	F	T	F	FT
F	F	T	F	F	F	F	F	T	T	F	F	F	F	F	TF	T	T	T	TF
F	F	F	T	F	F	F	F	F	F	F	T	T	F	F	TF	T	F	T	FT
F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	TF	T	F	T	TF

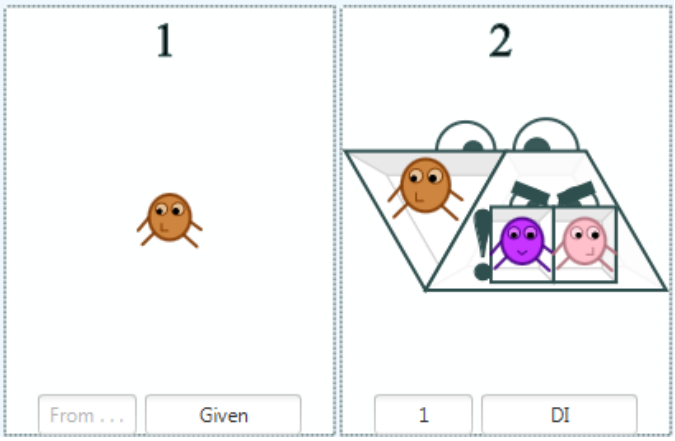
That is the truth table for "A" when it is disjoined with three other propositions, "B," "C," and "D," as well as with the complex, made-up proposition "(!B && (C -> !D))." So long as you unite "A" with these other propositions using disjunction, this truth table shows that the result will be true whenever "A" is true. Allow me to circle the relevant portions of the table for you

A	B	C	D	(A	||	B)	(A	||	C)	(A	||	D)	(A	||	(!B	&&	(C	->	!D)))
T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	FT	F	T	F	FT
T	T	T	F	T	T	T	T	T	T	T	T	F	T	T	FT	F	T	T	TF
T	T	F	T	T	T	T	T	T	F	T	T	T	T	T	FT	F	F	T	FT
T	T	F	F	T	T	T	T	T	F	T	T	F	T	T	FT	F	F	T	TF
T	F	T	T	T	T	F	T	T	T	T	T	T	T	T	TF	F	T	F	FT
T	F	T	F	T	T	F	T	T	T	T	T	F	T	T	TF	T	T	T	TF
T	F	F	T	T	T	F	T	T	F	T	T	T	T	T	TF	T	F	T	FT
T	F	F	F	T	T	F	T	T	F	T	T	F	T	T	TF	T	F	T	TF
F	T	T	T	F	T	T	F	T	T	F	T	T	F	F	FT	F	T	F	FT
F	T	T	F	F	T	T	F	T	T	F	F	F	F	F	FT	F	T	T	TF
F	T	F	T	F	T	T	F	F	F	F	T	T	F	F	FT	F	F	T	FT
F	T	F	F	F	T	T	F	F	F	F	F	F	F	F	FT	F	F	T	TF
F	F	T	T	F	F	F	F	T	T	F	T	T	F	F	TF	F	T	F	FT
F	F	T	F	F	F	F	F	T	T	F	F	F	F	F	TF	T	T	T	TF
F	F	F	T	F	F	F	F	F	F	F	T	T	F	F	TF	T	F	T	FT
F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	TF	T	F	T	TF

"A" is true on the first eight lines of the table *and so is every proposition of which "A" forms a part*. When "A" is false, all bets are off. Sometimes the whole proposition is true and sometimes not. But when "A" is true, the whole thing is true, no matter what you stick on the other side of the disjunction symbol. And that's all the Disjunction Introduction rule needs. After all, the rules

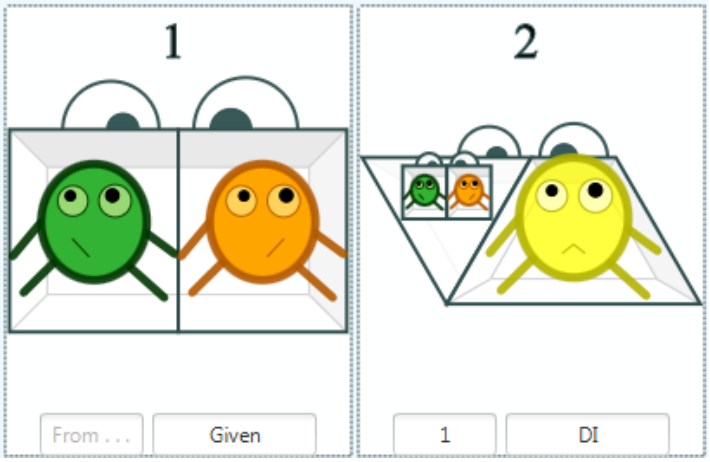
of logic are supposed to tell you what conclusions you can legitimately draw from what premises; they are supposed to help you engage in valid deductions. But an argument is valid if, and only if, when the premises are true, the conclusion must be true as well. So, we can legitimately reason from the truth of "A" to the truth of "A || [literally any proposition whatsoever]."

Example 1



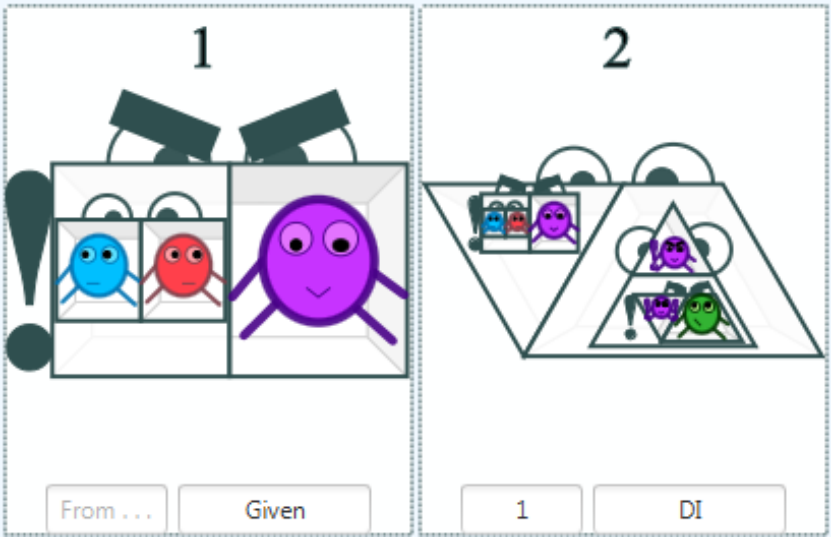
Paragons, man. I tell you. They are weird. Where do they come from? Where do they get the thing in the right chamber? It's magic, I tell you!

Example 2



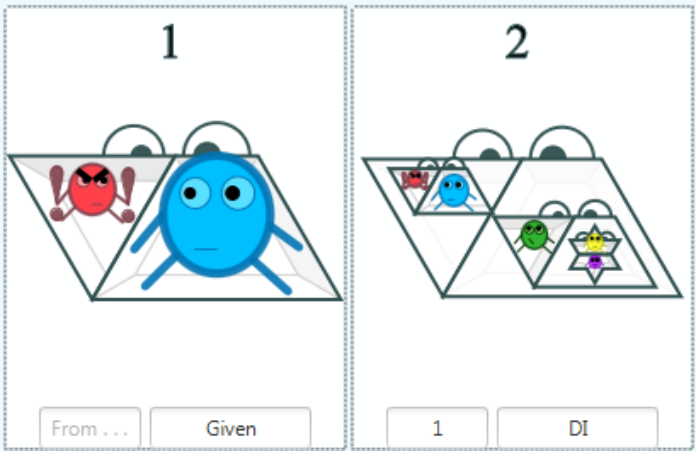
In this example, we think that having the Paragon in Card 2 would be useful for winning the game. Maybe we'll be able to use the yellow Character later to make some strategic move. Whatever the case, we use the magical Disjunction Introduction rule on Card 1, to produce what we have in Card 2.

Example 3



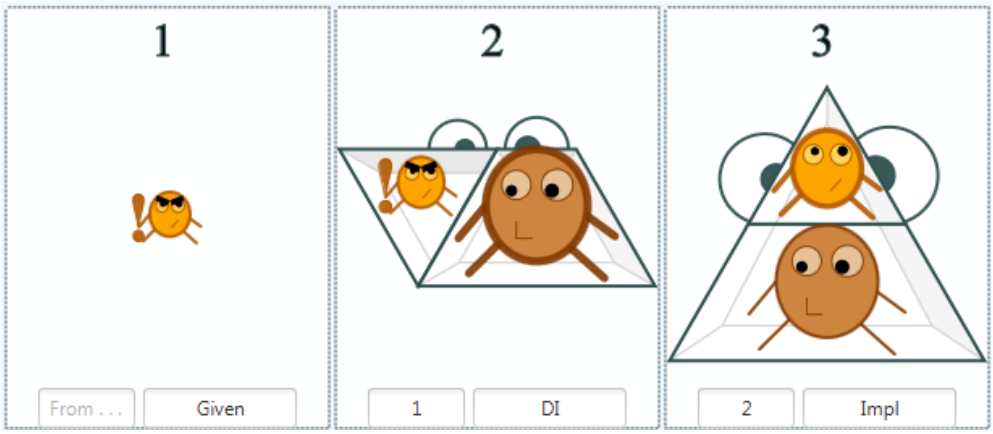
The purpose of this example is twofold. First, it shows that the Club in Card 1 has to stick with its Package, even after we use Disjunction Introduction to create the Paragon. Second, it shows that you can introduce practically whatever you want in the right chamber of the Paragon, when you use Disjunction Introduction. Of course, in the middle of a game, you'll never just introduce something randomly. You'll always use Disjunction Introduction to introduce a Paragon with very specific contents, because you at least suspect, and often *know*, that you'll be able to use it to win the game.

Example 4



In this example, we use a Paragon to anchor the introduction of a new Paragon, containing a Pulsar in its second chamber. Why? Because we can, and because Pulsars look cool. Also, because we'll need that Pulsar later in the Game, probably.

Example 5



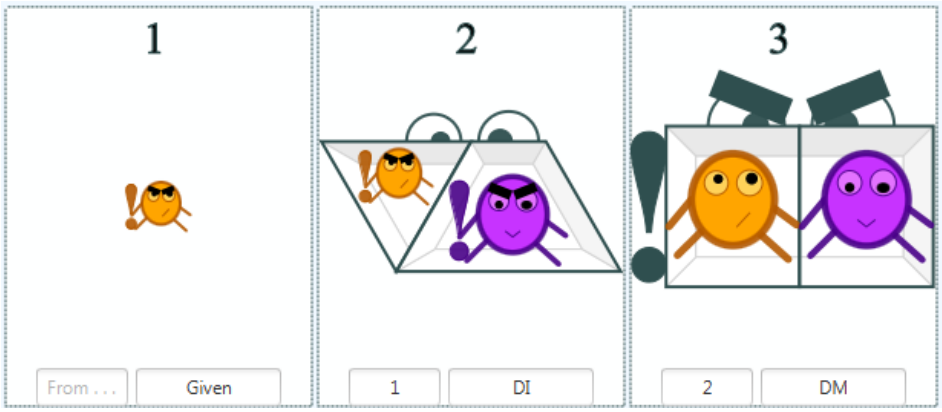
This example actually employs two different rules: Disjunction Introduction and Implication. I wanted to include it, however, to show a way in which Implication (with Disjunction Introduction) can be used to heal and comfort, in addition to being used to introduce fear and loathing. We start with just an orange Character with a Club. The Club alerts us to the fact that the orange Character is frightened and angry, and thus to the fact that the orange Character must have fallen out of the attic of some Pyramid, before we came across it. But how can we get it back into the attic of a Pyramid?

Well, it's actually very easy to get anything into the attic of a Pyramid. All you have to do is get it by itself in some Card, then use Disjunction Introduction on it to get it into a Paragon, and then use Implication on the Paragon. In the process, of course, the thing will gain a Club—if it doesn't have one already—or lose a Club—if it does already have one—but you'll have gotten it into the attic of a Pyramid.

So, this is what we do to help out the poor frightened and angry orange Character in Card 1. We introduce a magic Paragon, and put the Character in the left chamber. Then we flip up the left chamber so it becomes the attic of a Pyramid. Now the orange Character is back home, and can finally relax, letting all its negative emotions fade away (along with its Club).

Now, you might ask why I put a brown Character in the right chamber of the Paragon. The answer, presumably, that I'll need it later in the game (rather than, say, a Pulsar with two Clubs containing a dinosaur). But remember, with Disjunction Introduction, you can put *anything* in the right chamber.

Example 6



Here's a similar example, but this time we introduce a purple Character with a Club in Card 2. This allows us to use De Morgan's Theorem to produce Card 3. We could've used Implication, of course, but sometimes what you need from a Paragon is not a Pyramid, but a Parcel.

2.3: Activation Games

2.3.0: Introduction

The games we have been playing to this point are the equivalent of "proofs in propositional logic." However, you will have noticed that I've used truth tables to defend some of the rules of the game. And, as it turns out, "doing truth tables" is a significant part of a normal symbolic logic course. The problem with this, however, is the symbols, and you can see how . . . symbolic (?) . . . truth tables can get if you look back at page 77.

Fortunately for us, therefore, Chambergon Battle Logic has its own version of truth tables, and it's much more colorful than what you'd get in a normal logic text. Before I show you what they look like, however, I need to explain the point of truth tables. And that means we have to do two sub-sections of horrendously tedious and symbol-heavy stuff. So, if you find your brain just shutting off at some point, skip ahead to 2.3.3 (p. 102).

2.3.1: The Basic Truth Tables

When you have a simple proposition (like, "It's snowing outside"), symbolic logic will treat it as either true or false, just by itself. When you have a complex proposition (like, "It's snowing outside if I look, but not when I don't"), however, symbolic logic treats the truth value of the whole proposition as arising from the truth values of its parts. The problem is that we, as logicians, don't know whether the parts of a given proposition are true or false. *Most of the time we don't even know what the parts are*. All we have are A's and B's and C's and so on.

Instead of saying that a complex proposition is true, or that it is false, therefore, we say that it is true *when its constituent propositions have a particular combination of truth values*, and false *when its constituent propositions have other combinations of truth values*. To do this, we list all the possible combinations in a table. Here are the basic truth tables.

Negation:

A	**!A**
T	FT
F	TF

This says that the complex proposition "!A" is false if "A" is true, and true if "A" is false. How do we know if "A" is true? We don't. We don't even know what "A" is. We just know it's *some proposition or other*. So, we give it a column and list the possibilities. On the first line, we list the possibility that "A" is true. Then, in the second column, we show that if "A" is true, the "!" makes it false. Then, on the second line, we list the possibility that "A" is false. On the same line, we show in the second column that if "A" is false, then the "!" turns the whole statement true.

Conjunction:

A	**B**	**(A**	**&&**	**B)**
T	T	T	T	T
T	F	T	F	F
F	T	F	F	T
F	F	F	F	F

This says that the complex proposition "(A && B)" is true only if both "A" and "B" are true. If either "A" or "B" is false, then the whole thing is false. Again, we don't know what "A" and "B" are (they just stand for some proposition and some other proposition), so we don't know if they are true or false. All we know are the possibilities. Either they are both true (the top line of the table), or they are both false (the bottom line of the table), or one is true while the other is false (the middle two lines of the table). The column under the "&&" shows "the result" or "the answer" for conjunctions; it shows what the truth value of the entire proposition is, given the truth values of the parts of the proposition.

Disjunction:

A	**B**	**(A**	**||**	**B)**
T	T	T	T	T
T	F	T	T	F
F	T	F	T	T
F	F	F	F	F

This says that the complex proposition "(A || B)" is true whenever either "A" or "B" is true. It only turns false if both "A" and "B" are false. (Again, we don't know when "A" and "B" are either true or false, since we do not even know what propositions they stand for. We just know that they are either true or false, and thus are either both truth, both false, or one of each.)

Conditional:

A	**B**	**(A**	**->**	**B)**
T	T	T	T	T
T	F	T	F	F
F	T	F	T	T
F	F	F	T	F

This says that the complex proposition "(A -> B)" is false if "A" is true and "B" is false (see line 2), but is true in all other cases.

Biconditional:

A	**B**	**(A**	**<->**	**B)**
T	T	T	T	T
T	F	T	F	F
F	T	F	F	T
F	F	F	T	F

This says that the complex proposition "(A <-> B)" is true if the truth values of "A" and "B" match, but is false if they differ.

2.3.2: Filling in More Complex Truth Tables

From these basic truth tables, logicians are then able to build more complex truth tables for propositions like "((A || F) -> (!(B && (A && Q)) <-> !B))." Here's how they do it.

First, they count the number of letters in the proposition, leaving out repeats. So, all the A's count as one, all the B's count as one, all the C's count as one, and so on. In the bizarre proposition, "((A || F) -> (!(B && (A && Q)) <-> !B))," then, there are four letters, "A," "B," "F," and "Q." (The other two letters are a repeated "A" and a repeated "B," which don't count as their own letters.)

Next, logicians will write out the letters from the proposition on a line, in alphabetical order, like this:

A	**B**	**F**	**Q**

Then, they split the letters into columns by drawing lines between them.

A	**B**	**F**	**Q**

But to figure out how long those lines should be, they need to figure out how many rows will be in the truth table. To figure that out requires a tiny bit of math. Start at the rightmost letter (in our case, that is "Q"). Call it "2." Now, move to the next letter to the left, and double that number. ("F" is "4," because "Q" was "2," and four is the double of two.) Now move to the next letter and double again. ("B" is "8," because "F" was "4," and eight is twice two.). Now move to the next letter and double again. ("A" is "16," because "B" was "8," and sixteen is two times eight.)

When you have only two letters, you end up with four rows. When you have three, you end up with eight rows. And when you have four letters, like we do, you need sixteen rows. (If we had five letters, we'd have to double again, and thus would need 32 rows.) The reason for this is that

each letter represents a proposition, and every proposition is either true or false. That's two possibilities. So, if you have two propositions, each of which has two possible truth vales, that creates 2 × 2 (= 4) possible truth value combinations. So, you'll need four lines to write out those combinations. And when you have three propositions, each of which has two possible truth values, that creates 2 × 2 × 2 (= 8) possible truth value combinations. So, you'll need eight lines to write out those combinations. And when you have four propositions (like we do), each of which has two possible truth values, that creates 2 × 2 × 2 × 2 (=16) possible truth value combinations. So, we'll need sixteen lines in our truth table.

In fact, there's a general mathematical equation to tell you how many lines you need. It looks like this: "$L = 2^n$." That says, "The number of lines you need ('L') is equal to the number two ('2'), raised to the number of letters ('n') in the proposition." In other words, the number of lines you need is equal to 2, multiplied times itself the same number of times as there are letters in the complex proposition you want to analyze. In our case, the equation would be "$L = 2^4 = 2 \times 2 \times 2 \times 2 = 16$," so we need to add sixteen lines to our table

	A	**B**	**F**	**Q**
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				

Then, we need to fill in all the possible truth value combinations. Start with the rightmost column, and write "T" on line 1, "F," on line 2, "T" on line 3, "F" on line 4, and so on, changing from "T" to "F" every line.

	A	B	F	Q
1				T
2				F
3				T
4				F
5				T
6				F
7				T
8				F
9				T
10				F
11				T
12				F
13				T
14				F
15				T
16				F

Now, move a column to your left, and double the number of T's and F's. Instead of having one T, then one F, use two T's in a row, then two F's in a row. Like this:

	A	B	F	Q
1			T	T
2			T	F
3			F	T
4			F	F
5			T	T
6			T	F
7			F	T
8			F	F
9			T	T
10			T	F
11			F	T
12			F	F
13			T	T
14			T	F
15			F	T
16			F	F

Then, move another column to your left, and double the number of T's and F's again. Instead of having two T's, then two F's, use four T's, then four F's.

	A	B	F	Q
1		T	T	T
2		T	T	F
3		T	F	T
4		T	F	F
5		F	T	T
6		F	T	F
7		F	F	T
8		F	F	F
9		T	T	T
10		T	T	F
11		T	F	T
12		T	F	F
13		F	T	T
14		F	T	F
15		F	F	T
16		F	F	F

And finally, move to your left one more time, and double the number of T's and F's again. Instead of four T's then four F's, use eight T's, then eight F's.

	A	B	F	Q
1	T	T	T	T
2	T	T	T	F
3	T	T	F	T
4	T	T	F	F
5	T	F	T	T
6	T	F	T	F
7	T	F	F	T
8	T	F	F	F
9	F	T	T	T
10	F	T	T	F
11	F	T	F	T
12	F	T	F	F
13	F	F	T	T
14	F	F	T	F
15	F	F	F	T
16	F	F	F	F

On each line of the table, we have now written a particular combination of the truth values for the four basic propositions we'll be working with. We have no idea what those propositions are. All we know is that we're calling them "A," "B," "F," and "Q." And we don't know whether they are true or not. We just know either they're all true (line 1), all false (line 16), or something in between (lines 2 – 15).

Next, we want to know what will happen to the truth value of the entire, complex proposition, "((A || F) -> (!(B && (A && Q)) <-> !B))," when "A," "B," "F," and "Q" have the truth values on each line of the table. So, we place that complex proposition at the top of the table, giving every part of the proposition that can be true or false its own column.

	A	**B**	**F**	**Q**	**((A**	**||**	**F)**	**->**	**(!**	**(B**	**&&**	**(A**	**&&**	**Q))**	**<->**	**!**	**B))**
1	T	T	T	T													
2	T	T	T	F													
3	T	T	F	T													
4	T	T	F	F													
5	T	F	T	T													
6	T	F	T	F													
7	T	F	F	T													
8	T	F	F	F													
9	F	T	T	T													
10	F	T	T	F													
11	F	T	F	T													
12	F	T	F	F													
13	F	F	T	T													
14	F	F	T	F													
15	F	F	F	T													
16	F	F	F	F													

Then, we "copy over" the truth value for each letter in the proposition on each line. For example, on line 1, "A" is true. You can see this by looking at line 1 of its column on the left side of the table. So, on line 1, we put a "T" under all the other A's.

	A	**B**	**F**	**Q**	**((A**	**||**	**F)**	**->**	**(!**	**(B**	**&&**	**(A**	**&&**	**Q))**	**<->**	**!**	**B))**
1	T	T	T	T	T							T					
2	T	T	T	F													
3	T	T	F	T													
4	T	T	F	F													
5	T	F	T	T													
6	T	F	T	F													
7	T	F	F	T													
8	T	F	F	F													
9	F	T	T	T													
10	F	T	T	F													
11	F	T	F	T													
12	F	T	F	F													
13	F	F	T	T													
14	F	F	T	F													
15	F	F	F	T													
16	F	F	F	F													

One line 2, the A's are true again. So we copy over another "T" for each "A."

	A	B	F	Q	((A	||	F)	->	(!	(B	&&	(A	&&	Q))	<->	!	B))
1	T	T	T	T	T							T					
2	T	T	T	F	T							T					
3	T	T	F	T													
4	T	T	F	F													
5	T	F	T	T													
6	T	F	T	F													
7	T	F	F	T													
8	T	F	F	F													
9	F	T	T	T													
10	F	T	T	F													
11	F	T	F	T													
12	F	T	F	F													
13	F	F	T	T													
14	F	F	T	F													
15	F	F	F	T													
16	F	F	F	F													

In fact, the A's are true on the first eight rows:

	A	B	F	Q	((A	||	F)	->	(!	(B	&&	(A	&&	Q))	<->	!	B))
1	T	T	T	T	T							T					
2	T	T	T	F	T							T					
3	T	T	F	T	T							T					
4	T	T	F	F	T							T					
5	T	F	T	T	T							T					
6	T	F	T	F	T							T					
7	T	F	F	T	T							T					
8	T	F	F	F	T							T					
9	F	T	T	T													
10	F	T	T	F													
11	F	T	F	T													
12	F	T	F	F													
13	F	F	T	T													
14	F	F	T	F													
15	F	F	F	T													
16	F	F	F	F													

And the A's are false on the last eight rows:

	A	B	F	Q	((A	||	F)	->	(!	(B	&&	(A	&&	Q))	<->	!	B))
1	T	T	T	T	T							T					
2	T	T	T	F	T							T					
3	T	T	F	T	T							T					
4	T	T	F	F	T							T					
5	T	F	T	T	T							T					
6	T	F	T	F	T							T					
7	T	F	F	T	T							T					
8	T	F	F	F	T							T					
9	F	T	T	T	F							F					
10	F	T	T	F	F							F					
11	F	T	F	T	F							F					
12	F	T	F	F	F							F					
13	F	F	T	T	F							F					
14	F	F	T	F	F							F					
15	F	F	F	T	F							F					
16	F	F	F	F	F							F					

Now that we have used T's and F's in the first "A" column to fill in the other "A" columns, we can cross them off.

	A	B	F	Q	((A	||	F)	->	(!	(B	&&	(A	&&	Q))	<->	!	B))
1	~~T~~	T	T	T	T							T					
2	~~T~~	T	T	F	T							T					
3	~~T~~	T	F	T	T							T					
4	~~T~~	T	F	F	T							T					
5	~~T~~	F	T	T	T							T					
6	~~T~~	F	T	F	T							T					
7	~~T~~	F	F	T	T							T					
8	~~T~~	F	F	F	T							T					
9	~~F~~	T	T	T	F							F					
10	~~F~~	T	T	F	F							F					
11	~~F~~	T	F	T	F							F					
12	~~F~~	T	F	F	F							F					
13	~~F~~	F	T	T	F							F					
14	~~F~~	F	T	F	F							F					
15	~~F~~	F	F	T	F							F					
16	~~F~~	F	F	F	F							F					

Then we do the same thing for the "B" columns that we did for the "A" columns. Every column under a "B" should look identical to the column we've already filled in for "B."

	A	B	F	Q	((A	||	F)	->	(!	(B	&&	(A	&&	Q))	<->	!	B))
1	~~T~~	T	T	T	T					T		T					T
2	~~T~~	T	T	F	T					T		T					T
3	~~T~~	T	F	T	T					T		T					T
4	~~T~~	T	F	F	T					T		T					T
5	~~T~~	F	T	T	T					F		T					F
6	~~T~~	F	T	F	T					F		T					F
7	~~T~~	F	F	T	T					F		T					F
8	~~T~~	F	F	F	T					F		T					F
9	~~F~~	T	T	T	F					T		F					T
10	~~F~~	T	T	F	F					T		F					T
11	~~F~~	T	F	T	F					T		F					T
12	~~F~~	T	F	F	F					T		F					T
13	~~F~~	F	T	T	F					F		F					F
14	~~F~~	F	T	F	F					F		F					F
15	~~F~~	F	F	T	F					F		F					F
16	~~F~~	F	F	F	F					F		F					F

Having used all the T's and F's under the first "B" column to fill in the other "B" columns, we can cross them off.

	A	B	F	Q	((A	||	F)	->	(!	(B	&&	(A	&&	Q))	<->	!	B))
1	~~T~~	~~T~~	T	T	T					T		T					T
2	~~T~~	~~T~~	T	F	T					T		T					T
3	~~T~~	~~T~~	F	T	T					T		T					T
4	~~T~~	~~T~~	F	F	T					T		T					T
5	~~T~~	~~F~~	T	T	T					F		T					F
6	~~T~~	~~F~~	T	F	T					F		T					F
7	~~T~~	~~F~~	F	T	T					F		T					F
8	~~T~~	~~F~~	F	F	T					F		T					F
9	~~F~~	~~T~~	T	T	F					T		F					T
10	~~F~~	~~T~~	T	F	F					T		F					T
11	~~F~~	~~T~~	F	T	F					T		F					T
12	~~F~~	~~T~~	F	F	F					T		F					T
13	~~F~~	~~F~~	T	T	F					F		F					F
14	~~F~~	~~F~~	T	F	F					F		F					F
15	~~F~~	~~F~~	F	T	F					F		F					F
16	~~F~~	~~F~~	F	F	F					F		F					F

Now that we've done the A's and B's, we can move on to the column for "F." (It's important to notice that at the top of the table, "F" stands for a proposition, like "Frank is funny." Inside the table, however, "F" is the opposite of "T," and doesn't stands for "The proposition above is *false*.") We make sure the blank column under "F" gets filled in exactly like the filled-in column under the other "F."

	A	B	F	Q	((A	||	F)	->	(!	(B	&&	(A	&&	Q))	<->	!	B))
1	~~T~~	~~T~~	T	T	T		T			T		T					T
2	~~T~~	~~T~~	T	F	T		T			T		T					T
3	~~T~~	~~T~~	F	T	T		F			T		T					T
4	~~T~~	~~T~~	F	F	T		F			T		T					T
5	~~T~~	~~F~~	T	T	T		T			F		T					F
6	~~T~~	~~F~~	T	F	T		T			F		T					F
7	~~T~~	~~F~~	F	T	T		F			F		T					F
8	~~T~~	~~F~~	F	F	T		F			F		T					F
9	~~F~~	~~T~~	T	T	F		T			T		F					T
10	~~F~~	~~T~~	T	F	F		T			T		F					T
11	~~F~~	~~T~~	F	T	F		F			T		F					T
12	~~F~~	~~T~~	F	F	F		F			T		F					T
13	~~F~~	~~F~~	T	T	F		T			F		F					F
14	~~F~~	~~F~~	T	F	F		T			F		F					F
15	~~F~~	~~F~~	F	T	F		F			F		F					F
16	~~F~~	~~F~~	F	F	F		F			F		F					F

Having used all the T's and F's under the first "F" column, we cross them out.

	A	B	F	Q	((A	||	F)	->	(!	(B	&&	(A	&&	Q))	<->	!	B))
1	~~T~~	~~T~~	~~T~~	T	T		T			T		T					T
2	~~T~~	~~T~~	~~T~~	F	T		T			T		T					T
3	~~T~~	~~T~~	~~F~~	T	T		F			T		T					T
4	~~T~~	~~T~~	~~F~~	F	T		F			T		T					T
5	~~T~~	~~F~~	~~T~~	T	T		T			F		T					F
6	~~T~~	~~F~~	~~T~~	F	T		T			F		T					F
7	~~T~~	~~F~~	~~F~~	T	T		F			F		T					F
8	~~T~~	~~F~~	~~F~~	F	T		F			F		T					F
9	~~F~~	~~T~~	~~T~~	T	F		T			T		F					T
10	~~F~~	~~T~~	~~T~~	F	F		T			T		F					T
11	~~F~~	~~T~~	~~F~~	T	F		F			T		F					T
12	~~F~~	~~T~~	~~F~~	F	F		F			T		F					T
13	~~F~~	~~F~~	~~T~~	T	F		T			F		F					F
14	~~F~~	~~F~~	~~T~~	F	F		T			F		F					F
15	~~F~~	~~F~~	~~F~~	T	F		F			F		F					F
16	~~F~~	~~F~~	~~F~~	F	F		F			F		F					F

Finally, we have to fill in the "Q" column.

	A	B	F	Q	((A	||	F)	->	(!	(B	&&	(A	&&	Q))	<->	!	B))
1	~~T~~	~~T~~	~~T~~	T	T		T			T		T		T			T
2	~~T~~	~~T~~	~~T~~	F	T		T			T		T		F			T
3	~~T~~	~~T~~	~~F~~	T	T		F			T		T		T			T
4	~~T~~	~~T~~	~~F~~	F	T		F			T		T		F			T
5	~~T~~	~~F~~	~~T~~	T	T		T			F		T		T			F
6	~~T~~	~~F~~	~~T~~	F	T		T			F		T		F			F
7	~~T~~	~~F~~	~~F~~	T	T		F			F		T		T			F
8	~~T~~	~~F~~	~~F~~	F	T		F			F		T		F			F
9	~~F~~	~~T~~	~~T~~	T	F		T			T		F		T			T
10	~~F~~	~~T~~	~~T~~	F	F		T			T		F		F			T
11	~~F~~	~~T~~	~~F~~	T	F		F			T		F		T			T
12	~~F~~	~~T~~	~~F~~	F	F		F			T		F		F			T
13	~~F~~	~~F~~	~~T~~	T	F		T			F		F		T			F
14	~~F~~	~~F~~	~~T~~	F	F		T			F		F		F			F
15	~~F~~	~~F~~	~~F~~	T	F		F			F		F		T			F
16	~~F~~	~~F~~	~~F~~	F	F		F			F		F		F			F

And, having used all the T's and F's in the first "Q" column to fill in the other "Q" column, we cross them off.

	A	B	F	Q	((A	||	F)	->	(!	(B	&&	(A	&&	Q))	<->	!	B))
1	~~T~~	~~T~~	~~T~~	~~T~~	T		T			T		T		T			T
2	~~T~~	~~T~~	~~T~~	~~F~~	T		T			T		T		F			T
3	~~T~~	~~T~~	~~F~~	~~T~~	T		F			T		T		T			T
4	~~T~~	~~T~~	~~F~~	~~F~~	T		F			T		T		F			T
5	~~T~~	~~F~~	~~T~~	~~T~~	T		T			F		T		T			F
6	~~T~~	~~F~~	~~T~~	~~F~~	T		T			F		T		F			F
7	~~T~~	~~F~~	~~F~~	~~T~~	T		F			F		T		T			F
8	~~T~~	~~F~~	~~F~~	~~F~~	T		F			F		T		F			F
9	~~F~~	~~T~~	~~T~~	~~T~~	F		T			T		F		T			T
10	~~F~~	~~T~~	~~T~~	~~F~~	F		T			T		F		F			T
11	~~F~~	~~T~~	~~F~~	~~T~~	F		F			T		F		T			T
12	~~F~~	~~T~~	~~F~~	~~F~~	F		F			T		F		F			T
13	~~F~~	~~F~~	~~T~~	~~T~~	F		T			F		F		T			F
14	~~F~~	~~F~~	~~T~~	~~F~~	F		T			F		F		F			F
15	~~F~~	~~F~~	~~F~~	~~T~~	F		F			F		F		T			F
16	~~F~~	~~F~~	~~F~~	~~F~~	F		F			F		F		F			F

Now that the columns for the basic propositions (inside the huge, complex proposition we're trying to analyze) are filled in, we can start figuring out what to put in the other columns. To do this, however, we have to go in the right order.

First, find the innermost pair parentheses. In other words, find the pair of parentheses that is inside the most other pairs of parentheses. (If there are two that are buried equally deep, start with one, then do the other.)

((A || F) -> (! (B && (A && Q)) <-> ! B))

In our case, the innermost parentheses bind the proposition, "(A && Q)." That pair is inside three other pairs of parentheses, while the others are only inside two or one (or none, since the outermost parentheses are . . . outermost).

Once we've identified the innermost parentheses, we look for any exclamation marks attached to the letters inside those parentheses. In our case, there are none. So, our job is to fill in the column under the "&&" in "(A && Q)." Since this is a conjunctive proposition, we have to remember the rule for conjunctions: the conjunction is true as a whole only if both the conjuncts (i.e., both the constituent propositions) are true. If either conjunct is false, the whole thing is false. If we follow that rule, we get the following:

	A	**B**	**F**	**Q**	**((A**	**||**	**F)**	**->**	**(!**	**(B**	**&&**	**(A**	**&&**	**Q))**	**<->**	**!**	**B))**
1	~~T~~	~~T~~	~~T~~	~~T~~	T		T			T		T	T	T			T
2	~~T~~	~~T~~	~~T~~	~~F~~	T		T			T		T	F	F			T
3	~~T~~	~~T~~	~~F~~	~~T~~	T		F			T		T	T	T			T
4	~~T~~	~~T~~	~~F~~	~~F~~	T		F			T		T	F	F			T
5	~~T~~	~~F~~	~~T~~	~~T~~	T		T			F		T	T	T			F
6	~~T~~	~~F~~	~~T~~	~~F~~	T		T			F		T	F	F			F
7	~~T~~	~~F~~	~~F~~	~~T~~	T		F			F		T	T	T			F
8	~~T~~	~~F~~	~~F~~	~~F~~	T		F			F		T	F	F			F
9	~~F~~	~~T~~	~~T~~	~~T~~	F		T			T		F	F	T			T
10	~~F~~	~~T~~	~~T~~	~~F~~	F		T			T		F	F	F			T
11	~~F~~	~~T~~	~~F~~	~~T~~	F		F			T		F	F	T			T
12	~~F~~	~~T~~	~~F~~	~~F~~	F		F			T		F	F	F			T
13	~~F~~	~~F~~	~~T~~	~~T~~	F		T			F		F	F	T			F
14	~~F~~	~~F~~	~~T~~	~~F~~	F		T			F		F	F	F			F
15	~~F~~	~~F~~	~~F~~	~~T~~	F		F			F		F	F	T			F
16	~~F~~	~~F~~	~~F~~	~~F~~	F		F			F		F	F	F			F

Now, I would strongly suggest we cross off all the T's and F's we've used, so they don't distract us.

	A	B	F	Q	((A	‖	F)	->	(!	(B	&&	(A	&&	Q))	<->	!	B))
1	~~T~~	~~T~~	~~T~~	~~T~~	T		T			T		~~T~~	T	~~T~~			T
2	~~T~~	~~T~~	~~T~~	~~F~~	T		T			T		~~T~~	F	~~F~~			T
3	~~T~~	~~T~~	~~F~~	~~T~~	T		F			T		~~T~~	T	~~T~~			T
4	~~T~~	~~T~~	~~F~~	~~F~~	T		F			T		~~T~~	F	~~F~~			T
5	~~T~~	~~F~~	~~T~~	~~T~~	T		T			F		~~T~~	T	~~T~~			F
6	~~T~~	~~F~~	~~T~~	~~F~~	T		T			F		~~T~~	F	~~F~~			F
7	~~T~~	~~F~~	~~F~~	~~T~~	T		F			F		~~T~~	T	~~T~~			F
8	~~T~~	~~F~~	~~F~~	~~F~~	T		F			F		~~T~~	F	~~F~~			F
9	~~F~~	~~T~~	~~T~~	~~T~~	F		T			T		~~F~~	F	~~T~~			T
10	~~F~~	~~T~~	~~T~~	~~F~~	F		T			T		~~F~~	F	~~F~~			T
11	~~F~~	~~T~~	~~F~~	~~T~~	F		F			T		~~F~~	F	~~T~~			T
12	~~F~~	~~T~~	~~F~~	~~F~~	F		F			T		~~F~~	F	~~F~~			T
13	~~F~~	~~F~~	~~T~~	~~T~~	F		T			F		~~F~~	F	~~T~~			F
14	~~F~~	~~F~~	~~T~~	~~F~~	F		T			F		~~F~~	F	~~F~~			F
15	~~F~~	~~F~~	~~F~~	~~T~~	F		F			F		~~F~~	F	~~T~~			F
16	~~F~~	~~F~~	~~F~~	~~F~~	F		F			F		~~F~~	F	~~F~~			F

Our next job is to take care of any exclamation marks that are attached to the parentheses we just handled. In the current case, there are none, so we move on to the innermost parentheses of the propositions we haven't dealt with.

((A ‖ F) -> (! (B && (A && Q)) <-> ! B))

So, now we deal with the "(B && (A && Q))." Except we've already dealt with the "A" and "Q," so we deal with whatever is under the "&&" between them instead.

First, we check for any exclamation marks inside the parentheses. There are none, so we take the truth vale under "B," and ask what would be the result if it were "conjoined" with the truth value under the "&&" between "A" and "Q." We write the result under the "&&" beside the "B."

	A	B	F	Q	((A	||	F)	->	(!	(B	&&	(A	&&	Q))	<->	!	B))
1	~~T~~	~~T~~	~~T~~	~~T~~	T		T			T	T	~~T~~	T	~~T~~			T
2	~~T~~	~~T~~	~~T~~	~~F~~	T		T			T	F	~~T~~	F	~~F~~			T
3	~~T~~	~~T~~	~~F~~	~~T~~	T		F			T	T	~~T~~	T	~~T~~			T
4	~~T~~	~~T~~	~~F~~	~~F~~	T		F			T	F	~~T~~	F	~~F~~			T
5	~~T~~	~~F~~	~~T~~	~~T~~	T		T			F	F	~~T~~	T	~~T~~			F
6	~~T~~	~~F~~	~~T~~	~~F~~	T		T			F	F	~~T~~	F	~~F~~			F
7	~~T~~	~~F~~	~~F~~	~~T~~	T		F			F	F	~~T~~	T	~~T~~			F
8	~~T~~	~~F~~	~~F~~	~~F~~	T		F			F	F	~~T~~	F	~~F~~			F
9	~~F~~	~~T~~	~~T~~	~~T~~	F		T			T	F	~~F~~	F	~~T~~			T
10	~~F~~	~~T~~	~~T~~	~~F~~	F		T			T	F	~~F~~	F	~~F~~			T
11	~~F~~	~~T~~	~~F~~	~~T~~	F		F			T	F	~~F~~	F	~~T~~			T
12	~~F~~	~~T~~	~~F~~	~~F~~	F		F			T	F	~~F~~	F	~~F~~			T
13	~~F~~	~~F~~	~~T~~	~~T~~	F		T			F	F	~~F~~	F	~~T~~			F
14	~~F~~	~~F~~	~~T~~	~~F~~	F		T			F	F	~~F~~	F	~~F~~			F
15	~~F~~	~~F~~	~~F~~	~~T~~	F		F			F	F	~~F~~	F	~~T~~			F
16	~~F~~	~~F~~	~~F~~	~~F~~	F		F			F	F	~~F~~	F	~~F~~			F

In the column we just filled in, only lines 1 and 2 have a “T.” That is because on all the other lines, either “B” is false or the “&&” between “A” and “Q” is false.

	A	B	F	Q	((A	||	F)	->	(!	(B	&&	(A	&&	Q))	<->	!	B))
1	~~T~~	~~T~~	~~T~~	~~T~~	T		T			T	T	~~T~~	T	~~T~~			T
2	~~T~~	~~T~~	~~T~~	~~F~~	T		T			T	F	~~T~~	F	~~F~~			T
3	~~T~~	~~T~~	~~F~~	~~T~~	T		F			T	T	~~T~~	T	~~T~~			T
4	~~T~~	~~T~~	~~F~~	~~F~~	T		F			T	F	~~T~~	F	~~F~~			T
5	~~T~~	~~F~~	~~T~~	~~T~~	T		T			F	F	~~T~~	T	~~T~~			F
6	~~T~~	~~F~~	~~T~~	~~F~~	T		T			F	F	~~T~~	F	~~F~~			F
7	~~T~~	~~F~~	~~F~~	~~T~~	T		F			F	F	~~T~~	T	~~T~~			F
8	~~T~~	~~F~~	~~F~~	~~F~~	T		F			F	F	~~T~~	F	~~F~~			F
9	~~F~~	~~T~~	~~T~~	~~T~~	F		T			T	F	~~F~~	F	~~T~~			T
10	~~F~~	~~T~~	~~T~~	~~F~~	F		T			T	F	~~F~~	F	~~F~~			T
11	~~F~~	~~T~~	~~F~~	~~T~~	F		F			T	F	~~F~~	F	~~T~~			T
12	~~F~~	~~T~~	~~F~~	~~F~~	F		F			T	F	~~F~~	F	~~F~~			T
13	~~F~~	~~F~~	~~T~~	~~T~~	F		T			F	F	~~F~~	F	~~T~~			F
14	~~F~~	~~F~~	~~T~~	~~F~~	F		T			F	F	~~F~~	F	~~F~~			F
15	~~F~~	~~F~~	~~F~~	~~T~~	F		F			F	F	~~F~~	F	~~T~~			F
16	~~F~~	~~F~~	~~F~~	~~F~~	F		F			F	F	~~F~~	F	~~F~~			F

Now, let’s cross off all the T’s and F’s we’ve used.

	A	B	F	Q	((A	||	F)	->	(!	(B	&&	(A	&&	Q))	<->	!	B))
1	~~T~~	~~T~~	~~T~~	~~T~~	T		T			~~T~~	T	~~T~~	~~T~~	~~T~~			T
2	~~T~~	~~T~~	~~T~~	~~F~~	T		T			~~T~~	F	~~T~~	~~F~~	~~F~~			T
3	~~T~~	~~T~~	~~F~~	~~T~~	T		F			~~T~~	T	~~T~~	~~T~~	~~T~~			T
4	~~T~~	~~T~~	~~F~~	~~F~~	T		F			~~T~~	F	~~T~~	~~F~~	~~F~~			T
5	~~T~~	~~F~~	~~T~~	~~T~~	T		T			~~F~~	F	~~T~~	~~T~~	~~T~~			F
6	~~T~~	~~F~~	~~T~~	~~F~~	T		T			~~F~~	F	~~T~~	~~F~~	~~F~~			F
7	~~T~~	~~F~~	~~F~~	~~T~~	T		F			~~F~~	F	~~T~~	~~T~~	~~T~~			F
8	~~T~~	~~F~~	~~F~~	~~F~~	T		F			~~F~~	F	~~T~~	~~F~~	~~F~~			F
9	~~F~~	~~T~~	~~T~~	~~T~~	F		T			~~T~~	F	~~F~~	~~F~~	~~T~~			T
10	~~F~~	~~T~~	~~T~~	~~F~~	F		T			~~T~~	F	~~F~~	~~F~~	~~F~~			T
11	~~F~~	~~T~~	~~F~~	~~T~~	F		F			~~T~~	F	~~F~~	~~F~~	~~T~~			T
12	~~F~~	~~T~~	~~F~~	~~F~~	F		F			~~T~~	F	~~F~~	~~F~~	~~F~~			T
13	~~F~~	~~F~~	~~T~~	~~T~~	F		T			~~F~~	F	~~F~~	~~F~~	~~T~~			F
14	~~F~~	~~F~~	~~T~~	~~F~~	F		T			~~F~~	F	~~F~~	~~F~~	~~F~~			F
15	~~F~~	~~F~~	~~F~~	~~T~~	F		F			~~F~~	F	~~F~~	~~F~~	~~T~~			F
16	~~F~~	~~F~~	~~F~~	~~F~~	F		F			~~F~~	F	~~F~~	~~F~~	~~F~~			F

And, as we always do after finishing what's inside a pair of parentheses, we deal with any exclamation marks attached to the parentheses. In the last case, there wasn't one. But in this case, there is. So, in the column for the exclamation mark beside the "(B," we write the opposite of whatever is in the "&&" column to the right of "(B." After all, that exclamation mark is negating whatever is inside the parentheses.

	A	B	F	Q	((A	||	F)	->	(!	(B	&&	(A	&&	Q))	<->	!	B))
1	~~T~~	~~T~~	~~T~~	~~T~~	T		T		F	~~T~~	T	~~T~~	~~T~~	~~T~~			T
2	~~T~~	~~T~~	~~T~~	~~F~~	T		T		T	~~T~~	F	~~T~~	~~F~~	~~F~~			T
3	~~T~~	~~T~~	~~F~~	~~T~~	T		F		F	~~T~~	T	~~T~~	~~T~~	~~T~~			T
4	~~T~~	~~T~~	~~F~~	~~F~~	T		F		T	~~T~~	F	~~T~~	~~F~~	~~F~~			T
5	~~T~~	~~F~~	~~T~~	~~T~~	T		T		T	~~F~~	F	~~T~~	~~T~~	~~T~~			F
6	~~T~~	~~F~~	~~T~~	~~F~~	T		T		T	~~F~~	F	~~T~~	~~F~~	~~F~~			F
7	~~T~~	~~F~~	~~F~~	~~T~~	T		F		T	~~F~~	F	~~T~~	~~T~~	~~T~~			F
8	~~T~~	~~F~~	~~F~~	~~F~~	T		F		T	~~F~~	F	~~T~~	~~F~~	~~F~~			F
9	~~F~~	~~T~~	~~T~~	~~T~~	F		T		T	~~T~~	F	~~F~~	~~F~~	~~T~~			T
10	~~F~~	~~T~~	~~T~~	~~F~~	F		T		T	~~T~~	F	~~F~~	~~F~~	~~F~~			T
11	~~F~~	~~T~~	~~F~~	~~T~~	F		F		T	~~T~~	F	~~F~~	~~F~~	~~T~~			T
12	~~F~~	~~T~~	~~F~~	~~F~~	F		F		T	~~T~~	F	~~F~~	~~F~~	~~F~~			T
13	~~F~~	~~F~~	~~T~~	~~T~~	F		T		T	~~F~~	F	~~F~~	~~F~~	~~T~~			F
14	~~F~~	~~F~~	~~T~~	~~F~~	F		T		T	~~F~~	F	~~F~~	~~F~~	~~F~~			F
15	~~F~~	~~F~~	~~F~~	~~T~~	F		F		T	~~F~~	F	~~F~~	~~F~~	~~T~~			F
16	~~F~~	~~F~~	~~F~~	~~F~~	F		F		T	~~F~~	F	~~F~~	~~F~~	~~F~~			F

Now, let's cross off all the T's and F's we just used.

	A	B	F	Q	((A	||	F)	->	(!	(B	&&	(A	&&	Q))	<->	!	B))
1	~~T~~	~~T~~	~~T~~	~~T~~	T		T		F	~~T~~	~~T~~	~~T~~	~~T~~	~~T~~			T
2	~~T~~	~~T~~	~~T~~	~~F~~	T		T		T	~~T~~	~~F~~	~~T~~	~~F~~	~~F~~			T
3	~~T~~	~~T~~	~~F~~	~~T~~	T		F		F	~~T~~	~~T~~	~~T~~	~~T~~	~~T~~			T
4	~~T~~	~~T~~	~~F~~	~~F~~	T		F		T	~~T~~	~~F~~	~~T~~	~~F~~	~~F~~			T
5	~~T~~	~~F~~	~~T~~	~~T~~	T		T		T	~~F~~	~~F~~	~~T~~	~~T~~	~~T~~			F
6	~~T~~	~~F~~	~~T~~	~~F~~	T		T		T	~~F~~	~~F~~	~~T~~	~~F~~	~~F~~			F
7	~~T~~	~~F~~	~~F~~	~~T~~	T		F		T	~~F~~	~~F~~	~~T~~	~~T~~	~~T~~			F
8	~~T~~	~~F~~	~~F~~	~~F~~	T		F		T	~~F~~	~~F~~	~~T~~	~~F~~	~~F~~			F
9	~~F~~	~~T~~	~~T~~	~~T~~	F		T		T	~~T~~	~~F~~	~~F~~	~~F~~	~~T~~			T
10	~~F~~	~~T~~	~~T~~	~~F~~	F		T		T	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~			T
11	~~F~~	~~T~~	~~F~~	~~T~~	F		F		T	~~T~~	~~F~~	~~F~~	~~F~~	~~T~~			T
12	~~F~~	~~T~~	~~F~~	~~F~~	F		F		T	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~			T
13	~~F~~	~~F~~	~~T~~	~~T~~	F		T		T	~~F~~	~~F~~	~~F~~	~~F~~	~~T~~			F
14	~~F~~	~~F~~	~~T~~	~~F~~	F		T		T	~~F~~	~~F~~	~~F~~	~~F~~	~~F~~			F
15	~~F~~	~~F~~	~~F~~	~~T~~	F		F		T	~~F~~	~~F~~	~~F~~	~~F~~	~~T~~			F
16	~~F~~	~~F~~	~~F~~	~~F~~	F		F		T	~~F~~	~~F~~	~~F~~	~~F~~	~~F~~			F

Next, we have to deal with whatever parentheses are innermost among those that are left.

((A || F) -> (! (B && (A && Q)) <-> ! B))

We only have two sets of parentheses left (inside the outermost pair), and both are "buried equally deep." So, we can do either one. However, since we've been working inside the left pair of parentheses this whole time, let's do that one first.

As always, we begin by asking whether there are any exclamation marks inside the parentheses. In this case, there is. There are two, in fact. However, we've already dealt with the "!" attached to the "(B," so we only have to take care of the "!" attached to the "B." So, in the column under that exclamation mark, we write the opposite of whatever is under the "B," since the "!" negates the truth value of the "B."

	A	B	F	Q	((A	||	F)	->	(!	(B	&&	(A	&&	Q))	<->	!	B))
1	~~T~~	~~T~~	~~T~~	~~T~~	T		T		F	~~T~~	~~T~~	~~T~~	~~T~~	~~T~~		F	T
2	~~T~~	~~T~~	~~T~~	~~F~~	T		T		T	~~T~~	~~F~~	~~T~~	~~F~~	~~F~~		F	T
3	~~T~~	~~T~~	~~F~~	~~T~~	T		F		F	~~T~~	~~T~~	~~T~~	~~T~~	~~T~~		F	T
4	~~T~~	~~T~~	~~F~~	~~F~~	T		F		T	~~T~~	~~F~~	~~T~~	~~F~~	~~F~~		F	T
5	~~T~~	~~F~~	~~T~~	~~T~~	T		T		T	~~F~~	~~F~~	~~T~~	~~T~~	~~T~~		T	F
6	~~T~~	~~F~~	~~T~~	~~F~~	T		T		T	~~F~~	~~F~~	~~T~~	~~F~~	~~F~~		T	F
7	~~T~~	~~F~~	~~F~~	~~T~~	T		F		T	~~F~~	~~F~~	~~T~~	~~T~~	~~T~~		T	F
8	~~T~~	~~F~~	~~F~~	~~F~~	T		F		T	~~F~~	~~F~~	~~T~~	~~F~~	~~F~~		T	F
9	~~F~~	~~T~~	~~T~~	~~T~~	F		T		T	~~T~~	~~F~~	~~F~~	~~F~~	~~T~~		F	T
10	~~F~~	~~T~~	~~T~~	~~F~~	F		T		T	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~		F	T
11	~~F~~	~~T~~	~~F~~	~~T~~	F		F		T	~~T~~	~~F~~	~~F~~	~~F~~	~~T~~		F	T
12	~~F~~	~~T~~	~~F~~	~~F~~	F		F		T	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~		F	T
13	~~F~~	~~F~~	~~T~~	~~T~~	F		T		T	~~F~~	~~F~~	~~F~~	~~F~~	~~T~~		T	F
14	~~F~~	~~F~~	~~T~~	~~F~~	F		T		T	~~F~~	~~F~~	~~F~~	~~F~~	~~F~~		T	F
15	~~F~~	~~F~~	~~F~~	~~T~~	F		F		T	~~F~~	~~F~~	~~F~~	~~F~~	~~T~~		T	F
16	~~F~~	~~F~~	~~F~~	~~F~~	F		F		T	~~F~~	~~F~~	~~F~~	~~F~~	~~F~~		T	F

Then, we cross off all the T's and F's we've just used.

	A	B	F	Q	((A	||	F)	->	(!	(B	&&	(A	&&	Q))	<->	!	B))
1	~~T~~	~~T~~	~~T~~	~~T~~	T		T		F	~~T~~	~~T~~	~~T~~	~~T~~	~~T~~		F	~~T~~
2	~~T~~	~~T~~	~~T~~	~~F~~	T		T		T	~~T~~	~~F~~	~~T~~	~~F~~	~~F~~		F	~~T~~
3	~~T~~	~~T~~	~~F~~	~~T~~	T		F		F	~~T~~	~~T~~	~~T~~	~~T~~	~~T~~		F	~~T~~
4	~~T~~	~~T~~	~~F~~	~~F~~	T		F		T	~~T~~	~~F~~	~~T~~	~~F~~	~~F~~		F	~~T~~
5	~~T~~	~~F~~	~~T~~	~~T~~	T		T		T	~~F~~	~~F~~	~~T~~	~~T~~	~~T~~		T	~~F~~
6	~~T~~	~~F~~	~~T~~	~~F~~	T		T		T	~~F~~	~~F~~	~~T~~	~~F~~	~~F~~		T	~~F~~
7	~~T~~	~~F~~	~~F~~	~~T~~	T		F		T	~~F~~	~~F~~	~~T~~	~~T~~	~~T~~		T	~~F~~
8	~~T~~	~~F~~	~~F~~	~~F~~	T		F		T	~~F~~	~~F~~	~~T~~	~~F~~	~~F~~		T	~~F~~
9	~~F~~	~~T~~	~~T~~	~~T~~	F		T		T	~~T~~	~~F~~	~~F~~	~~F~~	~~T~~		F	~~T~~
10	~~F~~	~~T~~	~~T~~	~~F~~	F		T		T	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~		F	~~T~~
11	~~F~~	~~T~~	~~F~~	~~T~~	F		F		T	~~T~~	~~F~~	~~F~~	~~F~~	~~T~~		F	~~T~~
12	~~F~~	~~T~~	~~F~~	~~F~~	F		F		T	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~		F	~~T~~
13	~~F~~	~~F~~	~~T~~	~~T~~	F		T		T	~~F~~	~~F~~	~~F~~	~~F~~	~~T~~		T	~~F~~
14	~~F~~	~~F~~	~~T~~	~~F~~	F		T		T	~~F~~	~~F~~	~~F~~	~~F~~	~~F~~		T	~~F~~
15	~~F~~	~~F~~	~~F~~	~~T~~	F		F		T	~~F~~	~~F~~	~~F~~	~~F~~	~~T~~		T	~~F~~
16	~~F~~	~~F~~	~~F~~	~~F~~	F		F		T	~~F~~	~~F~~	~~F~~	~~F~~	~~F~~		T	~~F~~

Now, we fill in the column under the double-headed arrow. Biconditionals are true if, and only if, their two component propositions have the same truth value. So, we have to compare the columns under the two exclamation marks. If they match, we write a "T" under the double-headed arrow. If they don't match, we write an "F."

	A	B	F	Q	((A	||	F)	->	(!	(B	&&	(A	&&	Q))	<->	!	B))
1	~~T~~	~~T~~	~~T~~	~~T~~	T		T		F	~~T~~	~~T~~	~~T~~	~~T~~	~~T~~	T	F	~~T~~
2	~~T~~	~~T~~	~~T~~	~~F~~	T		T		T	~~T~~	~~F~~	~~T~~	~~F~~	~~F~~	F	F	~~T~~
3	~~T~~	~~T~~	~~F~~	~~T~~	T		F		F	~~T~~	~~T~~	~~T~~	~~T~~	~~T~~	T	F	~~T~~
4	~~T~~	~~T~~	~~F~~	~~F~~	T		F		T	~~T~~	~~F~~	~~T~~	~~F~~	~~F~~	F	F	~~T~~
5	~~T~~	~~F~~	~~T~~	~~T~~	T		T		T	~~F~~	~~F~~	~~T~~	~~T~~	~~T~~	T	T	~~F~~
6	~~T~~	~~F~~	~~T~~	~~F~~	T		T		T	~~F~~	~~F~~	~~T~~	~~F~~	~~F~~	T	T	~~F~~
7	~~T~~	~~F~~	~~F~~	~~T~~	T		F		T	~~F~~	~~F~~	~~T~~	~~T~~	~~T~~	T	T	~~F~~
8	~~T~~	~~F~~	~~F~~	~~F~~	T		F		T	~~F~~	~~F~~	~~T~~	~~F~~	~~F~~	T	T	~~F~~
9	~~F~~	~~T~~	~~T~~	~~T~~	F		T		T	~~T~~	~~F~~	~~F~~	~~F~~	~~T~~	F	F	~~T~~
10	~~F~~	~~T~~	~~T~~	~~F~~	F		T		T	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	F	F	~~T~~
11	~~F~~	~~T~~	~~F~~	~~T~~	F		F		T	~~T~~	~~F~~	~~F~~	~~F~~	~~T~~	F	F	~~T~~
12	~~F~~	~~T~~	~~F~~	~~F~~	F		F		T	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	F	F	~~T~~
13	~~F~~	~~F~~	~~T~~	~~T~~	F		T		T	~~F~~	~~F~~	~~F~~	~~F~~	~~T~~	T	T	~~F~~
14	~~F~~	~~F~~	~~T~~	~~F~~	F		T		T	~~F~~	~~F~~	~~F~~	~~F~~	~~F~~	T	T	~~F~~
15	~~F~~	~~F~~	~~F~~	~~T~~	F		F		T	~~F~~	~~F~~	~~F~~	~~F~~	~~T~~	T	T	~~F~~
16	~~F~~	~~F~~	~~F~~	~~F~~	F		F		T	~~F~~	~~F~~	~~F~~	~~F~~	~~F~~	T	T	~~F~~

Then, we cross off all the T's and F's we've just used.

	A	B	F	Q	((A	||	F)	->	(!	(B	&&	(A	&&	Q))	<->	!	B))
1	~~T~~	~~T~~	~~T~~	~~T~~	T		T		~~F~~	~~T~~	~~T~~	~~T~~	~~T~~	~~T~~	T	~~F~~	~~T~~
2	~~T~~	~~T~~	~~T~~	~~F~~	T		T		~~T~~	~~T~~	~~F~~	~~T~~	~~F~~	~~F~~	F	~~F~~	~~T~~
3	~~T~~	~~T~~	~~F~~	~~T~~	T		F		~~F~~	~~T~~	~~T~~	~~T~~	~~T~~	~~T~~	T	~~F~~	~~T~~
4	~~T~~	~~T~~	~~F~~	~~F~~	T		F		~~T~~	~~T~~	~~F~~	~~T~~	~~F~~	~~F~~	F	~~F~~	~~T~~
5	~~T~~	~~F~~	~~T~~	~~T~~	T		T		~~T~~	~~F~~	~~F~~	~~T~~	~~T~~	~~T~~	T	~~T~~	~~F~~
6	~~T~~	~~F~~	~~T~~	~~F~~	T		T		~~T~~	~~F~~	~~F~~	~~T~~	~~F~~	~~F~~	T	~~T~~	~~F~~
7	~~T~~	~~F~~	~~F~~	~~T~~	T		F		~~T~~	~~F~~	~~F~~	~~T~~	~~T~~	~~T~~	T	~~T~~	~~F~~
8	~~T~~	~~F~~	~~F~~	~~F~~	T		F		~~T~~	~~F~~	~~F~~	~~T~~	~~F~~	~~F~~	T	~~T~~	~~F~~
9	~~F~~	~~T~~	~~T~~	~~T~~	F		T		~~T~~	~~T~~	~~F~~	~~F~~	~~F~~	~~T~~	F	~~F~~	~~T~~
10	~~F~~	~~T~~	~~T~~	~~F~~	F		T		~~T~~	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	F	~~F~~	~~T~~
11	~~F~~	~~T~~	~~F~~	~~T~~	F		F		~~T~~	~~T~~	~~F~~	~~F~~	~~F~~	~~T~~	F	~~F~~	~~T~~
12	~~F~~	~~T~~	~~F~~	~~F~~	F		F		~~T~~	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	F	~~F~~	~~T~~
13	~~F~~	~~F~~	~~T~~	~~T~~	F		T		~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	~~T~~	T	~~T~~	~~F~~
14	~~F~~	~~F~~	~~T~~	~~F~~	F		T		~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	~~F~~	T	~~T~~	~~F~~
15	~~F~~	~~F~~	~~F~~	~~T~~	F		F		~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	~~T~~	T	~~T~~	~~F~~
16	~~F~~	~~F~~	~~F~~	~~F~~	F		F		~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	~~F~~	T	~~T~~	~~F~~

After taking care of any negations inside a pair of parentheses, then figuring out the truth values for the main connective in those parentheses, we always take care of any exclamation marks attached to the outside of the parentheses. But there are none, so we can move on to whatever pair of parentheses is now the "innermost."

At present, the only pair of parentheses that is inside the outermost pair, and which we haven't filled in all the way, is the pair in "(A || F)." So, we check for any exclamation marks inside it. There are none. So we fill in the column under its main connective.

	A	B	F	Q	((A	||	F)	->	(!	(B	&&	(A	&&	Q))	<->	!	B))
1	~~T~~	~~T~~	~~T~~	~~T~~	T	T	T		~~F~~	~~T~~	~~T~~	~~T~~	~~T~~	~~T~~	T	~~F~~	~~T~~
2	~~T~~	~~T~~	~~T~~	~~F~~	T	T	T		~~T~~	~~T~~	~~F~~	~~T~~	~~F~~	~~F~~	F	~~F~~	~~T~~
3	~~T~~	~~T~~	~~F~~	~~T~~	T	T	F		~~F~~	~~T~~	~~T~~	~~T~~	~~T~~	~~T~~	T	~~F~~	~~T~~
4	~~T~~	~~T~~	~~F~~	~~F~~	T	T	F		~~T~~	~~T~~	~~F~~	~~T~~	~~F~~	~~F~~	F	~~F~~	~~T~~
5	~~T~~	~~F~~	~~T~~	~~T~~	T	T	T		~~T~~	~~F~~	~~F~~	~~T~~	~~T~~	~~T~~	T	~~T~~	~~F~~
6	~~T~~	~~F~~	~~T~~	~~F~~	T	T	T		~~T~~	~~F~~	~~F~~	~~T~~	~~F~~	~~F~~	T	~~T~~	~~F~~
7	~~T~~	~~F~~	~~F~~	~~T~~	T	T	F		~~T~~	~~F~~	~~F~~	~~T~~	~~T~~	~~T~~	T	~~T~~	~~F~~
8	~~T~~	~~F~~	~~F~~	~~F~~	T	T	F		~~T~~	~~F~~	~~F~~	~~T~~	~~F~~	~~F~~	T	~~T~~	~~F~~
9	~~F~~	~~T~~	~~T~~	~~T~~	F	T	T		~~T~~	~~T~~	~~F~~	~~F~~	~~F~~	~~T~~	F	~~F~~	~~T~~
10	~~F~~	~~T~~	~~T~~	~~F~~	F	T	T		~~T~~	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	F	~~F~~	~~T~~
11	~~F~~	~~T~~	~~F~~	~~T~~	F	F	F		~~T~~	~~T~~	~~F~~	~~F~~	~~F~~	~~T~~	F	~~F~~	~~T~~
12	~~F~~	~~T~~	~~F~~	~~F~~	F	F	F		~~T~~	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	F	~~F~~	~~T~~
13	~~F~~	~~F~~	~~T~~	~~T~~	F	T	T		~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	~~T~~	T	~~T~~	~~F~~
14	~~F~~	~~F~~	~~T~~	~~F~~	F	T	T		~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	~~F~~	T	~~T~~	~~F~~
15	~~F~~	~~F~~	~~F~~	~~T~~	F	F	F		~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	~~T~~	T	~~T~~	~~F~~
16	~~F~~	~~F~~	~~F~~	~~F~~	F	F	F		~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	~~F~~	T	~~T~~	~~F~~

And we cross off all the T's and F's we've just used.

	A	B	F	Q	((A	||	F)	->	(!	(B	&&	(A	&&	Q))	<->	!	B))
1	~~T~~	~~T~~	~~T~~	~~T~~	~~T~~	T	~~T~~		~~F~~	~~T~~	~~T~~	~~T~~	~~T~~	~~T~~	T	~~F~~	~~T~~
2	~~T~~	~~T~~	~~T~~	~~F~~	~~T~~	T	~~T~~		~~T~~	~~T~~	~~F~~	~~T~~	~~F~~	~~F~~	F	~~F~~	~~T~~
3	~~T~~	~~T~~	~~F~~	~~T~~	~~T~~	T	~~F~~		~~F~~	~~T~~	~~T~~	~~T~~	~~T~~	~~T~~	T	~~F~~	~~T~~
4	~~T~~	~~T~~	~~F~~	~~F~~	~~T~~	T	~~F~~		~~T~~	~~T~~	~~F~~	~~T~~	~~F~~	~~F~~	F	~~F~~	~~T~~
5	~~T~~	~~F~~	~~T~~	~~T~~	~~T~~	T	~~T~~		~~T~~	~~F~~	~~F~~	~~T~~	~~T~~	~~T~~	T	~~T~~	~~F~~
6	~~T~~	~~F~~	~~T~~	~~F~~	~~T~~	T	~~T~~		~~T~~	~~F~~	~~F~~	~~T~~	~~F~~	~~F~~	T	~~T~~	~~F~~
7	~~T~~	~~F~~	~~F~~	~~T~~	~~T~~	T	~~F~~		~~T~~	~~F~~	~~F~~	~~T~~	~~T~~	~~T~~	T	~~T~~	~~F~~
8	~~T~~	~~F~~	~~F~~	~~F~~	~~T~~	T	~~F~~		~~T~~	~~F~~	~~F~~	~~T~~	~~F~~	~~F~~	T	~~T~~	~~F~~
9	~~F~~	~~T~~	~~T~~	~~T~~	~~F~~	T	~~T~~		~~T~~	~~T~~	~~F~~	~~F~~	~~F~~	~~T~~	F	~~F~~	~~T~~
10	~~F~~	~~T~~	~~T~~	~~F~~	~~F~~	T	~~T~~		~~T~~	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	F	~~F~~	~~T~~
11	~~F~~	~~T~~	~~F~~	~~T~~	~~F~~	F	~~F~~		~~T~~	~~T~~	~~F~~	~~F~~	~~F~~	~~T~~	F	~~F~~	~~T~~
12	~~F~~	~~T~~	~~F~~	~~F~~	~~F~~	F	~~F~~		~~T~~	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	F	~~F~~	~~T~~
13	~~F~~	~~F~~	~~T~~	~~T~~	~~F~~	T	~~T~~		~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	~~T~~	T	~~T~~	~~F~~
14	~~F~~	~~F~~	~~T~~	~~F~~	~~F~~	T	~~T~~		~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	~~F~~	T	~~T~~	~~F~~
15	~~F~~	~~F~~	~~F~~	~~T~~	~~F~~	F	~~F~~		~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	~~T~~	T	~~T~~	~~F~~
16	~~F~~	~~F~~	~~F~~	~~F~~	~~F~~	F	~~F~~		~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	~~F~~	T	~~T~~	~~F~~

Then, we look for any exclamation marks attached to the outside of the parentheses we just filled in. There are none. So, we are free to fill in the column under the single-headed arrow, using the column under the double pipes and the column under the double-headed arrow. Since we are

filling in a column under a single-headed arrow, we are dealing with a conditional statement. Conditional statements (at least, the conditional statements as we deal with them in symbolic logic) are only false if their antecedent is true while their consequent is false. (This happens on lines 2, 4, 9, and 10.) The rest of the time, they are true.

	A	B	F	Q	((A	||	F)	->	(!	(B	&&	(A	&&	Q))	<->	!	B))
1	~~T~~	~~T~~	~~T~~	~~T~~	~~T~~	T	~~T~~	T	~~F~~	~~T~~	~~T~~	~~T~~	~~T~~	~~T~~	T	~~F~~	~~T~~
2	~~T~~	~~T~~	~~T~~	~~F~~	~~T~~	T	~~T~~	F	~~T~~	~~T~~	~~F~~	~~T~~	~~F~~	~~F~~	F	~~F~~	~~T~~
3	~~T~~	~~T~~	~~F~~	~~T~~	~~T~~	T	~~F~~	T	~~F~~	~~T~~	~~T~~	~~T~~	~~T~~	~~T~~	T	~~F~~	~~T~~
4	~~T~~	~~T~~	~~F~~	~~F~~	~~T~~	T	~~F~~	F	~~T~~	~~T~~	~~F~~	~~T~~	~~F~~	~~F~~	F	~~F~~	~~T~~
5	~~T~~	~~F~~	~~T~~	~~T~~	~~T~~	T	~~T~~	T	~~T~~	~~F~~	~~F~~	~~T~~	~~T~~	~~T~~	T	~~T~~	~~F~~
6	~~T~~	~~F~~	~~T~~	~~F~~	~~T~~	T	~~T~~	T	~~T~~	~~F~~	~~F~~	~~T~~	~~F~~	~~F~~	T	~~T~~	~~F~~
7	~~T~~	~~F~~	~~F~~	~~T~~	~~T~~	T	~~F~~	T	~~T~~	~~F~~	~~F~~	~~T~~	~~T~~	~~T~~	T	~~T~~	~~F~~
8	~~T~~	~~F~~	~~F~~	~~F~~	~~T~~	T	~~F~~	T	~~T~~	~~F~~	~~F~~	~~T~~	~~F~~	~~F~~	T	~~T~~	~~F~~
9	~~F~~	~~T~~	~~T~~	~~T~~	~~F~~	T	~~T~~	F	~~T~~	~~T~~	~~F~~	~~F~~	~~F~~	~~T~~	F	~~F~~	~~T~~
10	~~F~~	~~T~~	~~T~~	~~F~~	~~F~~	T	~~T~~	F	~~T~~	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	F	~~F~~	~~T~~
11	~~F~~	~~T~~	~~F~~	~~T~~	~~F~~	F	~~F~~	T	~~T~~	~~T~~	~~F~~	~~F~~	~~F~~	~~T~~	F	~~F~~	~~T~~
12	~~F~~	~~T~~	~~F~~	~~F~~	~~F~~	F	~~F~~	T	~~T~~	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	F	~~F~~	~~T~~
13	~~F~~	~~F~~	~~T~~	~~T~~	~~F~~	T	~~T~~	T	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	~~T~~	T	~~T~~	~~F~~
14	~~F~~	~~F~~	~~T~~	~~F~~	~~F~~	T	~~T~~	T	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	~~F~~	T	~~T~~	~~F~~
15	~~F~~	~~F~~	~~F~~	~~T~~	~~F~~	F	~~F~~	T	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	~~T~~	T	~~T~~	~~F~~
16	~~F~~	~~F~~	~~F~~	~~F~~	~~F~~	F	~~F~~	T	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	~~F~~	T	~~T~~	~~F~~

This last-filled-in column is the “answer” or “result” for the truth table. We can make it clear that it is the answer by leaving it circled and crossing everything else off.

	A	B	F	Q	((A	||	F)	->	(!	(B	&&	(A	&&	Q))	<->	!	B))
1	~~T~~	~~T~~	~~T~~	~~T~~	~~T~~	~~T~~	~~T~~	T	~~F~~	~~T~~	~~T~~	~~T~~	~~T~~	~~T~~	~~T~~	~~F~~	~~T~~
2	~~T~~	~~T~~	~~T~~	~~F~~	~~T~~	~~T~~	~~T~~	F	~~T~~	~~T~~	~~F~~	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	~~T~~
3	~~T~~	~~T~~	~~F~~	~~T~~	~~T~~	~~T~~	~~F~~	T	~~F~~	~~T~~	~~T~~	~~T~~	~~T~~	~~T~~	~~T~~	~~F~~	~~T~~
4	~~T~~	~~T~~	~~F~~	~~F~~	~~T~~	~~T~~	~~F~~	F	~~T~~	~~T~~	~~F~~	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	~~T~~
5	~~T~~	~~F~~	~~T~~	~~T~~	~~T~~	~~T~~	~~T~~	T	~~T~~	~~F~~	~~F~~	~~T~~	~~T~~	~~T~~	~~T~~	~~T~~	~~F~~
6	~~T~~	~~F~~	~~T~~	~~F~~	~~T~~	~~T~~	~~T~~	T	~~T~~	~~F~~	~~F~~	~~T~~	~~F~~	~~F~~	~~T~~	~~T~~	~~F~~
7	~~T~~	~~F~~	~~F~~	~~T~~	~~T~~	~~T~~	~~F~~	T	~~T~~	~~F~~	~~F~~	~~T~~	~~T~~	~~T~~	~~T~~	~~T~~	~~F~~
8	~~T~~	~~F~~	~~F~~	~~F~~	~~T~~	~~T~~	~~F~~	T	~~T~~	~~F~~	~~F~~	~~T~~	~~F~~	~~F~~	~~T~~	~~T~~	~~F~~
9	~~F~~	~~T~~	~~T~~	~~T~~	~~F~~	~~T~~	~~T~~	F	~~T~~	~~T~~	~~F~~	~~F~~	~~F~~	~~T~~	~~F~~	~~F~~	~~T~~
10	~~F~~	~~T~~	~~T~~	~~F~~	~~F~~	~~T~~	~~T~~	F	~~T~~	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	~~F~~	~~F~~	~~T~~
11	~~F~~	~~T~~	~~F~~	~~T~~	~~F~~	~~F~~	~~F~~	T	~~T~~	~~T~~	~~F~~	~~F~~	~~F~~	~~T~~	~~F~~	~~F~~	~~T~~
12	~~F~~	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	~~F~~	T	~~T~~	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	~~F~~	~~F~~	~~T~~
13	~~F~~	~~F~~	~~T~~	~~T~~	~~F~~	~~T~~	~~T~~	T	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	~~T~~	~~T~~	~~T~~	~~F~~
14	~~F~~	~~F~~	~~T~~	~~F~~	~~F~~	~~T~~	~~T~~	T	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	~~F~~	~~T~~	~~T~~	~~F~~
15	~~F~~	~~F~~	~~F~~	~~T~~	~~F~~	~~F~~	~~F~~	T	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	~~T~~	~~T~~	~~T~~	~~F~~
16	~~F~~	~~F~~	~~F~~	~~F~~	~~F~~	~~F~~	~~F~~	T	~~T~~	~~F~~	~~F~~	~~F~~	~~F~~	~~F~~	~~T~~	~~T~~	~~F~~

It tells us in what circumstances the proposition "((A || F) -> (!B && (A && Q)) <-> !B))" is true, and in what circumstances it is false. It turns out that it is only false when (1) "A," "B," and "F," are true, but "Q" is false (line 2), or (2) when "A," and "B" are both true, but "F," and "Q" are both false (line 4), or (3) when "A" is false, but "B," "F," and "Q" are true (line 9), or (4) when "A" and "Q" are both false, while "B" and "F," are both true (line 10). In the twelve other possible combinations of truth and falsity for "A," "B," "F," and "Q," "((A || F) -> (!B && (A && Q)) <-> !B))" will turn out to be true.

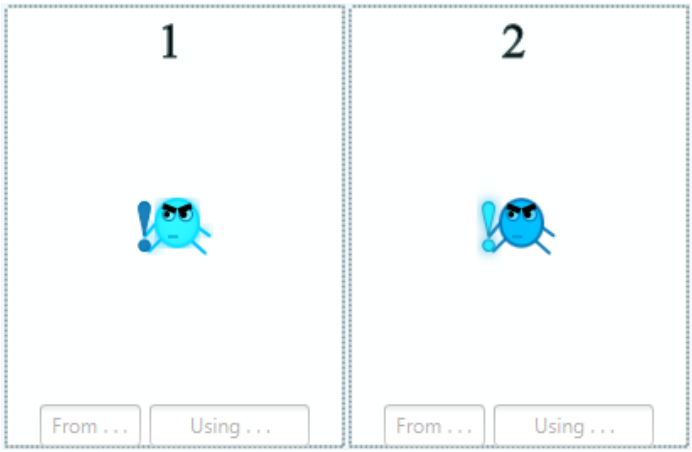
2.3.3: Activation Rules for Chambergon Battle Logic

In symbolic logic, every proposition has a truth value. It is either true or false. We can use "T" for "true" and "F" for "false." But we can also use "1" for true and "0" for false. This is what computers do. Well, they don't really have 1's and 0's in them. It's more like they have "electricity-on" and "electricity-off."

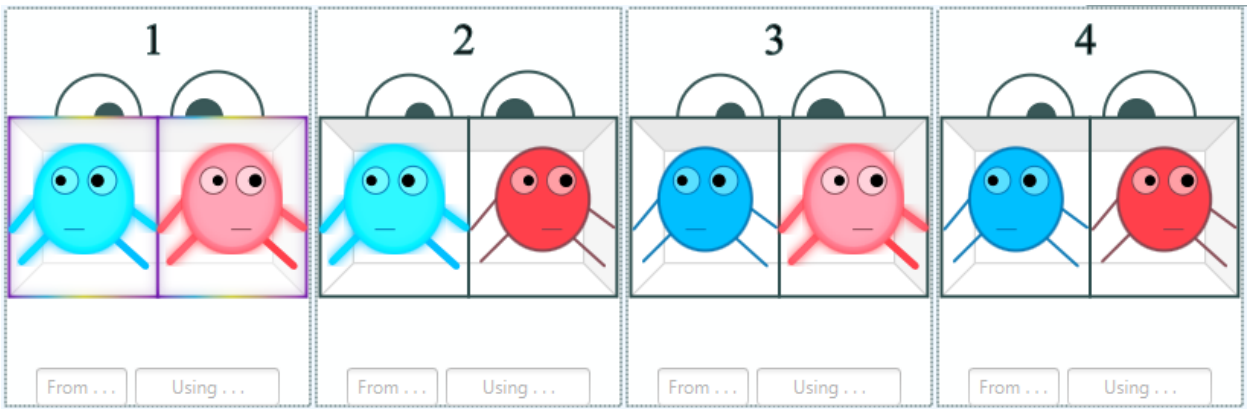
In Chambergon Battle Logic, we use the word "activated" for "electricity-on," and "deactivated" for "electricity-off." When a Character, Club, or Chambergon is activated, it lights up. When it is deactivated it . . . lights down? What's the opposite of "lighting up"? Anyway, a Chambergon will light up—it will become activated—if the contents of its two chambers are lit up in the right combination. Here are the rules.

- A Club always has the opposite activation state to whatever it is attached to.
- A Parcel is only activated when the contents of both chambers are activated.
- A Paragon is activated when the contents of either chamber are activated.
- A Pyramid is *de*activated when the contents of its attic are activated, but the contents of its basement are deactivated. Otherwise, it is activated.
- A Pulsar is activated whenever the contents of its attic and basement are in the same activation state. If they're both activated, then the Pulsar is activated. If they are both deactivated, the pulsar is activated. The Pulsar is only deactivated when one of its two contents is activated while the other isn't.

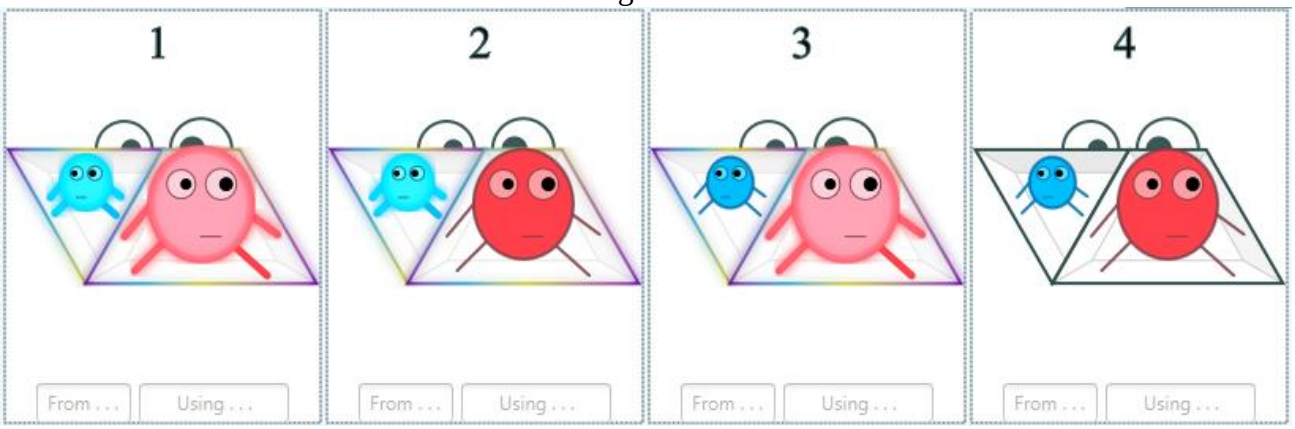
Clubs:



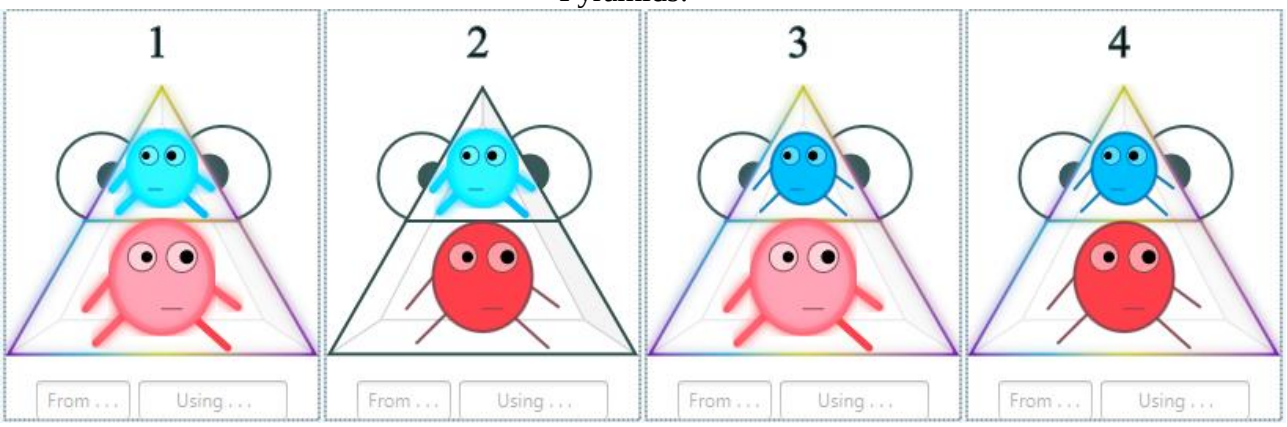
Parcels:



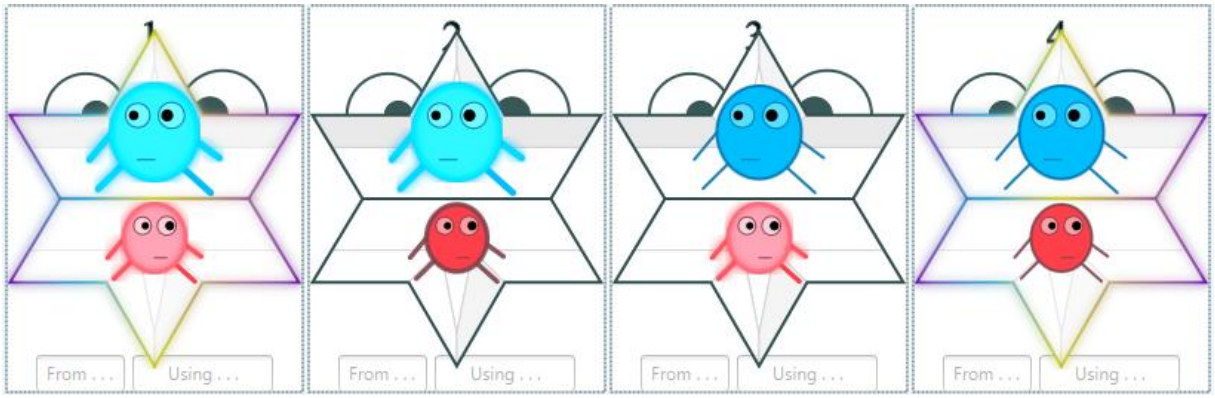
Paragons:



Pyramids:

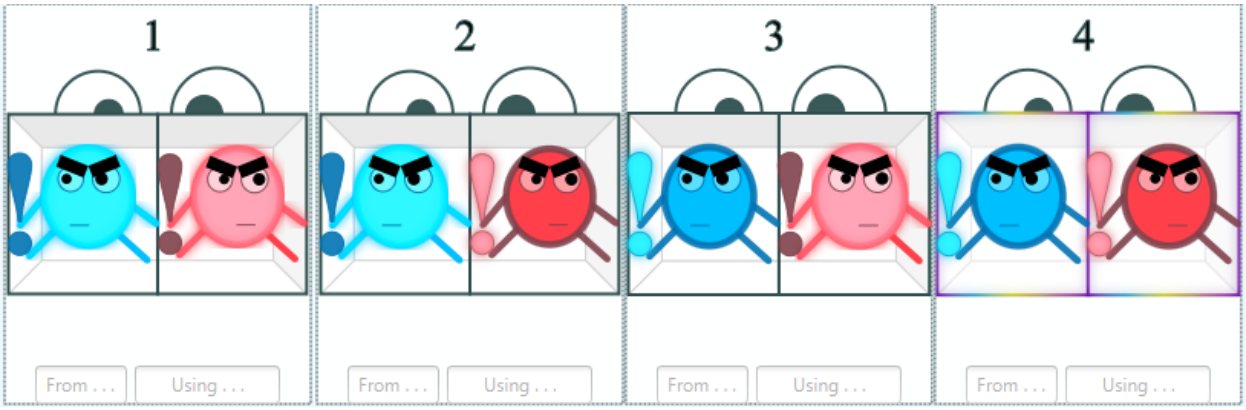


Pulsars:

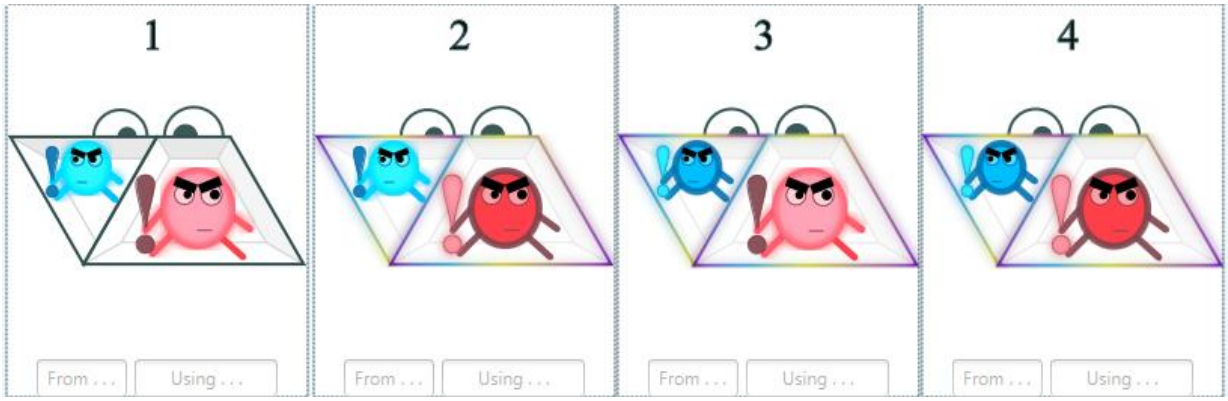


Now, you may ask what happens if the Character inside the Chambergon has a Club. The answer is that the Chambergon registers the state of the Club, not the state of the Character. If the Club is activated, then "the contents of that chamber are activated," as far the Chambergon is concerned. If the Club is deactivated, the contents of that chamber are deactivated, as far as the Chambergon is concerned.

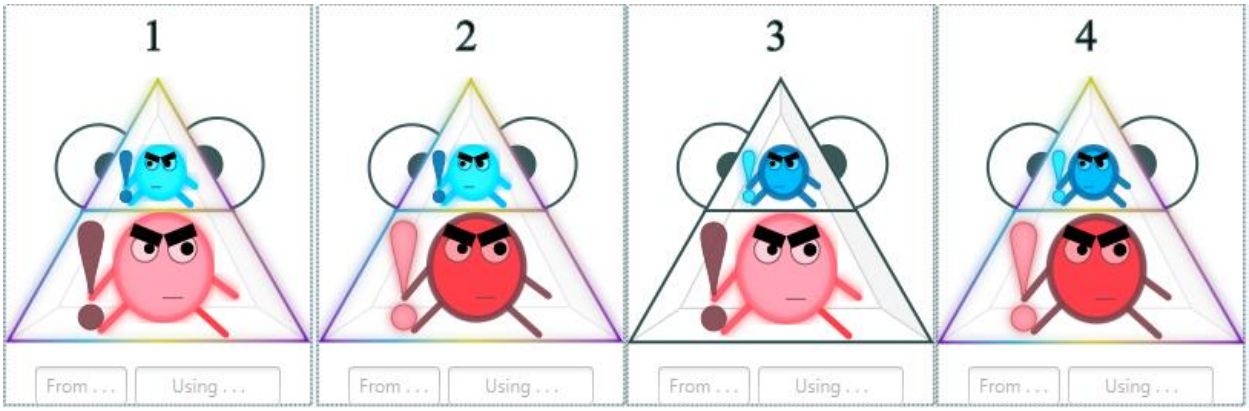
In the following, the Parcel lights up only in Card 4, because it is only in Card 4 that both its contents are activated. "What about Card 1?" you may ask, of course. But in Card 1, the two Clubs are deactivated, and it's the Clubs that count.



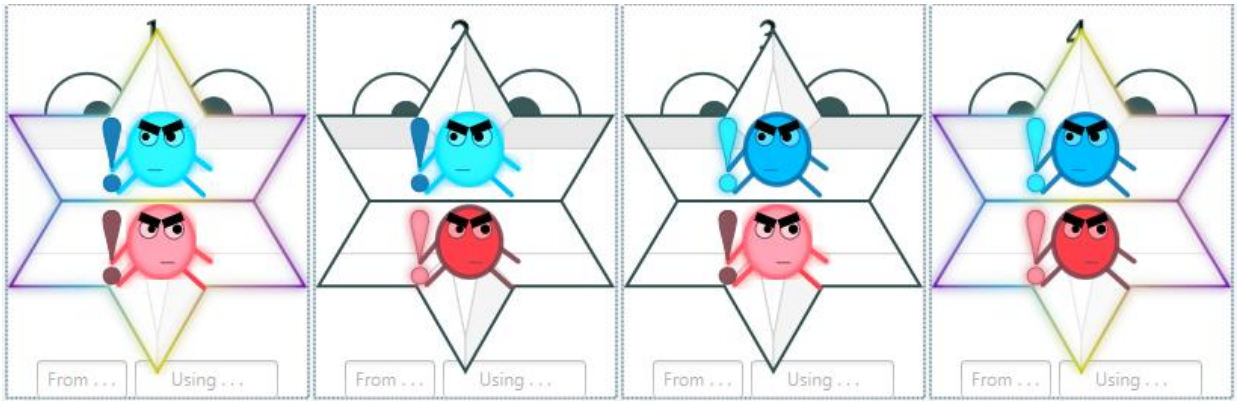
In Cards 2, 3, and 4 in the following series, the Paragon lights up because at least one of its two contents is activated. What matters, once again, is the state of each Club. So, if the Club in one chamber is activated, the whole Paragon lights up. If neither Club is activated, however (like in Card 1), the Paragon switches off, even though both the Characters are activated. Since the Clubs are deactivated, both contents count as being deactivated.



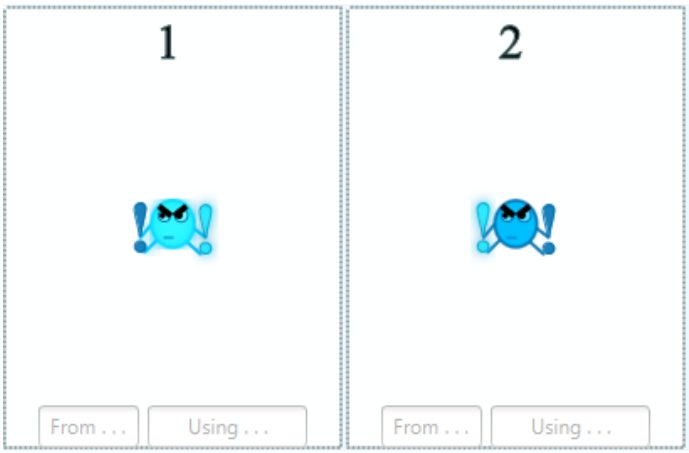
In Card 3, below, the Pyramid fails to light up because the contents of its attic are activated (that is, the Club in its attic is lit up) while the contents of its basement are not activated. Once again, it is the Clubs that matter. The fact that the Character in the basement of Card 3 is activated is irrelevant.



The Pulsar lights up in Cards 1 and 4 because in Card 1, the contents of both its chambers are deactivated (since the Clubs are deactivated, and that is all that matters to the Chambergon), and because in Card 4, the contents of both its chambers are activated (since the Clubs are activated, and that is all that matters to the Chambergon).

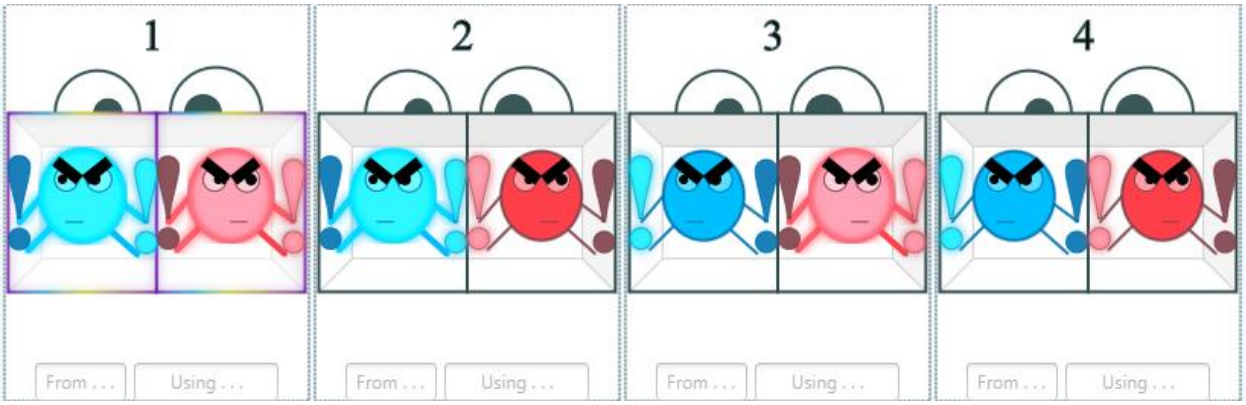


Things get even weirder when there are two Clubs involved.

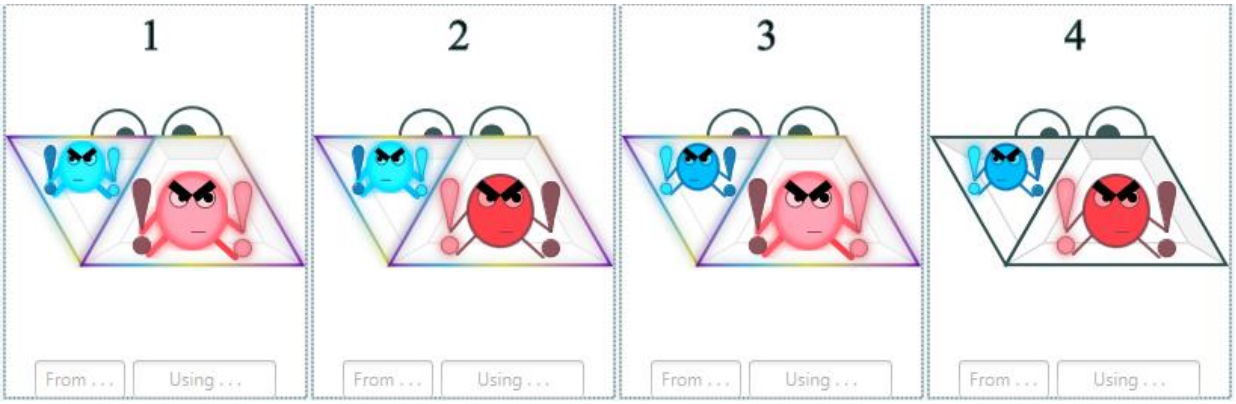


Notice that the second Club (the one on our right) always has the same activation state as the Character. (This is because a double-negated proposition always has the same truth value as the proposition by itself.)

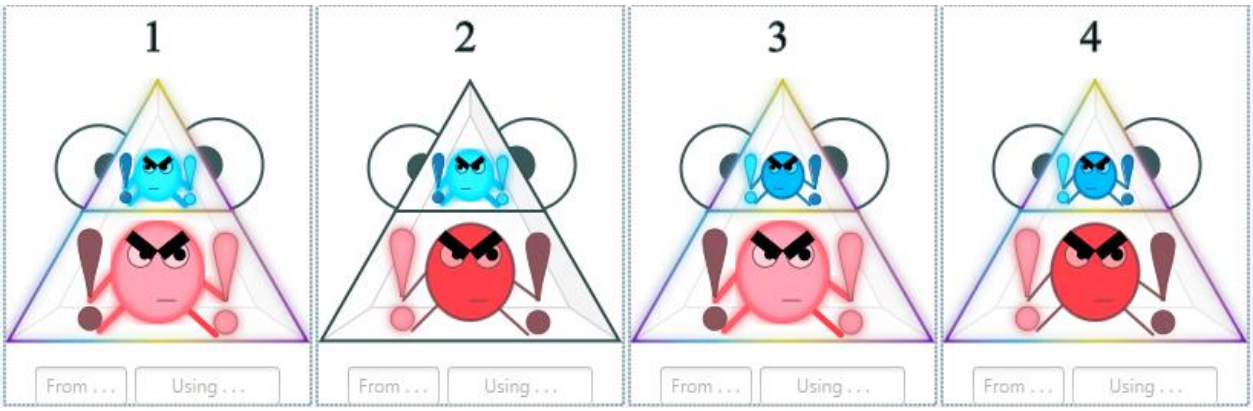
If we put double-negated Characters into a Chambergon, the Chambergon will register the activation state of the second Club, rather than the first Club or the Character.



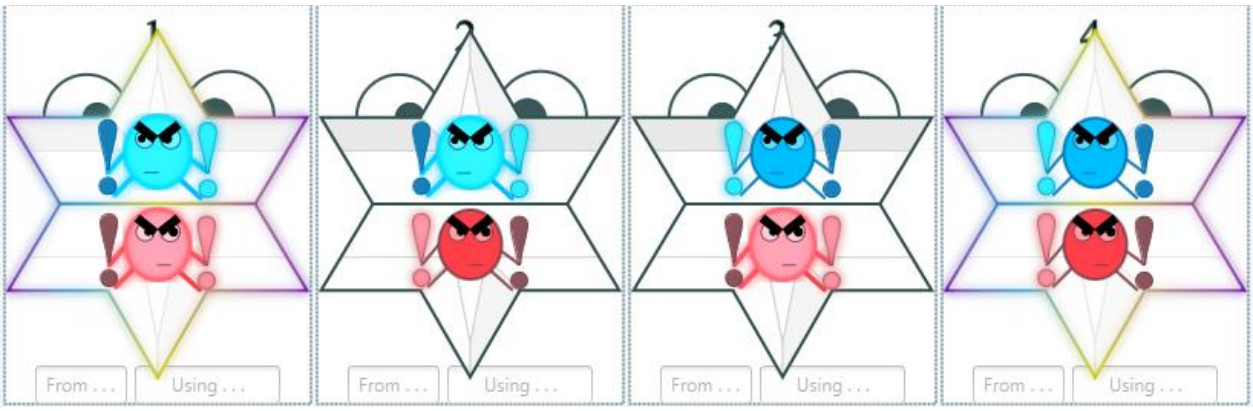
In the example above, Card 1's Parcel is the only one to light up, because it is the only one that considers both its contents to be activated. (It is the state of each Character's second Club that matters.) Cards 2 and 3 have activated Clubs in both their chambers, but if a Character has two Clubs, only the second Club counts.



The Paragon in Card 1 above is activated because the second Clubs in both its chambers are activated. The Paragon in Card 2 is activated because the second Club in its left chamber is activated. The Paragon in Card 3 is activated because the second Club in its right chamber is activated. The Paragon in Card 4, however, is not activated, because both the second Club in its left chamber and the second Club in its right Chamber are deactivated. It doesn't matter that the first Club in both chambers is activated. The state of a Character's first Club is overridden (negated) by the Character's second Club.



Look at the second Club (the Club on our right) in each chamber of the Pyramids above. Only in Card 2 is there a problem. In Card 2, the contents of the attic count as being activated, since that is the state of the second Club in the attic. The contents of the basement of the Pyramid in Card 2 count as deactivated, however, since that is the state of the second Club in the basement, and it overrides the activated state of the first Club.

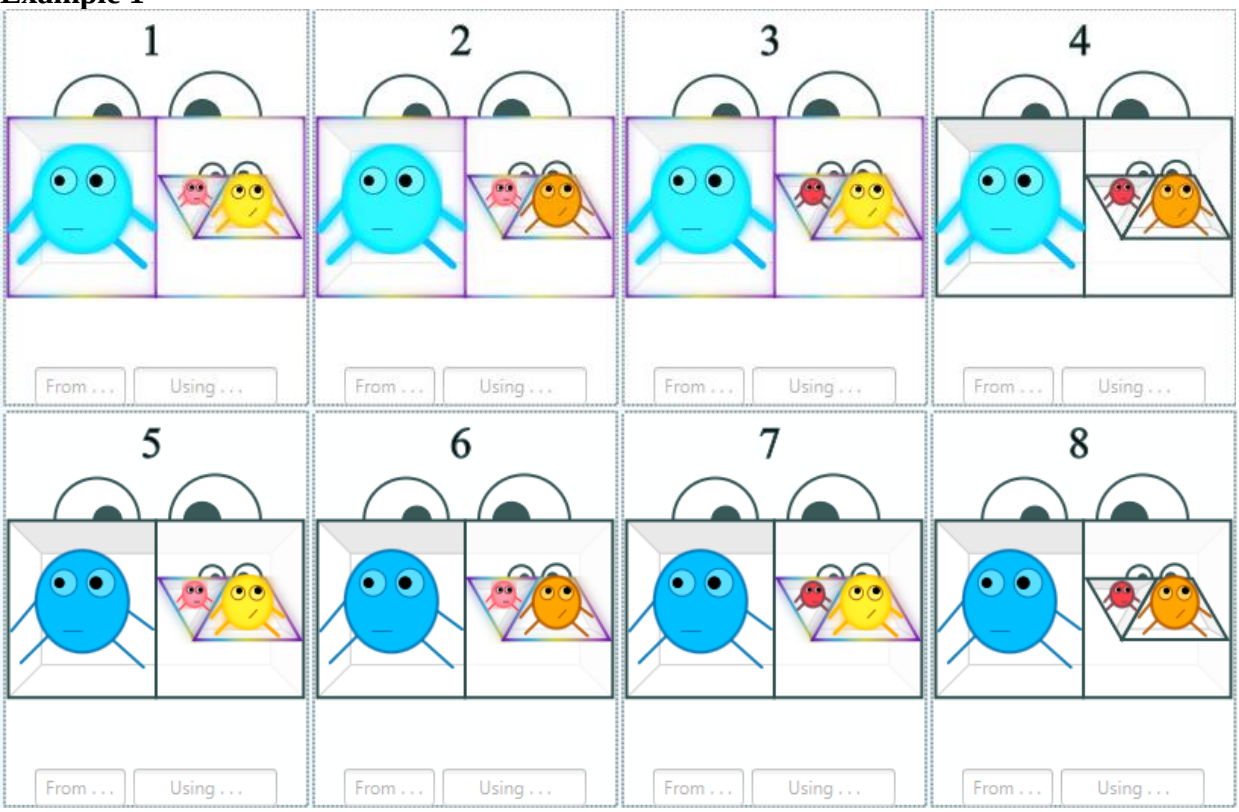


In Card 4, above, the Pulsar is activated even though the second Clubs in both its chambers are deactivated. This is because all a Pulsar cares about is whether the activation states of its two chambers match. In Card 4, therefore, it is the matching *deactivations* of its two second Clubs that makes the Pulsar light up, not the matching *activations* of the two first Clubs. The state of a first Club, as I keep saying, is overridden by the state of a second Club, if a Character or Chambergon is carrying two Clubs.

2.3.4: More Complex Examples

If having two Clubs in one chamber weren't complicated enough, you will recall that we can put Chambergons inside other Chambergons, and that Chambergons can have Clubs, just like Characters.

Example 1



The Parcels in Cards 1, 2, and 3 are activated, because the blue Characters in their left chambers are activated *and* the Paragons in their right chambers are activated. The Paragons in their right chambers are activated because either the red or orange Characters in them are activated (and sometimes both are). Unfortunately, both the red and orange Characters are deactivated in Card 4, which means the Paragon in that Card will not light up. But since it does not light up, the Parcel takes the contents of its right chamber to be deactivated, and a Parcel will only light up if the contents of both chambers are activated.

In Cards 4 – 7, the fact that the blue Character is deactivated keeps the Parcel as a whole from lighting up, even though the Paragon in the right chamber is activated in each. Then, in Card 8, nothing is activated, and thus the Parcel is not activated either.

Example 2

In this example, the blue Character is activated in Cards 1–4, which leads the Paragon in each Card to be activated, which leads the Club attached to the Paragon to be deactivated. Officially, then, the contents of Cards 1–4 are deactivated, since the activation state of the Paragon is overridden by the Club.

Just the opposite would be true in all four of Cards 5–8. However, the Parcel in Card 6 lights up, which lights up the Paragon, which deactivates the outermost Club.

(The reason the Parcel in Card 6 is activated is that the red Character is activated, but the orange Character's deactivation is overridden by its Club. Thus, the contents of the Parcel's two chambers are both technically activated.)

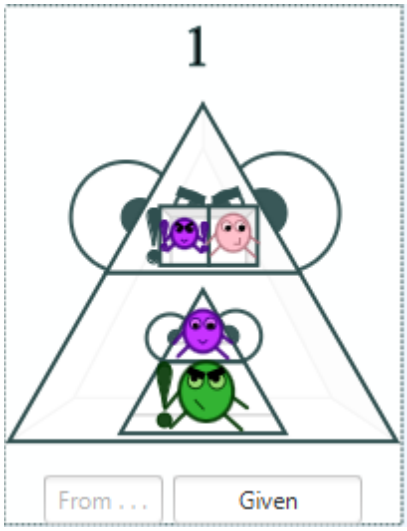
2.3.5: Activation Games Strategy

Step 1: Figure out how many cards are needed.

When you open a game, and find that the Goal Box is (a) empty, and (b) glowing red, you know you have an activation game. There will only be one Given card, and your first task will be to create the appropriate number of new Cards. The total number of Cards you will need will depend on how many different colors are in the one Given card.

Step 1a: Count the number of colors in the Given Card.

In the following Card, there are three colors: purple, pink, and green. It doesn't matter that there are two purple Characters. What matters is that they are both the same color, and so only add one additional color to the total count.



Step 1b: Do some simple math.

Count the number of colors, then multiply that number of 2's together. In this case, there are three colors, so we'd multiple three 2's together: 2 × 2 × 2. Since two times two is four, and two times four is eight, that means we need eight total cards.

If there had only been two colors in the first Card, we would multiply *two* 2's together. If there had been five colors in the first Card, we would multiply *five* 2's together. Notice: we would *not* multiply 5 × 2 (which equals 10). We would multiply 2 × 2 × 2 × 2 × 2 (which equals 32).

I cannot insist on that last point enough. If you skipped it, you absolutely must go back and read it.

To get the total number of cards, you do *not* multiply the number of colors in the first card times 2. Instead, you take the number of colors and multiply that many 2's times each other. So, if there are six colors, you have to multiply six twos together (*not* multiply six times two).

And if there had only been one color in the first Card, we would have multiplied only one 2 together. That is, we would have just written "2," and told ourselves, "We only need two Cards."

Step 1c: Give yourself the correct number of Cards.
After you figure out how many Cards you need, add more Cards (using the "New Card") button till you reach the number you need. (You already have one Card given, so you just need to add the rest.)

Step 2: Fill in all the Cards with whatever is in Card 1.
Just copy and paste whatever is in Card 1 into all the other Cards. You will end up with two, or four, or eight (or sixteen, or thirty-two, or sixty-four, etc.) identical Cards.

Step 3: Activate/Deactivate the Characters
You must now activate and deactivate the right Characters in the right pattern in every single Card. Here's how to do it.

Step 3a: Look at Card 1. Pick one of the colors in Card 1. Characters with this color should be activated in a "1 Card On, 1 Card Off" pattern. That is, every Character with that color in Card 1 should be activated. Then skip one card and go to Card 3 (if it exists), where you activate all the Characters with that color again. Then skip one card, and go to Card 5 (if it exists), where you activate all the Characters with that color again. Continue this "1 Card On, 1 Card Off" pattern till you run out of Cards.

Step 3b: Look at Card 1. Are there any colors in it besides the one you already did? If not, you're done; go to Step 4. If there are other colors, pick one. Characters with this color should be activated in a "2 Cards On, 2 Cards Off" pattern. That is, every Character with that color in Cards 1 and 2 should be activated. Then skip two Cards and go to Cards 5 and 6 (if they exist), where you activate all the Characters with that color again. Then, skip two Cards and go to Cards 9 and 10 (if they exist), where you activate all the Characters with that color again. Continue this "2 Cards On, 2 Cards Off" pattern till you run out of Cards.

Step 3c: Look at Card 1. Are there any colors in it besides the two you already did? If not, you're done; go to Step 4. If there are other colors, pick one. Characters with this color should be activated in a "4 Cards On, 4 Cards Off" pattern. That is, every Character with that color in Cards 1, 2, 3, and 4 should be activated. Then skip four Cards and go to Cards 9, 10, 11, and 12 (if they exist), where you activate all the Characters with that color again. Then skip four Cards and go to Cards 17, 18, 19, and 20, where you activate all the Characters with that color again. Continue this "4 Cards On, 4 Cards Off" pattern till you run out of Cards.

Step 3d: Look at Card 1. Are there any colors in it besides the three you already did? If not, you're done; go to Step 4. If there are other colors, pick one. Characters with this color should be activated in an "8 Cards On, 8 Cards Off" pattern. Then, if there are any colors left, pick one and use a "16 Cards On, 16 Cards Off" pattern for it. And if there are

any colors left, pick one and use a "32 Cards On, 32 Cards Off" pattern for it, etc. (For each new color, you double the number of Cards you do/skip in a row: 2, 4, 8, 16, 32, 64, etc.)

Step 4: Activate/deactivate the Characters' Clubs.
Go through the Cards one at a time. In each Card, look for any Characters holding a Club. If no Characters are holding a Club, skip this step and go to Step 5. If, however, a Character is holding a single Club, make sure that Club is activated when the Character is deactivated (and deactivated when the Character is activated). And if a Character is holding two Clubs, make sure the one on our left has the opposite activation state to the Character (i.e., it should be activated when the Character is deactivated, and deactivated when the Character is activated). Then, make sure that the one on our right has the same activation State as the Character (i.e., it should be activated when the Character is activated, and deactivated when the Character is deactivated).

Step 5: Activate/deactivate the Character's Chambergons.
In other words, go through each Card and find the Chambergons that contain only Characters (or Characters with Clubs). Ignore the Chambergons that also happen to contain another Chambergon.

For each Chambergon that contains only Characters (or Characters with Clubs) decide whether it should be activated using the following rules:

- A Parcel is only activated when the contents of both its chambers are activated.
- A Paragon is activated whenever the contents of at least one of its chambers are activated.
- A Pyramid is only deactivated when its attic's contents are activated, but not its basement's contents.
- A Pulsar is only activated when the activation/deactivation state of the contents of one chamber matches the activation/deactivation state of the contents of the other chamber.

When deciding whether the contents of a given chamber are activated or deactivated, remember that "Clubs are What Count" (and that the second Club—the Club on our right—overrides the first Club).

Step 6: Activate/deactivate those Chambergons' Clubs.
If the Chambergons you dealt with in Step 5 (the ones that only contain Characters or Characters with Clubs) have Clubs, make sure they are properly activated/deactivated. If the Chambergon is activated and has one Club, that Club should be deactivated, while if the Chambergon is deactivated the Club should be activated. The same goes for Chambergons with two Clubs, except that the Club on our right should have the same activation/deactivation state as the Chambergon.

Step 7: Activate/deactivate any Chambergons that contain a Character (or Character with Clubs) in one chamber, but another Chambergon (or Chambergon with Clubs) in the other chamber.

Step 8: Activate/deactivate those Chambergon's Clubs.

Step 9: Activate/deactivate any Chambergons that contain other Chambergons (or other Chambergons with Clubs) in both chambers.
Start with the Chambergons that contain pieces in both chambers that you have already activated/deactivated. Don't jump to the outermost Chambergon and work your way in. Keep working your way from the inside out.

Step 10: Make sure to activate/deactivate each Chambergon's Clubs before you "move up/out" to activating/deactivating the Chambergon that contains it.

Once you have done the ten steps above for each Card, you are done. You have performed the Chambergon Battle Logic equivalent of "filling in the truth table" for the proposition represented by the contents of Card 1.

2.3.6: The Point of Activation Games

Logic is about arguments, and arguments are about showing that one thing must be true since one or more other things is true. A valid argument, in fact, is one in which if the premises are true, the conclusion has to be true. It's as if the truth of the premises flows into the conclusion, making it true too. Or, rather, it's as if the premises and conclusion are structured in such a way that if there is truth in the premises, it can flow into the conclusion and make it true too.

As we've noted before, however, symbolic logic is primarily used by people working with computers, and computers care about 1's and 0's—or, rather, about electricity's being one or off—rather than about truth. To put the issue in electronic terms, then, logic is the study of how to structure complex electronic circuits such that if you power up the inputs (the "premises") electricity will reliably flow through and power up the output (the "conclusion").

Imagine, therefore, that Chambergons represent particular electronic circuits or "gates." If we want to power them up—and then use them as inputs for calculations or what-have-you—we first have to connect them to power sources in the right way. But we have two challenges. First, the primary power sources in Chambergon Battle Logic are Characters and Clubs. They, as it were, are direct links to the Primal Energy at the Heart of Reality. Second, different types of Chambergon react in different ways to different inputs. You can't get a Paragon to light up by feeding two deactivated Characters into it. But you can light up a Pyramid by plugging in two deactivated Characters. (It simply needs to be grounded, as it were.)

To know how to construct complex circuits through which electricity will reliably flow (so we can perform calculations or run searches or control robots or whatever), then, we need to know in exactly what combinations of activated and deactivated Characters we can get larger, more complex Chambergons to light up. This is what we discover by playing Activation Games. We can then use this information, in a way we will discuss later, to determine if an entire argument forms a coherent circuit.

Chapter 3: Level 3 Powers

3.0: A Logician's Three Fundamental Moves

If you give us any one of the three basic Chambergons (i.e., Parcels, Paragons, or Pyramids), we now know how to change it into any one of the others. If you give us a Parcel, we can give you a Paragon (using De Morgan's Theorem). If you give us a Paragon, we can give you a Pyramid (using Implication). So, if you give us a Parcel, we can give you a Pyramid. (Later, you will remember this and say, "Ah! That was a Hypothetical Syllogism. Foreshadowing!") But both De Morgan and Implication work in the other direction too, so if we started with a Pyramid, we could end with a Paragon or Parcel.

Changing one shape into another is one of the fundamental "moves" a logician makes. Another is giving and taking away Clubs. We can now add or remove two Clubs (using Double Negation) or add or remove only one Club (using De Morgan or Implication).

A third fundamental move is taking pieces out of, or putting them into, other pieces. This is like giving and taking away Clubs, in that the logician is adding things together, or taking them apart. The difference, however, is that you add (or remove) Clubs to (or from) the outside of other pieces, while we also need to be able insert (or extract) pieces into (or from) inside other pieces.

The logician's three fundamental moves, then, have to do with:

1. adding something to (or removing something from) a shape's outside,
2. adding something to (or removing something from) a shape's inside, or
3. changing the shape itself.

Every game we logicians play amounts to making these three moves in particular combinations. At each point in each puzzle, we have a choice: either we change one shape into another, or we add or remove Clubs from something, or take something out of (or put something into) something else. Which move we make at which point in the game depends on what we have available, and where we want to end up.

Our current problem is we don't know how to get things out of, or into, Pyramids directly.

- We can get things into a Parcel using Conjunction Introduction.
- We can get things out of a Parcel using Conjunction Elimination.
- We can get things into a Paragon using Disjunction Introduction.
- We can get things out of a Paragon using Disjunction Elimination.

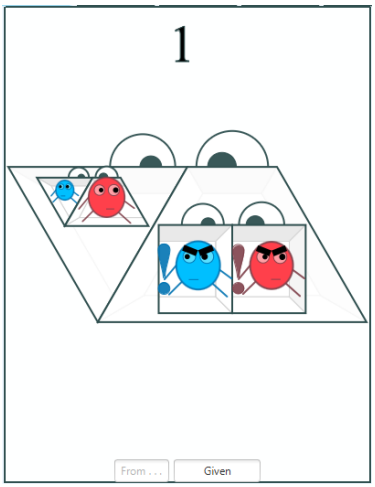
But if we want to get things into a Pyramid, we have to first get it into a Parcel or Paragon, and then change the shape into a Pyramid. And if we want to get something out of a Pyramid, we first have to change the Pyramid into a Parcel or Paragon, and then extract the contents we are after.

We could save ourselves a step, then, if we only had a way of getting things into or out of Pyramids themselves. Fortunately for us, there are ways in which this can be done. The problem, however, is that there is not just one way of getting something into a Pyramid, nor just one way of getting something out of a Pyramid. There is neither a single "Conditional Introduction" rule, nor a single "Conditional Elimination" rule. Rather, there are multiple ways of getting things into and out of Pyramids.

In this chapter, we will learn one method for getting things out of Pyramids, one for getting things that are in separate Pyramids together into a single Pyramid, and one for getting things out of two Pyramids into a single Paragon. In the next chapter, we will learn a second way for getting something out of a Pyramid, as well as one (of two) way(s) for moving pieces from the attic to the basement (or from the basement to the attic) of a Pyramid. Then, in chapter 5, we'll learn yet another way of getting things out of a Pyramid, and yet another way of getting things into a Pyramid.

Pyramids, in other words, are the most complex pieces in the game of Chambergon Battle Logic. I'm not sure why this is. Perhaps it's because you can make a Pyramid out of either a Paragon or a Parcel (by way of a Paragon), and thus the rules for both pieces apply in some kind of morphed form to Pyramids. (But, then, that would imply that Paragons are more complex than Parcels, and I'm not sure if that's true.) This is really interesting. I shall have to ponder it more.

In any event, you're going to either hate Pyramids for being so complex, or love them for opening so many new possibilities. Or maybe you'll just be indifferent to them. In other words:



(Imagine that the blue Characters represent, "You will hate Pyramids," red Characters represent, "You will love Pyramids," and the Characters with Clubs represent the denials of those two propositions.)

3.1: The Power of Modus Ponens (A Landfill/Inference Rule)

The motto for this rule is, "Match the attic, get the basement." (Its name is a Latin phrase for "the way of positing/placing/laying down/asserting/affirming.") A Pyramid is like a sign that says:

> Whatever is in my attic depends on/rests on/relies on whatever is in my basement. You can't have whatever's in the attic without also having whatever's in the basement, because attics can't float in thin air. So, if you see whatever's in my attic running around by itself, you can bet that whatever's in my basement is also out there running around too.

If the outermost Chambergon in one Card is a Pyramid, and if all you have in another Card is an exact match for whatever is in the attic of the Pyramid, you get to place whatever is in the basement of the pyramid in another card/card/card, all by itself.

Another way of thinking about Modus Ponens is to think of Pyramids like vending or gumball machines. Whatever is in the attic is the price, and whatever is in the basement is the candy bar, bag of chips, or gumball you'll get for that price. So, if you put the right coin in the machine (something that exactly matches whatever is in the attic) out will pop the "prize" in the basement.

Restriction: This rule won't work if the Pyramid has a Club. Likewise, it won't work if the Pyramid is inside some other Chambergon (even if that Chambergon is itself a Pyramid).

The logical justification for this rule is rather simple. A conditional statement says that if one proposition is true, then another one must be true too. Given that fact, if you know that the first proposition is true, then you can conclude that the second one must be true as well.

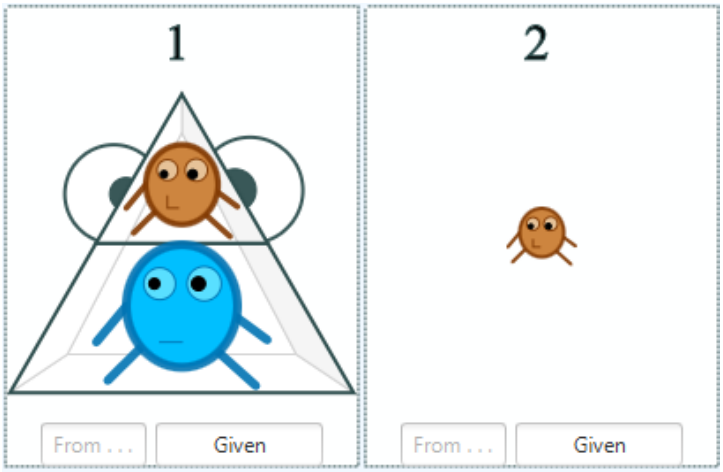
For example: (1) "If you take a logic class, you will learn a lot of amazing and wondrous things." (2) "You are taking a logic class at McDaniel this semester." Therefore, (3) "You will learn a lot of amazing and wondrous things."

In symbols, Modus Ponens looks like this:

Proposition	**Justification**
1. (A –> B)	Given
2. A	Given
∴ 3. B	1, 2 MP [where "∴" stands for "therefore."]

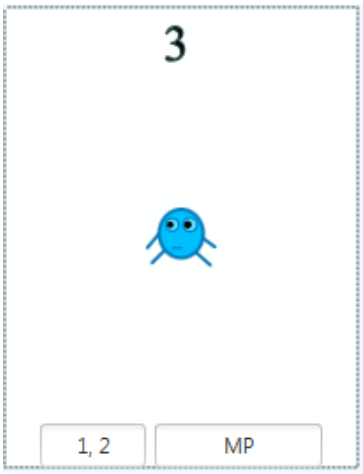
In Chambergon Battle Logic, however, Modus Ponens looks like this:

Example 1



If we are given these two Cards, what third Card can we create, if we use Modus Ponens?

The answer is this:



In Chambergon Battle Logic, brown Characters stand for "A," and blue Characters stand for "B." This example, then, is the same as the "symbols" example written on the previous page, with "1. (A –> B)" being the first Card, "2. A" being the second Card, and "3. B" being the third Card.

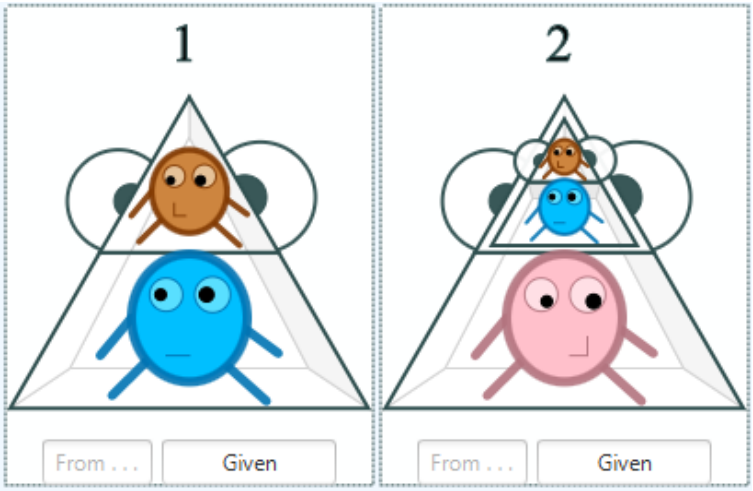
The Pyramid in Card 2 shows us that all brown Characters in this particular game depend on blue Characters. Or, to put it another way, each brown Character is supported by, or backed up by, a blue Character. (This is why the brown Character is placed on top of the blue Character in the Pyramid; it "rests on" or is "upheld by" the blue Character.)

After looking at Card 1, we then turn to Card 2, where we are delighted to discover a brown Character wandering around by itself. Given what the Pyramid in Card 1 tells us, we can now conclude that there must be a blue Character out there as well, supporting the brown Character. It's just the blue Character has been hiding, so we haven't seen him yet. But we can find him,

given what we know from Cards 1 and 2. We create Card 3 by using Modus Ponens to drive the blue Character out of hiding.

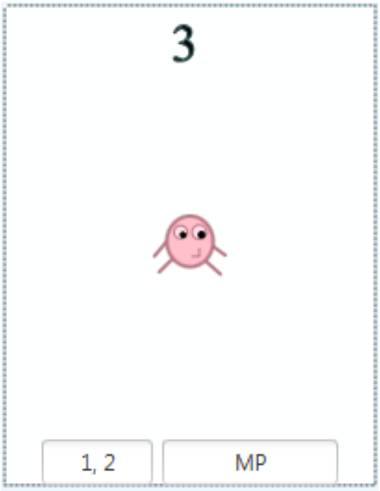
(It is as if the brown Character were an advance scout for the main body of an army, while the blue Character is the army itself. The first Card tells us that scouts are supported by armies. The second Card tells us we've found a scout. So, we get to conclude in the third Card that there is an army out there.)

Example 2



Imagine that we were given these two Cards. If we were given these two Cards, what third Card could we create, using Modus Ponens?

The answer is:

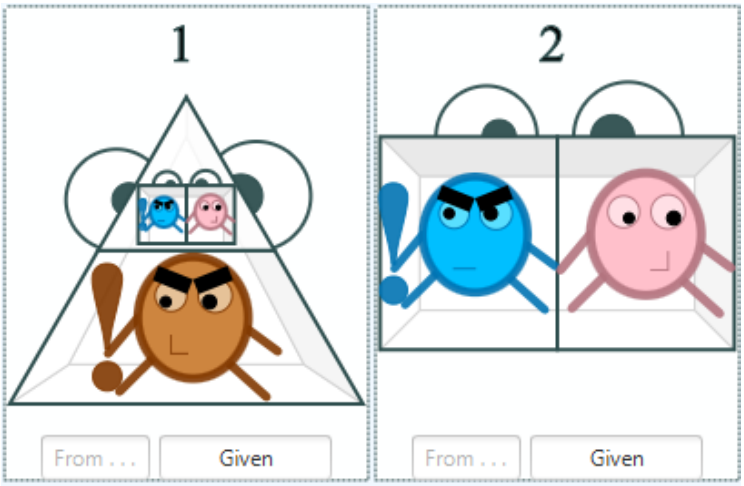


In this example, the tables have turned. Here, the Pyramid in the first Card acts as the "payment" we put into the attic of the Pyramid in the second Card. Modus Ponens tells us that if we match the attic of a Pyramid, we get whatever's in the basement. It's just that in this example, what's in the attic of the Pyramid in Card 2 is itself a Pyramid. It's as if we put a gumball machine into another gumball machine as payment for the pink gumball we get in Card 3.

The Pyramid in Card 1 tells us that brown Characters rest on, or depend on, blue Characters. The second Card, however, tells us that this dependence itself depends on pink Characters. The pink Character supports the blue Character's supporting the brown Character. (If the brown Character is like a scout, and the blue Character is like the army supporting the scout, the pink Character is like the taxpayers who make it possible for the army to support the scout.)

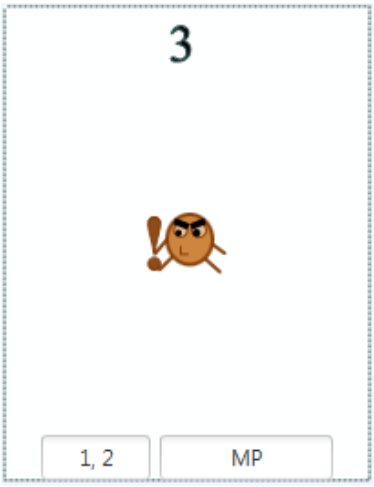
So, given that we know from Card 1 that blue Characters are needed to support brown Characters, and given from Card 2 that pink Characters are needed to support blue Characters' supporting brown Characters, we conclude in Card 3 that there must be a pink Character "out there somewhere." Otherwise, the support relationships in Cards 1 and 2 would break down.

Example 3



Imagine that we were given the above two Cards to start a game. What third Card could we make from them, using Modus Ponens?

The answer is:



In this example we have something very similar to Example 2. We have a Chambergon (in Card 2) that matches the content of the attic of some Pyramid (in Card 1). The difference here is twofold.

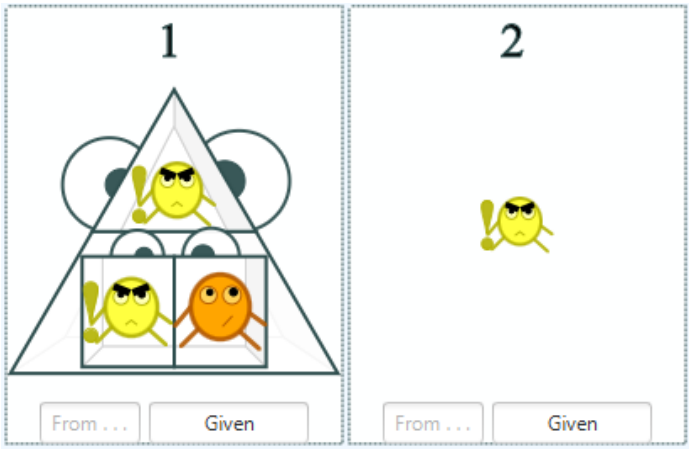
First, the order is different. We are given the Pyramid first, and then the thing that matches the content of its attic second. This is fine. The order in which you get the Pyramid and the matching Character or Chambergon is irrelevant, so long as you have both available to you somewhere in the game. You may see the gumball machine first, then pull the change from your pocket, or you may already be holding the change in your hand when you stumble across the gumball machine. Either way, you have the machine and the payment, you're ready to buy some candy.

The second difference is that the matching Chambergon in this example is a Parcel, rather than being another Pyramid (like in the previous example). This, likewise, is irrelevant. So long as (a) you have a Character or Chambergon as the main occupant of some Card, and (b) that Character or Chambergon exactly matches the content of the attic of a Pyramid that is *itself* the main occupant of some *other* Card, then (c) you can use Modus Ponens.

Now, the Pyramid in Card 1 contains two Clubs, and we know that a Pyramid with Clubs cannot be paid to "spit out" whatever is in its basement. It will reject all attempts at payment, like one of those annoying vending machines that always spits your dollar bill back out.

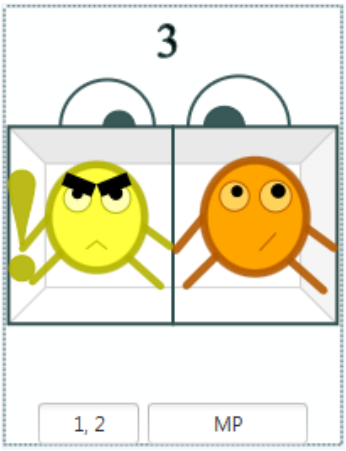
In this case, however, the Clubs are not a problem. If the Pyramid itself had a Club, then we couldn't use Modus Ponens. But since the Clubs are inside the Pyramid, everything is clear. What this Pyramid tells us is that every blue/pink Parcel (in which the blue Character has a Club) is supported by a brown Character with a Club. Since we discover a blue/pink Parcel, in which the blue Character has a Club, in Card 2, we then can use Modus Ponens to drive the "A" Character with a Club out of hiding.

Example 4



Imagine that we open a game, and find the two Cards above as our "Givens." What third Card could we create from them, using Modus Ponens?

The answer is this:



Notice here that if we had been given a yellow Character (with no Club) in Card 2, we wouldn't have been able to use Modus Ponens. It wouldn't have exactly matched the content of the attic of the Pyramid in Card 1. What the Pyramid in Card 1 tells us is that yellow Characters who have Clubs can only function on the battlefield if they are supported by a yellow/orange Parcel, in which the yellow Character also has a Club. The Pyramid in Card 1 tells us nothing about yellow Characters who don't have Clubs. They may be able to operate completely independently, for all we know.

So, since (a) we've discovered a yellow Character with a Club, in Card 2, and (b) we know from Card 1 that yellow Characters with Clubs rest on, or depend on, yellow/orange Parcels in which the yellow Character also has a Club, we conclude (c) that there must be just such a Parcel out there. As a result, we put the Parcel that we know must be out there in Card 3.

Now, you might ask why the yellow Character with a Club in Card 2 doesn't attack the attic of the Pyramid, like it would if we have a Paragon in Card 1 with a yellow Character in one of its Chambers. There are two reasons it doesn't do so. First, it only hates other yellow Characters who have no Clubs, and so it has no motivation for attacking. Second, even if it wanted to attack, it wouldn't be able to reach. The yellow Character with the Club in Card 2 is on ground level, while the yellow Character with the Club in Card 1 is above ground level in the attic of a Pyramid.

3.2: The Power of Hypothetical Syllogism (A Landfill/Inference Rule)

We've already seen one way in which Pyramids can be used together. If one Pyramid exactly matches the contents of the attic of another Pyramid, we can use Modus Ponens to extract the contents of the basement of the second Pyramid. With Hypothetical Syllogism, however, we combine two Pyramids to make a third.

The motto for Hypothetical Syllogism is: "If the basement of one Pyramid matches the attic of the other, they crash and cancel out; so you build a third Pyramid with what's left." What happens here is very similar to what goes on in Constructive Dilemma, which we will study next. When you have two Pyramids, and the attic of one matches the basement of the other, whatever is in the matching attic is excited to see its compatriot in the basement of the other Pyramid.

Steering its own Pyramid towards the other, it accidentally crashes into it; this is what happens when people drive Pyramids overenthusiastically. When the dust clears, the matching attic and basement have been destroyed, and all that's left over is the attic and basement that didn't match. They then have no choice but to form a new Pyramid together.

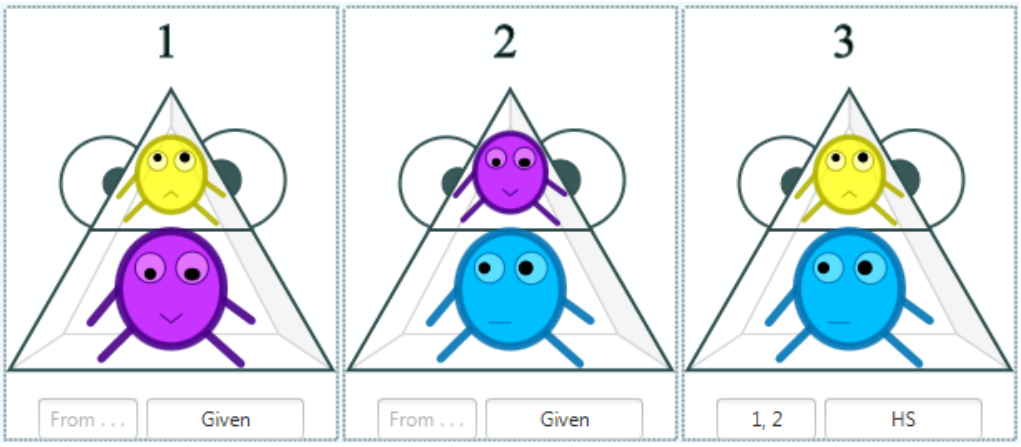
Restriction: If either Pyramid has a Club, this rule won't work. The Club will get in the way, and nothing will happen.

The logical justification for Hypothetical Syllogism is straightforward. Imagine that someone in Washington, DC says, "If you want to get to New York, you've got to go through Philly. But if you want to go through Philly, you've got to go through Baltimore first." The person in question just made two conditional statements: "If you want to get to New York, you've got to go through Philly," and, "if you want to go through Philly, you've got to go through Baltimore first." Having been told this, we would be perfectly justified in responding, "So, if we want to get to New York, we've got to go through Baltimore first." We would summarize the other person's two conditional statements with a conditional statement of our own.

Let's use "B" for "you've got to go through Baltimore first," "P" for "you've got to go through Philly," and "Y" for "you want to reach New York." If we do that, then our Hypothetical Syllogism would look like this

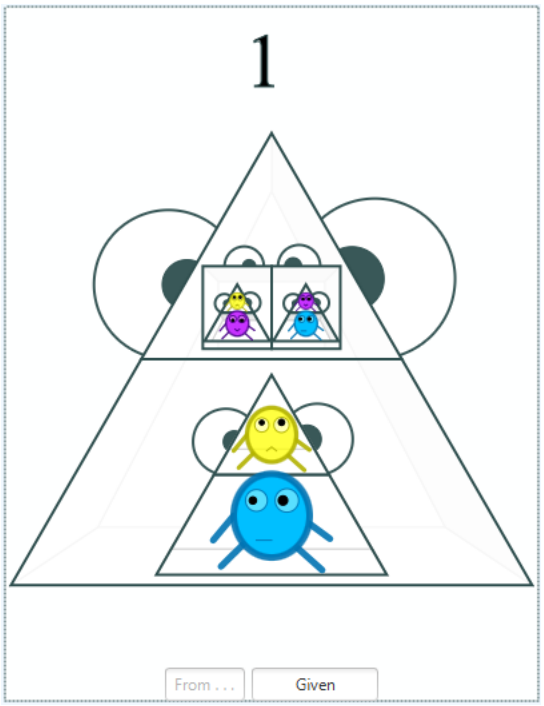
Proposition	**Justification**
1. (Y –> P)	Given
2. (P –> B)	Given
∴ 3. (Y –> B)	1, 2 HS

In Chambergons, that would look like this:



This says that if yellow Characters depend on purple Characters (Card 1), and purple Characters depend on blue Characters (Card 2), yellow Characters must ultimately depend on blue Characters (Card 3).

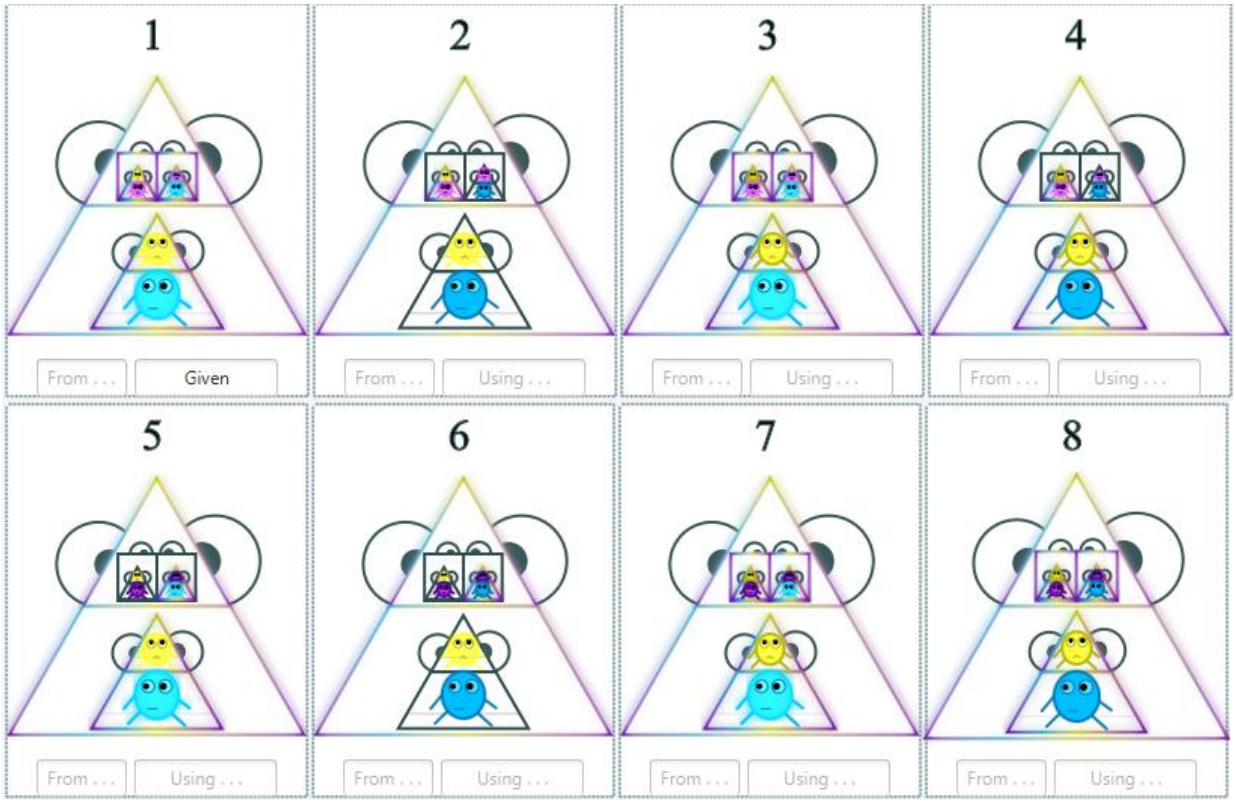
Now, take a moment and think about the logical structure of the previous sentence. Do you see the “if” near its beginning and the “and” about a third of the way in? Do you see where the “then” would go, if it hadn’t been left implicit? That sentence would be represented like this, in Chambergon Battle Logic:



If we want to know whether the proposition that that represents is true, we have to know whether its parts are true. Luckily for us, we know how to check all the possible combinations of truth and falsity. We count the number of colors in the above Card (there are three) and multiply that number of 2’s together ($2 \times 2 \times 2 = 8$). We then create that many identical cards and treat the whole thing like an Activation Game.

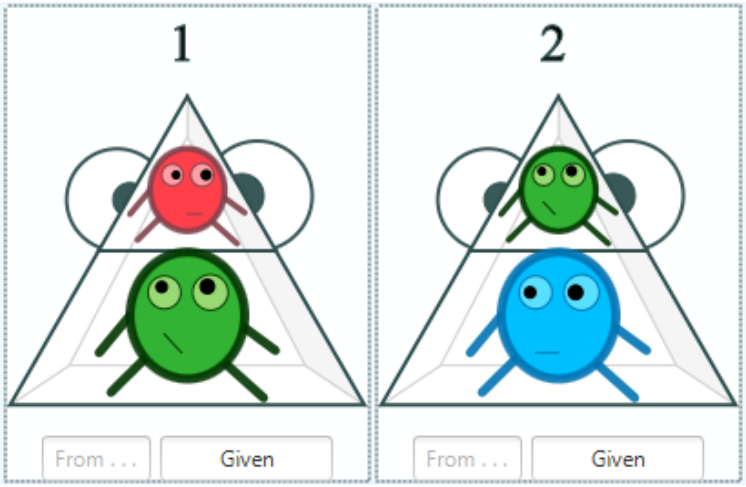
To make sure that every possible combination of activation and deactivation (truth and falsity) is covered, we activate the blue Characters in every other Card, the yellow Characters in a 2-on/2-off pattern, and the purple Characters in a 4-on/4-off pattern. Then we figure out whether the internal Pyramids should be activated or not. Then we figure out if the Parcels in the attics of the outside Pyramids should be activated. Then we figure out whether the outside Pyramids should be activated.

If we do this, we obtain the following result.



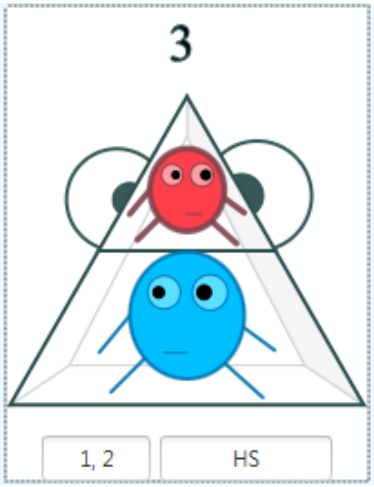
It turns out that no matter whether all three Character types are activated (Card 1) or deactivated (Card 8), or some combination of the two (Cards 2–7), the outermost Pyramid always winds up being activated. That is, propositions that have the form "(((A -> B) && (B -> C)) -> (A -> C))" are always true. And that means if you ever have two conditional propositions, where the consequent of one is identical to the antecedent of the other, you can just replace the consequent of the first with the consequent of the second (or, alternatively, you can replace the antecedent of the second with the antecedent of the first).

Example 1



Here, we are presented with two Pyramids. The basement of the first matches the attic of the second. What can we create with these two Cards, using Hypothetical Syllogism?

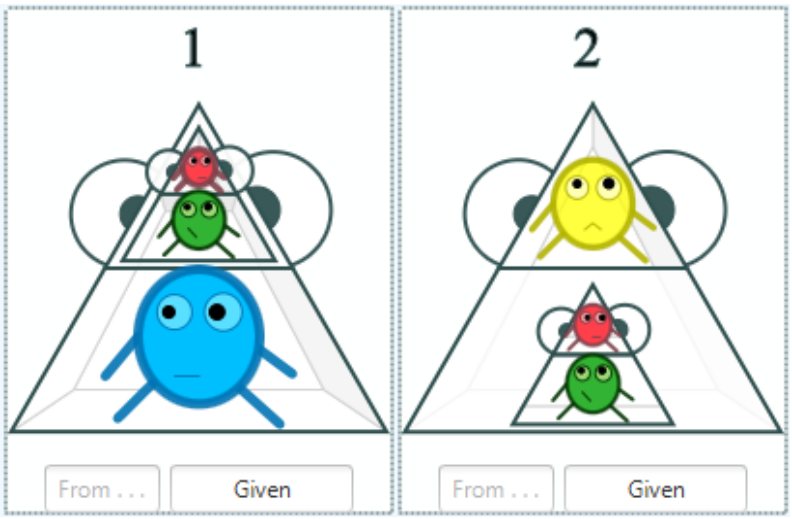
The answer is this:



This example is about as straightforward as Hypothetical Syllogism gets. The content of the basement of one Pyramid matches the content of the attic of the other, and so we create a new Pyramid using the non-matching attic and basement.

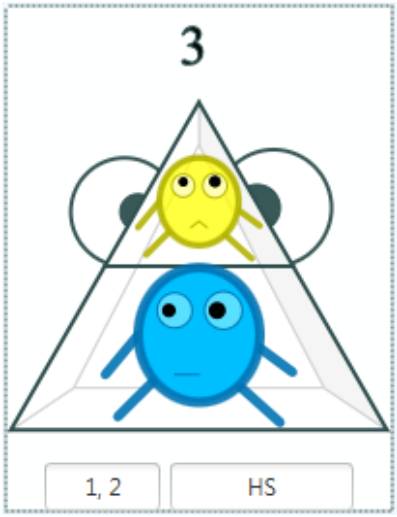
(Can you see how this might happen if the Pyramids in Cards 1 and 2 were to collide, such that the basement of the first crushed the attic or the second, or such that the attic of the second ruptured the basement of the first? All that would be left in the rubble would be the attic of the first, and the basement of the second.)

Example 2



If we were given these two Cards at the start of a game, what third Card could we create, using Hypothetical Syllogism?

The answer is this:

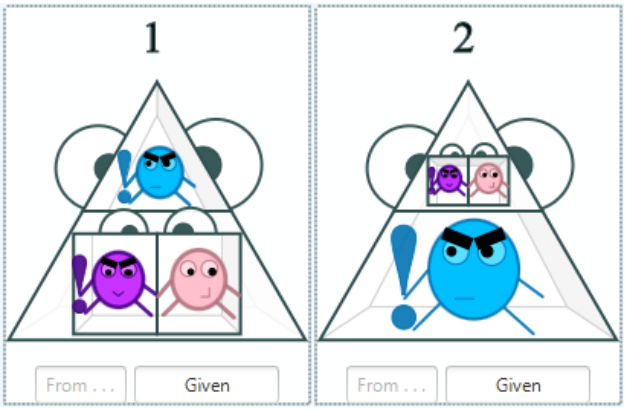


The attic of the Pyramid in Card 1 contains another Pyramid. But that "attic Pyramid" is identical to the Pyramid in the basement of the Pyramid in Card 2. This match between the contents of one attic and another basement is all we need for Hypothetical Syllogism.

One lesson of this example is that it doesn't matter how complicated the contents of the matching attic and basement are. In the first example, one green Character matched the other. In this example, what matches is a red/green Pyramid. You find it both in the attic of one Pyramid and the basement of the other.

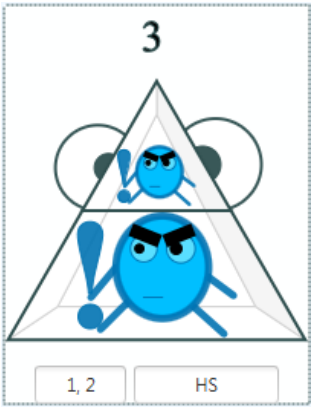
Another lesson is that order doesn't matter. That is, the Pyramid with the matching attic comes first in this example, even though it came second in the previous example. It doesn't matter. So long as you have one Pyramid (that is the main occupant of some Card somewhere) whose attic matches the basement of another Pyramid (that is the main occupant of some other Card somewhere), you can use Hypothetical Syllogism.

Example 3

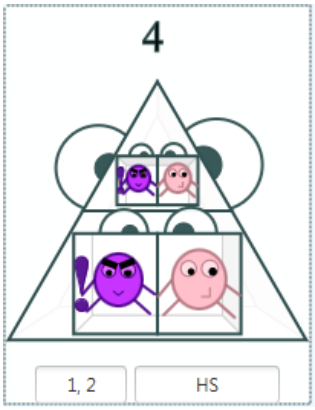


If we were given the above Cards to start a new game, what could we produce using Hypothetical Syllogism?

The answer is that we could produce either this:



. . . or this:



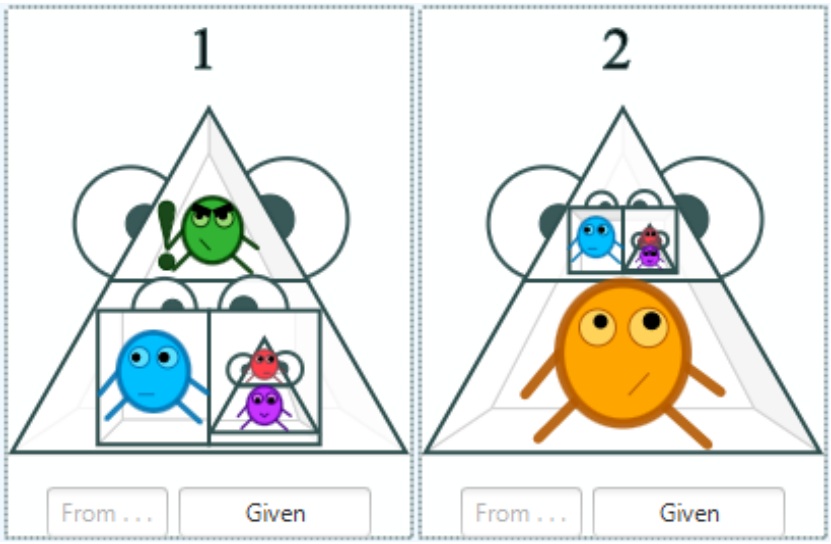
Either Card 3 or Card 4 would be a legitimate product of applying Hypothetical Syllogism to Cards 1 and 2.

There are three things to notice with this example. First, when the attic of one Pyramid matches the basement of the other, while its basement matches the attic of the other, you can use Hypothetical Syllogism twice, and produce two different results. This is what happens above, in Cards 3 and 4. Remember, just because you've used a Character or Chambergon once, doesn't mean you can't use it again. And just because you've used a Character or Chambergon in one way, doesn't mean you can't use it in another.

Second, it doesn't matter that the blue and purple Characters all have Clubs. That doesn't get in the way of Hypothetical Syllogism. All that matters is that both blue Characters match, in that both are blue Characters and both have Clubs. If either hadn't had a Club, or if either of the Pyramids in Cards 1 and 2 had Clubs, then we would've run into problems.

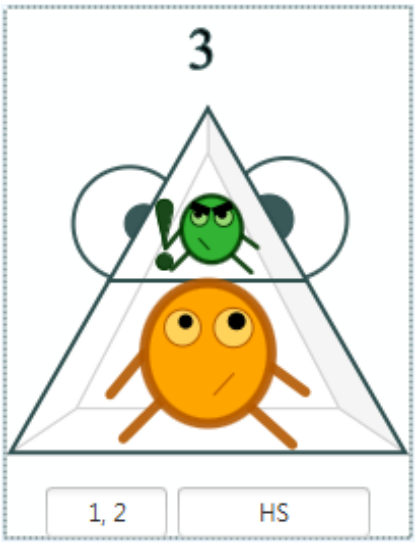
Third, it doesn't matter that the Pyramid you end up constructing has the same thing in its basement and its attic. That's cool. This is simply evidence that the two Pyramids you began with matched in two ways: not only did the attic of one match the basement of the other, but the basement of the first also matched the attic of the second.

Example 4



Imagine that we begin a new game, and find that we have been given the two Cards above. What can we do with them? Specifically, if we were to use Hypothetical Syllogism, what new Card could we produce?

The answer is this:



What we produce is greatly simplified, compared to either of the Cards used to produce it. Both those Cards contain Pyramids that contain a Parcel that contains another Pyramid. Through Hypothetical Syllogism, however, we have been able to simplify them down to a single Pyramid containing two Characters.

The reason this worked is that the complexities of the original Pyramids (in Cards 1 and 2) matched each other exactly. In the basement of one Pyramid, there was a Parcel that was identical to the Parcel in the attic of the other Pyramid. And those two Parcels were identical

because they each contained a blue Character in the left chamber, and a Pyramid in the right. And those two "right chamber Pyramids" were identical because each contained a red Character in its attic and a purple Character in its basement.

If you have this kind of match—a "match all the way down"—between what is in the attic of one Pyramid and what is in the basement of another (and if those two Pyramids are the outermost pieces in their respective Cards), then you can use Hypothetical Syllogism to combine them. The new Pyramid you produce may be more simple than the originals—as in the current example—but it may not be. It all depends on what is in the "non-matching" chambers of the two Pyramids you're combining.

3.3: The Power of Constructive Dilemma (A Landfill/Inference Rule)

You may or may not have noticed, but this is the third landfill rule we've done in a row. There are some recycling rules for Pyramids, but landfill rules predominate (for some reason). Constructive Dilemma, however, is different from the other Pyramid landfill rules because it requires three Cards as input, rather than two. (If you're into trivia, Constructive Dilemma is the only rule in logic that requires three Cards as input. Again, I don't know why this is, but it is.)

The motto for Constructive Dilemma is, "Two Pyramids can invade a matching Paragon and repopulate it." If you have two Pyramids available (that is, if you have two Cards whose primary occupants are Pyramids), as well as a Paragon available, and the contents of the attic of one Pyramid matches the contents of the left chamber of the Paragon, while the contents of the attic of the other Pyramid matches the contents of the right chamber of the Paragon, you can construct a second Paragon from the *basements* of the two Pyramids.

What has actually happened, however, is that the two Pyramids have punctured the two chambers of the Paragon (the one, having flipped upside down, invades the left chamber from above, and the other invades the right chamber from below), the matching contents have perished in the collision, and the contents of the basements of the Pyramids have moved into the now vacant chambers.

Why would the Pyramids do this, though? Why would they invade the Paragon, eliminating not only the occupants of their attics, but also the occupants of the Paragon? The answer is, "fellow feeling." Paragons are the tanks of the Semantic Realm, and require their occupants to live in close quarters for days, weeks, and years on end. As you know from any family car trip, this inevitably leads to the contents of both chambers being seriously irritated by the contents of the other chamber, and this produces a strong desire on the part of both parties to leave the Paragon. However, their sense of duty—and their massive irritation with whoever is in the other chamber—means that neither will let the other desert its post. Desiring to free its match from such a horrible situation, the Character or Chambergon in the attic of each Pyramid recklessly guides its Pyramid toward the Paragon prison, each trying to free its match trapped within those horrible walls. But they are overzealous, and what actually occurs is a catastrophe.

But why should the occupants of the basements of the two Pyramids end up trapped inside the Paragon at the end of the process? Well, they don't do it on purpose. Each is just "coming upstairs" to see what in the world has happened—to check on the occupant of the attic. After all,

it's the job of the basement of a Pyramid to support the attic, and the occupants of the basements feel a certain protectiveness toward the occupants of the attic.

Of course, once they get upstairs, they find themselves not in the attic of a Pyramid, but in the chamber of a Paragon. They look across the internal barrier between chambers, and their eyes meet. They say, "In honor of our fallen comrades, let us carry on the work they were doing, here in this Paragon." And so they do. It's just that after a while of driving around in the tank that is the Paragon, they become thoroughly disgusted and annoyed with each other, and the whole cycle starts all over again.

Restriction: This rule will *not* work if either of the two Pyramids, or the initial Paragon, has a Club. If any of the original three Chambergons has a Club, that Club will get in the way of the process and nothing will come of it. The rule will also only work if the two Pyramids and the Paragon are available in three different Cards (that is, it will only work if none of them are contained within larger Chambergons).

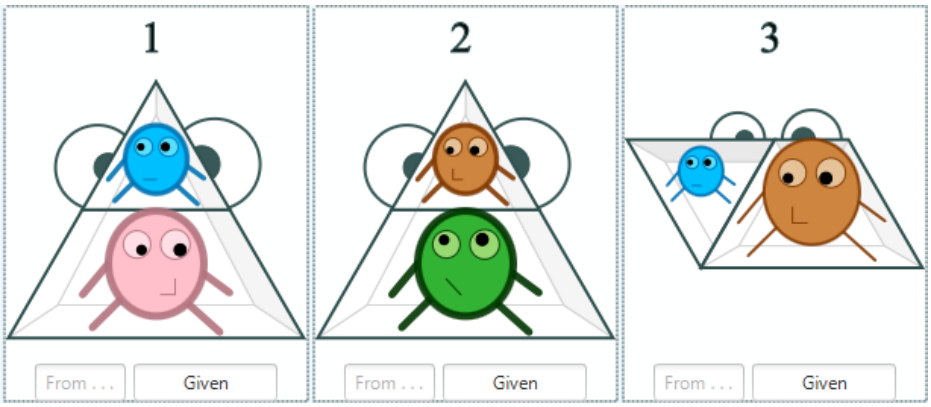
Note: The reason this rule is called "Constructive Dilemma" is that with it you are "constructing a dilemma." A "dilemma" after all, is a situation in which you have two choices: you can choose this, or you can choose that. It's an "or" kind of situation. This or that. What you are doing in Constructive Dilemma is creating a new "or" situation. After all, the Paragon embodies "or" statements, and you're making a new Paragon. So, you're constructing a dilemma.

The logical justification for this rule goes as follows. Imagine that you have a choice between two options. Perhaps you can go to a theme park with your friend's family, or stay home and study for a big exam. We would represent that as a Paragon with "I go to the theme park" in the left chamber, and "I stay home to study" in the right chamber.

Now, imagine that each choice has a particular consequence. If you go to the theme park, you will fail the exam, for instance. But if you stay home and study, you'll be ostracized for being uncool. We would represent both of these situations as Pyramids. One would have, "I go to the theme park," in its attic and, "I fail the exam," in its basement. The other would have, "I stay home to study," in its attic and, "I am ostracized for being uncool," in its basement.

If you are asked to choose between going to the theme park and staying home to study, therefore, you are ultimately being asked to choose between failing your exam, on the one hand, and being ostracized for being uncool, on the other. Your initial dilemma leads to another dilemma, because the Paragon representing your initial choice had two Pyramids "descending" from it.

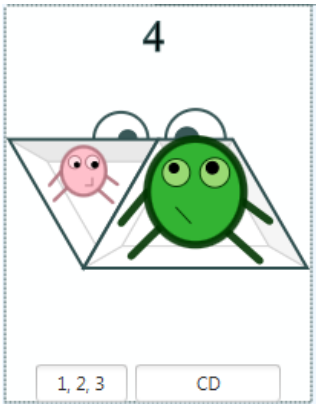
Example 1



Here, we are given three Cards. Two are Pyramids and one is a Paragon. The contents of the attic of Pyramid in Card 1 match the contents of the left chamber of the Paragon in Card 3. Similarly, the contents of the attic of the Pyramid in Card 2 match the contents of the right chamber of the Paragon in Card 3.

What can we do with these three Cards? Seeing two Pyramids, we might be tempted to try Hypothetical Syllogism. But that would require the attic of one Pyramid to match the basement of the other, and that doesn't happen here. Instead, we notice the match between the Pyramids' attics and the chambers of the Paragon.

So, we do this:

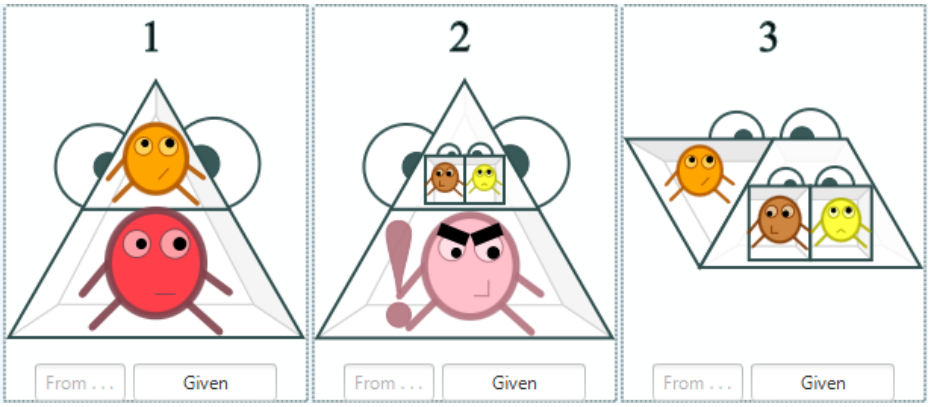


Notice how the left chamber of the Paragon in Card 3 looks like the attic of the Pyramid in Card 1, only upside down. Notice furthermore how the right chamber of the Paragon in Card 3 looks like the attic of the Pyramid in Card 2, only with the top chopped off. This means we can draw a new Paragon in Card 4, with (a) the blue Character in the left chamber of the Paragon in Card 3 replaced by the pink Character that supports the blue Character in the Pyramid in Card 1, and (b) the brown Character in the right chamber of the Paragon in Card 3 replaced by the green Character that supports the brown Character in the Pyramid in Card 2.

What has happened is as described above. The blue and brown Characters in the attics of the Pyramids, wishing to free their comrades from the prison of the Paragon, accidentally destroy their comrades. The pink and green Characters in the basements of the Pyramids, coming

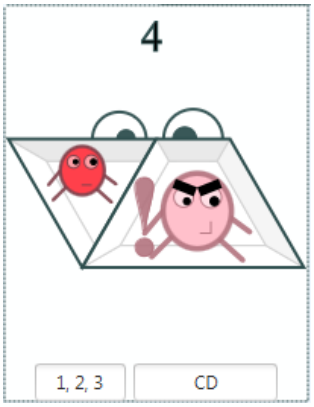
upstairs to see what is happened, then find themselves trapped in the very chambers from which they were attempting to free the blue and brown Characters.

Example 2



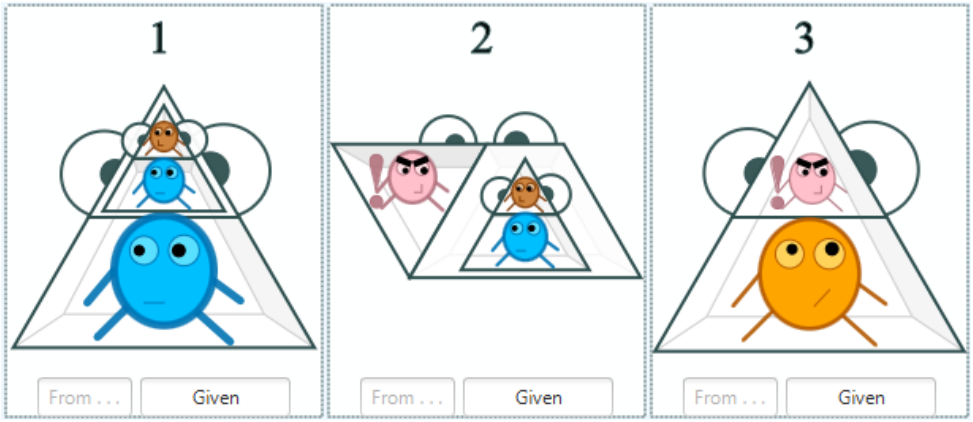
Here, we have another set of two Pyramids and a Paragon. Furthermore, the attics of the two Pyramids match the contents of the Paragon. We can use Constructive Dilemma, therefore. But what would we produce if we did?

The answer is this:



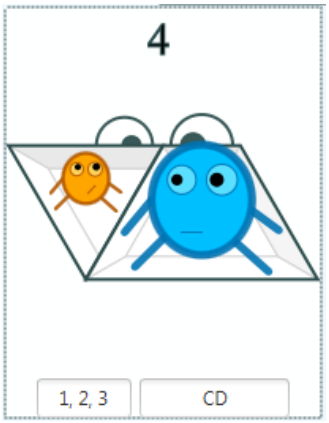
The lesson in this example is that it doesn't matter how complex the contents of the Paragon and the attics of the Pyramids are. So long as the contents of each chamber of the Paragon (here, in Card 3) match the contents of the attic of at least one Pyramid (here, in Cards 1 and 2), then we can use Constructive Dilemma.

Example 3



Once again, we have two Pyramids and a Paragon, and the attics of the Pyramids match the chambers of the Paragon. Given these three Cards, what can we use Constructive Dilemma to create?

The answer is this:

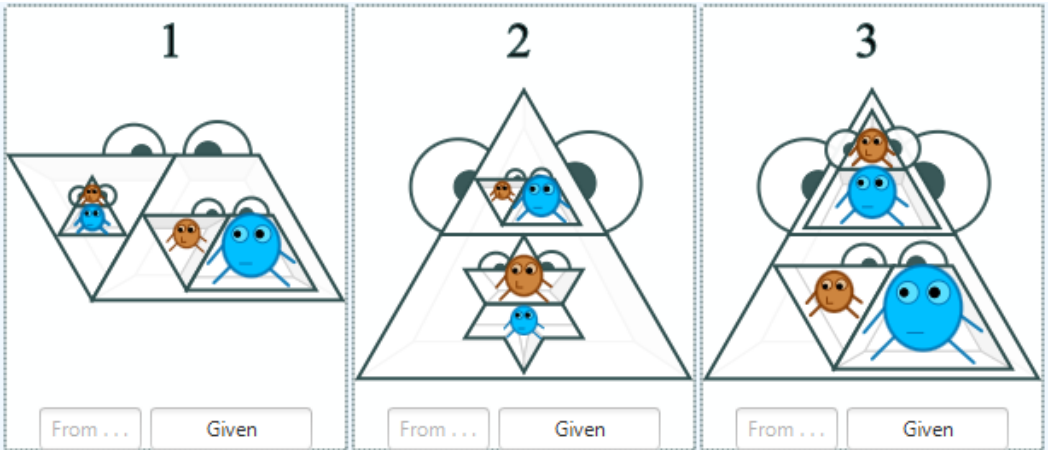


This example shows three things. First, it doesn't matter how complex the contents of the attics of the Pyramids and of the chambers of the Paragon are.

Second, it doesn't matter if said contents have a Club. It *would* matter if any of the two Pyramids or the Paragon themselves had a Club. But if their *contents* have Clubs, that doesn't matter.

Third, it doesn't matter in what order we get the two Pyramids and original Paragon. So long as we get two Pyramids and a Paragon, and the contents of the attic of one Pyramid matches the contents of one chamber of the Paragon, and the contents of the attic of the other Pyramid matches the contents of the other chamber of the Paragon, then we can use Constructive Dilemma.

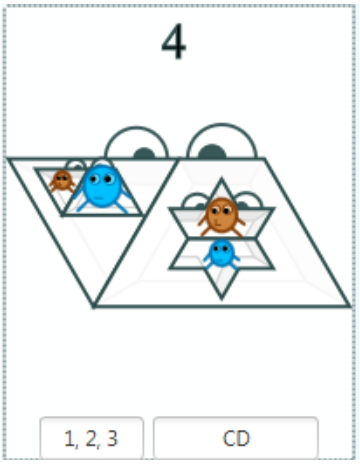
Example 4



Our three “Givens” look like a total mess, so start with the Paragon. Does one of its chambers contain something that is identical to what is in the attic of one of the two Pyramids? (Yes. Its left chamber contains a Pyramid that is identical to the Pyramid in the attic of the Pyramid in Card 3.) And does the other chamber contain something that is identical to what is in the attic of the other Pyramid? (Yes. Its right chamber contains a Paragon that is identical to the Paragon in the attic of the Pyramid in Card 2.)

So, what can we do with these three Cards, now that we have access to Constructive Dilemma?

The answer is this:



The point of this example is much like the previous. First, the complexity of the contents of the Paragon and the Pyramids doesn’t matter. Just so long as the contents of the attics of the Pyramids match the contents of the chambers of the Paragon, we’re good to go. And second, the order in which we get our original Paragon and Pyramids is irrelevant. So long as we get one Paragon and two Pyramids, and the contents match as described above, we’re good to go.

Chapter 4: Level 4 Powers and Encoding Games

4.0: Chapter Overview

We've got two Pyramid rules to cover in this chapter, as well as a rule that applies to Parcels and Paragons. I saved the "Parcels and Paragons" rule for now because it is the same as the second Pyramid rule we're going to cover—with one important difference. So, I thought, "Why not cover both rules at once? Then we can talk both about how they are the same and how they are different."

However, we also will be covering a third type of game. Our focus has been on pure puzzle-solving games to this point, but we have also been playing Activation Games. The new type of game we will discuss in this chapter is a variation on puzzle-solving games called "Encoding Games" (though some also involve decoding). These games are crucial to the success of our mission, so we will be paying them a lot of attention.

What was that? You didn't realize we had a mission? Oh. Well, we do. In fact, you have been participating in a top-secret project over the past few weeks and if you are able to master Encoding Games, you just may earn yourself a job offer from an organization currently carrying on the single most important battle in human history.

Drama.

4.1: The Power of Modus Tollens (A Landfill/Inference Rule)

The motto for this rule is, "Club the Basement, Freak Out the Attic." (Its name is a Latin phrase for "the way of denying/removing/picking up/taking back/negating.") If you have a Character or Chambergon that is the outermost occupant of its Card, and if that Character or Chambergon exactly matches the Character or Chambergon in the *basement* of a Pyramid that is the outermost occupant of *its* Card—*except the lone Character or Chambergon has exactly one more Club than the Character or Chambergon in the basement of the Pyramid*—then the following drama (more drama!) will occur.

The available Character or Chambergon will notice the Character or Chambergon in the basement of the Pyramid. It will see that it has one more Club than the Character or Chambergon in the basement of the Pyramid. It will be outraged at the existence of the Character or Chambergon in the basement, and will attempt to destroy it with its extra Club. To do this, however, it will have to destroy the basement of the Pyramid. This will cause the attic of the Pyramid to lose its support, and whatever is in the attic will fall out. Because of the fall, whatever was in the attic will be shocked, frightened, and enraged. And thus, whatever was in the attic will now have a Club.

That is, the thing that falls out of the attic will gain a Club, unless it already has one or more. In that case, when it falls out of the attic, it will simply have been restored to its original level. And, as we all know, this means it will relax, losing a Club.

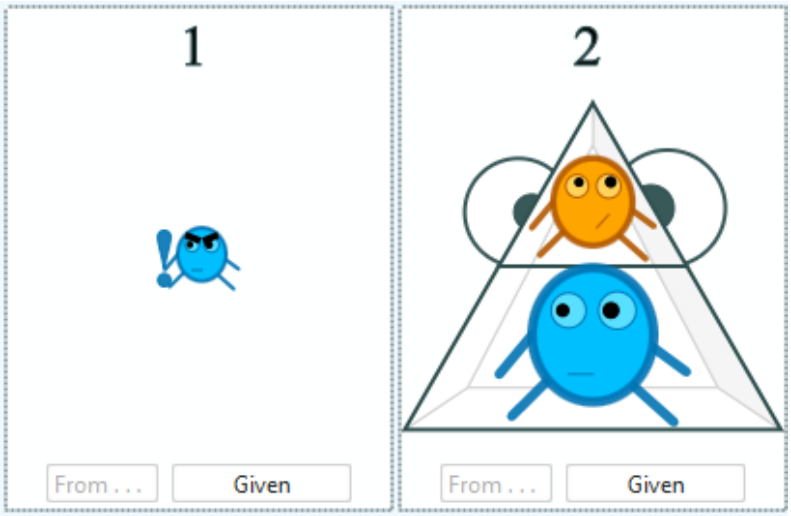
Restriction: This will not work if the Pyramid has a Club. If it does, it will fight off the attack, and nothing will end up happening. Likewise, this rule will not work if the Pyramid is inside some other Chambergon (even if that other Chambergon is a Pyramid).

The logical justification for this rule is the same is as follows. A Pyramid expresses a particular relationship between two propositions. It says that the truth of the proposition in the attic is a sufficient condition for the truth of the proposition in the basement. Likewise, it says that the truth of the proposition in the basement is a necessary condition for the truth of the proposition in the attic.

Now, imagine that we are told that "B" is a necessary condition of "A." In other words, if "A" is going to be true, "B" has to be true to. But imagine we know that "B" isn't true. If "B" isn't true, then we can conclude that "A" is also not true. For example, we all know that the truth of, "Things are falling," depends on the truth of, "There is gravity." But if we are in a place where there is no gravity. From that fact, we can also conclude that things are not falling.

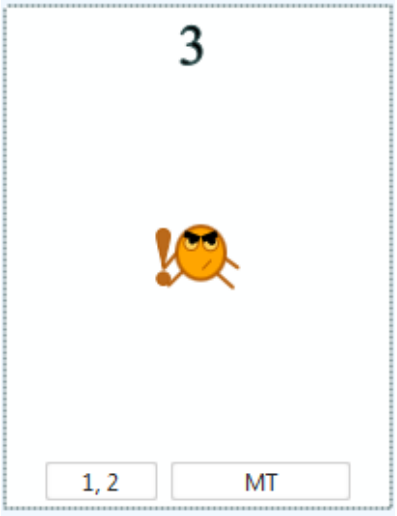
The logical justification for this rule could also be stated like this: Modus Tollens is identical to Implication plus Disjunction Introduction. If you figure out what that sentence means, you have a real knack for this logic stuff.

Example 1



Imagine that a game began with those two Cards, but we didn't have any set goal. Being perfectly free to do whatever is logically legal, furthermore, imagine that we decide to employ Modus Tollens. What would we produce, given these two Cards, using Modus Tollens.

There is but one answer, and that answer is:

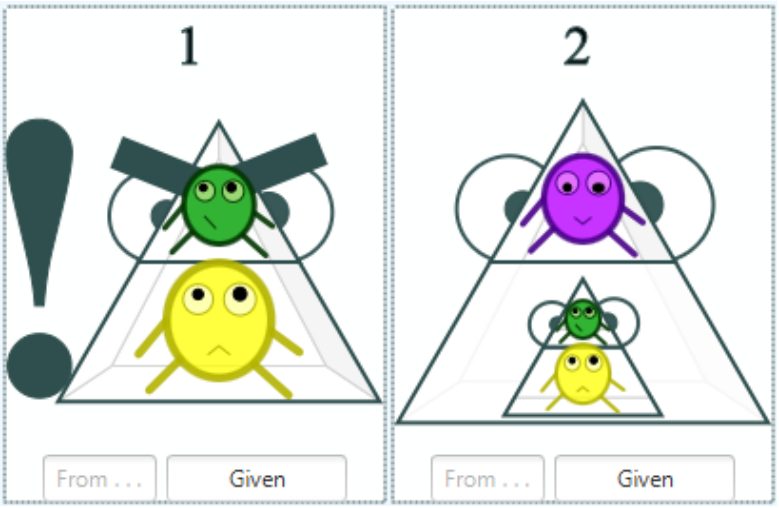


In Card 1 we have a blue Character with a Club. The Club, you will recall, has filled him with anger towards all blue Characters who have no Clubs. Spying just such a blue Character in the basement of the Pyramid, the blue Character with a Club from Card 1 is enraged. But to actually attack the blue Character in the basement, it must use its Club to destroy the basement of the Pyramid. Otherwise, the blue Character in the basement will remain safely behind the walls of the Pyramid.

When the blue Character with a Club destroys the basement, trying to get at the blue Character with no Club, the foundations of the Pyramid become unstable, and the orange Character in the attic comes tumbling out. The sudden fall frightens the orange Character, furthermore, and it exclaims. That is, it ends up with a Club of its own. Thus, we have Card 3.

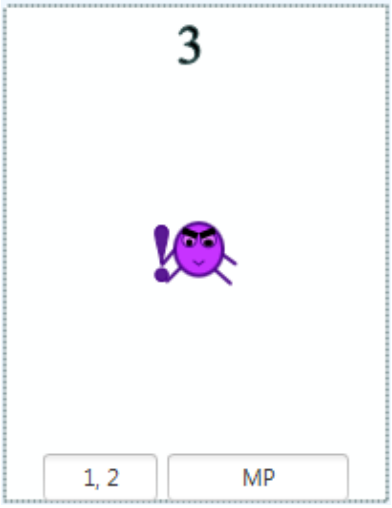
As you can see, anger and violence only beget more anger, and thus a potential for more violence (since the orange Character now has a Club . . .).

Example 2



Imagine that we were given the two Cards above, and want to use Modus Tollens on them. If we did so, what third Card would we produce?

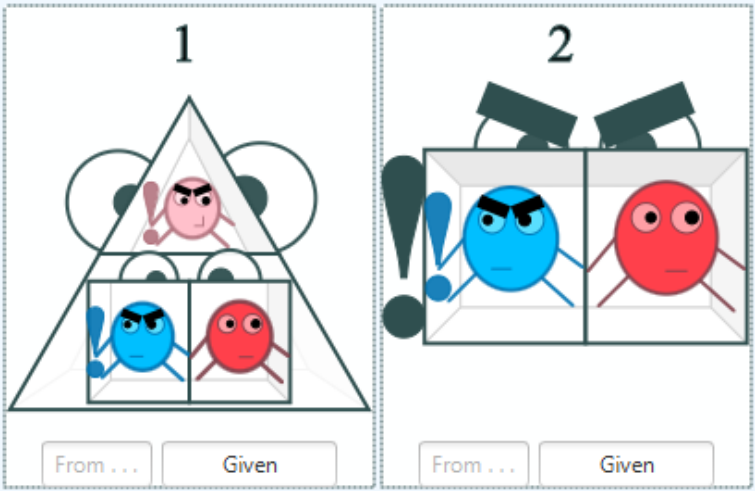
The answer is:



In this example, we have a match between the content of Card 1 and the content of the basement of the Pyramid in Card 2, *except* that the Pyramid in Card 1 has a Club, while the Pyramid in the basement in Card 2 has no Club. This example is essentially identical to the previous example, therefore, except that the thing with the Club outside the Pyramid, and the thing without a Club in the basement of the Pyramid, are themselves Pyramids, rather than simple Characters.

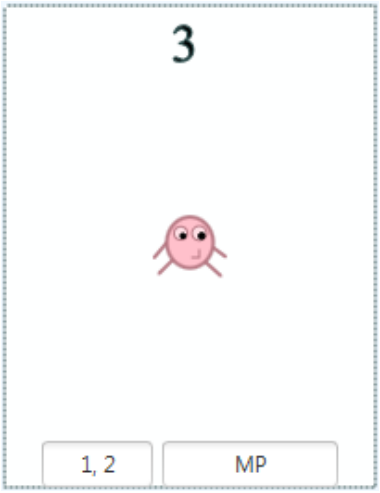
The principle, however, is the same. The green/yellow Pyramid with a Club in Card 1 attacks the green/yellow Pyramid in the basement of the Pyramid in Card 2, destroying the basement, and causing the orange Character to come tumbling out. The sudden fall, however, frightens the orange Character, so it ends up with a Club of its own.

Example 3



Even though these two Cards look like a mess, we can apply Modus Tollens to them. If we did, what third card would we produce?

The answer is this:

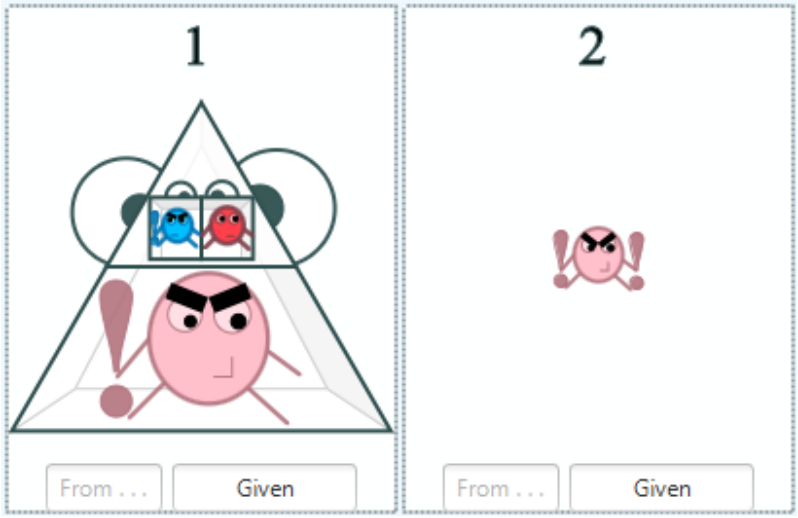


In this example, the pink Character in the attic of the Pyramid in Card 1 is not happy where it is. We can tell this by the Club it has. We can make it happy, however, if we return it to ground level. And we can do this if we can find something to destroy the basement of the Pyramid. It just so happens that the very thing we need is given to us in Card 2. In Card 2, we have exactly the same thing as is in the basement of the Pyramid in Card 1, except this new thing has exactly one more Club than the thing in the basement.

So, the Parcel in Card 2 attacks its rival in the basement of the Pyramid in Card 1, causing the pink Character to drop to ground level. This restores the pink Character to its original "home," and calms it. Thus, its Club disappears.

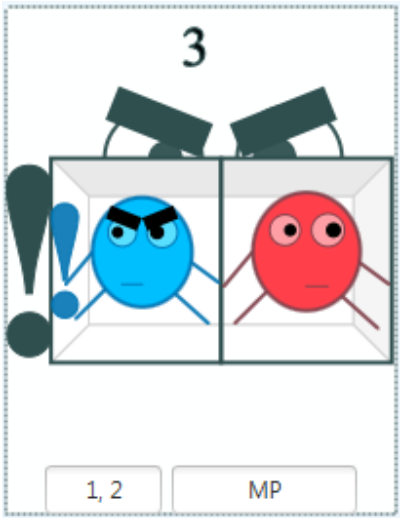
Notice that the chief difference between this example and the two previous is that in it the Pyramid comes first, and the thing that is supposed to attack the basement of the Pyramid comes second. This is to remind us that order doesn't matter. So long as we have a Pyramid by itself in some Card in the game, and in some other Card we have something that matches whatever is in the basement of the Pyramid (except that this other thing has one more Club), then we can use the Modus Ponens rule.

Example 4



Imagine that we were given the two Cards above. We can apply Modus Tollens to them. If we do, however, what would we produce?

The answer is this:



This example shows us two things. First, it doesn't matter whether whatever is in the attic of the Pyramid is a Character or a Chambergon. When it falls out of the Pyramid, it gains a Club if it doesn't already have one, or loses a Club if it already has one.

The second thing this example shows us is that it doesn't matter if the thing in the basement of the Pyramid has a Club, so long as the thing we want to attack the basement has *one more* Club. In this case, it's a pink Character with one Club in the basement, and a pink Character with two Clubs outside. Two Clubs beats one, so the pink Character with two Clubs destroys the basement, and the Package in the attic comes tumbling out.

4.2: The Power of Contraposition (A Recycling/Equivalence Rule)

The motto of this rule is, "You can swap the basement and attic of a Pyramid anywhere, but remember to deal with the Clubs." No matter whether a Pyramid is the main occupant of its Card, or has a Club, or is inside another Chambergon, you can swap the contents of its basement and attic. The only catch is that if either has Club, it will lose one Club in the transaction, while if either has no Club, it will gain a Club.

Remember the reason for this: If a Character has a Club, this means it is has been frightened into anger. It is not comfortable where it is, since that is not where it originally belonged. If it is at ground level (e.g., in the basement of a Pyramid), and has a Club, this means it fell there from the attic of some Pyramid. Restore it to the attic, and it will relax. Its Club will disappear. Likewise, if it is above ground level (i.e., in the attic of a Pyramid), and has a Club, this means it was suddenly lifted there from ground level. Restore it to the basement, and it will relax. Its Club will disappear.

The opposite, of course, occurs if the contents of the attic or basement have no Club. If either has no Club, then it is comfortable where it is. It is at home. Drop it suddenly into the basement, or raise it suddenly into the attic, and it will be traumatized. In shock, it will react with anger, gaining a Club.

"Why call this 'Contraposition'?" you wisely ask, in your wisdom and wiseness.

"Because you're taking the things inside the Pyramid, and putting them in positions contrary to where they used to be. One starts in the basement, and you put it in the attic. The other starts in the attic, and you put it in the basement."

"That's not really why it's called 'Contraposition,' is it? Isn't it because you're producing the 'contrapositive' of the original proposition, instead of the 'inverse' or 'converse' of the original?"

"If you're so smart, why don't *you* teach the course?" I respond.

"Maybe I will."

"Oh, you will, will you?"

"Just watch me."

"Fine."

"Fine!"

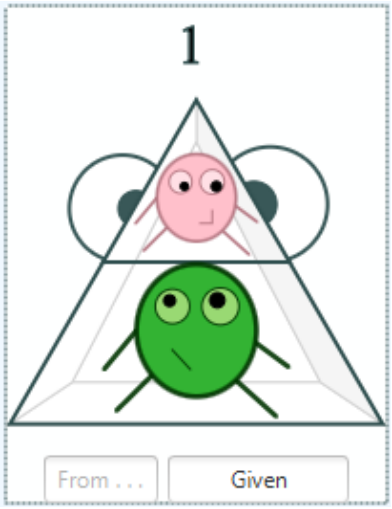
Restrictions: None. So long as you remember the way Clubs work when you are lifting or dropping things, you can swap the contents of a Pyramid's two chambers whenever and wherever you want.

The logical justification for this rule is essentially the same as the logical justification for Modus Tollens. If we know that the truth of one proposition is a necessary condition for the truth of another, and that the "necessary" one is false, we can conclude that the other is false too. For example, "If it is Saturday, then you can stay home," says that, "You can stay home," has to be true whenever, "It is Saturday," is true. So, let's imagine that "You can stay home" is not true today. If "You can stay home" is not true today, that means "It is Saturday" can't be true either. So, from "If A is true, then B is true" we can conclude, "If B is false, then A is false." That is, from "If A then B," we can conclude, "If not-B then not-A."

But this works even if we start off with "negations" instead of positive assertions. For example, "If it is not Saturday, then you can't stay home," says that "you can't stay home" must be true whenever "it is not Saturday" is true. (In other words, on every day but Saturday, you've got to go out and do something.) But that means that if "you can't stay home" were false, then "it is not Saturday" must be false too. In other words, if you can stay home, it must be Saturday.

So, if you start with "If A then B," you can change it to "If not-A then not-B." And if you start with "If not-A then not-B," you can change it to "If B then A." That, my friends, is Contraposition.

Example 1

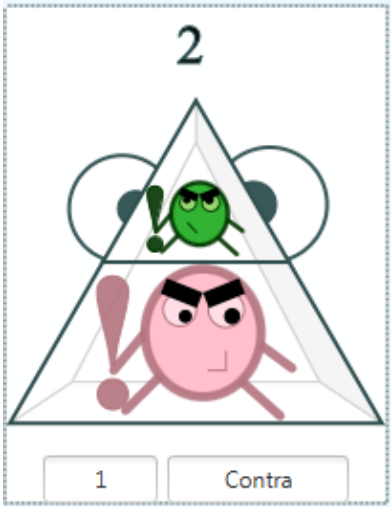


Imagine that this were the only Card we were given at the beginning of a Game. What could we do with it?

Well, we could apply Tautology to it, and that means we could double the Pyramid, or either of the Characters in the Pyramid. Or we could use Double Negation. Or we could use Implication. But today, we want to apply Contraposition and see what happens.

So, imagine that we use the Contraposition power on Card 1. What would be produced?

The answer is this:



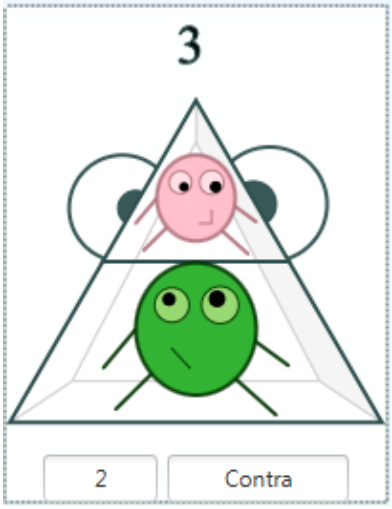
Contraposition flips the Pyramid's contents. The green Character moves to the attic and the pink Character to the basement. But neither is happy with its new position, and thus both end up with a Club.

So, imagine that we have done this, and thus have both Cards 1 and 2 available to us. What could we do next?

Well, we could apply Conjunction Introduction and put the two Pyramids into a Parcel. But let's just focus on the new Card for a moment. Can we apply Contraposition to it again?

The answer, of course, is, "Yes." You can apply Contraposition to any Pyramid anywhere. But what would be produced if we did?

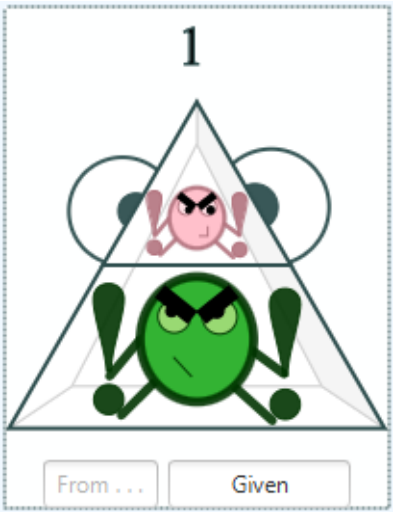
The answer is this:



Contraposition always moves whatever is in the basement into the attic, and whatever is in the attic into the basement. In this case, the characters are being moved from places where they are not happy (and thus have Clubs), to places where they are happy. Thus they lose their Clubs.

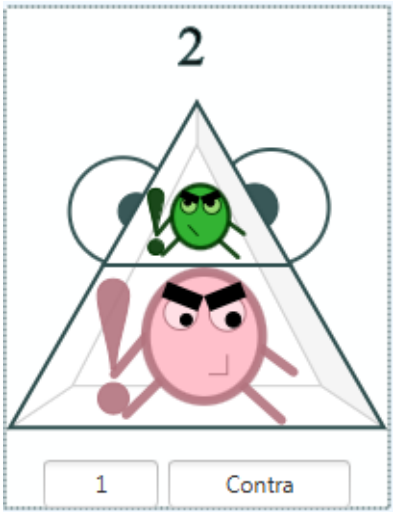
In this example, we start off with Characters that are comfortable where they are. If we swap them, therefore, both go into shock. This is what happens in Card 2. The only way to get them to calm down is to put them back where they came from, which is what we do in Card 3. If we apply Contraposition twice in a row to the same Pyramid, in other words, we end up back where we started. (That is why Contraposition is a recycling rule.)

Example 2



This time, we start off with Clubs. We are given a Pyramid containing Characters, both of which have two Clubs. What would happen if we apply Contraposition to this Card?

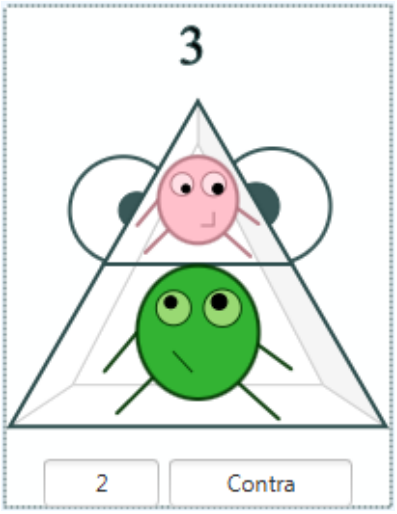
The answer is this:



Notice that the Character started off being absolutely miserable where they were. So, we move them to the opposition positions (from basement to attic, from attic to basement). This helps them calm down a bit, so each loses a Club. But since they started with two Clubs, they each have one left over.

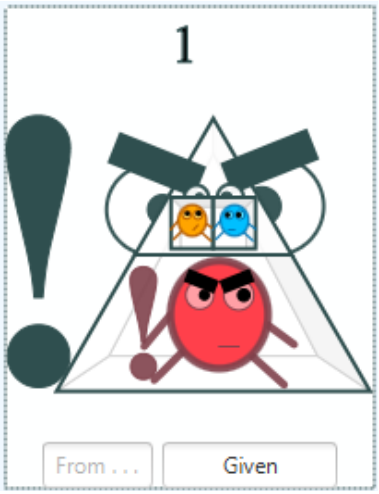
So, imagine that we apply Contraposition *again*. What would using Contraposition on Card 2 produce?

The answer is this:



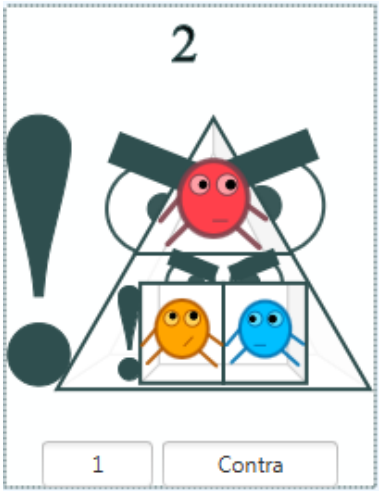
Notice that in this example, we applied Contraposition twice in a row, but didn't get back to where we started. Each application reduced the number of Clubs carried by the Characters by one. They went from two Clubs to one Club, and from one Club to zero. So, they started off with two Clubs and ended with none. *And yet*, we know that having two Clubs is equivalent to having none, since you can move back and forth between these states just by applying Double Negation. So we have, in an important sense, ended up back where we started.

Example 3



In this game, we start with the Card above. If we apply Contraposition to it, what would we produce?

The answer is this:



Notice that the red Character loses its Club, since it is back where it wanted to be. However, the Parcel has gained a Club because it was perfectly happy being where it was. Notice furthermore that the Pyramid itself hasn't lost its Club. This is because it has remained at ground level, where it is unhappy. Its insides have been rearranged, but it itself hasn't moved.

Now, imagine that we don't like the results of applying Contraposition to Card 1. What if we applied Contraposition to Card 2? What would we produce?

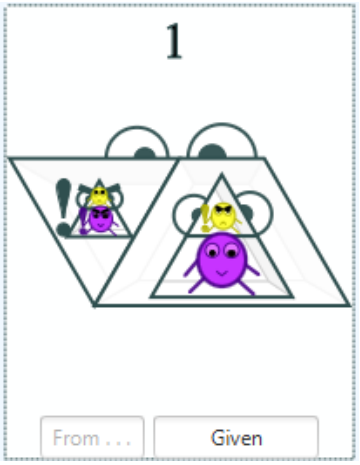
The answer is this:

The lesson of this example is twofold. First, it doesn't matter if the Pyramid in question has a Club. You can still "Contrapose" (or "transpose") it. Second, it doesn't matter if the contents of the chambers are Characters or Chambergons. You can still "Contrapose" them.

Note that in this example, there is no way to make both the occupant of the basement and the attic happy at the same time (at least, not while keeping the outer shape a Pyramid). Whoever is in the basement is going to be miserable. However, if we were to apply Implication, and then De Morgan, we could make both happy.

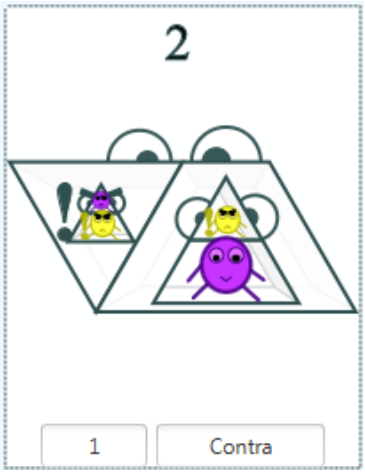
(If you can figure out what that last sentence means, then you are doing very well and feel good about your logical abilities.)

Example 4



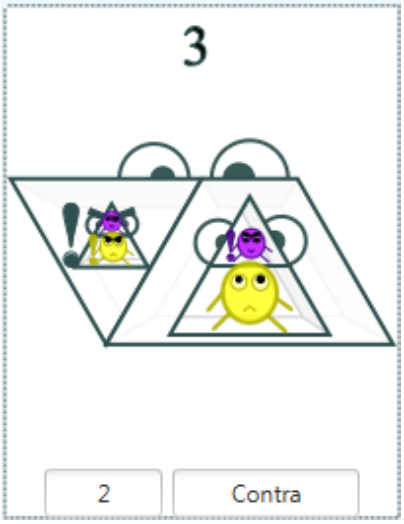
Imagine that we were given the Card above. What would happen if we applied Contraposition to it?

One answer is this:



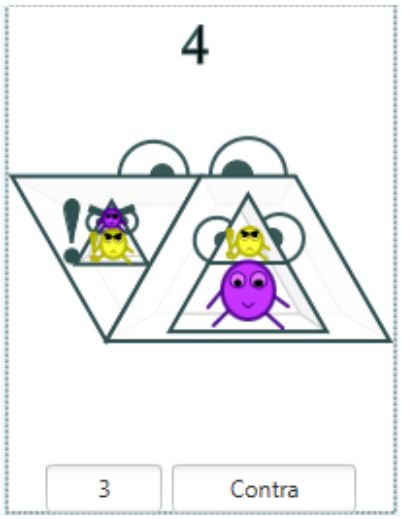
Notice that the Characters in the left Pyramid have swapped spots. The purple Character has lost its Club, while the yellow Character has gained a Club.

Now, using Contraposition, we could have swapped the contents of the other Pyramid instead. So, let's imagine that we apply Contraposition and do just that. If we do so, the following Card would result.

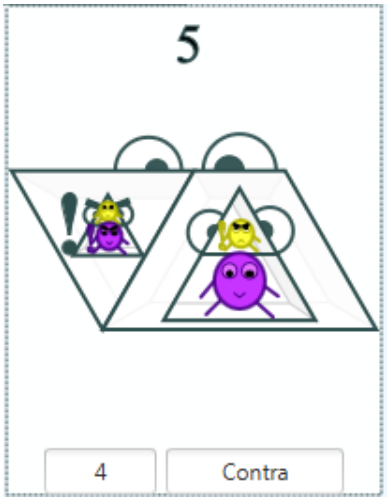


Card 3 contains the same contents as Card 1, but the contents of both Pyramids have now been contraposed. This gives the purple Character a Club, since it is no longer in the position where it was happy. But it removes the Club from the yellow Character, since it has been moved to where it is happy.

If we were to apply Contraposition again, this time applying it to Card 3, we could either contrapose the contents of the Pyramid on the right, or contrapose the contents of the Pyramid on the left. Let's start with the right.

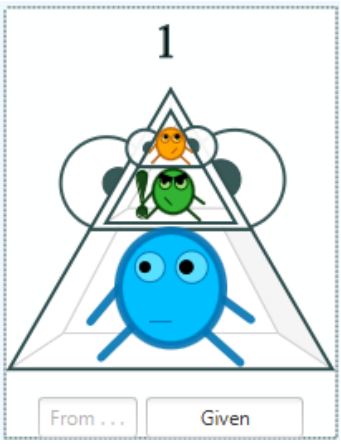


Notice that with Card 4, we have arrived back at where we were in Card 2. And we could go all the way back to Card 1, as it were, with only one more application of Contraposition.



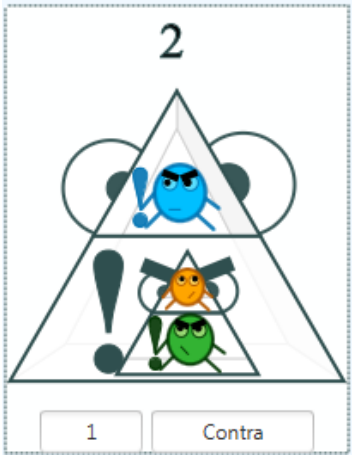
The lesson of this example is that even Pyramids inside other Chambergons can be "Contraposed." It doesn't even matter what kind of Chambergon the Pyramid is inside. In this case, it just happens to be a Paragon, and there just happen to be two Pyramids inside it.

Example 5



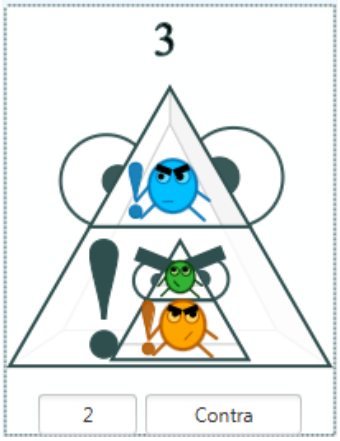
Imagine that we were given the Card above to start a game. What would happen if we applied Contraposition to it?

There are actually two answers to that question. One is this:



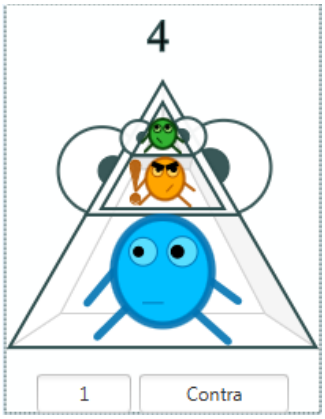
This is a kind of unfortunate move to make, however, since we started with a happy Pyramid in the attic, and a happy Character in the basement, and ended with both being miserable. Once we're in this position, however, we can ask what applying Contraposition would produce.

As before, there are two answers. One of them is this:



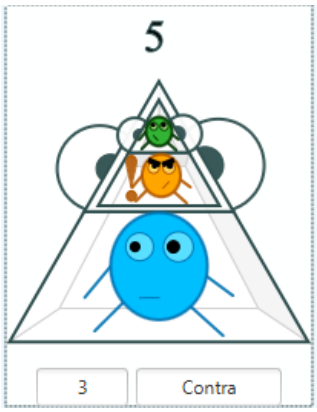
In moving from Card 2 to Card 3, we "contrapose" not the outside Pyramid, but the inside Pyramid. The orange and green Characters swap spots, and also moods.

Now, I mentioned above that when we apply Contraposition to Card 1, we have two different options regarding what we produce. Card 3 was one. Card 4 shows the other.



In this Card we have taken the contents of Card 1 and contraposed the Characters in the Pyramid in the attic. We contraposed the same Characters in Card 3, but in that Card they were in the basement. Furthermore, we could produce exactly the same result by applying Contraposition to Card 3, swapping the spots of the blue Character and the inner Pyramid.

Doing so would produce this:



Notice, therefore, how Cards 4 and 5 are identical, even though they were achieved by applying the same rule to different Cards.

4.3: The Power of Commutation (A Recycling/Equivalence Rule)

The motto for this rule is, "You can rearrange the contents of a Parcel or Paragon whenever, wherever." If you have two Characters or Chambergons in a Parcel or Paragon (or one Character and one Chambergon inside a Parcel or Paragon), with one in the left chamber, and one in the right, you can switch them. This rule works even if the Parcel or Paragon is inside another Chambergon, and even if the Parcel or Paragon has a Club. Simply interchanging the contents of the chambers isn't a big enough thing for anybody to notice, so you can get away with it.

<u>Restrictions:</u> None. This rule works for Parcels and Paragons that are in other Chambergons, and that have Clubs, in addition to working for Clubless Parcels and Paragons that are not in other Chambergons.

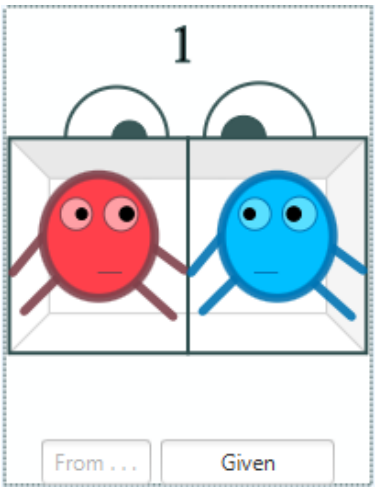
The logical justification for this rule is that it doesn't matter what order you say two propositions in if you have joined them with either a conjunction or disjunction. Saying, "I am over five feet tall and I have green hair" is the same as saying, "I have green hair and I am over five feet tall." Similarly, saying, "You are either an American or you are a foreign national" is the same as saying, "You are either a foreign national or an American."

Commutation in logic, therefore, is essentially identical to commutation in math. As we all know, "2 + 3" equals "5," but "3 + 2" also equals "5." Similarly, "4 × 5" equals "20," and "5 × 4" equals "20." You can rearrange the order of two numbers being added, or two numbers being multiplied, without changing the result.

However, you cannot change the order of two numbers being divided or subtracted without changing the result. Commutation in mathematics only applies to addition and multiplication, just like Commutation in logic only applies to conjunction and disjunction.

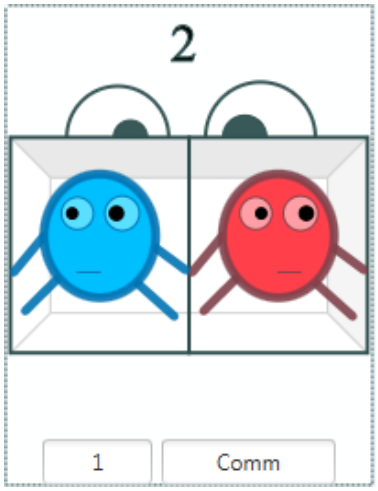
It is possible rearrange the contents of a conditional (i.e., of a Pyramid), of course, since that is Contraposition. However, while Contraposition requires us to change Clubs, Commutation does not. In Commutation, you just "swap chambers." Everyone remains at ground level, and thus no one is frightened into having a Club, or comforted into losing a Club. In Contraposition, however, when the chamber-swap happens, each thing is moving to a new position relative to ground level. Either it is moving into an attic or into a basement, and thus it either gains or loses a Club.

Example 1



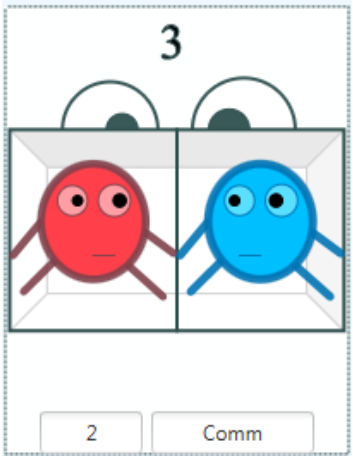
Here we have a simple Parcel containing two Characters. What would be the result of applying Commutation to this?

The answer is:



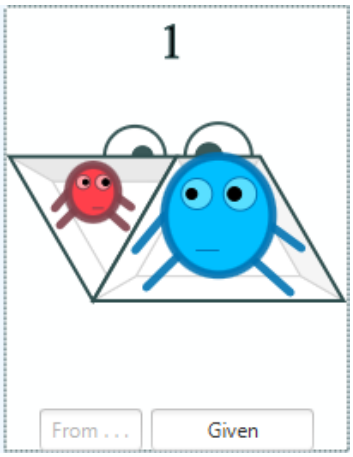
Notice that neither Character has left ground level, and since both were happy at ground level, both remain without Clubs. Having applied Commutation once, however, does not mean we cannot apply it again. If we used Commutation on Card 2, what third Card would we produce?

The answer is this:

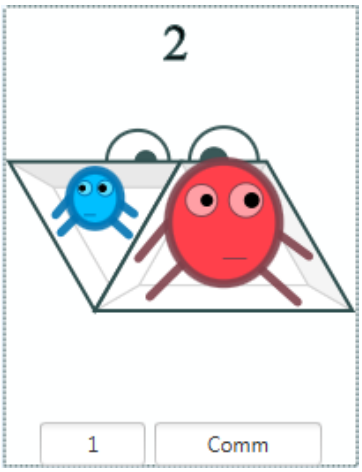


And that is identical to Card 1. Once again, no Clubs have to be added or removed, since Commutation changes chambers without changing chamber levels.

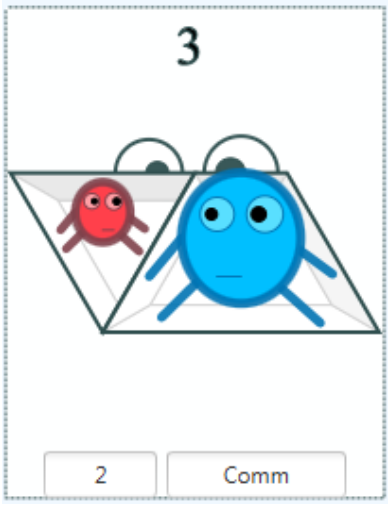
Now, imagine that we had begun with a Paragon instead of a Parcel.



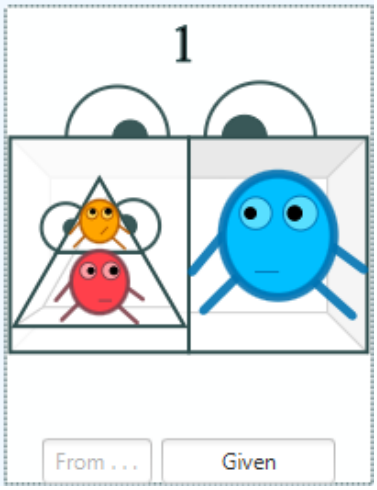
Here, in Card 1, we have two Characters at ground level in a Paragon. The red Character seems to be “higher” than the blue Character, in that it’s feet are further from the ground than the blue Characters. But what matters to the red Character is that the chamber in which it is standing is at ground level. If we were to apply Commutation to this Paragon, then, we would end up with this:



Neither Character has moved into the attic of a Pyramid, so neither needs to gain a Club. If we applied Commutation another time, furthermore, we would end up back where we started:

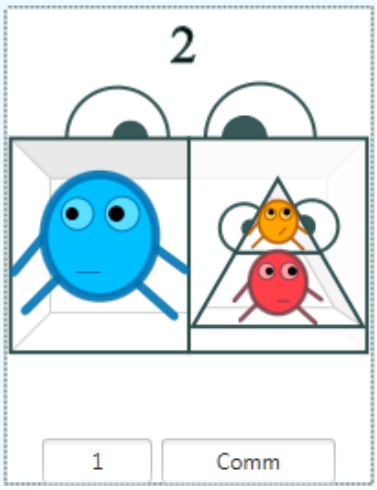


Example 2

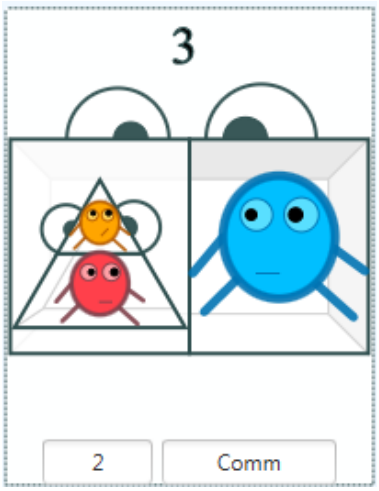


In this example there are two Chambergons. One is a Pyramid, however, and Commutation does not apply to Pyramids. So, if we use Commutation on the Card above, what second Card would be produced?

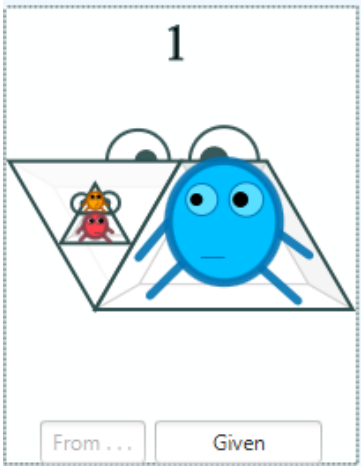
The answer is this:



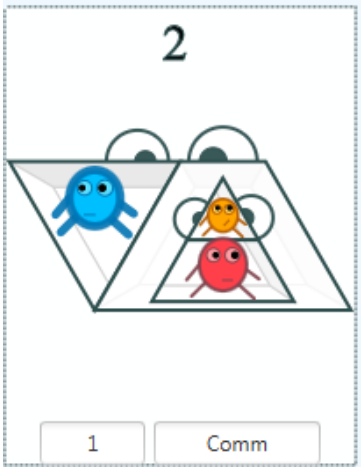
The blue Character has swapped spots with the Pyramid, but neither has left ground level, and thus both remain happy. If, however, we are unhappy with the new arrangement, all we have to do is apply Commutation again to produce this:



Note that the Commutation rule works on both Characters and Chambergons, so long as they are in Parcels. Note furthermore that much the same series of Cards could have been produced by Commutation even if the outermost piece had been a Paragon.



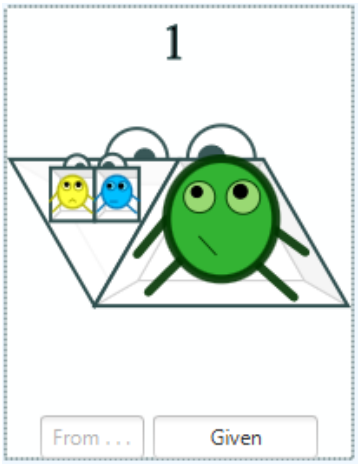
If we are given this Card to begin, we have a Paragon, and thus both the Pyramid and the Character are in chambers at ground level. We can then apply Commutation to produce this:



In this Card, both the Character and the Pyramid remain in chambers at ground level, but now each occupies the chamber formerly occupied by the other. They have “commuted,” but neither has gained a Club since neither has entered a chamber at a different level than they were in before.

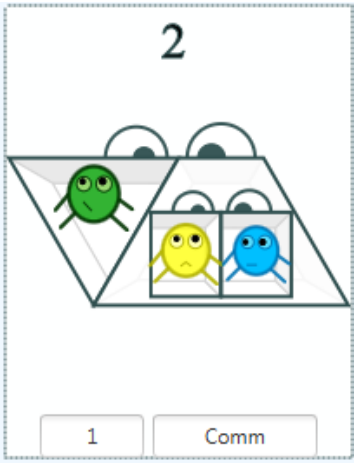
Then, just as before, we can apply Commutation to commute the contents back to where they began.

Example 3

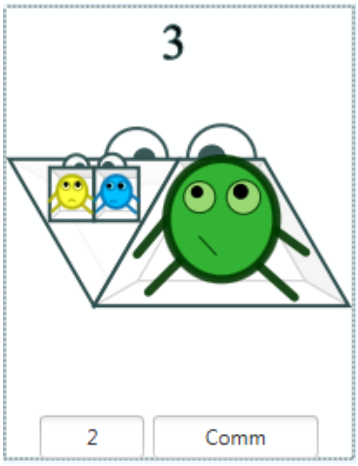


If we were given the Card above, and wanted to exercise our newly-acquired Commutation power, we would have two choices. We could either commute the outer Paragon, swapping the contents of its chambers, or commute the inner Parcel, swapping the contents of its chambers.

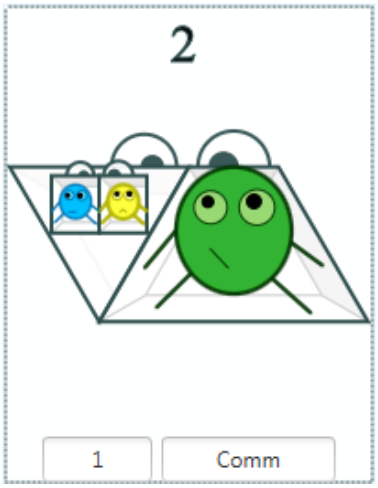
Let's commute the Paragon first. If we do so, we get this:



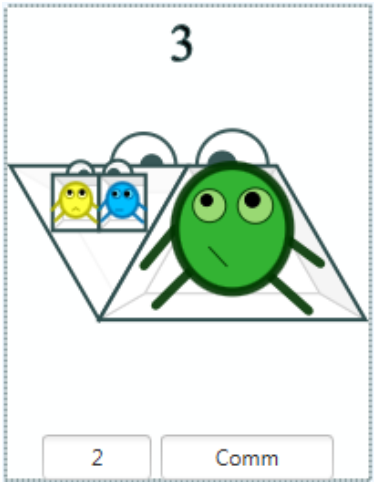
But, of course, Commutation works in the other direction too. So, we can put things back where they started if we wish, just by applying Commutation again.



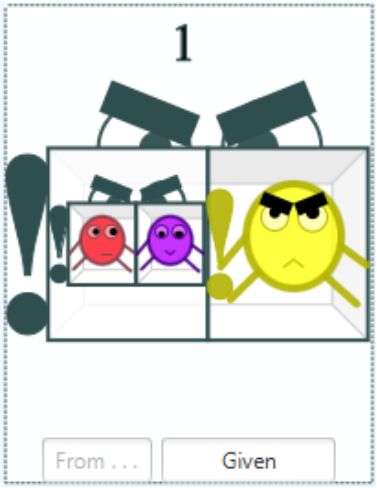
But now, imagine that we hadn't commuted the outer Paragon in Card 2, but had instead commuted the inner Parcel. In that case, Card 2 would have looked like this:



Then, we could have applied Commutation again to put things back the way they were when we started.



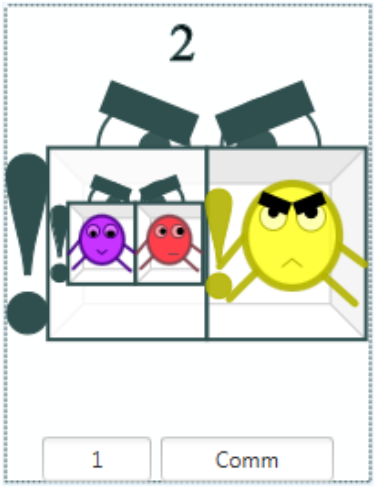
Example 4



Imagine that we were given the Card above. There are Clubs all over the place, but they do not get in the way of Commutation. Commutation does not remove something from, or insert something into a Chambergon. It simply rearranges the contents of the Chambergon.

If we apply Commutation to the above Card, this would allow us to rearrange either the outer Parcel or the inner Parcel. Let's rearrange the inner Parcel first.

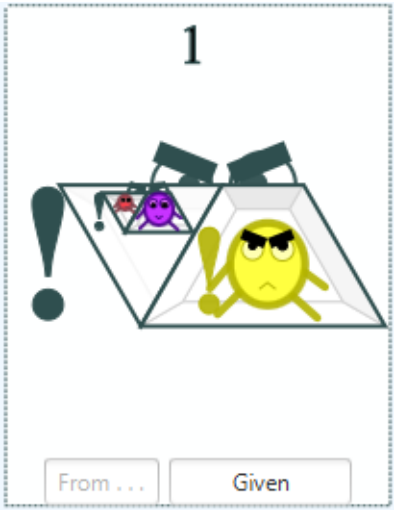
If we do this, we produce the following Card.



But now, let's rearrange the outer Parcel's contents, using Commutation. If we do that, we end up with:

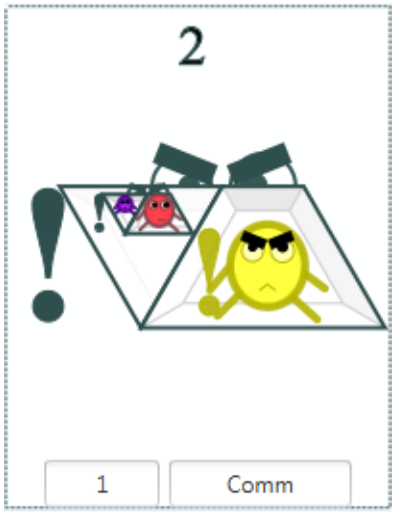
Notice that in moving from Cards 2 to 3, we have moved around two things (a Character and a Parcel) that have Clubs. However, neither loses its Club in the move, since neither has moved into a chamber at a different level than its previous Chamber.

Now, let's look at the same example, but using Paragons instead of Parcels.



Given this Card, we have many options. We could De Morgan the outer Paragon. We could De Morgan the inner Paragon. We could "Implicate" either of the Paragons. But we are going to commute the inner Paragon.

If we did so, we'd end up with this:



Notice, once again, that the Clubs outside the two Paragons do not get in the way of Commutation. We haven't tried to open either of them. We've just "shaken" them, as it were, to rearrange the contents of the inner one. But we could also use Commutation to rearrange the contents of the outer Paragon, as follows.

4.4: A Logician's Fourth Fundamental Move

We now have learned the fourth fundamental move that logicians make. Here are the first three:

1. The Outside of Shapes: Adding Clubs or removing Clubs.
2. The Inside of Shapes: Inserting things into or removing things from pieces.
3. The Shapes Themselves: Changing one shape into another.

The fourth move is this:

4. The Inside of Shapes: Rearranging the contents of shapes.

Or perhaps it would be better to put it as it is done in the chart on the following page.

A LOGICIAN'S FUNDAMENTAL MOVES

1. **The Pieces Themselves**
   a. *Changing one piece into another*
      i. Changing Parcels
         a. into Paragons: De Morgan's Theorem
      ii. Changing Paragons
         a. into Parcels: De Morgan's Theorem
         b. into Pyramids: Implication
      iii. Changing Pyramids
         a. into Paragons: Implication

2. **The Outside of Pieces**
   a. *Adding Clubs*
      i. Adding one Club: De Morgan, Implication, Contraposition
      ii. Adding two Clubs: Double Negation

   b. *Removing Clubs*
      i. Removing one Club: De Morgan, Implication, Contraposition
      ii. Removing two Clubs: Double Negation

3. **The Inside of Pieces**
   a. *Adding pieces into pieces:*
      i. Into a Parcel: Conjunction Intro
      ii. Into a Paragon: Disjunction Intro, Constructive Dilemma
      iii. Into a Pyramid: Hypothetical Syllogism

   b. *Removing pieces from pieces*
      i. Out of a Parcel: Conjunction Elim
      ii. Out of a Paragon: Disjunction Elim
      iii. Out of a Pyramid: Modus Ponens, Modus Tollens, Constructive Dilemma

   c. *Rearranging the contents of pieces*
      i. Of a Parcel: Commutation
      ii. Of a Paragon: Commutation
      iii. Of a Pyramid: Contraposition

4.5: Encoding Games

4.5.0: The Background Story

Characters represent simple propositions. We never encounter a proposition itself, however, here in the "Somatic Realm" (the physical world). Instead, we encounter things like sentences, which express propositions. The propositions expressed by sentences belong to what Plato would call "the World of Forms," but which we call the "Semantic Realm." That is where Characters live,

operate the “machinery” we call “Parcels,” “Paragons,” and “Pyramids,” and engage in battles using Clubs.

The logical conflicts that take place in the Semantic Realm are expressed by arguments, proofs, debates, and wars of words here in the Somatic Realm. If anyone were able to control the Semantic Realm directly, rather than working with Somatic signs and words, there is no telling what humans might be convinced to do. Our operatives who work in the Semantic Realm (yes, there is an “us,” and yes, we have found a way into the Semantic Realm) are working tirelessly to keep the Forces of Sophism at bay.

Our operatives must be able to put together the tools and weapons needed to counter the work of the Sophists from whatever they happen to have “lying around” in the Semantic Realm. In each situation, they find themselves surrounded by a different collection of what we call “Characters,” “Parcels,” “Paragons,” and so on, but rarely is what they have “to hand” enough win the current battle they are fighting. Instead, they must use what they have been “given” to construct what they actually need.

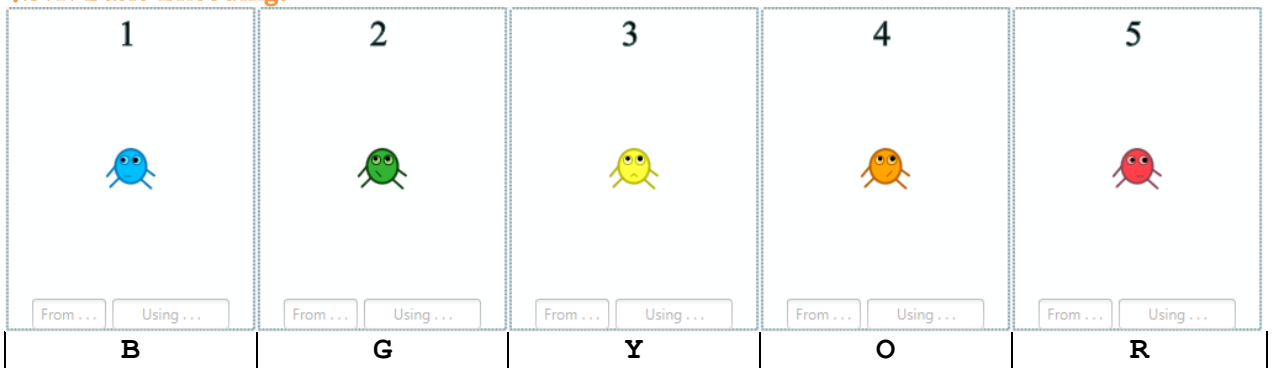
That is where your work comes in. In solving the puzzles you are given in this course, you are spelling out once and for all how to design, assemble, or extract a particular tool or weapon that our operatives might find useful, if they happen to be given a certain set of Characters and Chambergons. The more of these puzzles we can work out, the more prepared our operatives will be, and the more we can succeed in our struggle against Sophism.

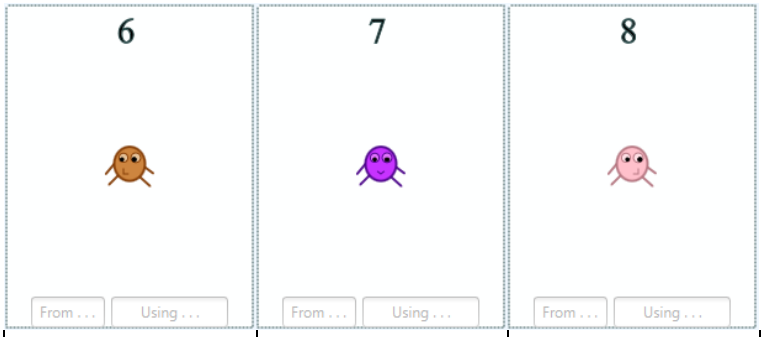
The solutions you find are archived for future use. When an operative in the Semantic Realm sends back a request for a situation we’ve already solved, we don’t want to waste time doing the solution work again. We want to just pull the solution out of storage, and send it along.

The problem is that our solutions are not saved in pictogram form. That would take up too much room on the archive’s servers. Instead, they are stored in textual code. That means that before we can send a solution to the archives for storage, we have to encode it into textual code, and before we can send a solution from the archives back to an operative, we have to decode it from textual code, back into pictograms.

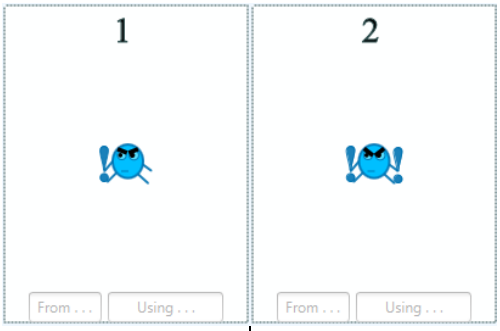
To do your job well, you must memorize the following encodings.

4.5.1: Basic Encodings

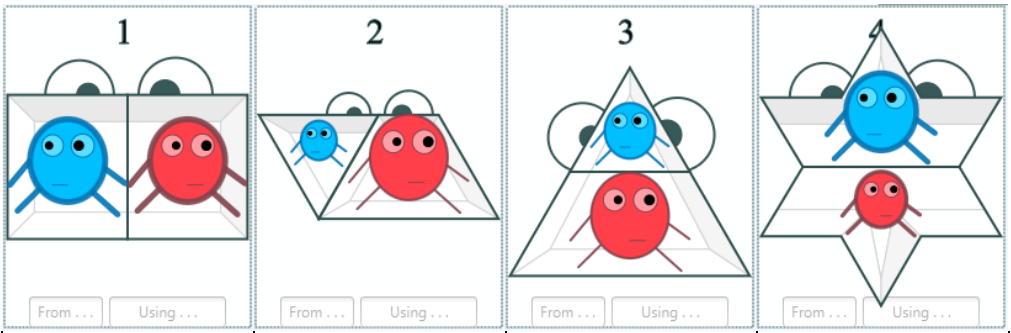




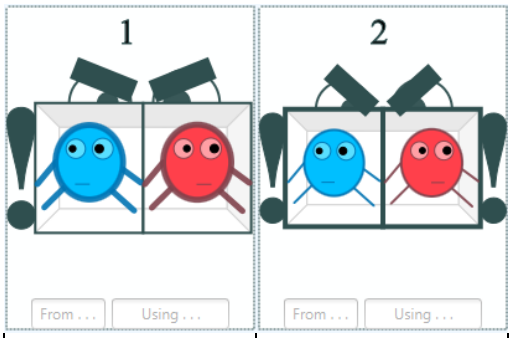
A	P	Z



!B	!!B



(B && R)	(B || R)	(B -> R)	(B <-> R)



!(B && R)	!!(B && R)

4.5.2: The Procedure for More Complex Encodings

Now, take a look at these two Cards.

How would we encode these? The answer is as follows.

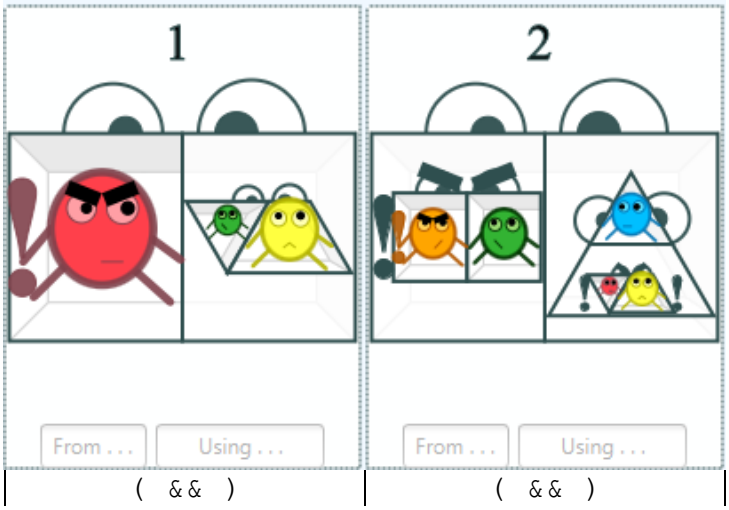
First: Encode the outermost piece in the Card.

If it is a Character, simply write its letter (and Clubs, if it has any.).

If it is a Chambergon, write its symbol between two parentheses. Leave a blank between the symbol and the parenthesis on either side, like this: "(||)."

In Card 1, the outermost piece is a Parcel. Parcels are encoded as, "(&&)."

In Card 2, the outermost piece is also a Parcel.

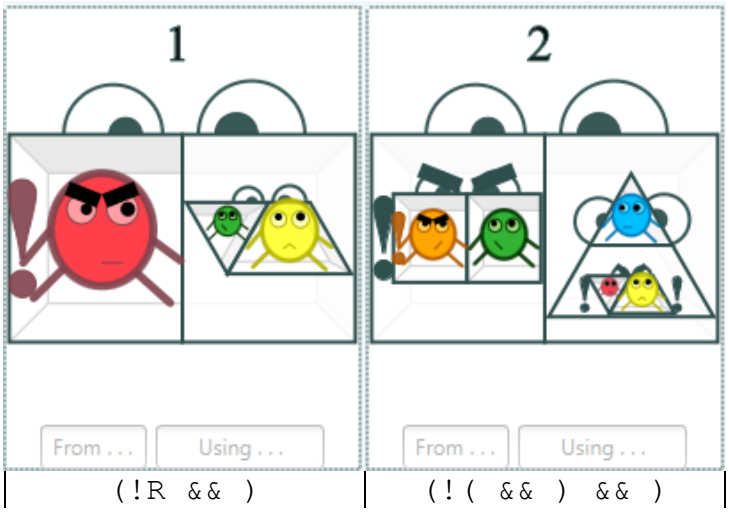


Second: If the outermost piece was a Chambergon, encode the outermost piece in the left chamber. Otherwise, you're done.

Write the symbol for whatever is the outermost piece in the left chamber on the left side of the symbol you've already written down. For example, if you first wrote "(||)," now write "(B ||)."

In Card 1, the outermost piece in the left chamber is a red Character with a Club. That is encoded as "!R."

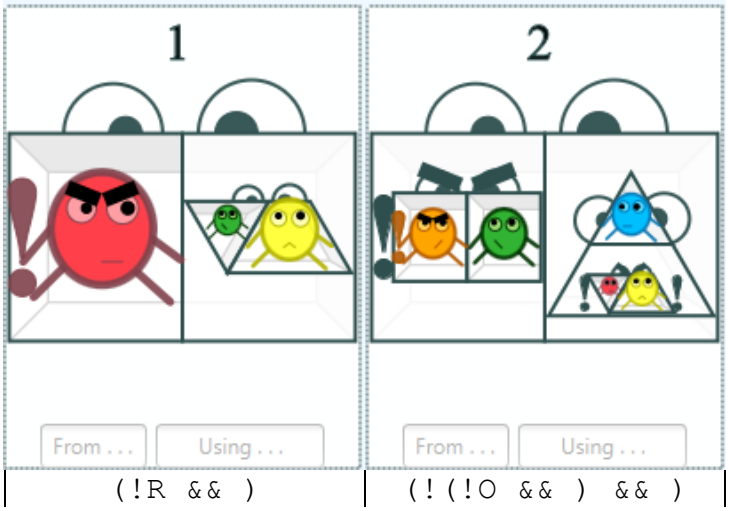
In Card 2, the outermost piece in the left chamber is another Parcel, with a Club. That is encoded as "(&&)."



Third: If the outermost piece in the left-hand chamber was a Character, you're done with the left-hand chamber. However, if it was another Chambergon, encode whatever is the outermost piece in *its* left chamber, first writing its symbol, and then filling in its chambers.

In Card 1, the left Chamber simply contained a red Character with a Club, so we're done encoding that chamber.

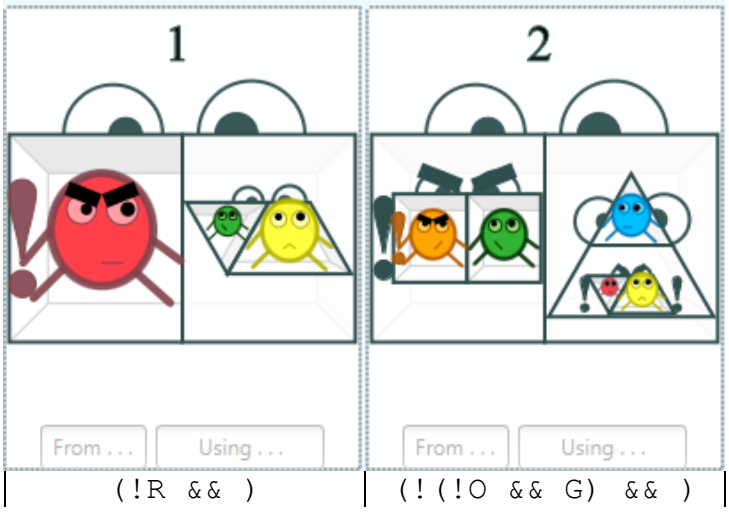
In Card 2, the left Chamber was a Parcel with a Club. So, now we have to encode *its* left chamber.



Fourth: After you've encoded the left-hand chambers "all the way down," fill in the right hand chambers, working your way "back up."

In Card 1, the left-hand chamber is already encoded all the way down.

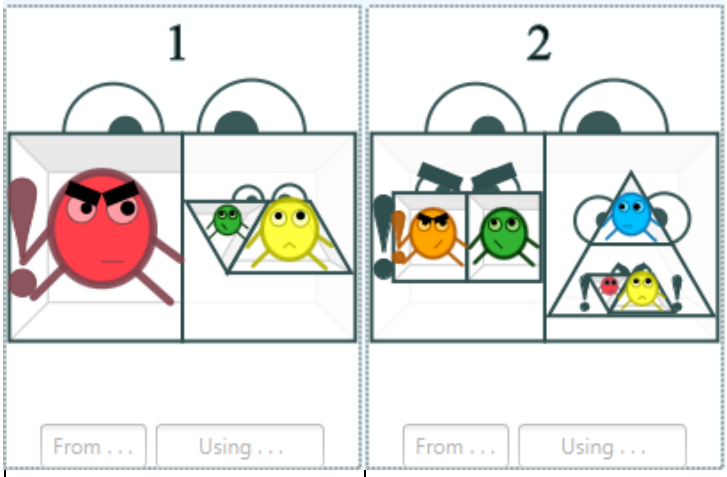
In Card 2, we've filled in the last left-hand chamber, and can now start working our way back up, filling in the right-hand chambers. The right-hand chamber of the inside Parcel contains a green Character.



We're now done with the left-hand chambers of the outermost Chambergons in both Cards. So, let's get started on the right-hand chambers.

In Card 1, the right-hand chamber's outermost piece is a Paragon, which is encoded "(||)." We'll put that on the right side of the "&&" under Card 1.

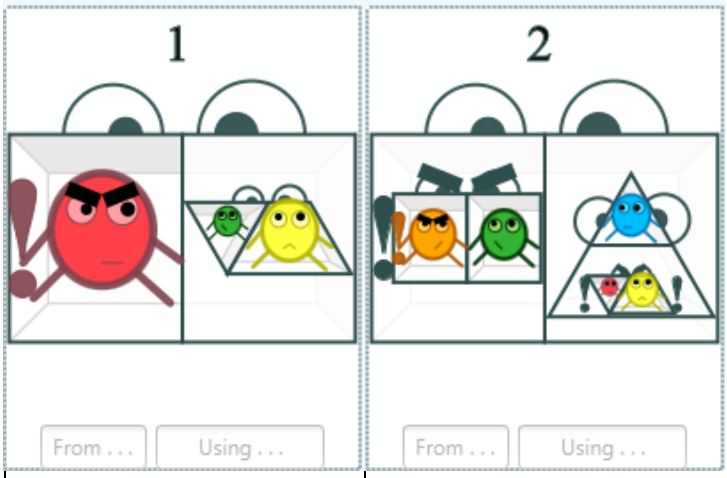
In Card 2, the right-hand chamber's outermost piece is a Pyramid, which is encoded "(->)." We'll put that on the right side of the original "&&" under Card 2.



| (!R && (||)) | (!(!O && G) && (->)) |

Although we're in the right-hand chamber of the outermost Chambergon in both Cards, whenever we start encoding a new Chambergon inside of the outermost Chambergon, we always start with its left-hand chamber (or its attic, if it is a Pyramid or Pulsar).

> In Card 1, the left-hand chamber of the Paragon in the right-hand chamber of the Parcel is a green Character. That is encoded as "G," and we will put it on the left side of the "||."
>
> In Card 2, the left-hand chamber is the attic of the Pyramid. It contains a blue Character, which is encoded as "B." We'll put it on the left side of the "->."

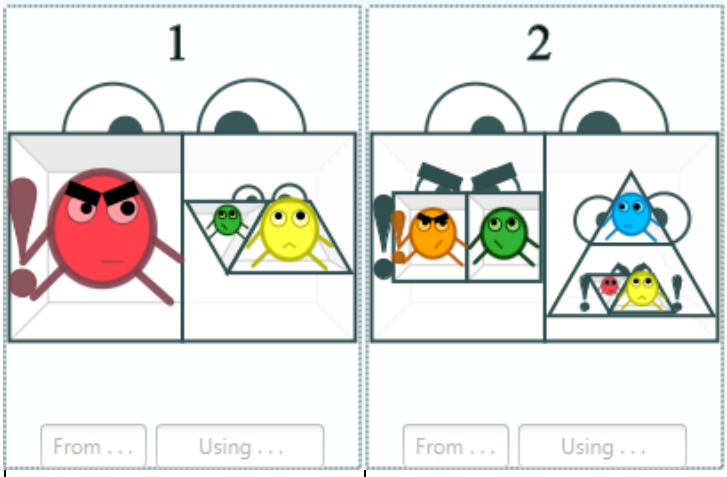


| (!R && (G ||)) | (!(!O && G) && (B ->)) |

Since the contents of the "left" chambers of the Chambergons in the right-hand chambers of the outermost Chambergons in both Cards were Characters, we can now move on to the right-hand chambers of the Chambergons in the right-hand chambers of the outermost Chambergons in the two Cards. (That was a horrible sentence.)

> In Card 1, the right-hand chamber of the Paragon is a yellow Character, which is encoded as "Y." We'll put that on the right side of the "||" symbol.

In Card 2, the right-hand chamber is actually the basement of the Pyramid. It contains a Paragon with two Clubs, which is encoded like this: “!!(||).” We’ll put that on the right side of the “->” symbol.

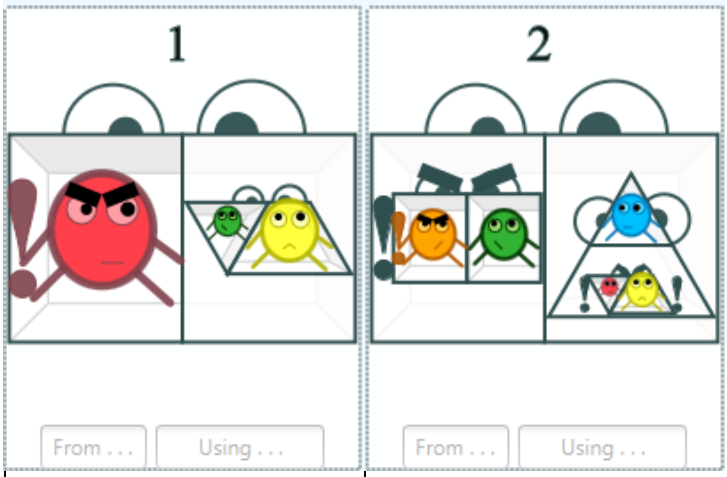


| (!R && (G || Y)) | (!(!O && G) && (B -> !!(||))) |

There are no more blank spaces left to fill for Card 1. Every Chambergon symbol in its encoding now has something in both its “left chamber” and its “right chamber.” Thus we are done with Card 1.

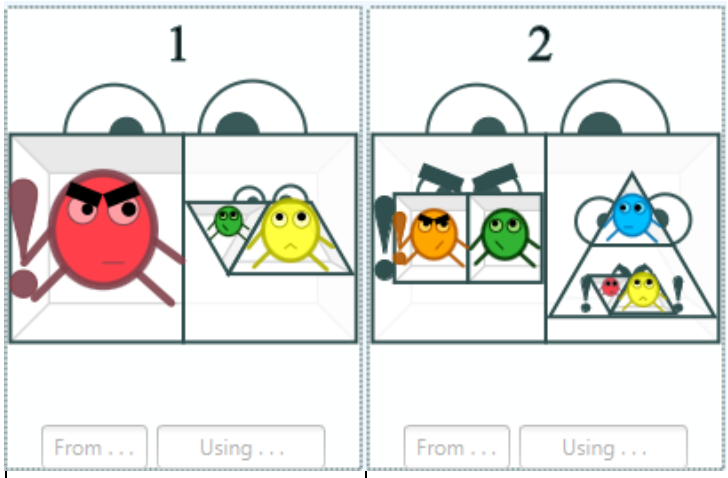
We still have more work to do with Card 2, however. Notice that the “||” has nothing to its left or right except parentheses. We still need to fill in those blanks.

So, we start with the left side first, and encode whatever is the outermost piece in the left-hand chamber of the Paragon inside the Pyramid inside the Parcel in Card 2. That piece happens to be a red Character, which is encoded like this: “R.” So, we put that to the left of the “||” symbol.



| (!R && (G || Y)) | (!(!O && G) && (B -> !!(R ||))) |

We've encoded our way all the way down in the left-hand chambers in the right-hand chamber of the outermost piece in Card 2 now, and can begin to work our way back up through the right-hand chambers. The yellow Character in the right-hand chamber of the Paragon is the only piece left to encode, however, so we simply put its "Y" to the left of the "||" symbol.



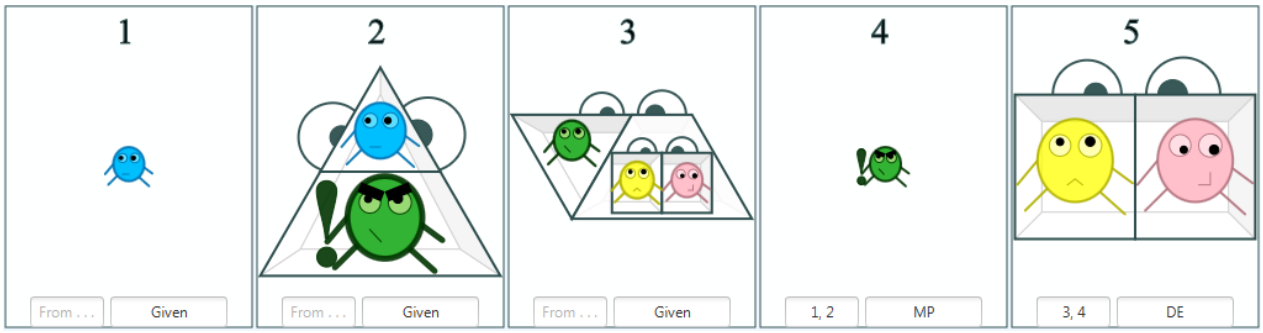
| `(!R && (G || Y))` | `(!(!O && G) && (B -> !!(R || Y)))` |

You can tell that the encoding for the Card 2 is now complete because each one of the Chambergon symbols in the code has something other than a parenthesis to both its left and right. The original "&&," of course, has a parenthesis to both its immediate left and immediate right, but those are the parentheses for the other "&&" on the left, and the "->" on the right. Thus, it actually has a "&&" to its left and a "->" to its right. And each of those symbols has the spaces to its left and right "filled in" by other symbols (i.e., by symbols other than mere parentheses).

If you've followed the encodings we just did for the Cards above, you now know how to encode any Card in Chambergon Battle Logic. Start with the outermost piece and work your way in. For each Chambergon you encounter, put its symbol between two parentheses. Then, fill in what is to the right and left of the symbol by encoding the outermost piece in its left (or top) chamber, and its right (or bottom) chamber. And make sure not to forge the Clubs as you're going along.

4.5.3: Encoding Entire Games

A normal puzzle-solving game in Chambergon Battle Logic is laid out in rows from left to right. To fully encode such a game, we simply write the code for each Card on its back. However, if we were to write down only the codes for the Cards, we would lay things out vertically. See the following example.




```
1. B                  Premise
2. (B -> !G)          Premise
3. (G || (Y && Z))    Premise
4. !G                 1, 2 MP
5. (Y && Z)           3, 4 DE
```

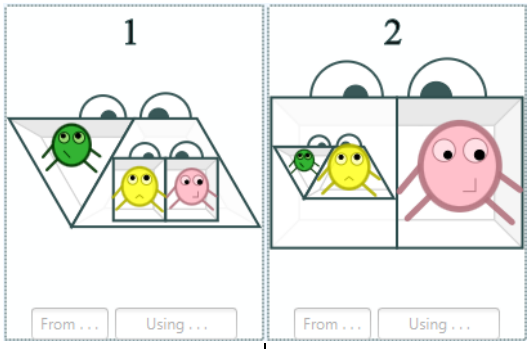
When a Game gets encoded, its Cards become lines, as you can see above. Card 1 becomes Line 1, and since Card 1 contains a blue Character, Line 1 contains a "B." Card 2 similarly becomes Line 2, and since Card 2 contains a Pyramid with a blue Character in its attic and a green Character with a Club in its basement, Line 2 contains a "B" and a "G" (with the "G" preceded by an exclamation mark) separated by the "->" code for Pyramids. (In a different coding system, there would be a "–", "~", or "¬", instead of the "!", and an "→" or "⊃" instead of the "->".)

Card 3 then becomes Line 3, which is kind of complicated. Card 3 contains two Chambergons, one of which is inside of the other. The Paragon is encoded by the double-pipe ("||"), while the Parcel is encoded by the double-ampersand ("&&"). To encode the fact that the yellow and pink belong together in the Parcel, we put parentheses around them.

Notice that instead of writing the Card number above the code for each Card, we write it to the right of the code, thus making it a line number. Notice furthermore that instead of writing the justification for each Card underneath the code for each Card, we write the justifications to the right. And notice finally that we write "Premise" instead of "Given." (Many systems just write "p" instead of using the whole word.)

4.5.4: More Coding Details

The placement of the parentheses in Line 3 is actually hugely important. Take a look at the following.

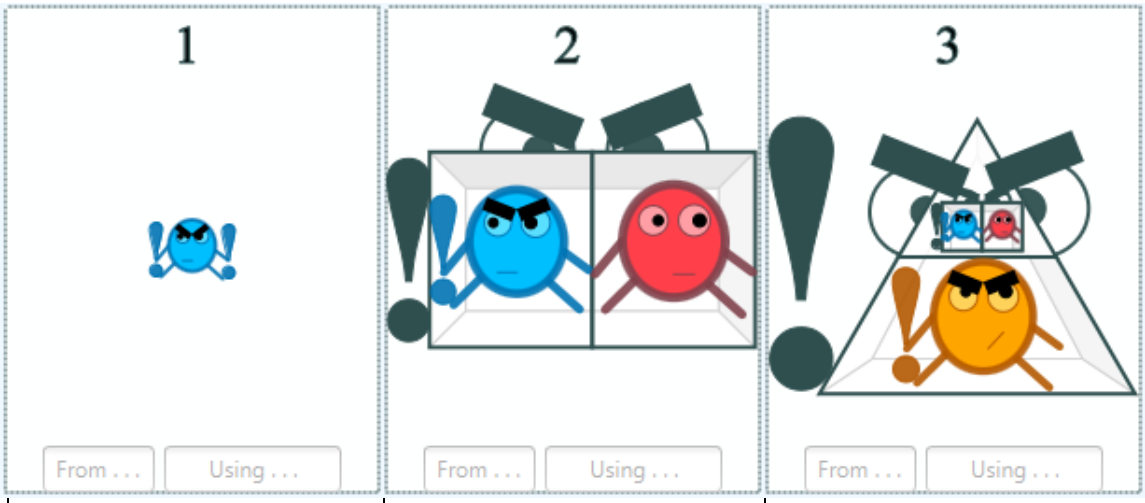


| (G || (Y && Z)) | ((G || Y) && Z) |

Notice that both Cards are encoded with the same letters ("G," "Y," and "Z"), the same Chambergon symbols ("||," and "&&"), and the same number of parentheses. The only difference is where the parentheses are. It is very difficult to tell the two Cards apart once they have been encoded, but when you see the icons, it becomes obvious that they are different. (This is one advantage of icons/pictograms over code.)

Notice that in code for both Cards, the Chambergon symbol that is the least buried inside parentheses corresponds to the Chambergon that is outermost in the Card. In Card 1, for example, the "||" is inside one pair of parentheses, while the "&&" is inside two pairs of parentheses. Likewise, in Card 1, the Paragon ("||") is the outermost Chambergon, while the Parcel ("&&") is inside it.

Where you put the parentheses in the code also matters when it comes to Clubs.



| !!B | !(!B && R) | !(!(!B && R) -> !O) |

Card 2 has two Clubs. One belongs to the blue Character, and one belongs to the Parcel. We encode this by putting one "!" outside the parentheses and the other inside. The "!" next to the parenthesis ("to the left") belongs to the "&&" inside the parentheses, while the "!" next to the blue belongs to the blue itself.

Now, look at the code for Card 3. If you can figure out why each of the exclamation marks is where it is, you are good to go.

Chapter 5: Level 5 Powers

5.0: Detour Rules

In this chapter we'll be dealing with three rules: Assumption, Conditional Proof, and Indirect Proof. Without the Assumption ("A") rule, Conditional Proof ("CP") wouldn't work at all. And without Assumption and Conditional Proof, Indirect Proof ("IP") wouldn't work in its normal application.

I call these rules "detour" rules because they're going to feel like going out of your way to reach your goal. In fact, when you use them, it's going to feel like you are giving up on your goal and heading in completely the wrong direction *on purpose*. However, these "wrong directions" are just detours, and there's a rather precise sequence of things you will do each time. Once you master that sequence, you'll feel a lot better about these rules, and they'll become something like second nature.

So, bear with me as I try to explain the Level 5 Powers. If you stick with it, they'll start to make sense.

5.1: The Power of Assumption (A Landfill/Inference Rule?)

This rule allows you to introduce a new Card and put anything into it. The problem is that in labeling the new Card an "Assumption," you are taking a major detour in your path toward a solution. You're not actually proving anything; you're just saying, "Now, let's assume that this Card is legit. I've got no evidence that it's legit, however, so it's just an assumption."

The way to think about Assumption Cards is to see them as pretend or make-believe "Givens." Every Card you justify using the "Assumption" rule is, in some sense, equivalent to adding another Card justified by the "Given" rule. The problem is that you *weren't* given the Cards that *you* introduce using "Assumption." After all, you invented them, rather than being given them. And since your job is to reach the goal based on what you were given—and nothing else—you can't actually claim to have solved the puzzle, or done your job, until you have effectively eliminated the new Card you created.

Every assumption you make in the course of a game, therefore, must be "discharged." When you introduce an Assumption Card, it's like opening a problem ticket with the IT department. Until the ticket is "closed" your problem isn't solved. If you haven't discharged every Assumption Card by the time you reach the Goal, therefore, you haven't actually reached the Goal. You've reached a mirage of the Goal.

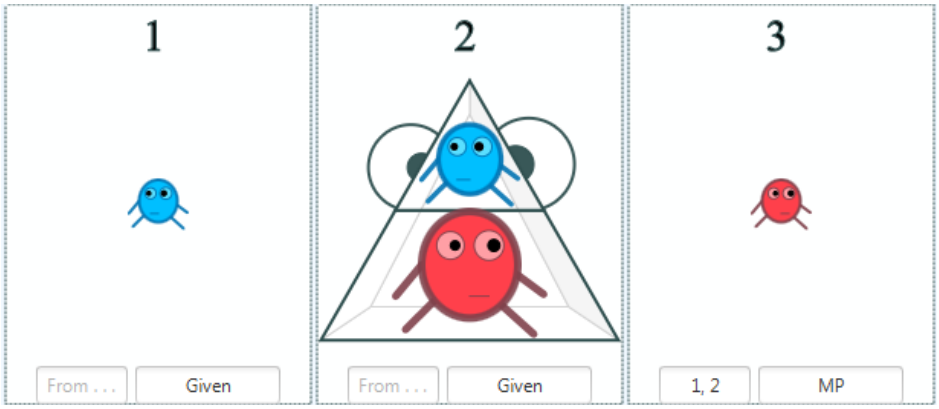
We'll learn about discharging assumptions when we discuss Conditional Proof, below. For the moment, however, let's just get used to the idea of Assumption Cards.

<u>Restrictions</u>: You can introduce an "Assumption" Card at any point in the game. However, until each assumption is "discharged" using its own application of Conditional Proof, you can't claim to have completed a game.

The logical justification for the Assumption rule is that it's always legitimate to make an assumption, just so long as you don't base your conclusions on your assumptions. If you were allowed to base your conclusion on an assumption you introduced midway through your argument—rather than basing your conclusion on the common ground you and the person you're arguing with have already agreed on—you could just "assume the conclusion," and never have to prove anything.

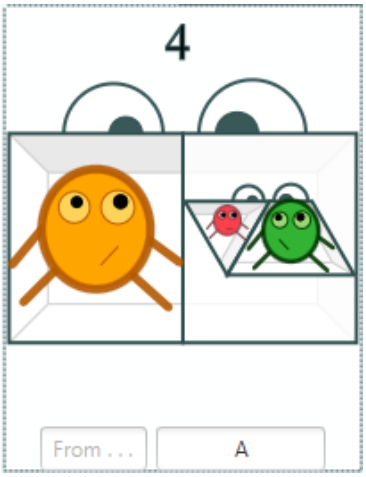
For example, you might say, "We both agree that democracy is the best form of government, and we both agree that politicians are usually untrustworthy. Now, I want to prove to you that if you believe those two things, then you should vote for Georgina Smithson-Smith in the next election. Here's my argument. First, democracy is the best form of government. Second, politicians are usually untrustworthy. Third, let's just assume that you should vote for Georgina Smithson-Smith. Therefore, fourth: you should vote for Georgina Smithson-Smith." And that's just not a fair or legitimate way to prove things to people.

Example 1



Let's imagine that we were given the first two Cards and have already derived the third Card using Modus Ponens. At this point in the game, what could we introduce using the Assumption rule?

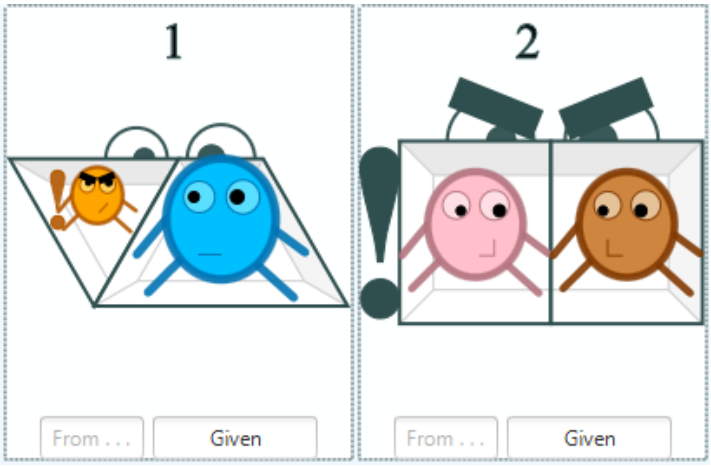
One of the infinite possible answers is this:



I say this is “one of the infinite possible answers” because we could have put any combination of Characters, Clubs, and Chambergons into Card 4. All that is required is that we put “A” or “Assumption” in the “Rule” box for the Card (and leave the “From” box blank).

Once we have introduced Card 4, however, we have placed ourselves in a kind of debt. Until we pay back that debt using Conditional Proof (which we will discuss in the next sections), we will be unable to finish the game.

Example 2



Imagine that we open a game and find the two Cards above to be our givens. What third Card could we introduce using Assumption?

There are an infinity of answers to that question, and one of them is this:

Notice that, just like in the Given Cards, Assumption Cards do not come "From" any previous Cards. Thus, the "From" box remains blank, while the "Rule" box is filled in.

Now, you might ask why we would want to introduce a blue Character with a Club in this situation. The answer may be that we want to do DE on Card 1. Or the answer may be that we want to do something completely different. It all ultimately depends on what goal we've been set.

5.3: The Power of Conditional Proof (A Landfill/Inference Rule)

The motto for this rule is: "You can borrow whatever you want, so long as eventually make it the attic of a Pyramid." It's as if there were a bank in the Semantic Realm that loves Pyramids so much its willing to give away custom attics for free. All you have to do is come up with the basement.

Since Conditional Proof allows you to create a Pyramid (by first pulling its attic out of thin air), you should think about using Conditional Proof whenever your Goal is a Pyramid. If you're trying to construct a Pyramid, and the bank is willing to give you the attic of that Pyramid for free, after all, why not give it a shot?

So, here's a bit of strategic advice: Whenever you see that your Goal is a Pyramid, try creating a new Card and putting whatever is in the attic of the Pyramid you're trying to construct into the new Card. All you have to do to justify this new Card is to type an "A" in its "Using" box.
"But why 'A'?," you ask. "A" stands for "Assumed," or "Assumption." But we can think of it as standing for "Attic." Notice that the letter "A" looks like a Pyramid that's missing its floor or foundation. It's kind of like an attic hanging in midair, without a basement. By typing "A" as the justification for your "out of thin air" Card, therefore, you are reminding yourself that you are going to have to eventually make whatever is in this Card into the attic of a Pyramid.

This means that after you've created the Card that has the attic of the Pyramid that is your Goal, you should try to use this new Card to create whatever is in the basement of your Goal Pyramid. Once you've created whatever is in the basement of your Goal Pyramid, your next Card should contain your Goal Pyramid itself, and should be justified by listing (1) the number of the Card you justified with "A" (the Card that contains the attic of the Pyramid you just created) and (2)

the number of the Card that contains the basement of the Pyramid you just created. Then, in the "Using" box, you should write "Conditional Proof," or "CP."

An interesting point about this rule is that once you've created your Goal Pyramid, you can no longer use any of the Cards between it and the original "A" Card. You can't even use the original "A" Card. All you can use is the "CP" Card. Everything else, as it were, was just a loan. It never really belonged to you, and so once the Pyramid construction project is complete, you have to give it back.

Think of it this way: the "A" Card opens a mini-game, and the "CP" Card closes it. Once that mini-game is over and closed, there's no going back into it. All you have available to you outside the mini-game (that is, all you have available to you later in the actual game) is the *results* of that mini-game (i.e., the Pyramid you built in it). Just like you can't take the air rifle, hammer, or throwing rings away from the games you play at a carnival or fair—but you *can* take your winnings—you cannot use the Cards that are inside a mini-game outside that mini-game. You can only use the Pyramid you won by playing it.

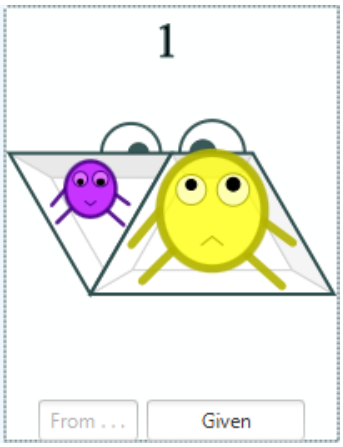
Now, this does not mean that you cannot use Cards from outside the mini-game while you're inside it. Cards between the "A" Card and the "CP" Card can appeal to Cards that came *before* the "A" Card, in other words. Just like you bring your money, strength, and skills into the games you play at carnivals, you can bring the Cards that were already there into the "mini-games" you initiate by laying down "A" Cards.

This is all really abstract at the moment, I know, because I haven't given you any examples. However, I need to make one final point. Since you can take out a loaner "A" Card whenever you want, you can take out two such Cards in a row. You can take out three in a row. You can take out as many as you want, in fact. You can even space them out. However, note the following two restrictions.

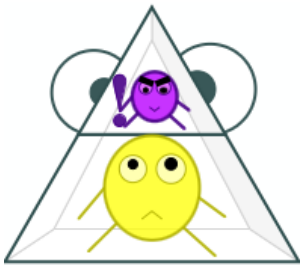
<u>Restrictions</u>: (a) If you take out a new loan before you've paid back the previous loan (by turning it into a Pyramid), you have to repay the *new* loan *before* you can pay back the old loan. Furthermore, (b) until each of your loans has been paid back (by turning it into a Pyramid), you cannot complete the Game. You cannot win if you are still in debt.

The logical justification for this rule is that all it allows you to do is show that if one thing were true, another would be true too. In other words, if the thing you're assuming turned out to be correct, then some other thing would have to be correct as well. That's what Conditional Proof allows you to "conclude." However, it doesn't allow you to conclude that your assumption actually is correct. It forces you to keep things hypothetical. Your assumption remains "hanging in the air" as the attic of the Pyramid you produce, rather than resting solidly on the ground.

Example 1



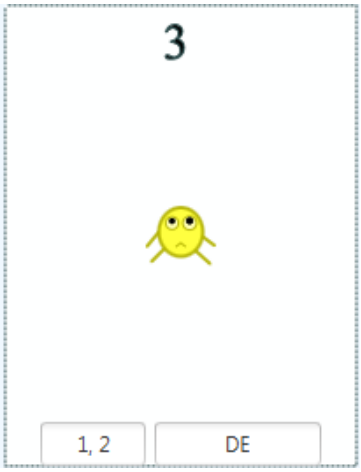
Imagine that we were given the Card above. And imagine that our Goal were the following:



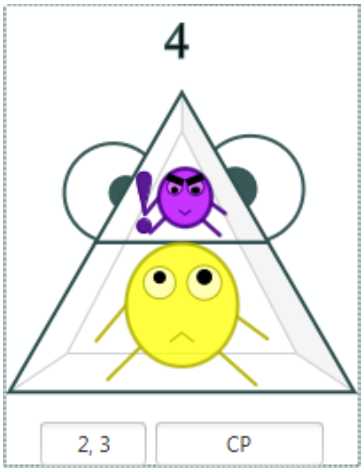
Our goal is a Pyramid. We know we can build Pyramids by taking out a loan of whatever is in the attic, and then creating whatever is in the basement. So, we do just that.

First, we take out a loan Card that has a purple Character with a Club in it.

Now we have the attic of our future Pyramid, and all we have to do is create the basement. So, we use the purple Character with a Club on the Paragon in Card 1. It destroys the purple Character in Card 1, freeing the yellow Character.



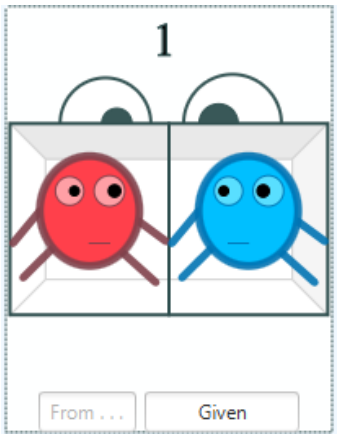
So, in Card 2 we have the attic of the Pyramid we want, and in Card 3 we have the basement of the Pyramid we want. All we have to do is put them together into an actual Pyramid, using Conditional Proof.



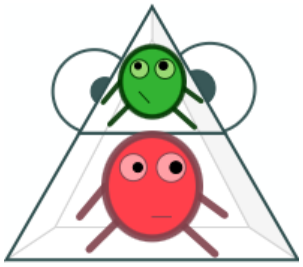
Our debut is now repaid. The assumption we made in Card 2 is now discharged. Any other Cards that might come after 4 could use the Pyramid in Card 4, but not the purple Character with the Club in Card 2 or the yellow Character in Card 3. Those are “locked away” within the mini-game that occurs between Cards 2 and 4.

It’s worth noting that in this case, our “shortcut” from Card 1 to the Pyramid we wanted was actually a “longcut.” We could’ve gotten there in just one Card, using Implication. The point of this example, then, was just to show you how Conditional Proof works. If you need to produce a Pyramid, you can do so using Conditional Proof, even if you could also do it using some other rule.

Example 2



Imagine that we were given the Card above, and asked to produce the following Pyramid.



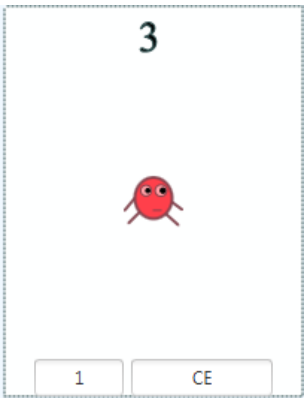
Seeing that we need to build a Pyramid, we decide to take out a loan. But what loan should we take out?

The answer is that we should make a new Card with whatever is in the attic of the Pyramid we want to build. Since the Pyramid we want to build only has a green Character in its attic, therefore, we only need to take out a single green Character as our loan.

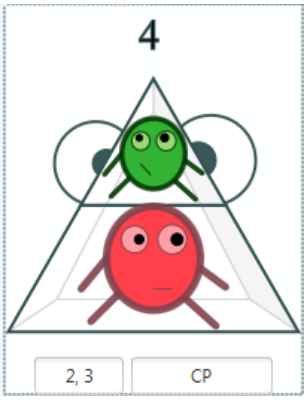
So, Card 2 ends up looking like this:

Now that we have the attic of our Pyramid, we need to make its basement. But the content of the basement of the Pyramid we are trying to build is just a single red Character. We can get that by applying CE to Card 1.

So, Card 3 ends up looking like this:

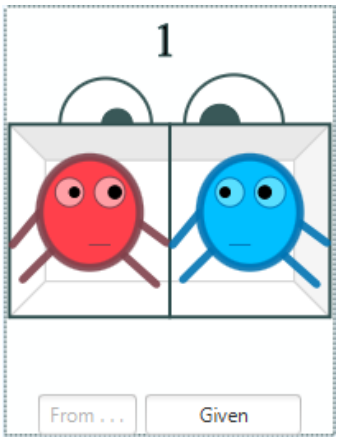


And this means we have created both the attic and the basement of our Pyramid. All we have to do is put them together using CP in Card 4.

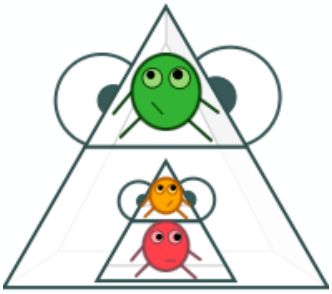


Notice that the two Cards that Card 4 cites are “2” and “3.” Card 2 is the source of the Pyramid’s attic, and Card 3 is the source of the Pyramid’s basement. Any other Cards that might come after 4 could use the Pyramid in Card 4, but not the green Character in Card 2 or the red Character in Card 3. If later Cards want to use the red Character, they will have to use CE on Card 1 again, to extract it.

Example 3

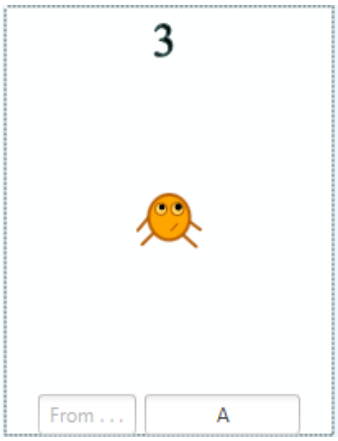


Let's start in the same place we did last time. Take the same given Card, but this time let's produce the more complex Pyramid below.

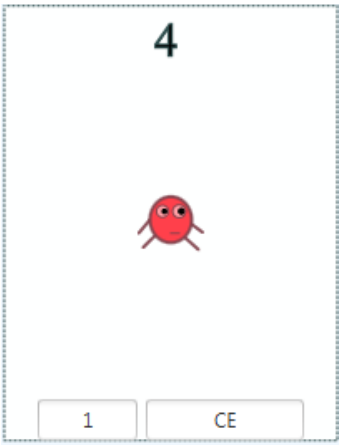


Notice that our goal actually contains two Pyramids. The outer Pyramid has a green Character in its attic. To build it, we will need to take out a green Character as a loan.

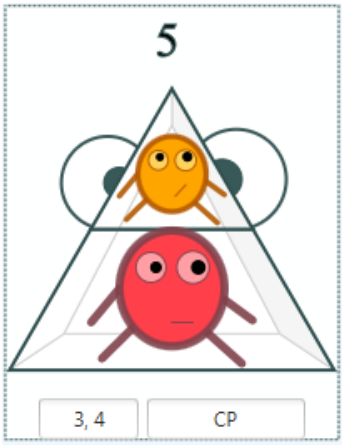
Next, we need to create the basement of the Pyramid we're trying to build. However, the basement is itself another Pyramid. So, we need to take out another loan.
The Pyramid in the basement has an orange Character in its attic. So, we take out a loan of an orange Character.



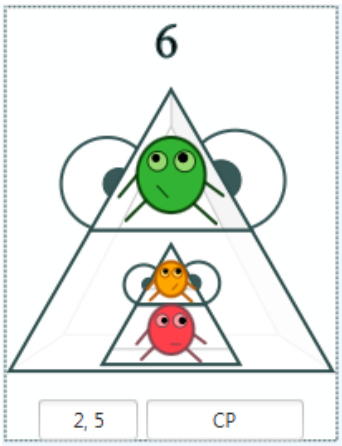
And that means our most immediate job is to create the basement of the Pyramid whose attic contains the orange Character. That Pyramid has a red Character in its basement. So, we extract the red Character from Card 1.



This gives us the basement of the Pyramid inside the Pyramid we are trying to build. We have the attic in Card 3, and the basement in Card 4, so we put the two parts together to make this:



And that means we now have the basement of the Pyramid we're supposed to be building. So, we take the contents of Card 5 as the basement of the new Pyramid, and the content of Card 2 as the attic of the new Pyramid, and construct the final Pyramid we were supposed to make.



The lesson of this example is that if you take out a second loan before your paid back your first loan, you have to pay back the second loan before you can pay back the first. Notice that Card 5 pays back the debt incurred by Card 3, and then Card 6 pays back the debt incurred by Card 2. Once Card 5 has been played, Cards 3 and 4 are no longer available. That mini-game is closed. But Card 2 is still available, because the first mini-game hasn't yet been closed. We use it to form the Pyramid in Card 6, which then puts Cards 2 through 5 all out of play. The only Card that remains available to Cards 7 through infinity, therefore, are Cards 1 and 6.

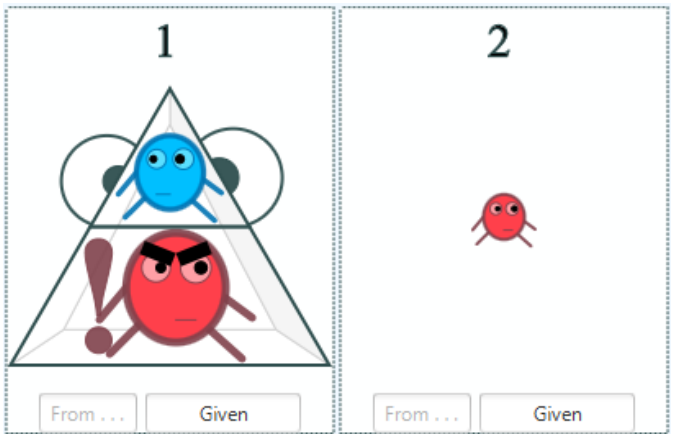
5.3: The Power of Indirect Proof (or "Reductio ad Absurdum") (A Landfill/Inference Rule)

The motto for this rule is: "If you want the opposite of something, put it in the attic of a collapsing Pyramid." This rule is odd, in that it typically depends on you using Conditional Proof first. That is, as it's normally used, it's a parasitic rule. It doesn't have to be, however. All you need is to have something in the attic of a Pyramid, and to have a Package in its basement whose two contents are "opposites" (i.e., two contents that are identical, except that one has one more Club than the other). If you have such a Pyramid, you can put whatever is in its attic in a new Card, and subtract a Club from it if it has one or more Clubs, or add a Club to it if it doesn't have any. (After all, it will have just fallen from the attic of a Pyramid, and thus will either be frightened or relaxed by the transition.)

<u>Restrictions:</u> This rule works even if the Pyramid has a Club. The Club just gets added to whatever was in its attic (after it falls).

The logical justification for this rule is that if something rests on a contradiction, then it must be false. After all, contradictions cannot be true, and thus anything that depends for its truth on a contradiction's being true will never be able to be true.

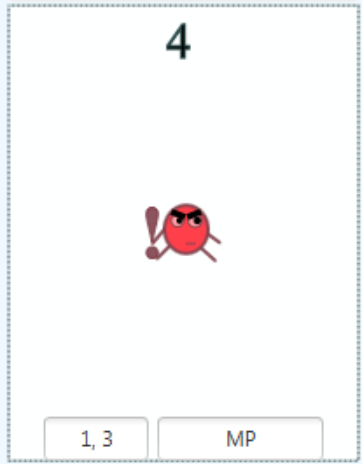
Example 1



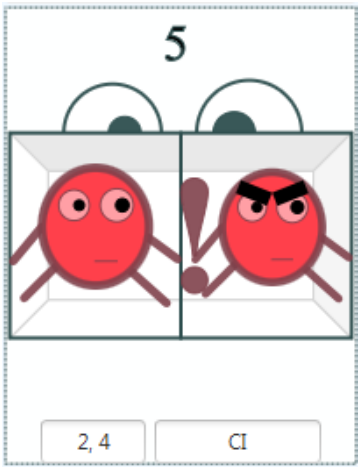
Imagine that we were given these two Cards, and were asked to produce the following Character:

We can use Indirect Proof to do this. We start by assuming the *opposite* of what we want. Since we want a blue Character with a Club, we assume a blue Character with *no* Clubs.

Now that we have the opposite of what we want, we try to show that this leads to a contradiction. So, we need to produce a Parcel that contains two opposite contents. We can do that if we put together the two red Characters, since one has a Club, and the other doesn't. To put them together, however, we need to get the red Character with a Club by itself. So, we use Modus Ponens to create Card 4.

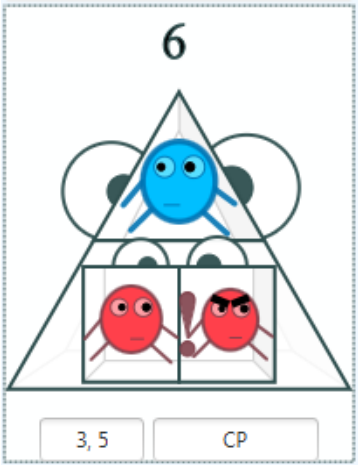


And then we use CI to put the two red Characters together.



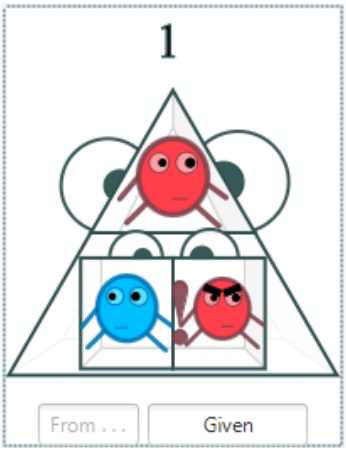
And when you put two opposites together like that in a Parcel, there will be war. If you were to put something on top of that Parcel, then, it would fall when the Parcel self-destructed.

Luckily for us, “putting something on top of the Parcel” is exactly what we need to do. The blue Character was a loan, and we need to pay it back by building a Pyramid. So, we use Conditional Proof to do just that.



All that is left for us to do, finally, is to point out that the basement of the Pyramid in Card 6 is fundamentally unstable and thus the entire Pyramid will collapse. When the collapse occurs, the blue Character will fall out and become frightened, producing Card 7.

Example 2

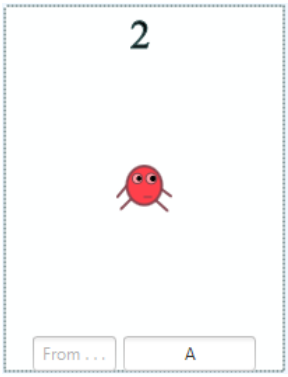


Imagine we were given the Card above, and asked to produce the Character below.

To get that Character, we would need to use Modus Ponens on Card 1. That would allow us to get the Parcel out of the basement of the Pyramid (and then we could extract the red Character with the Club, using Conjunction Elimination).

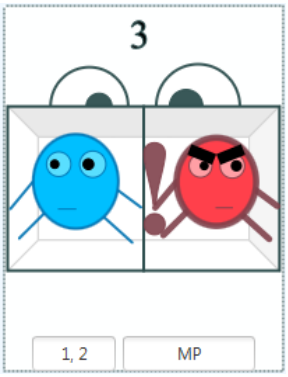
To use Modus Ponens, though, we would need a single red Character by itself (since that is what is in the attic of the Pyramid). But we haven't been given a single red Character by itself. So, we're stuck.

When we find ourselves stuck like this, often the way out is to use Indirect Proof. What we do is assume the *opposite* of what we want to produce. Since we want to produce a red Character with a Club, we make an Assumption Card that contains a red Character *without* a Club.

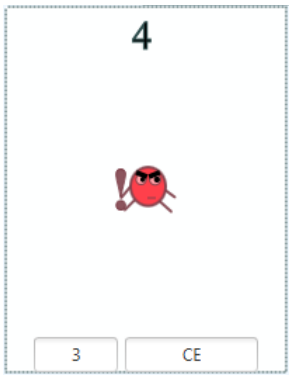


Now our job is to produce a Parcel that contains two opposite contents (i.e., it should contain the same thing in both chambers, but one thing should be holding a exactly one more Club than the other one is). It may not be clear to us at first what the Parcel should contain, however, so we should at least do what we can. What we *can* do is use Modus Ponens on Card 2 on Card 1. So we do that.

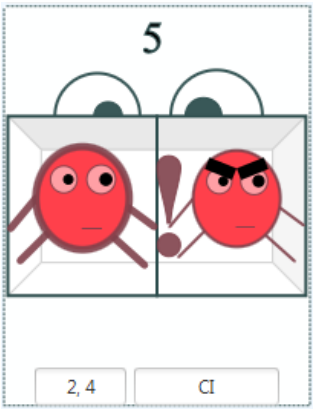
Modus Ponens on Cards 1 and 2 would produce this:



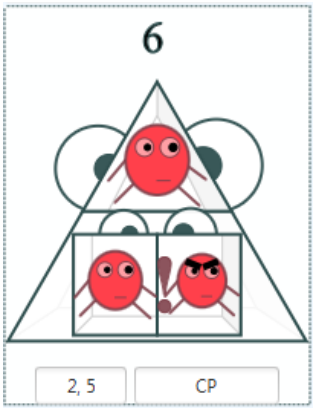
This comes from the basement of the Pyramid in Card one. And perhaps now it becomes clearer to us what our “contradictory Parcel” might contain. We have a red Character without a Club in Card 2, and we have a red Character with a Club in Card 3. All we have to do is extract the red Character with the Club, so we can put it together with the red Character from Card 2.



Now we have our contradiction. Card 2 has a red Character without a Club, and Card 4 has a red Character with a Club. They are opposites, and all we have to do is put them together into a Parcel to make the contradiction explicit.



But now we have to “discharge” the assumption from Card 2, or we won’t be able to finish the game. That means we have to build a Pyramid using the content from Card 2 as the attic and the content from Card 5 basement.

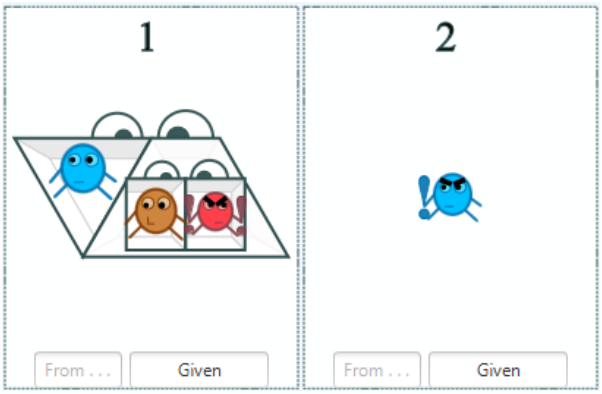


The Pyramid in Card 2 represents a fundamentally-unstable situation. The red Character with the Club in the basement is going to tear the basement apart trying to get at the other red Character in the basement (the one without a Club). In doing that, however, it will destroy the foundation of the Pyramid, and this will cause the whole thing to collapse. In the collapse, the red Character without a Club in the attic will fall out, onto ground level. In the fall, it will become frightened (since it was happy in the attic). And that means it will end up with a Club.

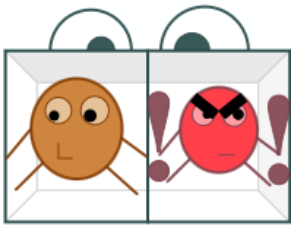
We win! That's what we were supposed to produce. However, you might ask why we couldn't stop way back at Card 4. After all, both Cards contain a single red Character with a Club. And a single red Character with a Club is what we were supposed to produce. So, why didn't we stop at Card 4?

The answer is that Card 4 was in the middle of the mini-game opened by the Assumption Card. We can't just stop there and claim we've won, since we're still in debt. We have to pay back the debt created by the Assumption Card before we can complete the game. So we have to keep going until we're in the clear, and have reached the goal outside the mini-game.

Example 3



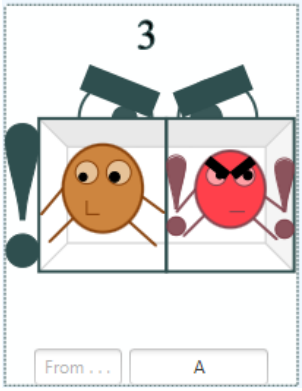
Here is a perfectly normal example. We are given the two Cards above, and asked to produce the Parcel below.



Fortunately for us, we know how to do this. All we have to do is apply Disjunction Elimination to Cards 1 and 2, and that will free the Parcel in the right Chamber of the Paragon in Card 1. That Parcel is identical to the goal we have been asked to produce, so we're done in three Cards.

However, the point of this example is to show how Indirect Proof can be used to solve even puzzles you know how to solve in other ways. The nice thing about Indirect Proof is that you can always use it, and you always use it in exactly the same way.

First, you figure out what you need to produce. In this case, it is the goal Parcel above. Then, you create and Assumption Card that has exactly the opposite of what you want to produce in it. So, let's do that.



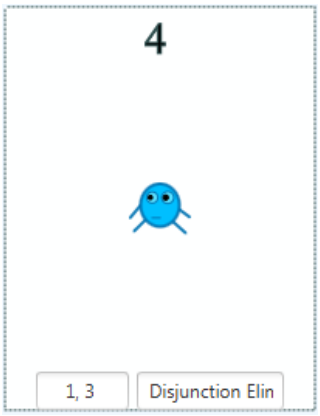
Now, all we have to do is show that we can use Card 3 to create a contradiction. That is, we need to use Card 3 to create a Parcel that contains two opposites in it. But that means we need to create the two opposites first.

So, in hopes of creating two opposites (two things that are identical, but one has exactly one more Club than the other), we use the Parcel in Card 3. "How do we use it?" you ask. "However we can," I answer. Or, more specifically, try to find a way to "use it on" or "use it with" one of the Cards that were already there before we made our assumption.

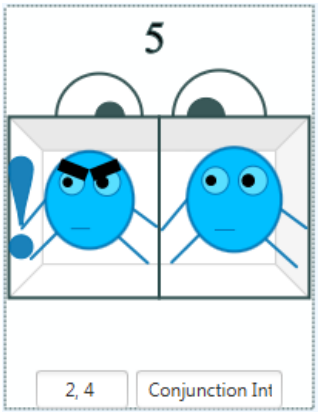
We had two Cards available to us before the Assumption Card, so we ask what we can do with those two Cards and our new Assumption Card. We could, for example, put the blue Character with the Club in Card 2, and the Parcel with the Club from Card 3 together into a Parcel in Card 4. Then maybe we could do De Morgan on that Card. However, we wouldn't end up producing the opposite of anything that way. (In other games, this might produce an opposite, but just not in this game.)

We could also put the Paragon from Card 1 into a Parcel with the “assumed Parcel.” But that wouldn’t end up helping produce an opposite either.

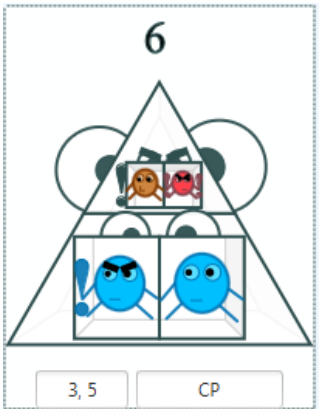
So, what if we used the Parcel with a Club in Card 3 to do Disjunction Elimination on the Paragon in Card 1? That would free the blue Character from the Paragon.



The reason doing this is a good idea is that we already have a blue Character with one Club in Card 2. That blue Character is the opposite of this new blue Character, since the new blue Character has no Clubs at all. If we put them together into a Parcel, then, we would have an explicit contradiction.



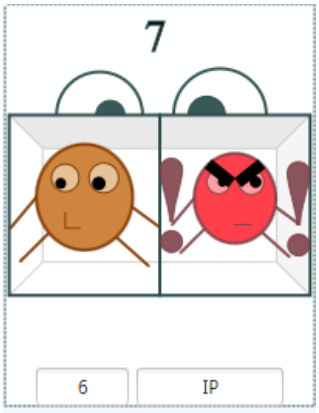
We now have what we wanted, and can build our Pyramid. Whenever we make an assumption, we have to build a Pyramid with the thing we created in the Assumption Card being in the attic. If we put something in the attic, with the Parcel from Card 5 in the basement, we’ll be creating a Pyramid that will immediately collapse, but that’s what we want.



In Card 6, we've used Conditional Proof to put the Parcel from the Assumption Card on top of the contradictory Parcel from Card 5. The blue Character with the Club will then set about attacking the blue Character without the Club, destroying the bottom level of the Pyramid and causing the whole thing to collapse.

In the collapse, the Parcel from the attic will fall out, losing its Club. After all, it was miserable up there in the attic, and now it is at ground level, where it wanted to be. "But wasn't it at ground level in Card 3, when we initially created it?" you wisely ask. "Not really," I respond. Card 3 gives us a picture of the attic of a Pyramid, in search of a basement. As far as the Parcel in Card 3 knew, it was in the attic of a Pyramid. It's just that it took us till Card 6 to find the basement of the Pyramid.

So, the Pyramid in Card 6 collapses, freeing the Parcel.

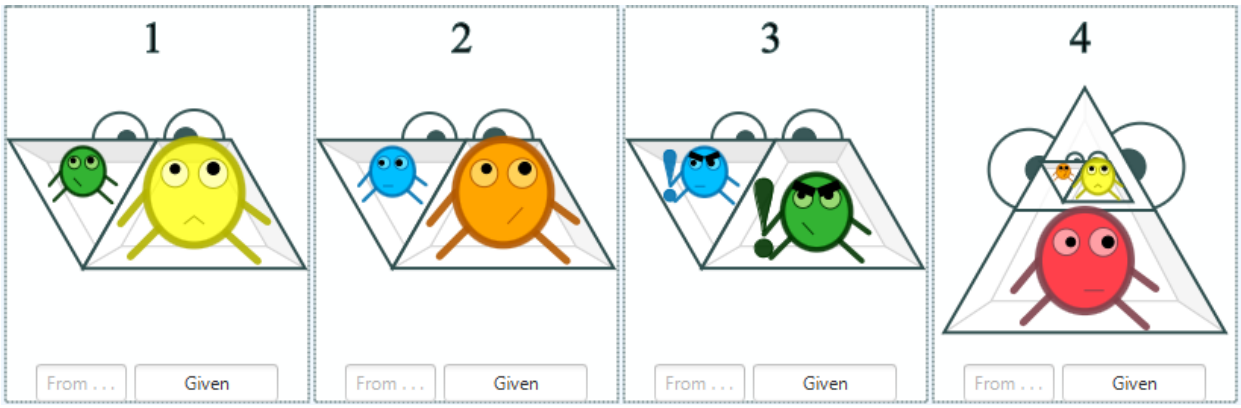


That Card cites Card 6 and Indirect Proof as its justification. It also contains the goal we were working toward all along. To reach the goal, we assumed the very opposite of our goal, showed that this produced a contradiction, and thus showed that anything resting on that contradiction had to fall.

We could have reached our goal much more quickly, if we had just used Disjunction Elimination right away. However, the point of this example was to show that Indirect Proof is always an

option. So, if you ever get stuck, try assuming the opposite of what you want, and then show how this leads to a contradiction.

Example 5
Okay, here's a weird example. Imagine that you were given the following four Cards:



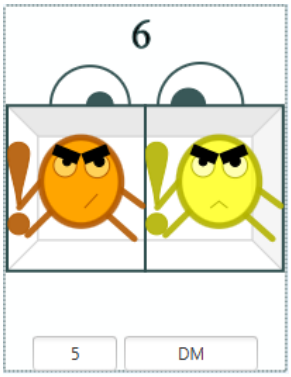
And imagine we were asked to produce the following Character.

We know we could get this Character from Card 4, if we only could "do a Modus Ponens" on it. But to do that, we'd need a Paragon containing one orange Character and one yellow Character.

Now, imagine we have no clue how to solve this puzzle. So, we decide to try Indirect Proof. And that means we'll need to assume the opposite of what we want, and show how that leads to a contradiction.

"What we want" is a red Character, by itself. So, we could assume a red Character with one Club. But we know how to get a red Character by itself, if only we had a Paragon with an orange Character and a yellow Character in it. So, even if we ultimately want a single red Character, what we "proximately want" is a Paragon with an orange Character and a yellow Character. So, let's assume the opposite.

Now, let's try to use that to create a contradiction. To ensure maximum possibility of success, let's get the yellow and orange Characters out of the Paragon. First, let's use De Morgan.

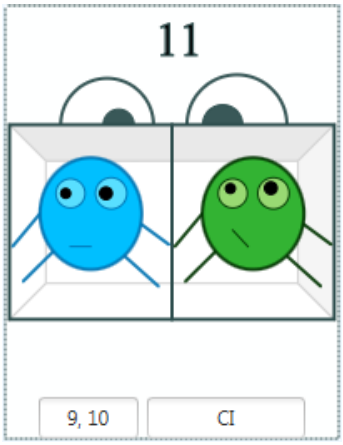


Then, let's use Conjunction Elimination twice.

Now these two Characters are free to wreak havoc. We see in Cards 1 and 2 that there are Paragons containing the opposites of our two new Characters, so let's do Disjunction Elimination on them both. The more Characters we release from their confines, the greater chance we have of putting together a contradiction.

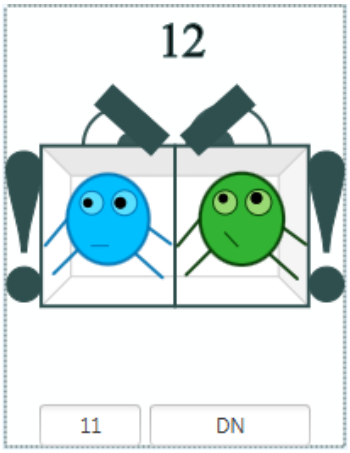
Now, we notice that we have a green Character and a blue Character available. Way back in Card 3 (one of the two Cards we haven't yet used), we had a blue Character with a green Character inside a Paragon. Do you think we could use our new green and blue Characters to create the opposite of what is in Card 3? Let's hope so, because the only other Card we haven't used is Card 4, and that's the Card we're supposed to do Modus Ponens on.

If we're going to make the opposite of the Paragon in Card 3, we'd have to put the green and blue Characters from Cards 9 and 10 together, using Conjunction Introduction.



Then, to get the opposite of what's in Card 3, we'd have to change the Parcel in Card 11 into a Paragon. And to do that, we need to do De Morgan.

If we're going to do De Morgan, however, we will need some Clubs. So first, we need to do Double Negation.

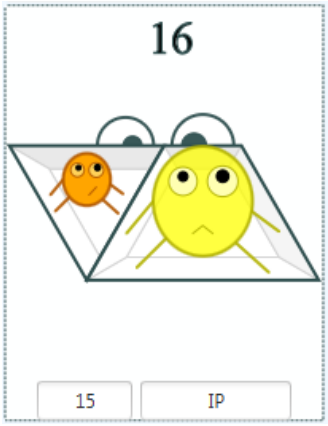


And now we can do De Morgan.

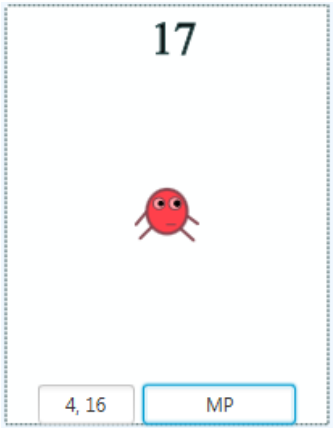
Revelation of revelations! The Paragon in Card 13 is the opposite of the Paragon in Card 3. Its contents are identical, but it has one more Club. Thus, we have our contradiction! All we have to do is make it explicit, as follows.

Next, as always, we build a Pyramid on top of the contradiction, with the contents of our Assumption Card in the attic.

But this Pyramid cannot stand. The Paragon with the Club will attack the Paragon without the Club, and the Pyramid as a whole will collapse. In the collapse, the Paragon will fall out, losing its Club.



We are *sixteen* Cards into this game, and we still haven't reached our conclusion. But at last we have what we need to produce the conclusion. The Paragon in Card 16 matches the attic of the Pyramid back in Card 4. This allows us to extract the red Character from the basement of that Pyramid, and a red Character was our goal all along.



That was insane. *Seventeen* Cards. We could have reached our goal much more quickly if we had used Implication on Cards 1 and 2, and then used Constructive Dilemma. But, once again, we have seen that Indirect Proof can be used to solve puzzles even when there are more “direct” ways of solving them.

Chapter 6: Level 6 Powers

6.0: Introduction

We spent the first half of the semester (Chapter 0–4) working our way up to the rules that deal with Pyramids. When we got to Contraposition, however, we "jumped back" to deal with the parallel rule (Commutation) for Parcels and Paragons. With those rules taken care of, we had all the "easy" powers in our hands.

With Chapter 5, we started in on the difficult powers. Since we finished the first half of the semester with Pyramid rules, we started with new Pyramid rules in Chapter 5. CP and IP are two of the most difficult powers you will have to master in symbolic logic, and you now know them inside and out.

There is one Pyramid rule left, however, and we will deal with it here in Chapter 6. While CP allowed you to put things into Pyramids, and IP allowed you to take things out of Pyramids, Exportation—the Pyramid rule we'll learn in this chapter—allows you to rearrange the contents of Pyramids. It is, in that way, similar to Contraposition.

Like Contraposition, furthermore, Exportation has a parallel rule that applies to Parcels and Paragons. That rule is called "Association." So, since we're dealing with Exportation in this chapter, we'll also learn Association.

Association, furthermore, has an extremely difficult hybrid-mutant cousin called "Distribution," that applies to Parcels and Paragons (like Association), but can involve mixing two different types of Chambergons (like Exportation). We will, therefore, learn Distribution in this chapter as well.

Each of the three rules for this chapter (Exportation, Association, and Distribution) is relatively straightforward. They're all recycling rules, and thus do not require multiple Cards as input. They can only be applied, furthermore, in specific, easily-recognizable situations.

However, I have found that *getting into the habit* of recognizing those situations is a huge challenge for students. Often the solution to a puzzle will require Exportation, Association, or Distribution, and students will find themselves in a situation where one of those three rules is applicable, but simply will not see it. So, these powers are going to take practice.

Furthermore, I have found that when students do recognize that Exportation, Association, or Distribution is applicable to a given Card, they often have a difficult time remembering exactly what applying the rule in question would produce. They apply the rule, but apply it incorrectly. In a "Powers On" game, this doesn't happen. You just let the computer do the work for you. But in Powers Off games, you will actually have to remember what gets doubled, what goes where, what shapes change, and so on. So, these powers are going to take practice.

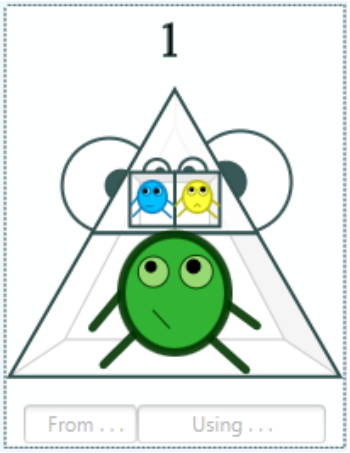
Once we have mastered these rules, furthermore, we'll be in a position to handle the Pulsar rules. We've been avoiding the Pulsar rules all semester, but in Chapter 7, we'll finally break down and talk about them.

For the moment, however, let's work on Exportation, Association, and Distribution.

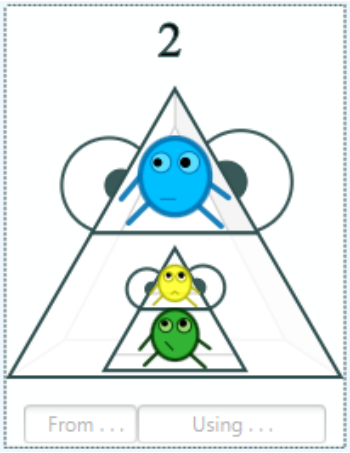
6.1: The Power of Exportation (A Recycling/Equivalence Rule)

Conjunction Introduction told you that you can pack things up into a box whenever you want. Conjunction Elimination told you it's always Christmas and you can open your presents whenever you want. Exportation combines the theme of packing and unpacking while adding Christmas trees to the mix.

This rule works for Pyramids that have a Package in the attic, like this:



Exportation also works for Pyramids that have a Pyramid in the basement, like this:



Think of the Pyramid in the basement as a Christmas tree, and the Package in the attic as holding the Christmas tree ornaments. If you have a Pyramid with a Package in the attic, this is *before* Christmas. You have all your Christmas tree ornaments packed away in the attic. The main ornament you have packed away in the attic is the star, or angel, or whatever you put on the top

of the Christmas tree. You take this ornament out of the *right* chamber of the Package in the attic, and put it on top of whatever is in the basement of the Pyramid. This forms a second Pyramid in the basement, with the "ornament" from the attic of the outside Pyramid now being in the attic of the inside Pyramid (and whatever was already in the basement of the outside Pyramid now being in the basement of the inside Pyramid). In doing this, you have put the ornament on top of the Christmas tree.

Now, if you have a Pyramid with Pyramid in the basement, this is *after* Christmas. You have to take the top ornament off the tree, and put it back in its Package in the attic. So, that's what you do. You take whatever is in the attic of the inside Pyramid, and stick it in the *right* chamber of a new Package in the attic. (The left chamber of that Package will contain whatever was already in the attic). The Pyramid in the basement disappears, leaving behind whatever was in *its* basement.

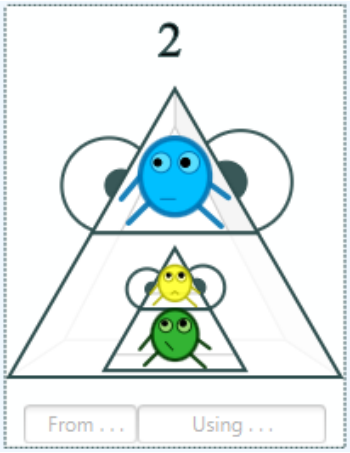
So, the motto for Exportation is: "You can pack a Package in the attic from a Pyramid in the basement, and vice versa."

NOTE 1: There is no giving or taking of Clubs in Exportation, even though we are dropping/raising things between attics and basements. Why not? It's because the thing being moved is being moved from one attic to another, and thus does not feel it has either been raised or lowered. Since it starts in an attic, and ends up in an attic, its feelings don't change. If it was happy where it was, it will be happy where it is. If it was miserable where it was, it will be miserable where it is.

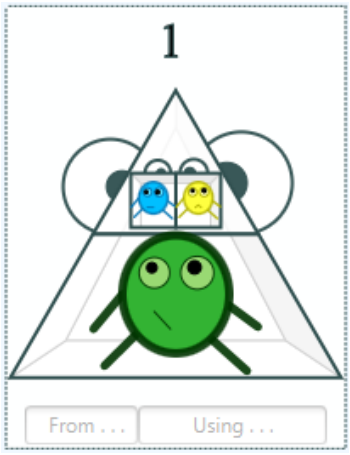
NOTE 2: You will notice that Exportation combines two of a logician's four fundamental rules. It is primarily a matter of rearranging the contents of a Chambergon. However, in rearranging those contents, you are removing something from one Chambergon, and putting it into another.

Restrictions: This rule will not work if the Package you're trying to "unpack" in the attic has a Club. Nor will it work if the Pyramid in the basement whose "ornament" you are trying to remove has a Club. You just can't take things out of things that have Clubs. However, if the outside Pyramid has a Club, that is fine, since you're just rearranging its innards. Likewise, it is fine if the outside Pyramid is itself inside some other Chambergon (including another Pyramid).

The logical justification for this rule is as follows. Think about what a Pyramid with a Pyramid in its basement means. We'll use the Card 2 from above again:

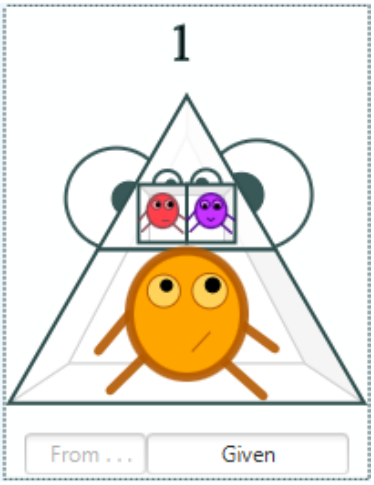


This Card says that if blue is going to be “held up,” you need to have yellow and green, and if yellow is to be held up, you need green. In other words, both blue and yellow require green for their support. In still other words:

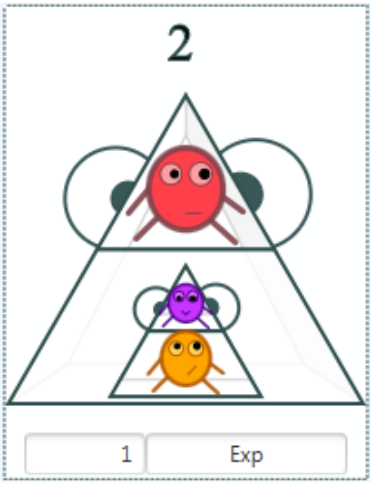


Do you see how Card 1, above, says that both blue and yellow rest on, or are supported by, green? After all, Parcels represent conjunctions (“and”), and there’s a blue Character and a yellow Character in a Parcel that is resting on top of a green Character. Cards 1 and 2, in other words, represent the same basic logical situation, and therefore it is logically legitimate to move from one to the other and back again. This is all the Exportation rule is saying.

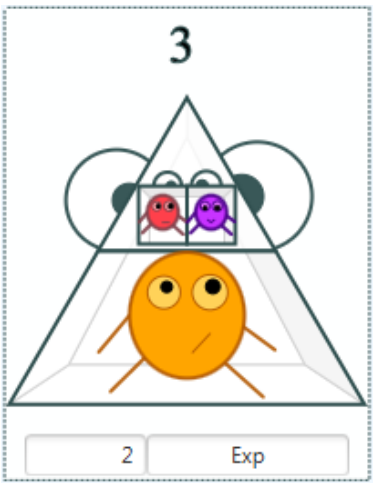
Example 1



Imagine that we were given the Card above. We notice that it contains a Pyramid with a Parcel in its attic, and the Parcel doesn't have a Club. This tells us that we can apply Exportation to it ("exporting" the purple Character to the basement). So, we do so and produce this:

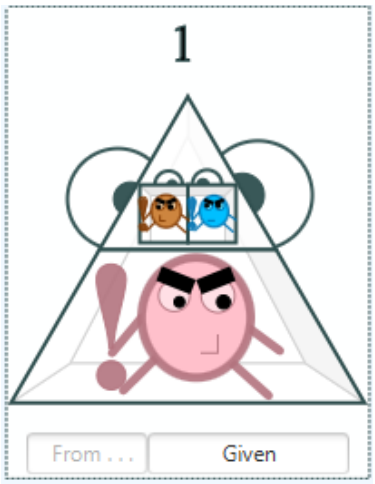


However, we now have a Pyramid with a Pyramid in its basement, and the basement Pyramid doesn't have a Club. This tells us we can apply Exportation again ("exporting" the purple Character to the attic). So, we do so, and produce this:



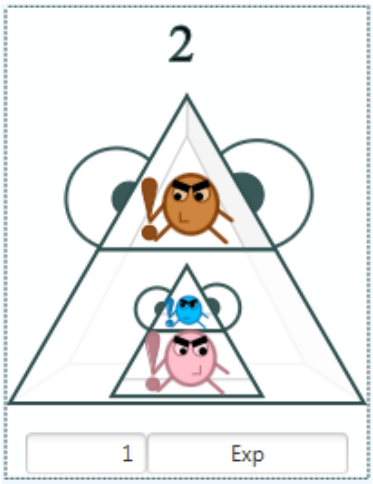
In other words, with two applications of Exportation in a row, we have ended up back where we started.

Example 2

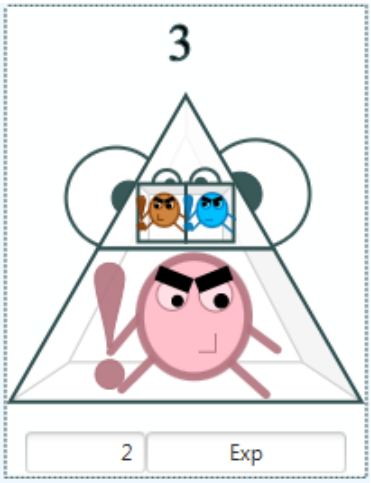


Now, let's imagine we are given the Card above and asked to "export" it. We see the Clubs everywhere and worry at first that Exportation won't be allowed. Then we realize that the Parcel in the attic of the Pyramid doesn't have a Club, so we're all good. Whether or not the thing we are moving from the attic to the basement (or vice versa) has a Club doesn't matter. What matters is whether the Parcel in the attic, or the Pyramid in the basement, has a Club.

In the above Card, we have a Parcel with no Clubs in the attic of a Pyramid. If we apply Exportation to it, we would produce this:

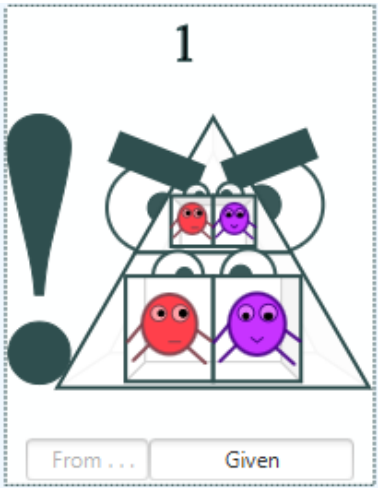


In Card 2, the blue Character has moved from the attic of the outside Pyramid (or, rather, from the Parcel in the attic of the outside Pyramid) to the attic of the Pyramid in the basement. But this puts us in a position to apply Exportation again. As before, all the Characters have Clubs, but the Chambergon inside the outer Pyramid does not have a Club. So, if we apply Exportation, we produce this:



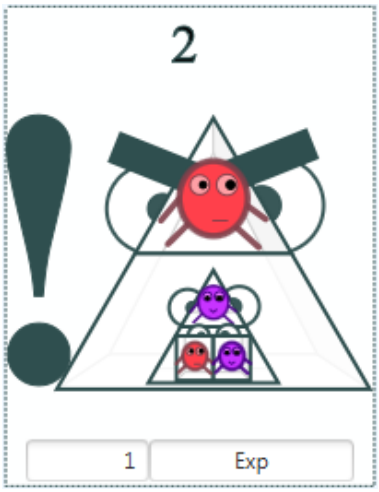
The lesson of this example is that if the occupant of the basement of the Pyramid has a Club, this will not stop Exportation from happening so long as there is a Parcel in the attic, and that Parcel doesn't itself have a Club. If the Parcel in the attic had a Club, no Exportation in this example would have been possible. After all, when a Parcel has a Club, you can't unpack it. It will smack your hand with the Club. Likewise, if the Pyramid in the basement in card 2 had a Club, we wouldn't be able to take the blue Character out of its attic and pack it up in a Parcel in the attic of the outside Pyramid.

Example 3



In this example, we have a Pyramid with a Club. This Club, however, does not keep us from applying Exportation, since we are not taking anything out of the Pyramid. Inside the Pyramid, we have two Parcels. The Parcel in the attic means we can apply Exportation to this Pyramid. The Parcel in the basement is irrelevant.

So, we apply Exportation to Card 1, and produce this:



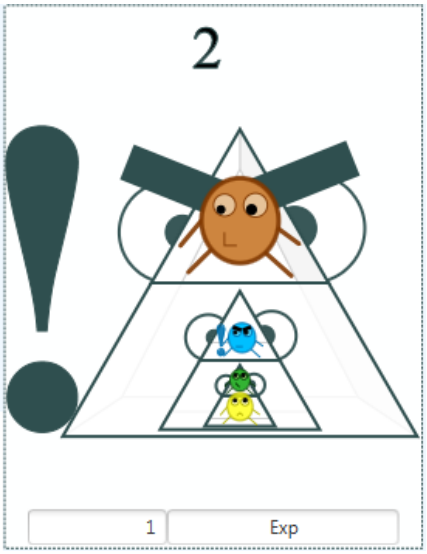
The purple Character from the attic Parcel has moved to the basement, but ended up in the attic of the new Pyramid that has appeared in the basement. This means we have a Pyramid with a Pyramid in its basement. And since the inner Pyramid does not have a Club, we can apply Exportation again to produce this:

This example has two lessons for us. First, the fact that the "outside" pyramid has a Club does not interfere with Exportation. Second, the fact that there is not only a Parcel in the attic, but also in the basement, does not interfere with Exportation. You simply treat the Parcel in the basement as if it were like any other occupant. When you drop the content of the right chamber of the attic Parcel into the basement, whatever is in the basement ends up in the basement of the new, inside Pyramid.

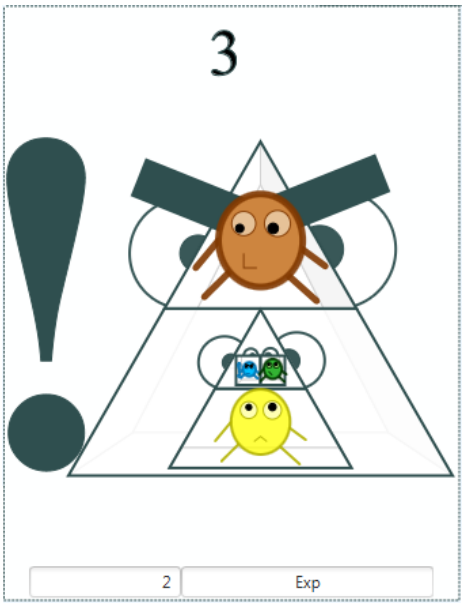
Example 4

Here, we have a situation similar to that in the previous example. We've got a Pyramid that contains two other Chambergons. The difference is that this time, the Chambergon in the basement is another Pyramid, rather than a second Parcel. This means we can "go either direction" with Exportation. We can either move the blue Character from the attic to the basement, or move the green Character from the basement to the attic.

Let's use Exportation to move the blue Character first:



In Card 2, we've created a new "inside" Pyramid with the blue Character in its attic. However, this new Pyramid now contains the old "inside" Pyramid inside its basement, and that means we can apply Exportation to it. If we do so, we produce this:



In Card 3, the green Character has moved from the attic of the innermost Pyramid into the attic of the Pyramid we created in Card 2. We've packed it up with the blue Character that had started off in the Parcel in the attic of the outermost Pyramid.

In Card 3 we have a standard Exportation situation. We have a Pyramid with a Pyramid in its basement, and the basement Pyramid has no Club. What is unusual about this situation is that the basement Pyramid's attic contains a Parcel. This means that we can either apply Exportation to the outer Pyramid, or to the inner Pyramid. Since we just did Exportation on the inner Pyramid, let's apply it to the outer Pyramid this time.

If we use Exportation on the outer Pyramid, the Parcel from the attic of the basement Pyramid will be moved to the attic of the outer Pyramid, like this:

In Card 4, the Parcel from the attic of the basement Pyramid has itself been packed away in another Parcel in the attic of the outer Pyramid.

Now, you will recall that way back in Card 1, we had a choice. Either we could move the green Character down, or the brown Character up. We just to move the green Character down, but let's go back to Card 1 and take the other option this time.

If we do so, we produce the contents of Card 5.

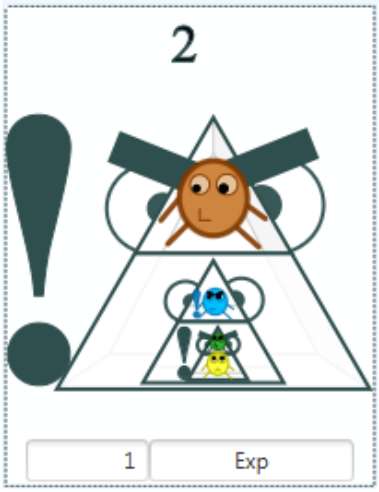
In Card 5, we have taken the brown “ornament” from the top of the basement “Christmas tree” and put it in a box with the other “ornaments” in the attic.

(Now, compare Card 4 with Card 5. Do you see how you could move back and forth between them using Commutation?)

Example 5

This example is identical to example 4, except that the inside Pyramid has a Club. This makes it impossible to Export in both directions (i.e., from both the attic into the basement, and the basement into the attic). Since the inside Pyramid has a Club, you cannot pack the content of its attic into a Parcel in the attic of the outside Pyramid. If you tried to do that, the inside Pyramid would fight you off with its Club. Something similar would be true if the Package in the attic had a Club. Then you wouldn’t be able to unpack whatever was in its right chamber and lower it on top of whatever was in the basement.

Given the restrictions created by the Club, then, the only thing we can do with Exportation is to lower the blue Character from the attic. If we do so, we produce this:



While the new inner Pyramid contains the old basement Pyramid in its basement, the old basement Pyramid still has its Club. That means we cannot perform Exportation on the new inner Pyramid. Instead, if we were to "export," we'd have to simply move the blue Character back to the attic of the outer Pyramid, like so:

6.2: The Power of Association (A Recycling/Equivalence Rule)

Exportation allows you to take something out of the attic of a Pyramid that's inside another Pyramid, and put it in a new Parcel in the attic of the outer Pyramid (or to take something out of the Parcel in the attic of a Pyramid, and put it in the attic of a new Pyramid in the outer Pyramid's basement). Much the same type of move can be made inside Parcels and Paragons themselves. However, unlike Exportation, which works with *either* a Pyramid in a Pyramid *or* a Parcel in a Pyramid, Association only works with Parcels inside Parcels, and Paragons inside Paragons.

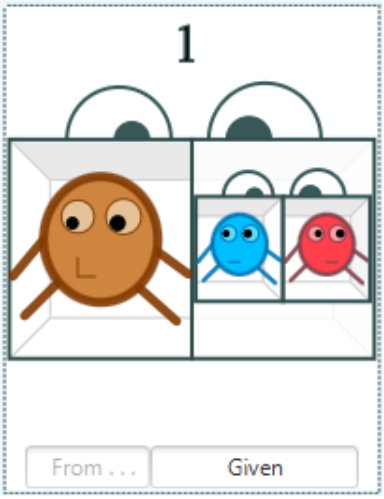
With Parcels, the motto for this rule is, “Parcels inside Parcels can be repackaged and repacked.” With Paragons, the motto is, “Paragons inside Paragons can pierce the chamber barrier and consume their companion.” This metaphor of “consumption” also requires a concomitant concept of excretion (fortunately or unfortunately, depending on what you think about such things). In consuming one Piece, the inner Paragon also must excrete a Piece to make room. Both mottos will be clearer once you see the examples below.

Restrictions: If the “outside” Parcel has *two* Parcels in it—or if the outside Paragon has two Paragons in in it–one in each of its chambers, you have to treat one of those “inside” Parcels/Paragons as if you could move it around, but couldn’t open it or rearrange its contents. You get to pick which one, but it has to be one or the other. The same thing goes for any “inside” Parcel that has a Club. If it has a Club, you can move it around, but can’t open it or rearrange its contents. Again, this will be clearer once you see the examples below.

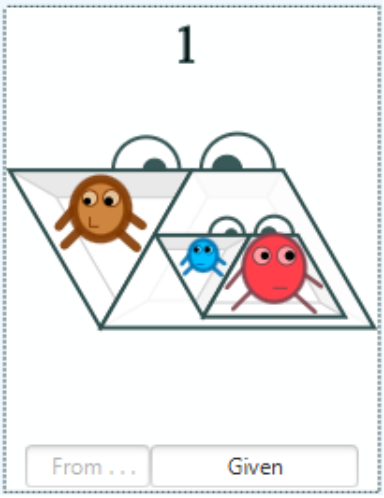
The justification for Association is that if you have a Parcel inside another Parcel (and the inside Parcel doesn’t have a Club), then everything’s in a Parcel with everything else. Similarly, if you have a Paragon inside a Paragon (and the inside Paragon doesn’t have a Club), then everything’s in a Paragon with everything else. It really doesn’t matter, therefore, how things “on the inside” are arranged. One internal setup is as good as the next. In other words, if you have blue, orange, and green all inside a Parcel, and two of them are inside a Parcel inside that Parcel, then it doesn’t matter whether the blue and orange are together inside the inner Parcel, or orange and green are together inside the inner Parcel. In the final analysis, they’re all in the outermost parcel, and each is “bound” to the next by a Parcel.

To put it in yet other words, the power of Association depends upon the fact that it doesn’t really matter who associates with whom inside a Parcel or Paragon, so long as everyone is in a Parcel or Paragon.

Example 1

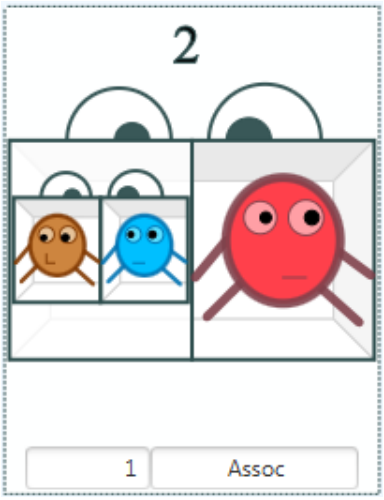


In this Card, we have a Parcel inside a Parcel. This means we can apply Association. However, the same would have been true even if we had had a Paragon inside a Paragon, like this:

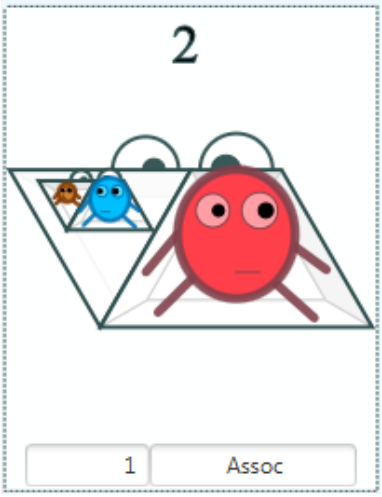


In either version of Card 1, we have three Characters bound together by two Chambergons of the same type. Since they are all Parcels (in the first version) or all Paragons (in the second version), it is ultimately irrelevant whether the inner Chambergon is in the left or right chamber of the outer Chambergon. We can move it, therefore, so long as we keep the overall order of the Characters the same.

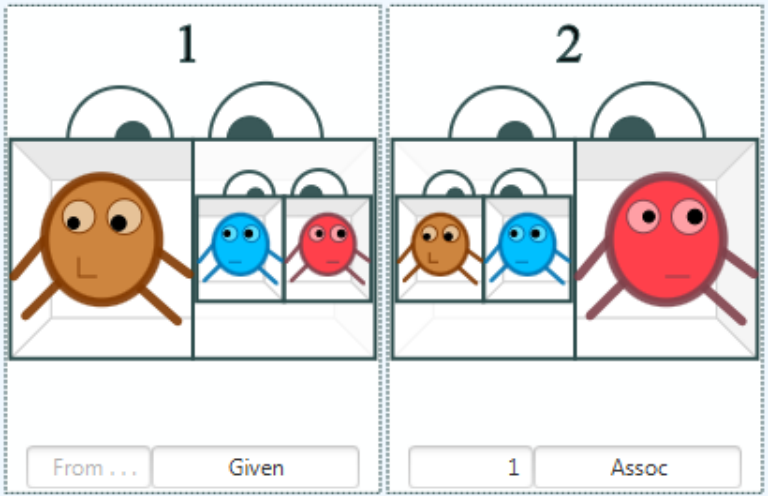
So, if we apply Association to Card 1, we produce this:



Or, if we are using Paragons:

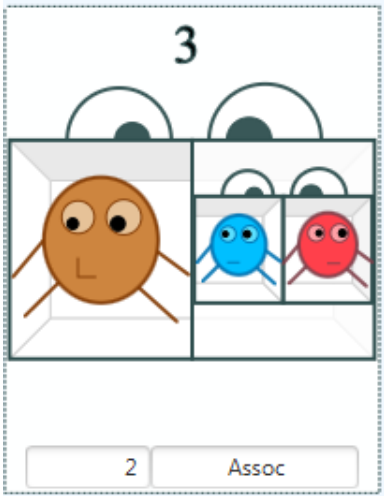


You will notice that the order of Characters has not changed. It started as brown–blue–red in Card 1, and ended as brown–blue–red in Card 2. All that has changed is that the inner Parcel or Paragon has changed chambers, and brought one of its contents along with it.

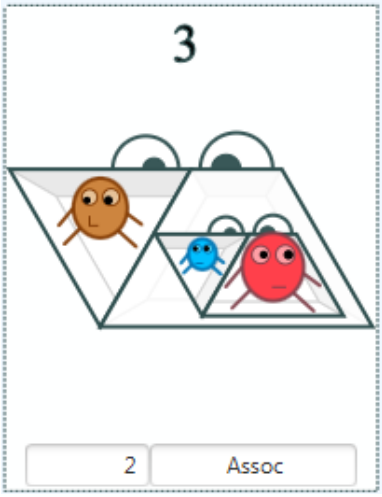


An application of Association looks, therefore, very much like an application of Commutation. The inner Chambergon, after all, has "swapped" to the other chamber. The difference, however, is that it has not replaced the thing that was in the other chamber, forcing that thing to switch chambers as well. Instead, it has *consumed* the thing in the other chamber, and left one of its own contents behind in its old chamber.

In both versions of Card 2, the application of Association leaves us in a position where another application of Association is legitimate. We still have a Paragon with a Paragon inside it, or a Parcel with a Parcel inside it. If we were to apply Association again, we would produce this:



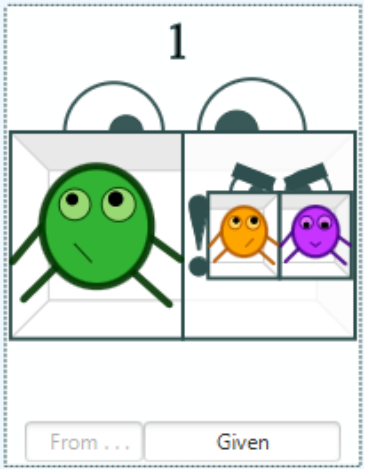
. . . or in the Paragon version, we would produce this:



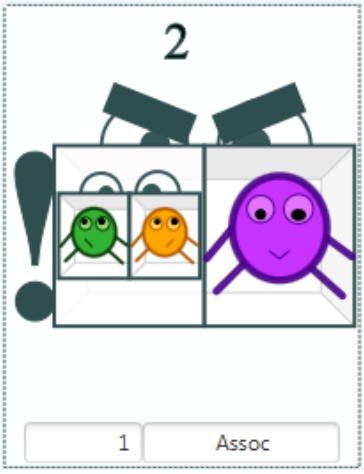
In either case, we end up back where we started.

Example 2

Compare the Card above to this Card:

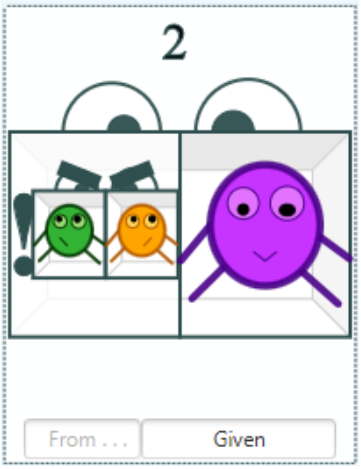


There is exactly one Club in both Cards, but the Clubs are placed differently. In the first Card, the Club is on the outside of the outermost Parcel. In the second, it is on the outside of the inner Parcel. This means that Association is applicable to the first Card. If we used Association on it, we would produce this:



However, if we tried to apply Association to the Card with a Club attached to the *inner* Parcel, nothing would happen. A Parcel with a Club can't be opened, and you have to be able to open the inner Parcel in order to take one of its contents out, and put another thing in.

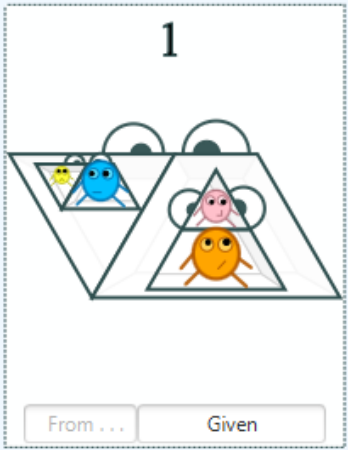
Now, compare Card 2 above, with the following alternate version of the same Card:



We can apply Association to the original version of Card 2 (the one with the Club on the outermost Parcel). If we did, we'd produce this:

We cannot apply Association to the alternate version of Card 2, however. The Club on its inside Parcel will keep us from opening that Parcel and replacing its contents.

Example 3

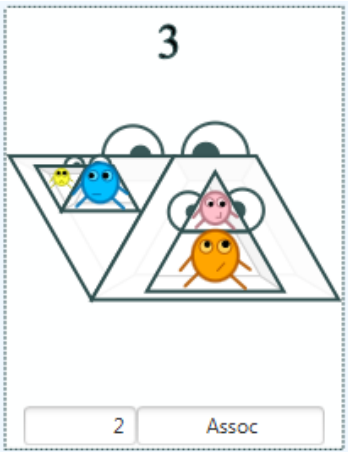


It will not be immediately obvious that Association is applicable to the above Card. However, once we think about it for a moment, we'll see that it is. What we have in Card 1 is a Paragon with another Paragon inside it. The inside Paragon, furthermore, does not have a Club. But a Paragon with a clubless Paragon inside it (or a Parcel with a clubless Parcel inside it) is all we need to apply Association. The fact that "the other chamber" happens to contain a Pyramid doesn't change that fact. We simply treat the Pyramid in the same way as we were treating the Characters in previous examples.

If we apply Association to Card 1, we would produce this:

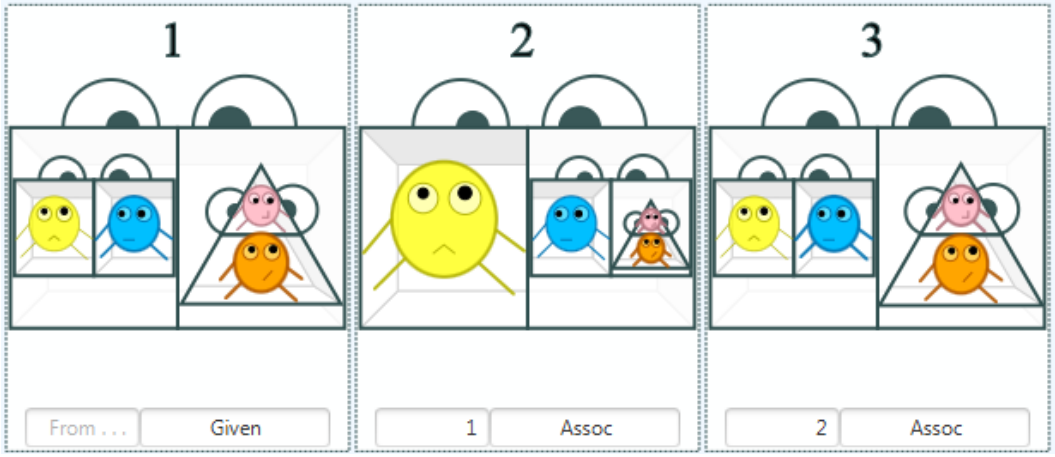
Notice that the order of the basic contents in Card 1 and 2 are the same: yellow–blue–pyramid. All that really changes is the inner Paragon, which shifts from one chamber to the other (and brings the blue Character along with it.

Notice furthermore that Card 2 presents us with essentially the same situation as Card 1. We have a Paragon with a Paragon inside it. The fact that the inner Paragon contains a Pyramid is irrelevant. So, we can apply Association again. If we do so, we will produce this:



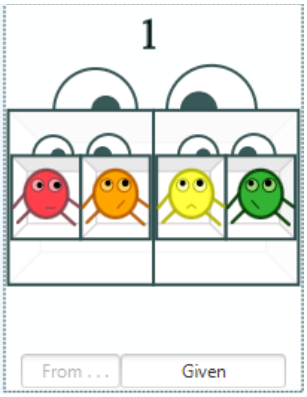
That is, we would have gotten back to where we started.

Now, recall that Association applies to both Paragons and Parcels. So, imagine that we had the same example, but every Paragon had been transformed into a Parcel. In that case, we would have had this:

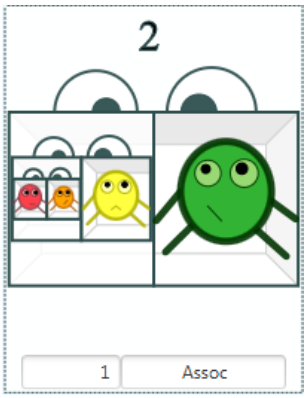


In each of the three Cards, we have a Parcel that contains a Parcel, and thus Association is applicable to each Card. We simply have to treat the Pyramid as if it were a single thing, like a Character.

Example 4



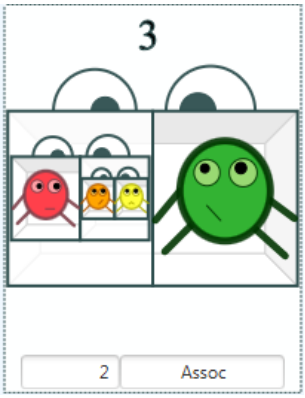
Imagine that you have the Card above. In this case, the Parcel contains *two* Parcels, and thus we can "do Association" in either direction. Either the Parcel on the left can swallow up the Parcel on the right, or vice versa. If we go with "vice versa," we produce this:



Notice that, as usual, the order of the fundamental contents does not change. It remains red–orange–yellow–green. What has changed is that the Parcel that was in the right chamber in Card 1 is now in the left chamber, and has left behind only the green Character in the right Chamber. Furthermore, the Parcel that moved has consumed the Parcel that was already in the left Chamber.

Due to the fact that one of the inner Parcels has swallowed up the other, we are once again faced with two possible ways of applying Association. We can either use association to restore things to the way they were when we began, or we can use Association on the inner Parcel that now itself contains another Parcel.

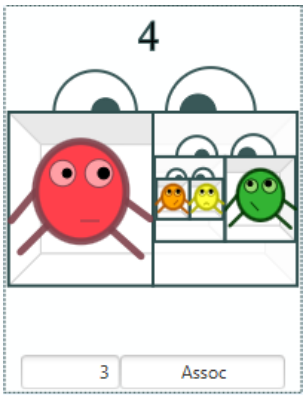
Imagine that we take the latter option. If we do so, we produce this:



As usual, the order of the Characters has remained the same, even though the innermost Parcel has switched chambers. It has moved from the left chamber of the “middlemost” Parcel to the right chamber. In the process, it left the red Character behind and picked up the yellow Character.

Now, imagine that we want to apply Association to the outermost Parcel again. After all, it contains a Parcel in its left Chamber, and that Parcel is clubless. What would we produce?

The answer is this:

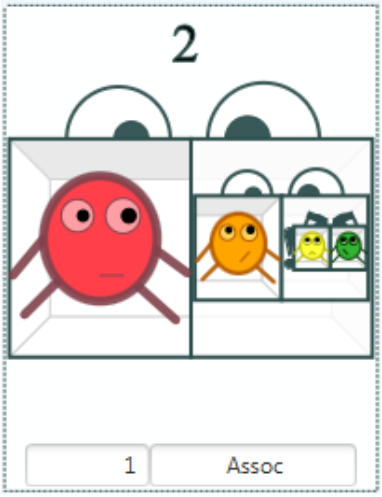


In this Card, the middlemost Parcel has moved from the left chamber to the right chamber of the outermost Parcel. It leaves the red Character behind, brings the innermost Parcel along with it, and picks up the green Character. And, once again, the order of the Characters has remained constant.

Example 5

This example is identical to the last, except I have added a Club. If I had added the Club to the outermost Parcel, it would not have affected the applicability of Association. However, since I have added it to one of the inner Parcels, that means that Parcel can no longer be “unpacked” or given new contents. Thus, we have to treat it in the same way we’d treat a single Character. If we want to apply Association, we’ll have to move the Parcel in the left chamber over to the right, rather than being able to move both the left and right Parcels.

Our only option for using Association—given this version of Card 1—therefore, is to do this:



6.3: The Power of Distribution (A Recycling/Equivalence Rule)

I don't know what you thought about the two previous rules, but this one is weird. "Distribution" may sound like "Exportation," since when you "export" a product you are "distributing" it to another country. However, the rule of Distribution doesn't involve moving one thing from one chamber to another. Instead, it involves flipping a Chambergon inside out, so that what was once on the outside is now on the inside.

This rule applies to both Parcels and Paragons. If a Parcel contains another Parcel or a Paragon in its right chamber, and that "other Parcel" or Paragon doesn't have a Club, then we can use Distribution. Similarly, if a Paragon contains another Paragon or a Parcel in its right chamber, and that "other Paragon" or Parcel doesn't have a Club, then we can use Distribution.

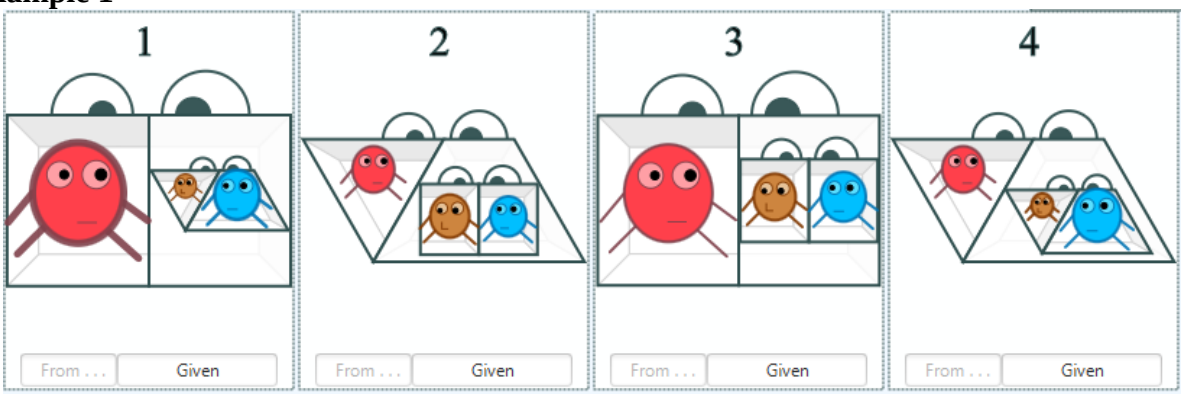
The motto for this rule is, "If there's a Paragon or Parcel in a Parcel's *right* chamber, you can stuff the outer Parcel into the inner Paragon or Parcel (twice); likewise, if there's a Parcel or Paragon in a Paragon's *right* chamber, you can stuff the outer Paragon into the inner Parcel or Paragon (twice)." That is, you can have the inside Paragon or Parcel swallow up the Parcel or Paragon that it's in. It's a freaky event made possible by warped spacetime or something.

Furthermore, after you've flipped things inside out, you can pop everything back into place by applying the same rule. That is, you can have the Parcel or Paragon regurgitate the Paragon or Parcel it swallowed. However, to explain exactly how all of this works, I'll need to show you some examples.

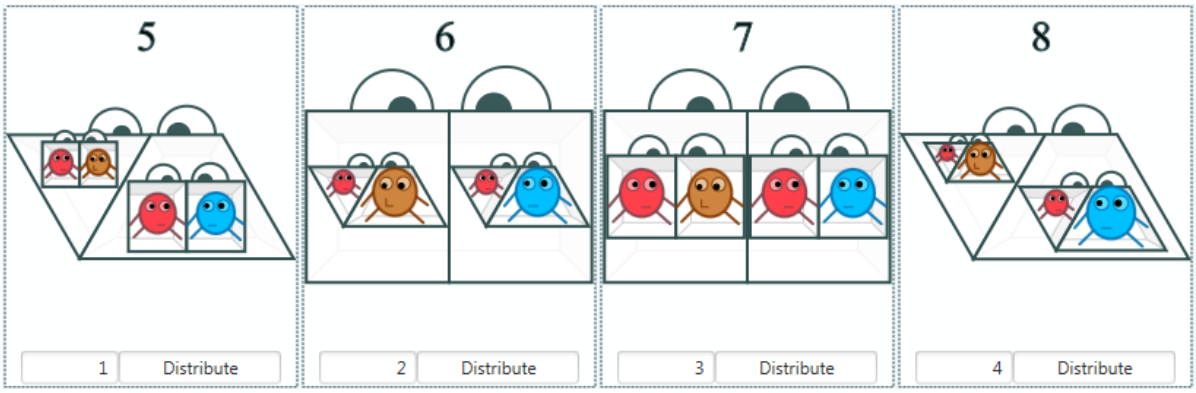
<u>Restrictions</u>: This only works on Parcels or Paragons that have a Paragon or Parcel in their *right* chambers. Furthermore, it won't work if the inner Paragon or Parcel has a Club. You won't be able to stuff anything inside it, because it will fight you off. It doesn't matter, however, whether the outer Parcel or Paragon has a Club. It will just stay where it is, and will become the Club of whatever Chambergon ends up being on the outside after you've finished the Distribution.

We'll deal with the logical justification for this rule below.

Example 1



Take a look at the four Cards above. We could do Distribution on any one of them. Each has a Parcel or Paragon on the outside that contains a Parcel or Paragon in its right chamber. If we "Distributed" each of the Cards above, we would produce the following four Cards.

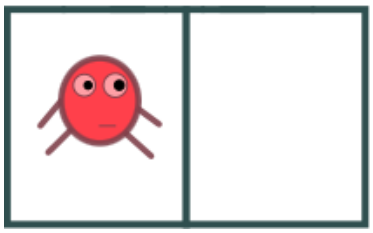


In the two rows of Cards above, Card 5 is Card 1 "Distributed," Card 6 is Card 2 "Distributed," and so on. So, to see what Distribution does to a Card in the first row, just look at the Card directly below it in the second row.

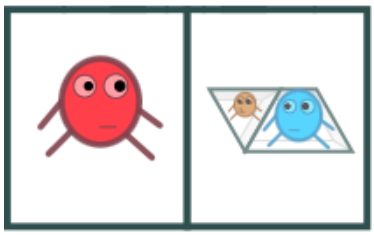
Now, look at what Distribution does. The outermost Chambergon in Card 5 is a Paragon. That Paragon used to be inside the Parcel in Card 1. Now it contains *two* versions of the Parcel from Card 1. The Chambergon that used to be on the inside is now on the outside, and the Chambergon that used to be on the outside is now on the inside *and doubled*.

The same thing occurs in the move from Card 2 to Card 6. The Paragon that began on the outside is doubled, and ends up inside the Paragon that used to be on the inside. And the same thing happens in the move from Card 3 to Card 7, and in the move from Card 4 to Card 8. The fact that the Chambergons on the inside and outside are the same might mask the fact, but after Distribution has been applied, the piece on the outside is the one that used to be on the inside, and the two pieces on the inside are the doubled versions of what used to be on the outside.

To help cement in your minds what is going on here, I would like to offer the following illustration. Imagine that a Parcel is actually a window with two panes of glass. On the left pane of glass, there is a sticker, or decal, that looks like a red Character.

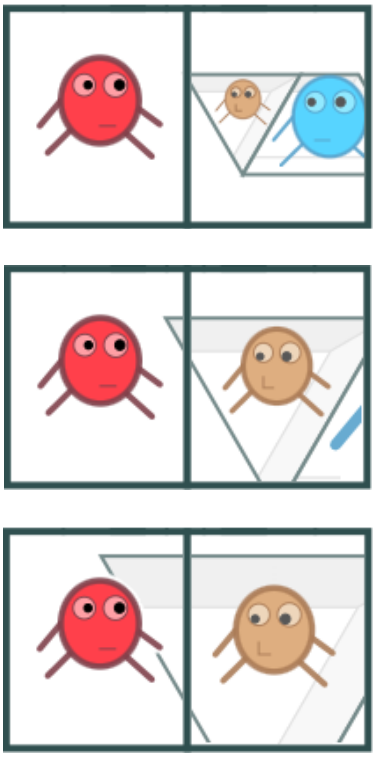


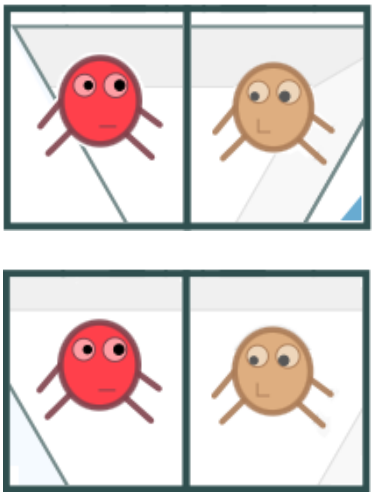
The right pane of glass is clear, however, and through it we can see a Paragon off in the distance. Since it's far away, it looks small. In fact, it looks small enough to fit within the right window pane.



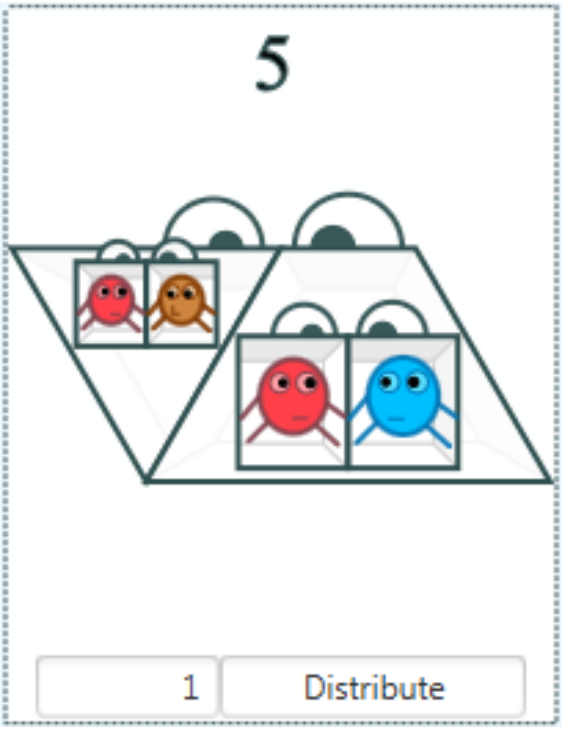
But if you got up close to the Paragon, you'd see that the brown Character and blue Character it contains are the same size as the red Character decal that we stuck to the left window pane.

Now, imagine that you are carrying the window. It's not stuck in a wall, but rather you are holding it in your hand. Imagine walking towards the Paragon, carrying the window. Through the right pane of the window, you see the Paragon and its Characters growing larger and larger, as you get closer to them.



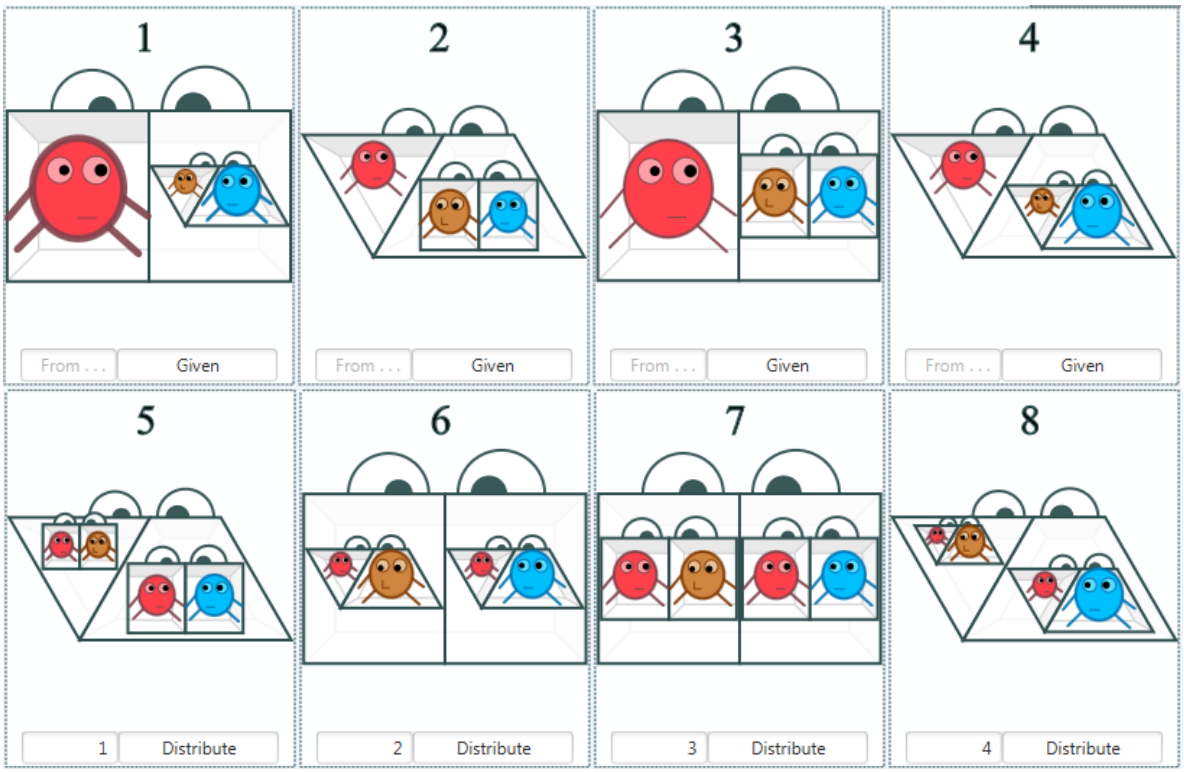


And that is where the Parcel in the left chamber of the Paragon in Card 5 comes from.



We could then repeat the process, “zooming in” on the blue Character instead. That would produce the Parcel in the right chamber of the Paragon.

Now, let’s go back to the example we were studying. It looked like this:



It is important to notice the pattern that all the Cards in the second row share. Each Card contains a Parcel or Paragon that contains two Parcels or Paragons, and the left chambers of those innermost Parcels or Paragons contain identical contents. In this example, every left chamber of the innermost Parcels and Paragons contains a red Character.

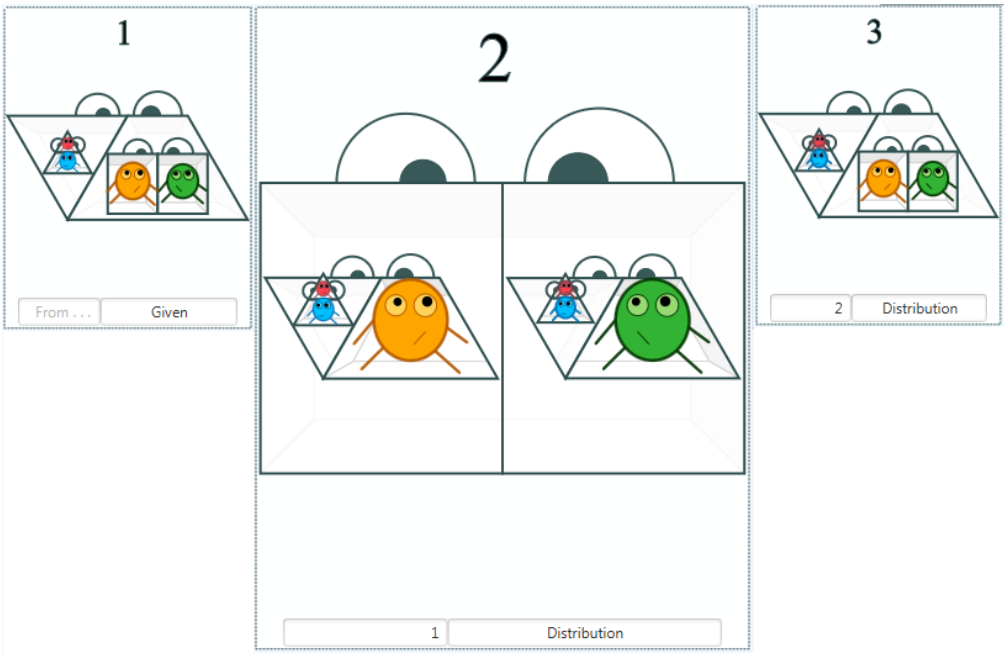
Whenever you see a Parcel or Paragon, therefore, that contains two Parcels or Paragons (one in each chamber) that do not have Clubs, and those two Parcels or Paragons have identical left chambers, you know you are looking at the product of a Distribution. Furthermore, since Distribution is a recycling rule, you know that you can apply it again to "undo" the distribution. You can not only go from Card 1 to Card 5, but from Card 5 to Card 1 (as it were).

To undo a Distribution (by apply Distribution) is, in effect, like refocusing your eyes after seeing double. In that way, Distribution is much like Tautology. It's just a bit more messy.

Before we move on to Example 2, I promised I would discuss the logical justification of this move. Look at Card 1. It has a red Character boxed up with a brown Character and a blue Character. That is, the red Character is boxed up with both a brown Character and with a blue Character. That's where the two Parcels come from in Card 5. However, the brown and blue Characters are being held together by the Paragon, and that bond can't just disappear without outside input. So, Card 1 says that a red Character is boxed up with a brown Character and a blue Character that are being held together by a Paragon, while Card 2 says that a red Character is boxed up with a brown Character and a blue Character, and that both sets are being held together by a Paragon.

Not buying it? Well, try it in normal logical language. Card 1 says, “Red and either brown or blue are true.” But that means either both red and brown are true, or both red and blue are true.” The logical justification is even easier to see if you start with Card 3. This says, “Red and ‘brown and blue’ are true.” But that means red is true and brown is true, while it also means red is true and blue is true. So red and brown are true and red and blue are true.

Example 2



This example was created by apply Distribution to Card 1. This produced Card 2. Then, we applied Distribution to Card 2, to produce Card 3.

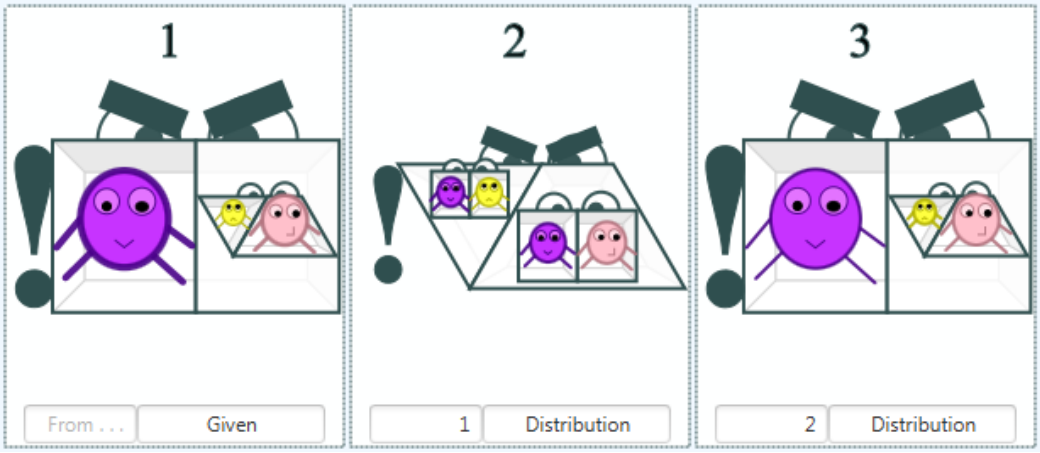
Here, once again, we have a Paragon with a Parcel in its right chamber. That means we can use the “Distribution” rule. So, we treat the Paragon like a window that has a Pyramid decal on its left pane, and zoom in on both the orange Character and the green Character in the Parcel. Thus, we end up with two Paragons, both with a Pyramid in the left chamber, but one with an orange Character in the right chamber, and the other with a green Character in the right chamber.

Now, imagine that Card 2 was the first Card we were given. How could we tell that we were allowed to use Distribution to move from Card 2 to Card 3? Well, we know that Distribution has to do with Paragons and Parcels, and in Card 2 we have two Paragons and a Parcel. Furthermore, we notice that the Chambergons in the left chambers of both Paragons match. Both Paragons have an orange/green Pyramid in their left chambers.

Even if Card 2 were actually the first Card in the whole game, therefore, the fact that the Parcel in Card 2 contains two Paragons having matching left chambers would clue us in that we are looking at the “zoomed in” result of a Distribution. Thus, we would realize that, if we wanted to, we could “zoom back out,” moving from Card 2 to Card 3. As we do this, the two Paragons, or windows, come into focus as one. This new single Paragon has an orange/green Pyramid in its

left chamber, while the orange Character and green Character are left in the Parcel. The orange-green Parcel now resides in the Paragon's right chamber.

Example 3



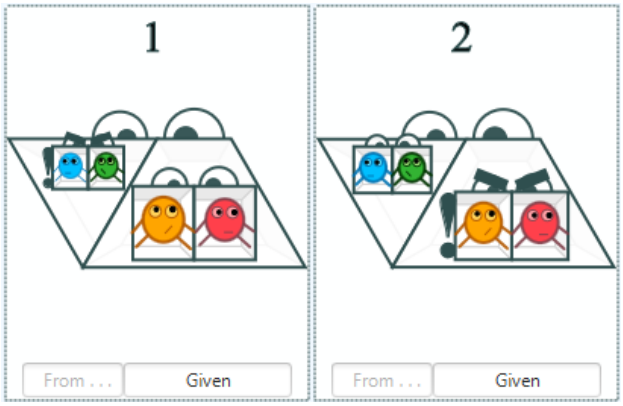
The main thing to notice here is the Club. It belongs to the Parcel, but doesn't travel with it when the Parcel splits in two and is swallowed by the Paragon. Rather, the Paragon takes it away from the Parcel. The reason for this is that the Club doesn't care what goes on "inside" the Chambergon to which it is attached. If that Chambergon wants to flip itself inside out, duplicate itself, and get consumed by the Paragon it contains, the Club is fine with this. In the eyes of the Club, nothing has really changed in the move from Card 1 to Card 2.

The thing to remember, then, is that with Distribution, if the outermost Chambergon has a Club, that Club stays with the outermost Chambergon, even if the outermost Chambergon changes from a Parcel to a Paragon, or from a Paragon to a Parcel. You see how the Club stays "outside," even when moving from Card 2 to Card 3? Yep. It just sits there, and lets everything else have all the fun.

Before we move on to the next example, I want you to take a look at Card 2 again, and imagine that there was no Card 1 before it. How would we know we could use Distribution on it? There are two tests to see if you can use Distribution. The first is if you have a Parcel or Paragon with a Paragon or Parcel (without a Club) in its right chamber. The second is if you have a Paragon or Parcel with a Parcel or Paragon in each chamber, and whatever is in the left chambers of the inner Parcels or Paragons matches.

In Card 2, we have a Paragon with a Parcel in each chamber. Furthermore, the contents of the left chambers of the two Parcels match. There is, specifically, a purple Character in both. We realize, therefore, that this is the zoomed in result of a Distribution, which we can undo by zooming back out again. When we do this, the two Parcels come into focus as one Parcel, with the purple Character in its left chamber, and what used to be the outside Paragon (in Card 2) in the right chamber.

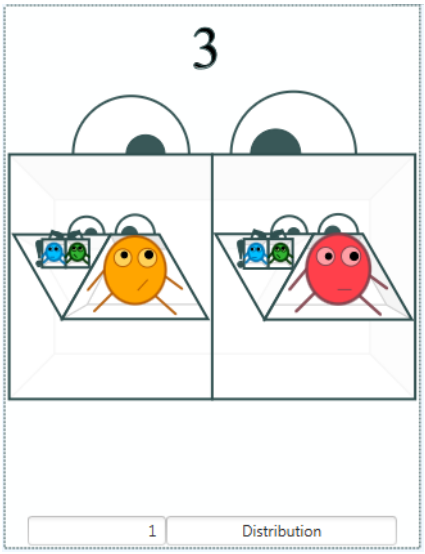
Example 4



In this example I want you to start by comparing Cards 1 and 2. Notice that both are identical except for where the Club is. In Card 1, the Club belongs to the Parcel in the left chamber, while in Card 2 it belongs to the Parcel in the Right Chamber.

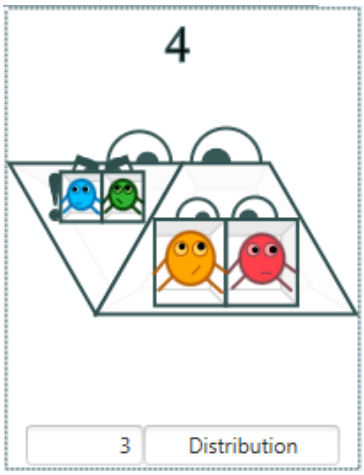
Now, ask yourself: Of these two Cards, which one can I perform Distribution on?

The answer to that question is below.



As you can see from this Card's "From" box, it is a Distribution of Card 1. We couldn't perform Distribution on Card 2 because the Club belonging to the rightmost Parcel meant that we couldn't open it up and stick the outer Paragon inside it. With Card 1, however, the Club belonged to the thing that was supposed to be doubled, rather than opened up. And that means the Club didn't get in the way.

In fact, if we look at Card 3, we see that it must be the result of a Distribution. It fits the bill. It is a Parcel or Paragon that contains two Parcels or Paragons (without Clubs), and the contents of the left chambers of those two Parcels or Paragons are identical. Given this, we know we can apply Distribution to "undo" what has been done. If we do so, we end up with Card 4:



With Card 4, things have come back into focus (as it were).

Chapter 7: The Level 7 Power(s) and Propositional Powers Summary

7.0: One Last Rule (and All the Others Too)

This is the final chapter in the first and fundamental part of the book. With it, we bring what logicians call "propositional" or "sentential" logic to a close. In addition to discussing the only power(s) specific to Pulsars (there's only one, but it feels like several), therefore, we are going to include an overview of all the other powers too.

Propositional logic analyzes propositions as wholes, and only deals with parts of propositions if those parts are themselves propositions (or negations). For propositional logic, therefore, there are only simple propositions ("Bob is kind," "Purple is a color," "Winter is coming," etc.), complex propositions (i.e., propositions that consist of other propositions, like "Bob is kind *and* purple is a color," and "Winter is coming *or* summer will last forever"), and negations of simple and complex propositions ("'Winter is coming' is false," "It is not the case that purple is a color and Bob is kind").

Starting with chapter 8, we will expand our world to include predicate logic. Before we can do that, however, we have to master Pulsars.

7.1: The Power of Biconditional Equivalence (A Recycling/Equivalence Rule)

There are actually two versions of this rule. The motto of the first is, "If you have a Pulsar, you can take it apart, and pack it up in a Package." To undo this version of the rule, you apply the following motto, "If you have two matching Pyramids in a Package, except that one is 'upside down', you can make them into a Pulsar."

This version of the rule is pretty straightforward. After all, a Pulsar is just two overlapping Pyramids. So, if you have a Pulsar, you can take it apart into its two constituent Pyramids, and then pack them away inside a Package. When you do this, you end up with two Pyramids that match, except that one is "upside down." That is, what is in its attic matches what is in the basement of the other, while what is in its basement matches what is in the attic of the other. So, if you ever find a Package containing two Pyramids, and they match except that one is upside down compared to the other, you can unpack them and put them back together into a Pulsar.

The motto of the second version of the rule is, "You can turn a Pulsar into a Paragon, if you throw in two Packages and two Clubs." This one is weirder. If you have a Pulsar, you can take whatever is in its upper chamber, and whatever is in its lower chamber, and repackage them together inside a Package. Then, you can do the same thing, except giving each an extra Club, and repackaging them inside a second Package. Then you put the two Packages in the two chambers of a Paragon. You can also undo this version of the rule, if you apply the following motto, "If you have a Paragon with two matching Packages, except that the contents of one

Package each has one more Club than the corresponding contents of the other Package, you can turn the Paragon into a Pulsar."

Restrictions: The "side" of Biconditional Equivalence that takes Pulsars apart has no restrictions. It can be performed on any Pulsar anywhere, even if the Pulsar has a Club. (The Club just stays with whatever shape is "outermost," like in Distribution.) However, the "side" of Biconditional Equivalence that puts Pulsars together has the following restrictions: (1) the Pyramids inside the Parcel must have identical contents, except that what is in the attic of one must be in the basement of the other, and vice versa, (2) if the Parcel containing the Pyramids has a Club, it must be given to the Pulsar, and (3) neither Pyramid can have a Club. Furthermore, (4) the Parcels inside the Paragon must have identical contents (whatever is in the left chamber of one must be identical to whatever is in the left chamber of the other, and the same for the right chambers) except that whatever is in the left chamber of one Parcel must have one more Club than whatever is in the left chamber of the other, and the same for whatever is in the right chamber (though the versions that have one more Club do not both have to be in the same Parcel), (5) if the Paragon has a Club, it must be given to the Pulsar, and (6) neither Parcel can have a Club.

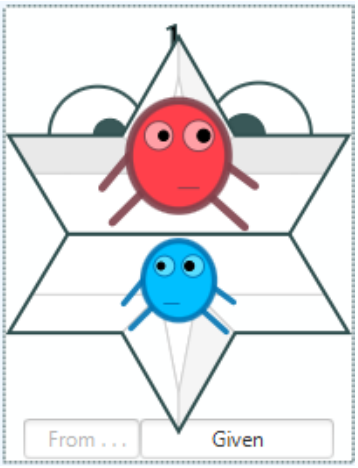
The logical justification of this rule is as follows. First, the version of the rule that allows you to turn Pulsars into Parcels with Pyramids is straightforward. Since a Pulsar is just two overlapping Pyramids, and whatever is in the basement of one is in the attic of the other, and vice versa (since one of the Pyramids is upside down), what we really have is two Pyramids "packaged" together that have identical, but reversed, contents.

In other words, a Pulsar says that whatever's in the attic depends on whatever's in the basement, and whatever's in the basement depends on whatever's in the attic. So, if we unpack that claim into its two parts, we have the attic depending on the basement (which gives us our right-side-up Pyramid) and the basement depending on the attic (which gives us our upside-down Pyramid). And since both Pyramids are legitimate ("true") we have to put them both together in a Parcel.

Second, the version of the rule that allows you to turn Pulsars into Paragons with Parcels is a little more roundabout. A Pulsar asserts the mutual dependence of two things upon each other. If such a claim is true—if the two things really do depend on each other—we would "activate" or "light up" the Pulsar. But we know by now that Pulsars only light up when the activation states of their attics and basements match. If both are activated, then the Pulsar lights up. If both are deactivated, then the Pulsar lights up.

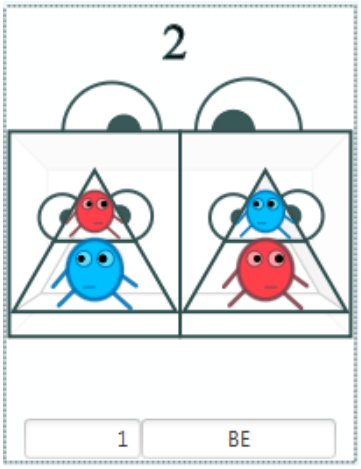
So, a Pulsar says that either both its contents are activated, or both are deactivated. To show that two things are activated (that both are true), we put them both in a Parcel. That's one of our two Parcels. To show that two things are deactivated, we negate each of them (adding a Club), and then stick both in a Parcel. That's the second of our two Parcels. Our problem is this: all we know is that the Pulsar as a whole is activated. But we don't know if this is because both its contents are activated or both are deactivated. One or the other is the case, but we don't know which. So, we put both our Parcels in a Paragon, since Paragons represent "or."

Example 1

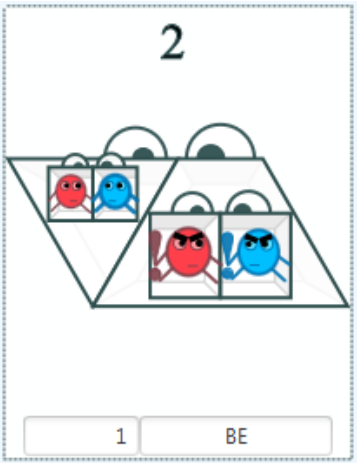


In Card 1, we are given the fact that red and blue are mutually-dependent. That is, if you have one, then you also have the other. We can apply Biconditional Equivalence to it to "unpack" it, but this could happen in one of two ways.

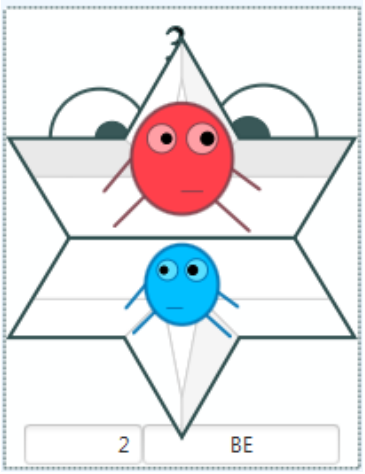
We can take apart the overlapping Pyramids, flip the upside-down one right-side-up, and pack them away in a Package.



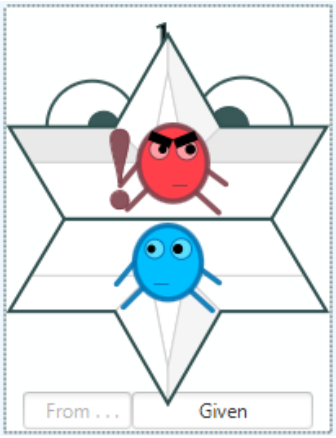
Or we can tilt the Pulsar counterclockwise and trim off its extra points, to form a Paragon. When we do so, however, the contents "resettle," with the content of the lower chamber of the Pulsar floating into the left chamber of the Paragon, but leaving behind a frightened "afterimage" or "ghost" of itself, and the content of the upper chamber of the Pulsar emitting a frightened "afterimage" or "ghost" of itself into the right chamber of the Paragon.



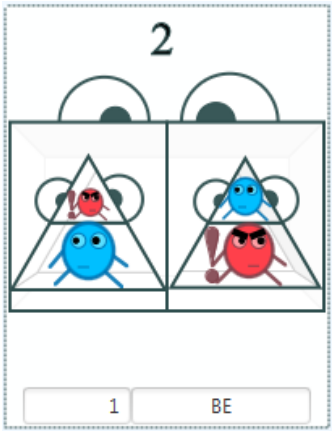
Given either version of Card 2, we can collapse things back into a Pulsar.



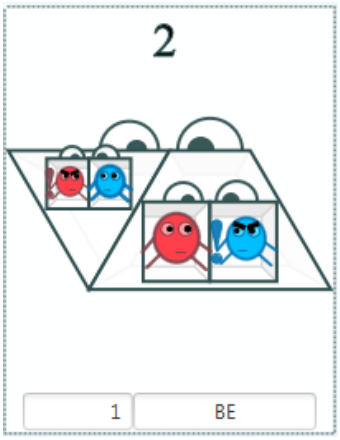
Example 2



Don’t let the Clubs fool you here. We simply have two Pyramids that have crashed, and when we take them apart, we see that the red Character with a Club is in the attic of one, but the basement of the other, and the same for the blue Character.



Alternatively, we can use BE to trim down and tilt the Pulsar into a Paragon.



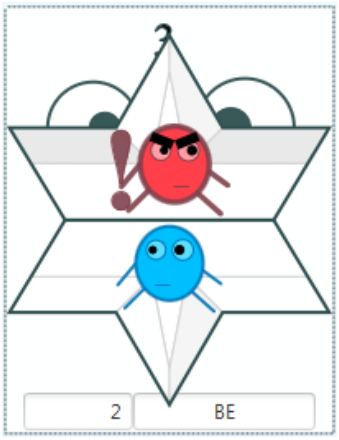
Notice that the Package in the left chamber of the Paragon (Card 2) contains identical copies of the contents of the Pulsar in Card 1, while the Package in the right chamber contains opposites of the contents of the Pulsar (that is, it contains the same contents, but with one more Club or one fewer Club). If we had started with Card 2, we could have used this fact (the fact that we have a Paragon containing Parcels whose contents are opposites to the contents of the other Parcel) to realize we can apply Biconditional Equivalence and create a Pulsar.

It would be easier to know how to reconstruct the Pulsar if we were given the first version of Card 2. The second version—the one with a Paragon—however, would be more confusing. Should the Pulsar have a red Character with a Club and a blue Character with no Club, or a red Character without a Club and a blue Character with a Club?

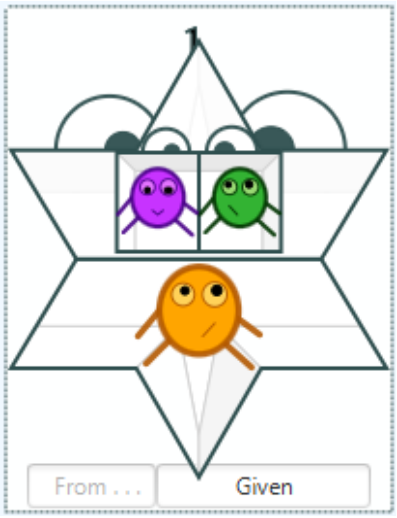
The answer is that it doesn't matter, logically speaking. If we put a red Character with a Club in the attic of our Pulsar, and a blue Character without a Club in the basement, that is fine. If we put a red Character with no Club in the attic and a blue Character with a Club in the basement, that is also fine. In fact, if we swapped basement and attic (putting the red Character in the basement and the blue Character in the attic), that would also be fine.

However, our convention in CBL will be to take the Package in the left chamber of the Paragon to represent the contents of the new Pulsar we construct. And thus we will effectively ignore the Package in the right chamber (since it simply contains “the opposites”).

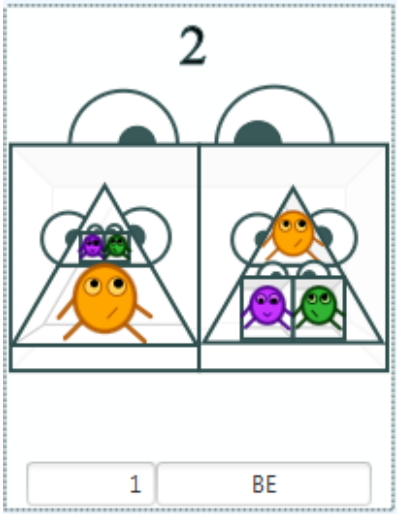
Therefore, whether we went with the first or second version of Card 2, we could apply BE again and produce Card 3:



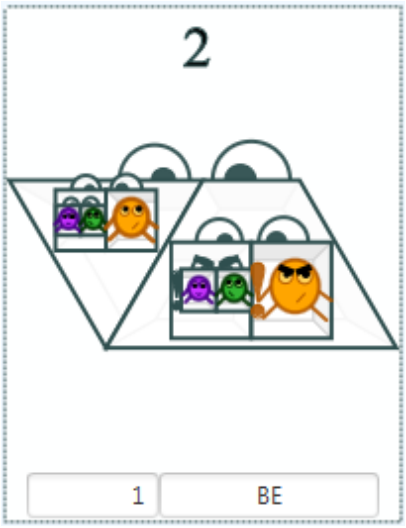
Example 3



Imagine we were given this Card to start a game. If having this Pulsar did us no good, we’d want to use Biconditional Equivalence on it to unpack it in some way. Our two options would be to unpack it into two Pyramids in a box . . . .

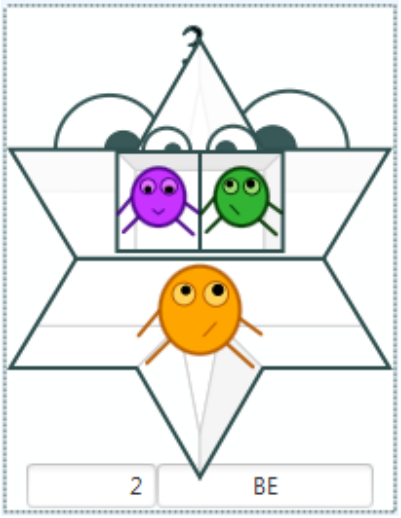


. . . or to unpack it into a two Parcels in a Paragon:



Our method for doing both would be the same as before. We would simply treat the Parcel in the attic of the Pulsar like we had been treating the red Characters in previous examples.

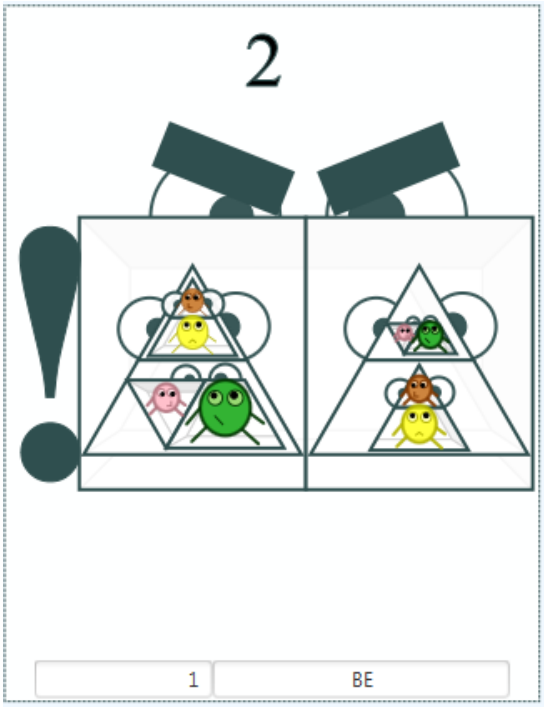
Furthermore, upon seeing either version of Card 2 above, we would know that we could "repack" things into a Pulsar, producing Card 3:



Example 4

In this example, we increase the complexity yet again. It is important, however, to remember two things. First, if we apply Biconditional Equivalence to Card 1, the Club will stay where it is, and become attached to the Parcel or Paragon that is outermost in Card 2. Second, the fact that the contents of the attic and basement of the Pulsar are both Chambergons is irrelevant. We simply treat them like we would Characters.

This means, therefore, that we can produce either of the following "Card 2s," using Biconditional Equivalence.



The two Pyramids in Card 2, above, contain the same contents as the Pulsar in Card 1, except the second of the two Pyramids has the contents reversed.

Notice, as before, that the two Parcels in this version of Biconditional Elimination end up containing the same things as the original Pulsar in their two chambers, except that the Parcel on the right has those contents with "opposite Clubs."

7.2: Summary of the All Propositional Logic Rules

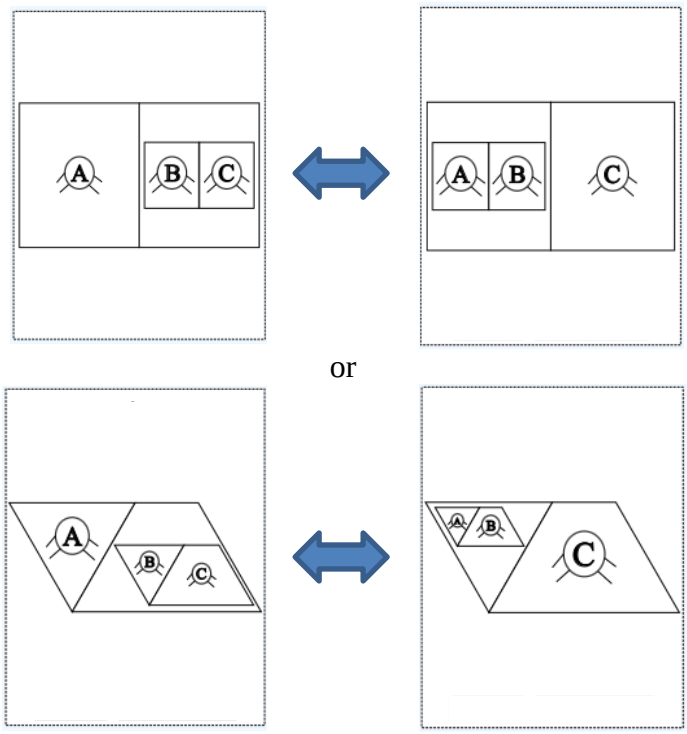
We have now seen the rules all the rules that apply to basic symbolic logic. There will be more rules to learn once we take up predicate logic in the following chapters, but for the moment we can summarize all the propositional logic powers.

In the following, the "" symbol indicates "recycling rules." This means:

(a) You can use the rule not only to change things, but also to change them back. (If you apply a recycling rule twice in a row, you can end up back where you began.)

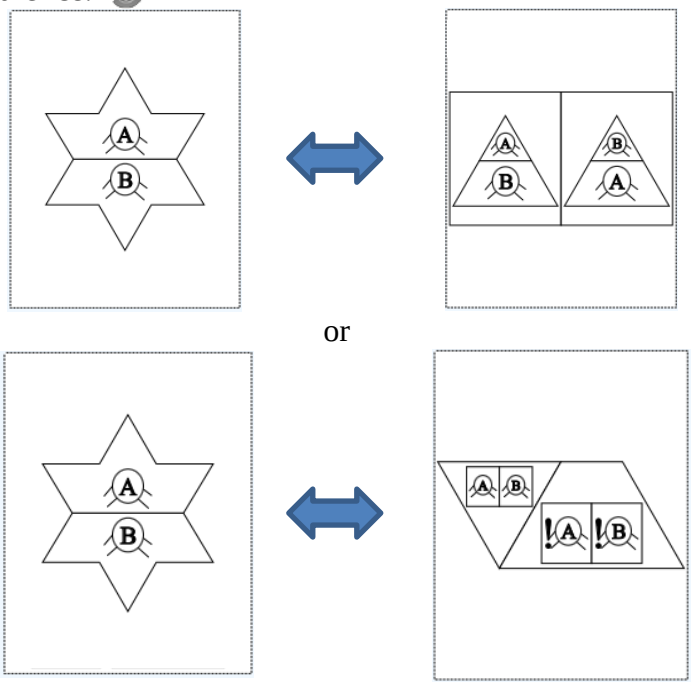
(b) You can use the rule not only on Chambergons/Characters that are the main ("outermost") occupants of their Cards, but also on Chambergons/Characters that are inside other Chambergons.

Association:



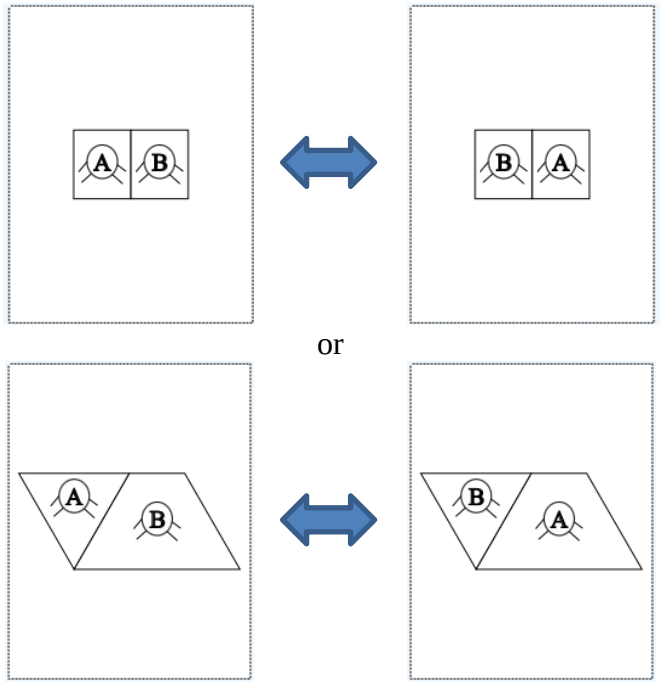
Note: this rule does not work if the inside Package or Paragon has a Club, but it does work if the outside Package or Paragon has a Club.

Biconditional Equivalence:



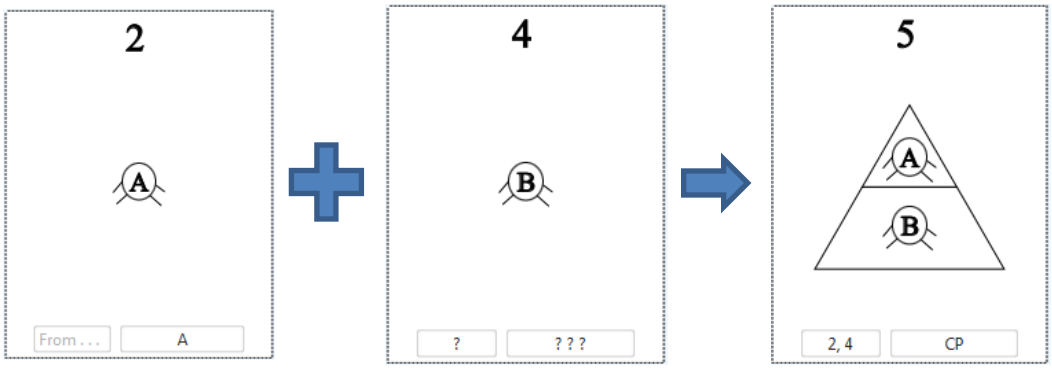
Note: this rule works even if the outermost shapes have Clubs,
but not if the inner Pyramids or Packages have Clubs.

Commutation:



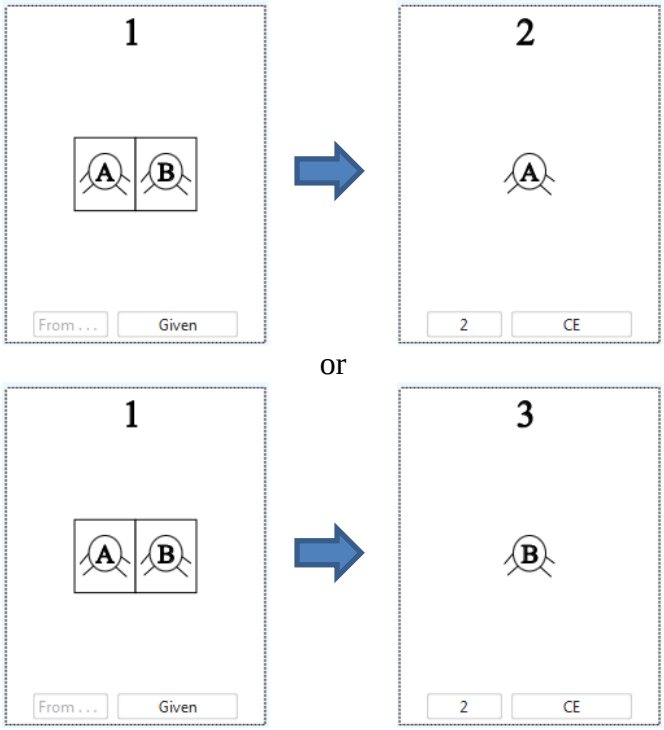
Note: this works whether or not the outside Package or Paragon has a Club,
and whether or not the contents of the Package/Paragon have Clubs.

Conditional Proof:



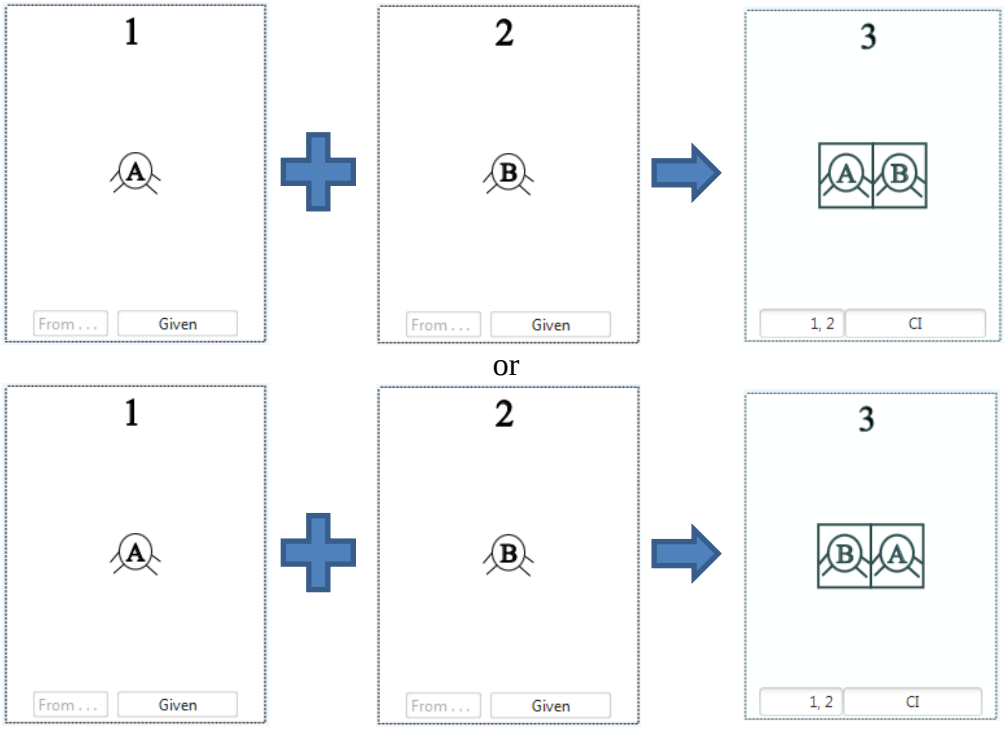
Note: this rule works even if contents of the loaner Card (Card 2, in the example above) and/or of the produced Card (Card 4, in the example above), have Clubs

Conjunction Elimination:



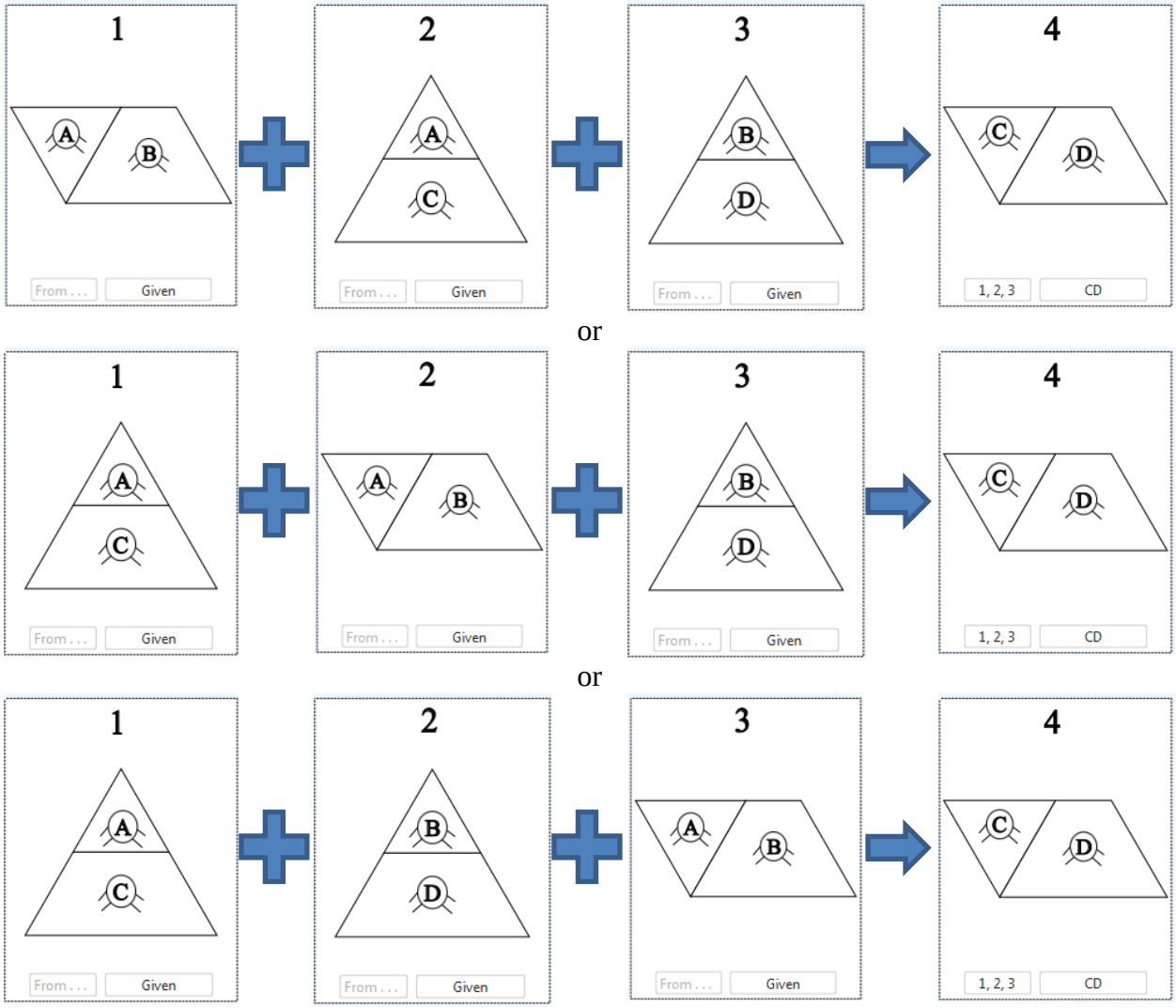
Note: this rule does not work if the outside Package has a Club.

Conjunction Introduction:



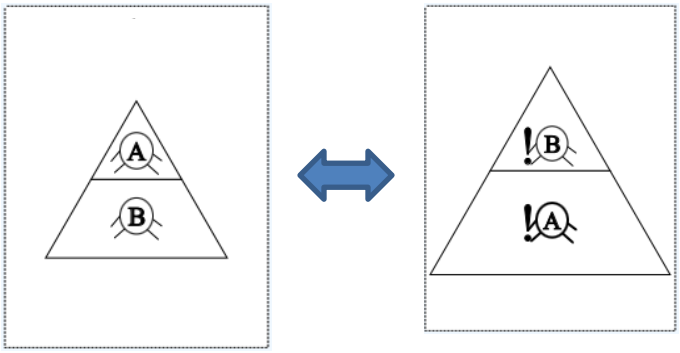
Note: this rule works even if the things you're packing up have Clubs.

Constructive Dilemma:



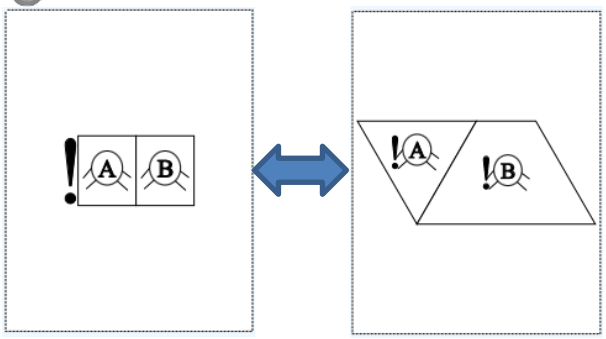
Note: this does not work if any of the Chambergons have Clubs.

Contraposition:

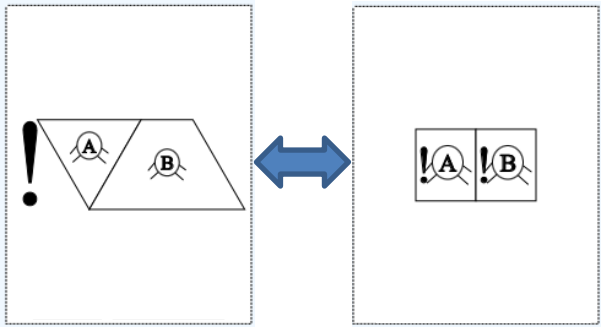


Note: this works even if the Pyramid has a Club.

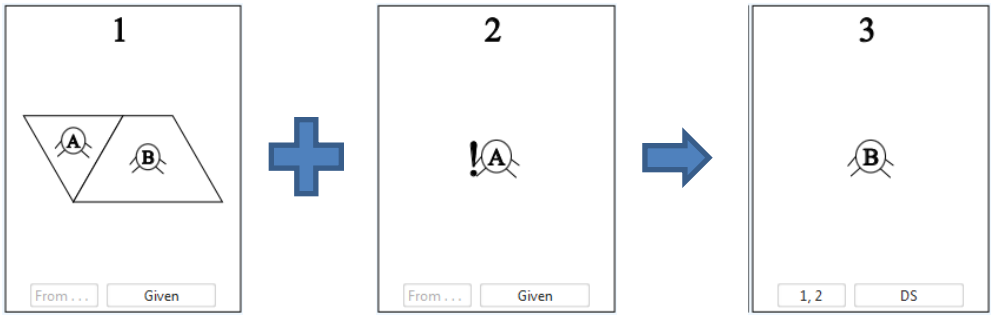
DeMorgan's Theorem:



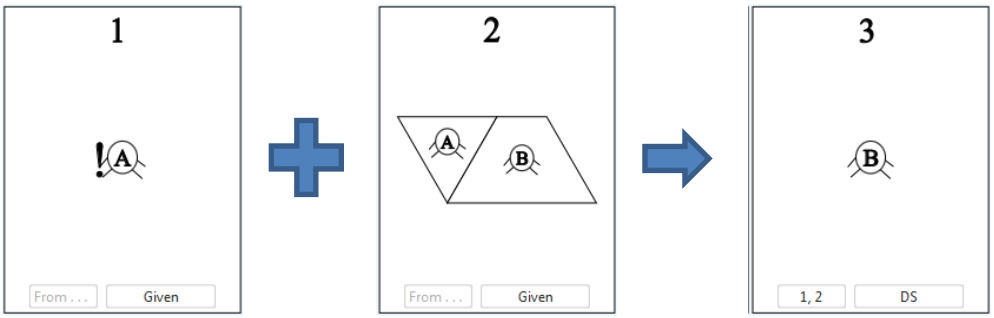
or



Disjunction Elimination:

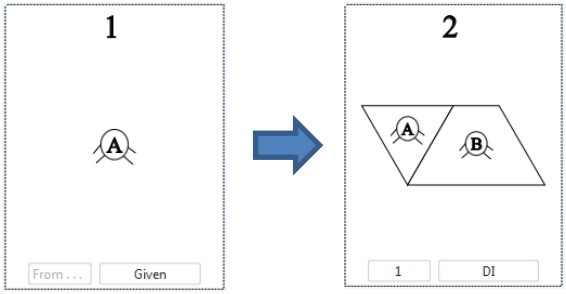


or



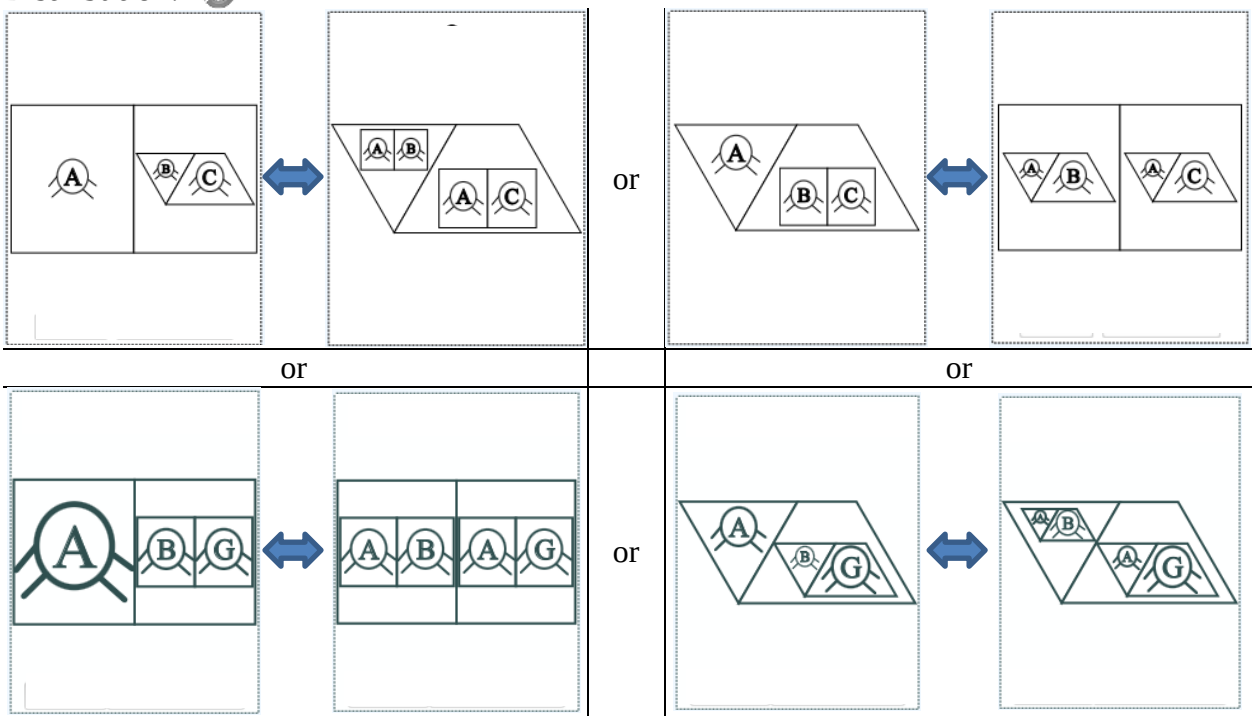
Note: this doesn't work if the Paragon has a Club.

Disjunction Introduction:



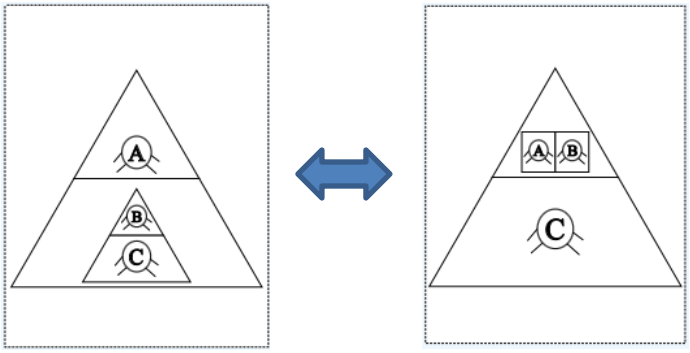
Note: This works, even if the thing to which you "add" the Paragon has a Club

Distribution:



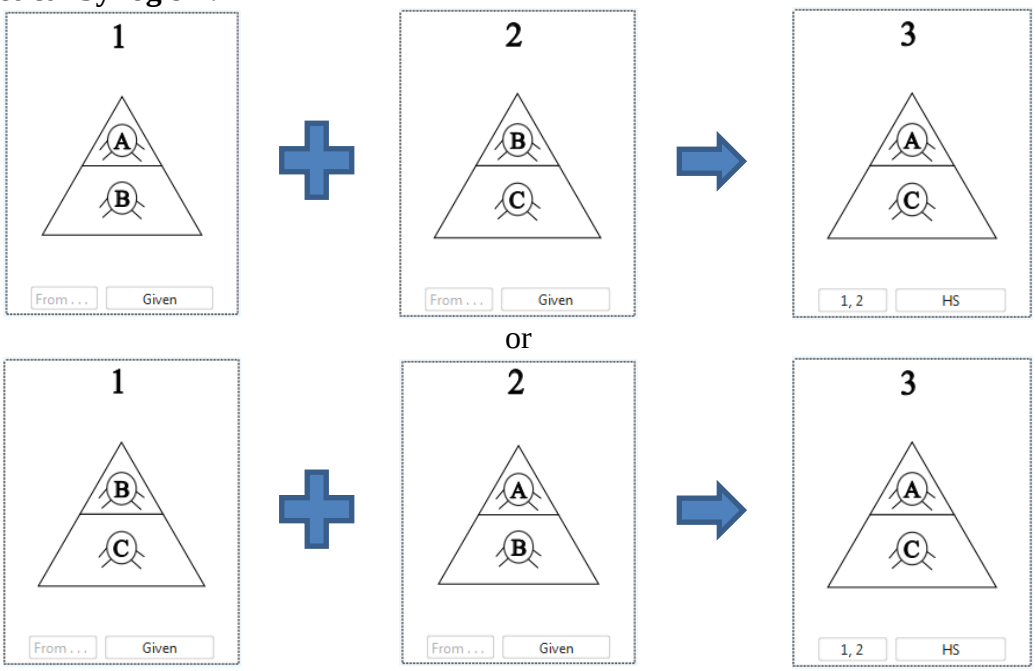
Note: this doesn't work if the inside Chambergons have Clubs, but if the outside Chambergons have Clubs, it does work. Those outside Club just "stay where they are."

Exportation:



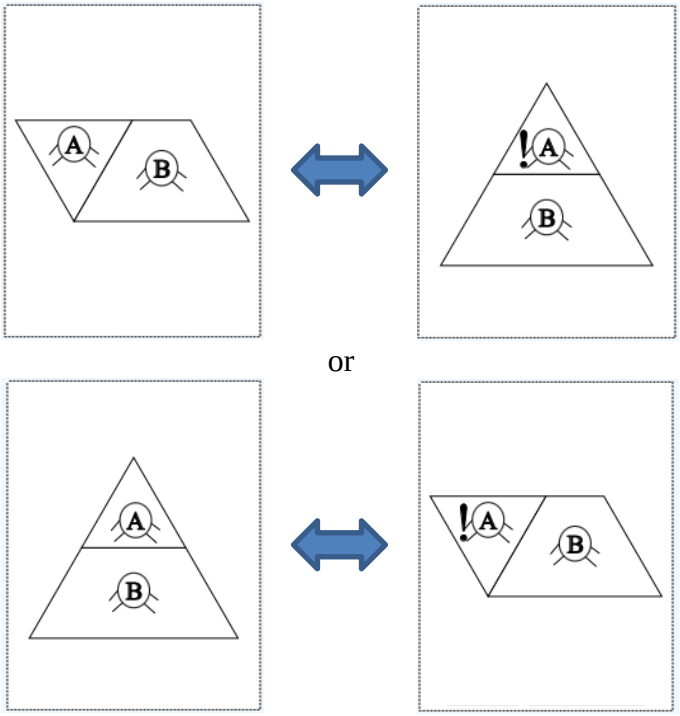
Note: this works even if the outermost Pyramid has a Club, but not if the inner Pyramid or the inner Package has a Club.

Hypothetical Syllogism:



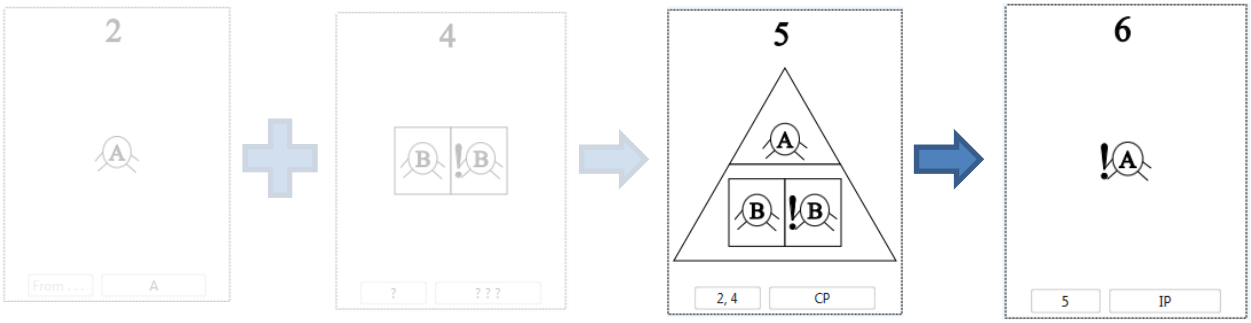
Note: this will not work if either of the first two Pyramids have Clubs.

Implication:



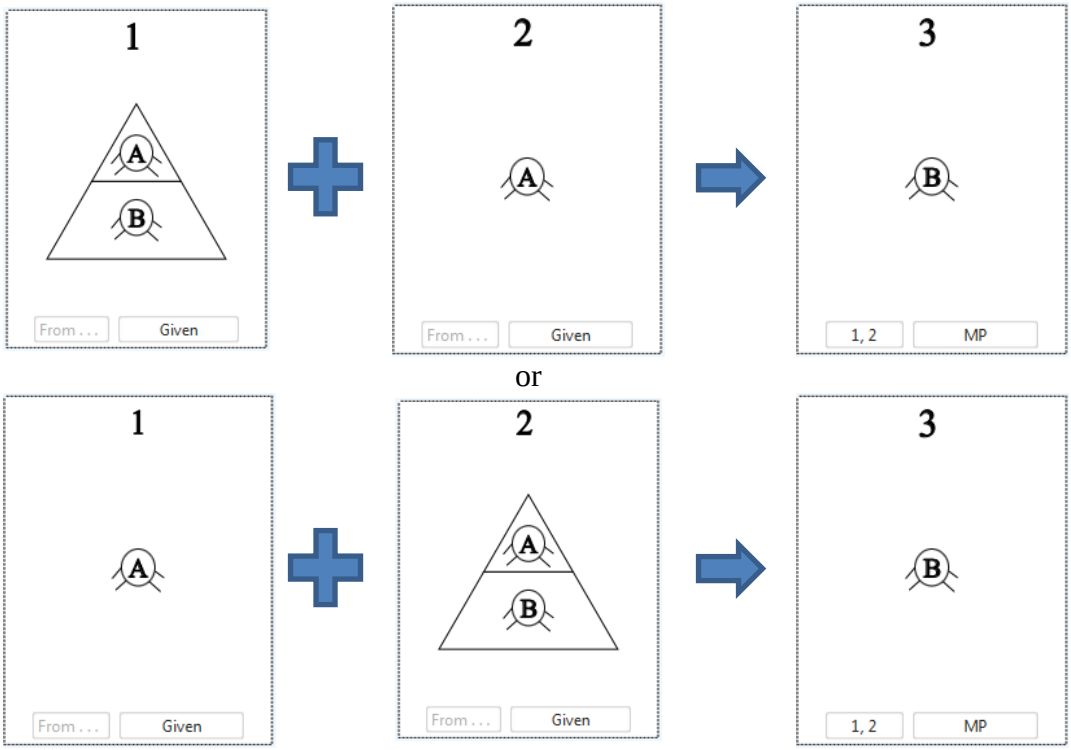
Note: this works even if the outside Chambergon has a Club.

Indirect Proof:



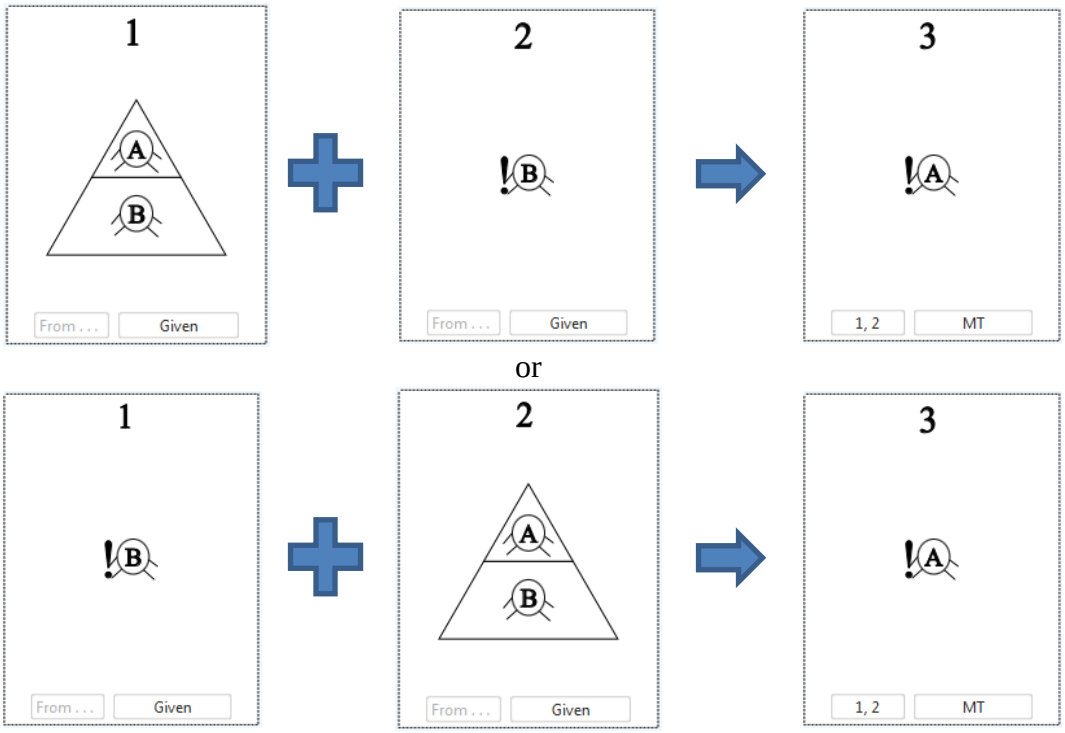
Note: this would work even if the Pyramid had a Club. That Club would just be added to whatever is in Card 6.

Modus Ponens:



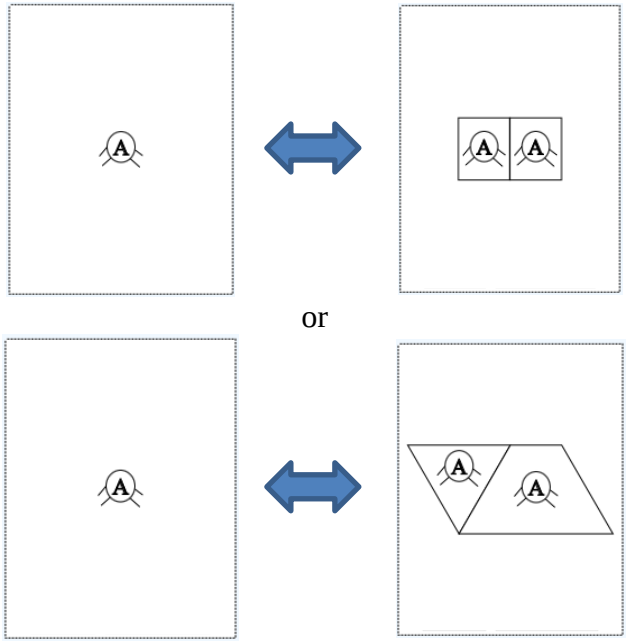
Note: this will not work if the Pyramid has a Club.

Modus Tollens:



Note: this will not work if the Pyramid has a Club.

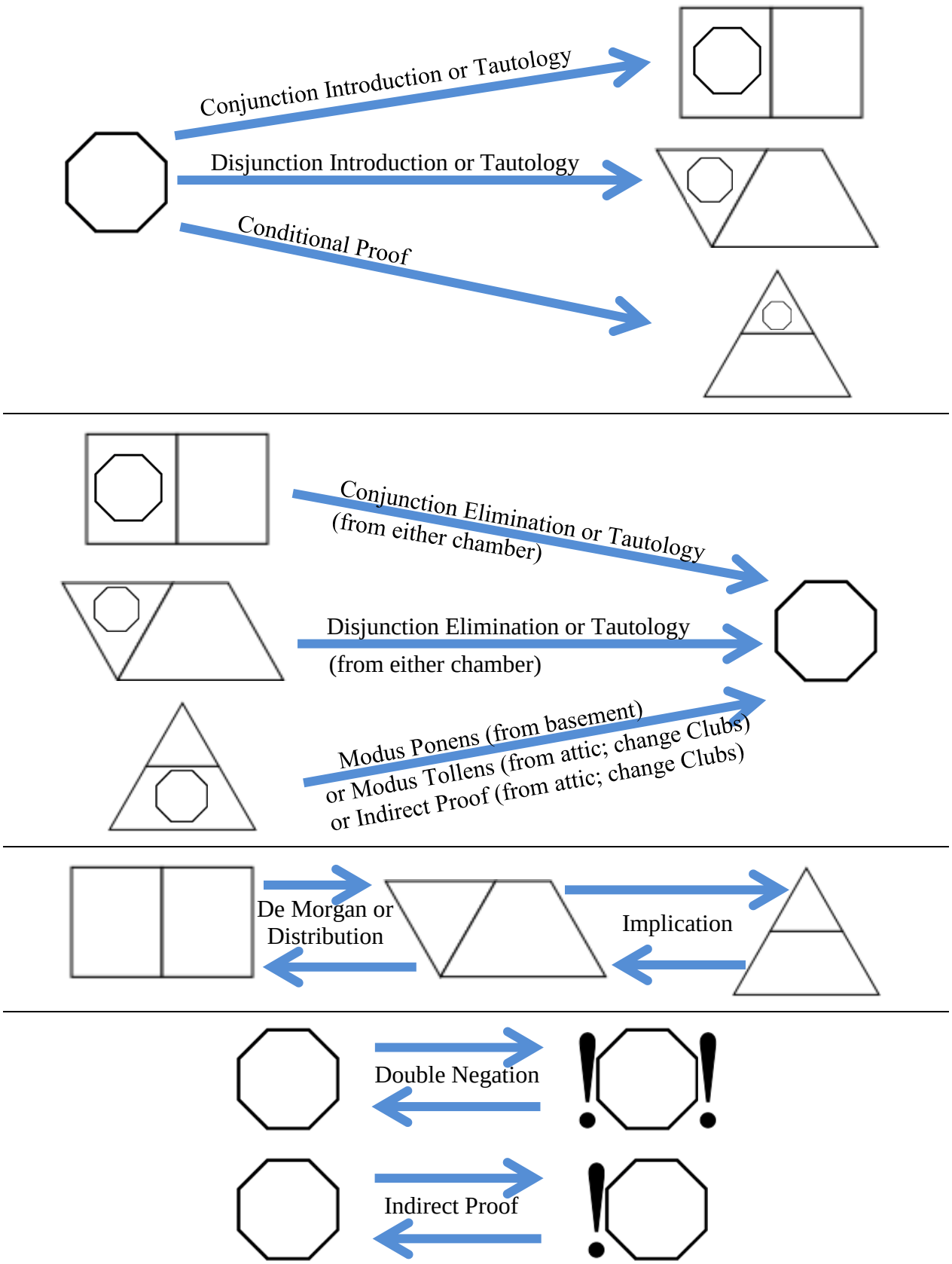
Tautology:



Note: this works even if there are Clubs involved.

7.3: Overview of the Main Rules

The summary above may be all you need as you move forward with the games. However, it might also be helpful to have at least the main rules presented in a more condensed form. In what follows, we will use an octagon to represent ("Stop Sign") any Character or Chambergon. Wherever you see an octagon, you can insert any Character or Chambergon you wish, and with any number of Clubs.


Conjunction Introduction or Tautology
Disjunction Introduction or Tautology
Conditional Proof
Conjunction Elimination or Tautology
(from either chamber)
Disjunction Elimination or Tautology
(from either chamber)
Modus Ponens (from basement)
or Modus Tollens (from attic; change Clubs)
or Indirect Proof (from attic; change Clubs)
De Morgan or
Distribution
Implication
Double Negation
Indirect Proof

7.4: Rules Abbreviations Complete

In the "Using" box of each Card, you can either write out the name of the rule, or use an abbreviation. The program accepts any number of different variations on each (and the program doesn't care about capitalization), but if it is not accepting the rule you are citing, double check your spelling.

Addition . . . . . Add [Same as Disjunction Introduction]
Association . . . . . Assoc
Assumed . . . . . A
Assumption . . . . . [Same as Assumed]
Biconditional Equivalence . . BE
Commutation . . . . . Comm
Conditional Proof . . . . . CP
Conjunction . . . . . .Conj [Same as Conjunction Introduction]
Conjunction Elimination . . . .CE, && Elim, & Elim, &&E, &E, v Intro, vI, etc.
Conjunction Introduction . . . CI, && Intro, & Intro, &&I, &I, ^ Intro, ^I, etc.
Constructive Dilemma . . . . . CD
Contraposition . . . . . .Contra
De Morgan's Theorem . . . . . DeM, DM
Disjunction Elimination . . . . DE, || Elim, | Elim, ||E, |E, v Elim, vE, etc.
Disjunction Introduction . . . .DI, || Intro, | Intro, ||I, |I, v Intro, vI, etc.
Disjunctive Syllogism . . . . . .DS [Same as Disjunction Elimination]
Distribution . . . . . .Dist, Dis
Double Negation . . . . . .DN
Exportation . . . . . Exp
Hypothetical Syllogism . . . . HS
Implication . . . . . Impl, Imp
Indirect Proof . . . . . IP
Modus Ponens . . . . . .MP
Modus Tollens . . . . . MT
Simplification . . . . . Simpl, Simp [Same as Conjunction Elimination]
Tautology . . . . . Taut
Transposition . . . . . .Trans [Same as Contraposition]

Chapter 8: Predicate Logic and Level 8 Powers

8.0: New Pieces

8.0.0: Entertainment for Characters

Propositional logic is not enough.

"What's propositional logic?" you ask.

"Propositional logic is what we've been doing the whole semester, up to this point."

"Oh," you respond. "Well then, I disagree. Propositional logic is *plenty*."

"Ah, you see, but that's where you're wrong," I respond.

"That's not an argument," you say. "You can't just say I'm wrong. You have to give me reasons why I'm wrong. Monty Python has a whole sketch about that. I demand premises for your conclusion! I need 'Given' Cards if you want me to reach the Goal, 'Propositional logic is not enough'."

I beam with pride. You've clearly got this logic thing down pat. "Okay then," I say, "Here's my reason: There are arguments we can't analyze with propositional logic, because when we do, they end up looking invalid."

You look puzzled. "Well, *are* they invalid? It's only a problem if they look invalid when they aren't, right?"

"Right," I respond. "Take the following argument, for example."

1. Americans are great.
2. Bob is American.
∴ 3. Bob is great.

Here's what that would look like in propositional logic.

1.	A	Given
2.	B	Given
∴ 3.	G	1, 2 ???

And here's what that would look like in CBL.

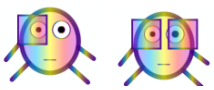
And that doesn't make any sense. The argument is valid. If the premises were true, the conclusion would have to be true. So, to handle arguments like the one above, we have to add five more Pieces to our game.

"But, I don't care about all that stuff. I just want to play games and earn points and LEVEL UP," you say.

Well, okay then. Think of it like this: the Characters that we've been playing with have lives. And that means they need entertainment. Now, it just so happens that the favorite entertainment of Characters is going to see 3D movies in their local theatres. We can already represent their calmness and anger and fear and rage, but if we are going to represent this "happiness and entertainment" side of their life, we will need icons for theatres and 3D Glasses.

And speaking of Glasses, some Characters are naturally farsighted or shortsighted, and that means they need normal eyeglasses too. (Well, "normal" in some sense of the word. Some of them wear Monocles, which don't really count as normal anymore.)

8.0.1: Square Lenses/Monocles/Glasses



Above you see what Characters look like if they're wearing square-lensed Monocles or Glasses. These Lenses, like everything else in Chambergon Battle Logic, stand for a particular linguistic or logical thing.

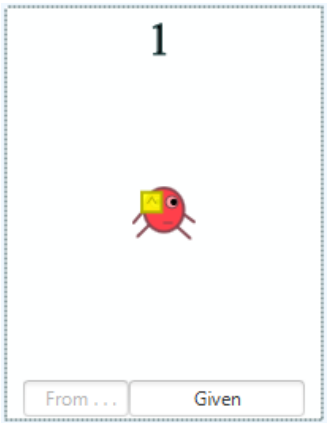
"Ugh! Not this logic stuff again!" you complain.

Hush your face. Parcels stand for conjunctions, and conjunction is usually represented by the word "and." Clubs stand for negation, and negation is usually represented by the word "not." Characters stand for whole propositions. And so on.

Because you are used to this "CBL things stand for linguistic and logical things" rule by now, you will be pleased and relieved to discover that it holds true for square Lenses too. Square

Lenses, you see, stand for general terms like “thing” or “object” or “person.” Furthermore, when a Character is wearing one or more Lenses, it ceases to represent an entire proposition and begins to represent the predicate that applies to the “thing,” “object,” or “person” represented by the Lens (or Lenses) it is wearing.

Here’s an example: A red Character with no Lenses usually represents the letter “R,” which might represent the simple proposition, “I am running.” A red Character with a square yellow Monocle, however, would represent “R(y),” with “R” representing the predicate, “is running,” and “y” representing the general subject, “something.”



There is a convention in symbolic logic that we use letters from the end of the alphabet (x, y, and z) to represent general terms, just like in algebra. Our way of following that convention in Chambergon Battle Logic will be to use *colors* “from the end of the alphabet (yellow [“y”] and pink [“z”]) plus colors from “the second half of the alphabet” (purple [“p”] and red [“r”]) for the sake of variety. We’ll usually save the colors from the beginning of the alphabet for circular Lenses.

Furthermore, Characters can wear different Lenses of different colors at once. For example:

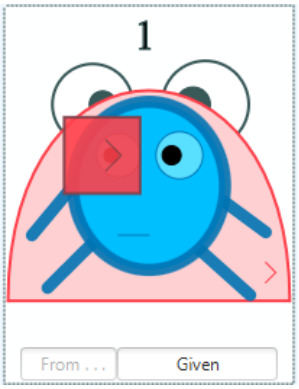
In a case like this, the Character represents a predicate that ties two general terms together. For instance, in this case the orange character might represent the predicate “is organizing,” while the blue square Lens might represent “someone,” and the red square lens might represent “an event.” (Thus, the orange Character plus its two lenses might represent, “Someone is organizing an event.”)

Whenever I see Characters wearing blue and red square Glasses like the one above,[27] I can’t help but see them as having on the old-fashioned 3D movie glasses theaters used to use back in the day. (If you don’t remember them, Google “3D Glasses paper.”) So, if you would like, you can

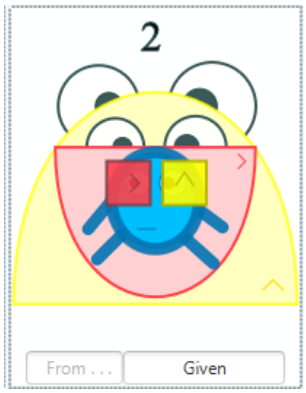
[27] This, of course, breaks the convention that square Lenses should come from the end of the alphabet. “Blue” is pretty clearly from the beginning.

ignore all that stuff about square Lenses representing general terms, and think of them like this: When a Character is wearing one or two square Lenses, it is actually 3D movie Glasses from a theater.

Now, at most theaters, the Glasses don't actually belong to the person wearing them. The theater owns the glasses and you have to give them back when you leave. For example, the following Character is wearing a 3D Monocle (I know, that makes no sense at all) that belongs to the theater she/he/it is in. It is red, and the "theater"—which is actually a "Dome"—is red.



Life can get complicated for Characters, however, once they've taken to wearing Monocles and Glasses. For example:



This guy is inside two theaters: one yellow and one red. Think of it like going to the movie theater (building), but being "in theater 13" (the room housing the 13th screen in the theater). You're in two different "theaters" at the same time, as it were. If a friend texted you and asked, "Where are you? I'm at the theater," he might just mean he's standing in the lobby of the building. To clarify the situation, you might respond, "Got seats in theater 13. You still in the lobby?" And your friend might respond, "Theater 13? Thought were meeting at Regal." And you might respond, "Theater 13 INSIDE Regal."

So, anyway, you can think of the guy above like that. The yellow "Dome" might represent the entire theater building, while the red "Bowl" represents the particular theater inside the building that the Character is in. What's weird about the situation above, however, is that one of the

lenses of the Character's 3D glasses belongs to the theater building as a whole (it is yellow), while the other belongs to the individual theater inside the building (it is red).

"Ugh, Tillman. Not more complications," you roar. "Why would a theater do that?"

The answer, fortunately, is rather simple. Imagine that you pay to see a 3D showing of *Toy Story 5*, but sneak into the IMAX screen that is showing *Doomtastic Explosion Time* instead. If you could use the same 3D glasses for both movies, this would make cheating way too easy. So, the theater company needs the glasses it hands out to work only for the movies they're supposed to be used for.

The problem is that it would cost too much money if the theater had to manufacture different types of 3D Glasses for each of its 25 screens. So, it splits the difference. Since each pair of Glasses needs two Lenses, and those Lenses have to be different anyway, it makes one Lens in each pair the same for all the Glasses, and the other Lens in each pair specific to the individual screen it's supposed to be used for.

"You just made all that up," you accuse. And you're right. I made it all up. We're playing a game here. We need a backstory. And backstories are allowed to be fiction.

"Fine," you say, grumpily.

"And speaking of fictional movie theaters," I continue, "the two lenses a Character wears don't necessarily have to be different colors. They can look identical. They can even be the same color as the Character itself."

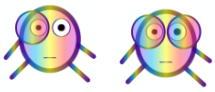
"ROOOOOOOOAAAAR," you roar once again.

Think of these as being like the newer 3D movie glasses that are clear. In those glasses, the two lenses look identical, even though they are actually polarized differently (if I understand correctly).

"But what if I'm colorblind?" you ask, even though you aren't.

Well, each Lens, Dome, and Bowl contains an arrow (actually, just an arrowhead), that points in one of eight directions (up, down, left, right, and the four diagonals). The directions correspond to the eight available colors, and are the same directions for the same colors. So, if the arrow in a Lens is pointing in the same direction as the arrow in a Bowl, then the Lens and Bowl are the same color. And if a Lens's arrow is pointing in the same direction a Character is looking, the Lens and Character have the same color.

8.0.2: Circular Lenses/Monocles/Glasses

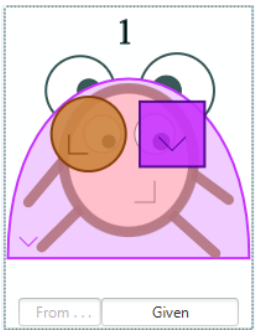


Above, you see what a Character wearing a circular Monocle (on the left) or full-blown pair of Glasses (on the right) looks like. While square Lenses represent general terms like “something,” or “someone,” circular Monocles and Glasses represent names for particular, known objects/things/people. While a square red Lens might represent “some person,” a circular red Lens might represent “Robert.” While a yellow square Lens might represent “some car or other,” a yellow circular Lens would represent, “this specific car right here.”

However, there is a convention in symbolic logic that we use letters from the beginning of the alphabet (a, b, and c) to stand for names or labels for specific objects. We’ll follow that convention in Chambergon Battle Logic by using *colors* “from the beginning of the alphabet” (brown [“a”] and blue [“b”]), with colors from roughly the first half of the alphabet (green [“g”] and orange [“o”]) thrown in for variety.

So, square Lenses are general terms, while circular Lenses are particular lables or names. However, all of that is kind of abstract and boring. While playing the game, just remember that Monocles and Glasses with *circular* Lenses *don’t* belong to movie theaters. They are the everyday streetwear of Characters. They have colors, just like the square Lenses, but those colors don’t tell you which Bowl or Dome they belong to.

For example, in the following Card, the square purple Lens belongs to the purple Dome, but the circular brown Lens does not. This is, first and foremost, because the circular Lens is not even the same color as the Dome, but it is also because the circular Lens is circular, and circular Lenses never belong to theaters (Domes/Bowls).



Notice that in addition to having colors, circular Lenses also have arrows in them that point in the same direction that a Character would be looking if the Character had the same color. The blue arrows in circular Lenses point left, for example, just like blue Characters look left. So, if you’re ever playing on a screen that isn’t displaying colors correctly (or if you’re colorblind), pay attention to the arrows.

8.0.3: Coins

The piece above is called a “Coin,” because it is meant to look like two coins stored side-by-side in a display card.

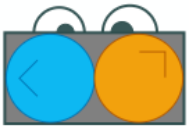
“Why not call it ‘Coins’, then, if there are two?” you might ask.

“Here’s why,” I respond, while doing a little jig of delight.

The problem with the display books, cards, or sheets that coin collectors use to store and display their collection is that some are opaque. If you slip a coin into the storage/display slot, you end up only being able to see one face of the coin.

A solution to this problem is to purchase two copies of each coin, and store/display them side-by-side—one with the front face of the coin turned outward, and the other with the back face turned outward. That’s what the “Coin” icon above is supposed to look like: two copies of the same coin displayed side-by-side, with the front of the coin visible on the left, and the back of the coin visible on the right.

The Coin above, however, is neutral. In actual games, Coins will look more like this:



The Coin above is blue on one side, and orange on the other. Both sides are being shown, side-by-side, but it’s the same Coin. Blue and orange are just two sides of the same thing.

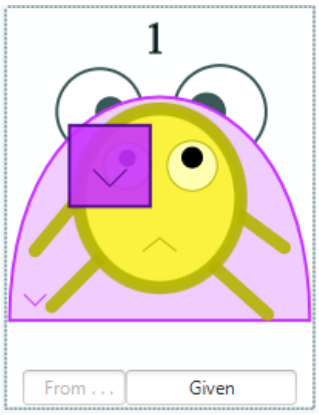
As with Characters, Lenses, Domes, and Bowls, Coins use directions and pointing to indicate their color. So, if one face of a Coin has its arrowhead pointing in the same direction as the arrowhead in a Lens, you know that face of the Coin is the same color as that Lens. Similarly, if the arrowhead on one face of a Coin is pointing in a different direction from that in which a Character is looking, the two must have different colors.

The colors on a Coin don’t matter to Characters, however. They only matter to lenses. They say that Lenses of one color would be another color if you just “turned them over.”

8.0.4: Domes and Bowls

The shape above is a Dome. Domes represent the word “all” or “every.” They are what logicians call “universal quantifiers,” since (a) they tell you “how many,” and (b) that how many is “all of

them." For example, the following Dome says that "all" or "every" square purple Lens belongs to a yellow Character.



However, you don't really have to remember all that while you are playing, any more than you have to remember what "$(3^3 \times 3^2) / 3^4$" actually means while you're solving an equation. (You just have to remember that when you multiply things with exponents, you add the exponents if the two things you're multiplying match—and when you divide things with exponents, you subtract the exponents if the two things you're dividing match.) Similarly, you can think of Domes as being "all-encompassing," like large movie theaters—the kind that belong to big national chains and have 20 different screens on multiple levels.

In contrast with Domes, there are Bowls.

That is what a neutral Bowl looks like. In actual games, however, Domes and Bowls tend to have the same colors as square Lenses—colors "from the end of the alphabet" (like yellow ["y"] and pink ["z"]), or at least from the second half of the alphabet (like purple ["p"] and red ["r"]). Bowls represent the phrase "there exist some." For example, the following Bowl says there exist some square red Lenses that are worn by purple Characters.

Bowls are also quantifiers, like Domes, but aren't universal. (Notice that Bowls are just upsidedown Domes, and vice versa.) Bowls are what logicians call "existential quantifiers." Instead of telling you that everything has a particular property, they tell you that something exists

that has a particular property. (After all, Bowls are smaller than Domes, so if Domes say "all," Bowls must say "some.")

But, once again, you don't really have to remember all that while playing the game. Instead, you can think of Bowls as being like smaller arthouse theaters, or like specific screens/theaters within a theater building.

Any Character or Chambergon can go inside a Dome or Bowl, and Domes and Bowls can go inside any Chambergon (even other Domes or Bowls). However, the main function of Domes and Bowls is to own/lend out square Monocles or Glasses.

8.0.5: The Logic of Lenses, Domes, and Bowls

Since other things can go inside Domes and Bowls, they are more like Chambergons than anything else. Domes and Bowls, in other words, are operators like "and," "or," and "if."

They differ from Chambergons, however, in two ways. First, they have only one "chamber," rather than two. This makes them more like Clubs than like Chambergons. Second, they have colors. This makes them more like Characters than like Chambergons.

However, it is helpful to think of the Character vs. Chambergon distinction as being parallel to the Lens vs. Dome/Bowl distinction. Chambergons are pointless without Characters to fill them. Likewise, Domes and Bowls are pointless without Lenses—being worn by Characters—inside them.

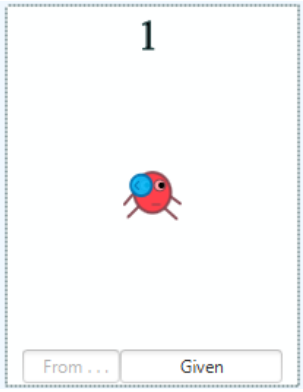
In propositional logic, furthermore, Characters are "atomic." You can't split them up or put anything inside them. In predicate logic, Characters can contain things (i.e., Lenses) and it is now the Lenses that are atomic. In fact, once Lenses and Domes/Bowls are introduced, Characters cease to represent only propositions. They now represent simple propositions if they have no Lenses, while representing predicates if they have one or more Lenses.

Let me give you an example. I want to translate the following argument into Chambergon Battle Logic.

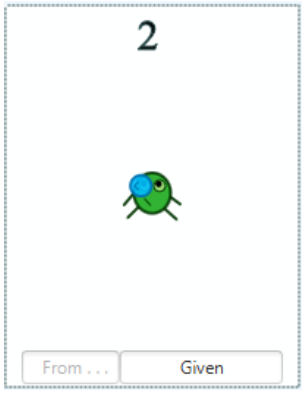
1. Bob is a Radical Reformer.
2. Bob is a Green (a member of the Green Party).
3. If a person is both a Radical Reformer and a Green, that person is from Yakutsk.

∴ 4. Bob is from Yakutsk.

I would symbolize the first line as "R(b)," with "R" standing for the predicate "is a Radical Reformer," and "b" standing for "Bob." "R(b)" would like this in CBL:

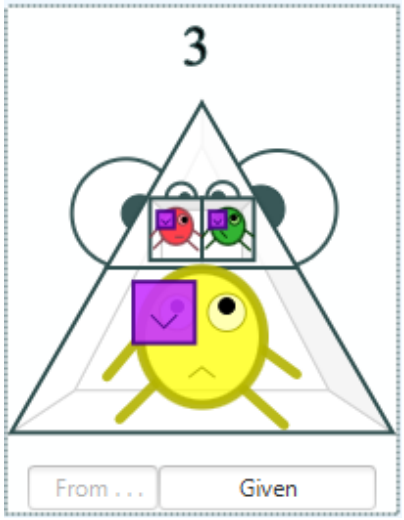


I would symbolize line 2 as “G(b),” with “G” standing for the predicate, “is a Green,” and “b” standing for “Bob” (once again). “G(b)” would look like this in CBL:



The third line—“If a person is both a Radical Reformer and a Green, then that person is from Yakutsk”—is more complicated. Let’s use “p” for the general term “person.” The “both/and” part of the proposition will be, “(R(p) && G(p)).” The “that person is from Yakutsk” part will be “Y(p).” And the “if/then” part of the proposition will be “(->).” So, when we put that all together, we get this: “((R(p) && G(p)) -> Y(p)).”

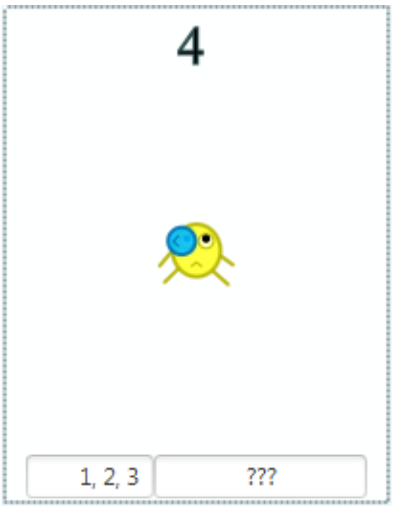
In CBL, “((R(p) && G(p)) -> Y(p)),” looks like this:



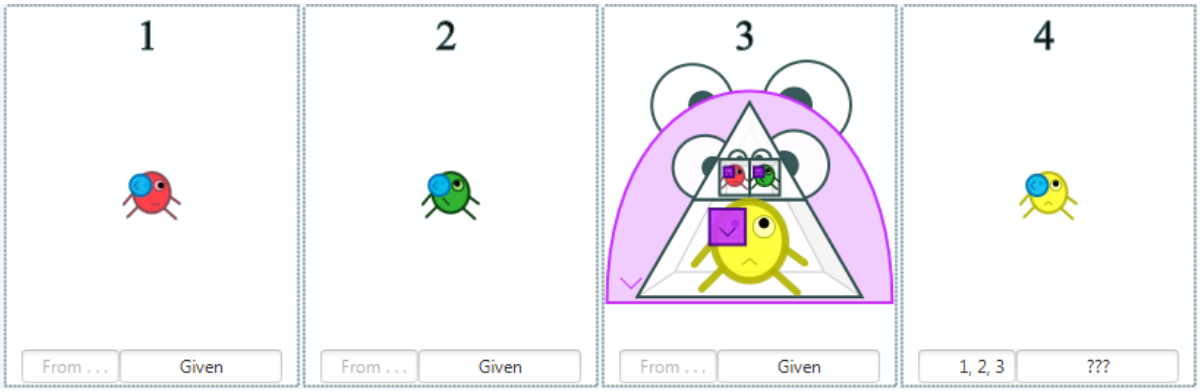
The reason we use square Lenses instead of circular Lenses is that "person" is a general term, not an actual name. But since "person" is a general term, we have to show whether we mean "all persons" or only "some persons." In the argument above, "all persons" is clearly meant. So, we have to put the Pyramid above in a Dome.

(Notice that the Dome is purple, to show that the purple Lenses belong to it.)

The fourth and final line of the argument—"Bob is from Yakutsk"—would be "Y(b)," which looks like this:



So, the whole argument looks like this:



All we need now is to know what rule to use in Card 4. We can put the red and green Characters together in a Parcel, and that would kind of match the attic of the Pyramid in Card 3. So, we could kind of pull a Modus Ponens. However, the Dome is in the way. So, we first have to get the Pyramid out of the Dome somehow. And even if we got the Pyramid out, Modus Ponens would get us a yellow Character with a square purple Lens, not a yellow Character with a circular blue Lens. So, we have to figure out how to change the shape and color of Lenses.

Our problem, then, is that we have a clearly valid argument, but we don't yet know the rules that would be necessary for proving that the argument is valid. We're closer to being able to fully analyze this kind of argument than we were back on p. 257 (when we were only using propositional logic), but we've got to learn some predicate logic laws before we can finish the job.

8.1: The Power of Quantifier Negation (A Recycling/Equivalence Rule)

Quantifier Negation ("QN") allows you to change Domes to Bowls, and Bowls to Domes. The only qualification is that if you "flip" the Dome or Bowl, you must also "flip" its number of Clubs, as well as "flipping" the number of Clubs of whatever Piece it contains.

Restrictions: None. This is a Recycling Rule, and so it can be used on Bowls and Domes that are the outermost Pieces in their Cards, as well as on Bowls and Domes that are inside other Chambergons.

The logical justification for this rule is rather simple. A Dome is a universal quantifier, so it says that everything has some property. But if everything has some property, then nothing exists that does not have that property.

"What?" you ask.

For example, if I say, "Everyone is looking happy today," that's the same thing as saying, "There isn't anyone who looks unhappy," or "There does not exist a single person who is not looking happy." Similarly, if I say, "Someone here is a genius," that's the same as saying, "It's not true that everyone here is an idiot," or, rather, "It is not the case, for everyone here, that each person here is not a genius."

"What?" you ask.

Well, look at it like this. You're a Dome or a Bowl, then some punk comes along and flips you upside down. Are you upset? Of course you are. You don't like being upside down. So you get angry—which means you gain a Club (just like in Implication or Contraposition). Furthermore, imagine that you are inside a Dome or Bowl when it gets flipped upside down. Your whole world has just been turned upside down. Are you going to be happy about this? No! You'll be scared out of your mind. And when you get scared, you get angry—which means you gain a Club.

However, if you were angry about your situation in the first place, the flipping the whole world upside down might just make you happy. In other words, if you have a Club already, you'll lose it if the Dome or Bowl you're in is flipped. Similarly, if a Dome or Bowl already has a Club, that means it's miserable. It considers itself to be in some way in the wrong position. You can make it happy(er) by raising or lowering it a level, just like anything else, but you can also make it happy(er) by changing its position in another way: flipping back "right-side-up."

Example 1a

Imagine we were given the Card above, and wanted—for some reason—to apply Quantifier Negation to it. Well, Domes are quantifiers, so we should be able to use QN on Card 1. However, we'd have to remember that Quantifier Negation not only negates quantifiers, but negates their contents as well.

So, apply QN to Card 1 would produce this:

The Dome has been flipped upside down, and is therefore angry. Furthermore, the Character's world has been turned upside down, and it is no longer happy either. So, both gain a Club.

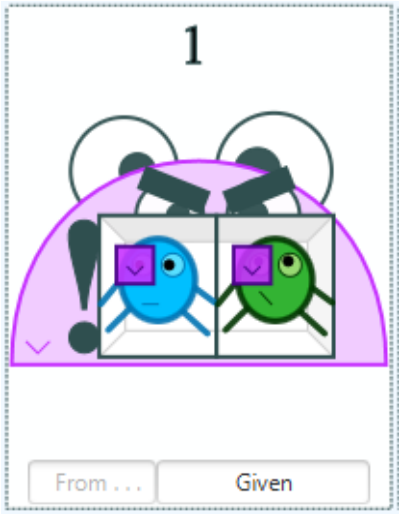
Now, imagine that all the anger we have created in Card 2 makes us feel guilty. So, remembering that Quantifier Negation is a recycling rule, we apply it again to undo our mistake. That produces this:

Quantifier Negation works by flipping the quantifier (the Dome or Bowl), and then "flipping" the Clubs inside and outside. So, since the Bowl in Card 2 has a Club, it now has zero Clubs. And since the Character in Card 2 has a Club, it now has zero Clubs as well. Everyone is happy because the world has been put right again.

Example 1b

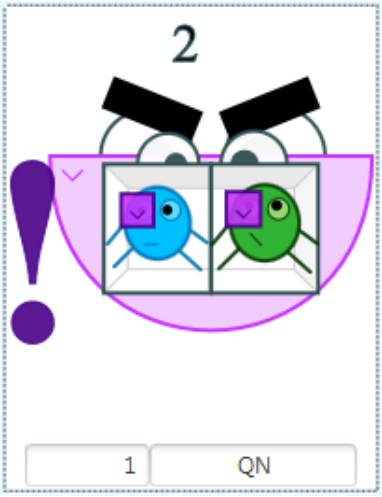
Example 1b is the same as Example 1a, except that we start with a Bowl. Everything is happy until we flip the Bowl upside down, using Quantifier Negation. This results in a Club for the Bowl and a Club for the Character. Things are then restored in Card 3, using Quantifier Negation again.

Example 2a



Unlike Example 1, we begin Example 2 with one Club "inside" the Dome. That is, the piece immediately inside the Dome—which is a Parcel, in this example—has a Club. However, the Dome still has no Club. So, what would happen if we applied Quantifier Negation?

The answer is this:

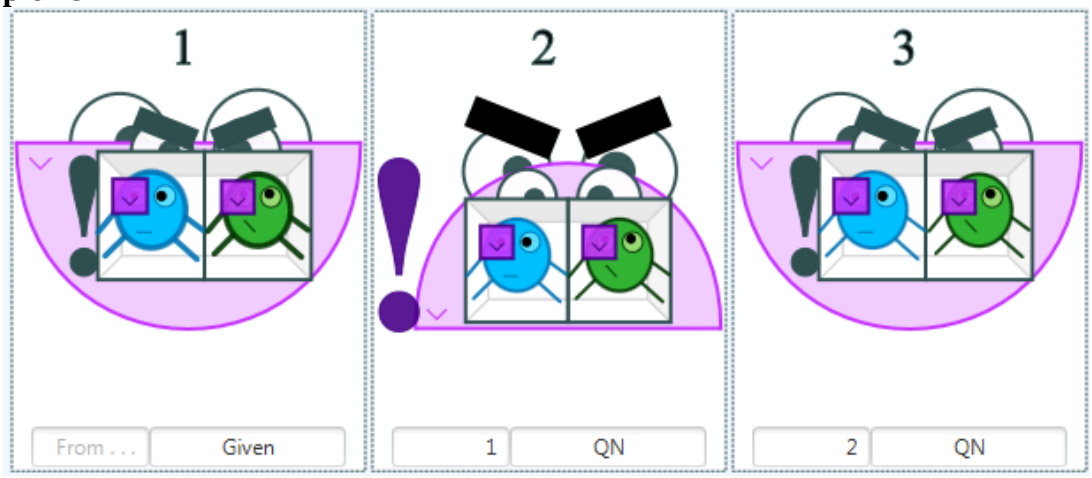


From the Parcel's point of view, it's world has been put back right again. So, it loses its Club. However, from the Dome's point of view, it has itself been turned upside down (since it is now a Bowl). So, it gains a Club.

In this example, we can't make both the Dome/Bowl and the Parcel happy. But if we'd rather things at least look happy on the outside, we can apply QN again. If we did so, we would end up with this:

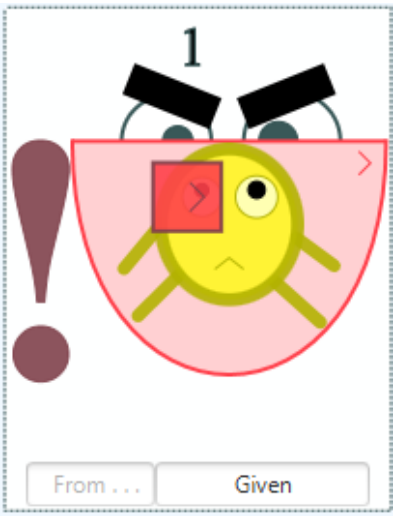
Notice how similar Example 2a is to an DeMorgan's Theorem. Instead of moving the Club further into the Parcel, however, we pull it out of the Dome, using Quantifier Negation. This flips the Dome into a Bowl, and leaves the Club outside with the Bowl. Then, using Quantifier Negation again, we move the Club back into the Bowl, which flips it into a Dome once more.

Example 2b



This example is the same as the previous, except that we start with a Bowl, flip it into a Dome, then flip it back into a Bowl. Notice in both examples that the Club simply moves from belonging to the Parcel to belonging to the outer Dome or Bowl. The things inside the Package are unaffected by Quantifier Negation.

Example 3a

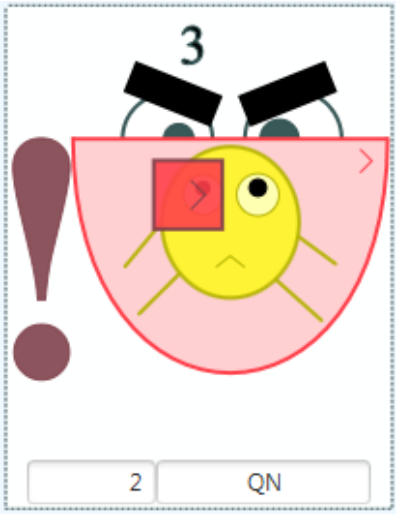


This example is much like the last. There is only one Club. However, this time, the Club starts on the outside. What would happen, therefore, if we applied QN?

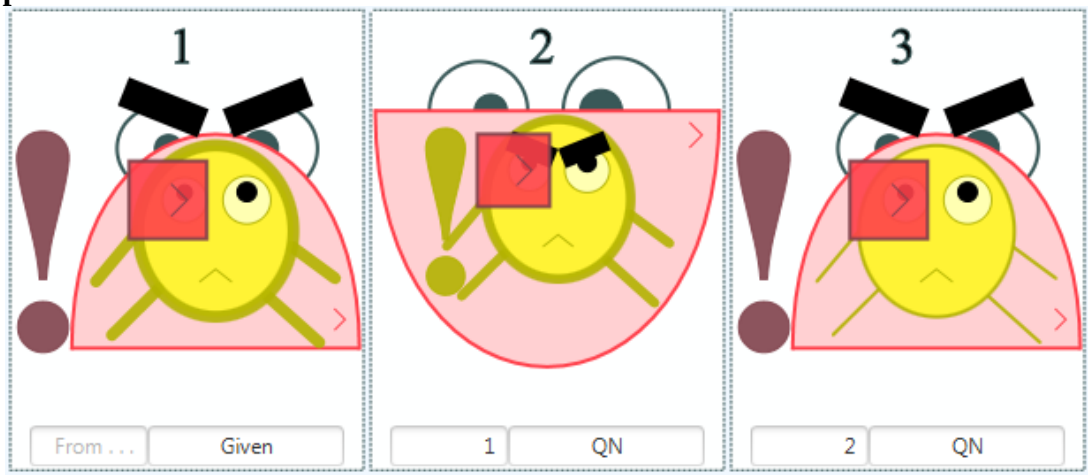
The answer is this:

The Bowl is now happy, since it's "back to being a Dome," which it evidently had wanted. The Character, however, is now unhappy, since its world has been turned upside down.

If we don't think Domes are real people, therefore, we might prefer to let the Character be the happy one. If we do, we can simply apply QN again, to produce this:



Example 3b



This example is the same as the previous. It just shows that you can start with a Dome that has a Club if you want to use Quantifier Negation, just as well as you can start with a Bowl with a Club.

Example 4a

Okay, here's a weird situation. This time, the Dome we might want to "quantifier negate" is inside a Pyramid, and that Pyramid has a Club. The question we have to ask ourselves is whether Quantifier Negation is still possible in a situation like this.

The answer, fortunately, is "yes." Quantifier Negation is a recycling rule, and recycling rules can be used on anything anywhere, even if what they're inside of has a Club. That's because recycling rules just rearrange things, and you can rearrange the insides of something without opening it. (You just have to give it a good shake, as it were.)

So, imagine we apply Quantifier Negation to Card 1, above. What would that produce?

The answer is this:

Notice that the Dome is now happy, being a Bowl rather than a Dome. And the Pyramid inside the Dome is now inside a Bowl, and that means its world is now "right-way up." Notice furthermore that the green Character in the basement of the inner Pyramid still has its Club. It

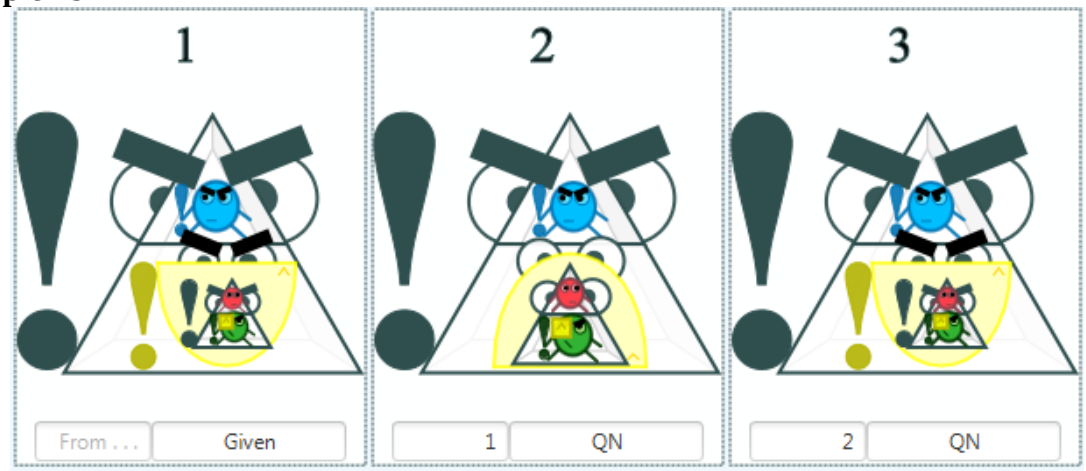
has no idea what's going on outside the Pyramid. It's still angry about having to be in the basement of a Pyramid.

So, remember that QN can be applied to Domes and Bowls anywhere, so long as you "flip" the Clubs of the Dome/Bowl and of whatever is *immediately* inside the Dome/Bowl. Anything that is inside whatever is inside the Dome/Bowl gets left alone.

Imagine, for example, that we want to apply Quantifier Negation again, this time using it on Card 2. If we did that, we'd produce the following:

Notice that the Bowl is miserable, since it has to be a Dome again, and the Pyramid is angry again, since it's world is once more upside down. However, notice that the Characters inside the Pyramid have neither gained nor lost Clubs. As far as they're concerned, their world (the Pyramid) has not changed.

Example 4b



Here we have the same example, but starting with a Bowl instead of a Dome. This is to emphasize the point that Bowls and Domes are essentially the same—each is just the "flipside" of the other, with Clubs adjusted.

8.2: The Power of Universal Elimination (A Landfill/Inference Rule)

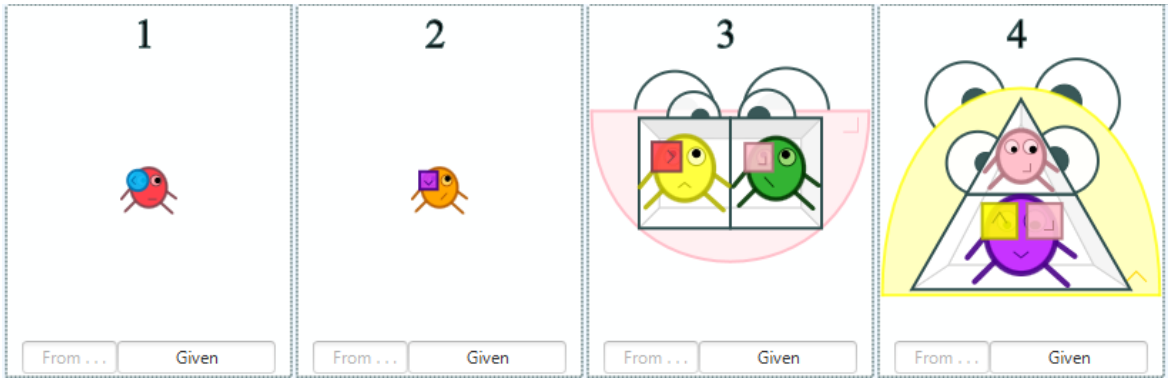
This is the rule for getting rid of Domes, and thus to free the Characters and Chambergons within them to roam about in the Wild. It will help us to deal with it if we think about two different situations in which it might be applied: (1) with only a Character inside the Dome, and (2) with a Chambergon, and hence multiple Characters, inside. However, before we get to those two situations specifically, we need to think more generally about the relationship between Lenses and Domes/Bowls.

8.2.0: Free and Bound Lenses

We need to adopt two special new terms. The first is term "'free' in a Card." Let us stipulate that if a Lens is not inside any Dome or Bowl in a particular Card, then it is "free in that Card." In fact, let us stipulate that even if a Lens is inside a Dome or Bowl, it is still free (in the Card it is in) if it has a *different* color from all the Domes or Bowls it is inside of. After all, if you steal a Lens or pair of Glasses from one theater and take them into another, the new theater doesn't control them, and they are outside the old theater that used to own them.

So, Lenses are free if they are not in a Dome or Bowl, and if they are in a Dome or Bowl, but are not the same color as that Dome/Bowl.

Take the following Cards as examples.



In Cards 1 and 2, the Lenses are not inside any Dome or Bowl, and thus are free.

In Card 3, the green Character's square pink Lens is bound by the pink Bowl, but the yellow Character's square red Lens is free. It is not inside a Bowl that matches its color.

In Card 4, the pink Lens is in a Dome, so it could be bound. However, the pink Lens is not the same color as the Dome it inside. So, the pink Lens is free.

The yellow Lens in Card 4 (like the pink Lens in Card 3), above, however, is not free. It is "bound" in that Card. It is bound because it is inside a Dome or Bowl that is the same color.

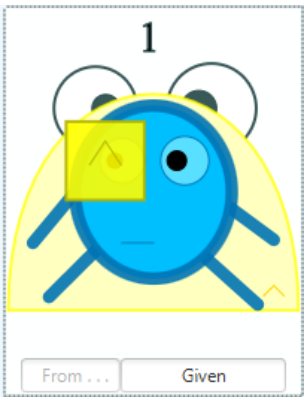
Think of a Lens as being bound when it belongs to one of the Domes or Bowls it is inside of, and think of it as belonging to a Dome or Bowl it is inside of when it has the same color as that Dome or Bowl.

8.2.1: Domes with a Character Inside

If a Dome (a) has no Clubs, (b) is the outermost Piece in some Card, and (c) contains a Character, look for any Lenses that Dome is binding. You can get rid of the Dome using Universal Elimination and *in the same move* you can change both the shape and the color of any Lens you free.

"What do you mean, 'freeing a Lens'?" you ask.

Well, if a Lens is bound by a Dome or Bowl, and you then get rid of the Dome or Bowl, the Lens isn't bound any more. So, by getting rid of the Dome or Bowl, you have freed it. Here's an example.

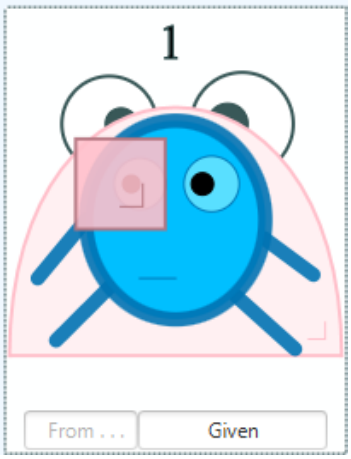


In the Card above, the square yellow Lens is bound by the yellow Dome. Now, imagine that we eliminate the Dome, using Universal Elimination. That would produce Card 2.

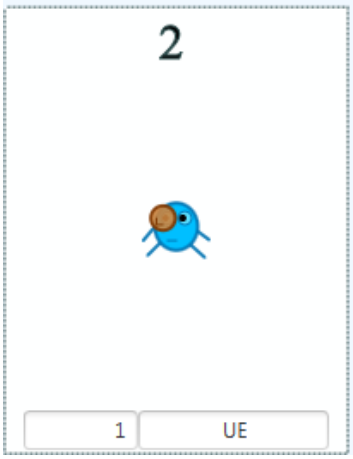
The yellow square Lens from Card 1 has been "freed" by Universal Elimination (removing the Dome), and is now free in Card 2.

Now, when you use Universal Elimination to remove a Dome and free a square Lens, *you can morph any Lens you free* it into a circular Lens that has a new color. (If possible, pick a color from the beginning of the alphabet—brown [“A”] or blue [“B”]—or green [“G”] or orange [“O”].)

For example, in Card 1 below, the square pink Lens is bound by the pink Dome.



If we use Universal Elimination to remove the Dome from Card 1, we would free the square pink Lens. And when you free a Lens from a Dome using Universal Elimination, you get to change both the shape and the color of the Lens. For example:



Now the Lens is circular, not square, and brown, not yellow. The act of removing the Dome and changing the Lens were the same: they’re both part of “Universal Elimination.”

“That’s a pretty powerful rule, if it lets you do two things at once,” you comment.

“True dat,” I respond. “It’s like De Morgan, which allows you both to move Clubs and to change shapes.”

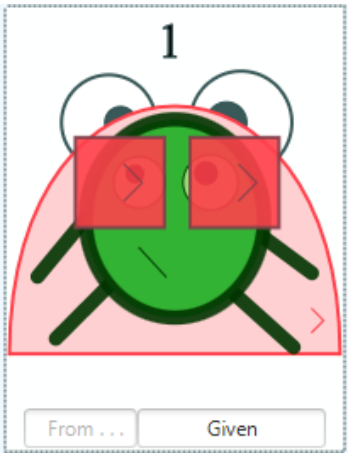
“Good point,” you acknowledge. “But just because I *can* do something doesn’t mean I should, or even that I want to. So, why would I want to change the Lens while removing a Dome?”

Well, if you got caught walking around outside the theater with their 3D glasses on, you’d get in trouble. But if you change their shape and color, no one will notice. And, fortunately for you, Domes represent national chains; thus, the technology for altering their proprietary 3D glasses to make them look normal is widespread. So, if you know someone who owns an eyewear specialty shop, it’s highly likely you’ll be able to get her or him to do alterations for you.
“I’m like a super-criminal now,” you exclaim with glee. “I have my own underground network.”

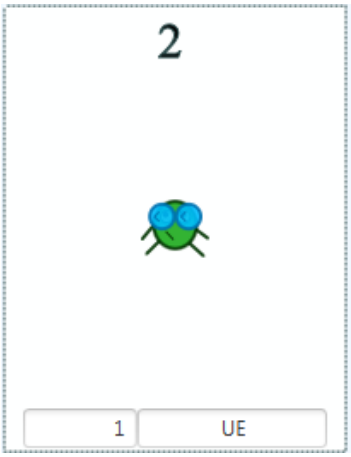
“Ah, but there’s a restriction,” I caution. The restriction is as follows:

(1) If the Dome you want to remove using Universal Elimination is binding more than one square Lens, then you must do the same thing to all the Lenses you free by removing the Dome.

For example, if you have the following Card, and want to eliminate its Dome so you can change the Lens on the left . . .



. . . then you have change the Lens on the right in exactly the same way:



They both started out as red squares, so if you use Universal Elimination to (remove the Dome and) change one Lens into a blue circle, you have to change the other Lens into a blue circle too.

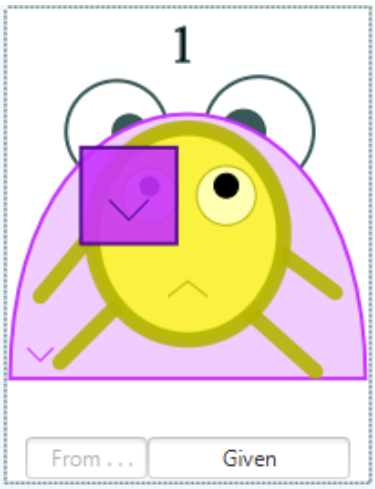
“But if I have to change the second one in exactly the same way as the first, why change the first one?” you ask. “I don’t like people making me do things.”

The answer is that the left Lens had the same color as the Dome you just removed. With that Dome gone, the red Lens is no longer bound. However, if you leave the Lens the same color and shape as it was, you might get caught by the cops on the street who watch for people stealing 3D movie glasses. So, it’s a good idea to change them to make them less recognizable.

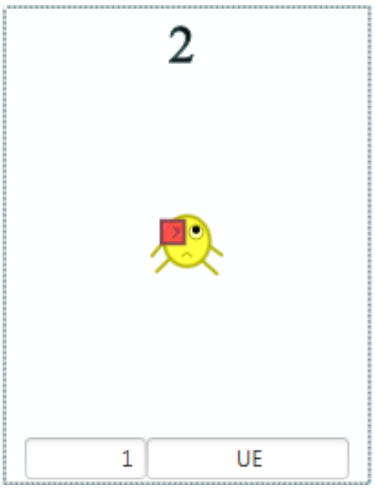
However, if you picked a matched pair of 3D glasses to begin with, you’re going to want them to stay matched after the alteration. You’re clearly a person who likes consistency. So, if both of a Character’s Lenses were bound by the Dome you just eliminated, then you can change them both so long as you change each Lens in exactly the same way as you change the other Lens.

When you remove a Dome using Universal Elimination, however, you do not *have* to change the shape of the Lenses that you free. You *can* simply change the colors of the freed Lenses. (That should hopefully be enough to keep you from getting caught.)

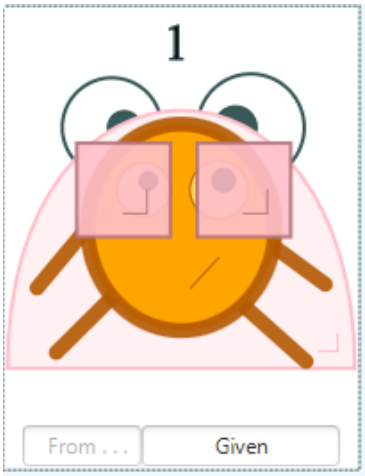
For example, if you start off with Card 1 below, the square purple Lens is bound by the Dome. .



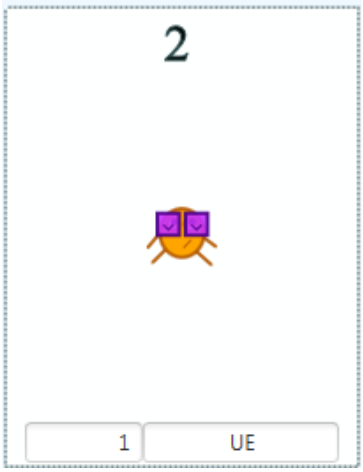
You can then use Universal Elimination to get rid of the Dome, and change the color of the Lens.



The same restriction listed above applies here. If the Character starts off with *two* square Lenses *and* both those lenses start off with the same color, you must do the same thing to *both* lenses when you free them. For example, the following Character is wearing two square pink Lenses inside a pink Dome.

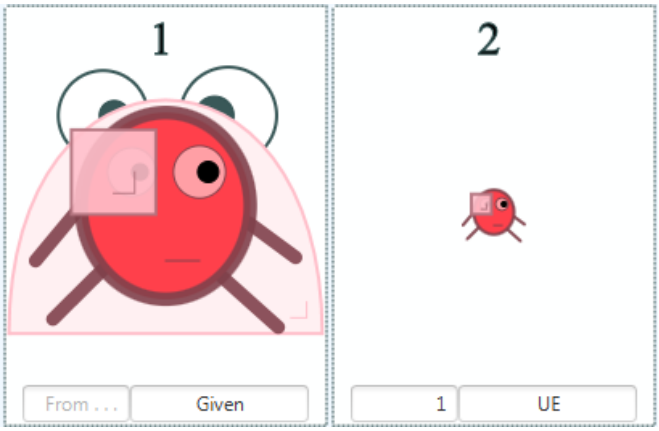


When we eliminate the Dome and change the Lens on the left or Right to a new color, we have to do the same thing to the other Lens as well:



Notice in all these examples that we are changing the *Lenses* that match the color of the Dome we're eliminating. This does not give us the right to change the colors of the *Characters* wearing the Lenses, even if they were the same color as the Dome we eliminated. Just because you have the money to pay for your glasses to be upgraded doesn't mean you have the money to pay for plastic surgery to alter your face.

And just because you know someone who can do special eyewear alterations, doesn't mean you have the money to pay for them. But that's okay (if you have the guts to handle to risk). You can just "not and say you did." For example:



The "new color" you apply to the Lenses after removing the Dome can be identical to the color they used to have. This is not going to help you blend in at all, or fool anyone into thinking those Glasses don't belong to the theater you stole them from. But sometimes you've just got to be brave.

Summary: If a Dome is the outermost piece in a Card, is Club-free, and contains a Character, you can use Universal Elimination to remove the Dome (leaving the Character behind) and can change any Lenses the Dome was binding *if* you do the same thing to all the Lenses the Dome was binding.

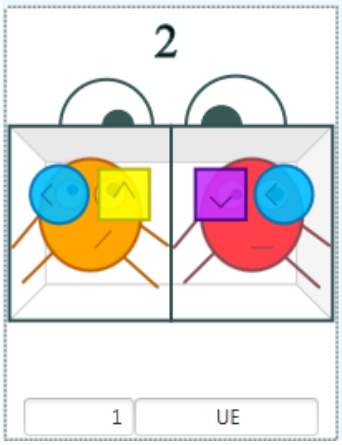
Very often, however, the main Piece inside a Dome is a Chambergon, not a Character. What can you do then?

8.2: Domes with a Chambergon Inside

If a Dome (a) has no Clubs, (b) is the outermost Piece in some Card, and (c) contains a Chambergon, then you can use Universal Elimination to get rid of the Dome, leaving just the Chambergon. When you do this, you can change any Lenses you free to circular Lenses with a color from (preferably) the first half of the alphabet.

For example, in the following Card, both Characters have one Lens that is bound by the Dome.

If we use Universal Elimination to remove the Dome from Card 1, we can change the Lenses we free (the square pink Lenses). For example:



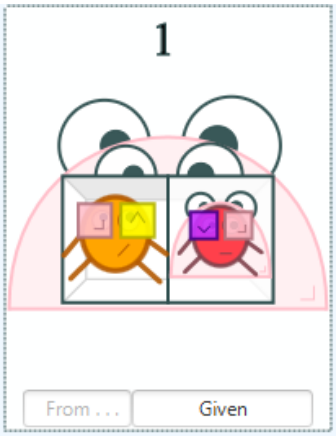
Notice how both square pink Lenses are now circular blue Lenses. With the Dome no longer there to bind them, we were free to change them.

There are restrictions on what we're allowed to do, however. The first is the same as before:

(1) If you change one of the Lenses you free, you have to do exactly the same to any other Lenses you free.

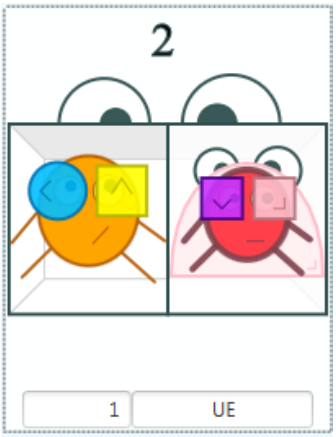
(2) However, you cannot change the shape or color of any Lens that that is still bound by some other Dome even after you remove the outermost Dome.

We obeyed restriction "(1)" by changing *both* square pink Lenses in the same way: they both ended up as circular blue Lenses. Restriction "(2)" is a little weirder, though. Take the following Card as an example.



Notice that in the Card above, there are two Domes. Both are pink, and one Dome is inside the Chambergon that is inside the other Dome. Notice furthermore that the orange Character has a square pink Lens that is bound by the outermost Dome, while the red Character has a square pink Lens that is bound by the inner pink Dome.

If we remove the outermost Dome, therefore, we can only change the square pink Lens on the left. That is the Lens we free by removing the outermost Dome. The square pink Lens on the right, however, is still bound by the inside Dome, and thus we are not free to change it.



The reason you are allowed to change both the shape and color of a Lens when removing the Dome that matches it (colorwise) is the same as before. Domes represent national cinema chains, so specialists all over the country will have had a reason to develop or purchase the technology necessary to alter the 3D glasses from those chains. Once you're outside the theater that loaned

you a square Lens (if you can make it out the door without being caught), therefore, you're free to visit your local black market eyewear emporium.

If you don't have enough money to change both the color and the shape of a square Lens when you free it, however, you can use Universal Elimination to change only the color. Just remember to change all the Lenses that you free to the same color, and remember not to change any Lenses that you haven't actually freed (i.e., don't change any Lenses that are still bound by Domes or Bowls that are still present even after you've gotten rid of the outermost Dome).

Furthermore, there is another restriction. When you use Universal Elimination to eliminate a Dome, you can change either the shape or color (or both) of any Lens you free, so long as:

(3) you do not bind any Lens you just freed to a new Dome or Bowl in the process.

In other words, if you change the color of a Lens you freed, and if the new color of the Lens matches another Dome or Bowl that the Lens is still in, you have bound it to that other Dome or Bowl. And that is illegal. Take the following, for example.

In the Card above, both the yellow Lenses are bound by the outer yellow Dome If we use Universal Elimination to remove that dome, we're allowed to change the color and shape of both Lenses. However, if we change the color of the two freed Lenses, we have to make sure that we don't use pink. If we used pink, we'd have to change both yellow Lenses to pink (since we have to do the same thing to all the Lenses we free), and that would mean the red Character's yellow Lens would turn pink end up being bound by the pink Dome it is inside of.

"What so bad about that?" you ask. "The red Character already has a pink Lens, so it's clearly okay with pink."

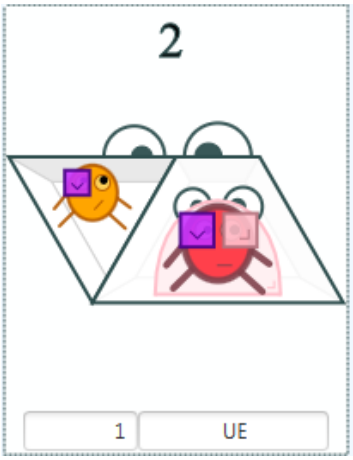
And that's true. The problem would be that if we free the red Character's yellow Lens by eliminating the yellow Bowl, but then change the color of the Lens to pink, we'd be binding the Lens to the pink Dome. We would have freed it from one Dome only to bind it to another. And that's just stupid. Why steal a Lens from one theater only to make it look like it belongs to another theater *that you are still inside*? You would have escaped the security guards from one

place only to put yourself in danger to the security guards at the other. If you're going to be a thief, at least be a smart thief.

So, there is one final restriction:

(4) If you cannot satisfy conditions 1, 2, and 3 all at once, you have to pick a different "new" color.

For example, we could use the following version of Card 2.

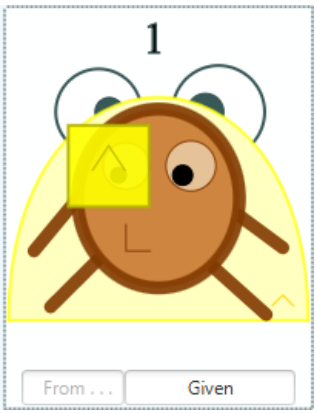


Notice that both the yellow Lenses are now purple, even though they haven't changed shape. That is perfectly alright. We've done the same thing to both Lenses, and we haven't freed a Lens only to bind it to a new Dome/Bowl in the same move.

In fact, we don't even have to change the color of the Lenses when we remove a Dome that they match. We can just take our chances with getting caught.

Summary: If a Dome is the outermost piece in a Card, is Club-free, and contains one Character (by itself) or more than one Character (inside one or more Chambergons), you can us Universal Elimination remove the Dome and can change the color, change the color and the shape, or leave unchanged both the color and shape of any Lenses you free *if* you do the same thing to all the other Lenses you free, *and if* you do not change any Lenses that are still bound (even with the outermost Dome gone), *and if* the change you make doesn't end up binding one of the freed Lenses to another Dome or Bowl.

Example 1a

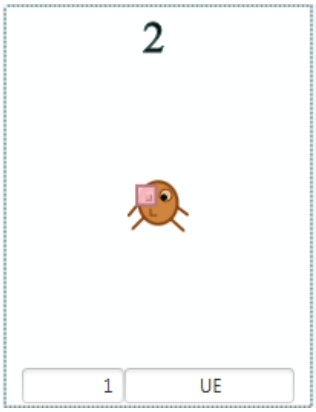


Imagine we have the Card above. (He looks like a teddy bear, doesn't he? [I've decided this one is a 'he.']) And imagine that we want to eliminate the Dome. What would Card 2 look like?

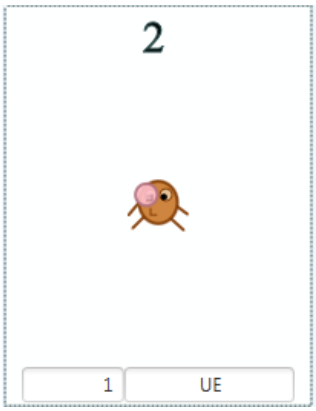
Here's one version:

In this version of Card 2, we could have changed the square yellow Lens, but we don't. We don't have to, so we don't.

But we could have. So, here's another version of Card 2.



In this version of Card 2, we change the yellow Lens when we remove the yellow Dome. We only change the color of the Lens however. That is perfectly fine since we aren't changing the color to match a Dome that the Character is still in (since there are no other Domes left).



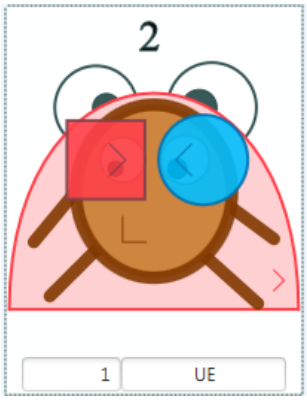
And in this version of Card 2, we change both the color and the shape of the Lens. (Given that we changed the shape, we don't have to worry about whether the new color matches any leftover Domes.)

Example 2a

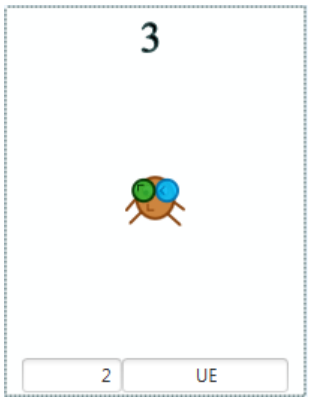
Imagine we are given the Card above, and want to free the Character from its imprisonment inside those two Domes. We have to work our way from the outside in. So, we imagine that we eliminate the outermost Dome, and change its matching Lens as follows.

What's wrong with this Card? The problem is that (a) we changed the square yellow Lens to a square red Lens, and (b) even though we eliminated the outermost Dome, there is still a red Dome left *and* (c) the "new" red Lens is inside that red Dome. That's like stealing a set of 3D glasses from one theater only to hand it over to another, when the point was to keep it for yourself.

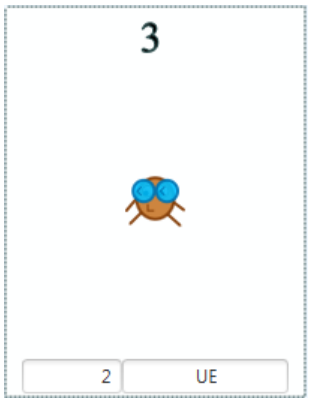
So, we scratch Card 2, and do it over like this:



That's better. We've eliminated the yellow Dome and changed the matching square yellow Lens to a circular blue Lens. Now, let's eliminate the red Dome, and change the matching square red Lens while we're at it.



In this version of Card 3, we change the square red Lens to a circular green Lens when we eliminate the red Dome. But we didn't have to change the shape, nor did we have to change the color. We could have left the Lens just as it was. Or, we could have changed it to a circular blue Lens, like this:



8.3: The Power of Existential Elimination (A Landfill/Inference Rule)

This is the rule for getting rid of Bowls. It's going to look really complicated at first, but in the end it is *identical* to Universal Elimination with only two added restrictions.

> First, Universal Elimination can be used to change *both* Lens colors *and* shapes, but Existential Elimination can only be used to change Lens colors.
>
> Second, how you use Universal Elimination in one Card is not restricted by what happened in previous Cards, while Existential Elimination *is* restricted by any previous Card that has a Character with a free Lens.

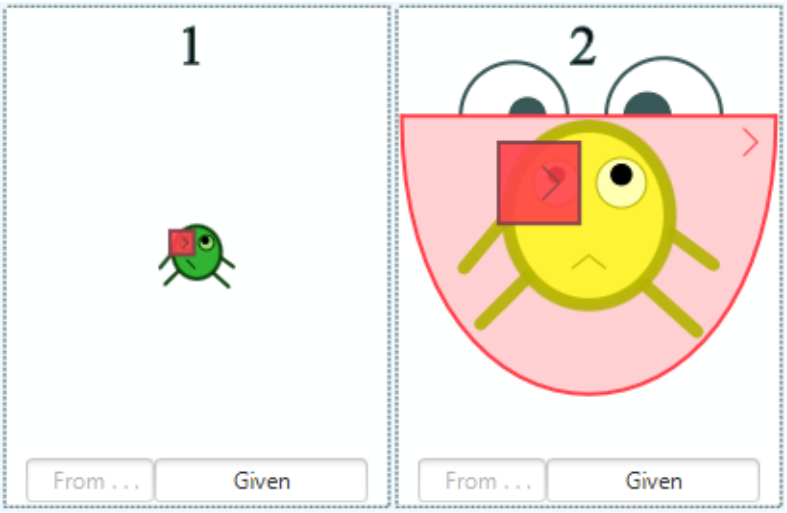
For more on these new restrictions, see below.

8.3.0: Characters inside Bowls

If a Bowl (a) has no Clubs, (b) contains a Character, and (c) is the outermost Piece in some Card, then you can get rid of the Bowl, leaving just the Character and its Lenses, so long as the Bowl you're removing doesn't match the color of any *free* Lens from a Card previous to the one you would be playing if you applied Existential Elimination ("EE"). If the Bowl *does* match some free square Lens from a Card previous to the new one you play by applying EE, then:

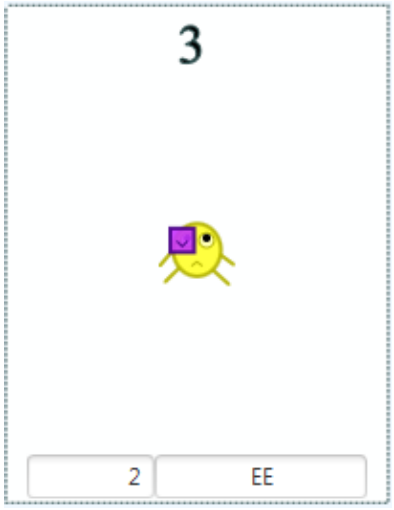
> (1) you must change the colors of all the Lenses you free (by removing the Bowl) to the same new color, and
> (1) you have to change the color of any Lenses you free to a new color that doesn't match any free Lenses in previous Cards.

Take the following example:

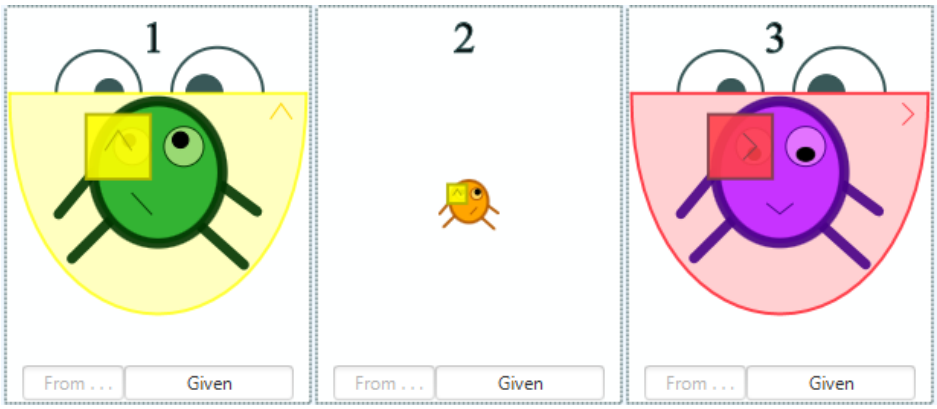


Imagine that we wanted to use Existential Elimination to get rid of the Bowl in Card 2. We notice that the Bowl is the same color as the Lens in Card 1, and we notice that that Lens in Card 1 is free. That means the color "red" has already been taken; so if we remove the Bowl in Card 2, we won't be able to leave the square red Lens in Card 2 red. We'll have to change its color to one that hasn't already been taken.

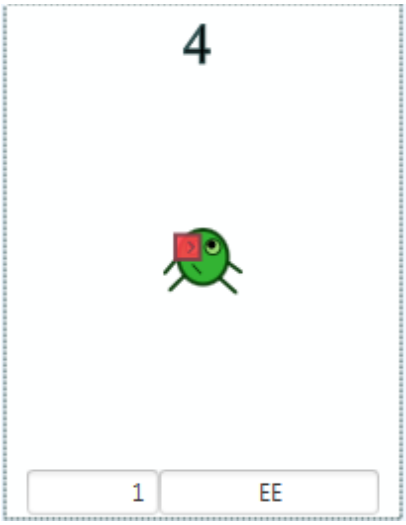
When we apply Existential Elimination to Card 2, therefore, we change the color of the Lens to purple (or yellow, or pink).



It is important to note here that "in a previous Card" means, "in a Card before the *new* Card we would play by apply Existential Elimination." It does not mean "in a Card before the Card to which we are applying Existential Elimination." Take the following example.

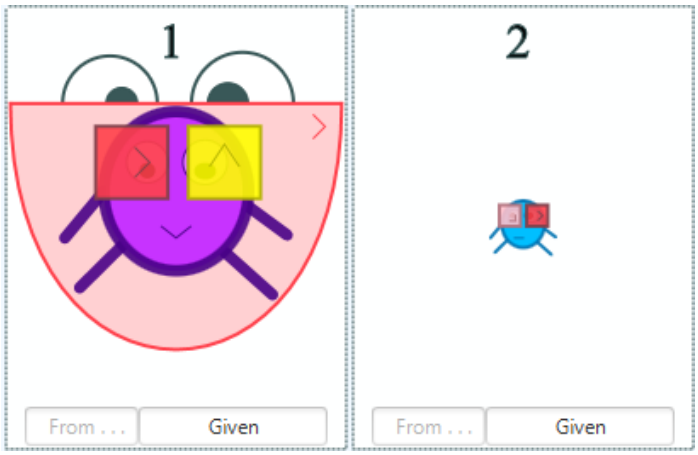


Imagine that we want to get rid of the yellow Bowl in Card 1. If we applied Existential Elimination to it, it would create Card 4. Thus, we have to ask ourselves whether the color yellow would already be taken by any Card before Card 4. As it happens, the answer is, "yes." The Lens in Card 2 is free, and it is yellow. Therefore, if we were to free the Lens from Card 1, and put it in Card 4 (attached to the green Character), we'd have to change its color to something other than yellow. For instance, we might use red.



Now, you may ask why we were allowed to use red when we couldn't use yellow. Isn't red already taken by the Lens in Card 3? The answer is, "no." The red Lens in Card 3 is bound by the red Bowl in Card 3, and only a free Lens can call "dibs" on a color.

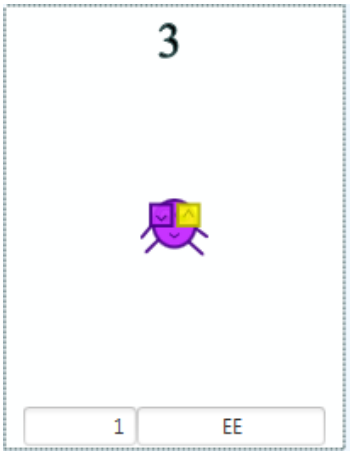
Here's another, even weirder example.



Imagine that we want to eliminate the Bowl in Card 1. That would leave us with a purple Character with two Lenses in Card 3. What color should those Lenses be?

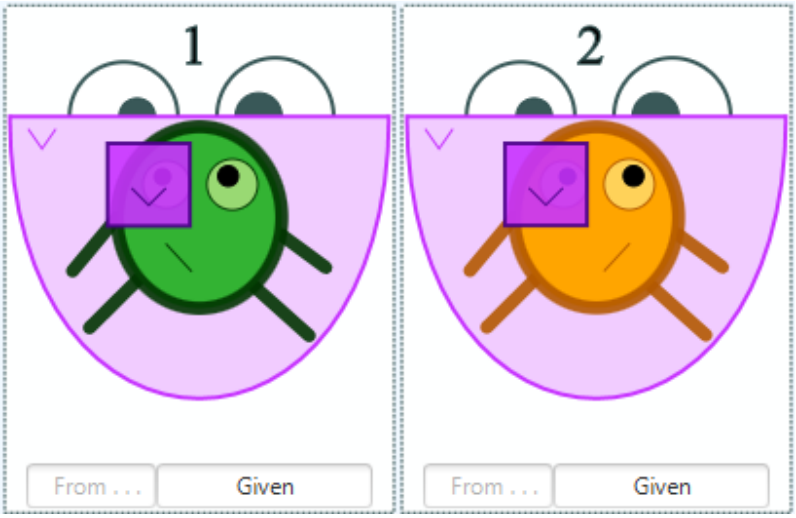
Well, the Lens on the right is yellow in Card 1, and isn't bound by the Bowl. So (since it was *not* one of the Lenses freed by eliminating the Bowl), we can't change its color. We *can* change the color of the square red Lens from Card 1, however. In fact, we must, since the red Lens in Card 2 has got dibs on the color red. Furthermore, since the yellow Lens in Card 1 is free, it has dibs on the color yellow. And the pink Lens in Card 2 has dibs on the color pink. So, if we apply Existential Elimination to Card 1, we'll have to change the color of the red Lens, but won't be allowed to use red, yellow, or pink as our new color.

So, let's use purple.

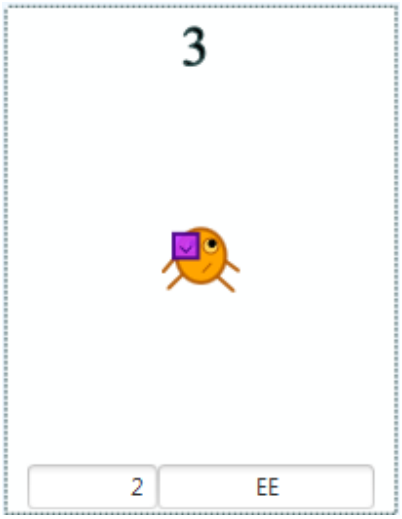


The fact that the Character is the same color as one of its Lenses is irrelevant. Characters don't call "dibs" on colors.

If the Bowl you're removing doesn't match any free square Lens in a Card prior to the one you're playing by applying Existential Elimination, then you don't have to change the color of any of the Lenses you free. Take the following example.



Imagine that we want to get rid of the Bowl in Card 2. If we did, that would free the square purple Lens. So, we ask ourselves if the color purple has already been taken by a free Lens in a previous Card. The answer, fortunately, is "no." There is a purple Lens in Card 1, but it is bound. So, we can leave the color of the Lens in Card 2 alone when we apply Existential Elimination.



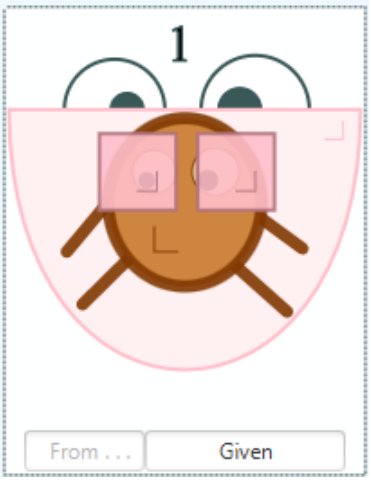
However, whenever you have an available Bowl (i.e., a Bowl that is the outermost piece in its Card) containing a Character, and you wish to remove that Bowl, leaving just the Character, you *can* change the colors on the matching square Lenses inside the Bowl, *even if you don't have to*.

Just remember that *if you change the colors*, you must abide by the conditions that:

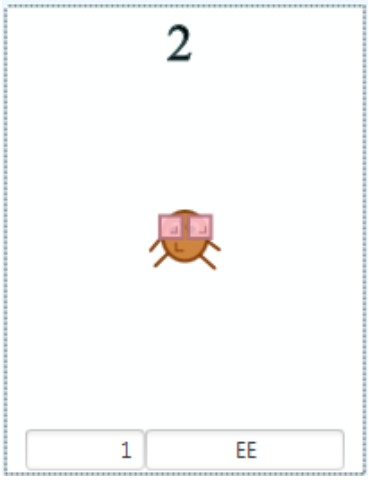
(1) you must change all the Lenses you free to the same new color, and
(2) you cannot use any color that has already been "taken" by a free Lens in a Card that is previous to the one you'd be playing by applying Existential Elimination.

Notice that those two conditions are the same as the previous two. It's just they're worth stating twice. Because repetition.

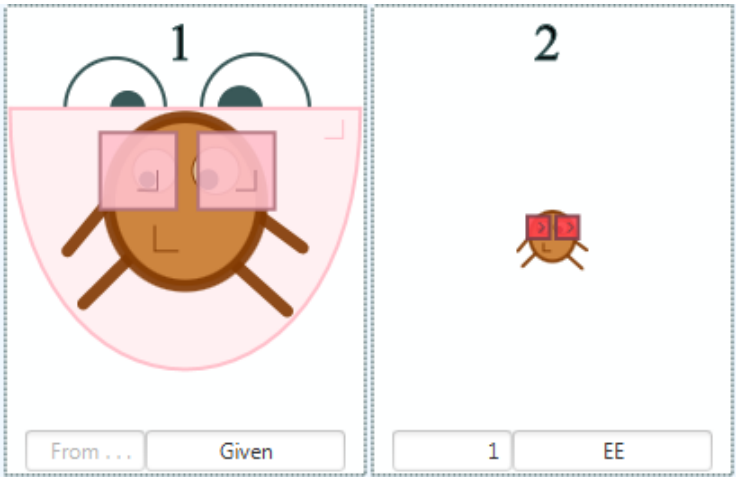
Now, one final example, to illustrate the restrictions:



The two square pink Lenses in the Card above are both bound by the Bowl in that same Card. If we want to eliminate the Bowl using Existential Elimination, we will have to do the same thing—whatever it is we choose to do—to both Lenses. Our options are to either leave them with the color they already have:



. . . or change them both to the same new color.



Notice: while you can use Existential Elimination to get rid of Bowls and change the colors of Lenses, you cannot use it to change the shape of Lenses.

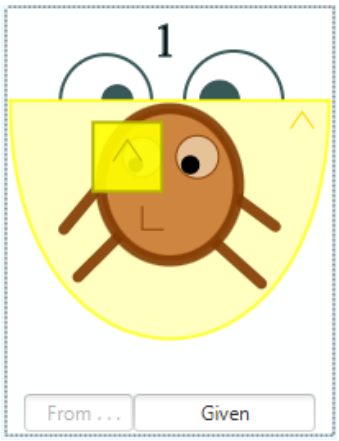
8.3.1: Chambergons inside Bowls

Bowls can contain more than just Characters, of course. If, for instance, a Bowl (a) has no Clubs, (b) is the outermost Piece in some Card, and (c) contains a Chambergon, then you can get rid of the Bowl, leaving just the Chambergon. However, all the conditions from above apply, with a few extra. Here are the restrictions.

(1) If you change the color of one freed Lens, you must change all the Lenses you free to the same color,
(2) whatever color the freed Lenses end up as cannot be the same as the color of any free Lenses in Cards previous to the one you play by applying Existential Elimination,
(3) you can only change the color of Lenses you actually free by removing the Bowl (and thus you cannot change the color of any Lenses that are still bound, even after you remove the outermost Bowl), and
(4) you must not change use a new color that would bind a Lens you are freeing to another Dome or Bowl that is still there even after you eliminate the outermost Bow.

Summary: Existential Elimination allows you to get rid of a Bowl if it is the outermost piece in a Card and if it has no Clubs. When you use Existential Elimination to eliminate a Bowl, you must simultaneously consider the colors of the Lenses the Bowl was binding. The color the freed Lenses *end up with* (whether you change their colors or not) cannot be the same as any Lens that is free in a Card previous to the new one you create. Furthermore, remember only to change the color of Lenses you actually free (do not touch any Lenses that are still bound, even after you've removed the outermost Bowl). And, make sure not to use a new color that would end up binding one of the Lenses you are trying to free to another Dome or Bowl.

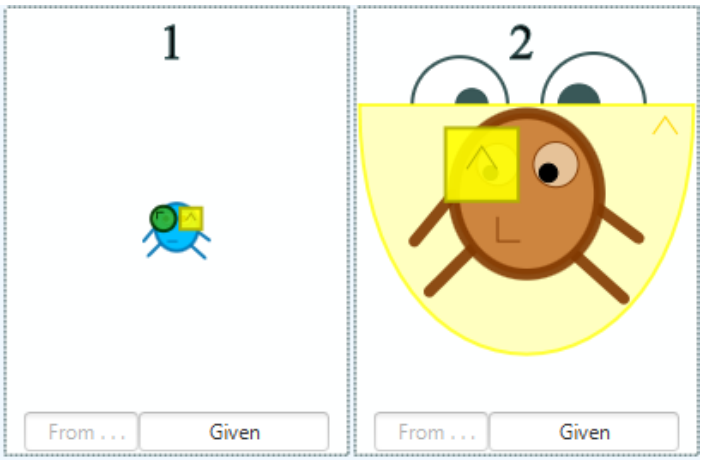
Example 1a



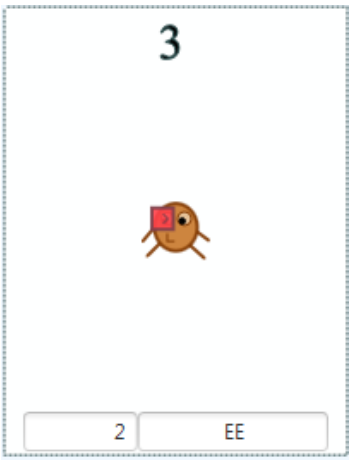
Imagine that we want to free the Lens in Card 1 (by eliminating the Bowl). If we apply Existential Elimination to Card 1, we might produce the following:

Since the color yellow was not taken by any free Lenses in any Card previous to Card 2, we are "free" to leave the yellow Lens from Card 1 yellow. Alternatively, of course, we could have changed the Lens.

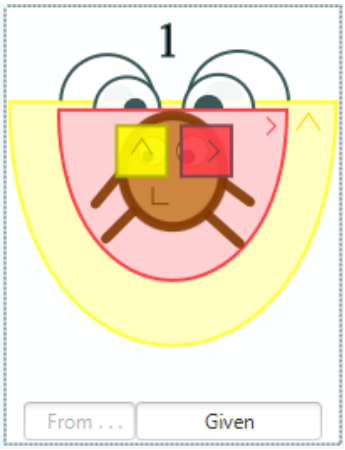
Example 1b



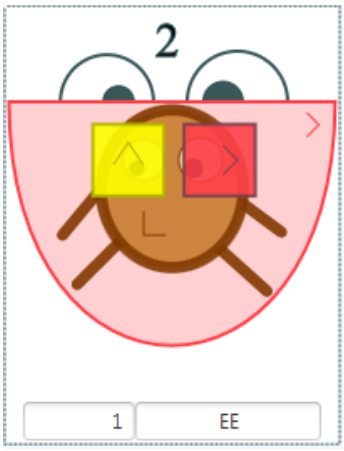
Imagine that, unlike the previous example, both the colors green and yellow were already claimed in a previous Card. If we want to apply Existential Elimination to Card 2, therefore, we are going to have to change the color of the yellow Lens. We'll need to pick something other than yellow or green.



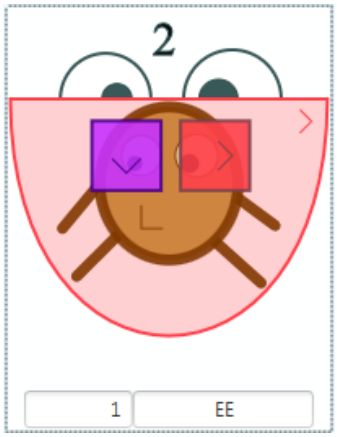
Example 2



Imagine we were given the Card above to start a game, and we'd like to eliminate the Bowls. We can do this if we apply Existential Elimination twice. On the first application, we might produce the following:



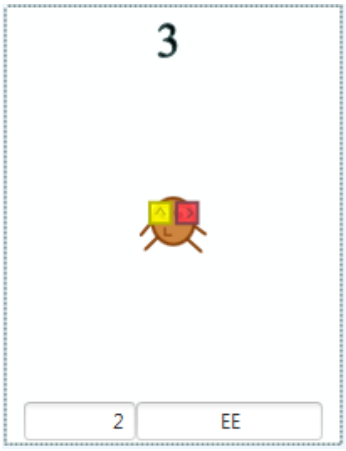
In this case, we would have left the yellow Lens as it was. We could have changed it, however, like this:



After all, purple was not already taken by a free Lens in a Card previous to 2. What we could *not* have done, however, is the following:

If we did this, we would have freed the yellow Lens only to bind it to the red Bowl by changing it to the wrong color. That is unacceptable.

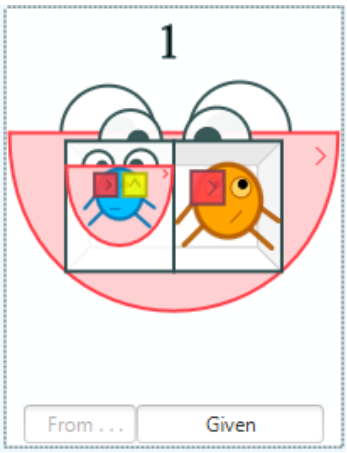
So, let's imagine that we stick with the first version of Card 2 (the one with the yellow Lens). If we apply Existential Elimination to it, we might produce the following:



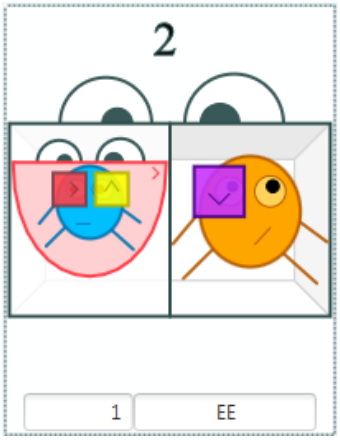
In this case, we would have left both the yellow and red Lenses alone, and just eliminated the Bowls. What we could *not* have done is this:

The color yellow was already “taken” by the free yellow Lens in Card 2. If we wanted the right Lens to end up yellow, then, we’d need to have used the version of Card 2 where we changed the yellow Lens to purple.

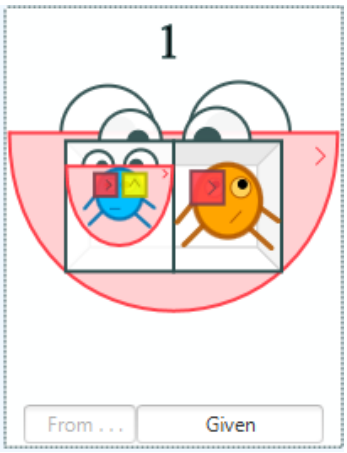
Example 3



If we were to eliminate the outermost Bowl in Card 1, we would free the red Lens belonging to the orange Character. The red Lens belonging to the blue Character, however, would still be bound by the inner Bowl. So, if we apply Existential Elimination to Card 1, we could only change the color of the red Lens belonging to the orange Character.



Notice that we do not change the orange Character's Lens to yellow, since yellow is already free in Card 1, and thus that color is taken. However, we could just as easily leave the orange Character's red Lens red, since no free Lens in a Card prior to 2 was red.



8.4: Why Universal Elimination and Existential Elimination Work Like They Do

Universal Elimination lets us get rid of Domes. Existential Elimination gets rid of Bowls. But they also allow us to change the Lenses the Domes and Bowls were binding. The annoying thing is that the rules for changing Lenses with Universal Elimination are different from the rules for changing Lenses with Existential Elimination.

Domes and Bowls are only supposed to bind square Lenses. This is because Domes and Bowls are quantifiers, and square Lenses represent general terms. General terms like "things" and "stuff" need quantifiers like "all" and "some" added to them. If I say, "Things are sharp," you'll want to know how many of them are sharp, and if any of them are dull. Adding in the quantifiers, we would get "All things are sharp," or "Not all things are sharp," "some things are sharp," "There are some things that are not sharp," and so on. And these are much more precise.

In contrast with square Lenses, circular Lenses are names or labels for particular objects. We do not need quantifiers for such things. If I say, "That tree is really tall," you won't ask, "How many trees?" If I say, "Fred is coming to dinner," you won't ask, "How many Freds?" So, circular

Lenses don't need quantification. They have "exactly one" built into them. And means they don't need Domes and Bowls.

One question that it would be legitimate to ask, nevertheless, is whether it is possible to change square Lenses (which need Domes and Bowls) into circular Lenses (which don't), and vice versa. After all, one of a logician's fundamental moves is changing the shapes of pieces. However, the conventions of symbolic logic (which reserve letters from the end of the alphabet for general terms, and letters from the beginning of the alphabet for names and particular labels) mean that to ask about changing circular Lenses into square Lenses (and vice versa) is also to ask about changing the color of Lenses.

For square Lenses, we only have four options: the two colors from the end of the alphabet (yellow ["y"] and pink ["z"]) and the two supplementary colors from the second half of the alphabet (purple ["p"] and red ["r"]). For circular Lenses, we also only have four options: the two colors from the beginning of the alphabet (brown ["a"] and blue ["b"]) and the two supplementary colors from the first half (or so) of the alphabet (green ["g"] and orange ["o"]). So, if I change a Lens from square to circular, I will need to change its color from one belonging to the second half of the alphabet to one belonging to the first half (and vice versa).

Now, Universal Elimination allows us to change any freed Lens from being square and one color, to being square and another. That amounts to changing Lenses back and forth between yellow, pink, purple, and red. Furthermore, Universal Elimination allows us to change any freed Lens from being square (and thus yellow, pink, purple, or red), to being circular (and thus brown, blue, green, or orange).

Existential Elimination, in contrast, only allows us to change freed Lenses back and forth between the square Lens colors (yellow, pink, purple, and red). Since Existential Elimination does not allow us to change the shape of Lenses, we can never use it to change a square Lens to one of the circular Lens colors (brown, blue, green, and orange).

But what accounts for this difference? Why can Universal Elimination change both color and shape, while Existential Elimination only allows us to change color?

The reason is that Domes cover everything, while Bowls only contain somethings. Domes are universal quantifiers, while Bowls are existential quantifiers.

That doesn't help? Okay, look at like this.

A Character wearing a square Monocle represents a predicate (the Character) applying to a name (the Monocle), or a property (the Character) applying to a particular object (the Monocle). If we then stick the Character with the Monocle into a Dome, we are saying that the property represented by the Character applies to *all* the things represented by the Monocle. We are saying, "All things have this property."

Now, if all things have a property, then *this thing here* has that property, *you* have that property, *I* have that property, and so on. For example, if, "All humans are evil," is true, then "George, that

guy over there, is evil" must also be true. If "everything is physical" is true, then "I am physical" must be true. If "All Americans are awesome" is true, then "President Obama is awesome" must also be true.

So, if we start with an "all things" statement, we can change it to a "this thing" statement and still be telling the truth. And that means if we have a square Lens in a Dome (an "all things" proposition), we can drop the Dome (which amounts to erasing the "all"), and change the square Lens to a circular Lens (which amounts to changing "things" to "this thing").

With Bowls, in contrast, things are different. Bowls represent the phrase "something exists," or "there exists some." So, if we then stick a Character (representing a property) with the Monocle (representing objects in general) into a Dome, we are saying that the property represented by the Character applies to *at least one* of the things represented by the Monocle. We are saying, "There exists something that has this property."

Now, if I am told that there exists something that has a property, I have no idea if it is *this thing here* that has that property, or if it is *you* who has that property, or if it is *I* who have that property, and so on. For example, if, "There is an evil human," is true, then I have no idea whether, "George, that guy over there, is evil" is also true. The most I can conclude is that "Someone is evil." Similarly, if "There exists something that is physical" is true, then I have no idea whether "I am physical" is true. The most I can conclude is that something is physical (it might be me; it might be something else). And if, "There are some Americans who are awesome," is true, then I have no idea whether, "President Obama is awesome," is also true. From, "There are some Americans who are awesome," it only follows that, "Some American is awesome."

So, if we start with a "there exists something" statement, we *cannot* change it to a "this thing" statement and still be telling the truth. The most we can say is that we're talking about something that is out there, but we're not sure who or what it is that we're talking about. And that means if we have a square Lens in a Bowl (a "there exists something" proposition), we can drop the Bowl (which amounts to erasing the "there exists"), but cannot change the square Lens to a circular Lens (which amounts to changing "something" to "this thing"). At most, we can change the color (which amounts to changing "something" to "some object," or "an item" to "an entity," or "someone" to "somebody").

"But why aren't you allowed to use a color that's already been 'taken' by a free Lens in a previous Card?" you ask.

The reason is that we have to avoid the chance of mislead assumptions. Imagine that someone made the following argument.

1. There is someone who is sitting in a chair at this very moment. Call that person "John Doe" (after all, we don't actually know who it is, and "John Doe" is the name we use for people when we don't know who they are).
2. There is someone who is running at this very moment. Call that person "John Doe" (after all, we don't actually know who it is, and "John Doe" is the name we use for people when we don't know who they are).

∴ 3. John Doe is both sitting in a chair and running at this very moment.

Now, it's pretty obvious that no one can be both running and sitting at the same time. The mistake occurred because we used the same "vague" name for both the person sitting and the person running. To avoid this kid of mistake, we really should have used different "vague" names (perhaps "X" and "Y").

To avoid making this kind of mistake in our CBL games, we have to make any Lenses we free using Existential Elimination end up with a color we haven't used before for any free Lenses. (Think of each color as a kind of "John Doe" or "X.")

"But why don't we have to follow the same rule when we use Universal Elimination?" you ask.

Examine the following argument.

1. There is someone who is sitting in a chair at this very moment. Call that person "John Doe" (after all, we don't actually know who it is, and "John Doe" is the name we use for people when we don't know who they are).
2. Everyone who sits in a chair is a very wise person.

∴ 3. John Doe is a very wise person.

This argument is perfectly legitimate. Even though the name "John Doe" is just a vague name we're using because we don't know the person's actual name, if it is true that everyone who sits in a chair is wise, and if it is true that someone is sitting in a chair, then we know that person—whoever the person is—must be wise.

8.5: Encoding the New Pieces

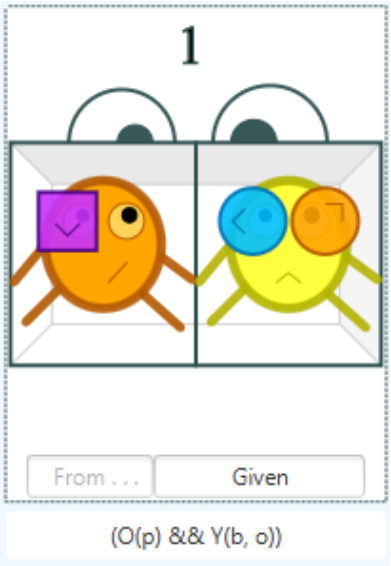
With new pieces comes the need to learn new symbols. Take a look at the following.

A blue Character (Card 1) continues to be symbolized by "B," of course. What is new is that if a Character has a lens (Cards 2 and 3), we add the lowercase version of the letter that its color represents inside parentheses. Notice that whether the Lens is circular or square does not show up in the symbolization beneath the Card. We simply record the letter that symbolizes its color.

If a Character has two Lenses, we put both their letters in the parentheses, separated by commas.

Order is important when you have two Lenses. Card 4 is "B(o, y)," not "B(y, o)." If we wanted "B(y, o)," we'd have to make the left Lens yellow and the right Lens orange. (Note, once again, that the fact that the orange Lens is circular while the yellow Lens is square does not show up in the symbolization.) Card 6, furthermore, emphasizes the need to distinguish between upper- and lowercase letters. A "B" stands for a blue Character, while a "b" stands for a blue Lens.

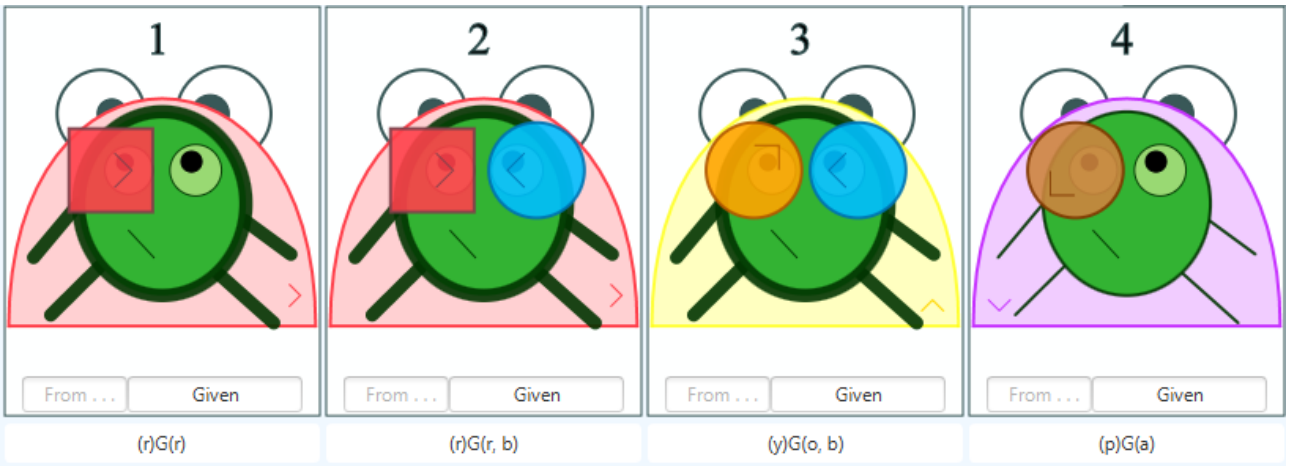
Put Characters with Lenses inside Chambergons is relatively straightforward. For example:



In this Card, the first and last parentheses mark the outer boundaries of the Parcel. The parentheses around the “p” show that the “O” owns the “p,” while the parentheses around the “b, o” show that they belong to the “Y.”

Even if we add Clubs, nothing really changes. The exclamation marks go in front of whatever they’re attached to, and they don’t attach to Lenses.

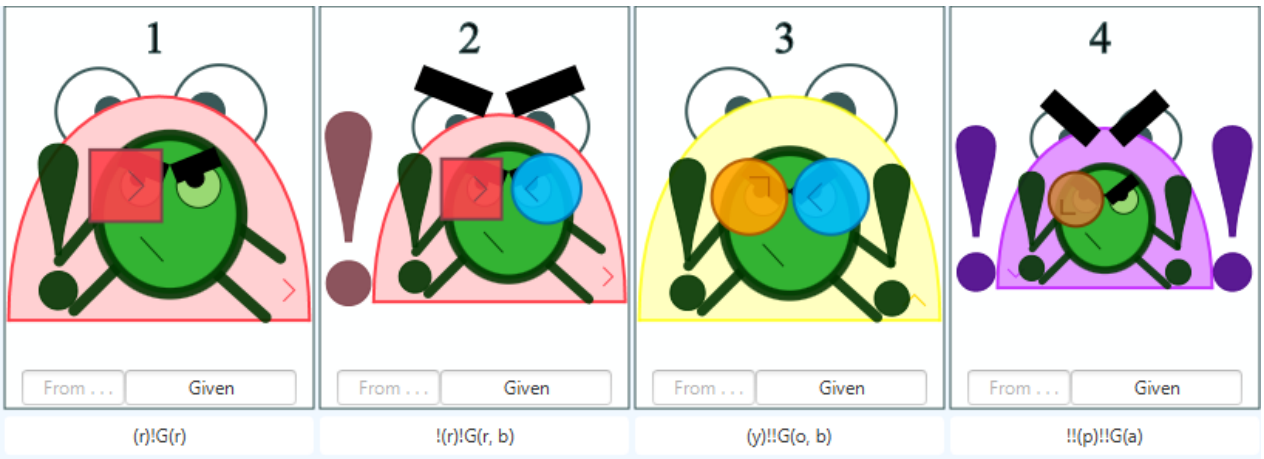
The only other real changes, therefore, have to do with Domes, Bowls, and Coins.



To symbolize Domes, we use lowercase letters (representing their colors), just like with Lenses. We then place the lowercase letter for the Dome inside parentheses, and attach it to the front of whatever piece it contains. In other words, we treat the symbol for a Dome in exactly the same way we would treat a the symbol for a Club.

In Card 1, we have a red Dome, which is symbolized as "(r)." It contains a green Character, so we put the "(r)" in front of a "G:" "(r)G." However, the green Character also has a red Lens. So, we have to write "(r)G(r)." The first "(r)" is for the Dome, while the second is for the Lens.

If we add Clubs, the Cards look like this:

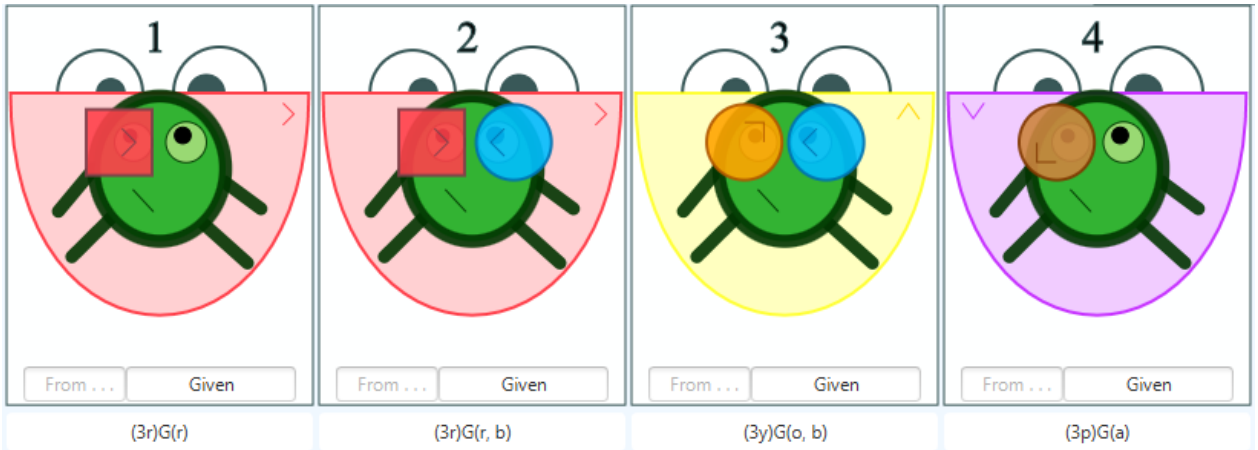


In Card 1, we put the "!" in front of the "G," like we would normally do, and tack the "(r)" that represents the Dome in front of the "!". In Card 2, the Dome itself has a Club, so we tack a "!" to its front as well. In other words, the general principle still holds that if a piece has Clubs, we put the exclamation marks that symbolize those Clubs in front of the piece.

Domes are the icon for what logicians call "universal quantifiers," and there have historically been two ways of representing universal quantifiers. The first is to use an upside-down "A," which looks like this: ∀. So, if you wanted to say "everything" or "all things" or "each and every

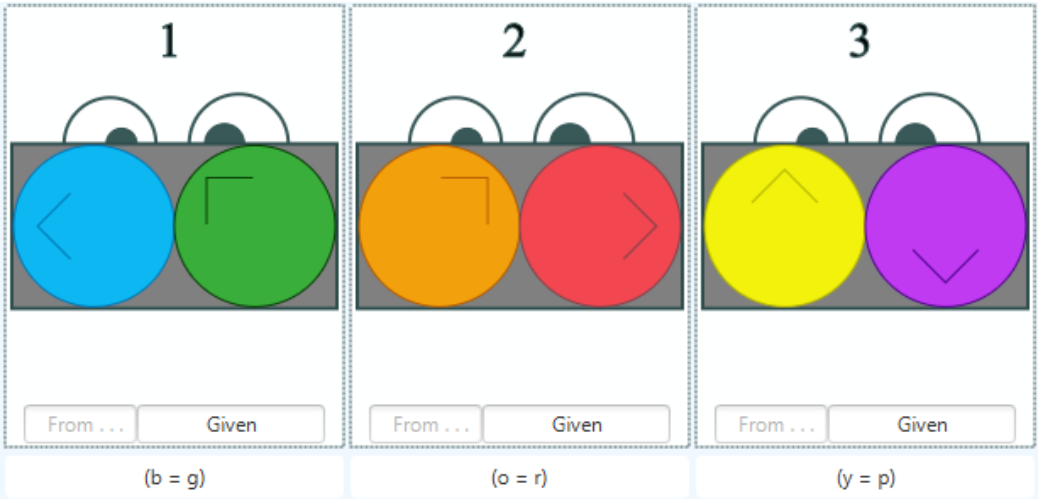
thing," you would write "∀x." The other way was simply to put the variable "x" in parentheses, like this: "(x)." Since we don't have the upside-down "A" character on our keyboards, we will be using the second option.

"But how do we represent Bowls?" you wisely ask. The answer is that logicians use a backwards "E" (since an upside-down "E" looks exactly like a right-side-up "E"). This is because Bowls are the icons for "existential quantifiers." So, if I want to say, "there exists something," or "some object is," or "an entity exists," I write, "∃x." Unfortunately, we don't have a backwards "E" on our keyboards either. So, we use the closest thing we've got: a "3."

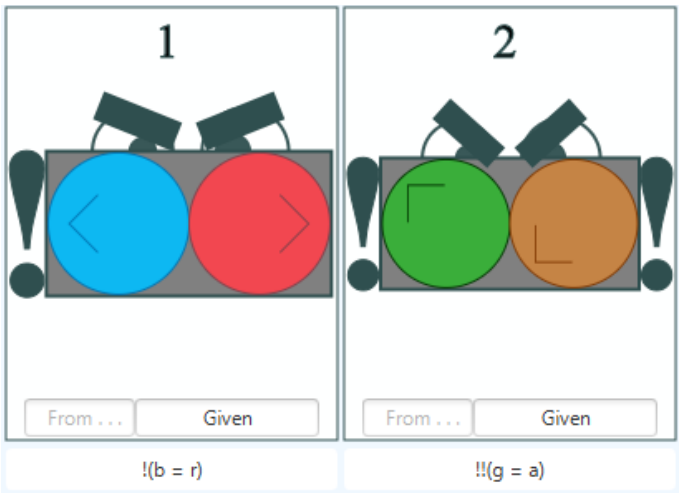


As you can see, the symbolizations of these Cards are identical to the ones on the top of p. 309. However, we have inserted a "3" into the parentheses representing the Bowls. While a red Dome is "(r)," a red Bowl is "(3r)." While a purple Dome is "(p)," a purple Bowl is "(3p)."

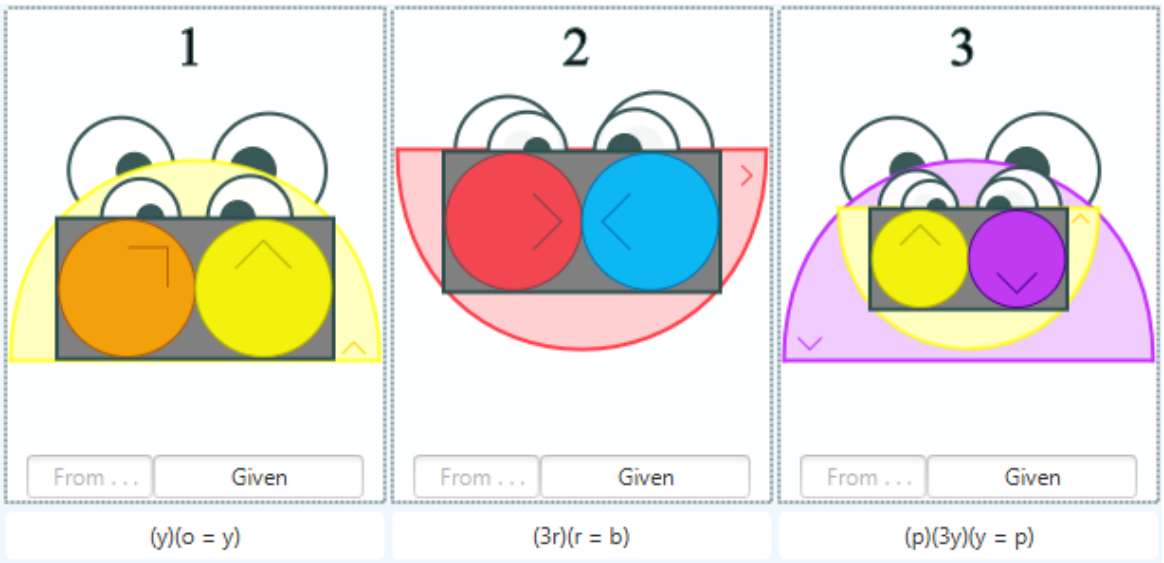
And finally, we should briefly show how to symbolize Coins. Coins represent identity statements. Thus, we symbolize them with the sign for equality.



Coins can have Clubs too.



And Coins can go inside Domes and Bowls.



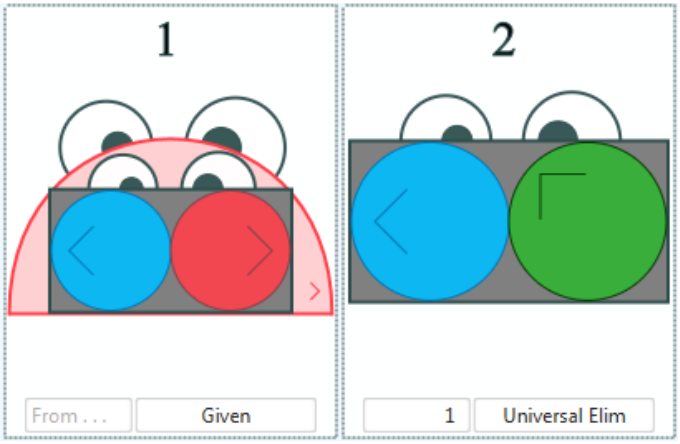
The only thing really new here is in Card 3. Notice that we have "nested" quantifiers: a Bowl inside a Dome. To symbolize this, we work from the outside in. First, we type the symbol for the Dome—"(p)"—then we type the symbol for the Bowl—"(3y)"—and then we type the symbol for the Coin: "(y = p)." This is how we always work when we have Bowls and Domes inside each other: work from the outside in, typing the symbols for each in the order that you come to them.

8.6: A Note about Coins

At level 8, coins have about as much complexity as Characters. We can't do much with them, in other words, other than giving/removing Clubs, putting them into Chambergons, and using them against Chambergons we don't like (e.g., in MT or DE). In Level 9, we'll learn a rule that applies to them only (unhelpfully called "Identity Elimination"), and that actually allows us to treat them as something more than hunks of material to be shifted around. However, even here in Level 8, we can do something interesting with Coins: we can change their colors.

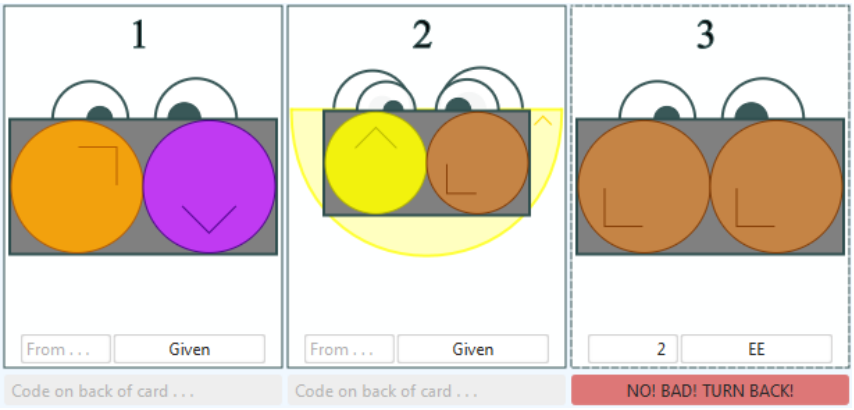
Coins, you see, are essentially the same as Lenses. They are, in fact, like two Lenses, joined together. Thus, Universal Elimination and Existential Elimination apply to Coins in the same way they apply to Lenses, with the exception that you cannot change the "shape" of a Coin; you can only change a Coin's colors.

Here's how to think about Coins, therefore. Think of a Coin as two circular Lenses joined together. They can be "bound" by Domes and Bowls, and be "freed" by removing Domes and bowls. What's unusual about them is that half a Coin can be bound while the other is free. For example:

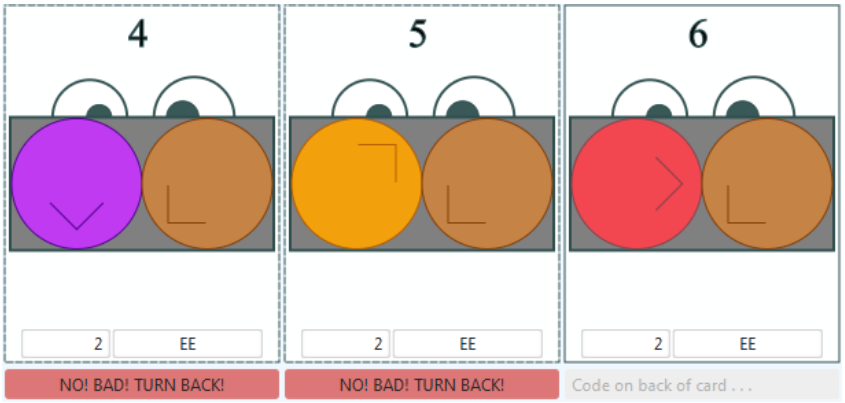


In Card 1, the blue "Lens" in the Coin is free, while the red one is bound by the red Dome. Thus, when we remove the red Dome using Universal Elimination, we are allowed to change the red "Lens" to be whatever color we want.

Now, take a look at this example:



Why is Card 3 wrong? The answer is that it tries to use Existential Elimination to change the yellow "Lens" in Card 2 to brown. But brown is already free in Card 2. That color is already taken, so we can't use Existential Elimination to turn anything brown. We have to try again. To that end, we try Card 4, below.



Card 4 tries to turn the yellow “Lens” in the Coin in Card 2 purple. The problem is that the purple “Lens” in the Coin in Card 1 has already taken that color. So, we can’t use Existential Elimination to turn anything purple. That means we have to try again, and we do so in Card 5. But Card 5 tries to turn the yellow Lens orange, and orange is also already taken. The Coin in Card 1 has an orange “Lens” in it, and that “Lens” is not bound by an orange Bowl or Dome. So, we have to pick a different color.

Finally, in Card 6, we get it right. We use Existential Elimination to turn the yellow “Lens” from Card 2 red. Red hasn’t already been taken, so we are free to use it.

With Universal Elimination, of course, we don’t have to worry about this kind of thing. But it’s important to keep in mind that the “Lenses” that make up Coins can be bound and free, and thus can have their colors changed—following the same rules as for normal square and circular Lenses—but the Universal and Existential Elimination powers.

Chapter 9: Level 9 Powers

9.0: The Last Three Powers

We have reached the final level at which new powers are introduced, and like Level 8, it contains three rules—one of which is easy and the others of which are more difficult. The more difficult powers are necessary, after all, since we need them to balance out the "elimination" rules we learned in Level 8. We can get rid of Domes and Bowls, but we also need to be able to introduce them.

The easy rule is called "Identity Elimination," and we'll learn it first. Unfortunately for us, there is no corresponding "Identity Introduction" rule at this level of symbolic logic. How, after all, could we logically prove that two things were identical to each other? The most obvious option would be to appeal to what philosophers call "the identity of indiscernibles." If a philosopher holds to the identity of indiscernibles, this means she or he believes that if one object has all the same properties as another object, this means the two objects are actually one and the same. If you can't tell them apart—because all their properties are identical—then they must not be two separate things.

The problem with appealing to the identity of indiscerinbles in order to create an "Identity Introduction" rule is that it would require "quantifying over predicates." We would need to be able to say things like, "All the predicates that apply to Bob also apply to Sam," as opposed to saying, "Only some of the predicates that apply to Bob also apply to Sam." And that would mean we would need Domes and Bowls that bind not Lenses, but Characters. (Characters, after all, stand for predicates when they are wearing glasses.) But Domes and Bowls we have so far don't bind Characters. They only bind Lenses. And that would mean we'd need to introduce another two types of pieces in order—Character-binding Domes and Character-binding Bowls—if we wanted to have an Identity Introduction power. That is something we can do, but isn't something we're going to do in this class. You'll need to take an advanced logic course if you want that.

Unlike Identity Introduction, however, we will be learning Universal Introduction and Existential Introduction. And yet, of these two powers, it would seem like Universal Introduction has the same problem as Identity Introduction. If you introduce a universal, you're going from claiming that some predicate applies to a particular thing (i.e., from a Character wearing a Lens that is not inside a Dome or Bowl of the same color) to claiming that some predicate applies to everything (i.e., to a Character wearing a Lens that is inside a Dome or Bowl of the same color). And how could you possibly prove—based on the fact that one thing has a particular property—that everything has a particular property?

The answer to that question is: magic. Actually, the answer is if you know that there is nothing special about the thing you are putting inside the Dome. If it was just an abstract stand-in for everything else from the very beginning, then anything that is true of it will be true of everything else too. The way this actually plays out in a given game, however, is something you'll have to wait to see below. We'll begin with Existential Introduction, since it is the easier of the two introduction rules for this chapter, and then move on to Universal Introduction.

9.1: The Power of Existential Introduction (A Landfill/Inference Rule)

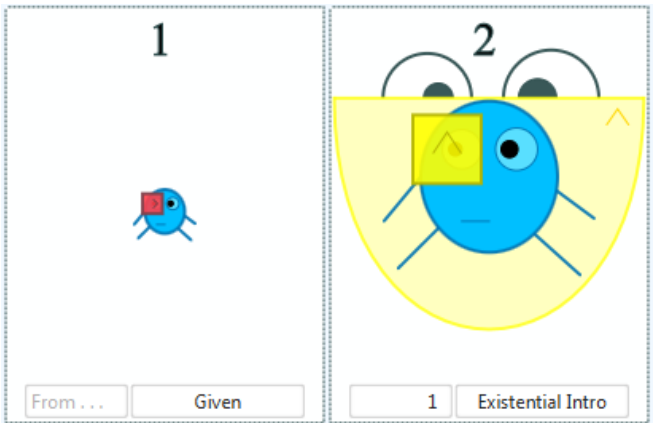
We have already learned how to get rid of Bowls, but have not yet seen how to "create" Bowls. That is what this rule is for.

9.1.0: Putting Characters into Bowls

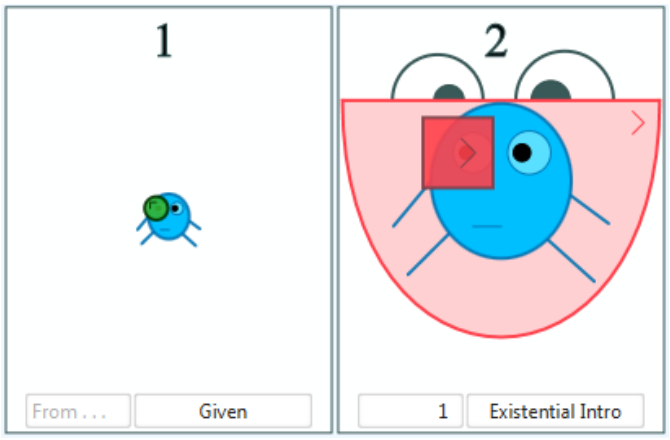
If a *Character* is alone in a Card (even if it has a Club), you can enclose it in a Bowl that matches one of its square Lenses whenever you want. Furthermore, you change the color on a Square Lens, or even change one of its circular Lenses into a new square Lens (usually with a new color), so long as:

(1) the new color doesn't match another Lens that the Character already had before you made the change, and
(2) you make the new Bowl you're adding match the new color.

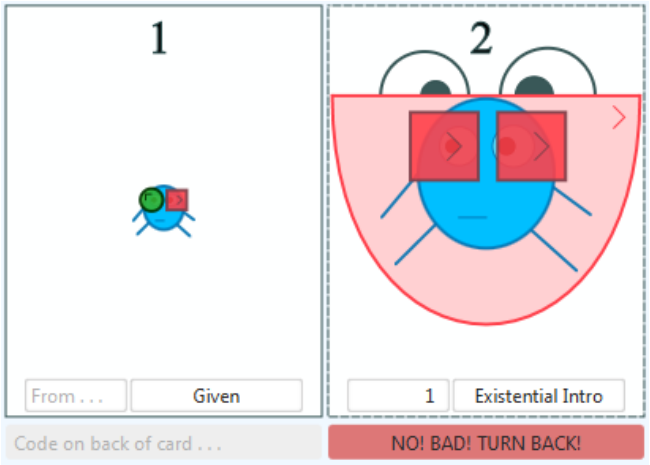
For example, the following is fine:



This, also, is fine:



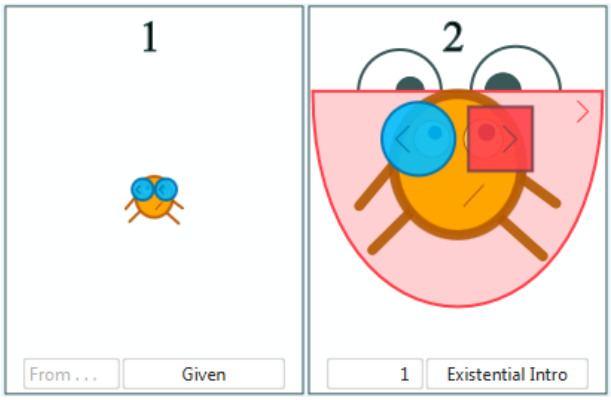
But this is not fine:



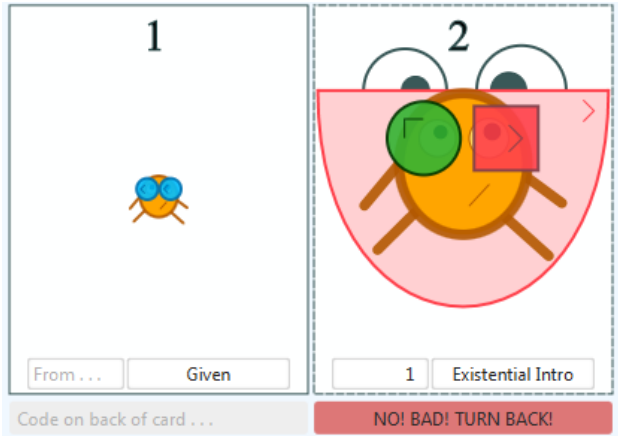
In the above example, the circular green Lens is changed to a square red Lens. But the Character already has a red Lens. So this is not allowed. Red is "already taken," so we would have to pick a different color. In Existential Introduction, if Lenses start out with different colors, they have to end with different colors.

However, just because two Lenses start out with the same color doesn't mean they have to stay the same color when you apply Existential Elimination. You can use Existential Elimination to change some Lenses, while leaving others that match them alone. The only restriction is that the Lenses you do change have to all have been the same color and shape to start, and they have to end as the same color and shape (even if you left others that have the same color and shape alone).

Here's an example:



In this example, we start off with two circular blue Lenses, and only change one of them to a red square Lens. That is perfectly fine. What is not fine is this:



We don't have to change both the circular blue Lenses, but if we do change them both, we have to change them in the same way.

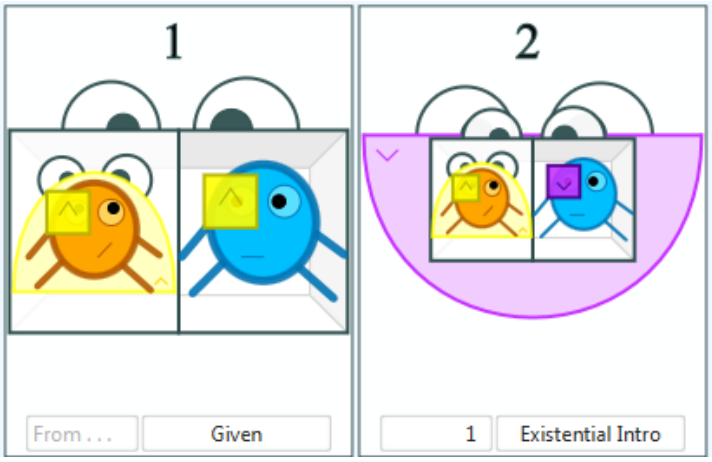
Now, you may wonder why it's okay to change one of the blue Lenses, but not the other, when you use Existential Introduction. The reason is that nobody wants to be bound. So, if one of a set of matching Lenses gets changed so that it is bound by a new Bowl, the other Lenses that are left unbound consider themselves lucky. They don't complain that now the bound lens is different from the rest of them. They just feel kind of sorry for it.

9.1.1: Putting Chambergons inside Bowls

If a *Chambergon* is the outermost Piece in a Card, even it has a Club, you can enclose it in a Bowl that matches one of the square Lenses inside it so long as the square Lens(es) in question isn't (or aren't) already in a matching Dome or Bowl inside the Chambergon. Furthermore, you can even change the color on the square Lens, or change a circular Lens into a square Lens (usually with same a new color), so long as:

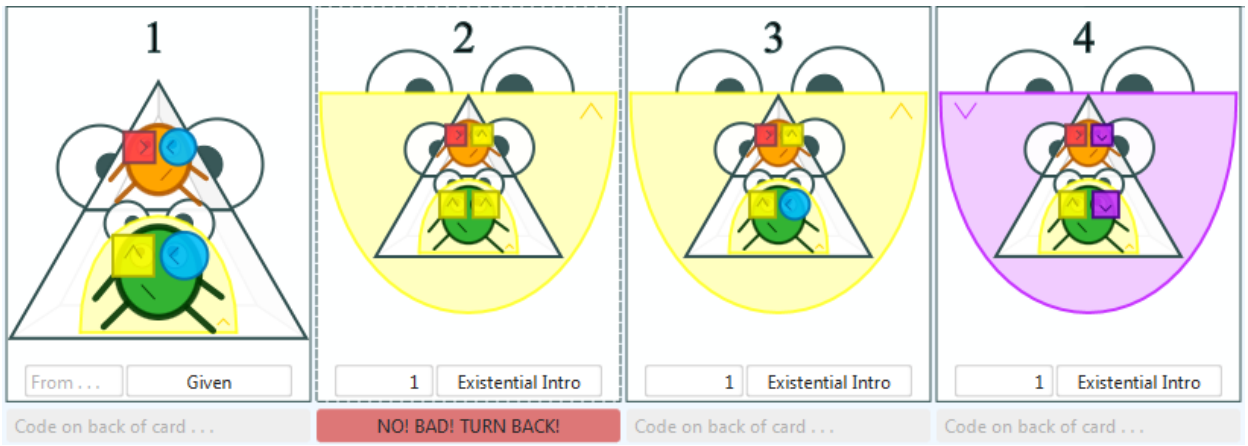
(1) the new color doesn't match another Lens that the Chambergon already has (unless that matching Lens is already inside a matching Dome or Bowl),
(2) the Lens(es) whose color you want to change isn't (aren't) already in a matching Dome or Bowl inside the Chambergon, and
(2) the new color doesn't match any Dome or Bowl its Lens(es) is (was) already in before you added the new Bowl.

The first restriction above is the same as before. If a color is already "taken" in the Card, you can't use Existential Introduction to change other Lenses to match it. The second restriction, however, is new. It says that you cannot change a Lens if it is already bound by some Dome or Bowl before you apply Existential Introduction. Here's an example.



In this example, the yellow Lens on the left is bound by a yellow Dome, but the yellow Lens belonging to the blue Character is free. When we introduce a bowl through Existential Introduction, therefore, we can only change the Lens belonging to the blue Character. Only it is free, and thus only it can be bound (and hence changed) by Existential Introduction.

The third restriction listed above—that you can't use Existential Introduction to change a Lens to match a Dome or Bowl it is already in—is a little weirder. Here's an example.



In this example we are trying to use Existential Introduction to change the two blue Lenses in Card 1. The problem is that the blue Lens belonging to the green Character is already in a Dome, so we can't use Existential Introduction to change it to have the same color as that Dome. When we try that—in Card 2—it doesn't check out. You can't use the introduction of Dome or Bowl as an excuse to bind a Lens to a different Dome or Bowl. Instead, we have to either leave the green Character's blue Lens unchanged (only changing the orange Character's blue Lens), like in Card 3, or we have to change the blue Lenses to a color other than yellow, like in Card 4.

In Card 3, above, you see once again that Existential Introduction can be used to change only one, or a few, of the matching free Lenses in a Card. You do not have to use it to change them all. The only restriction is that if you use Existential Introduction to change more than one of the previously-free lenses in a Card, you can only change lenses that match each other and you have

to make the same change to them all. So, in Card 4, we changed both the blue Lenses, and thus had to change them both to the same thing. We couldn't have changed one to red and the other to green.

9.2: The Power of Universal Introduction (A Landfill/Inference Rule)

Universal Introduction is almost identical to Existential Introduction, except it has a couple extra restrictions. Well, also, you use Universal Introduction to put things into Domes, not Bowls. But other than that—and its extra restrictions—it's basically the same as Existential Introduction.

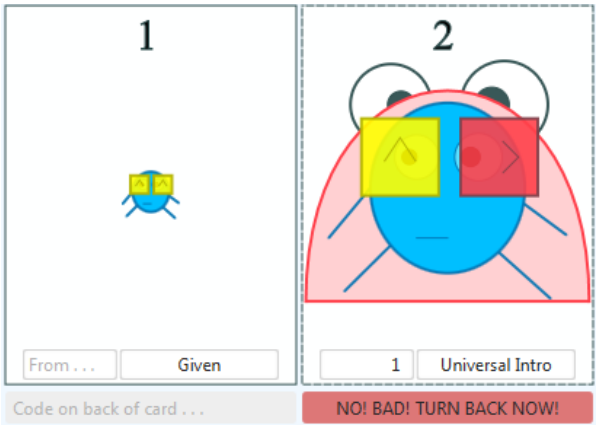
9.2.0: Putting Characters inside Domes

If a *Character* is the outermost Piece in some Card (that is, if it isn't inside of some Chambergon), you can enclose it in a Dome that matches one or more of its Lenses, so long as:

(1) the Lens(es) you are trying to bind is (are) not circular (only square Lenses can be bound by Universal Introduction),
(2) all the Character's Lenses that match the one you are trying to bind are changed—if they are changed—to the same new color,
(3) no Lens that matches the one(s) you are trying to bind has ever been taken out of Bowl (in that game) *even if the Lens wasn't freed for the first time by getting rid of that Bowl*, and
(4) you aren't in the middle of working out the consequences of assuming that a Lens matching the one you are trying to bind was free.

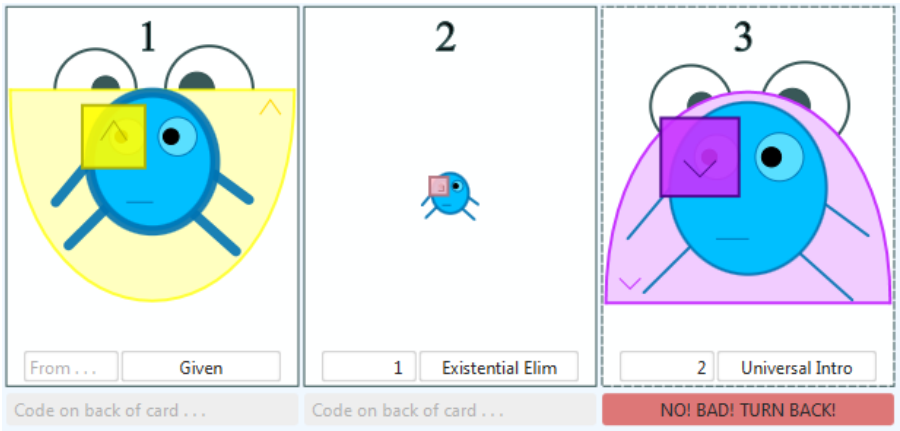
The first restriction is straightforward. Universal Introduction does not work on circular Lenses. It only works on square Lenses. This is because Domes represent universal quantifiers, while circular Lenses represent particular objects. The problem is that, at least in the modern way of thinking, a universal statement can never be justified by a particular statement. For example, "Keven plays football" does not justify the claim, "Everyone plays football."

The second restriction is also straightforward. It says that if you change the color of a square Lens using Universal Introduction, you have to change all the matching Lenses in that Card in the same way. For example, the following is wrong.



In Card 2, we use Universal Introduction to introduce a red Dome, and to change one of the square yellow Lenses from Card 1 to red. However, we didn't change both the yellow Lenses to red, and thus we failed. Universal Introduction applies "universally." If we use it to change one Lens in a Card, we have to change all the other matching Lenses in the same way.

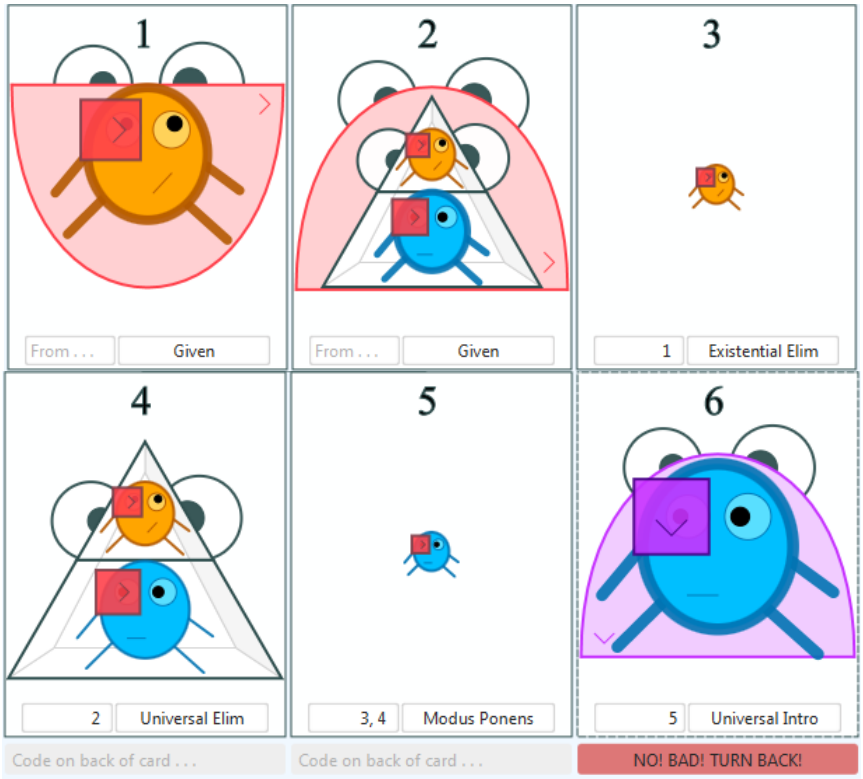
In contrast with the first and second restrictions, the third restriction is harsh. Basically, it means that any Lens that is free in a Card justified by the "Existential Elimination" rule can never go into a Dome. It has become unclean and will never be allowed into a major City. For example, this is not allowed:



The move from Card 1 to Card 2 is fine. We're just applying Existential Elimination to change the color of the square Lens in Card 1 to pink. The move from Card 2 to Card 3 is a problem, however. In it, we are trying to bind the pink Lens from Card 2 by using a purple Dome. The change in color from Card 2 to Card 3 is fine. But the pink Lens—no matter what color it ends up as—originally came from a Bowl. And that means we cannot use a Dome to bind it.

If you're interested in the logical reason for this, it goes as follows. Bowls are "existential quantifiers." They amount to the phrase, "there exists some." Domes, in contrast, are universal quantifiers. They amount to the phrase, "for all" or "for every." Now, imagine that Card 1 above stands for the sentence, "There exists some youngster who is bold." (The Bowl stands for, "There exists some youngster," while the blue Character stands for, "is bold," and the yellow lens stands for "the youngster.") If that's what Card 1 stands for, then Card 2 stands for the sentence, "Someone is bold." But that would mean Card 3 stands for, "Everyone is bold." But just because some youngster is bold doesn't mean everyone is bold.

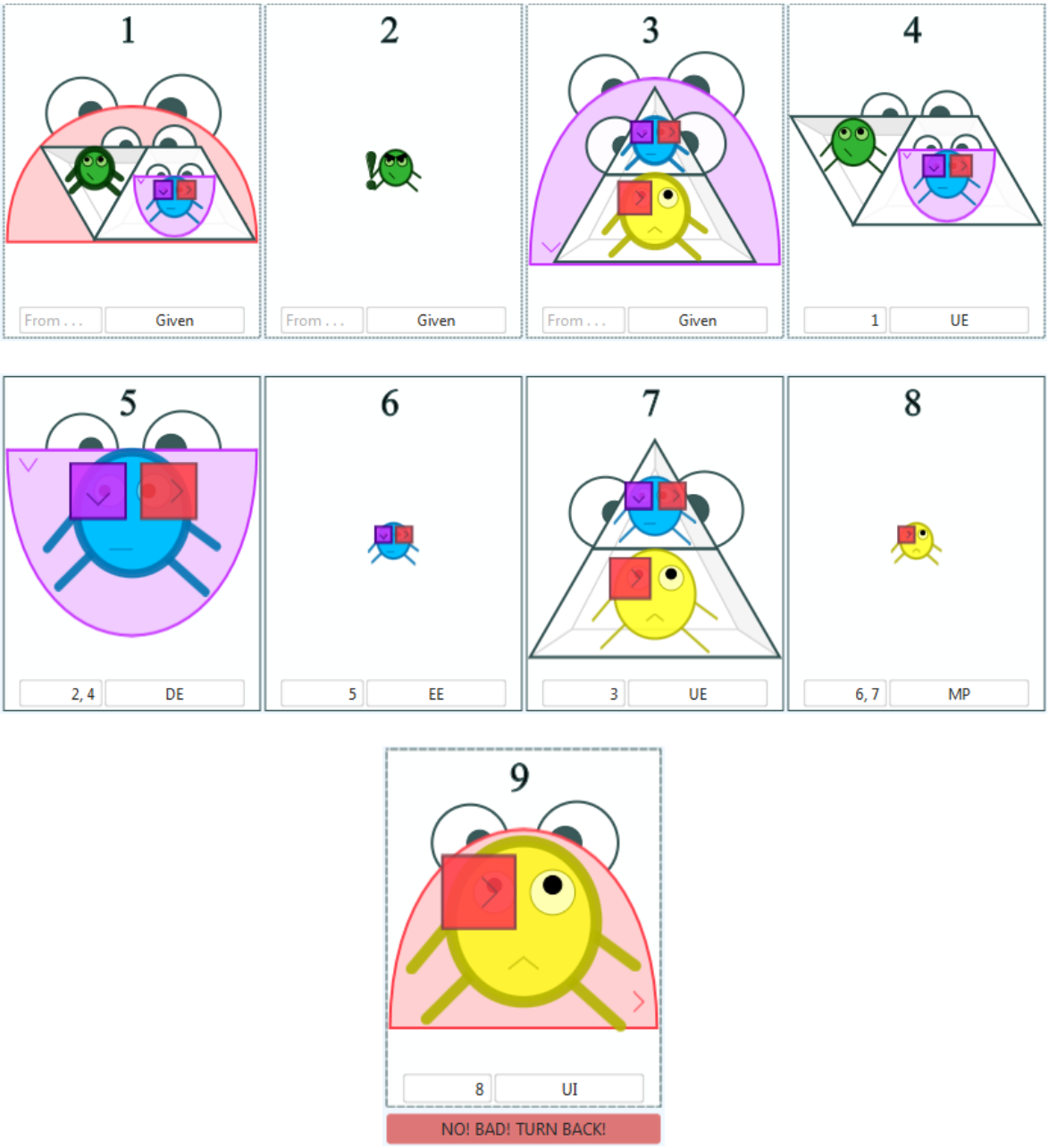
The problem with our third restriction for Universal Elimination, however, is that we don't just have to make sure we're not applying a Dome to a Card that was produced by Existential Elimination. We have to make sure we're not trying to use a Dome to bind a Lens that matches any other Lens that was freed by Existential Elimination. Take the following, for example.



Cards 1 and 2 are given, and we want to use them to extract the blue Character from the basement of the Pyramid in Card 2. To do this, we first have to unpack both Cards, taking their contents out of their Dome or Bow. Thus, Card 3 is an Existential Elimination of Card 1, and Card 4 is a Universal Elimination of Card 2. We can then use Cards 3 and 4 in a Modus Ponens move to create Card 5. This is all fine. The problem is if we want to then put that blue Character back in a Dome. After all, it started off in a Dome in Card 2, so it might feel most comfortable if it was back in familiar territory. Alas, however, we can't.

Card 6 is illegitimate because it tries to use a Dome to bind the red Lens from Card 5. But back in Card 3, there is also a free red Lens. It is just that Card 3 is justified by the "Existential Elimination" rule. That means red Lenses are "tainted," so far as Universal Introduction is concerned. Since one red Lens came from a Bowl, Universal Introduction thinks that all red Lenses come from Bowls. And that means Universal Introduction won't allow red Lenses to enter a Dome.

But things are even worse than this. Take a look at the following example.



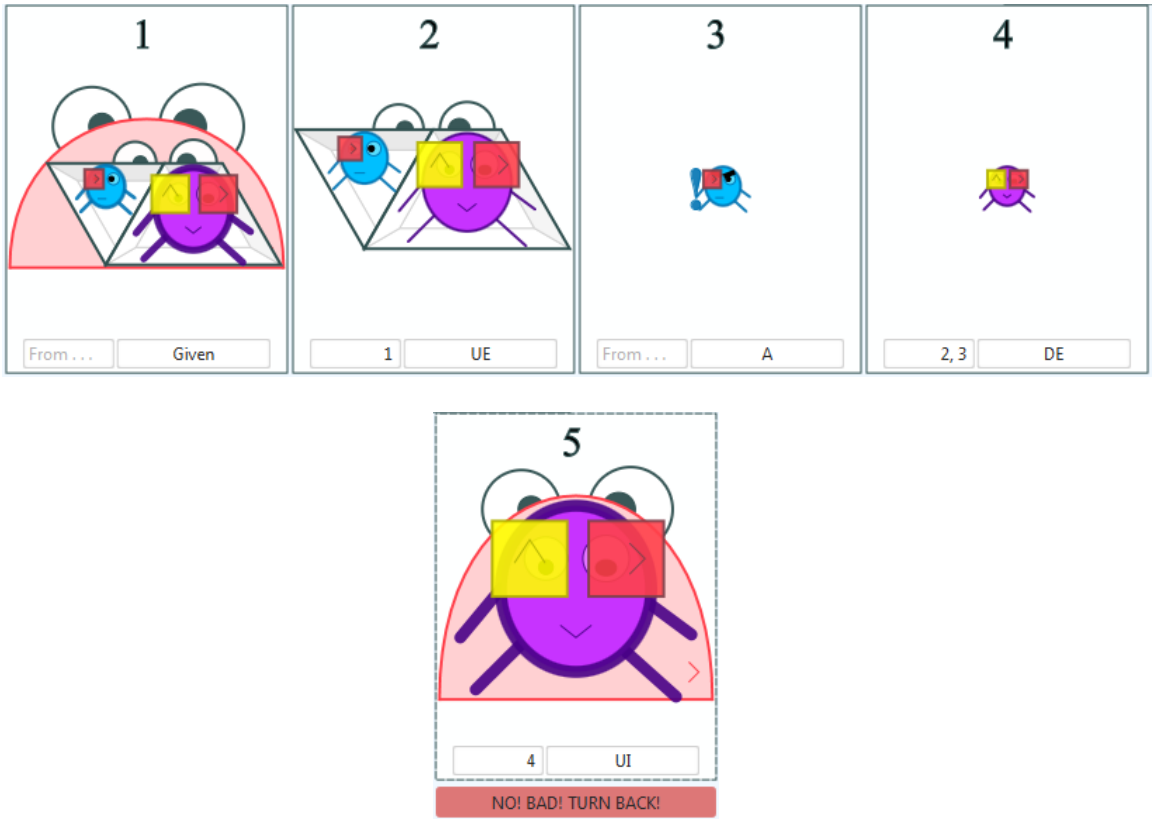
The real game being played here can be seen in Cards 2, 4, 6, 7, and 8. Card 2 frees the blue Character from the Paragon in Card 4. Card 6 frees the yellow Character from the basement of the Pyramid in Card 7. The rest of this game is just taking things out of, or putting things into, Bowls and Domes. But you will notice that Card 9 does not work. Why?

Card 9 tries to bind the red Lens from Card 8 in a matching Dome. The problem is that another red Lens is free back in Card 6, and that Card is justified by Existential Elimination. If you look at Card 5, in other words, you'll see that the red Lens in Card 6 used to be in a Bowl. Now, the red Lens was not bound by that bowl in Card 5. It was originally bound by a red Dome back in Card 1. But the fact that it was removed from a Bowl, and thus is free in a Card justified by "Existential Elimination" means that it can never be put back into a Dome. Or, rather, it means

that no red Lens can ever be put into a Dome *even if you change the color of the Lens when you put it in the Dome.*

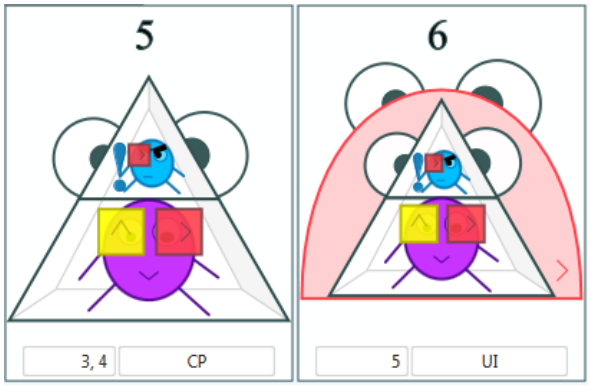
So, before you use Universal Introduction on a Card, you have to look back through all the Cards that appear before the new Card you want to create and look for any that are Existential Elimination Cards. Then, you have to check those Existential Elimination Cards to see what colors the free Lenses in them are. If the Lens you want to put in the Dome using Universal Introduction has the same color as a Lens that is free in one of the Existential Elimination Cards, you cannot use Universal Introduction to bind that Lens.

The fourth restriction on Universal Introduction is less harsh. What it says is that you can't just take out a loan of a Card containing a free Lens, and then stick that Lens inside a Dome. Take the following, for example.



The reason Card 5 doesn't work is that it is trying to bind the red Lens from Card 4. But look back at Card 3. That is an Assumption Card, and it too contains a free red Lens. Therefore, until we discharge that assumption (using Conditional Proof) we will not be able to use Universal Introduction to bind any red Lenses to Domes.

In other words, *if* Cards 5 and 6 looked like the following, then we would fine.



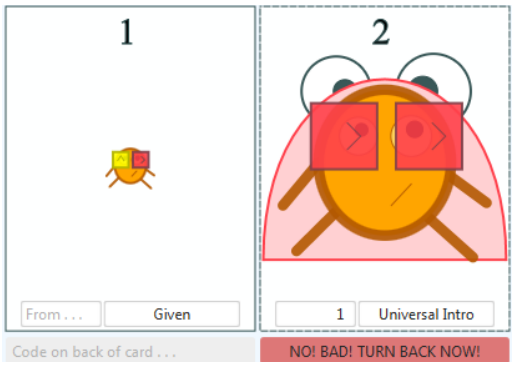
Notice that this version of Card 5 is a CP Card. It "discharges" the assumption from Card 3, and puts us outside the mini-game created by the Assumption Card. Since Card 5 closes the mini-game, it is technically outside it, and that means we can use Universal Introduction on it to bind its red Lenses.

So, in summary: if you are in the middle of a "mini-game" (between an Assumption Card and a CP Card), you can't use Universal Introduction unless the Lens you're trying to bind doesn't match one of the free Lenses in the Assumption Card.

Now, other than those four restrictions, Universal Introduction has the same restrictions as Existential Introduction, namely:

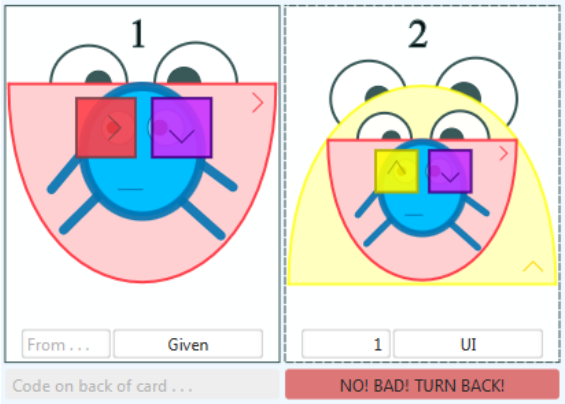
(5) You can't use Universal Introduction to change the Lens(s) in a Card to match another Lens that is already in the Card before you made the change,
(6) you can't use Universal Introduction to bind or change any Lenses that are already bound (i.e., that are already in another Dome or Bowl that they match), and
(7) you can't use Universal Introduction to change a Lens so that it becomes bound by a Dome or Bowl that it was already in.

Restriction 5 is the old, "If there's already a red Lens in the Card, you can't use Universal Introduction to turn a yellow Lens red" rule. Similarly, "If there's already a blue Lens in the Card, you can't use Universal Introduction to turn a purple Lens blue." Etc. So, that means the following move is illegitimate:



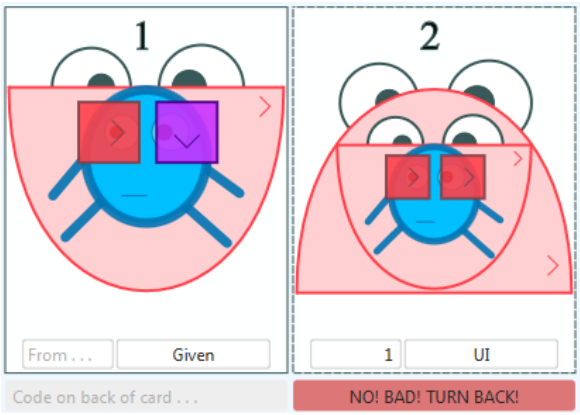
If we're going to bind the yellow Lens by adding a Dome, we can't change it to a red Lens and add a red Dome. We have to either leave it yellow and add a yellow Dome, or change it to some other color (like purple) and add a matching Dome. The Card already has a red Lens, so we can't use Universal Introduction to change the yellow Lens to red.

Restriction 6 is, similarly, our old friend, "If a Lens is bound by a Dome or Bowl (that is, if it is inside of a Dome or Bowl that has the same color as it), then it doesn't matter what you do outside that Dome or Bowl. You can't change a Lens that is bound by one Dome/Bowl simply by using Universal Introduction to add another Dome/Bowl to the Card." For example:



In this example, we try to change the red Lens from Card 1 to a yellow Lens by adding a yellow Dome. But the red Lens is already bound by a red Bowl, so it is not free to be changed. We can change the purple Lens—it is free in both Cards—but we cannot change the red Lens like we do in Card 2.

Restriction 7, finally, is another restriction we've encountered before. When you're trying to bind a Lens by changing its color to match a new Dome, you can't change it to match another Dome or Bowl it is already in. In other words, the following is wrong:



In this example, we try to bind the purple Lens in Card 1 by changing it to match the new red Dome we add in Card 2. But the problem is that in changing it to match the new Dome, we change it to match the Bowl it was already in. So we fail to bind it to the new Dome, and end up

binding it to the old Bowl. This is illegitimate. If we want to bind it to the new Dome, we have to use yellow, or purple, or something besides red.

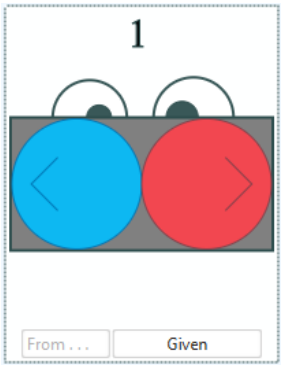
9.2.1: Putting Chambergons inside Domes

All the restrictions that we've just been talking about apply to using Universal Introduction on Chambergons too. If a *Chambergon* is the outermost Piece in some Card, you can enclose it in Dome that matches one or more of the square Lenses inside it so long as you follow Restrictions 1–7, discussed above.

9.3: The Power of Identity Elimination (A Landfill/Inference Rule)

This power has a stupid name. I don't like it. You could call it, "The Rule of Identity," but that sounds silly too. I guess we'll just have to get over it.

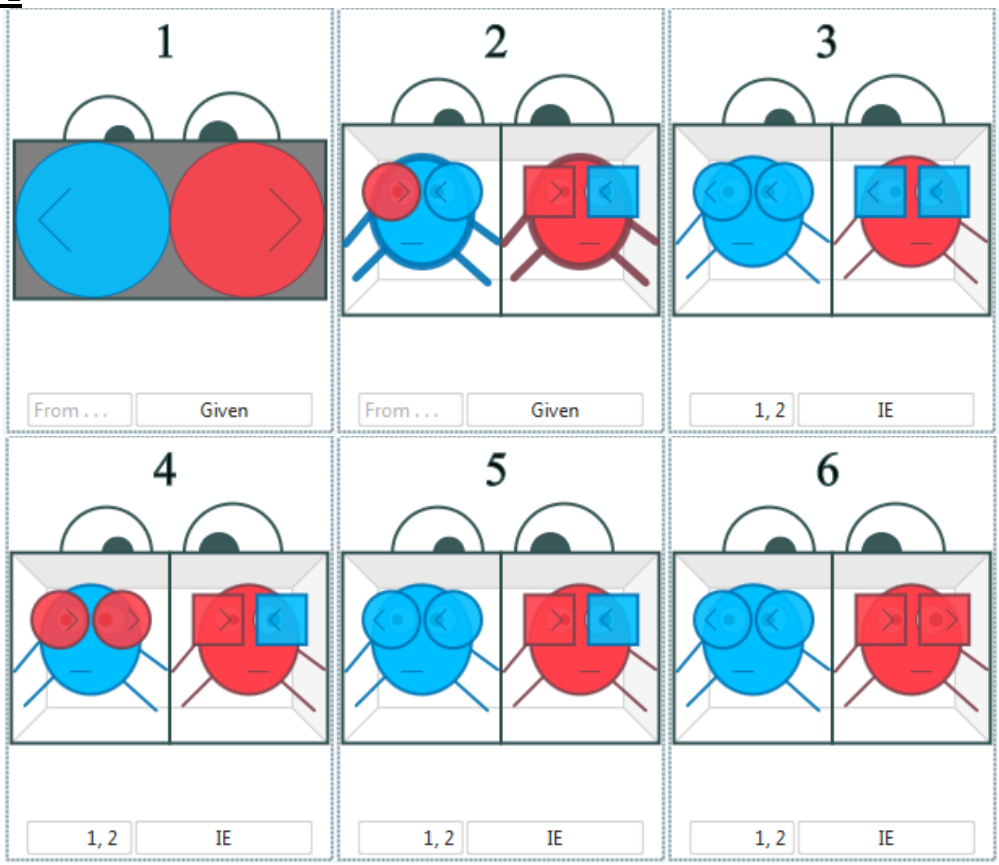
Identity Elimination is the power that allows us to change the colors of Lenses. For example, the following Coin . . .



. . . says that blue and red are two sides of the same coin. So, it gives us permission to change any blue Lens red, and any red Lens blue, so long as the Lens whose color we're changing is free (not bound). Furthermore, if we are given the Card above, we can change some of the free blue or red Lenses in another Card to the other color, or all of them (in that Card). We can change one and leave the others, change two and leave the others, change three and leave the others (etc.), or change them all (in the Card we're citing).

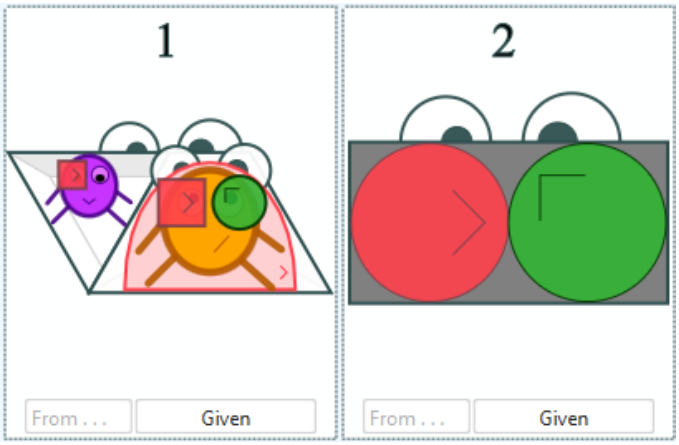
<u>Restrictions:</u> To be able to apply Identity Elimination, you have to have a Coin that is the outermost piece in a Card, and that Coin cannot have any Clubs. Furthermore, the Lenses whose color you want to change have to be free (that is, they cannot be inside a Dome or Bowl whose color they match). Finally, you cannot change any Lens to match a Dome or Bowl it is already in. (That is, you cannot use Identity Elimination to bind a Lens to a Dome or Bowl.)

Example 1



Take a look at Cards 1 and 2 to start. Card 1 tells us that red and blue Lenses are interchangeable. Card 2 then gives us two red Lenses and two Blue Lenses. Based on Card 1, we can change both of the red Lenses to blue (Card 3), or we can change one of the blue Lenses to red (Cards 4 and 6), or we can change one of the red Lenses to blue (card 5), etc. Since all the Lenses in Card 2 are free, and since they are all either red or blue, we can change them all, or leave them alone, as we like. Notice, however, that we cannot change the colors of the Characters. Coins only allow us to change the colors of Lenses.

Example 2



In this example, we are told that green and red are equivalent. Card 1 gives us two red Lenses and one green Lens. So, what does Identity Elimination produce? The red Lens belonging to the purple Character is free, and thus we can change it to green. The red Lens belonging to the orange Character is bound by the red Dome, and thus cannot be changed. The green Lens belonging to the orange Character is free, and thus could be changed. However, we would have to change it to red, which would bind it to the red Dome, which is not allowed.

Therefore, the only possibility is the following:

9.4: A Note about Lens Colors and Shapes for UI and IE

In algebra, lowercase letters from the beginning of the alphabet (especially "c") are used as constants, and lowercase letters from the end of the alphabet (especially "x") are used as variables. Thus, you have equations like . . .

$$ax^2 + bx + c = 0$$

. . . in which "a," "b," and "c" represent unknown numbers that are constant (held fixed), while "x" represents an unknown number that is variable (allowed to change from one value to the next, beginning at negative infinity and extending all the way to positive infinity).

In symbolic logic, there is the same distinction. Lowercase letters from the beginning of the alphabet (i.e., “a,” “b,” “c”) are treated as constants, while lowercase letters from the end of the alphabet (i.e., “x,” “y,” “z”) are treated as variables. Where the constants end and the variables begin in the alphabet depends on who you’re talking to, but the basic idea is that constants are from the beginning (or first half), and variables are from the end (or second half).

In Chambergon Battle Logic, we use colors rather than letters, so we try to abide by the normal distinction between constants and variables by thinking about the first letter of the color names in English, or about the letter key we use to create them when two colors share the same first letter. Arranged in alphabetical order, we have:

brown = “a”
blue = “b”
green = “g”
orange = “o”
purple = “p”
red = “r”
yellow = “y”
pink = “z”

Having eight letters, we divide them in half and assign the first half to the constants, and the second half to the variables.

Constants:	brown = “a”
	blue = “b”
	green = “g”
	orange = “o”
Variables:	purple = “p”
	red = “r”
	yellow = “y”
	pink = “z”

We then assign a different Lens shape to each type.

Constants:	brown = “a”	Circular Lenses
	blue = “b”	
	green = “g”	
	orange = “o”	
Variables:	purple = “p”	Square Lenses
	red = “r”	
	yellow = “y”	
	pink = “z”	

That means circular Lenses *should* be brown, blue, green, or orange, and square Lenses *should* be purple, red, yellow, or pink. And since Domes and Bowls only bind square Lenses, Domes and Bowls should also only be purple, red, yellow, or pink.

Constants:	brown = “a” blue = “b” green = “g” orange = “o”	Circular Lenses	
Variables:	purple = “p” red = “r” yellow = “y” pink = “z”	Square Lenses	(Domes and Bowls)

Now, you will have noticed that I don’t always follow the guidelines above. Sometimes square Lenses have a circular color, and sometimes circular Lenses have a square color. However, it is especially important to follow the guidelines above when you are using Universal Introduction and Identity Elimination. You cannot use Universal Introduction to bind a Lens that is circular. Nor can you use Universal Introduction to bind a Lens that has a “circular” color. And if you use Identity Elimination to change a Lens from a circular color to a square color (or vice versa), you have to change the shape of the Lens too.

So, if you are playing the game and find that Universal Introduction or Identity Elimination aren’t working in places where you think they should work, use the chart above to make sure that you are following the proper guidelines for Lens color and shape.

Note: Coins have a square outline, but appear to contain two circular Lenses. There’s nothing you can do about their shape, so don’t worry about it. However, do note that when you use a Coin to change some Lens’s color (i.e., when using Identity Elimination), you have to change the shape of the Lens if you are moving the Lens from one “color group” (e.g., constants) to the other (e.g., variables).

A final note: In logic, “constants” stand not for “constant numbers,” but for proper names (e.g., “Sally,” “John”) or “indexicals” (e.g., “this,” “that,” “the tree over there”)—terms that pick out a single, actually-existing person, place, object, entity, etc. Variables, similarly, do not stand for numbers whose values can change, but for vague or generalized names (e.g., “something,” “someone,” “everything,” “everyone”).

9.5: Decoding Games (The Other Kind of Encoding Games)

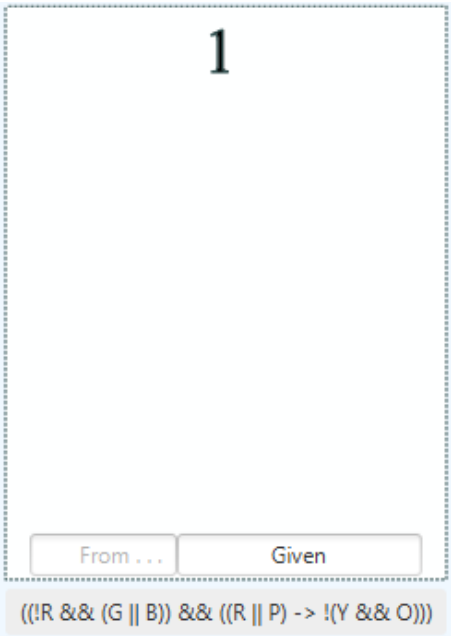
Encoding games come in two flavors. The normal kind are those in which you actually encode the icons and pictograms into code, using the box below each card. The other kind are actually *decoding* games. You are given the code for each card, and then have to fill each card with the right pieces. What’s more, you have to figure out what to put in the “justification boxes” too (the “From” and “Using” boxes). These “decoding” encoding games pose a special challenge, therefore, and we should take a minute to talk strategy.

First, the easiest Card to decode in a decoding game is always the last Card. After all, the last Card is the one that completes the game, and thus its contents will be identical to the contents in the “Goal” box. So, just copy and paste the contents of the goal into the final Card.

From there, you may be able to work backwards, copying and pasting pieces from the final Card into other Cards. However, in many cases, what's in the final Card will only have a very slim overlap with what's in all the others. So, the second strategy point we need to make is this: For each Card, find its "main connective," and work from there.

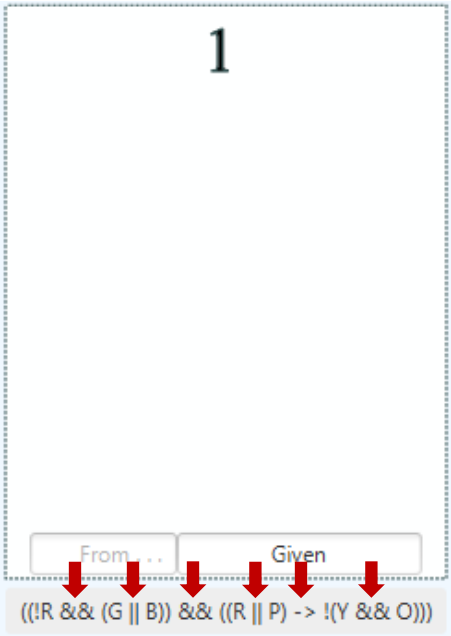
9.5.0: Decoding Propositional Logic Games

The main connective in a line of Code is the symbol that holds the whole line together. Put in another way, it is the symbol to which the outermost parentheses correspond. But let me give you an example and a step-by-step process.



Step 1: Identify the connectives.

> This step is not strictly necessary, but it is good to get into the habit of looking at a line of code and immediately focusing on the connectives in it.



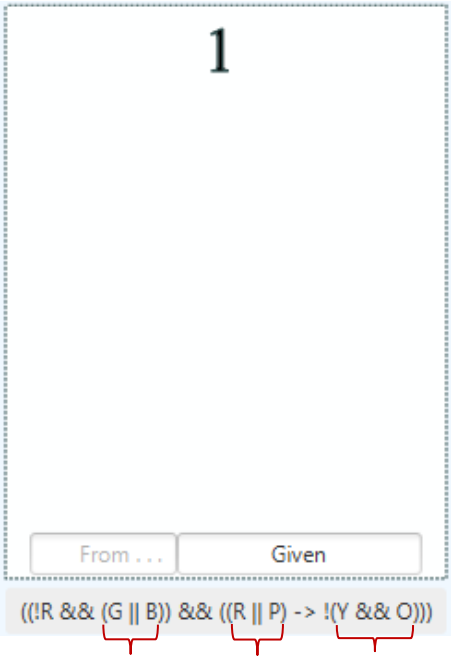
Step 2: Find the main connective (the connective that belongs to the outermost pair of parenthesis.)

Each pair of parentheses contains one open-parenthesis—"("—and one close-parenthesis—").”

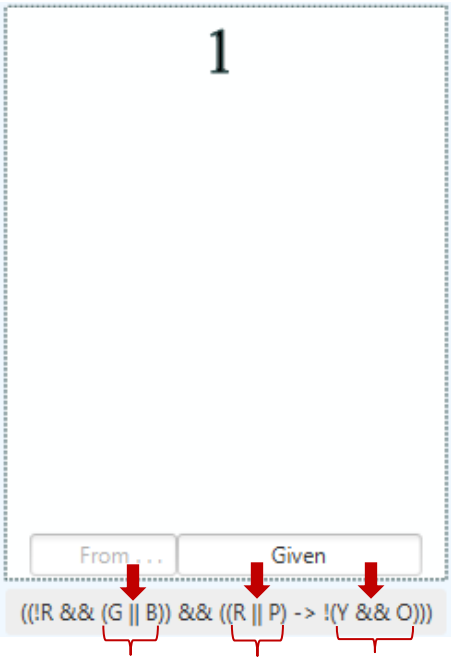
Scan through the line of code, noting when you reach a parenthesis. Sometimes you'll encounter multiple open-parentheses "in a row." (That is, each parenthesis you scan over in a stretch of code will be an open-parenthesis.) Sometimes you'll encounter multiple close-parentheses "in a row."

What you're looking for, however, are when you pass an open-parenthesis and the next parenthesis you reach is a close-parenthesis. When that happens—when one parenthesis is an open-parenthesis and the very next one you come to is a close-parenthesis—you have found an "innermost" pair.

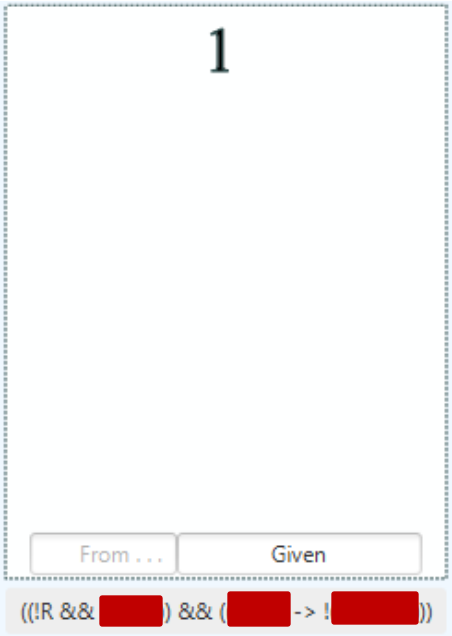
To find the connective that belongs to the outermost pair of parentheses, you need to first mark off all the innermost pairs.



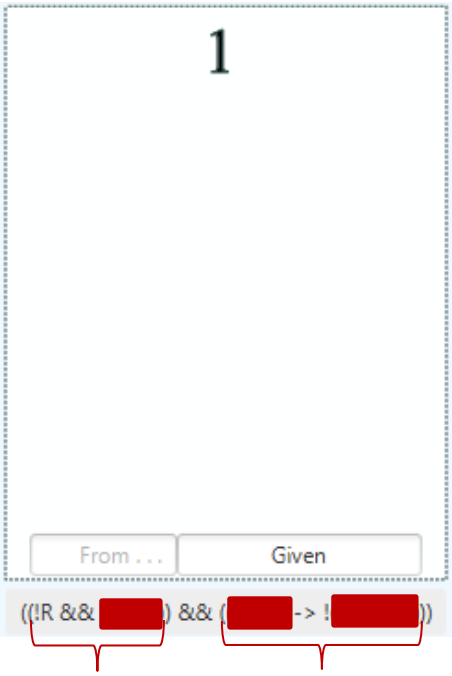
Once you've found the innermost pairs of parentheses, you can tell which connectives they belong to very easily. The connective a pair of parentheses belongs to is the only connective in that pair.



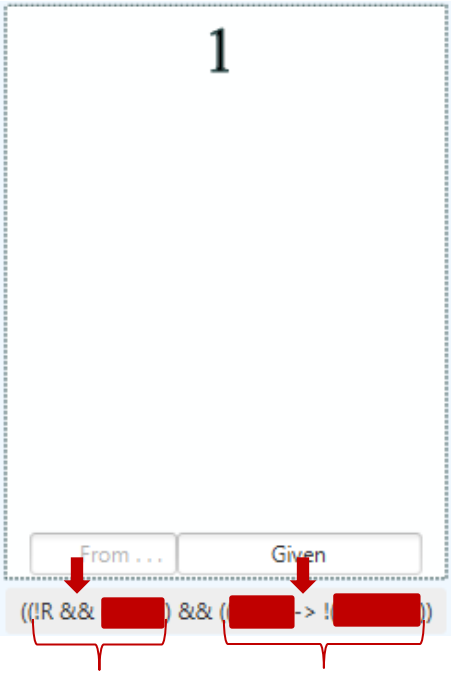
Now, "blank out" or "cover over" or "ignore" the innermost parentheses and everything inside them.



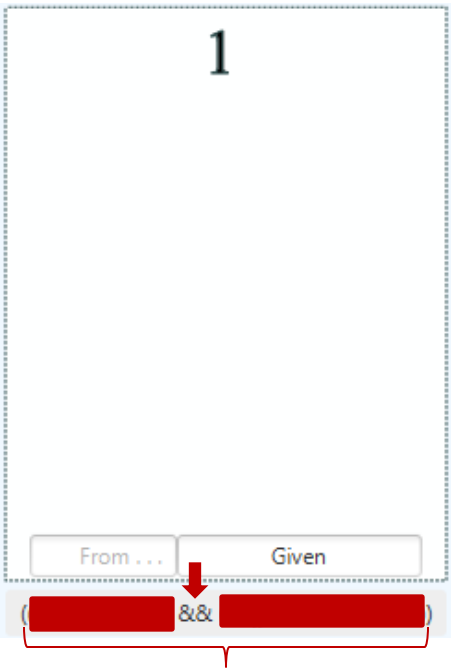
Now, run through the code again, looking for the "new" innermost parentheses. That is, ignoring the pairs of parentheses you just marked, look for the places where you encounter an open-parenthesis, and the next one you meet is a close-parenthesis.



Now—just as before—mark the connectives that are inside the latest pairs of innermost parentheses.

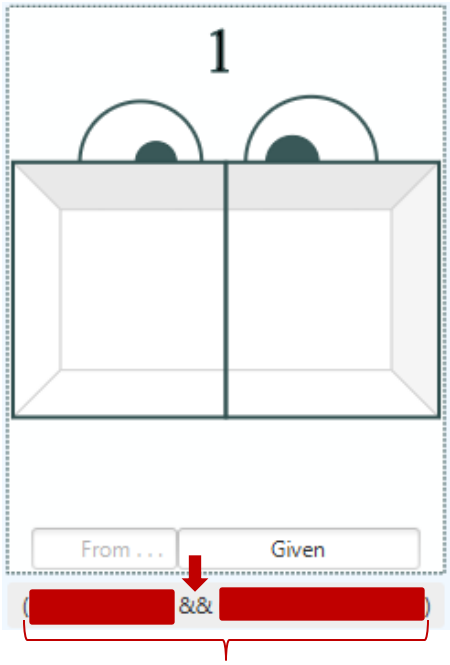


Now, block off the latest “innermost” pairs of parentheses and repeat the process.



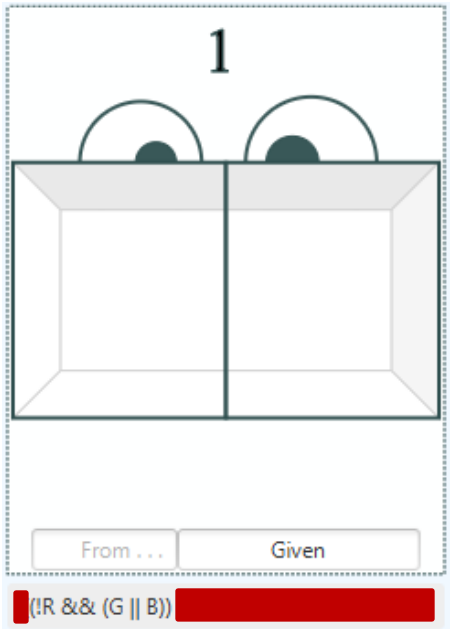
When the “innermost” pair of parentheses is actually the *outermost*—because you’ve already identified and blocked off all the others—you’ve found the main connective.

Step 3: Drop the piece that corresponds to the main connective into the Card.



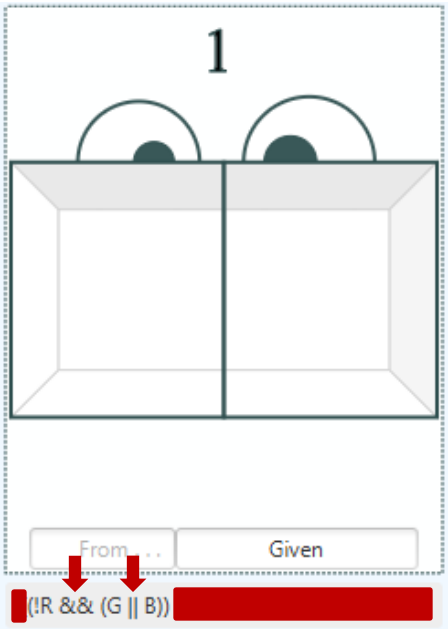
If the outermost pair of parentheses had a Club or two in front of them, we'd add those as well. But since they don't in this case, we move on.

Step 4: Block off the main connective and everything to its right, as well as the outermost parenthesis on the left.



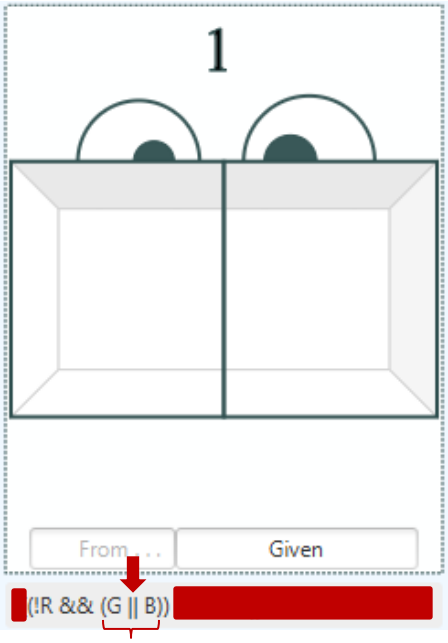
Step 5: Repeat Steps 1 through 3 for the code to the left of the main connective.

Step 1 (for Step 5): Identify the connectives.

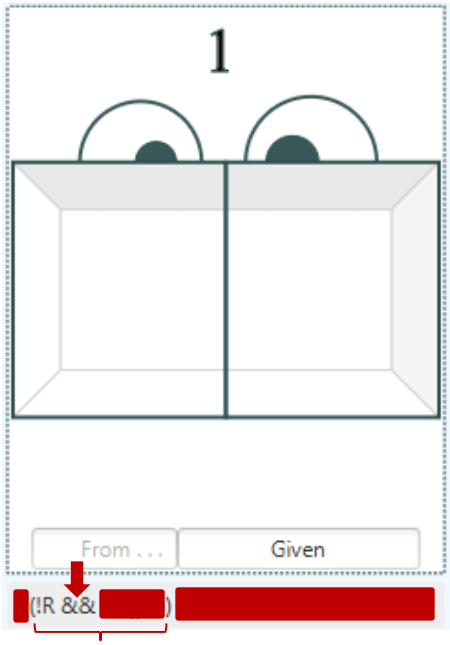


Step 2 (for Step 5): Find the main connective (the connective that belongs to the outermost pair of parenthesis.)

First, find the innermost pairs of parentheses.

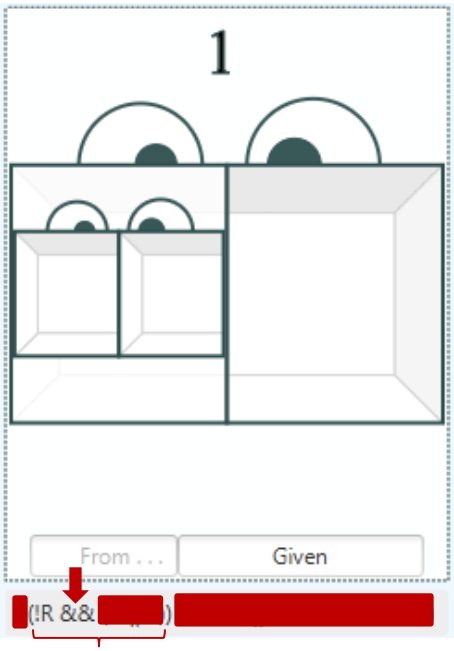


Second, block off the innermost pairs of parentheses and find the next-most-inner pairs of parentheses.



Third, once the "innermost" pair of parentheses is actually the *outermost*—because you've already identified and blocked off all the others—you've found the main connective.

Step 3 (for Step 5): Drop the piece that corresponds to the main connective for the left "half" of the code into the left-hand chamber (or the top chamber, if the piece you're filling is a Pyramid or Pulsar).



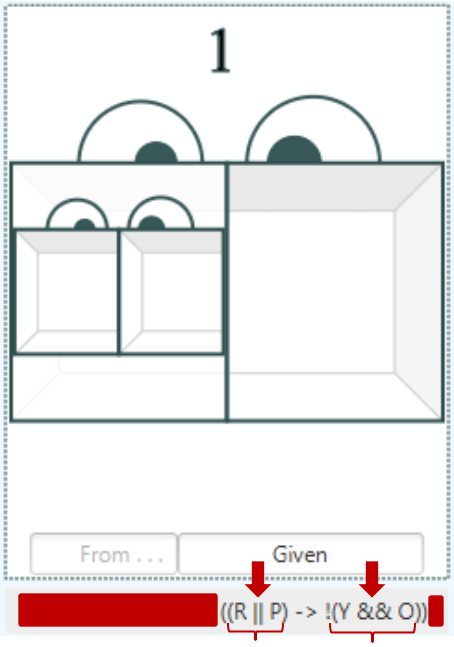
Step 6: Unblock the code to the right of the right of the code's main connective, and block the code to the left of the main connective instead. (Leave the outermost parenthesis on the right blocked.)

Step 7: Repeat Steps 1 through 3 for the right half of the code.

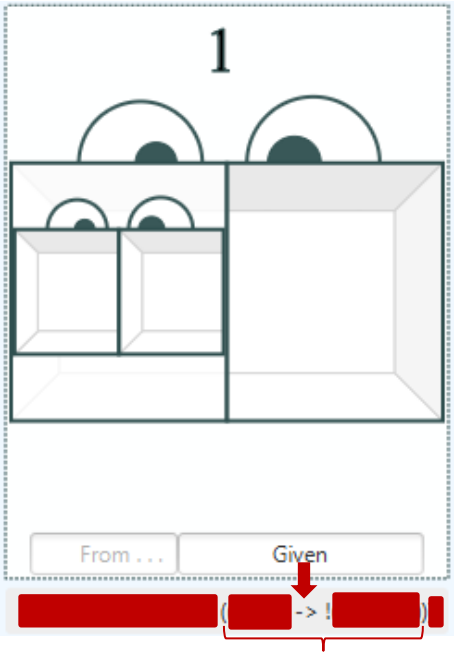
Step 1 (for Step 7): Identify the connectives.

Step 2 (for Step 7): Find the main connective (the connective that belongs to the outermost pair of parenthesis.)

First, find the innermost pairs of parentheses.



Second, block off the innermost pairs of parentheses and find the next-most-inner pairs of parentheses.

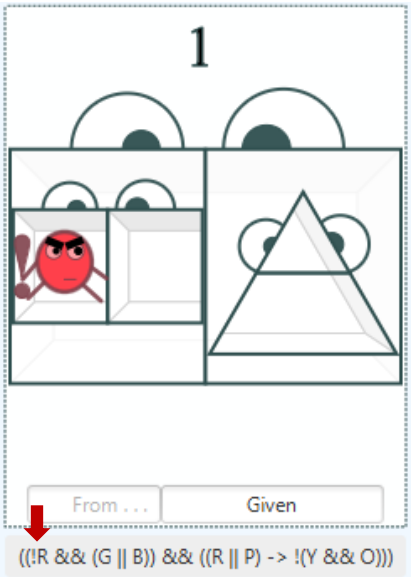


Third, once the "innermost" pair of parentheses is actually the *outermost*—because you've already identified and blocked off all the others—you've found the main connective.

Step 3 (for Step 7): Drop the piece that corresponds to the main connective for the right "half" of the code into the right-hand chamber (or the lower chamber, if the piece you're filling is a Pyramid or Pulsar).

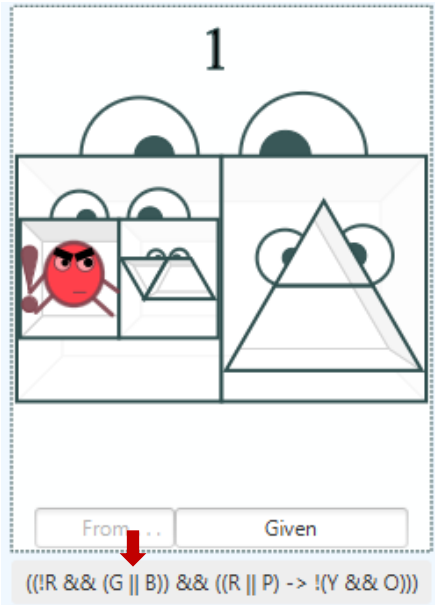
Step 8: Repeat Steps 1 through 7 for the left and right "halves" of the code.

Or, in other words, the left "half" of the code is "(!R && (G || B)." The "&&" marks the middle line of the Parcel in the left chamber. To its left is "!R." This tells you what to put in the left chamber of the Parcel in the left chamber.



To the right of the "&&" in the left half of the code is "(G || B)." This tells you what to put in the right chamber of the Parcel on the left. You look for its main connective in the

same way as before. Fortunately, this time there is only one connective, so we drop the piece that corresponds with it into the right chamber of the Parcel we're currently working on.



Now we have a choice. Either we keep working on the Paragon we just created, or we move to the other half of the code. I suggest we fill in the Paragon.

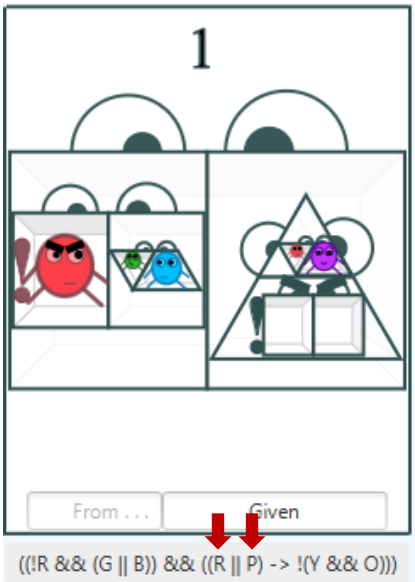
We then have to take on the right half of the code. On the right side of the main connective "&&," we have "((R || P) -> !(Y && O))." This corresponds to the Pyramid, as we've already seen, since its main connective is, "->." To the right of the "->" is "(R || P)," so our job is to find its main connective. That, fortunately, is easy once again, since

there is only one connective. So, we insert the piece corresponding to "||" into the upper chamber of the Pyramid.

The Paragon goes in the upper chamber of the Pyramid since the "||" is to the left of the "->." To the right of the "->" is "!(Y && O)." Its main, and only connective, tells us to put a Parcel into the basement of the Pyramid.

However, the fact that there's an "!" attached to the front of the parentheses belonging to the "&&" tells us this Parcel needs a Club.

And now we have all the connectives in the code covered. From here, it's just Characters. We need an "R" and a "P" in the Parcel.

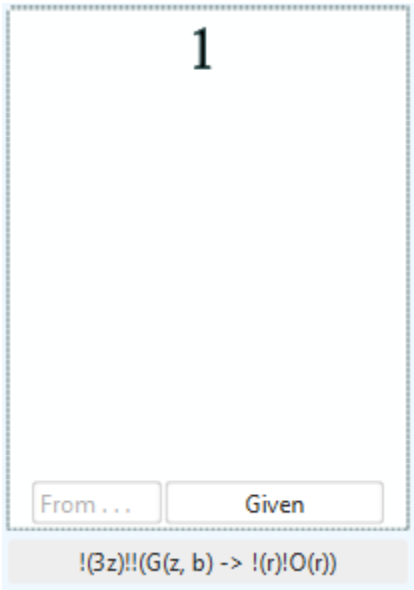


And we need a "Y" and an "O" in the Parcel.

9.5.1: Predicate Logic Games

"But what about predicate logic games?" you ask.

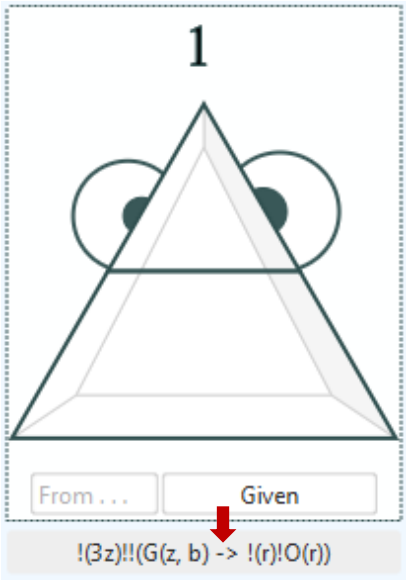
"They're the same," I respond. You look for the main connective of the whole line, then the main connective on the left, then the main connective on the right, and so on. For example:



In this Card, the main connective is the arrow. In fact, it's the only connective.

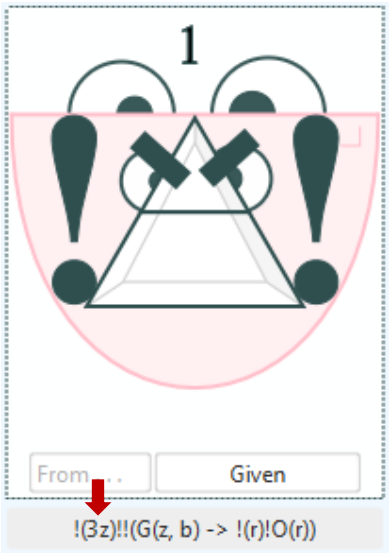
"But aren't the Clubs connectives? And aren't the '(3z)' and '(r)' connectives?"

No. They're operators, not connectives (see Intro.1.11 on p. 19). What we want are the connectives, and there's only one in this case. So, we drop a Pyramid into the Card.



The pair of parentheses belonging to the arrow, however, have a number of symbols attached to them on the left. Specifically, they have "!(3z)!!" attached to them. So, "work from the inside, out," moving from right to left. Add the two Clubs first.

Then add the "(3z)."



But the "(3z)" has an "!" attached to it, so we add a Club to the Bowl.

Now that we've taken care of the externals, we can move to the contents. To the left of the arrow is "G(z, b)." This tells us we need a Green character in the attic of the Pyramid, and that the Character should have two lenses (one pink and one blue).

We make the pink Lens square, and the blue Lens circular, because of the conventions we discussed in the previous section (p. 331).

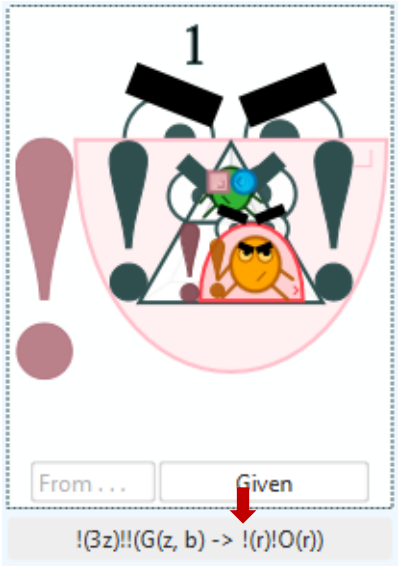
Having filled the attic, we must now fill the basement using the code to the right of the arrow. How should we interpret "!(r)!O(r)"? When we have no connectives to latch onto, we look for capital letters.

The only capital letter in "!(r)!O(r)" is "O." So we drop an orange Character into the basement of the Pyramid.

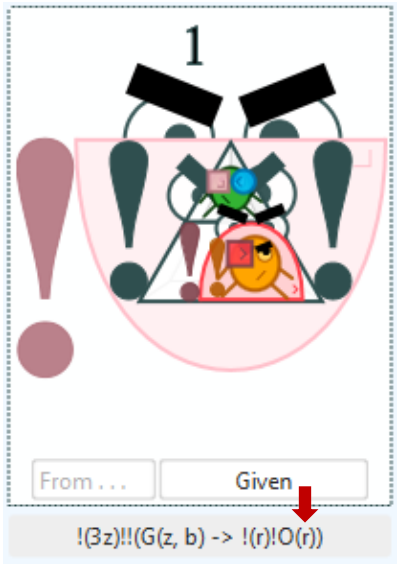
The "O" has a number of things tacked onto it, both on the left and on the right. On the left, it has "!(r)!," so we work from the inside, out (from right to left). First, we add a Club.

Then we add the “(r).”

And finally, we add the “!” attached to the “(r).”



That takes care of the "!(r)!" on the left of the "O." On the right of the "O," we have another "(r)." Since it's on the right, rather than on the left, of the "O," it represents the Lens that orange Character is wearing or holding, not the Dome that contains the Character. So, we give the Character a red Lens.



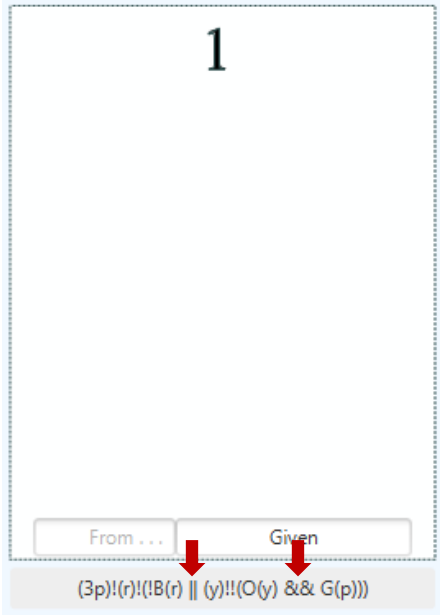
That, in general, is the method you should use till you get used to these games. Find the connectives and drop in the Chambergons they represent. Then attach any Clubs and Domes or Bowls. Then, fill the Chambergons with Characters, and after you've dropped in the Characters, attach their Domes, Bowls, and Clubs (using the symbols to the left of the Character's symbol) and their Lenses (using the symbols to the right of the Character's symbol). Or, if the Chambergons contain more Chambergons, or fill the Chambergons with other Chambergons, then attach their Domes, Bowls, and Clubs.

Think of the Chambergons and Characters, in other words, as the skeleton, and the Clubs, Domes, Bowls, and Lenses as the flesh. You've got to construct the skeleton before you can flesh things out.

9.5.2: More Complex Predicate Logic Games

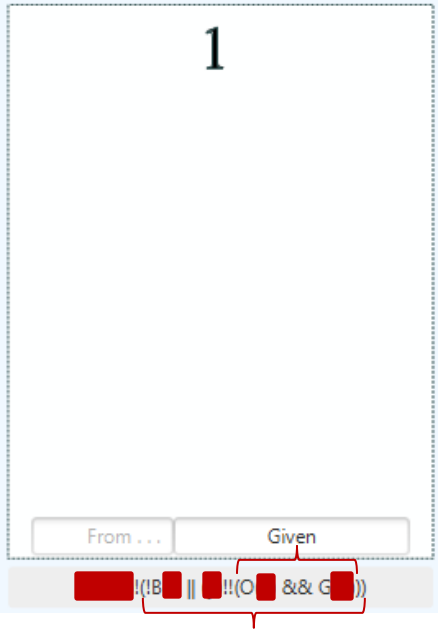
Take a look at the following Card's code. How should we decode it?

First, we have to identify the connectives, and decide which one is the main connective.



To figure out whether the "||" or the "&&" is the main connective, we have to figure out which is innermost and which is outermost. So, ignoring the parentheses that mark the Domes, Bowls, and

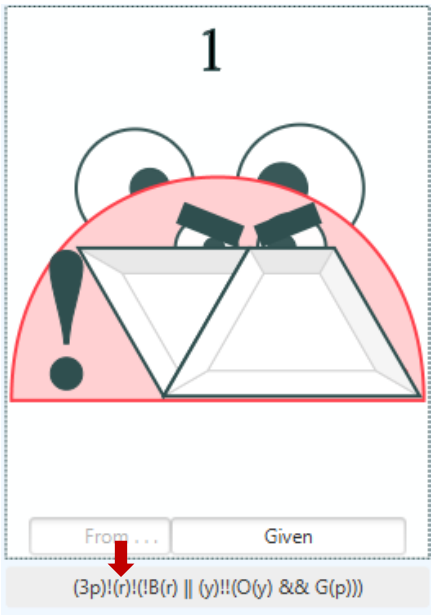
Lenses, we look for the pair of parentheses that do not contain any other pair (i.e., the pair where there is no other parenthesis between them, other than the parentheses that mark the Domes, Bowls, and Lenses).



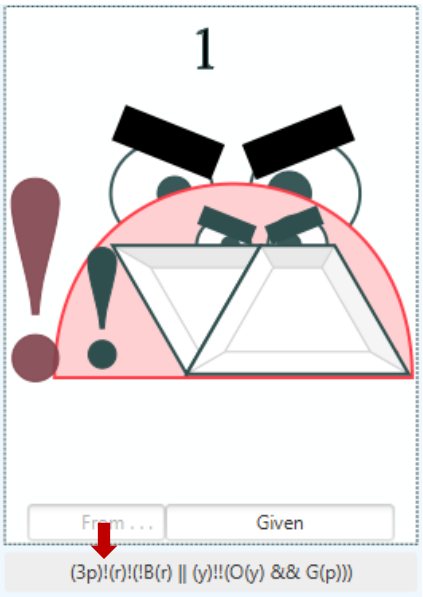
So, the "&&" is inside the pair of parentheses that contains no other pair, which makes that pair "innermost." But the main connective belongs to the outermost pair of parentheses, which as we now see, belongs to the "||." This means we'll drop a Paragon into the Card first.

Now we work from the parentheses for the "||" outward, moving from right to left. First, we come to a Club.

Then we come to a red Dome.



Then we come to another Club.



And then we come to a purple Bowl.

Having reached the end of the things “tacked onto” the Paragon, we can now turn to the job of filling it. On the left side of the “||” we have “!B(r).” This represents a blue Character with a Club and a red Lens.

On the right side of the "||" there is the "&&," with its parentheses, the things attached to the parentheses, and everything inside the parentheses. As always, we start with the connective.

Then, as always, we add whatever is tacked onto the outside of the parentheses for the connective, moving from right to left.

First, we come to a pair of Clubs.

Then, we come to a yellow Dome.

Having taken care of everything tacked onto the parentheses belonging to the "&&," we now turn to the contents of the parentheses.

On the left side of the "&&" we have "O(y)." This tells us we need an orange Character with a yellow Lens in the left Chamber of the Parcel.

Finally, on the right side of the "&&," we have "G(p)," which tells us we need a green Character with a purple Lens.

And we're done.