9 Towards Modal Predicate Logic

9.1 Predicate logic recap

In these last two chapters, we are going to explore what happens when we add modal operators to the language of first-order predicate logic. I'll begin by reviewing the syntax and semantics of classical, non-modal predicate logic.

The language $\mathfrak{L}_P$ of first-order predicate logic consists of *predicates* $F^1, F^2, \ldots$, $G^1, G^2, \ldots$, *individual constants* (or *names*) $a, b, c, \ldots$, *individual variables* $x, y, z, \ldots$, the logical symbols $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\forall$, $\exists$, and the parentheses (and). Individual variables and constants are also called *(singular) terms*.

An *atomic sentence* is formed by conjoining a predicate with some terms. Each predicate takes a fixed number of terms, as indicated by its numerical superscript: F^1 is a *one-place* predicate that combines with one term to form a sentence, F^2 is *two-place*, and so on. In practice, we usually omit the superscripts, because context makes clear what kind of predicate is in play. For example, in the sentence $Fa \vee Fb$, it is clear that F is a one-place predicate.

If you remove all names from an English sentence that contains no logical operators, you get a predicate. From 'Bob is hungry', you get the predicate '– is hungry'; from 'Bob is in Rome', you get the two-place predicate '– is in –', and from 'Bob saw Carol's father in Jerusalem', you get the three-place-predicate '– saw –'s father in –'. When you translate from English, you normally translate English names into $\mathfrak{L}_P$-names, and English predicates into $\mathfrak{L}_P$-predicates. So 'Bob is in Rome' might become Fab, where a translates 'Bob', b 'Rome', and F '– is in –'.

From atomic sentences, other sentences are formed in the usual way by means of the truth-functional operators $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$.

Another way to construct a longer sentence from a shorter sentence is to add a quantifier in front of the shorter sentence. A *quantifier* is an expression of the form $\forall\chi$ or $\exists\chi$, where χ is some variable. A quantifier is said to *bind* the variable it contains: $\forall x$ binds x, $\exists y$ binds y, and so on.

English quantifiers are usually restricted to a particular subclass of the things under discussion: '*all whales* are mammals', '*some students* went home'. By contrast, $\forall x$ and $\exists x$ are unrestricted, corresponding roughly to 'everything is such that ...' and 'something is such that ...'. We translate restricted quantifiers by combining unrestricted quantifiers with truth-functional connectives. 'All whales are mammals' is equivalent to 'Everything is either not a whale or a mammal'; so it can be translated as $\forall x(Wx \to Mx)$. 'Some students went home' could be translated as $\exists x(Sx \wedge Hx)$.

Variables are book-keeping devices whose function resembles that of pronouns in English. Thus $\exists x(Sx \wedge Hx)$ might be read as 'something is such that *it* is a student and *it* went home'. By using different variables (x, y, z, ...), we can disambiguate statements with nested quantifiers. Consider

> Every dog barked at a tree.

This can mean that there is a particular tree at which all the dogs barked, but it can also mean that each dog found some tree to bark at – possibly different trees for different dogs. The first reading could be translated as

$$\forall x(Dx \to \exists y(Ty \wedge Bxy)),$$

the second as

$$\exists y(Ty \wedge \forall x(Dx \to Bxy)).$$

Some more terminology. Recall that the *scope* of an operator (token) in a sentence is the shortest well-formed subsentence in which it occurs. In $\exists y(Ty \wedge \forall x(Fx \to Bxy))$, the scope of the quantifier $\forall x$ is the subsentence $\forall x(Fx \to Bxy)$. If a token of a variable lies in the scope of a quantifier that binds the variable, it is called *bound*, otherwise it is *free*. In $\forall x(Fx \to Bxy)$, all occurrences of x are bound, but y is free.

A sentence containing free variable is called *open*. Sentences that aren't open are *closed*. Intuitively, only closed sentences make complete statements. For this reason, some authors reserve the word 'sentence' for closed sentences, referring to open sentences as 'formulas'. (Others call every $\mathfrak{L}_P$-sentence a 'formula'.)

Exercise 9.1

Translate the following sentences into $\mathfrak{L}_P$.

(a) Keren and Keziah are sisters of Jemima.
(b) All myriapods are oviparous.
(c) There's somebody at the door.
(d) Not every student likes logic.
(e) Every student who likes logic likes something.

Like sentences of modal propositional logic, sentences of predicate logic are interpreted relative to a model. A model of predicate logic first of all specifies an *individual domain* D over which the quantifiers are said to range. If we read $\forall x$ as 'everything is such that' and $\exists x$ as 'something is such that', the relevant "somethings" are the members of the domain D.

The remainder of a model is an *interpretation function* V that assigns

(a) to each name a member of D,
(b) to each one-place predicate a subset of D, and
(c) to each n-place predicate (for $n > 1$) a set of n-tuples from D.

An "n-tuple from D" is simply a list of length n, all elements of which are in D. Repetitions are allowed, so if Bob is a member of D, then $\langle$Bob, Bob$\rangle$ counts as a 2-tuple from D. (2-tuples are more commonly called *pairs*.) We can subsume condition (b) under condition (c) by assuming that a 1-tuple from D is a member of D.

Definition 9.1

A **(classical) first-order model** is a pair $\langle D, V \rangle$ consisting of

- a non-empty set D, and
- a function V that assigns to each name a member of D, and to each n-ary predicate a set of n-tuples from D.

As always, the purpose of a model is to represent a conceivable scenario together with an interpretation of the non-logical vocabulary. The non-logical vocabulary of $\mathfrak{L}_P$ are the names and predicates, which is why these are interpreted by V.

We assume that in any relevant scenario there some things we want to talk about; these things are represented by the domain. The members of D are often called *individuals*, but this should not be taken to imply anything about their nature. An individual might be a rock, a person, a symphony, a sentence, a number, or a possible world. Every $\mathfrak{L}_P$-name is assumed to pick out one of these individuals. (Different names can pick out the same individual, and there can be individuals that aren't picked out by any name.) Intuitively, a predicate expresses a property or relation that may be instantiated by the individuals in the domain. However, to determine the truth-value of a sentence like Fa or $\exists xFx$ in a given scenario, we only need to know which individuals have the property expressed by F. Similarly, to determine the truth-value of sentences like Rab or $\forall x\exists yRxy$, we only need to know which pairs of individuals stand in the relation expressed by R. That's why the interpretation function in a first-order model simply assigns sets of individuals or n-tuples of individuals to predicates. Fa is true in a given model iff the individual assigned to a (in the model) is a member of the set assigned to F; that is, iff $V(a) \in V(F)$. Likewise, Rab is true in a model iff the pair of individuals assigned to a and b – the pair $\langle V(a), V(b)\rangle$ – is in the set assigned to R.

In this way, the truth-value of every closed atomic sentences is determined. For truth-functionally complex sentences, the standard rules apply: a negated sentence $\neg A$ is true iff the corresponding sentence A is not true; $A \wedge B$ is true iff A and B are both true; and so on.

When we turn to quantified sentences, we face a problem. We can't define the truth-value of $\forall xFx$ in terms of the truth-value of Fx, because an open sentence like Fx doesn't have a truth-value. Interpretation functions only assign individuals to names; they say nothing about variables. Even if we changed this and said that x should also be interpreted as picking out a member of the domain, we would have to ignore this interpretation when we evaluate $\forall xFx$. We want $\forall xFx$ to be true iff Fx is true *no matter which individual is assigned to* x. We therefore define truth not just relative to a model, but relative to a model *and an assignment of individuals to variables*.

To illustrate, consider a model with just two individuals, Alice and Bob, which are picked out by the names a and b. Let $V(F)$ be the set { Alice }, a set that only contains Alice. So Fa is true and Fb false. The sentence Fx is neither true nor false, for the variable x does not refer to any particular individual. All we can say is that Fx is "true of" Alice and "false of" Bob. That is, Fx is true if we assign Alice to x

and false if we assign Bob to x. $\exists x Fx$ is true because there is an individual (Alice) of which Fx is true. Equivalently, $\exists x Fx$ is true because there is some assignment of individuals to variables relative to which Fx is true. $\forall x Fx$ is false because it is not the case that every assignment of individuals to variables renders Fx true.

So we'll define truth relative to a model $M = \langle D, V \rangle$ and a variable assignment g. A *variable assignment* is a function that maps variables to members of D. When we have nested quantifiers, as in $\forall x \exists y Gxy$, we need to consider variable assignments that differ from other assignments with respect to a particular variable. $\forall x \exists y Gxy$ is true iff, no matter what individual is assigned to x, there is some assignment of an individual to y (but holding fixed the assignment to x) that makes Gxy true. Equivalently: $\forall x \exists y Gxy$ is true iff for every variable assignment g, there is some variable assignment g' that differs from g at most in what it assigns to y, and Gxy is true relative to g'.

Let's say that a variable assignment g' is an χ-*variant* of a variable assignment g iff g' differs from g at most in the value it assigns to χ (where χ is any variable). Let's also introduce $[\tau]^{M,g}$ as shorthand for the individual picked out by a term τ in a model $M = \langle D, V \rangle$ relative to assignment g:

$$[\tau]^{M,g} =_{\text{def}} \begin{cases} V(\tau) & \text{if } \tau \text{ is a name} \\ g(\tau) & \text{if } \tau \text{ is a variable.} \end{cases}$$

Now we can state the standard semantics of first-order predicate logic. ('$M, g \models A$' is pronounced 'A is true in M relative to g').

Definition 9.2: Semantics of first-order predicate logic

If $M = \langle D, V \rangle$ is a first-order model, ϕ is an n-place predicate (for $n \geq 1$), $\tau_1, \tau_2 \ldots$ are terms, χ is a variable, and g is a variable assignment, then

(a)	$M, g \models \phi\tau_1 \dots \tau_n$	iff $\langle [\tau_1]^{M,g}, \dots, [\tau_n]^{M,g} \rangle \in V(\phi)$.
(b)	$M, g \models \neg A$	iff $M, g \not\models A$.
(c)	$M, g \models A \wedge B$	iff $M, g \models A$ and $M, g \models B$.
(d)	$M, g \models A \vee B$	iff $M, g \models A$ or $M, g \models B$.
(e)	$M, g \models A \rightarrow B$	iff $M, g \models B$ or $M, g \not\models A$.
(f)	$M, g \models A \leftrightarrow B$	iff $M, g \models (A \rightarrow B)$ and $M, g \models (B \rightarrow A)$.
(g)	$M, g \models \forall \chi A$	iff $M, g' \models A$ for all χ-variants g' of g.
(h)	$M, g \models \exists \chi A$	iff $M, g' \models A$ for some χ-variant g' of g.

Definition 9.2 settles the truth-value of every $\mathfrak{L}_P$-sentence in every (first-order) model, relative to any assignment function.

We can also define a concept of truth that is just relative to a model. Let's say that an $\mathfrak{L}_P$-sentence is *true in a model M* iff it is true in M relative to *every* assignment function g for M.

Finally, we say that an $\mathfrak{L}_P$-sentence is *valid* (in classical first-order logic) iff it is true in all (classical, first-order) models. Equivalently: An $\mathfrak{L}_P$ sentence is valid iff it is true in all models relative to all assignment functions.

On the present definition, $Fx \rightarrow Fx$ is valid, even though it does not make a complete statement, due to the free variable x. To avoid this, we could restrict the concept of validity to closed sentences.

Exercise 9.2

Define a first-order model in which $\forall x Fx \rightarrow \exists x Fx$ is false. Demonstrate that the sentence is false in your model by applying all relevant clauses from definition 9.2.

Exercise 9.3

The above definition of truth in a model uses the method of supervaluation that we met in section 7.4. Give examples to illustrate the following claims.

(a) If a sentence A is not true in a model, it does not follow that $\neg A$ is true in the model.

(b) A disjunction $A \vee B$ can be true in a model even though neither A nor B is true in the model.

9.2 Modal fragments of predicate logic

Much of the power and complexity of predicate logic comes from its ability to handle nested quantifiers with different variables. For some applications, these complexities aren't needed, and we can simplify the semantics.

Consider a fragment $\mathfrak{L}_P^1$ of $\mathfrak{L}_P$ with only one variable x, no names, and only one-place predicates. In $\mathfrak{L}_P^1$, we have sentences like Fx, $\forall xGx$, $\forall x\exists x(Fx \to Gx)$, but not Fa or $\forall x\exists y(Fx \to Gy)$.

Following definition 9.1, a model for $\mathfrak{L}_P^1$ consists of a non-empty set D and an interpretation function V that assigns to each predicate a subset of D. Intuitively, $V(F)$ specifies which members of D have the property expressed by F. We could equivalently assume that V assigns to each individual in D a truth-value: 1 if the individual has the relevant property, 0 if it doesn't. So we can rephrase definition 9.1 for $\mathfrak{L}_P^1$ as follows:

A **model of** $\mathfrak{L}_P^1$ is a pair $\langle D, V \rangle$ consisting of

- a non-empty set D, and
- a function V that assigns to each predicate of $\mathfrak{L}_P^1$ and each member of D a truth-value.

We can also simplify definition 9.2. Since $\mathfrak{L}_P^1$ has only one variable x, an assignment function for $\mathfrak{L}_P^1$ only needs to tell us which individual in D is picked out by x. So we can represent an entire assignment function for $\mathfrak{L}_P^1$ by a member of D. This leaves us with the following semantics.

If $M = \langle D, V \rangle$ is a model for $\mathfrak{L}^1_P$, d is a member of D, and ϕ is a predicate, then

(a) $M, d \models \phi x$ iff $V(\phi, d) = 1$.
(b) $M, d \models \neg A$ iff $M, d \not\models A$.
(c) $M, d \models A \wedge B$ iff $M, d \models A$ and $M, d \models B$.
(d) $M, d \models A \vee B$ iff $M, d \models A$ or $M, d \models B$.
(e) $M, d \models A \rightarrow B$ iff $M, d \models B$ or $M, d \not\models A$.
(f) $M, d \models A \leftrightarrow B$ iff $M, d \models (A \rightarrow B)$ and $M, d \models (B \rightarrow A)$.
(g) $M, d \models \forall x A$ iff $M, d' \models A$ for all $d' \in D$.
(h) $M, d \models \exists x A$ iff $M, d' \models A$ for some $d' \in D$.

These definitions look a lot like definitions 2.1 and 2.2 from chapter 2. The only difference is that the sentence letters from chapter 2 are now called predicates, the box is written $\forall x$, the diamond $\exists x$, and we always append the letter x to sentence letters: we write $\forall x F x$, not $\forall x F$. But it doesn't really matter in logic how a symbol is called or how it is written.

The upshot is that propositional modal logic, interpreted as in chapter 2, is a disguised *fragment of first-order predicate logic*. The sentence letters of $\mathfrak{L}_M$ are disguised (one-place) predicates, the box and the diamond are disguised quantifiers. If we adopted the orthographic convention to write the box as $\forall x$, the diamond as $\exists x$, and to always append the letter x to (capitalised) sentence letters, $\mathfrak{L}_M$ would look just like $\mathfrak{L}^1_P$, and it would have the same semantics.

If we use Kripke semantics to interpret $\mathfrak{L}_M$, we get a different fragment of first-order predicate logic. The box and the diamond are still disguised quantifiers, but this time they are restricted by the accessibility relation. We could drop the disguise by writing $\Box p$ as $\forall y(Rxy \rightarrow Py)$ and $\Diamond p$ as $\exists y(Rxy \wedge Py)$. The fragment of $\mathfrak{L}_P$ that corresponds to $\mathfrak{L}_M$-sentences has two variables x and y and one two-place predicate 'R' (in addition to the one-place predicates); it no longer has unrestricted quantifiers.

What's the point of the disguise? Why didn't we write boxes and diamonds as $\mathfrak{L}_P$-quantifiers all along? There are several answers.

One is that we often use the box and the diamond to formalize pre-theoretic concepts of which it is not obvious that they can be understood as a quantifiers over accessible worlds. Some hold that the correct semantics for obligation and permission,

for example, is not Kripke semantics, but neighbourhood semantics. The language of modal propositional logic is neutral on this disagreement. Or think of provability logic, where the box formalizes mathematical provability. As it turns out, one can give a Kripke semantics for provability, but nobody thinks this somehow reveals what provability really means. In provability logic, $\Box A$ means that A is derivable from the axioms and rules of (say) ZFC set theory; it would not be illuminating to write this as $\forall y(Rxy \to Ay)$.

One might also argue that the syntax of modal logic conveniently resembles the surface form of English statements we want to formalize. In 'Bob knows that it is raining', for example, the object of Bob's knowledge is specified by the sentence 'it is raining'. It therefore seems appropriate to formalize the sentence in terms of an operator K_b that applies to a sentence, p. If we "dropped the disguise", the formalization would be $\forall y(Rxy \to Py)$. The sentence 'it is raining' would have to be translated by a *predicate* P – a predicate that is true of all and only the worlds at which it is raining.

There is a deeper point here. Sentences of modal logic are interpreted *at a world* in a model. Modal logic looks at models "from the inside", from the perspective of a particular world. Predicate logic, by contrast, describes models "from the outside", from a God's eye perspective. If we want to say that a particular individual has property P in predicate logic, we need to pick out that individual among all the elements of the domain, perhaps by a name. We can then say Pa. In modal logic, we can simply say p to express that the internal point from which we're looking at the model has the relevant property.

For many applications, this internal perspective is very natural. When we think about what is possible or about what the future will bring, our thinking takes place at a particular time, in a particular world. We are looking at the structure of times and worlds from the inside. When I say that it is raining, I mean that it is raining *here and now*, in *this world*. I don't need a way to pick out the relevant time and place and world from a God's eye perspective. I can pick them out simply as the time and place and world at which I currently find myself.

There are other, more pragmatic reasons to use the modal language $\mathfrak{L}_M$ rather than $\mathfrak{L}_P$. The language of boxes and diamonds is simpler than the language of first-order predicate logic. It has a simpler syntax, a simpler semantics, and allows for simpler proofs. For almost all the conceptions of validity we have studied (K-validity, S4-validity, etc.), there are efficient mechanical procedures to determine whether

an arbitrary $\mathfrak{L}_M$-sentence is valid or invalid, By contrast, there is no mechanical procedure at all to determine, for an arbitrary $\mathfrak{L}_P$-sentence, whether it is valid or invalid.

You may wonder how this is possible given that $\mathfrak{L}_M$-sentence are just $\mathfrak{L}_P$-sentences in disguise. The reason is that while every $\mathfrak{L}_M$-sentence is a disguised $\mathfrak{L}_P$-sentence, not every $\mathfrak{L}_P$-sentence can be disguised as an $\mathfrak{L}_M$-sentence. There are many things one can say in $\mathfrak{L}_P$ that can't be said in $\mathfrak{L}_M$. Consider, for example, the claim that a model's accessiblity relation is irreflexive. This is easy to express in $\mathfrak{L}_P$: $\forall x \neg Rxx$. But it is impossible to express in $\mathfrak{L}_M$. There is no $\mathfrak{L}_M$-sentence that is true at a world in a model iff the model's accessibility relation is irreflexive.

That's why modal propositional logic, interpreted as in chapter 2 or 3, is a disguised *fragment* of predicate logic. It is a simple and computationally attractive fragment that takes an "internal" perspective on models.

Exercise 9.4

Explain why there is no $\mathfrak{L}_M$-sentence that is true at a world in a model iff the model's accessibility relation is irreflexive.

Exercise 9.5

Can you give another example of an $\mathfrak{L}_P$-sentence (about a Kripke model) for which there is no equivalent $\mathfrak{L}_M$-sentence?

9.3 Predicate logic proofs

With the help of definition 9.2, we could in principle work out whether various $\mathfrak{L}_P$-sentences are valid or invalid. But the process is tedious. Various proof systems for classical predicate logic provide an alternative.

Let's look at the tree method for classical predicate logic. I will begin with an example. Our target sentence is $\forall x(Fx \wedge Gx) \rightarrow \forall x Fx$. So we start the tree with its negation:

$$1. \quad \neg(\exists x(Fx \wedge Gx) \rightarrow \exists x Fx) \qquad \text{(Ass.)}$$

There is no world label because we're not doing modal logic. Next, we apply the standard rule for negated conditionals:

2.	$\exists x(Fx \wedge Gx)$	(1)
3.	$\neg\exists xFx$	(1)

Node 2 says that $Fx \wedge Gx$ is true of some individual. To expand this node, we introduce a new name a for that individual, and infer $Fa \wedge Ga$.

4.	$Fa \wedge Ga$	(2)
5.	Fa	(4)
6.	Ga	(4)

Nodes 5 and 6 expand the conjunction on node 4. Next, we expand node 3, which says that Fx is true of nothing. It follows that Fa must be false:

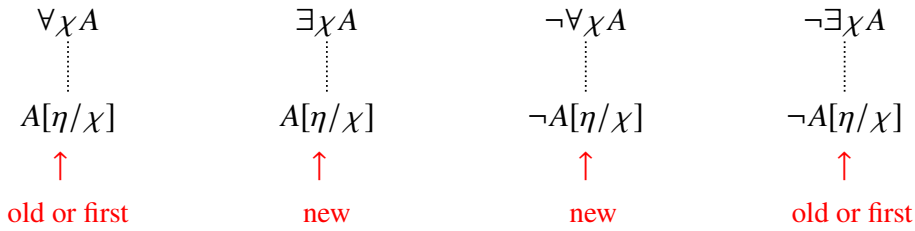
7.	$\neg Fa$	(3)
	x	

The tree is closed because the sentence on node 7 is the negation of the sentence on node 5.

To state the general rules, we need some more notation. If A is a sentence, χ is a variable, and η is a name, let $A[\eta/\chi]$ be the sentence obtained from A by replacing all free occurrences of χ with η. So $Fx[a/x]$ is Fa, but $\forall xFx[a/x]$ is $\forall xFx$ because this sentence contains no free occurrences of x.

The general rule for nodes of type $\exists\chi A$ is that their expansion adds a corresponding node of the form $A[\eta/\chi]$, where η is a "new" name that does not already occur on the relevant branch. If the resulting node has been added to every open branch below $\exists\chi A$, that node can be ticked off. $\forall\chi A$ nodes can be expanded several times, once for each "old" name. So if $\forall xA$ occurs on a branch, and the branch contains the names a and b, we can add both $A[a/x]$ and $A[b/x]$. If there is no old name on a branch, we are allowed to expand $\forall\chi A$ with a new name. $\forall\chi A$ nodes are never ticked off.

Here is a summary of the quantifier rules; 'old or first' means that the relevant name either already occurs on the branch or it is introduced as the first name on the branch.



Exercise 9.6

Give tree proofs for the following sentences.

(a) $\forall x Fx \to Fa$
(b) $\forall x(Fx \to Gx) \to (\forall x Fx \to \forall x Gx)$
(c) $\forall x(Fx \wedge Gx) \leftrightarrow (\forall x Fx \wedge \forall x Gx)$
(d) $\exists x \forall y Gxy \to \forall y \exists x Gxy$
(e) $\exists y \forall x(Fy \to Fx)$

There are also axiomatic calculi for predicate logic. I'll give one example.

To minimize the number of axioms, we treat $\exists\chi A$ as an abbreviation of $\neg\forall\chi\neg A$. We can then extend the axiomatic calculus for propositional logic by the following axiom schemas:

(**UI**) $\forall\chi A \to A[\eta/\chi]$

(**DI**) $\forall\chi(A \to B) \to (A \to \forall\chi B)$, if χ is not free in A

We also need one new rule, in addition to *Modus Ponens*:

(**Gen**) If A occurs on a proof, then $\forall\chi A$ may be appended

The resulting calculus is sound and complete: all and only the sentence that are valid in classical first-order logic can be derived. The above tree rules are also sound and complete.

Exercise 9.7

As in chapter 4, the completeness proof for trees shows that if a sentence is valid then any fully expanded tree for that sentence will close (provided the tree rules are applied in a certain order). Why doesn't this contradict the claim I made in the previous section: that there is no mechanical procedure to determine, for an arbitrary $\mathfrak{L}_P$-sentence, whether the sentence is valid? (Tree proofs count as "mechanical", so that's not the problem.)

9.4 Modality de dicto and de re

We are now ready to add boxes and diamonds to the language of first-order predicate logic. This gives us the **standard language of first-order modal logic**, or $\mathfrak{L}_{MP}$. The sentences of $\mathfrak{L}_{MP}$ are defined as follows.

1. An n-place predicate followed by n terms is an $\mathfrak{L}_{MP}$-sentence.
2. If A is an $\mathfrak{L}_{MP}$-sentence, then so are $\neg A$, $\Diamond A$, and $\Box A$.
3. If A and B are $\mathfrak{L}_{MP}$-sentences, then so are $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$ and $(A \leftrightarrow B)$.
4. If A is an $\mathfrak{L}_{MP}$-sentence and χ is a variable, then $\forall \chi A$ and $\exists \chi A$ are $\mathfrak{L}_{MP}$-sentence.
5. Nothing else is an $\mathfrak{L}_{MP}$-sentence.

We continue to interpret the box and the diamond as (disguised) quantifiers. So $\mathfrak{L}_{MP}$ effectively has two kinds of quantifiers: overt quantifiers of the form $\forall \chi$ and $\exists \chi$, and the disguised quantifiers $\Box$ and $\Diamond$. This is only useful if the two kinds of quantifiers range over different things. In applications of modal predicate logic, the box and the diamond usually range over possible worlds (or times), while the overt quantifiers range over things like people, rocks, ghosts, etc. which are assumed to inhabit the worlds (or times).

To illustrate, consider the following inference.

Bob knows that all humans are mortal.	$\mathsf{K}_b \forall x(Hx \rightarrow Mx)$
Socrates is human.	Hs
Therefore: Socrates is mortal.	Ms

The knowledge operator K_b is a quantifier over the worlds compatible with Bob's (implicit) knowledge. So $\mathsf{K}_b \forall x(Hx \rightarrow Mx)$ says that $\forall x(Hx \rightarrow Mx)$ is true at every world compatible with Bob's knowledge. $\forall x(Hx \rightarrow Mx)$ is assumed to quantify not over worlds, but over things that exist relative to a world. $\forall x(Hx \rightarrow Mx)$ is true at a world iff every inhabitant of the world is either not human or mortal. (The inference is valid because the accessibility relation for knowledge is reflexive.)

Another example. Imagine there is a lottery in which we know there is exactly one winning ticket. If we read the box as 'it is certain that' and W as '– is the winning ticket', we could use

$$\Box\exists xWx$$

to express that at every epistemically possible world there is a winning ticket. Compare the following statement:

$$\exists x\Box Wx.$$

This says that there is an individual such that at every epistemically accessible world, *it* is the winning ticket. This is only true if we know which ticket is the winning ticket. Suppose we know that ticket #7 is the winning ticket. Then there is an individual – ticket #7 – such that at every epistemically accessible world, this individual is the winning ticket. Worlds where a different ticket wins are inaccessible. But if we don't know which ticket will win then there are accessible worlds where ticket #1 wins, others where ticket #2 wins, and so on. So there is no individual of which we are certain that it is the winning ticket. In that case, $\exists x\Box Wx$ is false.

Sentences like $\exists x\Box Wx$ are called **de re**, which is Latin for 'of a thing'. Intuitively, $\exists x\Box Wx$ assert *of* the winning ticket that it has a modal property, namely the property of being the certain winner. By contrast, $\Box\exists xFx$, merely states that the proposition (Latin, *dictum*) $\exists xFx$ is certain. Sentences like this are called **de dicto**.

In general, an $\mathfrak{L}_{MP}$-sentence is *de re* if it contains a variable that is free in the scope of some modal operator. So to determine whether a sentence A is *de re*, first identify all subsentences of A that form the scope of a modal operator. In $\exists x\Box Wx$, there is one such subsentence: $\Box Wx$. Next, check if at least one of these subsentences contains a free variable. ($\Box Wx$ contains the free variable x.) If yes, the sentence A is *de re*.

If a sentence contains a modal operator and is not *de re*, then it is *de dicto*. So $\forall x(Fx \to \Box Gx)$ and $\exists y\Box(\forall xFx \to Fy)$ are *de re*, but $\Box\forall xFx \to Fa$ is *de dicto*. $\forall xFx \to Fa$ is neither *de dicto* nor *de re*, because it isn't modal.

There is, in fact, no consensus on how to classify sentences like $\Box Fa$ which contain a name, but no free variable, in the scope of a modal operator. One might argue that $\Box Fa$ is *de dicto* because it attributes a modal status – say, necessity – to the proposition Fa. But one might also interpret the sentence as attributing a modal property to the individual a: the property of being necessarily F. The sentence should then be classified as *de re*. Which of these two perspectives is more adequate depends on the precise semantics of $\mathfrak{L}_{MP}$. We therefore have to postpone the question until the next chapter, where we will consider some options for developing a semantics of $\mathfrak{L}_{MP}$.

Many natural-language sentences are ambiguous between a *de re* reading and a *de dicto* reading. Consider 'something necessarily exists'. This can mean either that there is an object which could not have failed to exist ($\exists x\Box Ex$); but it can also mean that it is necessary that something or other exists ($\Box\exists xEx$). The first reading is *de re*, the second *de dicto*.

Exercise 9.8

Translate the following sentences into modal predicate logic. (Some of them are ambiguous.)

(a) John must be hungry.
(b) Anyone who is a cyclist must have legs.
(c) Every day might be our last.
(d) If water is H_2O then cucumbers might contain H_2O.
(e) If anyone wants to leave early, they should do so quietly.
(f) Everyone who bought a ticket is allowed to enter.
(g) One day, all those who are rich will be poor.

Exercise 9.9

Which of your translations from the previous exercise are *de re* and which are *de dicto*?

On some interpretations of the modal operators, it is questionable whether *de re* sentences make any sense. Suppose we interpret the box as 'it is analytic that' or 'it is provable that'. The things that are analytic or provable are sentences or propositions. The sentence '$\forall x F x \to F a$', for example, is provable in classical predicate logic, and 'all vixens are female foxes' is analytic in English. (Remember that a sentence is analytic if it is true in virtue of its meaning.) It is not clear what it could mean to say that something is provable or analytic *of* a particular thing.

To illustrate the problem, let's introduce the name 'Julius' for whoever invented the zip. The sentence 'Julius invented the zip' is then analytic. (In fact, 'Julius invented the zip' entails that someone invented the zip, which is not analytic. So we should really use 'If anyone invented the zip, then Julius invented the zip'. I will ignore this complication.) But is it analytic *of* the person who invented the zip that they invented the zip? The problem is that this person has multiple names, and depending on which name we plug into the schema '— invented the zip', we sometimes get an analytic truth and sometimes not. For 'Julius', the sentence is analytic; for whatever name the inventor of the zip was given by his or her parents, the sentence is not analytic.

This kind of worry was prominently raised by W.V.O. Quine in the 1940s. It has since faded, mostly because philosophers have turned their attention away from analyticity to other interpretations of the box for which the problem is thought not to arise. But we will return to the matter in section 10.4.

9.5 Identity and descriptions

In applications of modal and non-modal predicate logic, it is often useful to have a special predicate for identity. So let's assume that (in $\mathfrak{L}_P$ and $\mathfrak{L}_{MP}$) we have a two-place predicate '='. This predicate is conventionally placed between its two arguments: we write '$a = b$', not '$=ab$'. We also write '$a \neq b$' instead of '$\neg(a = b)$'.

Unlike the other predicates of $\mathfrak{L}_P$ and $\mathfrak{L}_{MP}$, the identity predicate counts as a logical symbol, so its meaning is held fixed: in any model, $a = b$ means that the individual picked out by a is the very same thing as the individual picked out by b. This is reflected by the following clause, which we add to the semantics of predicate logic:

$$M, g \models \tau_1 = \tau_2 \quad \text{iff } [\tau_1]^{M,g} = [\tau_2]^{M,g}.$$

It is easy to see that the sentence $a = a$ is now valid, because a and a are guaranteed

to pick out the same individual. More interestingly, since the function of a name in classical predicate logic is just to pick out an individual, it never matters which of two names we use if they pick out the same individual. That is, if $a = b$ is true, then replacing some or all occurrences of a in a sentence with b never affects whether that sentence is true. This principle is known as **Leibniz' Law**.

In the tree method for (non-modal) predicate logic, the two principles I just mentioned are reflected by two new rules. First, if η is an "old" name (that already occurs on a branch) we can extend the branch by adding a node for $\eta = \eta$. Second, if an identity statement $\eta_1 = \eta_2$ occurs on a branch, and some sentence A on the branch contains η_1, then we may add a new node with the same sentence A except that one or more occurrences of η_1 in A are replaced by η_2, or one or more occurrences of η_2 by η_1. Let $A[\eta_2//\eta_1]$ stand for any sentence that results from A by replacing one or more occurrences of η_1 by η_2. The new rules can then be summarized as follows.

Existence	Leibniz' Law	Leibniz' Law
⋮	$\eta_1 = \eta_2$	$\eta_1 = \eta_2$
$\eta = \eta$	A	A
↑	⋮	⋮
old	$A[\eta_2//\eta_1]$	$A[\eta_1//\eta_2]$

Here is a tree for $(Raa \wedge a=b) \rightarrow Rab$, using Leibniz's Law.

1.	$\neg((Raa \wedge a=b) \rightarrow Rab)$	(Ass.)
2.	$Raa \wedge a=b$	(1)
3.	$\neg Rab$	(1)
4.	Raa	(2)
5.	$a=b$	(2)
6.	Rab	(4, 5, LL)
	x	

Exercise 9.10

Give tree proofs for the following sentences.

(a) $\forall x(x = x)$
(b) $\forall x \forall y(x = y \rightarrow y = x)$

Exercise 9.11

Show that the second version of the Leibniz' Law rule is redundant: we could reach $A[\eta_1//\eta_2]$ from $\eta_1 = \eta_2$ and A with the other rules.

In the axiomatic method, the two identity principles are often represented by the following axiom schemas:

(**E**) $\eta = \eta$
(**LL**) $\eta_1 = \eta_2 \rightarrow (A \rightarrow A[\eta_2//\eta_1])$

In modal extensions of predicate logic, both principles are controversial. For example, Leibniz' Law can arguably fail if a name occurs in the scope of a model operator. Consider the following inference:

> It is analytic that Julius invented the zip.
> Julius = Whitcomb L. Judson.
> Therefore: It is analytic that Whitcomb L. Judson invented the zip.

The conclusion clearly doesn't follow from the premises, but the inference seems to be licensed by Leibniz's law. Another well-known example:

> Lois Lane believes that Superman can fly.
> Superman = Clark Kent.
> Therefore: Lois Lane believes that Clark Kent can fly.

Again, we will return to this problem in section 10.4. In the remainder of the present section, I want to highlight some other things we can do with the identity predicate, apart from making claims about identity.

You have already encountered one other use in earlier chapters. Suppose we want to express that some relation R is strongly connected, meaning that for any two things,

either the first stands is R-related to the second or the second is R-related to the first. Without an identity predicate, this is impossible. With an identity predicate, it is easy:

$$\forall x \forall y (x \neq y \rightarrow (Rxy \vee Ryx)).$$

We can also use identity to express numerical quantifiers. For example, we can express 'there are at least two Fs' as

$$\exists x (Fx \wedge \exists y (Fy \wedge x \neq y)).$$

'There is exactly one F' can be expressed as

$$\exists x (Fx \wedge \forall y (Fy \rightarrow x = y)).$$

Exercise 9.12

Can you express the following in $\mathfrak{L}_P$ with identity?

(a) There are exactly two Fs.
(b) There are no more than three Fs.

Another important use of the identity predicate is to formalise statements involving definite descriptions. A definite description is a complex noun phrase, typically of the form 'the F', that purports to pick out a particular object. 'The current Prime Minister', 'the highest mountain in Scotland', and 'Bob's father' are definite descriptions.

The language of standard predicate logic does not have anything corresponding to the definite article 'the'. The only way to pick out an individual in $\mathfrak{L}_P$ is by a name. But there are good reasons not to translate descriptions as names.

For one, we would thereby miss logical connections between descriptions and predicates. 'The current Prime Minister is not Prime Minister' is a logical contradiction, but that can't be brought out if we translate 'the current Prime Minister' as a simple name.

Another reason not to translate descriptions as names is that descriptions often give rise to a *de re*/*de dicto* ambiguity. Consider the following sentence:

The Pope might have been Italian.

This has two readings. It can mean either that the actual Pope, Jorge Mario Bergoglio, might have been Italian (*de re*). Alternatively, it can mean that the following might have been the case: some Italian person is Pope (*de dicto*). There is no way to account for these two readings in $\mathfrak{L}_{MP}$ if we translate 'the Pope' as a name.

So how should we formalize statements involving definite descriptions? Russell (1905) argued that a statement of the form 'the F is G' is true just in case (i) there is exactly one (relevant) F, and (ii) this one F is also G. This we can easily express in the language of predicate logic if we have an identity predicate:

$$\exists x(Fx \wedge \forall y(Fy \rightarrow x = y) \wedge Gx).$$

Following Russell, we can translate 'The current Prime Minister is not Prime Minister' as

$$\exists x(Px \wedge \forall y(Py \rightarrow x = y) \wedge \neg Px),$$

which is indeed contradictory: it is true in no model. We can also account for the two readings of 'the Pope might have been Italian'. The *de re* reading is

$$\exists x(Px \wedge \forall y(Py \rightarrow x = y) \wedge \Diamond Ix).$$

The *de dicto* reading is

$$\Diamond\exists x(Px \wedge \forall y(Py \rightarrow x = y) \wedge Ix).$$

Exercise 9.13

Give two translations for each of the following sentences, one *de re* and one *de dicto*.

(a) Hilary Clinton might have been the 45th US President.
(b) Your dog could have eaten my breakfast.
(c) Alice believes that the student representative is rude.