Modal Logic
A Practical and Rigorous Introduction

Adam Bjorndahl and Chris Grossack

November 22, 2019

Contents

add a preface, and acknowledge everyone who's helping revise

Part I

Core Concepts

Chapter 1

Introduction

When talking about the real world, we often use *modifiers* to subtly change the meaning of what we are saying. To give a purely linguistic example, consider the sentences "It will rain tomorrow" and "It might rain tomorrow". The first expresses certainty, whereas the second expresses uncertainty. Indeed, we could also use "Valerie knows it is raining" and "Valerie thinks it might be raining" in a similar way. The first sentence expresses an amount of certainty that the second sentence lacks. How can somebody judge the truth of a sentence like this? In every case, the primitive idea "It is raining" is a very easy thing to check the truth of. However, "Valerie knows it is raining" might be very hard to prove. **Modal Logic** takes up the study of these *modifiers*, officially called **Modalities**, and gives a formal framework for reasoning about how they change the truth of a sentence. Of course, modalities show up in many guises in addition to the linguistic examples above. Modalities are also prevalent in branches of philosophy, mathematics, and computer science, so let's talk about some examples.

In programming language theory, a branch of computer science dedicated to the design of programming languages, it is often desirable to have a *type system* which guarantees certain correctness properties of code before the it is executed. The ever-elusive dream in this field is to have a programming language where if your code compiles, it must be correct. Unfortunately, this often requires a lot of verbosity on the part of the programmer, and balancing programmer quality-of-life with guaranteed program safety is a big topic in the area. *Side effects*, the things a code does to actually interact with a world, are particularly difficult to formalize in a type system. Thankfully, there has been a lot of research recently in the use of modalities to model these side effects. There is also reserach into using Temporal Logic to model program execution. This allows us to automatically prove certain programs match their specification, in other words, to prove that our code is correct automatically.

In philosophy, as in linguistics, it is useful to take possibly ambiguous sentences and make them precise. This allows us to reason formally about the truth of arguments. However, many arguments in philosophy use concepts like *knowledge* and *obligation* and *morality*. In trying to make arguments involving these terms, it is often useful to rephrase them in the language of modal logic. Here, there is no ambiguity, and so it is immediately apparent whether a proof is actually a proof.

In mathematics, modality finds uses primarily in mathematical logic, though its uses are often varied and creative. In understanding Gödel's seminal incompleteness theorems, an important notion is that of *provability*. Surprisingly, the notion of being provable is a modality, and this has shed some light on Gödel's theorems, and related ones. Modalities also arise in topology and in a type theoretic foundation for mathematics as *closure operators*, and these modalities are intimately related. This is a reflection of a much deeper link between geometry and logic, related to category theory, which is very efficiently studied via these related modalities. In Homotopy Type Theory, modalities are used to represent *truncation*, a way of simplifying a complicated infinite structure by only looking it its first finitely many substructures. Finally, set theoretic *forcing*, the act of custom building a set theoretic universe to prove a property about the axioms of set theory itself, has modal content.

Of course, we still have not said what makes something a modality. The characterizing feature is that modalities are not *truth functional*. As an example, let K be the modality "Valerie knows". Thus, if φ means "it is raining", then $K\varphi$ means "Valerie knows it is raining". A truth functional operator only depends on the truth value of its input. In that way, it is a *function* of the *truth* value of its input. So there are only four truth functional operators of one input: $T\varphi$ which is always true, $F\varphi$ which is always false, $I\varphi$ which is true whenever φ is, and

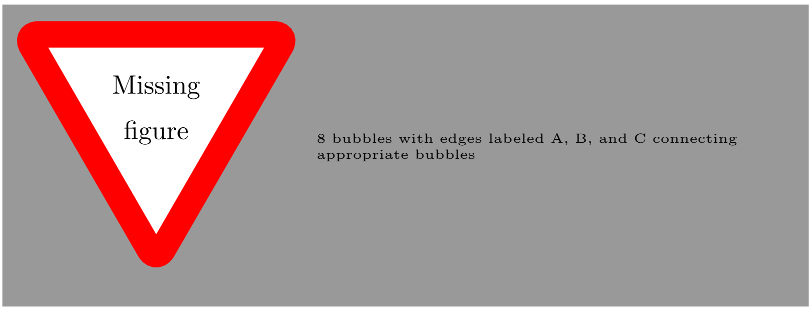
$N\varphi$ which is true whenever φ isn't. Modalities are useful because they care about more than just the truth value of their input. For instance, If φ is "it is raining by Valerie" and ψ is "it is raining in Bermuda", and let's say they're both true. If Valerie is not in Bermuda, then $K\varphi$ is true, but $K\psi$ might not be, even though φ and ψ are both true. It is incredibly common for the truth of something to depend on more than just the truth of its input, and this is the reason for the ubiquity of modalities in certain areas of study.

This book is meant as a rigorous treatment of Modal Logic, formalized mathematically with *Kripke Semantics*. We will prove fundamental results for a variety of modal logics, and ideally have fun doing so. Typically examples will come from Epistemology (where modal formulas are interpreted as what Valerie knows) or Temporal Logic (where modal formulas are interpreted as what the future will be like) but there will be others along the way. At the end, we will work with two modal logics which are currently the subject of active study. With all this said, let's dive in!

1.1 A Puzzle

> Three people are sitting in a room, each wearing either a red or blue hat.
>
> A moderator comes in and says "At least one of you is wearing a red hat."
>
> Each player can see the other players' hats, but not their own.
>
> Players then take turns either announcing their hat color (if they know it), or announcing that they do not know their hat color. The first person to know their hat color wins!
>
> Will this game always end? Further, can everyone figure out what their hat color is eventually?

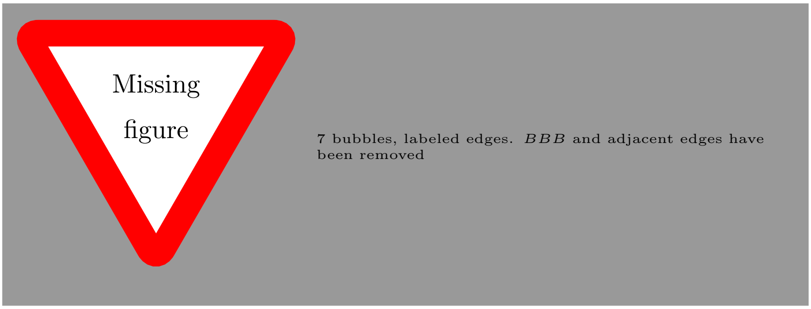
Somewhat surprisingly, the game always has a winner! Though it is not possible, in general, for every player to know their own hat color. Perhaps more surprisingly, the moderator's announcement is extremely important! Let's talk about why.



First, let's translate the game into logical terms. Let's name the three players Alyss, Bob, and Cam. Then there are 8 "possible worlds", one for each assignment of hats to players. We will denote the world where Alyss's hat has color X, Bob's has color Y, and Cam's has color Z by XYZ. For example, the world RRB is the world where Alyss and Bob have red hats, and Cam has a blue hat.

We will add three modalities, $\widehat{A}$, $\widehat{B}$, and $\widehat{C}$ representing what Alyss, Bob, and Cam *consider possible*. For example, Alyss cannot tell RRR and BRR apart. So if the real world is RRR, Alyss will *consider it possible* that BRR is the real world. Symmetrically, if BRR is the real world, Alyss will *consider it possible* that RRR is the real world. We can represent this by adding edges between worlds which Alyss, Bob, and Cam cannot distinguish.

Now we are equipped to solve the problem. The above graph has a node for every *possible world*, and we will remove nodes from the graph as we learn more information about the world. For example,

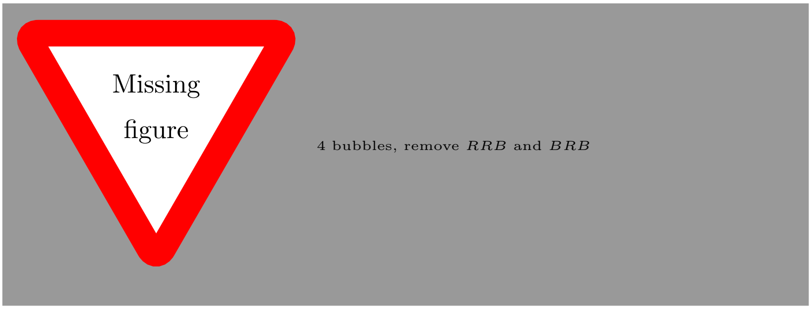


since the moderator announced that BBB is not a possible world, we can consider the following graph instead.

So now, let's start with Alyss. If she sees two blue hats, then she *knows* she must be wearing a red hat. This is modeled in the above diagram by the lack of $\widehat{A}$ arrows coming out of the RBB vertex. If instead, she does not see two blue hats, then she does not know what hat she has (modeled by the fact that every other world *does* have a $\widehat{A}$ edge). So she passes.

The fact that she passes, however, tells Bob and Cam that they aren't living in RBB, because if they were, then Alyss would not have passed! So we can update our model of what the possible worlds are to exclude RBB:

Next is Bob's turn. If *he* sees Alyss wearing red and Cam wearing Blue, then he knows he is in the RRB world. Again, this is because the $\widehat{B}$ edges show Bob's *uncertainty*, and a lack of $\widehat{B}$ edges at the RRB world means Bob can be sure his hat color is red. Similarly, if Bob



sees two blue hats, then he knows his hat must be red, because the BRB world has no $\widehat{B}$ edge.

Then, if Bob passes, the other players know that they are not in RRB or BRB, since otherwise Bob would have won. They update their model of the possible worlds to match:

Finally, it is Cam's turn. But there are no $\widehat{C}$ edges left! So no matter what the state of the world is, Cam knows what their hat color must be! The game stops.

Notice, though, that Alyss and Bob are still uncertain about what world they are in. Even though Cam knows, Alyss and Bob are unable to determine their hat color. It was also extremely important that a moderator announce BBB was not a possible world, as this small bit of knowledge is what added the imbalance that made it possible to gain any knowledge at all.

Ex. 1.1 —

Play the hat game with Alyss, Bob, and Cam using

a. RRR as the real world

b. RBR as the real world

c. RBB as the real world

1.2 Applications

This is all well and good in the abstract, and I love puzzles as much as anybody else does. However, one would like to know that there are applications to more substantial problems. Which begs a question:

Why should we care?

Even though we outlined a great deal of examples at the start of this chapter, I suppose we can take a moment to give an explicit example.

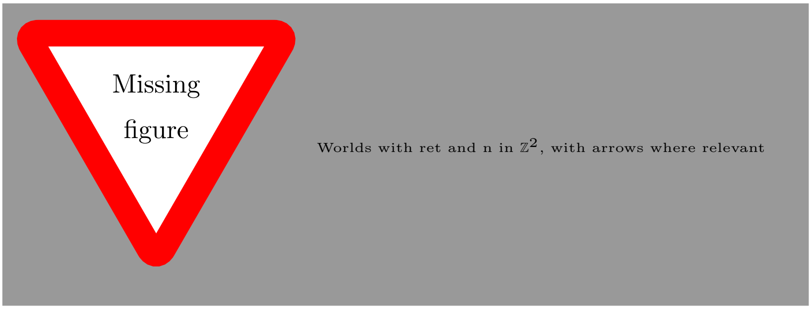
It is often the case in Computer Science that we have code which may or may not be correct. Testing code is important, but the only way to be truly sure that your code is correct is to *prove* that it is correct. But proving things is time consuming, so it would be nice if we could automate the process of proving certain programs correct. To that end, consider the following program:

```
def fact(n):
  ret = 1
  while n > 0:
    ret = ret * n
  return ret
```

What does it mean for this program to be correct? We need to have a specification of what the program *should* do to be able to prove that the program actually does it. Let's introduce that specification:

```
def fact(n):
#@Requires: n >= 0
#@Ensures: fact(n) = n!
  ret = 1
  while n > 0:
    ret = ret * n
    n = n-1
  return ret
```

Now a human can prove (by induction on n) that when $n \geq 0$, `fact`$(n) = n!$, and so the program `fact` matches its specification. That is, it is correct. But what does this have to do with modal logic?



As before, we introduce a set of possible worlds, this time corresponding to the various states the program execution might be in. We add edges from one world to another according to how what line of the program we are on. If a world reads a **`return ret`** instruction, then we show no edges, but we circle the world to show the computation has ended. [1]

The specification for this program says that if we start at a world where $n \geq 0$, then we will eventually find a world where we return, and further **`ret`** will be $n!$ in that world. We will see (in Chapter 6) how this can be checked by a computer at compile time. This means the programmer does not have to prove their code correct to know that it is – we offload this (routine) work to the compiler. Obviously this will not work for every program, so there is some care needed. The fact that we can do it at all, though, is both cool and useful.

The way we will prove this, though, is by creating a modal logic which models program execution. Then, using general results developed in the rest of the text, we will be able to get the above result for free!

[1] This construction is somewhat ad-hoc so that we don't have to get into the details of Propositional Dynamic Logic, which will be the subject of Chapter 6.

Chapter 2

Syntax and Semantics

2.1 Introduction

Central to the study of logic is the differentiation between *formulas*, which are strings of symbols, and *truth*, which is a judgement we can make about formulas. As a simple example, consider the sentence

$$\forall x. \exists y. x + y = 0.$$

If we take our domain to be $\mathbb{N}$, then the sentence is false. If we instead work over $\mathbb{Z}$, the same sentence is true.

In logic, we take this observation to an extreme, and we define complementary notions of **Syntax** and **Semantics**. The syntax defines the symbols which we are allowed to use, and which strings of symbols are meaningful. The string $+\forall 0$, for instance, uses the same symbols as the sentence above, but is not meaningful. A meaningful string of symbols is called a **Formula**. We have rules of syntax which make precise which strings of symbols are formulas. Dually, the semantics describe which formulas are *true*. As the above example shows, different formulas are true in different settings. Because of this, we must keep track of a **Model**, which gives us a way to interpret our formulas. Keep in mind that just as in the above example, the truth or falsity of a formula depends on the model in which we are interpreting it!

Of course we often think of things that are true as the things which have been *proven*. We formalize the notion of a proof with a **Deduction System**. A deduction system is a set of *axioms* and *inference rules*. The axioms are sentences which we say are provable by default, and the inference rules give ways of taking previously proven formulas and using them to prove new formulas. Much of the book will be dedicated to showing that, for a particular deduction system, a formula is *provable* if and only if it is *true in every model*.

In this chapter, we will introduce the syntax and semantics of a basic modal logic with one modality: $\Box$ (pronounced "box"). We also introduce a deduction system, called **System K**, whose provable statements (called **Theorems**) are exactly the formulas which are true in every model. In the exercises, you will play with a simpler (non-modal) logic (Classical Propositional Logic, or CPL) and a more complicated modal logic, **Multi-Agent Epistemic Logic**, which is what we used in the introduction to make sense of the red and blue hat game.

2.2 Syntax: The Basic Modal Language

The **Basic Modal Language** is defined recursively by the following **Grammar** :

$$
\begin{aligned}
\varphi, \psi = p & \qquad \text{(primitive propositions)} \\
\mid \varphi \wedge \psi & \qquad \text{(conjunction)} \\
\mid \neg \varphi & \qquad \text{(negation)} \\
\mid \Box \varphi & \qquad \text{(modality)}
\end{aligned}
$$

This means that p is always a formula. It is the base case of our definition. Inductively, if φ and ψ are formulas which have been defined, we can *also* define $\varphi \wedge \psi$, $\neg\varphi$, and $\Box\varphi$. Here p is called a **Primitive Proposition**. These are the basic things we wish to examine. We consider a set $\texttt{PROP}$ from which these propositions come. We might consider $\texttt{PROP} = \{$"it is raining", "it is sunny", "it is cloudy"$\}$. In this case, we can form sentences such as:

- $\neg$"it is raining" (it is *not* raining)
- "it is cloudy" $\wedge$ "it is raining" (it is cloudy *and* it is raining)
- $\Box$"it is sunny"

$\Box$ can be interpreted in a number of ways based on what one wishes to study. Some common interpretations of this last example (and their associated fields) are:

- "Valerie knows that it is sunny" (epistemic logic)
- "It will always be that it is sunny" (temporal logic)
- "It is necessary that it is sunny" (alethic logic)

We also have common abbreviations:

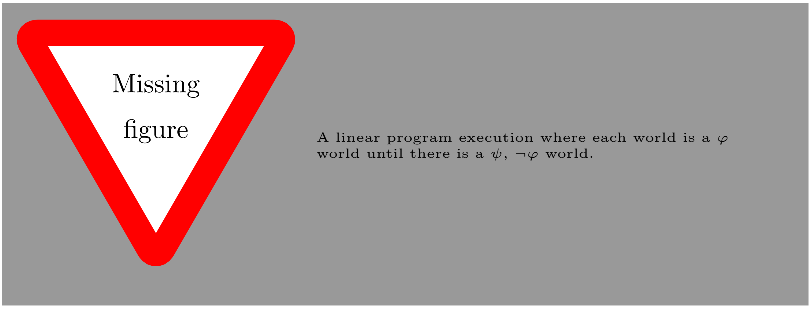
$\varphi \vee \psi$	$\neg(\neg\varphi \wedge \neg\psi)$	(disjunction)
$\varphi \to \psi$	$\neg(\varphi \wedge \neg\psi)$	(implication)
$\varphi \leftrightarrow \psi$	$(\varphi \to \psi) \wedge (\psi \to \varphi)$	(bi-implication)
$\Diamond\varphi$	$\neg\Box\neg\varphi$	(interpretation varies)

These abbreviations have the below interpretation. A reader unfamiliar with these abbreviations should convince themself that the abbreviations actually correspond to the interpretation we give here.

- "it is sunny" $\vee$ "it is cloudy" (it is raining *or* it is cloudy)
- "it is raining" $\to$ "it is cloudy" (*if* it is raining *then* it is cloudy)
- "it is raining" $\leftrightarrow$ "it is cloudy" (it is raining *if and only if* it is cloudy)
- $\Diamond$"it is sunny" (interpretation varies)

The interpretation of $\Diamond$"it is sunny" depends on the chosen interpretation of $\Box$. For example, it could mean any of the following:

- "Valerie considers it possible that it is sunny" (epistemic logic)
- "It will at some point be that it is sunny" (temporal logic)
- "It is possible that it is sunny" (alethic logic)



2.2.1 Extensions of the Basic Modal Language

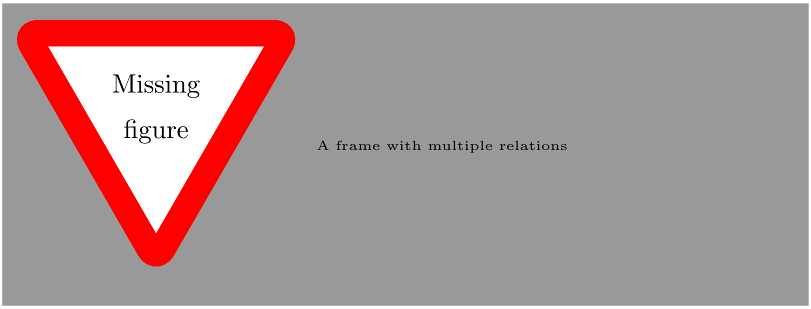
There are other, more expressive modal logics, too! Consider the following grammar, which defines the language of **Linear Temporal Logic** (LTL):

$$
\begin{array}{lll}
\varphi, \psi = p & & \text{(primitive propositions)} \\
& \mid \varphi \wedge \psi & \text{(and)} \\
& \mid \neg\varphi & \text{(not)} \\
& \mid \mathsf{N}\varphi & \text{(Next)} \\
& \mid \varphi \mathsf{U} \psi & \text{(Until)}
\end{array}
$$

This logic allows us to discuss properties which change over time, and to model when certain properties might be true in the future. This is useful in modeling program excution, where we want to know that certain "failure" states are never reached.

We have the same basic setup, but we have two modalities now: N and U. $\mathsf{N}\varphi$ holds if, in the next program state, φ holds. We also have a *binary* modality, where $\varphi\mathsf{U}\psi$ is true if and only if "φ holds *until* ψ holds". That is, in every future program state, φ is true. *However*, once ψ becomes true, all bets are off, and φ is allowed to be false again. You should be thinking of the picture in figure 2.2.1

When reasoning about knowledge, it is often the case that you have multiple people whose knowledge is important. Say you are modelling a poker game, and you want to be able to express that one person knows that another person knows that a third person is



bluffing! That's a lot of expressive power for a language to have, but it is still well within our grasp.

Given a set A of agents, we define a grammar for **Multi-Agent Epistemic Logic** as follows:

$$
\begin{aligned}
\varphi, \psi = \; & p && \text{(primitive propositions)} \\
& \mid \varphi \wedge \psi && \text{(and)} \\
& \mid \neg \varphi && \text{(not)} \\
& \mid K_a \varphi && \text{(Agent } a \text{ knows)}
\end{aligned}
$$

Here we have added a K_a for every $a \in A$, so we can express formulas like those from the introduction. In this setting we typically denote $\neg K_a \neg \varphi$ by $\widehat{K_a}\varphi$, read as "Agent a considers φ possible".

We also have to level up our models to make sense of these new formulas. Now instead of one relation R, our frames have a relation R_a for every $a \in A$. We then define our semantics exactly like before, but one relation at a time (see figure 2.2.1).

Ex. 2.1 —

Consider the following grammar
(where `n` stands for any of 0, 1, 2, 3...):

$p, q, r =$	`n`	(natural numbers)
	$\mid$ `t`	(true)
	$\mid$ `f`	(false)
	$\mid p{+}q$	(addition)
	$\mid$ `if` p `then` q `else` r	(conditional)

Which of the following are syntactically correct?

a. $5 + 3$

b. $6 + 2 + 1$

c. $5 +$ `t`

d. $3 * 4$

e. `if` 5 `then` `f`

f. `if` $5 + 3$ `then` `f` `else` $2 + 1$

g. $3 +$ `if` `t` `then` 2 `else` $4 + 1$

Ex. 2.2 —

a. Write down a grammar for a language which can express

- primitive propositions
- conjunction
- negation
- Valerie Knows
- Valerie Believes

b. Translate some sentences about belief and knowledge from english

Ex. 2.3 —
Translate the following sentences between Epistemic Logic and English, assuming any claims about which activities are fun are primitive.

a. Valerie knows modal logic is fun.
b. Valerie knows that if modal logic is fun, then so is mathematics.
c. If computer science is fun, then Valerie thinks it might be.
d. Valerie doesn't know that modal logic is fun.
e. Valerie knows that doing homework isn't fun.

Ex. 2.4 —

a. Explicitly write down a grammar for **Multi-Agent Epistemic Logic** with 3 agents. This language should be able to express
   - primitive propositions
   - conjunction
   - negation
   - Agent 1 knows
   - Agent 2 knows
   - Agent 3 knows
b. Translate some sentences from MAEL to english
c. Translate some sentences from english to MAEL

Ex. 2.5 —
Translate the following sentences between LTL and English, assuming any claims about the weather are primitive. Let's say that each time-step is one day, so that $\mathsf{N}\varphi$ means "Tomorrow, φ holds"

a. It will be sunny until it is raining.
b. If it is windy, then tomorrow it will be cloudy.
c. If it snows tomorrow, then it will snow everyday until it is sunny.
d. As long as it is cold and windy, it will be rainy.
e. "It is sunny" U "It is cloudy"
f. N "It is windy"
g. N ("It is hot" U "It is raining")

Ex. 2.6 —
Create a grammar which allows us to express statements about what will happen tomorrow and what Valerie knows. Use it to formalize the following sentence:

> If Valerie knows it will rain tomorrow, she will bring an umbrella.

(Assume that "Valerie carries an umbrella", as well as "It is raining" are primitive.)

2.3 Semantics: Kripke Frames and Models

Now we have a language which we can use to write down formulas. However, just as with $\forall x.\exists y.x+y=0$, the truth of these formulas will depend on how we *interpret* each of the symbols. The interpetation of the language we choose is called the **Semantics** of our language.

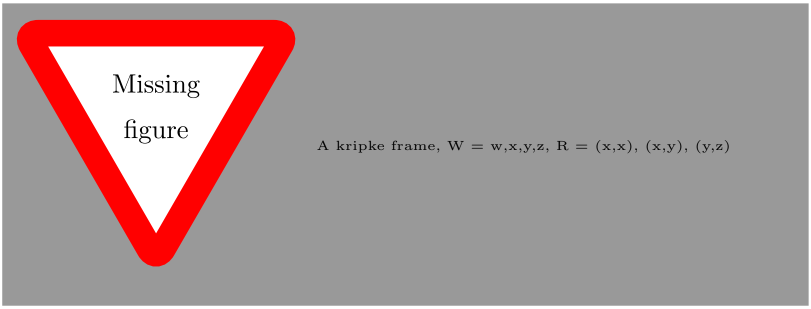
There are many ways to interpret the basic modal language, though we will spend most of this book discussing **Kripke Semantics**, which is the classically used and studied semantics. Kripke Semantics are phrased in terms of **Relational Structures**, and so we will take a quick detour into the world of relations. Here we will encounter some fundamental examples and definitions which will serve us well for the rest of the book.

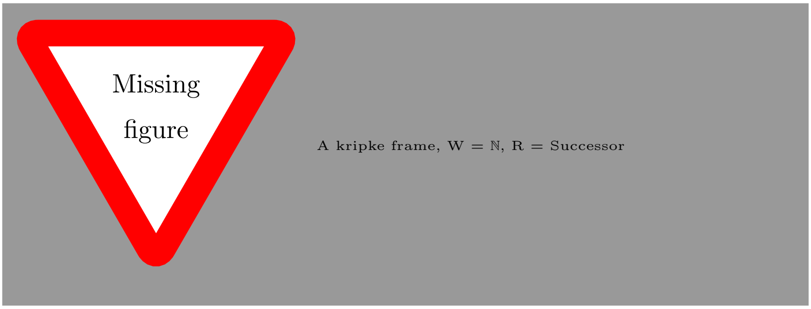
In chapter 5 we will describe another family of models in which we can interpret modal formulas, which will give rise to a **Topological Semantics** for modal logic. For now, however, let us continue with the definition of a Relational Structure:

2.3.1 Relational Structures

Definition 2.1. A **Relational Structure**, or **Frame**, $\mathfrak{F} = (W, R)$ is a set W, and a *relation* $R \subseteq W \times W$

We frequently refer to $x \in W$ as a *world* or *state*. We will also abuse notation, and write $x \in \mathfrak{F}$ when we mean $x \in W$ for $\mathfrak{F} = (W, R)$. We write $R(x, y)$, or xRy if $(x, y) \in R$, which is to say that $x \in W$ is *related* to $y \in W$ by R. We will often say that "x sees y" if xRy, using the intuition that every world can only see in some directions, and can





only see one "step" away from itself. Note that if x sees y, that does NOT mean y sees x. Finally, we write $R(x) = \{y \in W \mid R(x, y)\}$

One can visualize this relation as a directed graph, whose vertices are given by W, and where there is an edge from x to y if and only if xRy. There are examples throughout this section.

2.3.2 Frame Properties

There are a number of properties a relation $R \subseteq W \times W$ can have. Indeed, we will show that many of these properties are expressible by formulas in the basic modal language. The interplay between properties of relations and modal formulas will discussed in depth in section 3.6, however we must first familiarize ourselves with these properties. We appologize for the rapid fire barrage of definitions the reader is about to encounter, but we want to make sure that we formally introduce everything we might need.

A kripke frame, W = $\mathbb{Z}$, R = $\leq$ (show the transitive arrows too)

The = relation on 7 worlds

The mod 3 relation on $\mathbb{Z}$ - say from -4 to 4?

Definition 2.2. $R \subseteq W \times W$ is called **Reflexive** if $\forall x \in W . xRx$.

In a reflexive relation, every world sees itself. For example, $=$ is reflexive, as $x = x$. A less trivial example might be congruence "mod 3". Here we say that $x \sim_3 y$ if and only if $3 \mid x - y$. Since $3 \mid 0$, we see $x \sim_3 x$. We might also take the "same family" relation on people. Since any one person is in the same family as themself, this relation is reflexive. Notice $x \leq y$ *is* reflexive, while $x < y$ isn't.

Definition 2.3. $R \subseteq W \times W$ is called **Symmetric** if $\forall x, y \in W . xRy \iff yRx$.

In a symmetric relation, x sees y exactly when y sees x. Graph theoretically, a symmetric relation corresponds to an undirected graph, compared to the directed graphs we normally work with. $=$, again, is symmetric (indeed if $x = y$, then $y = x$ too). A slightly more whimsical relation is the "handshake" relation. If we have a room of business people P, then we can put $p_1 R p_2$ whenever p_1 and p_2 have shaken hands. Clearly this relation is symmetric, since if p_1 has shaken p_2's hand, then p_2 must have also shaken p_1's hand. As a non-example, $\leq$ is *not* symmetric, since $3 \leq 5$, but $5 \not\leq 3$.

Definition 2.4. $R \subseteq W \times W$ is called **Transitive** if $\forall x, y, z \in W . xRy$ and yRz implies xRz.

The classic example of a transitive relation is $\leq$. If $x \leq y$ and $y \leq z$, then $x \leq z$. Another way of phrasing this is that if x sees y, then x sees everything y does. Note $=$ is also transitive, since $x = y$ and $y = z$ implies $x = z$. As another example, consider the "branching time" model shown in 2.3.1.

Definition 2.5. $R \subseteq W \times W$ is called an **Equivalence Relation** if it is reflexive, symmetric, and transitive. Equivalence relations are often written as $\sim$.

An equivalence relation is a relation which "looks like equality". That is, we can group W into clumps based on xRy, and we will have xRy if and only if they fall into the same clump. This will be extremely useful in the future when we talk about *Filtration* in section 4.4. Consider P, the set of all people, with the relation $p_1 \sim p_2$ iff p_1 and p_2 have the same birthday. As a slightly mathier example, consider the "congruent mod 3" example from earlier.

In an antisymmetric relation, the only way for two worlds to see each other is for them to actually be the same. Consider $\leq$ again. If $x \leq y$ and $y \leq x$, then $x = y$.

Definition 2.6. A relation $R \subseteq W \times W$ is a **Partial Order** if it is reflexive, transitive, and *antisymmetric*. This last condition says that xRy and yRx can only be true if $x = y$. Partial Orders are often denoted $\preceq$.

Definition 2.7. If $\preceq \subseteq W \times W$ is a partial order with the following bonus property: $\forall x, y.(x \preceq y \text{ or } y \preceq x)$

Then $\preceq$ is called a **Total Order**

Orders, like it says on the tin, let us put events in an order. If w_1 comes before w_2, we write $w_1 \preceq w_2$. It makes sense for these to be transitive, beacuse if Mozart came before The Beatles, and The Beatles came before Louis Cole, it stands to reason Mozart came before Louis Cole. Reflexivity is a bit more subtle, since it doesn't necessarily make sense to say Debussy came before Debussy. However this is the view that mathematicians have adopted. It doesn't make much difference, though, since we can define $x \prec y$ as $x \preceq y \wedge x \neq y$ to avoid this situation if we want to.

A total order lets us put events in an order in such a way that for any two events, one definitely came first. Think about timelines, for example. For any two composers, we can always say which of them started composing first. There is no ambiguity.

With a partial order, however, we can't *always* say which event came first. Consider a branching timeline, where every decision splits the timeline in two. In a timeline exactly like ours, but where Beethoven never became a composer, music would have developed completely differently. Does it make sense to say that a band in their timeline, which doesn't exist in ours, "came before" a band which exists in ours but not theirs? Partial Orders reflect this ambiguity, by allowing $x \preceq y$ to be undefined sometimes.

You are already familiar with a number of total orders. $(\mathbb{Z}, \leq)$ and $(\mathbb{R}, \leq)$ are both total orders. One partial order is $\subseteq$. Consider 2^X, the set of subsets of X. Then $\subseteq$ defines a partial order on 2^X. Note for $A, B, C \subseteq X$: $A \subseteq A$ (so $\subseteq$ is reflexive), $A \subseteq B$ and $B \subseteq A$ implies $A = B$ (so $\subseteq$ is antisymmetric), and if $A \subseteq B$ and $B \subseteq C$ then $A \subseteq C$ (so $\subseteq$ is transitive).

Definition 2.8. A relation $R \subseteq W \times W$ is called **Serial** if $R(x)$ is nonempty for every x.

Serial relations say that you can never "get stuck" as you move around the worlds of a model. No matter where you are, there is always an edge to follow. Almost all of the examples we have seen so far are serial. As a non-example, consider figure 2.3.1. w and z are not related to any other world, and so the frame is not serial. Notice y *is* related to z, however.

Finally, we give the definitions for *classes* of frames. That is, a collection[1] of frames with some property. Much of this book will deal with the connection between certain classes of frames and certain kinds of modal logics, but we cannot discuss the connections before we have some definitions in place.

Definition 2.9. $\mathcal{C}_{\text{all}}$ is the class of all frames.

$\mathcal{C}_{\text{refl}}$ is the class of reflexive frames.

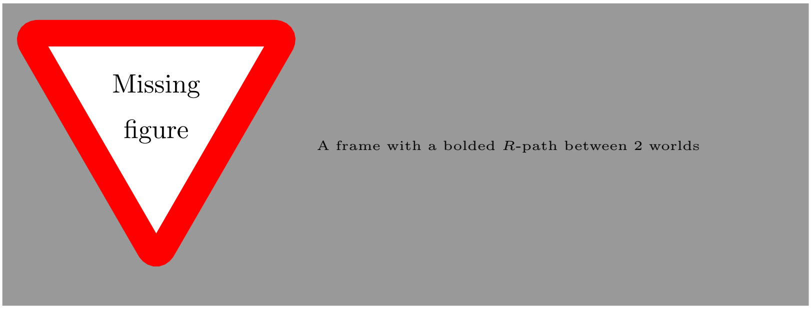
$\mathcal{C}_{\text{trans}}$ is the class of transitive frames.

$\mathcal{C}_{\text{refl,trans}}$ is the class of all reflexive, transitive frames.

$\mathcal{C}_{\text{pos}}$ is the class of all posets.

$\mathcal{C}_{\text{serial}}$ is the class of serial frames.

[1] which might be bigger than a set, for those with some set theoretic background



2.3.3 Operations on Relational Structures

Oftentimes we will want to take a given relation and modify it slightly to give it extra properties. The most useful are defined here:

Definition 2.10. Given a relation $R \subseteq W \times W$, its **Reflexive Closure** is the relation $R' = R \cup \{(w, w) \mid w \in W\}$.

This one is pretty obvious. We have a relation, but we want it to be reflexive. Let's just add in all the stuff it takes to be reflexive. As an example, the reflexive closure of $<$ is $\leq$.

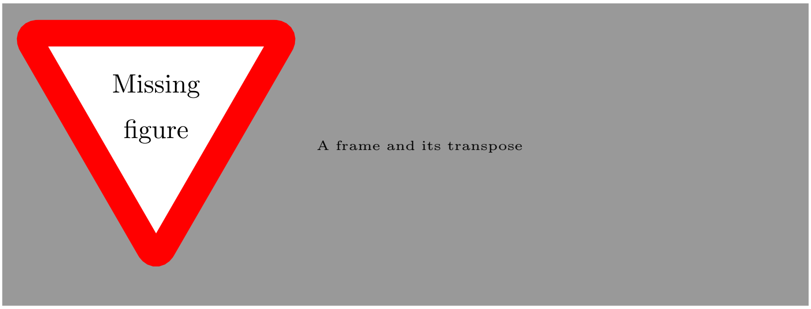
Definition 2.11. As a kind of auxiliary definition, an **R-path from x to y** is a chain $xRz_0Rz_1Rz_2R \ldots Rz_nRy$.

In figure 2.3.3, the bolded path is an R-path, since we see $xRz_0Rz_1Rz_2Ry$.

Definition 2.12. Given a relation $R \subseteq W \times W$, its **Transitive Closure** R^+ is given by xR^+y if and only if there is an R-path from x to y. A slightly more formal definition might be

$$xR^+y \iff \begin{cases} xRy \\ xRz \text{ and } zR^+y \end{cases}$$

This one is well motivated – it makes sense that we might *want* to make a relation transitive. However it might be less clear that this new relation is actually transitive. Luckily, this is the content of exercise 2.13. As an example, the transitive closure of $+1$ from 2.3.1 is $<$. This is because $x < y \iff xSz_1Sz_2 \ldots z_nSy$, where xSy means $x + 1 = y$.



Oftentimes we will want to do both of these. In fact, we want to do it so often it gets its own name:

Definition 2.13. Given a relation $R \subseteq W \times W$, its **Reflexive-Transitive Closure** R^* is given by first taking its transitive closure, then taking its reflexive closure.

The reflexive transitive closure of $+1$ is $\leq$.

Definition 2.14. Given a relation $R \subseteq W \times W$ and a subset $X \subseteq W$, define the **Restriction of** R **to** X by $R \upharpoonright X = R \cap X \times X$.

Intuitively, restricting a relation corresponds to only looking at certain worlds of W. If we ignore all the worlds which are not in X, we cannot talk about edges between these worlds. However, we want to keep as much information as possible, so any edge we *can* keep, we want to. This will be important when we talk about *Generated Submodels* in section 4.3.

Definition 2.15. Given a relation $R \subseteq W \times W$, its **Transpose** [2] R^T is defined as xR^Ty if and only if yR^Tx.

The transpose of a relation lets us move *backwards* along edges.

[2] R^T is sometimes called R^{op}

Ex. 2.7 —
Say whether the following relations are Reflexive, Transitive, Symmetric, or Serial. Also specify if they are a Partial or Total Order, or an Equivalence Relation.

a.
b.
c.
d.
e.

Ex. 2.8 —
Compute the reflexive, transitive, and reflexive-transitive closures of the following relations.

a.
b.
c.

Ex. 2.9 —
A **partition** of a set X is a family of nonempty subsets $A_i \subseteq X$ such that $A_i \cap A_j = \emptyset$ and $\bigcup A_i = X$. Thus every $x \in X$ is in exactly one A_i.

a. Show every partition induces an equivalence relation $\sim$ where $x \sim y$ if and only if x and y are in the same A_i
b. Show every equivalence relation $\sim$ induces a partition of X.

Ex. 2.10 —
$(n_1, d_1) \sim (n_2, d_2) \subseteq \mathbb{Z} \times \mathbb{N} \iff n_1(d_2 + 1) = n_2(d_1 + 1)$ is eq rel. Relate to $\mathbb{Q}$.

Ex. 2.11 —
The only relation which is symmetric and antisymmetric is $=$

Ex. 2.12 —

Definition 2.16. A relation $R \subseteq W \times W$ is called **Euclidean** if whenever xRy and xRz, we also have yRz.

Show a reflexive, euclidean relation is an equivalence relation.

Ex. 2.13 —
Prove the transitive closure of a relation is actually transitive.

Ex. 2.14 —
We can represent a relation as a matrix with 0/1 valued entries. Given a relation $R \subseteq W \times W$, define $\mathbf{A}_R[x, y] = 1 \iff xRy$.

a. What can we say about $\mathbf{A}_R$ if R is symmetric? What about Reflexive? Transitive?
b. Prove $\mathbf{A}_{R^T} = {\mathbf{A}_R}^T$. That is, the matrix of the transpose of R is the transpose of the matrix of R.
c. What must $\mathbf{A}_R$ look like if R is a total order?

Ex. 2.15 —
We seemingly made an arbitrary choice with the reflexive-transitive closure: We decided to first take its reflexive closure, then its transitive closure. If we define the transitive-reflexive closure to be "first take the reflexive closure, then take the transitive closure", would we get a different relation? Find a relation R in which they differ, or prove they are always the same.

2.3.4 Models

We've spent a lot of time on frames and properties of relations, but it is still unclear what these might have to do with interpreting modal formulas. How can a frame say if $\Box p \to q$ is true or false? The answer is: it can't. We need a little bit more information, which this section provides.

Definition 2.17. A **Kripke Model** $\mathfrak{M} = (W, R, v) = (\mathfrak{F}, v)$ is a frame $\mathfrak{F}$ equipped with a **valuation function** $v : \mathtt{PROP} \to 2^W$, which sends a primitive proposition to the set of worlds where it is true (as usual, $2^W = \{A \mid A \subseteq W\}$ is the *powerset* of W).

As with frames, we will abuse notation and say that $w \in \mathfrak{M}$ if actually $w \in W$ for $\mathfrak{M} = (W, R, v)$. If $\mathfrak{M} = (\mathfrak{F}, v)$, we call $\mathfrak{F}$ the **Underlying Frame** of $\mathfrak{M}$.

For the following examples, let $\mathtt{PROP} = \{p, q\}$ Each world is tagged with p or q, based on the valuation. If $w \in v(p)$, then we write a p

$\mathfrak{M}_1$, W = w,x,y,z, R = (x,x), (x,y), (y,z), p worlds are x,y, q worlds are w,x,z

$\mathfrak{M}_2$, W = a,b,c, R = (a,a), (b,b), (c,c), (a,b), (b,c), (a,c), p worlds are a b, q worlds are b c next to w, likewise for q. This gives us a way of saying "w thinks that p is true." The next obvious question is to ask "How can we know if w thinks that φ is true?"

2.3.5 Kripke Semantics

In modal logic, we interpret formulas *at a particular world* $w \in \mathfrak{M}$. Thus, each world is a sort of microcosm of CPL (the logic of just $\wedge$ and $\neg$), and $\Box$ will allow us to see what nearby worlds think.

To that end, for $w \in \mathfrak{M} = (W, R, v)$, we recursively define $w \vDash \varphi$ (read as "*w satisfies φ*", or "*w thinks that φ is true*") as follows:

$$\begin{aligned}
(\mathfrak{M}, w) \vDash p &\iff w \in v(p) \\
(\mathfrak{M}, w) \vDash \neg\varphi &\iff (\mathfrak{M}, w) \nvDash \varphi \\
(\mathfrak{M}, w) \vDash \varphi \wedge \psi &\iff (\mathfrak{M}, w) \vDash \varphi \text{ and } (\mathfrak{M}, w) \vDash \psi \\
(\mathfrak{M}, w) \vDash \Box\varphi &\iff \forall x \in R(w).(\mathfrak{M}, x) \vDash \varphi
\end{aligned}$$

We will abuse notation and write $w \vDash \varphi$ if the model $\mathfrak{M}$ is clear from context. We will also occasionaly write $\mathfrak{M}, w \models \varphi$, dropping the parentheses. Additionally, given a set of formulas A, we will write $\mathfrak{M}, w \vDash A$ if $\mathfrak{M}, w \vDash \varphi$ for each $\varphi \in A$.

The valuation function says which basic propositions are true at each world, and we can recursively say which larger formulas are true at a given world. Notice that if a formula does not use $\Box$, then it has the same truth value it would in propositional logic. The power of modal logic, and indeed of $\Box$, comes from the ability to quantify over nearby worlds. Before we go too much further, though, there are explicit semantics for some of our common abbreviations:

Theorem 2.18. $\mathfrak{M}, w \vDash \varphi \vee \psi \iff \mathfrak{M}, w \vDash \varphi$ or $\mathfrak{M}, w \vDash \psi$

Proof.

$$\begin{aligned}
\mathfrak{M}, w \vDash \varphi \vee \psi &\iff \mathfrak{M}, w \vDash \neg(\neg\varphi \wedge \neg\psi) \\
&\iff \mathfrak{M}, w \nvDash \neg\varphi \wedge \neg\psi) \\
&\iff \mathfrak{M}, w \nvDash \neg\varphi \text{ or } \mathfrak{M}, w \nvDash \neg\psi \\
&\iff \mathfrak{M}, w \vDash \varphi \text{ or } \mathfrak{M}, w \vDash \neg\psi
\end{aligned}$$

■

Theorem 2.19. $\mathfrak{M}, w \vDash \Diamond\varphi \iff \exists x \in R(w).\mathfrak{M}, x \models \varphi$

Proof.

$$\begin{aligned}
\mathfrak{M}, w \vDash \Diamond\varphi &\iff \mathfrak{M}, w \vDash \neg\Box\neg\varphi \\
&\iff \mathfrak{M}, w \nvDash \Box\neg\varphi \\
&\iff \text{not every } x \in R(w) \text{ has } \mathfrak{M}, x \vDash \neg\varphi \\
&\iff \exists x \in R(w).\mathfrak{M}, x \nvDash \neg\varphi \\
&\iff \exists x \in R(w).\mathfrak{M}, x \vDash \varphi
\end{aligned}$$

■

There are a number of derived concepts relating to the truth of certain formulas at certain worlds, which we will now define.

Definition 2.20. The **Theory** of a world $w \in \mathfrak{M}$ is

$$\mathrm{Th}(w) = \{\varphi \mid w \vDash \varphi\}$$

Definition 2.21. The **Denotation** of a formula φ, written $v(\varphi)$ or $[\![\varphi]\!]$ is the extension of the valuation v to all formulas. That is,

$$v(\varphi) = [\![\varphi]\!] = \{x \in \mathfrak{M} \mid x \vDash \varphi\}$$

Definition 2.22. We say that a model $\mathfrak{M}$ **Validates** a formula φ (written $\mathfrak{M} \vDash \varphi$) if $\mathfrak{M}, x \vDash \varphi$ for every $x \in \mathfrak{M}$.

Similarly, we say a frame $\mathfrak{F}$ validates φ if $(\mathfrak{F}, v) \vDash \varphi$ for every valuation function v. That is, every model we build on $\mathfrak{F}$ validates φ. Totally expanded, this says that in every world x of every model $\mathfrak{M}$ we can build on $\mathfrak{F}$, $\mathfrak{M}, x \vDash \varphi$. A strong property indeed!

We can go one stronger, though. Given a collection $\mathcal{C}$ of frames[3], we say $\mathcal{C} \vDash \varphi$ if every $\mathfrak{F} \in \mathcal{C}$ satisfies φ.

Finally, given a set A of formulas, we allow ourselves to write $\mathfrak{M} \vDash A$ to mean $\mathfrak{M} \vDash \varphi$ for every $\varphi \in A$. The definitions for $\mathfrak{F} \vDash A$ and $\mathcal{C} \vDash A$ are similar.

[3] which might be a proper class

Thus, in the models shown above:

- $\mathfrak{M}_1, w \models q \wedge \neg p$
- $\mathfrak{M}_1, x \models p \to q$
- $\mathfrak{M}_1, x \models p \vee q$
- $\mathfrak{M}_1, y \models \Box q$
- $\mathfrak{M}_1, x \models \Box p$
- $\mathfrak{M}_1, w \models \Box p$
- $\mathfrak{M}_1, w \models \Box(p \wedge \neg p)$

Notice $w \models \Box(p \wedge \neg p)$, even though this is a contradiction! This is because w does not see any worlds, and so, vacuously, every world it sees is contradictory. You will show in exercise 2.19 that this is the only way this may happen.

2.3.6 An Example Proof

At this point, we know enough to prove that every frame satisfies certain properties. For example, let's show

$$\models \Diamond(p \vee q) \leftrightarrow \Diamond p \vee \Diamond q$$

Proof. Let $\mathfrak{M} = (W, R, v)$ be a model. Let $w \in W$ be a world.

To show $\to$, we may assume $w \models \Diamond(p \vee q)$, as otherwise the claim is true vacuously. But if $w \models \Diamond p \vee q$, then for some $w' \in R(w)$ we have $w' \models p \vee q$. But then $w' \models p$ or $w' \models q$. Say $w' \models p$. Then w sees a world modeling p, and so $w \models \Diamond p$. If instead $w' \models q$, then $w \models \Diamond q$. Either way, we have $w \models \Diamond p \vee \Diamond q$.

To show $\leftarrow$, say $w \models \Diamond p \vee \Diamond q$, as otherwise the claim is vacuously true. Then for some $w' \in R(w)$, $w' \models p$ or $w' \models q$. But then $w' \models p \vee q$ and so $w \models \Diamond(p \vee q)$. ■

2.3.7 Interpretations

Now, let's get back to why this might be useful, and discuss how these frames actually model various situations of interest.

In an Epistemic Interpretation of the basic modal language, we say that xRy if Valerie cannot distinguish between x and y. If she is in x, she cannot tell if she is in x or y. Notice this implicitly places some assumptions on our frame! Surely x should be indistinguishable from itself, after all. It also stands to reason that if Valerie cannot tell x from y, she shouldn't be able to tell y from x either! So we are dealing with the class of Reflexive, Symmetric Frames.

Now if x is the "real world" and it is sunny in x, it is still possible that Valerie doesn't *know* that it is sunny. Perhaps she is inside, and she has no way of knowing if it is sunny or cloudy outside. However, if Valerie then looks out a window and sees the brilliant afternoon sun, she knows that it is sunny. Why? Because now every world which she considers possible has to take into account the sun she's looking at. Said more formally, after Valerie opens the window, if $y \in R(x)$, then $y \vDash \text{Sunny}$. This is another way of saying $x \vDash \Box\text{Sunny}$, and we can see that $\Box$ lines up with our intuition about knowledge (at least in this case).

We might also consider a Temporal Interpretation of the basic modal language. Here we consider possible words as being various moments in time, with xRy whenever x comes temporally before y. Again, this places some assumptions on our frames. Clearly we want to work in the class of Partial Orders, or, if we want time to be linear, Total Orders.

If x is the "current moment", then we can ask what $\Box\text{Sunny}$ means. It says that at all future times y, $y \vDash \text{Sunny}$. A quick way of saying this is "henceforth Sunny". From x onwards, it will be sunny.

Ex. 2.16 —
Identify $[\![p]\!]$, $[\![\Box q]\!]$, and $[\![\Diamond p]\!]$ in figures 2.3.4 and 2.3.4 above.

Ex. 2.17 —
Which of the following models validates $\Diamond(q \vee \neg p)$?

- The model in figure 2.3.4
- The model in figure 2.3.4
- $\Big(\{0,1\}, \{(0,1),(1,0)\}, v(p) = \{0\}, v(q) = \emptyset\Big)$
- $\Big(\mathbb{Z}, \{(x, x+1) \mid x \in \mathbb{Z}\}, v(p) = \text{odds}, v(q) = \text{multiples of } 3\Big)$

Ex. 2.18 —
Prove that any frame $(X, =)$ validates $\Box p \to p$. That is, we have $xRy \iff x = y$. This is often called the **Discrete Frame**.

Ex. 2.19 —
Show that $(W, R, v), w \models \Box(p \wedge \neg p)$ if and only if $R(w) = \emptyset$.

Ex. 2.20 —
Find Th(x) and Th(y) in figure 2.3.4, and Th(a), Th(b), and Th(c) in 2.3.4.

Ex. 2.21 —
Find Th(0) and Th(1) in $(\{0,1\}, \{(0,0),(0,1),(1,1)\}, v)$ with $v(p) = \{0\}$ and $v(q) = \{1\}$.

Ex. 2.22 —
Prove *every* model validates $\Diamond(p \vee q) \to \Diamond p \vee \Diamond q$

Ex. 2.23 —
Consider the following grammar:

$$
\begin{aligned}
\varphi, \psi = x_i & \quad \text{(primitive propositions)} \\
\mid \varphi \wedge \psi & \quad \text{(and)} \\
\mid \neg\varphi & \quad \text{(not)}
\end{aligned}
$$

This grammar defines the syntax for Classical Propositional Logic (CPL), which is the basis for the Basic Modal Language. A model of CPL is merely a function $v : \mathbb{N} \to \{T, F\}$, where we say $v \vDash x_i$ if and only if $v(i) = T$. We can extend the definition to all formulas in the same way we did for Kripke Semantics.

a. What should the semantics be for $\varphi \wedge \psi$ and $\neg\varpi$?

b. Show that every model satisfying $p \wedge (p \to q)$ also satisfies q.

Ex. 2.24 —
Show every CPL tautology is true at every world of every kripke model.

Ex. 2.25 —
Explain why $\Diamond\varphi$ corresponds (epistemically) to Valerie considering φ possible. How would we express that she knows φ isn't true, and how are these two ideas related?

Ex. 2.26 —
Explain why $\Diamond\varphi$ corresponds (temporally) to "φ will happen eventually". How would we express that φ will never happen, and how is these ideas related?

Ex. 2.27 —
Belief modality – Explain some assumptions for how it works, and ask for what class of frames we should use to interpret it.

Ex. 2.28 —
Something inovlving MAEL. Maybe interpreting formulas?

Ex. 2.29 —
Define a modality C for common knowledge, and define $\mathfrak{M} \vDash C\varphi$ if φ is true, $K_i\varphi$ is true for each i, $K_jK_i\varphi$ is true for every i and j, $K_kK_jK_i\varphi$ is true for each i, j, k, and so on. This encodes the notion that φ is **Common Knowledge**, since not only is φ true. Everyone *knows* φ is true. And everyone knows that everyone knows that φ is true. And so on.

a.

b.

c.

2.4 Provability: System K

Of course, the Semantics are only half of the story. Logic is, in large part, about proving things – so how do we prove things? And what does it mean for something we prove to be *true*? We literally just talked about truth being relative to a particular modal.

We will answer these questions by introducing a Deductive System which lets us (purely formally) push symbols around. If we set up the system correctly, any string of symbols which we can reach by playing by the rules of our system will be **Valid** on some class of models. That is, the formulas which are derivable will not be true *sometimes*, they will be true at every world of every model. What is somewhat more surprising is that every Valid formula will be derivable!

2.4.1 Deduction Systems

A **Deduction System** Λ, for our purposes, is a mathematical object which we can use to *derive* theorems. These theorems will be sentences in a language, for instance, the Basic Modal Language defined above. It consists of **Axioms**, which are formulas which we assume as proven, and **Inference Rules**, which are rules for turning existing proofs of formulas into new proofs of other formulas.

Ideally, we would know that all of the formulas we can prove are valid. A deduction system which only proves valid formulas is called **Sound**. Conversely, it would be nice if every valid formula was provable in our system. This property is called **Completeness**, and it turns out the deduction system we will describe enjoys both of these properties, as we will prove in future chapters.

Think of a deduction system as being a description of rules by which one can shuffle symbols around. The allowable starting positions are the *axioms*, and the *rules of inference* describe how one can move the symbols. Then a *proof* is simply a legal sequence of steps which arrives at the desired symbol order. It is a certificate that you didn't cheat over the course of the game.

We will write our Axioms as

$$\frac{}{\text{Axiom}}$$

and our Rules of Inference as

$$\frac{\text{Antecedent 1} \qquad \cdots \qquad \text{Antecedent n}}{\text{Conclusion}}$$

This notation says that whenever we have derived the sentences above the line, we are allowed to derive the sentence below the line. This is why axioms are listed with nothing above the line – they correspond to things which are derivable with no extra information.

Definition 2.23. Given a deduction system Λ, a Λ-**Derivation** is a list of formulas, each of which is either an axiom, or a rule of inference where each antecedent comes from earlier in the list.

We say $\Lambda \vdash \varphi$ (read "Λ *entails* φ" or "Λ *proves* φ") or φ is a **theorem** of Λ if some Λ-Derivation ends in φ.

We will give examples of derivations in the next section, once we have a concrete system to work with.

2.4.2 System K

System K will be the system on which we base all other modal logics, and will have many results proven about it. It has the following Axioms:

$$\frac{}{\varphi \text{ a CPL tautology}}\ \text{CPL}$$

$$\frac{}{\Box(\varphi \to \psi) \to \Box\varphi \to \Box\psi}\ \text{Distribution or K}$$

And the following Rules of Inference:

$$\frac{\varphi \to \psi \qquad \varphi}{\psi}\ \text{Modus Ponens}$$

$$\frac{\varphi}{\Box\varphi}\ \text{Necessitation}$$

Intuitively, what do these rules mean? Though we have not formally shown it, the theorems of this logic will be true at every world of every model. To that end, we would like our axiom schemes to be true at every world of every model. Further, we would like our rules of inference to preserve that truth.

The first axiom scheme tells us that everything which was true in propositional logic remains true here, if we are only looking at one world. This formalizes the idea from earlier that each individual world of $\mathfrak{M}$ should look "like a microcosm" of propositional logic. For example, for every φ and ψ, the following are axioms:

$$\varphi \wedge \psi \to \varphi$$

$$\varphi \vee \psi \to \psi \vee \varphi$$

$$\varphi \vee \neg\varphi$$

Importantly, these are true *for every formula* φ and ψ, even if, say, φ itself refers to $\Box$. So, for example, substituting $\varphi = \Box\xi \to \xi$ and $\phi = \sigma \wedge \Box\xi$, we see the following are axioms:

$$(\Box\xi \to \xi) \wedge (\sigma \wedge \Box\xi) \to (\Box\xi \to \xi)$$

$$(\Box\xi \to \xi) \vee (\sigma \wedge \Box\xi) \to (\sigma \wedge \Box\xi) \vee (\Box\xi \to \xi)$$
$$(\Box\xi \to \xi) \vee \neg(\Box\xi \to \xi)$$

One more example which will become important when we start doing proofs is $\varphi \wedge (\varphi \to \psi) \to \psi$. This is another theorem of propositional logic (as you showed in exercise 2.31), and so is an axiom of K.[4]

The distribution axiom scheme tells us something about how we want $\Box$ to behave. Namely, for any formulas φ and ψ,

$$\Box(\varphi \to \psi) \to \Box\varphi \to \Box\psi$$

is an axiom. This makes sense, since if at some world x of some model both $\Box(\varphi \to \psi)$ and $\Box\varphi$ are true, then for every world x sees, $\varphi \to \psi$ and φ are true. But by propostional logic, we know that ψ must be true at every world x sees. That is, $\Box\psi$ should be true.

Modus ponens tells us how to combine two existing theorems to create a new theorem. Namely, if we can derive the truth of both $\varphi \to \psi$ and φ, then we are allowed to derive the truth of ψ.

Finally, Necessisitation tells us how to derive $\Box\varphi$. If we have derived φ, then we are allowed to derive $\Box\varphi$. If we have proven φ, then it is valid (though we have not officially shown this yet). But if every world of every model satisfies φ, then in particular for any world x, every world it can see satisfies φ. That is, x satisfies $\Box\varphi$.

2.4.3 Some Example Derivations

Let's see a few derivations in K. The definition of derivations might be somewhat confusing, but hopefully seeing some examples will clarify what it. The concept itself is not scary at all.

Throughout this, p, q, r, etc. will be primitive propositions.

$$K \vdash p \wedge q \to p$$

1. $p \wedge q \to p$ CPL

[4]If taking all of propositional logic as axioms leaves a sour taste in your mouth, we could instead take any axiomatization of propositional logic, and the deduction system would remain unchanged.

Since $p \wedge q \to p$ is valid in CPL, it is an axiom of K. That's all there is to it. More generally, since we are allowed *any* sentences φ or ψ in the CPL axiom, the following is also a K-derivation for any sentences φ and ψ in the basic modal language:

1. $\varphi \wedge \psi \to \varphi$ CPL

As a specific example, the following is a K-derivation (taking $\varphi = \Box p \to q$ and $\psi = p \vee \Diamond q$):

1. $(\Box p \to q) \wedge (p \vee \Diamond q) \to (\Box p \to q)$ CPL

Ok, let's do a slightly more complicated derivation. From now on we will use φ and ψ freely, knowing that we can substitute any specific formula we want, as in the previous example. Let's show that

$$K \vdash \Box(\varphi \wedge \psi) \to \Box\varphi$$

1. $\varphi \wedge \psi \to \varphi$	CPL
2. $(\varphi \wedge \psi \to \varphi) \to \Box(\varphi \wedge \psi) \to \Box\varphi$	Distribution
3. $\Box(\varphi \wedge \psi) \to \Box\varphi$	Modus Ponens (2,1)

This tells us that every world of every model satisfies $\Box(\varphi \wedge \psi) \to \Box\varphi$. Or, it will once we prove the soundness theorem. Notice in the last step we used Modus Ponens to derive our result, and we listed in the description which two formulas were used. It is not necessary to include these, as there are only finitely many steps before any instance of Modus Ponens, so we can try every possible pair of formulas to check that the proof is valid. In the interest of human sanity, though, it is worth the extra time, especially as proofs get long.

As an example of proofs getting long, let's see one last (slightly more complicated) K-derivation. We'll show

$$K \vdash \Box(\varphi \wedge \psi) \to \Box\varphi \wedge \Box\psi$$

1. $\varphi \wedge \psi \to \varphi$	CPL
2. $\varphi \wedge \psi \to \psi$	CPL
3. $\Box(\varphi \wedge \psi \to \varphi)$	Necessitation (1)
4. $\Box(\varphi \wedge \psi \to \psi)$	Necessitation (2)
5. $\Box(\varphi \wedge \psi \to \varphi) \to \Box(\varphi \wedge \psi) \to \Box(\varphi)$	Distribution
6. $\Box(\varphi \wedge \psi \to \psi) \to \Box(\varphi \wedge \psi) \to \Box(\psi)$	Distribution
7. $\Box(\varphi \wedge \psi) \to \Box\varphi$	Modus Ponens (5,3)
8. $\Box(\varphi \wedge \psi) \to \Box\psi$	Modus Ponens (6,4)
9. $(\Box(\varphi \wedge \psi) \to \Box\varphi) \to \Big((\Box(\varphi \wedge \psi) \to \Box\psi) \to \Box(\varphi \wedge \psi) \to \Box\varphi \wedge \Box\psi\Big)$	CPL
10. $(\Box(\varphi \wedge \psi) \to \Box\psi) \to \Box(\varphi \wedge \psi) \to \Box\varphi \wedge \Box\psi$	Modus Ponens (9,7)
11. $\Box(\varphi \wedge \psi) \to \Box\varphi \wedge \Box\psi$	Modus Ponens (10,8)

Well, that certainly escalated quickly, didn't it. Even simple derivations can be quite long – it's the unfortunate price we pay for formalism. When we aren't allowed any shortcuts, we see how complicated even our simple reasoning really is. It's honestly kind of encouraging, and puts mathematics into perspective (at least for me). Epistemically this theorem says nothing but "If Valerie knows (φ and ψ), then she knows φ and she knows ψ." This is intuitively obvious. Of *course* it should be true at every world of every model. The derivation system seems woefully obtuse, but I promise it will be one of the most powerful tools in our arsenal, because it can be completely automated. We will see in section 4.8 that, for many classes of interest, we can write a computer program which tells us if a given sentence is valid on that class of frames, no matter how complicated that sentence is. This computer program is extremely useful in applications, and is only possible because of this proof-theoretic formalism. Indeed, it is unweildy for us humans to work with, but it is a simple recursive definition for a computer.

Ex. 2.30 —
It is OK to cite every theorem of CPL as an axiom because the theory of CPL is **Decidable**. We can write a computer program which takes a formula in the language of CPL and *tells* us whether it is or isn't a theorem of CPL. Write such a computer program. (Hint: any formula can only refer to finitely many variables, and each variable can only take finitely many values.)

Ex. 2.31 —
Show that $\varphi \wedge (\varphi \to \psi) \to \psi$ is valid in CPL.

Ex. 2.32 —
Step 9 in the above proof stands out somewhat, and it is worth talking about it. We used it in order to create an "and" using only Modus Ponens. We claimed that it was an instance of CPL, but this might be difficult to see.

a. Show $(A \to B) \to ((A \to C) \to (A \to B \wedge C))$ is true in CPL.
b. Show step 9 in the above proof is an instance of this axiom.

2.5 Other Deduction Systems

Modal Logic has been used to model many things in its history, and differing applications have differing needs. It is common to expand system K by adding new *bonus* axioms which reflect the assumptions at hand, and there are standard names for the most frequently used extensions.[5] The price we pay for these new axioms, which let us derive more things, is a restriction of soundness. Each of these logics will only be sound on a subclass of all frames. However, in most settings, this is a small price to pay, and the restriction of frames can be viewed as ignoring those frames which are irrelevant for some application. We will show that the restrictions mentioned below are correct in section 3.6, but if you are feeling adventurous you can try to convince yourself

[5] Unfortunately, the naming convention is rather convoluted, as you will soon see.

that any frame outside of the mentioned class will have a model and a world which thinks the axiom is false.

Axiom T	$\Box\varphi \to \varphi$
Axiom D	$\Box\varphi \to \Diamond\varphi$
Axiom 4	$\Box\varphi \to \Box\Box\varphi$
Axiom 5	$\Diamond\varphi \to \Box\Diamond\varphi$

Axiom T is the easiest to explain, and the most common axiom in use. In almost every interpretation of $\Box$, we want to know that something which has some qualified notion of truth should also be true. For instance, if Valerie knows it is Sunny, then it should also *actually be Sunny*! Epistemically, Axiom T corresponds to the *Factivity of Knowledge*, and it forces our frames to be reflexive.

Axiom D also makes sense epistemically – It says that if Valerie knows that it is sunny, she considers it possible that it is sunny. Intuitively, *knowing* something should be more restrictive than *considering*, and this axiom enforces that. Temporally, this axiom says that if φ is true at all future times, then there is actually a time in the future where φ is true. A snappy way of saying this is that *time has no end* – if there were a way to have φ true in the future, but never actually be true. . . then there must not be a future time. Axiom D forces our frames to be serial.

Axiom 4 is typically used in logics of time. If, starting today, it will never rain again, then it is also true that, starting in a few weeks, it will never rain again. Every future of a few weeks from now is also a future of today. When Axiom 4 is used epistemically, it says that "If Valerie knows φ, then she knows she knows φ." This is called *Positive Introspection*, and forces our frames to be transitive. That said, Axiom 4 tends to not be used alone. If we additionally add Axiom T, then we have a system valid on transitive and reflexive frames – Partial orders satisfy both of these, and the preference of a system which also satisfies T is similar to the preference of $\preceq$ over $\prec$ in partial orders. Axiom 5 is a levelled up version of Axiom 4, and there isn't much else to say. It is such a strong axiom that its frames are heavily constrained (they must be equivalence relations). For our purposes, Axiom 5 won't come up very often, but it is good to know about.

Ex. 2.33 —
We can view axiom D as being a weakened version of axiom T, and indeed many philosphers treat it that way. Let's justify this intuition.

a. Show that if $\mathfrak{M} \vDash T$, then $\mathfrak{M} \vDash D$
b. Find a model $\mathfrak{M} \vDash D$ but $\mathfrak{M} \nvDash T$
c. In terms of frames, why does it make sense that T should imply D?
d. Epistemically, why does it make sense that T should imply D?
e. Epistemically, why does it make sense that D should *not* imply T?

Ex. 2.34 —
Show that if $\mathfrak{M}$ is a model of S4, then $\mathfrak{M} \vDash \Diamond\Diamond\varphi \to \Diamond\varphi$

Ex. 2.35 —
Show that if $\mathfrak{M}$ is a model of S5, then $\mathfrak{M}$ models S4

Ex. 2.36 —
There is heated debate in some communities between S4 and S5. Show that the difference is really one of Axiom B, which we define to be $\varphi \to \Box\Diamond\varphi$. That is, show $\mathfrak{M} \vDash$ S5 if and only if $\mathfrak{M} \vDash$ S4 and $\mathfrak{M} \vDash B$.

Ex. 2.37 —
Show that, in S4, we can expand and condense repeated modalities. Precisely, if we define $\Box^n\varphi$ to be $\underbrace{\Box\cdots\Box}_{n}\varphi$, then $\text{S4} \vdash \Box^n\varphi \leftrightarrow \Box\varphi$.

Similarly, show $\text{S4} \vdash \Diamond^n\varphi \leftrightarrow \Diamond\varphi$

Ex. 2.38 —
Show that, in S5, we can expand and condense any modalities. Precisely, if we have a block of modalities, the truth only depends on the innermost one. As two examples:

$$\Box\Diamond\Box\Box\varphi \leftrightarrow \Box\varphi$$

$$\Diamond\Diamond\Box\Diamond\Diamond\varphi \leftrightarrow \Diamond\varphi$$

Ex. 2.39 —
Proof system for MAEL – we want T for each agent

Ex. 2.40 —
Proof system for LTL

Ex. 2.41 —
Recall from exercise 2.2 the logic of Knowledge and Belief. Say we axiomatize this with $B\varphi \to BK\varphi$ (this axiom is called ???)

a. What does this axiom mean, philosophically?

b. Prove something using this axiom

Ex. 2.42 —
Derive the following theorems of T, then prove they are valid semantically (assuming every model of T is reflexive)

a. $\mathrm{T} \vdash$

b. $\mathrm{T} \vdash$

c. $\mathrm{T} \vdash$

Ex. 2.43 —
Derive the following theorems of S4, then prove they are valid semantically (assuming every model of S4 is reflexive and transitive)

a. $\mathrm{S4} \vdash$

b. $\mathrm{S4} \vdash$

c. $\mathrm{S4} \vdash$

Ex. 2.44 —
Here is a language with a 3-ary relation symbol, here are its semantics. Prove some basic facts.

Ex. 2.45 —Consider the frame $\mathfrak{F} = (\mathbb{R}^2, R)$ where $(x_1, y_1)R(x_2, y_2)$ if and only if $x_1^2 + y_1^2 = x_2^2 + y_2^2$. Find a geometric interpretation of R.

Ex. 2.46 —More problems with geometric interpretations!!

Ex. 2.47 —Show that if $(\mathfrak{M}, x) \vDash \varphi \to \psi$ and $(\mathfrak{M}, x) \vDash \varphi$ then $(\mathfrak{M}, x) \vDash \psi$

Ex. 2.48 —Something to do with a temporal logic other than LTL

Ex. 2.49 —The muddy children puzzle

Chapter 3

Soundness and Completeness

3.1 Introduction

We have seen how to use Kripke Semantics to provide truth values to formulas in the basic modal language. Additionally, we have seen the definition of a proof system, K, which allows us to derive sentences in this language. We have hinted at the fact that the proof system and the semantics are related and in this chapter we will prove the following theorem:

$$K \vdash \varphi \iff \mathcal{C}_{\text{all}} \vDash \varphi$$

This says that a formula φ should be *derivable* in K if and only if that formula is *valid*, or true at every world of every model. The forward direction $K \vdash \varphi \implies \mathcal{C}_{\text{all}} \vDash \varphi$ is called soundness. After we prove this theorem, we will show how to modify it to quickly proe other soundness results. For instance:

$$T \vdash \varphi \implies \mathcal{C}_{\text{refl}} \vDash \varphi$$

3.2 The Soundness Theorem

First, let us fully define the notion of **Soundness** which we have been discussing:

Definition 3.1. A derivation system Λ is called **sound** with respect to a language $\mathcal{L}$ and a class of models $\mathcal{C}$ if and only if for every $\varphi \in \mathcal{L}$

$$\Lambda \vdash \varphi \implies \mathcal{C} \vDash \varphi$$

Theorem 3.2. K is a sound axiomatization of $\mathcal{L}$ with respect to $\mathcal{C}_{\text{all}}$

Proof. Assume $K \vdash \varphi$, that is, assume there is a K-derivation of φ. We proceed by induction on the length of this derivation.

If the derivation has length 1, then it must be a lone instance of an axiom scheme, so it suffices to show that each axiom scheme is true at every world of every model. Let (W, R, v) be a model, and $x \in W$ a world. That $x \vDash \varphi$ when φ is a CPL tautology was exercise 2.24 from chapter 2.

So we must show $x \vDash \Box(\varphi \to \psi) \to \Box\varphi \to \Box\psi$. If $x \nvDash \Box(\varphi \to \psi)$ or $x \nvDash \Box\varphi$ then by the definition of $\to$ in $\mathcal{L}$ (recall $\varphi \to \psi$ is an abbreviation for $\neg\varphi \vee \psi$) we are done. So assume $x \vDash \Box(\varphi \to \psi)$ and $x \vDash \Box\varphi$. Then by the definition of $\Box$, $\forall y \in R(x).y \vDash \varphi \to \psi$ and $\forall y \in R(x).y \vDash \varphi$. So then $\forall y \in R(x).y \vDash \psi$, and $x \vDash \Box\psi$, as desired.

If the derivation has length $n+1$, then the $n+1$th line is an instance of an axiom scheme, an instance of modus ponens, or an instance of necessitation. In the axiom scheme case, the base case proof still works. In the modus ponens case, in order to derive ψ we must have φ and $\varphi \to \psi$ earlier in the proof, by definition. Then, by induction, φ and $\varphi \to \psi$ are both true at every world in every model. If x is some point of some model, then in particular φ and $\varphi \to \psi$ are true at x, so by exercise 2.31 ψ is true at x, as desired. Finally, in the necessitation case, to derive $\Box\varphi$ we must have derived φ earlier in the proof. So by induction φ is true at every point of every model. Then let x be a point of some model with relation R. If $y \in R(x)$, then $y \vDash \varphi$, since by induction *every* world of every model satisfies φ. In particular so does y. Then $\forall y \in R(x).y \vDash \varphi$ and $x \vDash \Box\varphi$, as desired. ■

This technique of inducting on the length of the proof will be extremely useful in proving results in modal logic. However, as lazy mathematicians, we should strive to save ourselves time when possible. Because many proof systems in Modal Logic are just K with bonus axioms, we can prove soundness of other systems efficiently by leaning on this proof. Notice that when proving that Modus Ponens and Necessicitation preserve the truth of sentences (the inductive step in the above proof) we used nothing about the axioms of K. We only made use of the truth of sentences occuring earlier on in the derivation. Thus, even if we add new axioms, this step of the proof goes unchanged! Let's use this observation to quickly prove a new soundness result:

Corollary 3.3. T is sound with respect to $\mathcal{C}_{\text{refl}}$

Proof. Assume $T \vdash \varphi$. We proceed by induction on the proof length. If the proof length is 1, then it must be an instance of an axiom scheme, however any axiom of T which is shared by K is valid on every model, in particular models in $\mathcal{C}_{\text{refl}}$. So it suffices to show that $\Box\varphi \to \varphi$ is valid on $\mathcal{C}_{\text{refl}}$. Let x be a world in a reflexive model, and assume $x \vDash \Box\varphi$. By reflexivity, $x \in R(x)$, so by the definition of $\Box\varphi$, $x \vDash \varphi$. As desired.

If the proof length is $n+1$, then notice that line $n+1$ is an instance of an axiom scheme, an instance of modus ponens, or an instance of necessitation. But in the proof of the soundness of K, we have shown these are truth preserving on all models, in particular models in $\mathcal{C}_{\text{refl}}$. So we are done. ■

This technique, of using the soundness of K to quickly prove soundness of extensions of K, is an extremely powerful tool in the characterization of modal logics. For instance, look how quickly we can prove that K4 is sound with respect to $\mathcal{C}_{\text{trans}}$ now that we are familiar with the technique!

Corollary 3.4. K4 is sound with respect to $\mathcal{C}_{\text{trans}}$

Proof. By the soundness of K, it suffices to show every world of every transitive model satisfies $\Box\varphi \to \Box\Box\varphi$. Let x be a world in a transitive model, and assume $x \vDash \Box\varphi$. Let $y \in R(x)$ and $z \in R(y)$ (if either of these sets are empty, then the claim is vacuously true). By transitivity, $z \in R(x)$ and so $z \vDash \varphi$. Thus $y \vDash \Box\varphi$ and $x \vDash \Box\Box\varphi$, as desired. ■

Of course, here is the fastest proof available to a textbook author:

Corollary 3.5. S4 is sound with respect to $\mathcal{C}_{\text{Poset}}$

Proof. Exercise 3.1 ∎

Ex. 3.1 — Prove S4 is sound with respesct to the class of posets.

Ex. 3.2 —Prove S5 is sound with respect to the class of equivalence relations.

Ex. 3.3 —Find an axiom which, when added to K, is sound with respect to the class of symmetric models

Ex. 3.4 — Show that the proof theory for Multi-Agent Epistemic Logic which takes as axioms CPL, as well as Distribution and T for each K_i individually, plus Modus Ponens and a Necessitation for each K_i is sound with respect to all frames where every relation R_i is reflexive.

Ex. 3.5 —Something with a higher-arity modality?

Ex. 3.6 — Define a logic w/ $\overleftarrow{\Box}$ and $\overrightarrow{\Box}$ and use it to talk temporally about the past. Show these axioms
K for $\overleftarrow{\Box}$ and $\overrightarrow{\Box}$
$\varphi \to \overleftarrow{\Box}\overrightarrow{\Box}\varphi$
$\varphi \to \overrightarrow{\Box}\overleftarrow{\Box}\varphi$
$\overleftarrow{\Box}\varphi \to \overleftarrow{\Box}\overleftarrow{\Box}\varphi$ $\overrightarrow{\Box}\varphi \to \overrightarrow{\Box}\overrightarrow{\Box}\varphi$
Necc for both and MP
are sound with respect to all models with a transitive relation.

Ex. 3.7 —Show belief + knowledge from 2.2 is sound wrt some axioms on some class of models.

3.3 Completeness

The topic of *completeness* is dual to the topic of *soundness*, which we discuseed in the previous section. Where a proof system is sound (with respect to a given class of models) when all its theorems are valid on that class of models, it is called complete if every valid sentence is a theorem. That is, Λ is complete with respect to a class $\mathcal{C}$ of models whenever

$$\mathcal{C} \models \varphi \implies \Lambda \vdash \varphi$$

While proofs of soundness are typically simple inductions on the length of derivation, proofs of completeness tend to be more involved, as they enable us to turn a true statement into its derivation, at least abstractly. Since derivations can be long and complicated, it is often easier to argue by contrapositive, and take a nontheorem (something which has no proof) and find a world of a model which thinks it is false. For System K, we will take this second approach, and we will do so spectacularly. We will construct *one* model, called the **Canonical Model** for K, written $\mathfrak{C}_K$, which simultaneously refutes all nontheorems! That is, if φ is not a theorem of K, then $\mathfrak{C}_K \not\models \varphi$, and so φ cannot be a tautology. It is incredibly non-obvious that such a model should exist, but in a certain sense, its construction is natural. We will bake into the model everything we could possibly want to know about tautologies of K. By using this approach, we will prove

$$\mathfrak{C}_K \models \varphi \implies K \vdash \varphi$$

and thus, since $\mathfrak{C}_K \in \mathcal{C}_{\text{all}}$, we will show

$$\mathcal{C}_{\text{all}} \models \varphi \implies K \vdash \varphi.$$

This will follow because, if $\mathcal{C}_{\text{all}} \models \varphi$, every model in $\mathcal{C}_{\text{all}}$ satisfies φ. But if *every* model satisfies φ, in particular $\mathfrak{C}_K$ does. But by the theorem we will prove, if $\mathfrak{C}_K \models \varphi$, we must have $K \vdash \varphi$, and completeness follows.

3.4 Maximally Consistent Sets

The big idea in the construction of $\mathfrak{C}_K$ is that of **Consistent Sets**. Like it says on the tin, a consistent set of formulas is a collection of formulas which are logically consistent. For example, the set $X = \{Kp, p\}$ is consistent, whereas $Y = \{Kp, \neg Kp\}$ is inconsistent.

Intuitively, the two facts "Valerie knows p" and "p is true" are consistent. It is entirely possible that both statements are true simultaneously, and so X is *consistent*. However, it cannot be the case that "Valerie knows p" and "Valerie doesn't know p" are true at the same time. Because of this, Y is *inconsistent*.

There is one last important observation before we make the definition: Since derivations hae finite length, any derivation using a set Σ to reach a contradiction can only refer to a finite number of sentences in Σ. This leads us to the following definitions:

Definition 3.6. If Λ is a proof system, a set of formulas Σ is called (Λ -)**Inconsistent** iff there is a finite subset $\{\varphi_1, \varphi_2, \varphi_3, \ldots, \varphi_n\} \subseteq \Sigma$ such that

$$\Lambda \vdash \neg(\varphi_1 \wedge \varphi_2 \wedge \cdots \wedge \varphi_n)$$

We often abbreviate such formulas as $\bigwedge_{i \leq n} \varphi_i$, or as $\bigwedge \Delta$, when Δ is a finite set of sentences.

Σ is called (Λ -)**Consistent** iff it is not inconsistent. That is, if

$$\Lambda \nvdash \neg \bigwedge \Delta$$

for each finite $\Delta \subseteq \Sigma$.

Further, we say Σ is (Λ –)**maximally consistent** iff:

- Σ is consistent
- $\forall \varphi \notin \Sigma \ . \ \Sigma \cup \{\varphi\}$ is not consistent.

That is to say, not only is Σ consistent, it is maximally so. Any extra formula we add will make Σ inconsistent.

But how does this help us construct a model that refutes every nontheorem? The answer, at first glace, is surprising: We will take the worlds of $\mathfrak{C}_K$ to be the maximally consistent sets! We will want our model to satisfy the following important (and extremely natural) relation:

$$\mathfrak{C}_K, \Sigma \vDash \varphi \iff \varphi \in \Sigma$$

Of course, to make this happen, we will need to cleverly construct a relation and a valuation function for $W = \{\Sigma \mid \Sigma \text{ maximally consistent}\}$. Before we do, however, there is one more important consideration: We know that every world should satisfy either φ or $\neg\varphi$, but if our intuitive definition is $\Sigma \vDash \varphi \iff \varphi \in \Sigma$, then we might have a problem: What if $\varphi, \neg\varphi \notin \Sigma$? Then it's not clear what Σ should think about φ. Thankfully, this can never happen. This is where we will use the fact that our worlds are *maximally* consistent sets.

Theorem 3.7. If Σ is maximally consistent, then for any formula φ, exactly one of φ and $\neg\varphi$ is in Σ.

Proof. It is clear that φ and $\neg\varphi$ cannot both be in Σ, as this would violate consistency. Then it suffices to show at least one of the two is in Σ.

Towards a contradiction, assume that φ and $\neg\varphi$ are not in Σ. By maximality of Σ, $\Sigma \cup \{\varphi\}$ is inconsistent. (since $\varphi \notin \Sigma$) So fix $\Gamma \subseteq \Sigma$ with Γ finite and $\neg \bigwedge(\Gamma \cup \{\varphi\})$ provable. That is to say, $\neg (\bigwedge \Gamma \wedge \varphi)$, or, equivalently, $(\bigwedge \Gamma) \to \neg\varphi$ is provable.

But since $\neg\varphi \notin \Sigma$, there must be some $\Delta \subseteq \Sigma$ with Δ finite and $\Delta \cup \{\neg\varphi\}$ inconsistent. Then $\bigwedge \Delta \to \neg\neg\varphi$ is provable by similar reasoning as above. So $\bigwedge \Delta \to \varphi$ and $\bigwedge \Gamma \to \neg\varphi$ are both provable. So $\bigwedge (\Delta \cup \Gamma) \to (\varphi \wedge \neg\varphi)$ is provable too. Namely $\bigwedge (\Delta \cup \Gamma) \to \bot$ is provable, so $\neg \bigwedge (\Delta \cup \Gamma)$ is provable. But $\Delta \cup \Gamma$ is a finite subset of Σ. So Σ is inconsistent, yielding the desired contradiction. ■

This gives us insight into the worlds of $\mathfrak{C}_K$, but we still need to define a valuation and a relation in order to make a model.

Thankfully, if we want to know $\Sigma \vDash \varphi \iff \varphi \in \Sigma$, there is only one possible choice of valuation function: for a primitive proposition, we will need $\mathfrak{C}_K, \Sigma \vDash p \iff p \in \Sigma$, so we set $v(p) = \{\Sigma \mid p \in \Sigma\}$.

Finally, we will need a relation on these worlds. And we want the relation to make the above proposition true. That is, if $\Box\varphi \in \Sigma$, then we want $\Sigma \vDash \Box\varphi$. But this happens exactly when $\Sigma R \Gamma \implies (\Gamma \vDash \varphi)$. We want this to be the case for every $\Box\varphi \in \Sigma$, so we will take this as a definition: $\Sigma R \Gamma \iff \forall(\Box\varphi \in \Sigma).\varphi \in \Gamma$ Now the last step is to cover our tracks. As mathematicians it is important to obscure the process and show nothing but the completed definition. If our result makes too much sense, it runs the risk of not being published! After all, truly deep results must be confusing on first read.[1] We should put all of this pre-work into a definition, and act like we were divinely inspired to come up with it. Look how, magically, the thing we want to prove just *works* when we define things in this way! Little do they know we cooked up the definition precisely to make the theorem true!

Definition 3.8. The **Canonical Model** for K is the model $\mathfrak{C}_K = (W, R, v)$ such that:

- $W = \{\Sigma \mid \Sigma \text{ maximally K-consistent}\}$
- $\Sigma R \Gamma$ iff $\{\varphi \mid \Box\varphi \in \Sigma\} \subseteq \Gamma$
- $v(p) = \{\Sigma \mid p \in \Sigma\}$

Theorem 3.9. $\mathfrak{C}_K, \Sigma \vDash \varphi \iff \varphi \in \Sigma$

Proof. We induct on formulas. For p primitive, $\mathfrak{C}_K, \Sigma \vDash p$ iff $p \in \Sigma$ by the definition of v. As for the inductive cases:

Say $\mathfrak{C}_K, \Sigma \vDash \varphi$ iff $\varphi \in \Sigma$, and $\mathfrak{C}_K, \Sigma \vDash \psi$ iff $\psi \in \Sigma$.

$$\begin{aligned} \mathfrak{C}_K, \Sigma \vDash \neg\varphi &\iff \mathfrak{C}_K, \Sigma \nvDash \varphi \\ &\iff \varphi \notin \Sigma \\ &\iff \neg\varphi \in \Sigma \end{aligned}$$

The last equality follows since exactly one of φ or $\neg\varphi$ is in Σ

$$\begin{aligned} \mathfrak{C}_K, \Sigma \vDash \varphi \wedge \psi &\iff \mathfrak{C}_K, \Sigma \vDash \varphi \text{ and } \mathfrak{C}_K, \Sigma \vDash \psi \\ &\iff \varphi \in \Sigma \text{ and } \psi \in \Sigma \\ &\iff \varphi \wedge \psi \in \Sigma \end{aligned}$$

We know $\varphi \to \psi \to \varphi \wedge \psi \in \Sigma$, since it is a theorem of CPL, and thus an axiom of K (note axioms of K are always K-consistent with any set of K-consistent formulas). But then, since Σ is maximally consistent, it is closed under implication (cf. exercise 3.11). So $\varphi \wedge \psi \in \Sigma$.

[1] I'm being facetious, but an unfortunate amount of mathematics is acutally done like this...

Conversely, we know $\varphi \wedge \psi \to \varphi$ and $\varphi \wedge \psi \to \psi$ are axioms of K too. So if $\varphi \wedge \psi \in \Sigma$, we must have $\varphi \in \Sigma$ and $\psi \in \Sigma$ too.

Finally, we get to see our magical definition in action:

$$\begin{aligned} \mathfrak{C}_K, \Sigma \vDash \Box\varphi &\iff \forall \Gamma \in R(\Sigma). \mathfrak{C}_K, \Gamma \models \varphi \\ &\iff \forall \Gamma \in R(\Sigma). \varphi \in \Gamma \\ &\iff \Box\varphi \in \Sigma \end{aligned}$$

Here the last equality is because (by negating our definition of R) $\Box\varphi \notin \Sigma$ if and only if some $\Sigma R \Gamma$ for some Γ with $\varphi \notin \Gamma$. ■

It is also important to have a way of constructing maximally consistent sets with certain properties. This will let us know that there is a world of $\mathfrak{C}_K$ which does what we want. For this, we will show that maximally consistent sets are plentiful. In fact, we can grow any consistent set into a maximally consistent set. The process for this is intuitively simple: One can imagine a greedy gnome, who starts with a bag full of consistent sentences. The gnome wants to own as many sentences as possible, while still knowing that they are consistent. The gnome lays out all the possible sentences in front of itself, and then starts walking down the line of sentences. Everytime the gnome gets to a new sentence, it checks if that sentence is consistent with the ones in its bag. If it is, the greedy gnome picks up the sentence to keep! If not, the gnome passes it by, hoping for better luck with the next sentence in the line. At the end of time, our (very tenacious) gnome will have collected a *Maximally Consistent* set of sentences in its bag, as the following theorem shows:

Theorem 3.10. Any consistent set can be grown to a maximally consistent set

Proof. Let Σ_0 be a consistent set. We will construct a maximally consistent set Σ such that $\Sigma_0 \subseteq \Sigma$. Since there are only countably many formulas in the basic modal language (cf. exercise 3.9), write them as $\{\varphi_i\}$ for $i \in \mathbb{N}$. Now we will go one at a time and decide which sentences are worth keeping.

We define $\Sigma_{i+1} :\equiv \begin{cases} \Sigma_i \cup \{\varphi_i\} & \text{if } \Sigma_i \cup \{\varphi_i\} \text{ is consistent} \\ \Sigma_i & \text{otherwise} \end{cases}$

Finally, we define $\Sigma = \bigcup_i \Sigma_i$

Clearly $\Sigma_0 \subseteq \Sigma$, but is Σ maximally consistent? First, why is Σ even consistent? By definition, each Σ_i is consistent. This is because Σ_0 is consitent, and for each i, Σ_{i+1} is only different from Σ_i (which is consistent by induction) if the new formula φ_i is consistent!

Then if Σ *weren't* consistent, there must be some finite set of formulas $\Delta \subseteq \Sigma$ which is inconsistent. Let φ_n be the largest of the $\varphi_i \in \Delta$. Then $\Delta \subseteq \Sigma_n$, which is consistent! We contradict, and see Σ is consistent.

We still have to show why Σ is *maximally* consistent. Let $\psi \notin \Sigma$ be some formula. We must show that $\Sigma \cup \{\psi\}$ is inconsistent. ψ must have shown up in our enumeration, so call it φ_k But $\varphi_k \notin \Sigma$, so $\varphi_k \notin \Sigma_{k+1}$, so by the definition of Σ_{k+1}, $\Sigma_k \cup \{\varphi_k\}$ must be inconsistent. Since $\Sigma_k \subseteq \Sigma$, we see that $\Sigma \cup \{\varphi_k\} = \Sigma \cup \{\psi\}$ is also inconsistent, so Σ is maximally consistent. ■

The proof of the above theorem hinges on the ability to enumerate the φ, but there are logics one might be interested in where this is not possible. But fear not! Using the axiom of choice, we can still save this argument. The only hitch is that the gnome might have to keep walking until the end of time many *many* times over. Readers familiar with transfinite induction should prove this in exercise 3.10. Readers unfamiliar with transfinite induction should take this on faith (or familiarize themselves with it and then prove it).

With all this machinery in place, the completeness proof seems easy!

Theorem 3.11. K is complete with respect to the class of all models

Proof. Let φ be a nontheorem of K. Then $\{\neg\varphi\}$ is a consistent set, so we can extend it to a maximally consistent set $\Sigma_{\neg\varphi}$. Now $\mathfrak{C}_K, \Sigma_{\neg\varphi} \models \neg\varphi$, and $\mathfrak{C}_K, \Sigma_{\neg\varphi} \not\models \varphi$. As desired. ■

This is somewhat remarkable, as it shows that there is exactly one model which can detect nontheorems of K. Understanding K as a proof system and understanding $\mathfrak{C}_K$ as a model are the same thing! Provability is exactly encoded by satisfiability in $\mathfrak{C}_K$. In exercise 3.12 you will finish proving this for yourself.

Ex. 3.8 —
Here is an alternate definition of R in the canonical model. Show that the soundness proof still works.

Ex. 3.9 —
Show that there are only countably many sentences in the BML (Hint: How many strings of symbols of length n are there, whether or not they are meaningful. Then how many (possibly meaningless) strings are there total? The number of meaningful srings will be a subset of this.)

Ex. 3.10 — (requires some familiarity with set theory)
Assuming the Axiom of Choice, show that every consistent set of formulas can be extended to a maximally consistent set of formulas for any modal logic.

Ex. 3.11 —
Prove that whenever Σ is maximally consistent, if $\varphi \to \psi \in \Sigma$ and $\varphi \in \Sigma$, then $\psi \in \Sigma$ too.

Ex. 3.12 —
Prove $K \vdash \varphi \iff \mathfrak{C}_K \vDash \varphi$.

3.5 Completenesss in Other Modal Logics

As was the case for soundness, now that we have completeness for K, it is quick to find similar completeness proofs for other modal logics.

To illustrate the general strategy, let's see if we can show that T is complete with respect to some class of models. We know that T is complete with respect to $\mathcal{C}_{\text{all}}$, since anything valid on all frames is a theorem of K, and any theorem of K is a theorem of T. We can make this sharper, though – T is sound with respect to $\mathcal{C}_{\text{refl}}$, and there are formulas which $\mathcal{C}_{\text{refl}}$ validates that $\mathcal{C}_{\text{all}}$ doesn't (T, for example). Can we show that T is complete with respect to $\mathcal{C}_{\text{refl}}$ too?

Theorem 3.12. T is complete with respect to $\mathcal{C}_{\text{refl}}$

Proof. As before, we argue by contrapositive. If φ is a nontheorem of T, we want to be able to refute it on a model. The difference now is that we want to refute it on a *reflexive* model. Let's do what we did before and see what happens.

We construct the canonicial model $\mathfrak{C}_T$ by taking as worlds all maximal $T-$consistent sets of formulas, and defining (as before)

$$\Sigma R \Gamma \iff \{\varphi \mid \Box\varphi \in \Sigma\} \subseteq \Gamma$$

As before, we have $\Sigma \models \varphi \iff \varphi \in \Sigma$, since our proof of that fact did not rely on what axioms we were using. Further, notice that, since T is an axiom, it is consistent with any T-provably consistent set of formulas. So for each φ, $\Box\varphi \to \varphi \in \Sigma$ (since these are the instances of axiom T). Then, by the definition of R, we see $\Sigma R \Sigma$, and $\mathfrak{C}_T$ is reflexive!

Finally if φ is a non-theorem of T, then $T \not\vdash \neg\varphi$ so we can extend $\{\neg\varphi\}$ to a maximally T-consistent set Σ. Then $\mathfrak{C}_T, \Sigma \models \neg\varphi$, and φ is invalidated by a reflexive frame!

Thus, T is complete with respect to $\mathcal{C}_{\text{refl}}$. ■

Again, let's speed things up a bit now that we're getting the hang of things.

Theorem 3.13. K4 is sound with repect to $\mathcal{C}_{\text{trans}}$

Proof. Let φ be a nontheorem of K4. Then $\{\neg\varphi\}$ is K4-consistent and can be extended to a maximally K4-consistent set Σ. We construct the canonical model $\mathfrak{C}_{K4}$ as usual, but recognize $\Box\varphi \to \Box\Box\varphi$ is consistent for every φ, since it is Axiom 4. Thus $\mathfrak{C}_{K4} \models K4$, and R must be transitive. Thus $\Sigma \not\models \varphi$ invalidates φ on a transitive model, proving the claim. ■

Ex. 3.13 —

a. Prove S4 is *not* complete with respect to the class of all posets.

b. Find a class of frames for which S4 *is* complete, and prove it.

Ex. 3.14 —Prove D is complete with respect to the class of serial frames

Ex. 3.15 — Prove T and 4 are not theorems of K (Hint: find a frame invalidating them, then argue based on K's completeness with respect to all frames)

Ex. 3.16 —Show that the proof theory for Multi-Agent Epistemic Logic outlined in exercise 3.4 is complete with respect to all frames where every relation R_i is reflexive.

Ex. 3.17 —Show the proof theory for (the one with $\overline{\Box}$) given in exercise 3.6 is complete with respect to the class of transitive frames.

3.6 Definability

Notice that there is a clear connection between graph theoretic properties of the underlying frame class, and certain axioms we could add to K. For instance, what must it mean about the underlying frame if the formula $\Box\varphi \to \varphi$ is true in every world of every model built on a certain frame?

It must mean that every world can see itself. That is, our frame is **Reflexive**. To see this, assume towards a contradiction that our frame is not reflexive. Then there exists a world x with $x \not R\, x$. Then consider the valuation where p is true everywhere *except* x. Then $x \vDash \Box p$ but $x \nvDash p$.

However, without prior knowledge, how could we have come up with this? Notice that if we insist on $\Box\varphi \to \varphi$ being true at every world, then we need to *ensure* φ must be true, knowing only that $\Box\varphi$ is true. Well, the only thing we *know* to be φ worlds are the worlds related to x, by the definition of $\Box\varphi$. So in order to ensure that x itself is a φ world, the only tool we have is to make sure x sees itself.

In this way, we have added an axiom to our logic, and in doing so, we have restricted the class of frames on which our logic is sound. Alternatively, we have restricted the class of frames which we are interested in, and we have updated our logic to reflect this.

Let's take a look at other potential axioms, some of their logical interpretations, and how they affect the frames on which we are defined.

3.6.1 Transitivity

Recall a frame is called **Transitive** if whenever xRy and yRz, then xRz. Consider the formula $\Box\varphi \to \Box\Box\varphi$. This says that "If every world x sees in one step satisfies φ, then every world x sees in two steps also satisfies φ. But again, the only way for us to prevent an adversary from stopping us is to guarantee that the world we reached in two steps was one of the things we already knew to be a φ world. That is, a world reachable in one step.

Towards a contradiction, assume a frame is not transitive, but satisfies $\Box\varphi \to \Box\Box\varphi$. Then fix xRy, yRz with $x \not R z$. Consider the valuation function where p is true everywhere except z. Then $x \vDash \Box p$, since x does not see z. However, $y \nvDash \Box p$, since y does see z. Then x cannot satisfy $\Box\Box p$.

3.6.2 Definability

Notice that not only are these formulas sufficient to show the desired frame condition, they are also minimal in the sense that they do not force any *other* frame conditions as well. Additionally, our frame conditions were minimal. Notice that, for $\Box\varphi \to \varphi$, we could have said that the collection of all frames with *only* self loops ensures that this formula is satisfied. However this is too restrictive. There are plenty of frames outside this class which also satisfy this axiom. We say the least restrictive class of frames which renders a set of axioms sound is *Defined by* those axioms. Formally:

Definition 3.14. A set of formulas A **Defines** a class of frames $\mathcal{C}_A$ iff

$$\mathfrak{F} \vDash A \iff \mathfrak{F} \in \mathcal{C}$$

For example, in the above sections, we proved half of definability for each of the frame properties. Namely, we showed that $\mathfrak{F} \vDash \varphi \Rightarrow \mathfrak{F} \in \mathcal{C}$. The opposite direction is easy, and was informally justified. It amounts to a soundness proof like those seen in section 3.2.

Ex. 3.18 —
Earlier, you showed that S4 was sound with respect to the class of posets. Show that S4 does *not* define the class of posets. What class *does* it define?

Ex. 3.19 —
Something definable using the logic of $\overrightarrow{\Box}$ and $\overleftarrow{\Box}$

Ex. 3.20 —
Show Symmetry is defined by $\varphi \to \Box\Diamond\varphi$

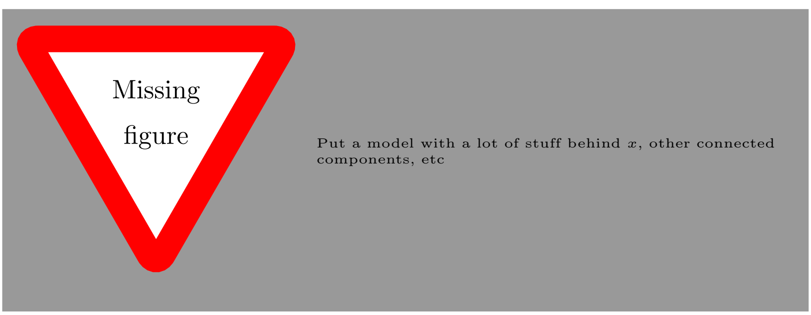
Chapter 4

Bisimulations and Operations on Frames

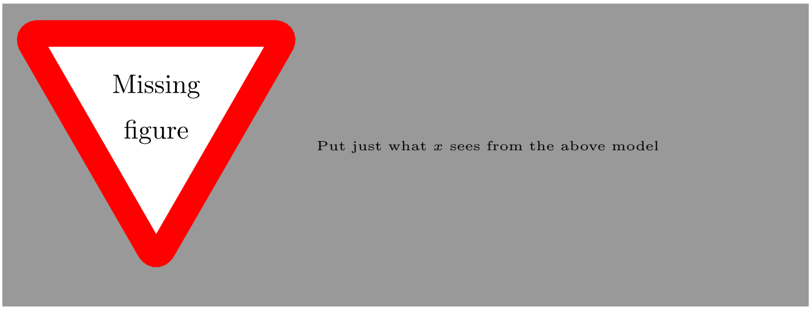
4.1 Introduction

Because truth of a modal formula is based on the model in which we interpret it, oftentimes we find ourselves analysing a model, when really we care about what the model *thinks* of a particular formula φ. Depending on what exactly we care about, we can often replace the model with a simpler one, at the cost of losing some information, which we hopefully don't care about.

As an example, say we are interested in understanding what formulas φ are true at x in the model below:

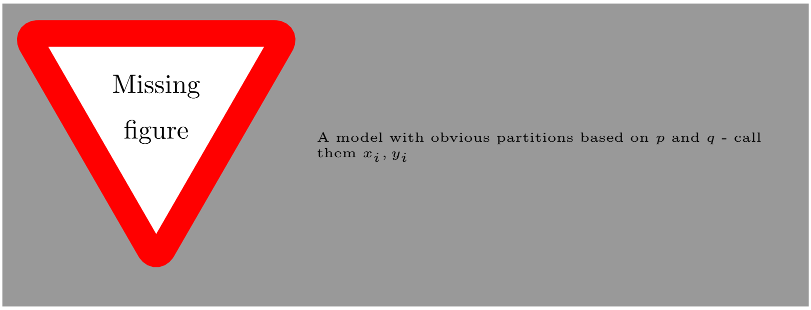


Since the truth of a modal formula at x only cares about what x can see, we can replace the complicated model above with the simpler version below:



Now we obviously lose a lot of information in making this transformation. $\mathfrak{M}_2$ carries no information regarding the truth of modal formulas at y, for instance. We have simplified the structure which we have to analyse, but in doing so we have lost the ability to reason about other worlds. Sometimes this is a wise trade-off to make, other times it is more useful to keep the entire model around. It all depends on what one is trying to study.

As another, perhaps less obvious, example of this trade-off, say we care about what each world thinks about, say, $\Box p \vee q$. Each world is important, but only with respect to whether it thinks $\Box p \vee q$ true or false. This is dual to the above example, in which we cared about every formula, but only one world. Note that, as far as $\Box p \vee q$ is concerned, we only care about a world's opinion on p and q, and the opinions of what worlds it sees. So to understand this model:



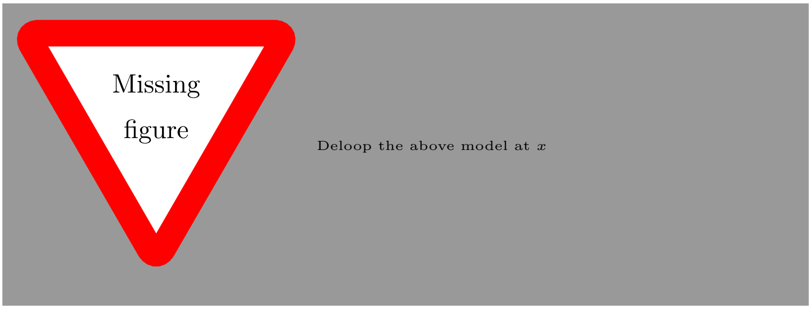
it suffices to understand this model

A filtration of the above model through $\Box p \vee q$ - call the classes X, Y

We can also preserve truth, while making the model simpler, in less obvious ways. Often cycles are hard to analyse, and in some settings we are willing to give up finiteness in order to "de-loop" our model. That is, we replace our model by one in which no world has a path to itself. For example:

A world with 2 states and a cycle

Here, we can construct the following model by "de-looping" our model at x:



Notice that, instead of $xRyRx$, in which x sees *itself* after two steps, instead x sees x_1 - a copy of x which looks exactly the same as far as modal formulas are concerned.

We will formalize these, and other, constructions in this chapter, and will give methods for proving rigorously what our intuition tells us: These constructions do actually preserve the properties we say they do. The main tool for this verification is called **Bisimulation**, and after properly introducing it, we will use it to show that all of the assertions above (indeed, some mild generalizations of the assertions above) are all true.

4.2 Bisimulations

A Bisimulation between two models $\mathfrak{M}_1$ and $\mathfrak{M}_2$ gives us a way to *relate* worlds in the models which "look the same" as far as modal formulas are concerned. That is, worlds which have the same theory.

What properties must we require of two worlds $w_1 \in \mathfrak{M}_1$ and $w_2 \in \mathfrak{M}_2$ to *guarantee* they will have the same theory? Obviously we need to know that $v_1(w_1) = v_2(w_2)$, as otherwise they will disagree on even primitive propositions! As for formulas involving $\Box$, we will need two conditions on B to do the job.

First, we need to know that $\mathfrak{M}_1$ is "powerful enough" to *simulate* $\mathfrak{M}_2$. What do we mean by this? Say x_0By_0, and $y_0R_2y_1$. We need to be able to *simulate* that move in $\mathfrak{M}_1$, by finding a x_1By_1 such that x_0Rx_1.

Intuitively, let's see why this condition is necessary: If $y_1 \vDash \varphi$, then $y_0 \vDash \Diamond\varphi$. If there were no $x_1 B y_1$, then there would be no $x_1 \vDash \varphi$, and so $x_0 \nvDash \Diamond\varphi$.

Similarly, we need to know that $\mathfrak{M}_2$ can simulate $\mathfrak{M}_1$. Say $x_0 B y_0$ and $x_0 R_1 x_1$. We need to find a y_1 such that $x_1 B y_1$ and $y_0 R_2 y_1$.

Again, let's intuitively see why we need this. Say $y_0 \vDash \Box\varphi$. Then that means every $y_1 \in R_2(y_0)$ satisfies φ. If $\mathfrak{M}_2$ cannot simulate $\mathfrak{M}_1$, then it is possible some $x_1^* \in R_1(x_0)$ goes unsimulated. If $x_1^* \nvDash \varphi$, then $x_0 \nvDash \Box\varphi$, so x_0 and y_0 have different theories. If, however, $\mathfrak{M}_2$ *can* simulate $\mathfrak{M}_1$, then x_1^* would have a matching y_1^* in $\mathfrak{M}_2$, contradicting the fact that every $y_1 \in R_2(y_0)$ satisfies φ.

So if we require that $\mathfrak{M}_1$ can simulate $\mathfrak{M}_2$, we are requiring every $y_1 \in R_2(y_0)$ to have a mate in $R_1(x_0)$ (for $x_0 B y_0$). By requiring $\mathfrak{M}_2$ simulate $\mathfrak{M}_1$, we are saying that *every* world that x_0 sees is related to some world y_0 sees. This is not unlike requiring both injectivity *and* surjectivity to say that a function is an equivalence (bijection).

With the intuition safely digested, we are prepared to see the formal definition:

Definition 4.1. A **Bisimulation** between models $\mathfrak{M}_1 = (W_1, R_1, v_1)$ and $\mathfrak{M}_2 = (W_2, R_2, v_2)$ is a relation $B \subseteq W_1 \times W_2$ such that whenever $w_1 B w_2$ the following properties hold:

- **Invariance**: $v_1(w_1) = v_2(w_2)$
- **Back**: $\forall y \in R_2(w_2) . \exists x \in R_1(w_1) . x B y$
- **Forth**: $\forall x \in R_1(w_1) . \exists y \in R_2(w_2) . x B y$

This definition is great and all, but let's spend some time unpacking what it means. At the end of the day, we want to be able to say that $x_1Bx_2 \implies T(x_1) = T(x_2)$, phrased differently:

$$x_1Bx_2 \implies \Big(\forall\varphi.(\mathfrak{M}_1, x_1) \vDash \varphi \iff (\mathfrak{M}_2, x_2) \vDash \varphi\Big)$$

That is, this relation should tell us that two worlds have the same theory. Importantly, notice that we *don't* ask for the converse. It might be the case theat y_1 and y_2 have the same theory, yet $y_1 \not B\, y_2$. This gives us extra flexibility, as it allows us to ignore worlds which we decide are unimportant for a particular application. Indeed, there are models $\mathfrak{M}_1$ and $\mathfrak{M}_2$ with worlds x_1 and x_2 such that x_1 and x_2 share a theory, but there is no bisimulation relating the two! You will show this in exercise 4.6.

Now that the caveats are out of the way, let's make this precise, and prove that our definition does the right thing:

Theorem 4.2. If B is a bisimulation between $\mathfrak{M}_1$ and $\mathfrak{M}_2$ and x_1Bx_2, then $\mathfrak{M}_1, x_1 \vDash \varphi \iff \mathfrak{M}_2, x_2 \vDash \varphi$

Proof. Let $x_1 \in \mathfrak{M}_1 = (W_1, R_1, v_1)$ and $x_2 \in \mathfrak{M}_2 = (W_2, R_2, v_2)$

Further, let $B \subseteq W_1 \times W_2$ be a bisimulation.

By induction on formulas φ, we will show $Th(x_1) = Th(x_2)$.

If φ is a primitive proposition p, then by the **Invariance** property of B, $v_1(x_1) = v_2(x_2)$ and so p is in one iff it is in both.

Inductively, then, consider $\neg\varphi$. Clearly

$$x_1 \vDash \neg\varphi \iff \text{not } x_1 \vDash \varphi \iff \text{not } x_2 \vDash \varphi \iff x_2 \vDash \neg\varphi$$

(since, by induction $x_1 \vDash \varphi \iff x_2 \vDash \varphi$)

Then, consider $\varphi \wedge \psi$.

$$x_1 \vDash \varphi\wedge\psi \iff x_1 \vDash \varphi \text{ and } x_1 \vDash \psi \iff x_2 \vDash \varphi \text{ and } x_2 \vDash \psi \iff x_2 \vDash \varphi\wedge\psi$$

Finally, consider $\Box\varphi$.

Assume $x_1 \vDash \Box\varphi$, and let $y_2 \in R_2(x_2)$. Then **Back** says there exists $y_1 \in R_1(x_1)$ such that y_1By_2. Then, since $x_1 \vDash \Box\varphi$, $y_1 \vDash \varphi$ and by induction, $y_2 \vDash \varphi$ too. So $x_2 \vDash \Box\varphi$.

Conversely, assume $x_2 \vDash \Box\varphi$, and let $y_1 \in R_1(x_1)$. Then **Forth** says there exists $y_2 \in R_2(x_2)$ such that y_1By_2. Then since $x_2 \vDash \Box\varphi$, $y_2 \vDash \varphi$ and by induction, $y_1 \vDash \varphi$ too. So $x_1 \vDash \Box\varphi$.

Thus, $x_1 \models \Box\varphi \iff x_2 \models \Box\varphi$, completing the proof. ■

Ex. 4.1 —
Give two models and a relation, prove it is a bisimulation

Ex. 4.2 —
Give two models and a relation, prove it is a bisimulation

Ex. 4.3 —
Give two models and a relation, prove it is a bisimulation

Ex. 4.4 —
Give two models and two worlds, find a bisimulation relating x and y

Ex. 4.5 —
Give two models and two worlds. Show there is *no* bisimulation relating x and y.

Ex. 4.6 —


These two frames are called the *hedgehogs* because of their resemblance to the pointy critters. The finite hedgehog is the model with one branch of each finite length, and the infinite hedgehog has one branch of each finite length, as well as one infinite length branch.
If we endow these frames with the valuation that thinks p is true at every world, (and we work with p the only primitive proposition), then show the following facts:

a. x and y have the same theory
b. There is no bisimulation relating x and y.

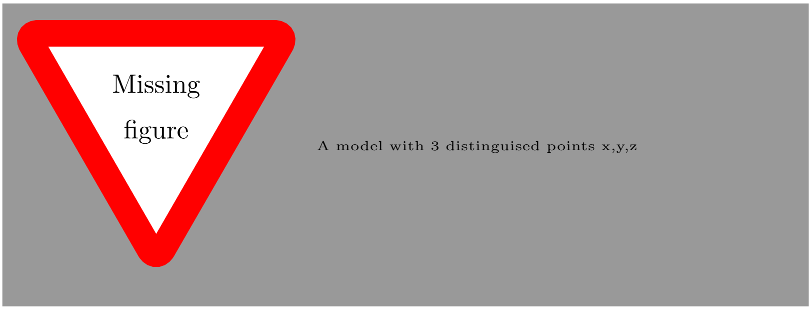
4.3 Generated Submodels

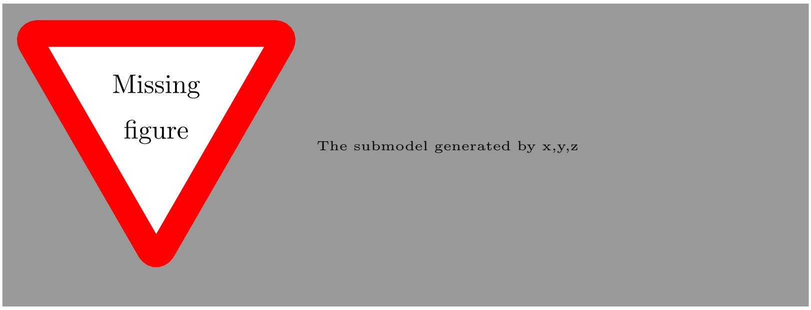
Generated Submodels provide possibly the most intuitive notion of a transformation that preserves truth for certain worlds in a model. Say $\mathfrak{M} = (W, R, v)$ is a model, and $x \in W$. Then let $W_x = \{w \in W | xR^*w\}$ be the worlds in W which x eventually sees (recall R^* is the *transitive closure* of R, as defined in 2.12). Then $\mathfrak{M}_x = (W_x, R\restriction_{W_x}, v\restriction_{W_x})$ is called the **Generated Submodel** of $\mathfrak{M}$ at x. See example 4.1 above. We will soon prove that $\mathfrak{M}_x, x \vDash \varphi \iff \mathfrak{M}, x \vDash \varphi$, so $\mathfrak{M}_x$ preserves the theory of x.

In fact, we can generalize slightly further already: Say $X \subseteq W$ is a set of worlds. Then put $W_X = \bigcup_{x \in X} W_x$. Then

$$\mathfrak{M}_X = (W_X, R\restriction_{W_X}, v\restriction_{W_X})$$

is the Generated Submodel of X, and it preserves the theory of every world in X!





Finally we arrive at the proof. We want to find a bisimulation B relating the generated submodel to the original model. Hmm... if only we had a natural way of identifying objects in W_X with objects in W...

I jest. Obviously we can put $xBy \iff x = y$, since $W_X \subseteq W$. I now claim that this is a bisimulation.

Theorem 4.3. If $\mathfrak{M}_X$ is a generated submodel of $\mathfrak{M}$, then $=\restriction_{W_X}$ is a bisimulation.

Proof. Said less snappily, we are saying $\{(w, w) \mid x \in W_X\}$ is a bisimulation.

Fix $w_1 \in \mathfrak{M}_X, w_2 \in \mathfrak{M}$ such that $w_1 B w_2$. We must prove **Invariance**, **Back**, and **Forth**.

Invariance is easy: If $w_1 = w_2$, then $v_X(w_1) = v(w_1) = v(w_2)$.

Back is slightly harder: Say $w_2 R z$. Since $w_1 = w_2$, $w_2 \in R^*(x)$ for some $x \in X$. Then $z \in R^*(x)$ too, so $z \in W_X$ and $w_1 R_X z$.

Forth is easy as can be: Say $w_1 R_X z$. Then zBz, and $w_1 R z$. $w_1 = w_2$, so $w_2 R z$ too. ■

4.4 Filtration

Dual to Generated Submodels are **Filtrations**. To make a Generated Submodel you start with a collection of worlds you care about, and you remember just enough about your model to preserve the theory of those worlds. Dually, to make a Filtration, you first start with a set of formulas, and you remember exactly enough of your model to preserve those formulas. Recall the example from above:

We can *filter* this model $\mathfrak{M}$ into a new model $\mathfrak{M}_\varphi$ by grouping $\mathfrak{M}$ into chunks as follows:

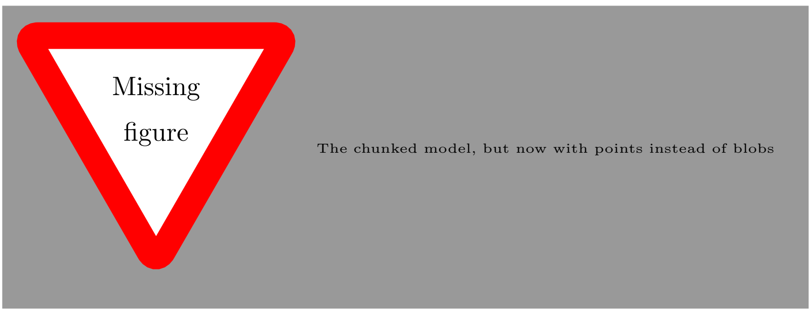
We will define a new relation on these chunks, by setting

$$R_\varphi(X, Y) \iff \exists x \in X. \exists y \in Y. R(x, y)$$

Copy the model of filtration from above (label the guys in the chunks as x_i, y_i)

Copy the post-filtration model

Now it is easy to check that x_i and X agree on φ, as do y_i and Y and z_i and Z. But clearly this "chunked" model is easier to analyse:



As with Generated Submodels, we will now move to a slightly more general definition for Filtration, though in spirit we are doing exactly as in the example above, but with multiple formulas.

Unlike with Generated Submodels, we will need some extra definitions in order to properly define filtrations.

Definition 4.4. A set of formulas Σ is **Subformula-Closed** if subformulas of $\varphi \in \Sigma$ are also in Σ.

Inductively:

- $\neg\varphi \in \Sigma$ implies $\varphi \in \Sigma$
- $\varphi \wedge \psi \in \Sigma$ implies $\varphi \in \Sigma$ and $\psi \in \Sigma$
- $\Box\varphi \in \Sigma$ implies $\varphi \in \Sigma$

By example, the following sets are subformula closed:

- $\{p \wedge \neg q, p, \neg q, q\}$,
- $\{\Box p \vee q, \Box p, q, p\}$,
- $\{\Box(p \to q), \Diamond r, p \to q, p, q, r\}$, and
- $\{p\}$

And the following aren't:

- $\{p \vee q, p\}$,
- $\{p \to q\}$,
- $\{\Box p \vee q, p \vee q, p, q\}$, and
- $\{\neg q\}$

Another definition we will need is the **Restricted Theory** of a world. Instead of considering *all* of the formulas, we only consider formulas in Σ.

Definition 4.5. If Σ is Subformula-Closed, then

$$T_\Sigma(w) = \{\varphi \in \Sigma | w \vDash \varphi\}$$

Clearly $\sim_\Sigma$ defined by $x \sim_\Sigma y \iff Th_\Sigma(x) = Th_\Sigma(y)$ is an equivalence relation, and we will denote the equivalence class of x by $[x]_\Sigma$, or just $[x]$ if Σ is clear from context.

The set of all equivalence classes will be denoted W_Σ

As in the definition of the Canonical Model (section 3.3), we know what we want to do, and we need to find a way to do it. We have an obviovus candidate for the worlds and valuation function for the filtration, but we need to find a relation which will preserve the restricted theory of every world once we quotient. To do this, we will need to know that $[x]R_\Sigma[y]$ whenever x and y used to be related. Some experimentation shows that this is actually enough. This leads us to the following definition (where we just add in everything we have to):

Definition 4.6. Let Σ be a subformula-closed set of formulas, let $\mathfrak{M} = (W, R, v)$ be a model, and let

$$R_\Sigma([x], [y]) \iff \exists x' \in [x].\exists y' \in [y].R(x', y')$$

Then $\mathfrak{M}_\Sigma = (W_\Sigma, R_\Sigma, v_\Sigma)$ (here $v_\Sigma([x]) = v(x)$) is called A **Filtration** of $\mathfrak{M}$ through Σ.

Of course, this definition is building to the following, somewhat predictable theorem. After all, we chose our relation to make this theorem true!

Theorem 4.7. Let $\mathfrak{M} = (W, R, v)$ a model and $\mathfrak{M}_\Sigma = (W_\Sigma, R_\Sigma, v_\Sigma)$ a filtration of $\mathfrak{M}$ through Σ. Then

$$\forall \varphi \in \Sigma.(\mathfrak{M}, x) \vDash \varphi \iff (\mathfrak{M}_\Sigma, [x]_\Sigma) \vDash \varphi$$

Proof. It is tempting to consider bisimulation as a proof technique here, as it is the title of the chapter. However recall bisimulation shows that the *entire* theory of a world is preserved across the bisimulation B, whereas we are only interested in preserving *part* of the theory (namely that part specified by Σ). Because of this, we have no recourse besides an honest inductive proof. It is precisely for this induction that we specify Σ be suformula-closed: we need to know that smaller formulas are also in Σ in order to apply the inductive hypothesis. That said, let's get on with the proof:

We induct on formulas. If $p \in \Sigma$, then $p \in v(x) \iff p \in v_\Sigma(x)$ by definition.

If $\neg\varphi \in \Sigma$, then $\varphi \in \Sigma$ by subformula-closedness. Then by induction

$$(\mathfrak{M}, x) \models \varphi \iff (\mathfrak{M}_\Sigma, [x]) \models \varphi$$

thus we see

$$(\mathfrak{M}, x) \models \neg\varphi \iff (\mathfrak{M}_\Sigma, [x]) \models \neg\varphi$$

If $\varphi \wedge \psi \in \Sigma$, then $\varphi \in \Sigma$ and $\psi \in \Sigma$ too. Now by induction

$$(\mathfrak{M}, x) \models \varphi \iff (\mathfrak{M}_\Sigma, [x]) \models \varphi$$

and

$$(\mathfrak{M}, x) \models \psi \iff (\mathfrak{M}_\Sigma, [x]) \models \psi.$$

From this we see

$$(\mathfrak{M}, x) \models \varphi \wedge \psi \iff (\mathfrak{M}_\Sigma, [x]) \models \varphi \wedge \psi$$

Finally say $\Diamond\varphi \in \Sigma$, then $\varphi \in \Sigma$ too.

First, if $(\mathfrak{M}, x) \models \Diamond\varphi$, then some $y \in R(x)$ satisfies φ. But by the definition of R_Σ $[y] \in R_\Sigma([x])$. Further, by induction $[y] \models \varphi$. Then $\mathfrak{M}_\Sigma, [x] \models \Diamond\varphi$.

Conversely, if $\mathfrak{M}_\Sigma, [x] \models \Diamond\varphi$, we know there is some $[y] \in R_\Sigma([x])$ with $[y] \models \varphi$. Then by induction, $y \models \varphi$. But then some $y' \in R(x)$ *also* satisfies φ and so $\mathfrak{M}, x \models \Diamond\varphi$. ■

Ex. 4.7 —
Our definition of the relation on the filtered model is less general than it could be. There are other relations which we could also use. Indeed,

Definition 4.8. Say $\mathfrak{M} = (W, R, v)$ is a model and Σ is subformula-closed. A relation $R_\Sigma \subseteq W_\Sigma \times W_\Sigma$ is called **Σ-appropriate** if for every $x, y \in W$, the following hold:

- $R(x, y) \implies R_\Sigma([x], [y])$
- If $[x]R_\Sigma[y]$, then for all φ, if $\Box\varphi \in \Sigma$ and $x \models \Box\varphi$ then $y \models \varphi$.

a. Show that any $\Sigma-$appropriate relation will define a version of a filtration preserving everything we want to.

b. Show the defintion we gave is the *smallest* $\Sigma-$appropriate relation, that is, if S_Σ is $\Sigma-$appropriate, then $R_\Sigma \subseteq S_\Sigma$.

c. Define a new relation R_Σ^{big} by

$$[u]R_\Sigma^{\text{big}}[v] \iff \forall\Diamond\varphi \in \Sigma. v \models \varphi \implies u \models \Diamond\varphi$$

a) Show it is $\Sigma-$appropriate

b) Show it is the *largest* $\Sigma-$appropriate relation

d. What is the difference between the filtration using R_Σ and R_Σ^{big}?

Ex. 4.8 —
Show that if $x, y \in \mathfrak{M}$, then $x \sim_\Sigma y \iff T_\Sigma(x) = T_\Sigma(y)$ is an equivalence relation.

Ex. 4.9 —
Show that v_Σ is well defined on propositions $p \in \Sigma$.

Ex. 4.10 —
Define a relation R_Σ^{trans} by

$$[u]R_\Sigma^{\text{trans}}[v] \iff \forall\Box\varphi \in \Sigma. u \models \Box\varphi \implies v \models \Box\varphi \wedge \varphi$$

a. Show that whenever $\mathfrak{M}$ is a transitive model, using R_Σ^{trans} gives a transitive filtration.

b. Show that using R_Σ as above might *not* ensure $\mathfrak{M}_\Sigma$ is transitive.

Ex. 4.11 —

a. Show that if $\mathfrak{M}$ is symmetric, using R_Σ as above might result in a filtration which is *not* symmetric.

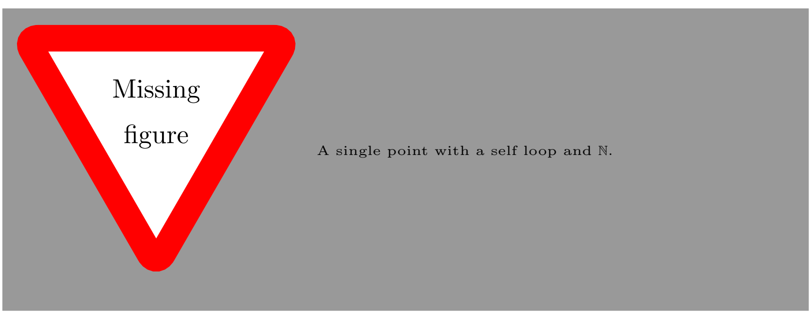
b. Find a relation R_Σ^{sym} such that, whenever we filter a symmetric model through Σ using this relation, results in a symmetric model.

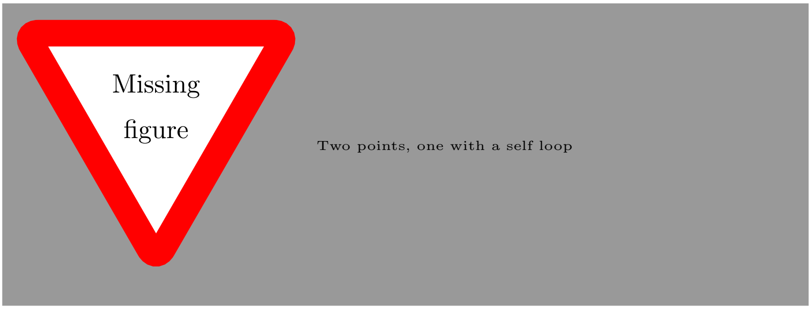
4.5 Unraveling

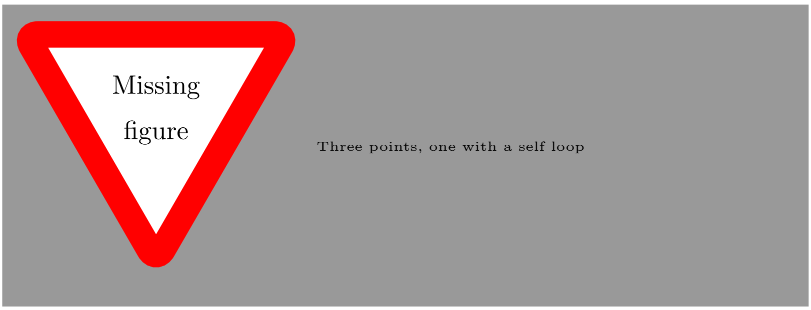
A lot of complexity in our frames comes from cycles. Partial orders are nice because we can always distinguish what could happen next from what has already happened. With cycles we allow ourselves to go back to where we've been. We will now outline a procedure for transforming (in a theory preserving way) any model into a model whose underlying frame has no cycles. The price we pay, however, is that a finite model may become infinite.

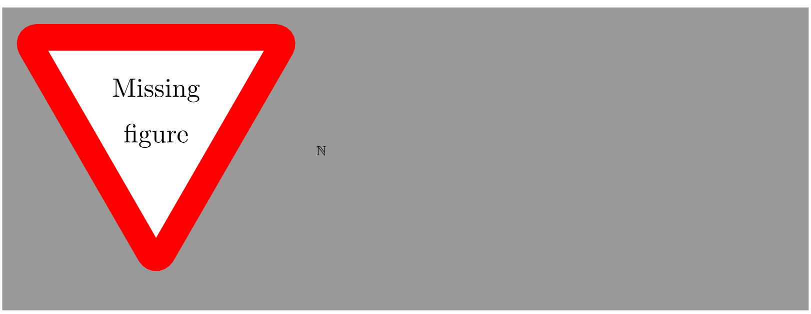
Definition 4.9. Let $w \in \mathfrak{M}$ such that there is an R-path from w to every other world in $\mathfrak{M}$. The **Unraveling** of a model $\mathfrak{M}$ about w is a model $\bar{\mathfrak{M}} = (\bar{W}, \bar{R}, \bar{v})$, where $\bar{W}$ is the set of R-paths starting at w, $\bar{p}\bar{R}\bar{q}$ iff $\bar{p} = (w, p_1, \ldots p_n)$, $\bar{q} = (w, q_1, \ldots q_{n+1})$, and $p_i = q_i$ where defined. and $\bar{v}(w, p_1, p_2, \ldots p_n) = v(p_n)$

The best way to explain this construction is with a set of pictures. It is not complicated, despite what the definition might lead you to believe:



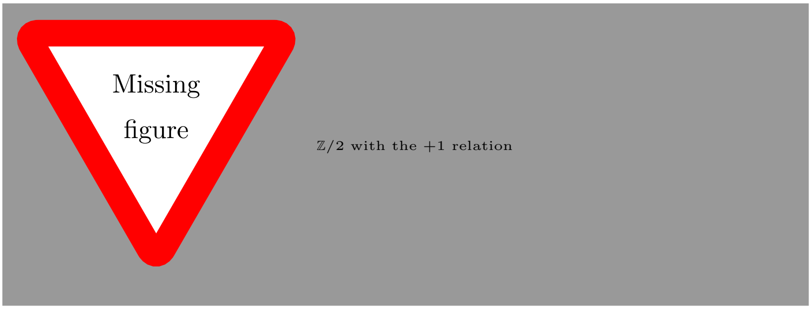






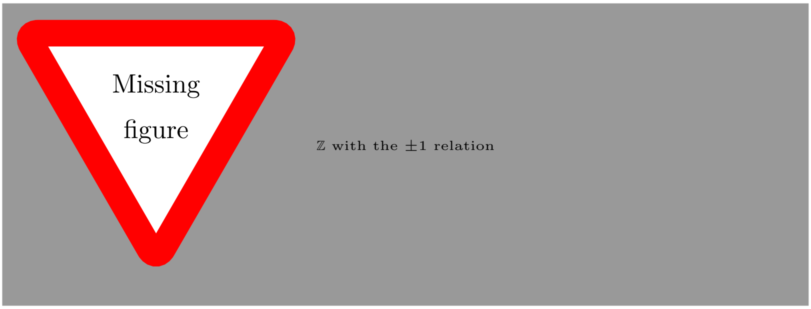
Here we started with one point, and we replaced its self loop with a new, equivalent point. Then we replaced the new point with a self loop, unwinding the loop until the self loop is lost to infinity.

Another example to keep in mind is the following:


Missing figure
$\mathbb{Z}/2$ with the $+1$ relation


Missing figure
uniwind one step


Missing figure
unwind another step



4.6 MultiAgent Logic

4.7 Proving Inexpressibility

We saw in chapter 3.6 how to show that certain classes of frames are *definable* by a sentence φ. But how can we know that a class of frames *isn't* definable? We need to argue against all possible sentences in our logic, a daunting task.

Say $\mathcal{C}$ is a class of models. We can compare all possible formulas at once by exhibiting a bisimulation between a model in $\mathcal{C}$ and a model outside of $\mathcal{C}$. Then we know that no formula can carve out $\mathcal{C}$, because no formula distinguishes between our two models, one of which is in $\mathcal{C}$ and one of which isn't!

As an example, consider the class $\mathit{2}$ of all models with exactly two worlds. Is there a formula which defines $\mathit{2}$? Towards a contradiction, say there were a formula φ such that $\mathfrak{M} \vDash \varphi \iff \mathfrak{M} \in \mathit{2}$. But then the model with two worlds from before (4.5) should satisfy φ. Unfortunately, since that model is bisimilar to a model with countably many worlds (and since, $|\mathbb{Z}|$ is, among other things, not equal to 2) we see that model satisfies φ too, even though it is not in $\mathit{2}$.

Similarly, one might want to know if we can define a predecessor operation in the basic modal language. Is there a formula φ_p such that $x \vDash \varphi_p \iff \exists y. x \in R(y)$ with $y \vDash p$? You might already have an idea of how to prove that no such φ_p exists:

Consider the generated submodel above (4.1). Notice that in the full model, $y \vDash p$ and $x \in R(y)$, so $x \vDash \varphi_p$. However, in the generated submodel, $x \not\models \varphi_p$. Thus there is no way to express φ_p in the basic modal language.

4.8 Finite Model Property

Possibly the most important use of filtration is proving the **Finite Model Property**. This property will allow us to show that K is decidable! What, though, does this mean?

We say a deduction system Λ is **Decidable** if there is a computer program which accepts φ as input, and returns True or False based on whether $\Lambda \vdash \varphi$.

As a rather naive example, we know that CPL is decidable. Let φ be a formula in CPL, and let n be the number of propositional constants in φ. Then there are 2^n possible valuations which are relevant for φ – each of the n variables can be assigned to either True or False. Then we can evaluate φ on each of these 2^n assignments and see if each valuation v makes φ true.

For more complicated logics, however, it is an interesting question to see if we can still decide whether a given formula is valid or not. There has been a lot of dedicated research in recent history to characterize how powerful we can make a logic while still having decidable theory. Obviously we cannot go too far, since First Order Logic, which is what we use (with set theory) as the foundation of contemporary mathematics, is not decidable. If it were, mathematics would be a very boring subject indeed! K strikes a nice balance - it is more interesting that CPL, and yet is still simple enough to be decidable. In this section we will outline why.

Let $\mathfrak{C}$ denote the canonical model for K. Then we know that φ is valid (that is, it is true in every model) if and only if $\mathfrak{C} \models \varphi$. A priori, however, this is not useful - $\mathfrak{C}$ is uncountable, and so it is not possible to compute with it effectively. It is for this reason that we introduce the **Finite Model Property**:

Definition 4.10. We say a logic Λ has the **Finite Model Property** if whenever φ is refuatable, it is refutable in a finite model.

It is a theorem of computability theory[1] that every logic (with "simple" axioms) with the finite model property is decidable (cf. 4.13). We will not appeal to this theorem, and will instead show that K is decidable by hand (using the finite model property directly). This is advantageous since it also provides a more efficient algorithm: We know the general algorithm promised to us by computabiltiy theory will halt, but we cannot know when. With our construction we can bound the runtime of our algorithm in terms of the complexity of φ (cf. 4.14).

Theorem 4.11. *K has the finite model property, moreover if φ is not valid and has n-many subformulas then it is invalidated by a model with at most 2^n worlds.*

Proof. Let φ be a formula invalidated by some model. Then, in particular, $\mathfrak{C} \not\models \varphi$. Let Σ be the subformulas of φ. Then Σ is subformula closed, and so we can consider $\mathfrak{C}_\Sigma$, the filtration of $\mathfrak{C}$ through Σ. Since each world in $\mathfrak{C}_\Sigma$ is characterized by which formulas in Σ that it thinks is true (indeed, this is how we defined our equivalence relation), there are at most $2^{|\Sigma|}$ many worlds. Since $\mathfrak{C}_\Sigma$ and $\mathfrak{C}$ agree on any formula in Σ, $\mathfrak{C}_\Sigma$ is a model with at most $2^{|\Sigma|}$ many worlds which also invalidates φ. ■

This theorem tells us everything we need to know! Given a formula φ, how can we tell if φ is true in every world of every model?

Well if it *isn't*, then there is a model with at most N worlds which refutes it! Here $N = 2^{|\Sigma|}$, as in the above proof. So we can simply try every possible model with fewer than N worlds (there are only finitely many 4.14). If one of the worlds in one of these models invalidates φ, then we know it is invalid! However, if none of the worlds in any of these models invalidates φ, then we know φ must actually be valid!

The fact that checking validity of a formula φ can be reduced to checking only finitely many models (indeed, only cheking models up to a size we can compute in advance!) is somewhat magical. In the exercises you will have the opportunity to explore related results for yourself, and show that many modal logics have this property.

[1] sometimes called recursion theory

Ex. 4.12 —

Show T is also decidable by applying a filtration argument to the canonical model for T.

Ex. 4.13 —

(If you know some computability theory)

a. Show semidecidable and cosemidecidable implies decidable[2]
b. Show any logic with semidecidable axioms has semidecidable theorems
c. Show any logic with the finite model property has cosemidecidable theorems, that is, the set of nontheorems is semidecidable.
d. Conclude that any logic with semidecidable axioms and the finite model property is decidable.

Ex. 4.14 —

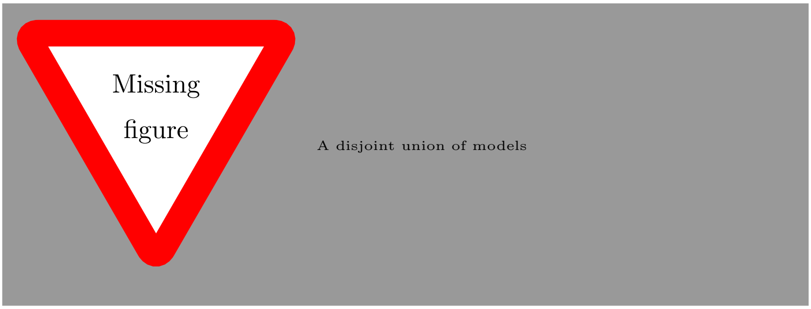
a. Find a formula for the number of models with at most N worlds
b. Implement the algorithm described above
c. Compute a (rough) big-O class for your implementation, in terms of the number of subformulas of φ.

Ex. 4.15 —

Say we have two models $\mathfrak{M}_1$ and $\mathfrak{M}_2$. We can construct their **Disjoint Union** ($\mathfrak{M}_1 \sqcup \mathfrak{M}_2$) as follows: $\mathfrak{M}_1 = (W_1, R_1, v_1)$ and $\mathfrak{M}_2 = (W_2, R_2, v_2)$, written $\mathfrak{M}_1 \sqcup \mathfrak{M}_2$, is $(W_1 \cup W_2, R_1 \cup R_2, v)$ where $v(w) = \begin{cases} v_1(w) & w \in W_1 \\ v_2(w) & w \in W_2 \end{cases}$

The unions in the above definition are assumed to be disjoint, that is $W_1 \cap W_2$ should be $\emptyset$. Notice we can always make this happen by renaming all of the worlds in W_2 if we need to.

[2] semidecidable is called recursively enumerable by some authors, and similarly decidable is called recursive in these circles.



a. Intuitively, $\mathfrak{M}_1$ and $\mathfrak{M}_2$ can't interact using modal formulas, since there is no edge from $\mathfrak{M}_1$ to $\mathfrak{M}_2$. Make this precise by proving there are two bisimulations, one from $\mathfrak{M}_1$ to $\mathfrak{M}_1 \sqcup \mathfrak{M}_2$ and one from $\mathfrak{M}_2$ to $\mathfrak{M}_1 \sqcup \mathfrak{M}_2$.

b. Construct models $\mathfrak{M}_1$ and $\mathfrak{M}_2$ such that $\mathfrak{M}_1$ is *not* a generated submodel of $\mathfrak{M}_1 \sqcup \mathfrak{M}_2$.

c. Show that for $w \in \mathfrak{M}$, $(\mathfrak{M} \sqcup \mathfrak{N})_w = (\mathfrak{M}_w)$

Ex. 4.16 —
Is the class of acyclic models definable?

Ex. 4.17 —
Every satisfiable formula of depth n is satisfiable on a tree of height $\leq n$.

Ex. 4.18 —
Is there a formula φ such that $x \models \varphi$ if and only if a p-world sees x?

Ex. 4.19 —
Let us say $B \subseteq W_1 \times W_2$ is a n-Bisimulation if we can simulate at least n moves (Say this better).

a. Show n-bisimulations preserve formulas of depth $\leq n$

b. If there is a n-bisimulation for every n, is there a real bisimulation?

Ex. 4.20 —
Show every model $\mathfrak{M} \models S5$ whose relation is complete (that is, $R = W \times W$) is bisimilar to a model where $w_1 \neq w_2 \iff Th(w_1) \neq Th(w_2)$.

Ex. 4.21 —
Show that every refutable formula can be refuted on a model satisfying
$x \neq y \implies Th(x) \neq Th(y)$

Part II

Extended Topics

Chapter 5

Topological Semantics

5.1 Introduction

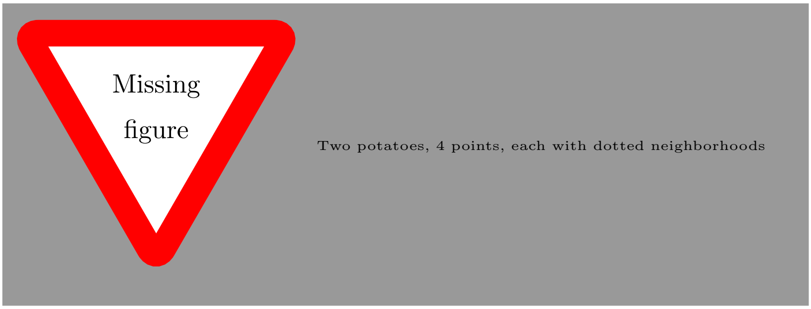
One of the fundamental principles of logic is the separation between the symbols we write, and their interpretation as true or false statements. One way we can show this explicitly in Modal Logic is to provide an entirely new way to interpret the symbols we write down. Instead of **Kripke Semantics**, which we have been using for the book thus far, we will introduce **Topological Semantics**.

Obviously this will require some background in topology, so let's start there.

5.2 Topology

Topology is a general way to describe geometric structure in mathematics. It is ubiquitous in the landscape of contemporary mathematics, and can be used creatively in many different areas. As a simple example of the "geometric structure" I'm referring to, consider the following two diagrams.

We can clearly see that x_0 is *close* to being a member of A_0 in a way that x_1 is *not* close to being a member of A_1. Similarly, y_0 is somehow *more* in A_0 than y_1 is in A_1. Unfortunately, in pure set theory, a point either is or is not in the set, there is no such notion of "closeness". We



will turn to **Topology** to give us this notion of closeness, which we will then use to interpret sentences in Modal Logic.

One way to explain why x_0 seems to be "closer" than x_1 to A_0 is by considering measurements which we are allowed to take. If $U \subseteq X$ is a subset of our space, then we can ask "is $x \in U$?" Of course, each of these measurements comes with some amount of ambiguity. We cannot tell where in U our point is, only that it *is* in U. Now we can see why this gives us a notion of closeness! If we limit ourselves to knowledge that these measurements can provide, then we *know* that $y_0 \in A_1$. We aren't sure exactly where it is, but no matter *where* it is, our measurement guarantees it is in A_1. x_1 is similar, there is a measurement we can take which guarantees it is *not* in A_0. x_0, however, is different. No matter what measurement we take, parts of that measurement will be in A and parts won't be. This corresponds to the idea that x_0 is "almost" in A_0. In the case of x_0, it happens to not be in A_0, while y_1 is in A_1, but barely. We formalize these ideas below:

Definition 5.1. A **Topology** on a set X is a set $\tau \subseteq 2^X$ of subsets of X satisfying the following 4 rules:

- $X \in \tau$
- $\emptyset \in \tau$
- If $U_i \in \tau$, then $\bigcup U_i \in \tau$
- If $U_1, \ldots, U_n \in \tau$, so is $U_1 \cap \ldots \cap U_n$

The sets $U \in \tau$ are called **Open**.

Notice the distinction between the last two clauses. We are allowed to union as many sets as we want, but we are only allowed to intersect finitely many!

Intuitively, what do these rules mean with respect to the "measurements" available to us? $X \in \tau$ and $\emptyset \in \tau$ say that we have access to useless measurements. Every x is in X, and no x is in $\emptyset$, so these measurements don't tell us anything. Of course we can always take a measurement which gives us no information!

If we think of an open set as one where we can know, for sure, that a point is inside it (because it is itself a measurement), it makes sense for a union of open sets to be open. After all, if x is in the union of open sets, then there must be an open set which contains it. But then the measurement that that open set corresponds to will say, definitively, that x is in the union! Taking intersections, though, is harder. We need to check that x is in each of the measurements, and so we restrict to finitely many.

Another important notion is that of a **closed** set. These are the complements of open sets. Where an open set is one that we can definitively say a point *is* inside it, a closed set is one where we can definitively say a point *isn't* inside it.

We will also define the **Interior** and **Closure** of a set A (written A° and $\overline{A}$) as the largest open set contained in A and the smallest closed set containing A. Equivalently:

$$A^\circ = \bigcup \{U \in \tau \mid U \subseteq A\}$$

$$\overline{A} = \bigcap \{U \mid U^c \in \tau \wedge A \subseteq U\}$$

The interior of A should be thought of as all of those points which are definitely in A. The closure, dually, should be thought of as all those points which are "close to A".

5.2.1 Examples

As the defining example of a topological space, we have to include $\mathbb{R}$ with its "usual topology". In $\mathbb{R}$ we already have a good idea of when two points are "close to" one another, and we can use this to create a topology on $\mathbb{R}$. All of topology is arguably based on trying to generalize this one example, so it is a very important motivator in the area.

Intuitively, x and y are close to each other if $|x - y|$ is *small.* And if $|x - y| < |z - w|$, then we say x and y are closer to each other than z and w are. So we might define an open set around x to be those y which are sufficiently close to x for some margin of error. Then

$$B_\varepsilon(x) = \{y \mid |x - y| < \epsilon\}$$

the **Open Ball of Radius** ε **centered at** x, should be open. We want this to be a topology, so we say an open set is any union of open balls. Formally, we set $U \in \tau$ if and only if U is a union of open balls.

Notice

$$B_\epsilon(x) = (x - \frac{\varepsilon}{2}, x + \frac{\varepsilon}{2})$$

and any open interval $(a, b) = \{x \mid a < x < b\} = B_{\frac{b-a}{2}}(\frac{a+b}{2})$ is an open ball.

Theorem 5.2. The usual topology on $\mathbb{R}$ is actually a topology.

Proof. X is open because $X = \bigcup_{x\in\mathbb{R}} B_1(x)$, and so is the union of open balls.

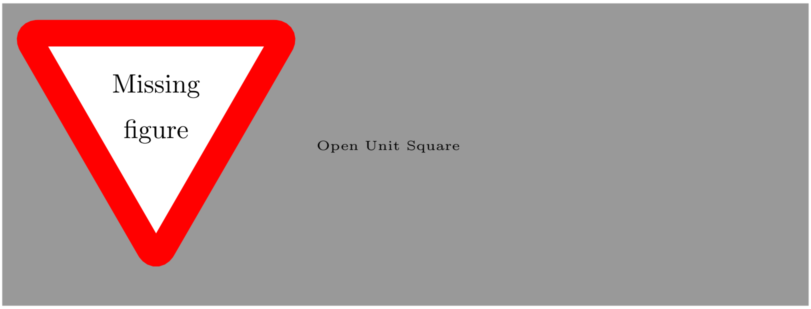
If U and V are both open, then why is $U \cap V$ open? Let $U = \bigcup_i U_i$ and $V = \bigcup_j V_j$ where each U_i and V_j is an open ball. Then

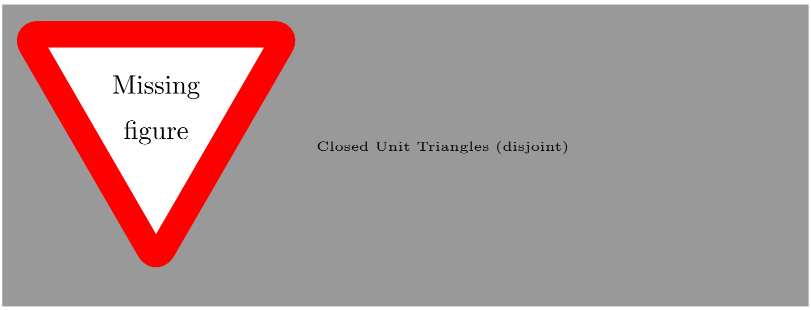
$$U \cap V = \bigcup_i U_i \cap \bigcup_j V_j = \bigcup_{i,j} U_i \cap V_j.$$

Since the intersection of open intervals is either empty or an open interval (cf. exercise 5.2) $U_i \cap V_j$ is either empty or an open interval. In this way $\bigcup_{i,j} U_i \cap V_j$ is the union of open intervals (plus possibly some empty sets that don't impact the union), and is itself open.

If U_i are open sets, then $\bigcup U_i$ is the union of unions of open intervals. But this is still a union of open intervals, and $\bigcup U_i$ is open. ∎

In addition to (a, b) being open, we also have (a, ∞) and $(-\infty, b)$ open. This is because $(a, \infty) = \bigcup_{a<b}(a, b)$ and $(-\infty, b) = \bigcup_{a<b}(a, b)$ are both unions of open sets. Also, the closed intervals $[a, b] = \{x \mid a \leq x \leq b\}$ are actually closed, since $[a, b] = ((-\infty, a) \cup (b, \infty))^c$. That is,





each $[a, b]$ is the complement of an open set. Finally, it is easily checked that $[a, b]^{\circ} = (a, b)$ and $\overline{(a, b)} = [a, b]$.

We can also work with "the usual topology" on $\mathbb{R}^2$. Now we take our open balls to be

$$B_\varepsilon((x_1, x_2)) = \{(y_1, y_2) \mid \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2} < \varepsilon\}.$$

Notice this is really saying the same thing as in $\mathbb{R}$ – $B_\varepsilon((x_1, x_2))$ is the set of those points less than ε away from (x_1, x_2).

Again, we call a set $U \subseteq \mathbb{R}^2$ open if it is the union of open balls. It is routine to verify that it is indeed a topology (cf. exercise 5.6)

In the figures above, we used dotted lines to indicate the region does not contain the line drawn, whereas a solid line indiates that the line is to be included in the set. Thus the first figure shows the set

$$\{(x, y) \mid 0 < x < 1 \text{ and } 0 < y < 1\},$$

which is open (since every point (x, y) in the square has an open ball which is completely contained in the square). Its closure is the square

$$\{(x, y) \mid 0 \leq x \leq 1 \text{ and } 0 \leq y \leq 1\}.$$

Notice the closed square is indeed closed, since its complement is open. The closed square is not open, however. This is because any open ball centered at a point on the boundary of the square has to contain points outside the edge of the square *must* contain points outside the square.

Similarly, the set consisting of the two triangles above is closed, since its complement is open. The interior of the above set is the same two triangles, but without their boundaries. As a more concrete example, the interior of the triangle

$$\{(x, y) \mid 0 \leq x, 0 \leq y, x + y \leq 1\}$$

is the set

$$\{(x, y) \mid 0 < x, 0 < y, x + y < 1\}.$$

There are more complicated examples, though. As one example, take $\mathbb{Q} \subseteq \mathbb{R}$. Then $\mathbb{Q}$ is not open (since any ball at a rational point must also contain irrationals) and is not closed (since any ball around an irrational point must contain rationals). Indeed, $\overline{\mathbb{Q}} = \mathbb{R}$ and $\mathbb{Q}^o = \emptyset$. Somewhat amazingly, this means the interior of $\overline{\mathbb{Q}}$ is $\mathbb{R}$ is not $\mathbb{Q}$. Similarly the closure of $\mathbb{Q}^o$ is $\emptyset$ is not $\mathbb{Q}$.

We can also explicitly give topologies on finite sets. For example, the set $\{a, b, c\}$ can be topologized with

$$\tau = \{\emptyset, \{c\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$$

Indeed there are 9 topologies on a 3 point space that are fundamentally different. A challenging exercise might be to try and find all of them! Somewhat interestingly, there is no known simple formula for $T(n)$, the number of unique topologies on a set of size n.

With this topology, we can quickly see what the open sets are. The complements of these will be closed sets, namely:

$$\{a, b, c\}, \{a, b\}, \{b\}, \{a\}, \emptyset$$

Now we can say, for instance, that the interior of $\{a, b\}$ is $\emptyset$, since it is the largest open set contained in $\{a, b\}$. Similarly, the closure of $\{a, c\}$ must be $\{a, b, c\}$ since it is the smallset closed set containing $\{a, b\}$.

Our last example is called 2^ω (pronounced "Cantor Space"), and is somewhat more exotic. First, 2^ω is the set of all of infinite binary strings, and can be identified with functions from $\mathbb{N} \to \{0, 1\}$[1]. Intuitively, two binary strings are "close to" each other if they look the same for a long time. To that end, we introduce (for each s a finite-length binary string) the basic open set $N_s = \{x \mid x$ has s as a prefix .

As an example, N_0 is the set of all strings starting with 0, N_{1101} is the set of all strings starting with 1101, and and every open set is the union of these basic opens.

These basic opens are special, though – unlike other topological space we've seen, these basic open sets are also closed! In the literature they are typically referred to as "basic clopen sets", and indeed N_0 is N_1^c. Similary, $N_{10} = (N_{00} \cup N_{01} \cup N_{11})^c$, and N_{1101} is the complement of the union of the 15 other basic clopens of length 4. Notice this does *not* mean every set is clopen! In exercise 5.13 you will give examples of a closed set that is not open, and an open set that is not closed.

Ex. 5.1 —

a. Show that the arbitrary intersection of closed sets is closed.

b. Show finite unions of closed sets are closed.

Ex. 5.2 —
Show that the intersection of two open intervals (a, b) and (c, d) is either empty or an open interval.

Ex. 5.3 —
Show that X and $\emptyset$ are both closed and open in any topology, such a set is called *clopen*.

Ex. 5.4 —
Show that $X^{\circ} = X$, $\overline{X} = X$, $\emptyset^{\circ} = \emptyset$, and $\overline{\emptyset} = \emptyset$.

[1] Indeed, 2^ω is just set-theorist lingo for all functions $\mathbb{N} \to \{0, 1\}$

Ex. 5.5 —
In the usual topology on $\mathbb{R}$, show that any two points $x, y \in \mathbb{R}$ can be separated by open sets. That is, there are U_x and U_y open with $x \in U_x$, $y \in U_y$, and $U_x \cap U_y = \emptyset$.

Ex. 5.6 —
We will show that the usual topology on $\mathbb{R}^2$ is indeed a topology.

a. Show that for any two open balls B_1 and B_2, there is an open ball $B_3 \subseteq B_1 \cap B_2$
b. Use the lemma from (a.) to conclude that the usual topology is a topology on $\mathbb{R}^2$.

Ex. 5.7 —
A question about the sorgenfried topology on $\mathbb{R}$?

Ex. 5.8 —
Show $\tau = \{\emptyset, \{x\}, \{x, y\}\}$ is a topology on $\{x, y\}$. This is called the *Sierpinski Topology.*

Ex. 5.9 —
For this exercise, we write Closure(A) and Interior(A) for $\overline{A}$ and A°. Additionally, we write Comp(A) for $X \setminus A$, the complent of A.

a. Prove Comp(Closure(Comp(A))) = Interior(A)
b. Prove Comp(Interior(Comp(A))) = Closure(A)

Ex. 5.10 —
Show each of the following:

a. $A^{\circ} \subseteq A$
b. $A \subseteq B \implies A^{\circ} \subseteq B^{\circ}$
c. $A^{\circ\circ} = A^{\circ}$
d. $A \cap B^{\circ} = A^{\circ} \cap B^{\circ}$

Ex. 5.11 —
Show each of the following:

a. $A \subseteq \overline{A}$
b. $A \subseteq B \implies \overline{A} \subseteq \overline{B}$

c. $\overline{\overline{A}} = \overline{A}$

d. $\overline{A \cup B} = \overline{A} \cup \overline{B}$

Ex. 5.12 —
Show that the elements x of the $\overline{A}$ are exactly the elements which are "close to" A. As above, this means that there is no measurement which confirms x is *not* in A. Formally, show $x \in \overline{A}$ if and only if every open set U containing x satisfies $U \cap A \neq \emptyset$.

Ex. 5.13 —

a. Show for any infinite string $s \in 2^\omega$, $\{s\}$ is closed but not open. (Hint: can you write it as an intersection of clopen sets? Can you write it as a union of clopen sets?)

b. Show for any $s \in 2^\omega$, $\{s\}^c$ is open but not closed.

Ex. 5.14 —

a. Show, for any set X, that $\tau = 2^X$ is a topology on X. This is called the **Discrete Topology**

b. Show, for any set X, that $\tau = \{\emptyset, X\}$ is a topology on X. This is called the **Indiscrete** or **Trivial Topology**

5.2.2 Continuous Maps

Not only are topological spaces important, but also the maps between them that preserve the topological structure. The relevant functions are called **continuous**, and are defined as follows:

Definition 5.3. If (X, τ_X) and (Y, τ_Y) are two topological spaces, then $f : X \to Y$ is called **Continuous** if for every $U \in \tau_Y$,

$$f^{-1}(U) \in \tau_X.$$

An equivalent (perhaps more intuitive definition) is the following

Theorem 5.4. A function $f : (X, \tau_X) \to (Y, \tau_Y)$ is continuous if and only if $f(\overline{A}) \subseteq \overline{f(A)}$

Proof. $\Longrightarrow$:
Since $f(A) \subseteq B \iff A \subseteq f^{-1}(B)$ for any sets A and B, it suffices to show $\overline{A} \subseteq f^{-1}(\overline{f(A)})$. But

$$A \subseteq f^{-1}(f(A))$$

so

$$\overline{A} \subseteq \overline{f^{-1}(f(A))}$$

(by exercise 5.11). Then

$$f(A) \subseteq \overline{f(A)}$$

so

$$f^{-1}(f(A)) \subseteq f^{-1}(\overline{f(A)})$$

and

$$\overline{f^{-1}(f(A))} \subseteq \overline{f^{-1}(\overline{f(A)})}$$

Finally, since f is continuous and $\overline{f(A)}$ is closed, $f^{-1}(\overline{f(A)})$ is closed too (as the diligent reader will prove in exercise 5.15). Then

$$\overline{f^{-1}(\overline{f(A)})} = f^{-1}(\overline{f(A)})$$

(since the closure of a closed set is itself).

Putting these together, we see

$$\overline{A} \subseteq \overline{f^{-1}(f(A))} \subseteq \overline{f^{-1}(\overline{f(A)})} = f^{-1}(\overline{f(A)}).$$

$\Longleftarrow$:
Again, by exercise 5.15, it suffices to show if $D \subseteq Y$ is closed, then $C = f^{-1}(D)$ is closed too.

Then

$$f(\overline{C}) \subseteq \overline{f(C)} = \overline{f(f^{-1}(D))} \subseteq \overline{D} = D$$

(using exercise 5.11 liberally, and recalling D is closed).

So $\overline{C} \subseteq f^{-1}(D) = C$, and thus (since $C \subseteq \overline{C}$) $C = \overline{C}$ and C is closed. ∎

Ex. 5.15 — Show that $f : X \to Y$ is continuous if and only if $f^{-1}(C)$ is closed (in X) for every C closed (in Y).

Ex. 5.16 —
Show the definition of continuity we gave aligns with the standard calculus definiton in $\mathbb{R}$ with the usual topology. That is $f : \mathbb{R} \to \mathbb{R}$ is continuous if and only if

$$\forall x_0.\forall \epsilon > 0.\exists \delta > 0.\forall x.(|x - x_0| < \delta \to |f(x) - f(x_0)| < \epsilon)$$

Ex. 5.17 —
Show the following functions are continuous

a. $f : \mathbb{R} \to \mathbb{R}$, $f(x) = x^2$ (here $\mathbb{R}$ has the usual topology)

b.

c.

d.

Ex. 5.18 —
Find an example of an open set $A \subseteq \mathbb{R}$ with the usual topology and a continuous function $f : \mathbb{R} \to \mathbb{R}$ such that $f(A)$ is *not* open.

Ex. 5.19 —
Show the only continuous functions from $\mathbb{R} \to 2^\omega$ are constant functions.

Ex. 5.20 —
This exercise refrences the definitions in exercise 5.14

a. Show every function out of a discrete space is continuous

b. Show every function into an indiscrete space is continuous

5.3 Topological Semantics

Now that we have a better understanding of topology, how can we do logic with it? We will now introduce a way of interpreting the symbols of the Basic Modal Language in terms of Topological Spaces instead of Kripke Structures.

Given a space X with topology τ, we additionally define a valuation function $v : \texttt{PROP} \to \mathbf{2}^X$ which tells us which points of X satisfy a given primitive proposition. Keep in mind $v(p)$ can be *any* set at all, not necessarily an open one.

Definition 5.5. A topological space (X, τ) equipped with a valuation function v as above is called a **Topological Model**

We start as we did with Kripke Semantics – by giving interpretations at a world. Let $x \in X$, with (X, τ, v) a topological model. Then we define $X, x \models \varphi$ recursively as follows:

$$
\begin{aligned}
X, x \models p &\iff x \in v(p) && \text{Primitive Propositions} \\
X, x \models \neg\varphi &\iff X, x \not\models \varphi && \text{Negation} \\
X, x \models \varphi \wedge \psi &\iff X, x \models \varphi \text{ and } X, x \models \psi && \text{And} \\
X, x \models \Box\varphi &\iff \exists U \in \tau . x \in U \text{ and } \forall u \in U . X, u \models \varphi && \text{Box}
\end{aligned}
$$

These semantics look exactly like Kripke Semantics with the exception of the interpretation of $\Box$. Intuitively this is because $\Box$ is the part of the Basic Modal Language that uses the extra structure. The semantics of $\Box$ are not entirely different, though. Topologically, we say that $x \models \Box\varphi$ if every world “close enough” to x models φ. This is similar to our familiar Kripke Frames, if we think about every world “close to” x as being those worlds only one step away.

Definition 5.6. As before, we say that φ is **Validated** by a model (X, τ, v) if every world has $X, x \vDash \varphi$. Similarly, a space (X, τ) validates φ if, for every valuation v, (X, τ, v) validates φ. Finally we say a class of spaces $\mathcal{C}$ validates φ if every space $X \in \mathcal{C}$ validates φ.

We write $(X, \tau, v) \vDash \varphi$, $(X, \tau) \vDash \varphi$, and $\mathcal{C} \vDash \varphi$ respectively for the three definitions of validation.

As before, we can directly reason about the semantics of our abbreviations by making use of the following theorem:

Theorem 5.7.

$$\mathfrak{M}, x \models \varphi \vee \psi \iff \mathfrak{M}, x \models \varphi \text{ or } \mathfrak{M}, x \models \psi$$

and

$$\mathfrak{M}, x \models \Diamond\varphi \iff \forall U \in \tau . x \in U \implies \exists y \in U . \mathfrak{M}, y \models \varphi$$

Proof. $\mathfrak{M}, x \models \varphi \vee \psi \iff \mathfrak{M}, x \models \neg(\neg\varphi \wedge \neg\psi)$, but this happens exactly when $\mathfrak{M}, x$ *doesn't* satisfy $\neg\varphi \wedge \neg\psi$. But then it must satisfy φ or ψ.

Similarly, $\mathfrak{M}, x \models \Diamond\varphi \iff \mathfrak{M}, x \models \neg\Box\neg\varphi$. Then for every open set around x, $\neg\varphi$ cannot be true. So every open set containing x also contains a y with $\mathfrak{M}, y \models \varphi$, as claimed. ■

As an example, consider $\mathbb{R}$ with its usual topology, and the valuation function $v(p) = \mathbb{Q}$, $v(q) = \{x \mid x > 0\}$, and $v(r) = [-2, 1]$.

Then $\frac{1}{2} \models r \wedge q \wedge p$, and $-\sqrt{2} \models r \wedge \neg p$. Similarly, $\pi \models q \wedge \neg p$ but $3 \models p \wedge q \wedge \neg r$.

More interestingly, we have $x \models \Diamond p$ for every x, and $0 \models \Diamond q \wedge \neg q$. We also see $0.999 \models \Box r$, and $-2 \models r \wedge \neg\Box r$. These are because every open ball around every real number contains a rational, every open set around 0 must contain an element of $\{x \mid x > 0\}$, the ball of radius 0.000001 centered at 0.999 is entirely contained inside $[-2, 1]$, but no ball of any radius centered at 2 can be contained in $[-2, 1]$.

If we again consider the finite topology on $\{a, b, c\}$ given by

$$\tau = \{\emptyset, \{c\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$$

with $v(p) = \{a\}$, $v(q) = \{b\}$ and $v(r) = \{a, c\}$, then

$a \models \Box r$, and $c \models \Box r$, but $b \models \neg r \wedge \neg\Box r$. Similarly, $b \models \Diamond r$ and $a \models \Diamond(p \wedge r)$.

We define the **Denotation** of a formula as before:

Definition 5.8. For some sentence φ, we (as usual) define its **Denotation** $[\![\varphi]\!] = \{x \in X \mid x \vDash \varphi\}$.

We now give a recursive definition of $[\![\varphi]\!]$:

$$\begin{aligned}
[\![p]\!] &= v(p) \\
[\![\neg\varphi]\!] &= [\![\varphi]\!]^c \\
[\![\varphi \wedge \psi]\!] &= [\![\varphi]\!] \cap [\![\psi]\!] \\
[\![\Box\varphi]\!] &= [\![\varphi]\!]^\circ
\end{aligned}$$

It is worth noting that our abbreviations have convenient interpretations too:

Theorem 5.9. $[\![\varphi \vee \psi]\!] = [\![\varphi]\!] \cup [\![\psi]\!]$ $[\![\varphi \rightarrow \psi]\!] = [\![\varphi]\!]^c \cup [\![\psi]\!]$ $[\![\Diamond\varphi]\!] = \overline{[\![\varphi]\!]}$

Proof. Exercise 5.26 ■

Ex. 5.21 —
Here is a model, here are some sentences, which worlds in the model satisfy the sentences?

Ex. 5.22 —
Here is a model, here are some sentences, what are $[\![\varphi]\!]$ for each sentence?

Ex. 5.23 —
Here is a model, here are some worlds, what are $Th(x)$ for each world?

Ex. 5.24 —
Directly show (using topological semantics) that the following sentences are valid

a.

b.

c.

Ex. 5.25 —
The following sentences *aren't* valid. Find a counterexample for each one

a.

b.

c.

Ex. 5.26 —
Prove theorem 5.9

Ex. 5.27 —
Show $\varphi \to \psi$ is valid if and only if $[\![\varphi]\!] \subseteq [\![\psi]\!]$.

5.4 Soundness

Because topological spaces have more structure than graphs, K is not a strong enough logic to provide a completeness result when we interpret the basic modal language in this setting. As an example, we know that $A^{\circ\circ} = A^{\circ}$ in a topological space. This tells us that $\Box\Box\varphi \leftrightarrow \Box\varphi$ should be valid on the class of all topological spaces. Of course, $\Box\varphi \to \Box\Box\varphi$ is not a theorem of K (3.15) and so our logic would not be complete. The correct logic is S4, which, as a reminder, is K augmented with the following axioms:

- $\Box\varphi \to \varphi$
- $\Box\varphi \to \Box\Box\varphi$

These can be interpreted, respectively as "If Valerie knows φ, then φ is true" and "If Valerie knows φ, then she *knows* she knows φ". That is, Valerie only knows true things, and she is aware of everything she knows.

Let's start with soundness – Though the diligent reader will have unwittingly done much of this proof already. We will first prove an incredibly useful lemma:

Theorem 5.10. $\varphi \to \psi$ is valid on a space (X, τ, v) if and only if $[\![\varphi]\!] \subseteq [\![\psi]\!]$.

Proof. $\varphi \to \psi$ is valid $\iff$ $[\![\varphi \to \psi]\!] = X$ $\iff$ $[\![\varphi]\!]^c \cup [\![\psi]\!] = X$ $\iff$ $[\![\varphi]\!] \subseteq [\![\psi]\!]$ ∎

Theorem 5.11. S4 is sound with respect to the class of all topological spaces

Proof. We begin by showing that each axiom is validated by the class of all topological spaces:

Any theorem of CPL is true at any world, for exactly the same reason as in Kripke Frames: the theorems of CPL do not refer to the topological structure, and therefore are true at each world individually.

For the modal axioms, S4 is axiom K, T, and 4, so it suffices to check that each of those is valid.

K ($\Box(\varphi \to \psi) \to \Box\varphi \to \Box\psi$):

Say $x \vDash \Box(\varphi \to \psi)$ and $x \vDash \Box\varphi$. Then there are open sets U and V containing x so that every $y \in U$ satisfies $\varphi \to \psi$ and every $y \in V$ satisfies φ.

Then $U \cap V$ is open, and every $y \in U \cap V$ satisfies $\varphi \to \psi$ and φ, thus $y \vDash \varphi$ too.

So $U \cap V$ witnesses the fact that $x \vDash \Box\varphi$.

T ($\Box\varphi \to \varphi$):

It suffices to show $[\![\varphi]\!]^{\mathrm{o}} \subseteq [\![\varphi]\!]$ by exercise 5.27. However this is true by exercise 5.10.

4 ($\Box\varphi \to \Box\Box\varphi$):

Again, it suffices to show $\varphi^{\mathrm{o}} \subseteq \varphi^{\mathrm{oo}}$. However this is also true by exercise 5.10.

As for the rules of inference:

Modus Ponens:

If $\varphi \to \psi$ and φ are both valid, then every world satisfies $\varphi \to \psi$ and φ. Then, since CPL works everywhere, ψ must be true at every world too.

Neccessitation:

If φ is valid, then $[\![\varphi]\!] = X$. But then $[\![\Box\varphi]\!] = [\![\varphi]\!]^{\mathrm{o}} = X^{\mathrm{o}} = X$ by exercise 5.4. ■

5.5 Completeness

We were able to prove soundness directly, using the topological semantics and a few helpful lemmas. This was not unlike the proof of soundness in the Kripke case, where we basically followed our nose to get the desired results. Completeness, as with Kripke frames, will be harder, but we will offload a lot of the work by adapting our Kripke completeness. It is *possible* to construct a canonical topological model, a single topological space which refutes every non-theorem of S4, however we will not take that approach here (though it will be available in exercise 5.36 for the extremely dedicated reader).

We know that every nontheorem of S4 can be refuted by a reflexive, transitive frame. In 4.4 we showed how to extend these results to finite frames by taking the reflexive transitive counterexample and filtering it to obtain a finite (and still reflexive, transitive) counterexample. Our technique here will be similar. By showing how to turn a reflexive transitive model into a topological model (in a truth preserving way), we will show that any Kripke counterexample can be turned into a topological counterexample. Thus showing the topological semantics are complete.

Theorem 5.12. Every reflexive, transitive Kripke Model (W, R, v) defines a Topological Model (X, τ, v') with the same theory.

Proof. Let (W, R) be a reflexive and transitive Kripke Frame. We call a subset $U \subseteq W$ an "up-set" if whenever $u \in U$ and uRv means $v \in U$ too.[2] We define a topology on W by declaring

$$\tau = \{U \subseteq W \mid U \text{ is an up-set}\}.$$

Notice τ is indeed a topology: W is an up-set, since it contains *everything* in particular everything bigger than any given element in w. If U and V are up-sets, then so is $U \cap V$. This is because, given $x \in U \cap V$ and xRy, $y \in U$ (since U is an up-set) and $y \in V$ (since V is an up-set). Then $y \in U \cap V$ and so $U \cap V$ is an up-set. Finally, given an arbitrary union $\bigcup U_i$ of up-sets U_i, if $x \in \bigcup U_i$, then $x \in U_i$

[2] This phrasing comes from partial orders, where we think of uRv as "u is less than v". Then an up-set is a set of elements which is "closed upwards". If u is in the set, then anything bigger than u is too.

for some particular i. If xRy, then $y \in U_i$ (since U_i is an up-set), and then $y \in \bigcup U_i$ too! So $\bigcup U_i$ is an up-set, and the set of up-sets is a topology on W!

We need one more fact before we move on, and that is $R(x)$ is open in this topology. It suffices to show $R(x)$ is an up-set, so let $y \in R(x)$ and yRz. Then, by transitivity xRz too. So $z \in R(x)$ and $R(x)$ is an up-set. Indeed, $R(x)$ is the *smallest* up-set containing x. If $x \in U$ and xRy, $y \in U$ too. So $R(x) \subseteq U$.

We now claim that a world $w \in W$ satisfies

$$(W, R, v), w \vDash \varphi \iff (W, \tau, v) \vDash \varphi$$

That is, worlds have the same theory if we interpret the semantics as kripke frames or topologically. We prove this, of course, by induction on the structure of formulas.

When p is primitive, then

$$(W, R, v), w \vDash p \iff w \in v(p) \iff (W, \tau, v) \vDash p.$$

Similarly,

$$\begin{aligned}(W, R, v), w \vDash \varphi \wedge \psi &\iff (W, R, v), w \vDash \varphi \text{ and } (W, R, v), w \vDash \psi \\ &\iff (W, \tau, v), w \vDash \varphi \text{ and } (W, \tau, v), w \vDash \psi \\ &\iff (W, \tau, v), w \vDash \varphi \wedge \psi\end{aligned}$$

and

$$\begin{aligned}(W, R, v), w \vDash \neg\varphi &\iff (W, R, v), w \nvDash \varphi \\ &\iff (W, \tau, v), w \nvDash \varphi \\ &\iff (W, \tau, v), w \vDash \neg\varphi\end{aligned}$$

In both of these cases, the middle $\iff$ comes from the inductive hypothesis.

Finally, we have

$$\begin{aligned}(W, R, v), w \vDash \Box\varphi &\iff \forall w' \in R(w).(W, R, v), w' \vDash \varphi \\ &\iff \forall w' \in R(w).(W, \tau, v), w' \vDash \varphi \\ &\iff (W, \tau, v), w \vDash \Box\varphi\end{aligned}$$

Here the middle $\iff$ is the inductive hypothesis, and the last $\iff$ is because any open set U witnessing $(W, \tau, v), w \vDash \Box\varphi$ must contain $R(w)$, so we might as well take $R(w)$ as the witness! ■

Corollary 5.13. S4 is complete with respect to the class of all topological models

Proof. It suffices to show that any nontheorem of S4 is refuted on some topological model. Let φ be a nontheorem of S4. By completeness of S4 with respect to Kripke Semantics of reflexive transitive graphs, fix a kripke model which refutes φ. Then by the above theorem, transform it into a topological model with the same theory. In particular, this topological model refutes φ, as desired. ■

Ex. 5.28 —
Show the class of topological spaces with a bonus property is sound wrt some new axioms

Ex. 5.29 —
We can discuss **definable properties** of a topological space, analogous to the definable classes of frames, by saying a class $\mathcal{C}$ of topological frames is **defined** by a set of formulas Σ whenever $(X, \tau) \in \mathcal{C} \iff (X, \tau) \vDash \Sigma$. Show the following properties are definable

a.

b.

c.

Ex. 5.30 —
Show S5 is sound with respect to topologies where every closed set is open. Such topologies are called "Locally indiscrete".

Ex. 5.31 —
Given a reflexive transitive frame $\mathfrak{F}$, call $F \subseteq \mathfrak{F}$ a *down-set* if and only if $x \in F$ and yRx implies $y \in F$. Show the down-sets are exactly the closed sets of the Alexandroff Topology associated to $\mathfrak{F}$.

Ex. 5.32 —

The topology we constructed for the completeness proof has the property that every x has a *smallest* open set containing it (namely $R(x)$). Such topologies are called **Alexandroff**.

a. Show a space is Alexandroff if and only if the intersection of any family of open sets is open (as opposed to just a finite intersection). This is the “standard” definition of Alexandroff Spaces.
b. Show S4 is complete with respect to the class of Alexandroff Spaces
c. Show we can run the construction in reverse. That is, show every Alexandroff space has an associated (reflexive, transitive) frame such that the theory of a world is the same regardless of if we use topological or kripke semantics.

Ex. 5.33 —
Show every finite topological space is Alexandroff

Ex. 5.34 —
Show S4 is complete with respect to finite topological spaces

Ex. 5.35 —
A topological space (X, τ) is called T_0 if for every two points $x, y \in X$ there is an open set U which contains exactly one of the two points.

a. Find a bijection between finite partial orders and finite T_0 spaces.
b. Show, moreover, that the order-preseriving maps betwen finite partial orders are in bijection with the continuous maps between the associated finite T_0 spaces.
   Explicitly, let $(X, \preceq_X), (Y, \preceq_Y)$ be finite partial orders and (TX, τ_X) and (TY, τ_Y) be the associated finite T_0 spaces. Show there is a bijection between functions $f : X \to Y$ satisfiying $x \preceq_X x' \implies f(x) \preceq_Y f(x')$ and continuous functions from TX to TY.

Ex. 5.36 —
As with the proof of Kripke Completeness, we will define a topological space from our maximally consistent sets. Indeed, let X be the set of S4-maximally consistent sets of formulas, and topologize X by

declaring each $U_{\Box\varphi} = \{\Sigma \in X \mid \Box\varphi \in \Sigma\}$ open. Then τ is the set of arbitrary unions of these basic open sets.
Intuitively, this is because we *want* $\Box\varphi$ to denote an open set (since we interpret $\Box$ as the interior operator). Eventually we will want $\Sigma \vDash \varphi \iff \varphi \in \Sigma$, as before, so we will preemptively make $[\![\Box\varphi]\!]$ open by *making* them open.

a. Show that τ is indeed a topology.
b. As before, define the valuation function v by $v(p) = \{\Sigma \mid p \in \Sigma\}$. Show that $\Sigma \vDash \varphi \iff \varphi \in \Sigma$.
c. Finally, conclude $(X, \tau, v) \vDash \varphi \iff \mathrm{S4} \vdash \varphi$, and thus S4 is complete with respect to the class of topological models.

Ex. 5.37 —
Something to do with MultiAgent Epistemic Logics?

5.6 Dynamic Topological Logic

5.6.1 S4C

Since topological spaces have richer structure than kripke structures, it makes sense to try using more powerful logics with this structure. An important one is **Dynamic Topological Logic**, which gives us a way to model the truth of certain propositions changing over time.

The language of DTL is the same as the basic modal language, with an extra symbol $\bigcirc$. We interpret $\bigcirc$ using a function $f : X \to X$, and say that

$$x \vDash \bigcirc\varphi \iff f(x) \vDash \varphi.$$

One way to interpret this is as time evolution. Whenever f is applied to all of X, points move around. Perhaps our topological space represents the water in a river. If we think about f as sending a particular water molecule to where in the river it is after a second passes, then repeatedly applying f allows us to approximate how the river flows.

These kinds of spaces are called **Dynamical Systems** and are a source of active study in both pure and applied mathematics. One way of studying these spaes is by studying the sentences of DTL which our system validates. It is this point of view which we will take in this section.

Definition 5.14. A **Dynamic Topological System** is a tuple (X, τ, f) where (X, τ) is a topological space and $f : X \to X$ is a continuous function.

Moreover, a **Dynamic Topological Model** is a tuple (X, τ, f, v) where (X, τ, f) is a dynamic topological system, and v is a valuation function.

With our semantics in hand, we use a proof theory **S4C**. In addition to the axioms and rules of inference for S4, we add the following rules mediating the interactions between $\bigcirc$ and $\Box$.

$$\frac{}{\bigcirc(\varphi \to \psi) \to \bigcirc\varphi \to \bigcirc\psi}$$

$$\frac{}{\bigcirc\neg\varphi \leftrightarrow \neg\bigcirc\varphi}$$

$$\frac{}{\bigcirc\Box\varphi \to \Box\bigcirc\varphi}$$

$$\frac{\varphi}{\bigcirc\varphi}$$

This proof theory is sound and complete with respect to the class of all dynamic topological models, and has the finite model property, as we will see.

Adam - do you know any good proofs of completeness/fmp? All the ones I've found are somewhat elaborate

5.6.2 True DTL

Adam - do you know any fun results to put here? A quick glance at the literature seems to show that we don't know that much about DTL, but maybe I was looking at the wrong papers

This actually brings us to the cutting edge of research! The logic DTL which is studied right now augments S4C with one more modality, $*$, which plays the role of "Henceforth". We say $*\varphi$ is true if $\bigcirc\varphi$, and $\bigcirc\bigcirc\varphi$, and $\bigcirc\bigcirc\bigcirc\varphi$, and so on are all true. In terms of the denotation, this says that

$$[\![*\varphi]\!] = \bigcap f^{-n}[\![\varphi]\!]$$

While sound axiomatizations for this logic exist, there are no known complete axiomatizations. In fact, there is evidence suggesting there is *no* complete, finite axiomatization. If we're already losing nice properties like completeness, one might suspect other nice properties to fail too, and indeed this logic is not decidable.

href some papers

.

This makes a certain amount of sense, since the addition of $*$ allows us to express the asymptotic behavior of f, and dynamical systems can behave quite chaotically. As a simple example, the mandelbrot set and other fractal patterns are related to nothing but the iteration of continuous functions – it seems reasonable that asking for a complete description of the asymptotic behavior might be too much.

Chapter 6

Propositional Dynamic Logic

6.1 Intro

One of the most active areas of research in modal logic right now is in Programming Language Theory and Program Verification. It should go without saying that, for certain applications, writing code which provably does what we expect it to, and perhaps more importantly, provably *can't* do what we don't expect it to, is incredily important.

It is unfortunate that proving properties about code, especially in enormous code bases, is incredibly difficult. Wouldn't it be nice if we could have our code prove *itself* correct when we compile it? This is one of the dreams of PL Theory – to have code which, if it compiles, must be correct. While many of the systems that exist today place a large burden on the programmer to assist the compiler in checking the correctness of the code, a lot of people are working very hard to make these tools easier to use.

6.2 Syntax

PDL is an extension of Temporal Logic, where we have a modality for each program we could possibly run. We write $[\pi]$ and $\langle\pi\rangle$ for the box and diamond modalities of a program π, and we most commonly want to write expressions of the form $\varphi \to [\pi]\psi$. We can read this as "If φ holds, then after we execute π, ψ will hold. These are typically referred to as preconditions and postconditions, and tell us what our program should do. There are various flavors of PDL for various applications (often they are meant to mimic real-world programming practices), but in this chapter we will describe a rather basic one. As we will see, though, it is already expressive enough to prove the correctness of a lot of C-like code (albeit with some help from the programmer).

First, we need to talk about what a program is. For us, there will be only a few basic constructs. In the exercises, you will explore logics with more "primitive programs". Even a small imperative programming language has features for assigning values to variables, if/then statements, and while loops. We will work with a programming language which *only* has these features, to show the fundamentals of PDL. Our grammar will be more complicated now, and will be made of several parts. This is because our logic and our programs will refer to each other!

$$\alpha, \beta ::= \pi \mid \varphi? \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^*$$

$$\varphi, \psi ::= p \mid \top \mid \varphi \wedge \psi \mid \neg\varphi \mid [\alpha]\varphi$$

We have two grammars, one for **programs** and one for **PDL** which refer to each other. The PDL system should be fairly familiar, with primitive propositions p, and familiar ways to combine formulas. We now explicitly add a symbol for $\top$ which means "true". This will be convenient when expressing that we make no assumptions on our code. We also have a whole *family* of box-like modalities $[\alpha]$ – one for every possible program! These mean "after every possible execution of α, φ is true". Since our code is nondeterministic in general, there might be multiple states we could end at. Of course, we still have the tyipcal abbreviations for $\vee$, $\to$, and $\leftrightarrow$. Plus, we write $\langle\alpha\rangle\varphi$ for $\neg[\alpha]\neg\varphi$. We interpret $\langle\alpha\rangle\varphi$ as "after *at least one* execution of α, φ is true".

Our programs are made from primitive programs π. Much like the primitive propositions tell us what questions we are allowed to ask, the primitive programs tell us what we are allowed to *do* in our programming language. We also have $\varphi?$, which is a harsh if-statement based on formulas φ. Program execution stops if φ is false, and continus if φ is true. We also have $\alpha \cup \beta$, which randomly picks one of α or β to run. $\alpha;\beta$ sequences programs, by running α and then running β. α^* runs α repeatedly some number of times, and is our basic looping construction.

In the interest of small examples, we will restrict the primitive programs and propositions fairly heavily. The propositions will only be allowed to compare two expressions of variables, where an expression is comprised of `+,*,-,/` and real-valued constants. The primitive programs are allowed to set variables to expressions involving existing variables. We will write this as $x := e$. In real world applications, one wants more programs and propositions, but we will keep it simple.

Let's write a few example programs to get the hang of these. This program sets x to 7, y to 3, and then (if $x + y > 9$) sets z to $x + y$.

$$x := 7\ ;\ y := 3\ ;\ (x + y > 9)?\ ;\ z := x + y$$

We can express "If T then α else β" as follows:

$$(T?\ ;\ \alpha) \cup (\neg T?\ ;\ \beta)$$

We can also express "While T α then β" with

$$(T?\ ;\ \alpha)^*\ ;\ \neg T?;\beta$$

Here's a program which computes $x!$ and puts it in y. We know that the starred section will loop until $x = 0$, because the program cannot do anything else until $?x = 0$ is true after the loop.

$$y := 1; (x > 0?\ ;\ y = y * x\ ;\ x = x - 1)^*\ ; x = 0?$$

Using this language, we can try to express the correctness of the factorial code above. We will abbreviate it as `fact` in the interest of conciseness.

$$(x > 0) \rightarrow [\texttt{fact}]y = x!$$

Ex. 6.1 —
Translate this code into PDL

Ex. 6.2 —
Translate this PDL into pseudocode

Ex. 6.3 —
Here is a version of PDL with arrays. Write code that does something.

6.3 Semantics

Now that we know what our logic is allowed to say, we need to know what it means when it says something. It is time to move on to semantics, and hopefully the interpretation of PDL should line up with our intuition for how programs execute. We still consider a set of possible worlds, where now a world represents the state the program is in. Using our same primitive programs and propositions, this will mean worlds keep track of values for each of the variables.

In general, we think of an interpretation of a primitive program π as being a relation $R_\pi \subseteq W \times W$. Just as we can interpret a primitive proposition p as a subset $v(p)$ of worlds. From these two base cases, we inductively define relations for every program α, as well as semantics for the truth of every formula φ.

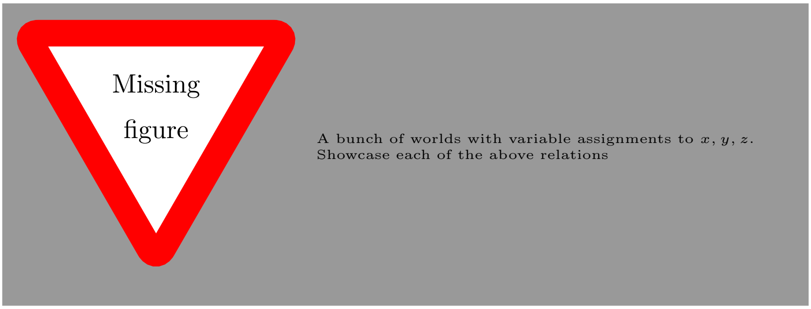
We interpret $\wedge$ and $\neg$ as we usually do, and $[\alpha]\varphi$ will, as usual, ask if every world related to the current world by R_α (which we will define shortly) satisfies φ. We will change our language somewhat for the current application, and we will say "after executing α" to mean the worlds reachable by R_α.

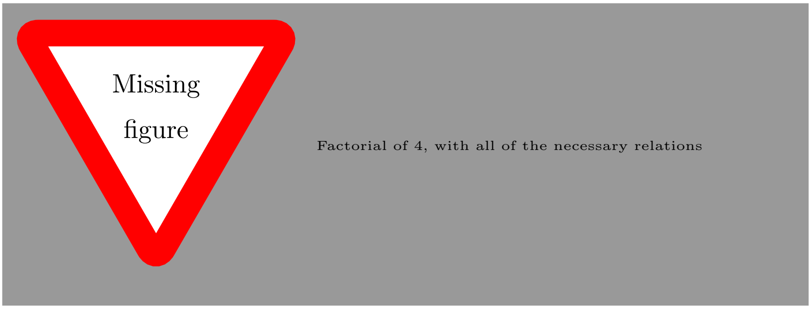
We define $R_{\varphi?}$ to be $\{(w, w) \mid w \models \varphi\}$. That is, $R_{\varphi?}$ only has self loops, and even then only has self loops at the worlds satisfying φ.

We define $R_{\alpha\cup\beta}$ is just $R_\alpha \cup R_\beta$, since we can execute either function nondeterministically.

Since $\alpha;\beta$ comes from running α, then running β, we will say $wR_{\alpha;\beta}w'$ if and only if $wR_\alpha w'' R_\beta w'$ for some w''. That is, if we can get from w to w' by first executing α, and then executing β.

Finally $R_{\alpha^*} = R_\alpha^*$, the reflexive transitive closure of R_α. The following example includes all of these, and afterwards we will see an example of the factorial function from above executing.





These semantics are designed to reflect the way that programs actually execute, and by reasoning formally about this system, we can show that our programs satisfy their specification. Of course, this model just abstracts program states. Reasoning about these semantics has not made our life any easier – indeed many *many* programmers get by by pretending programs are just variable assignments (maybe with some other assignments and state too) and basically tracing code executions in their head. The real power of PDL comes from the proof theory, which lets us say that certain programs will satisfy their specification without having to reason about the semantics at all.

Ex. 6.4 —
Here's a short program α, here's 4 worlds w. What is $R_\alpha(w)$ in each case?

Ex. 6.5 —
Same as above, for a different α.

Ex. 6.6 —
What should $[\![\varphi]\!]$ be in PDL? Give a recursive construction like those seen in 2.21 and 5.8

Ex. 6.7 —
Show $[\varphi?]\varphi$ is valid on each model of PDL.

6.4 Proof Theory

The time has come to introduce a proof theory for PDL. In chapter 2, we mentioned that proof theory, while tedious, was one of the most powerful parts of logic, and this is where we finally see it pay dividends. Reasoning about possible states of program execution and relations on this set of worlds, we can instead mindlessly push symbols around – a task well suited for computers. Without further ado, we take the following (plus all of CPL) as axioms:

$$\frac{}{[\alpha](\varphi \to \psi) \to [\alpha]\varphi \to [\alpha]\psi}$$

$$\frac{}{[\varphi?]\psi \leftrightarrow (\varphi \to \psi)}$$

$$\frac{}{[\alpha\ ;\ \beta]\varphi \leftrightarrow [\alpha][\beta]\varphi}$$

$$\frac{}{[\alpha \cup \beta]\varphi \leftrightarrow [\alpha]\varphi \wedge [\beta]\varphi}$$

$$\frac{}{[\alpha^*]\varphi \leftrightarrow \varphi \wedge [\alpha][\alpha^*]\varphi}$$

$$\frac{}{[\alpha^*](\varphi \to [\alpha]\varphi) \to \varphi \to [\alpha^*]\varphi}$$

We also take the following rules of inference:

$$\frac{\varphi \to \psi \qquad \varphi}{\psi}$$

$$\frac{\varphi}{[\alpha]\varphi}$$

The first axiom says that every program functions like a $\Box$ modality. Such modalities are called *normal*, and they are without contest the most common modalities studied. The second is for $\varphi?$. Recall $[\varphi?]\psi$ is true vacuously if a world doesn't satisfy φ, since then the relation is empty. If φ is satisfied, then we want to know ψ is as well. That is exactly asking if $\varphi \to \psi$. The third says that x sees y after running $\alpha \,;\, \beta$ if and only if x sees y after running α, then after running β. The fourth says that the only way to guarantee a formula holds after running α or β is to guarantee it holds after running α and after running β individually. Since we don't know which will run, we have to check both. Then we say that $[\alpha^*]\varphi$ is true exactly when φ is true at the current world, and (if we execute α once) $[\alpha^*]\varphi$ stays true in the future. Finally, we say that we are "allowed to induct" to prove things about α^*. If everywhere I can reach by running α thinks that $\varphi \to [\alpha]\varphi$ is true, and I am in a φ world, then I must be in an $[\alpha]\varphi$ world. So I take take one step forward along R_α and end up in a φ world. But then this new world must also satisfy $\varphi \to [\alpha]\varphi$ and so after taking another step I find myself in a φ world. Proceeding inductively, we see that every world reachable by some number of α executions is a φ world. That is, our starting world was a $[\alpha^*]\varphi$ world. This axiom is typically called the "Loop Invariant" axiom, as it lets us reason about loop structures by asserting that a loop invariant holds. By a "loop invariant", we mean a formula φ that is true when we enter the loop, and which is preserved under each iteration of the loop. We shall see that these loop invariants are the main source of programmer overhead, as finding the correct loop invariant to prove a particular fragment of code correct can be extremely difficult.

6.4.1 Soundness and Completeness

At this point, I'm sure you have a pavlovian reaction to seeing an axiom system presented, and indeed we will prove that these axioms are sound and complete with respect to models of PDL. Because these proofs are routine, and almost exactly like the proofs from chapter 3, we will leave many of the repeated details to the reader in the interest of space.

Soundness is easy, as usual. We leave the proof as an exercise.

Theorem 6.1. The stated axioms are sound with respect to the class of all models of PDL

Proof. Exercise 6.8 ■

The more interesting theorem is completeness, which we will show now. We will use the same technique as before, by creating a canonical model with maximally consistent sets as worlds. However this time, the proof will not *quite* work, since (as we will see) R_α^* will not be the same as R_{α^*}. After modifying the model slightly to fix this defect, we will obtain a canonical model. We will emphasize the details that have changed, or which require extra care in the PDL setting. Most of the proof works exactly as in the completeness of K, though.

Theorem 6.2. The stated axioms are complete with respect to the class of all models of PDL

Proof.

I need to wave my hands WAY less vigorously in this proof. Unfortunately, I ran out of time :(

As before, we will construct a model $\mathfrak{C}_{\mathrm{PDL}}$ as follows:

- W is the set of maximally consistent sets of formulas.
- The relation for a program α is given by

$$R_\alpha = \{(\Sigma, \Delta) \mid \{\varphi \mid [\alpha]\varphi \in \Sigma\} \subseteq \Delta\}$$

- The valuation of some primitive proposition p is given by

$$v(p) = \{\Sigma \mid p \in \Sigma\}$$

A simple induction on formulas shows that

$$\mathfrak{C}_{\mathrm{PDL}}, \Sigma \models \varphi \iff \varphi \in \Sigma$$

There is only one, mild issue: Notice this model was defined in terms of *all* programs α, not just the primitive programs π. Unfortunately, it does not work to just define R_π for primitive programs and extend the definitions as we have done before. Because of this, we will need to verify manually that, for instance $R_{\alpha\cup\beta} = R_\alpha \cup R_\beta$. Thankfully, almost all of these work out.

What is tragic, however, is that R_{α^*} is **not** R_α^*. So this model is not *really* a model of PDL. Luckily for us, it is close enough to being a model of PDL that we can fix it on a formula by formula basis.

Let φ be a nontheorem of PDL. Then $\{\neg\varphi\}$ is consistent, and so we can grow it to a maximally consistent set Σ (cf. exercise 6.9). Now $\mathfrak{C}_{\text{PDL}}, \Sigma \not\models \varphi$, and so all we need to do is turn this into an honest-to-goodness model of PDL. We will define a levelled up notion of filtration to create such a model.

Call a set Γ of formulas **closed** if:

- Γ is subformula closed
- $[\varphi?]\psi \in \Gamma$ implies $\varphi \in \Gamma$
- $[\alpha;\beta]\varphi \in \Gamma$ implies $[\alpha][\beta]\varphi \in \Gamma$
- $[\alpha \cup \beta]\varphi \in \Gamma$ implies $[\alpha]\varphi, [\beta]\varphi \in \Gamma$
- $[\alpha^*]\varphi \in \Gamma$ implies $[\alpha][\alpha^*]\varphi$

Notice we can close any set of formulas by first closing it under subformulas, and then adding in all of the things which the later rules require. If our initial set of formulas was finite, we will even be left with a finite set Γ, since the complexity of the programs on the right of the implication is less than the complexity of the programs on the left (exepct for $[\alpha^*]$, but it is clear that $[\alpha][\alpha^*]\varphi$ will not require we add any additional programs for Γ to be closed).

This tells us that the smallest closed set of formulas Γ containing $\neg\varphi$ is finite and subformula closed. So we can filter through it in order to get a finite model, which we will now verify is a real model of PDL. Since filtrations preserve truth (of formulas in Γ), this will be a model refuting φ. Indeed, it will be a *finite* model, which will get us decidability for free!

Let Π_Γ be the set of all programs including atomic programs in members of Γ, all programs $\psi?$ for $\psi \in \Gamma$, and closed under ;, $\cup$, and *.

We define a new model $\mathfrak{M}_\Gamma$ as usual, with equivalence classes of worlds of $\mathfrak{C}_{\text{PDL}}$ based on truth of sentences in Γ and the obvious valuation function. We define R_π^Γ for $\pi \in \Pi_\Gamma$ to be as in section 4.4 for each R_π defined on $\mathfrak{C}_{\text{PDL}}$. We further define

$$R_{\varphi?}^\Gamma = \{([\Sigma], [\Sigma]) \mid \mathfrak{C}_{\text{PDL}}, \Sigma \models \varphi\}$$

and we inductively define R^Γ_α for the extensions by ;, $\cup$, and * as usual.

Finally, we claim $\mathfrak{M}_\Gamma, [\Sigma] \models \varphi$ whenever $\varphi \in \Gamma$ and $\mathfrak{C}_{\mathrm{PDL}}, \Sigma \models \varphi$. We have to check that the filtered version of R_α is Γ-appropriate (cf. exercise 4.7) for each program α. To save the reader some page-flipping, it suffices to show

- $xR_\alpha y \implies [x]R^\Gamma_\alpha[y]$
- if $[x]R^\Gamma_\alpha[y]$, then for every $\Box\varphi \in \Gamma$, if $x \models \Box\varphi$ then $y \models \Box\varphi$

We will show this by an induction on α.

The case of primitive $\pi \in \Pi_\Gamma$ holds by definition.

If $\varphi? \in \Pi_\Gamma$, suppose $\Sigma R_{\varphi?}\Delta$. Then if $\psi \in \Sigma$, $\varphi \to \psi \in \Sigma$, so $[\varphi?]\psi \in \Sigma$. So $\psi \in \Delta$. Thus $\Sigma \subseteq \Delta$, and so $\Sigma = \Delta$ since Σ is a maximal. Moreover, since $[\varphi?]\varphi$ is valid (exercise 6.7), we have $\varphi \in \Delta = \Sigma$ and so the first property holds by definition of $R^\Gamma_{\varphi?}$.

For the second property, suppose $[\Sigma]R^\Gamma_{\varphi?}[\Delta]$. Then $[\Sigma] = [\Delta]$ and $\Sigma \models \varphi$. So if $[\varphi?]\psi \in \Gamma$ and $\Sigma \models [\varphi?]\psi$ we have $\Sigma \models \varphi \to \psi$, and so $\Sigma \models \psi$. But then $\Delta \models \psi$ too, since Σ and Δ have the same theory restricted to Γ. Thus $R^\Gamma_{\varphi?}$ is Γ-apropriate.

For the inductive cases, we will use the following idea: Given Σ, let θ_Σ be a formula satisfying

$$\theta_\Sigma \in \Delta \iff [\Sigma]R^\Gamma_\alpha[\Delta]$$

Add a lemma showing such a formula exists. cf. 114 in Goldblatt

Then to show $\Sigma R_\alpha \Delta$ implies $[\Sigma]R^\Gamma_\alpha[\Delta]$, it will suffice to show $[\alpha]\theta_\Sigma \in \Sigma$, as then $\theta_\Sigma \in \Delta$ and $[\Sigma]R^\Gamma_\alpha[\Delta]$, as needed.

For $\alpha;\beta \in \Pi_\Gamma$, inductively assume R^Γ_α and R^Γ_β are both Γ-appropriate. Let θ_Σ be such that $\theta_\Sigma \in \Delta \iff [\Sigma]R_{\alpha;\beta}[\Delta]$.

For the first condition, say $\Sigma R_\alpha \Psi R_\beta \Delta$. By induction, we have $[\Sigma]R^\Gamma_\alpha[\Psi]R^\Gamma_\beta[\Delta]$ and thus $[\Sigma]R^\Gamma_{\alpha;\beta}[\Delta]$ by definition. So $\theta_\Sigma \in \Delta$, and $[\alpha][\beta]\theta_\Sigma \in \Sigma$. Thus $[\alpha;\beta]\theta_\Sigma \in \Sigma$ and we see $[\Sigma]R^\Gamma_{\alpha;\beta}[\Delta]$ by the argument outlined above.

For the second condition, let $[\Sigma]R^\Gamma_{\alpha;\beta}[\Delta]$. Then for some $[\Psi]$, we have $[\Sigma]R^\Gamma_\alpha[\Psi]R^\Gamma_\beta[\Delta]$. If $[\alpha;\beta]\varphi \in \Gamma$ and $\Sigma \models [\alpha;\beta]\varphi$ we have $\Sigma \models [\alpha][\beta]\varphi$ and is also in Γ by one of the closure conditions. Then induction gives $[\beta]\varphi$ true at $[\Psi]$ and thus φ true at $[\Delta]$, as required.

For $\alpha \cup \beta \in \Pi_\Gamma$, we again take a formula

$$\theta_\Sigma \in \Delta \iff [\Sigma] R^\Gamma_{\alpha\cup\beta} [\Delta]$$

Then by induction and the definition of $R^\Gamma_{\alpha\cup\beta}$, we see $\theta_\Sigma \in \Delta$ whenever $\Sigma R_\alpha \Delta$ or $\Sigma R_\beta \Delta$. So $[\alpha]\theta_\Sigma$ and $[\beta]\theta_\Sigma$ are both in Σ, and the first condition is met.

I'm torn between leaving part 2 as an exercise like goldblatt, and actually doing it...

Finally, we work with α^*. The fact that $R_{\alpha^*} = R^*_\alpha$

Finish stealing this proof from Goldblatt 9.8

Thus the filtration is an honest-to-goodness model of PDL, and, importantly, still refutes $\neg\varphi$, as desired. ■

Corollary 6.3. PDL is decidable

Proof. In the proof of completeness we constructed a finite model refuting a nontheorem φ. Indeed by being more careful with our bookkeeping, we can bound the size of the Γ we filter through, and thus the size of the model which will refute it. Simply checking all models up to this size will give the desired decision procedure. ■

Ex. 6.8 —
Prove the soundness theorem for PDL

Ex. 6.9 —
Following the example in chapter 3, prove that every PDL-consistent set of formulas can be extended to a maximally consistent set of formulas.

Ex. 6.10 —Complete the proof of completeness by verifying the following:

a. $R_{\alpha;\beta} = R_\alpha \circ R_\beta$
b. $R_{\alpha\cup\beta} = R_\alpha \cup R_\beta$
c. $R_{\varphi?} = \{(\Sigma, \Sigma) \mid \Sigma \models \varphi\}$

Ex. 6.11 —
Show

$$\mathrm{PDL} \vdash [\alpha^n]\varphi \leftrightarrow [\alpha]^n\varphi$$

Where

$$\alpha^n = \underbrace{\alpha; \alpha; \ldots; \alpha}_{n \text{ times}}$$

and

$$[\alpha]^n\varphi = \underbrace{[\alpha][\alpha]\ldots[\alpha]}_{n \text{ times}}\varphi$$

Ex. 6.12 —
Show $\mathrm{PDL} \vdash [\alpha^*]\varphi \rightarrow [\alpha]^n\varphi$

add an index

actually set up the bibliography