

# 9 Towards Modal Predicate Logic

## 9.1 Predicate logic recap

In these last two chapters, we are going to add the resources of first-order predicate logic to those of propositional modal logic. Let's begin by reviewing the syntax and semantics of classical, non-modal predicate logic.

The language  $\mathcal{L}_P$  of first-order predicate logic consists of *predicates*  $F^0, F^1, F^2, \dots, G^0, G^1, G^2, \dots$ , *individual constants* (or *names*)  $a, b, c, \dots$ , *individual variables*  $x, y, z, \dots$ , the logical symbols  $\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \forall, \exists$ , and the parentheses ( and ). Individual variables and constants are also called (*singular*) *terms*.

Atomic sentences of  $\mathcal{L}_P$  are formed by conjoining a predicate with zero or more 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.  $Fa \vee Gab$ , for example, is well-formed only if  $F$  is one-place and  $G$  two-place.

In English, a predicate is what is what you get when you remove all names from a sentence. Removing ‘Bob’ from ‘Bob is hungry’ yields the predicate ‘– is hungry’. From ‘Bob is in Rome’, we get the two-place predicate ‘– is in –’. From ‘Bob saw Carol’s father in Jerusalem’, we could get the three-place-predicate ‘– saw –’s father in –’. When we translate from English, we normally translate English names into  $\mathcal{L}_P$ -names and (logically simple) English predicates into  $\mathcal{L}_P$ -predicates. ‘Bob is in Rome’ might become  $Fab$ , where  $a$  translates ‘Bob’,  $b$  ‘Rome’, and  $F$  ‘– is in –’.

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

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

In English, quantifier expressions are usually restricted to a particular subclass of the things under discussion: ‘*all whales* are mammals’, ‘*some students* went home’. The  $\mathcal{L}_P$ -quantifiers  $\forall x$  and  $\exists x$  are unrestricted. They roughly correspond to ‘everything is such that ...’ and ‘something is such that ...’. We can 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  $\forall x(Wx \rightarrow Mx)$ . ‘Some students went home’ could be translated as  $\exists x(Sx \wedge Hx)$ .

Variables are book-keeping devices. They function somewhat like pronouns in English.  $\exists x(Sx \wedge Hx)$  might be read as ‘something is such that *it* is a student and *it* went home’. By using different variables ( $x, y, z, \dots$ ), 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

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

the second as

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

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  $\exists y(Ty \wedge \forall x(Fx \rightarrow Bxy))$ , the scope of the quantifier  $\forall x$  is the subsentence  $\forall x(Fx \rightarrow Bxy)$ . If an occurrence of a variable lies in the scope of a quantifier that binds the variable, then the occurrence is called *bound*, otherwise it is *free*. In  $\forall x(Fx \rightarrow Bxy)$ , all occurrences of  $x$  are bound, but  $y$  is free.

A sentence containing free variables 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  $\mathcal{L}_P$ -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) Fred has a new car.
- (d) Not every student loves logic.
- (e) Every student who loves logic loves 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’ then 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 zero-place predicate a truth-value,
- (c) to each one-place predicate a subset of  $D$ , and
- (d) to each  $n$ -place predicate with  $n > 1$  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  $\langle \text{Bob}, \text{Bob} \rangle$  counts as a 2-tuple from  $D$ . (2-tuples are more commonly called *pairs*.) We can subsume condition (c) under condition (d) by assuming that a 1-tuple from  $D$  is a member of  $D$ . We can subsume (b) under (d) by identifying the truth-value False with the empty tuple  $\emptyset$  and the truth-value True with  $\{\emptyset\}$ . (Don’t worry if you find this confusing or objectionable. We won’t be using zero-ary predicates.)

**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  $n$ -place 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 are 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. In order to determine the truth-value of a sentence like  $Fa$  or  $\exists x Fx$  in a given scenario, however, we only need to know which individuals in the domain have the property expressed by  $F$ . Similarly, to determine the truth-value of sentences like  $Rab$  or  $\forall x \exists y Rxy$ , 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  $\forall x Fx$  in terms of the truth-value of  $Fx$ , because an open sentence like  $Fx$  doesn't have a truth-value. Interpretation functions interpret names and predicates; 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 if we evaluate  $\forall x Fx$ . We want  $\forall x Fx$  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  $b$  respectively. 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  $D$ . If 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  $y$  such that  $Gxy$  is true relative to  $g'$ .

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

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

This is a compact way of saying that (1) for any variable  $\chi$ ,  $[\chi]^{M,g}$  is the individual assigned to  $\chi$  by  $g$ , and (2) for any name  $\eta$ ,  $[\eta]^{M,g}$  is the individual assigned to  $\eta$  by the interpretation function of  $M$ .

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,  $\phi^n$  is an  $n$ -place predicate (for  $n \geq 0$ ),  $\tau_1, \dots, \tau_n$  are terms,  $\chi$  is a variable, and  $g$  is a variable assignment, then

- (a)  $M, g \models \phi^n \tau_1 \dots \tau_n$  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  $M, g \not\models A$  or  $M, g \models B$ .
- (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  $\chi$ -variants  $g'$  of  $g$ .
- (h)  $M, g \models \exists \chi A$  iff  $M, g' \models A$  for some  $\chi$ -variant  $g'$  of  $g$ .

Clause (a) says that, for example,  $Fa$  is true in a model  $M$  relative to an assignment  $g$  iff in that model, the predicate  $F$  applies to the individual picked out by  $a$ . Clauses (b)-(f) say that the truth-functional operators are interpreted in the standard fashion. Clauses (g) and (h) tell us how quantified sentences are interpreted.  $\exists x Fx$ , for example, is true relative to  $M$  and  $g$  iff  $Fx$  is true relative to some assignment function  $g'$  that differs from  $g$  at most in what it assigns to  $x$ .

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

We can also define a concept of truth relative to a model, without reference to an assignment function. Let's say that an  $\mathcal{L}_P$ -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  $\mathcal{L}_P$ -sentence is **valid** (in classical first-order logic) iff it is true in all (classical, first-order) models. Equivalently: An  $\mathcal{L}_P$  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, many authors restrict the concept of validity to closed sentences.

**Exercise 9.2**

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

**Exercise 9.3**

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

Following definition 9.1, a model for  $\mathcal{L}_P^1$  consists of a non-empty set  $D$  and an interpretation function  $V$  that assigns to each predicate a subset of  $D$ . That is, for  $\mathcal{L}_P^1$  definition 9.1 can be simplified as follows:

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

- a non-empty set  $D$ , and
- a function  $V$  that assigns to every  $\mathcal{L}_P^1$ -predicate a subset of  $D$ .

We can also simplify definition 9.2. Since  $\mathcal{L}_P^1$  has only one variable  $x$ , an assignment function for  $\mathcal{L}_P^1$  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  $\mathfrak{L}_P^1$ ,  $d$  is a member of  $D$ , and  $\phi$  is an  $\mathfrak{L}_P^1$ -predicate, then

- (a)  $M, d \models \phi x$  iff  $d \in V(\phi)$ .
- (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  $M, d \not\models A$  or  $M, d \models B$ .
- (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 and written in uppercase, the box is written  $\forall x$ , the diamond  $\exists x$ , and we always append the letter  $x$  to sentence letters: we write  $\forall x Fx$ , not  $\forall x F$ . But it doesn't really matter how a symbol is called or how it is written.

The upshot is that propositional modal logic, interpreted as in chapter 2, can be regarded as 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  $\mathfrak{L}_P^1$ , and it would have the same semantics.

If we use chapter 3's Kripke semantics rather than the simple semantics from chapter 2 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 now 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 reasons.

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 quantifiers over 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; it would not be illuminating to write this as  $\forall y(Rxy \rightarrow Ay)$ .

One might also argue that the syntax of modal logic conveniently resembles the surface form of English statements that we may want to formalize. In ‘Bob knows that it is raining’, for example, the object of Bob’s knowledge is specified by ‘it is raining’. It seems appropriate to formalize the sentence in terms of an operator  $K$  that applies to a sentence,  $p$ . If we “dropped the disguise”, the formalization would be  $\forall y(Rxy \rightarrow Py)$ . The sentence ‘it is raining’ would have to be translated by a predicate  $P$  – a predicate that applies to 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 a 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. If 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 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  $\mathcal{L}_M$  rather than  $\mathcal{L}_P$ . 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  $\mathcal{L}_M$ -sentence is valid or invalid. By contrast, there is no mechanical procedure at all to determine, for an arbitrary  $\mathcal{L}_P$ -sentence, whether it is valid or invalid.

You may wonder how this is possible given that  $\mathcal{L}_M$ -sentences are just  $\mathcal{L}_P$ -sentences in disguise. The reason is that while every  $\mathcal{L}_M$ -sentence is a disguised  $\mathcal{L}_P$ -sentence, not every  $\mathcal{L}_P$ -sentence can be disguised as an  $\mathcal{L}_M$ -sentence. There are many things one can say in  $\mathcal{L}_P$  that can't be said in  $\mathcal{L}_M$ . The  $\mathcal{L}_P$ -sentence  $\forall xRxx$ , for example, states that  $R$  is reflexive. No sentence of  $\mathcal{L}_M$  has this meaning: there is no  $\mathcal{L}_M$ -sentence that is true at a world in a model iff the model's accessibility relation is reflexive.

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

Since  $\Box A \rightarrow A$  corresponds to reflexivity, one might think that  $\Box p \rightarrow p$  is true at a world in a model iff the model's accessibility relation is reflexive. (a) Explain why this is not correct. (b) Can you also show that there is no  $\mathcal{L}_M$ -sentence that is true at a world in a model iff the model's accessibility is reflexive?

### 9.3 Predicate logic proofs

If we want to know whether a given  $\mathcal{L}_P$ -sentence is valid or invalid, we could in principle work through definition 9.2. Various proof systems for classical predicate logic offer a more streamlined approach.

Let's look at the tree method for classical predicate logic. Suppose we want to test whether  $\exists x(Fx \wedge Gx) \rightarrow \exists x Fx$  is valid. As always, we start the tree with the negation of the target sentence:

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

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. | $\neg \exists x Fx$       | (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) |
|----|----------------|-----|

We expand the conjunction on node 4.

- |    |      |     |
|----|------|-----|
| 5. | $Fa$ | (4) |
| 6. | $Ga$ | (4) |

Next, we expand node 3, which says that  $Fx$  is true of nothing. In particular then,  $Fx$  can't be true of  $a$ . So we add  $\neg Fa$ :

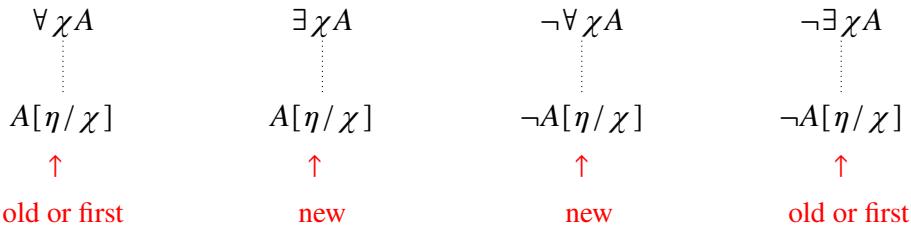
- |    |           |     |
|----|-----------|-----|
| 7. | $\neg Fa$ | (3) |
|    | $\times$  |     |

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

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

The general rule for expanding nodes of type  $\exists \chi A$  is that you add a node  $A[\eta/\chi]$ , where  $\eta$  is a “new” name that does not already occur on the relevant branch. If this node has been added to every open branch below  $\exists \chi A$  then the  $\exists \chi A$  node can be ticked off.  $\forall \chi A$  nodes can be expanded multiple times, once for each “old” name. So if  $\forall x A$  occurs on a branch, and the branch contains the names  $a$  and  $b$  then 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 9.5**

Give tree proofs for the following sentences.

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

There are also axiomatic calculi for predicate logic. We can, for example, use the following axiom schemas:

- ( $\forall\exists$ )  $\neg\exists\chi A \leftrightarrow \forall\chi\neg A$
- (UI)  $\forall\chi A \rightarrow A[\eta/\chi]$
- (DI)  $\forall\chi(A \rightarrow B) \rightarrow (A \rightarrow \forall\chi B)$ , if  $\chi$  is not free in  $A$

To these we would add the following rules. As in earlier chapters,  $\Gamma \vDash_P A$  means that  $A$  is a truth-functional consequence of (the sentences in)  $\Gamma$ .

- (CPL) If  $\Gamma \vDash_P A$  and all members of  $\Gamma$  are on a proof, then one may add  $A$ .
- (Gen) If  $A$  occurs on a proof, then one may add  $\forall\chi A[\eta/\chi]$ .

These axioms and rules are sound and complete: everything that can be proved is valid, and every valid (closed) sentence can be proved. The above tree rules are also sound and complete.

**Exercise 9.6**

The completeness proof for first-order trees (like the proof in chapter 4) 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 sensible 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  $\mathcal{L}_P$ -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  $\mathcal{L}_{MP}$ . The sentences of  $\mathcal{L}_{MP}$  are defined as follows.

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

We continue to interpret the box and the diamond as (disguised) quantifiers. So  $\mathcal{L}_{MP}$  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, in which I've written the box as 'K'.

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

The knowledge operator  $K$  is a quantifier over the worlds compatible with Bob's (implicit) knowledge.  $K \forall x(Hx \rightarrow Mx)$  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  $w$  iff  $Hx \rightarrow Mx$  is true of every inhabitant of  $w$ , meaning that every inhabitant of  $w$  is either not human or mortal. The inference is valid because the accessibility relation for knowledge is reflexive.

Imagine a lottery. Let's read the box as 'it is certain that' and  $W$  as '– is a winning ticket'. Can you see what is expressed by the following two statements?

- (1)  $\square \exists x Wx$
- (2)  $\exists x \square Wx$

(1) says that it is certain that some ticket wins: at every epistemically accessible world there is a winning ticket. (2) says that there is a particular ticket of which we are sure that it will win: there is an individual such that at every epistemically accessible world, *it* is the winning ticket. (2) is only true if we know which ticket is the (or a) winning ticket.

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

In general, an  $\mathcal{L}_{MP}$ -sentence is *de re* whenever it contains a variable that is free in the scope of some modal operator. To determine whether a sentence  $A$  is *de re*, first identify all subsentences of  $A$  that constitute the scope of a modal operator. (In  $\exists x \square Wx$ , there is one such subsentence:  $\square Wx$ .) Next, check if at least one of these subsentences contains a free variable. ( $\square Wx$  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  $\forall x(Fx \rightarrow \square Gx)$  and  $\exists y \square (\forall x Fx \rightarrow Fy)$  are *de re*, but  $\square \forall x Fx \rightarrow Fa$  is *de dicto*.  $\forall x Fx \rightarrow Fa$  is neither *de dicto* nor *de re*, because it isn't modal.

There is no consensus on how to classify sentences like  $\square Fa$  that contain a name, but no free variable, in the scope of a modal operator. One might argue that  $\square Fa$  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  $\mathcal{L}_{MP}$ . We therefore have to postpone the question until the next chapter, where we will consider some options for developing a semantics of  $\mathcal{L}_{MP}$ .

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 ( $\Box \exists x Ex$ ). The first reading is *de re*, the second *de dicto*.

### Exercise 9.7

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 anyone wants to leave early, they should do so quietly.
- (e) Everyone who bought a ticket is allowed to enter.

### Exercise 9.8

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

On some interpretations of the modal operators, one may question whether *de re* sentences are intelligible. 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. That  $2+2=4$ , for example, is provable in ZFC, 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 analytic. (In fact, ‘Julius invented the zip’ entails that someone invented the zip, which is not analytic. We should really use ‘If anyone invented the zip, then Julius invented the zip’. Let’s 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. Let’s assume that  $\mathcal{L}_P$  and  $\mathcal{L}_{MP}$  have the two-place predicate ‘=’. The identity predicate is conventionally placed between its two arguments: we write ‘ $a = b$ ’, not ‘ $= ab$ ’. We also sometimes abbreviate ‘ $\neg a = b$ ’ as ‘ $a \neq b$ ’.

Unlike the other predicates of  $\mathcal{L}_P$  and  $\mathcal{L}_{MP}$ , the identity predicate counts as a logical symbol. 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:

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

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**.

To reflect these facts, the tree method for (non-modal) predicate logic must be extended by two new rules. First, if  $\eta$  is an “old” name (that already occurs on a branch) then we can always add a node  $\eta = \eta$  to the branch. Second, if an identity statement  $\eta_1 = \eta_2$  occurs on a branch, and some sentence  $A$  on the branch contains

$\eta_1$ , then we may add a new node with the same sentence  $A$  except that one or more occurrences of  $\eta_1$  in  $A$  are replaced by  $\eta_2$ , or one or more occurrences of  $\eta_2$  by  $\eta_1$ . Let  $A[\eta_2//\eta_1]$  stand for any sentence that results from  $A$  by replacing one or more occurrences of  $\eta_1$  by  $\eta_2$ . The new rules can then be summarized as follows.

Self-Identity	Leibniz' Law	Leibniz' Law
$\vdots$ $\eta = \eta$ $\uparrow$ <b>old</b>	$\eta_1 = \eta_2$ $A$ $\vdots$ $A[\eta_2//\eta_1]$	$\eta_1 = \eta_2$ $A$ $\vdots$ $A[\eta_1//\eta_2]$

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 9.9

Use the tree method to check which of the following sentences are valid.

- (a)  $\forall x(x=x)$
- (b)  $\forall x \forall y(x=y \rightarrow y=x)$
- (c)  $(a=b \wedge b=c) \rightarrow a=c$
- (d)  $Rab \rightarrow \forall x(x=a \leftrightarrow Rxb)$
- (e)  $\forall x \forall y \forall z(x \neq y \wedge y \neq z \rightarrow x \neq z)$

**Exercise 9.10**

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 approach, the two facts about identity are often represented by the following axiom schemas:

- (SI)       $\eta = \eta$
- (LL)       $\eta_1 = \eta_2 \rightarrow (A \rightarrow A[\eta_2//\eta_1])$

Once we add boxes and diamonds to the language of predicate logic, the seemingly harmless axioms and rules for identity become problematic. 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.

**Exercise 9.11**

- (a) Give an axiomatic proof of  $\Box \exists x x = a$ , using (SI), (UI), (CPL), ( $\forall \exists$ ), (CPL), and (Nec), in this order. (b) Can you see why we might not want to count  $\Box \exists x x = a$  as a logical truth in some applications of modal logic? At which point do you think the proof goes wrong?

We will return to these issues 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 connected, meaning that for any two things, either

the first is  $R$ -related to the second or the second is  $R$ -related to the first. This can't be expressed without an identity predicate. With an identity predicate, it is easy:

$$\forall x \forall y (Rxy \vee x = y \vee Ryx).$$

We can also use identity to express numerical quantifiers. For example, we can express 'there are at least two  $F$ s' 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  $\mathcal{L}_P$  with identity?

- (a) There are exactly two  $F$ s.
- (b) There are no more than three  $F$ s.

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 'Carol's father' are definite descriptions.

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

One reason is that we would thereby miss logical connections between descriptions and predicates. 'The current Prime Minister is not Prime Minister' is a logical contradiction, but this 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  $\mathcal{L}_{MP}$  if we translate ‘the Pope’ as a name.

A better translation for statements involving definite descriptions was proposed by Bertrand Russell in 1905. Russell argued that a statement of the form ‘the  $F$  is  $G$ ’ is true just in case there is exactly one (relevant)  $F$ , and this one  $F$  is also  $G$ . If we have an identity predicate, we can easily express this in the language of predicate logic:

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

Following Russell, we might translate ‘The current Prime Minister is not Prime Minister’ as

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

This is indeed a contradiction: 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) Hillary Clinton might have been the 45th US President.
- (b) Smith’s murderer could have been a woman.
- (c) Alice believes that the student representative is rude.