less flexible ones, like GaveScruffy(x, y) and GaveCarl(x, y). This produces a more expressive language, and one that makes the logical relations between various claims more perspicuous.

Names can be introduced into a first-order language to refer to anything that can be considered an object. But we construe the notion of an "object" pretty flexibly—to cover anything that we can make claims about. We've already seen languages with names for people and the blocks of Tarski's World. Later in the chapter, we'll introduce languages with names for sets and numbers. Sometimes we will want to have names for still other kinds of "objects," like days or times. Suppose, for example, that we want to translate the sentences:

objects

Claire gave Scruffy to Max on Saturday.
Sunday, Max gave Scruffy to Evan.

Here, we might introduce a four-place predicate Gave(w, x, y, z), meaning *w gave x to y on day z*, plus names for particular days, like last Saturday and last Sunday. The resulting translations would look something like this:

Gave(claire, scruffy, max, saturday)
Gave(max, scruffy, evan, sunday)

Designing a first-order language with just the right names and predicates requires some skill. Usually, the overall goal is to come up with a language that can say everything you want, but that uses the smallest "vocabulary" possible. Picking the right names and predicates is the key to doing this.

Exercises

1.8 ✎ Suppose we have two first-order languages: the first contains the binary predicates GaveScruffy(x, y) and GaveCarl(x, y), and the names max and claire; the second contains the ternary predicate Gave(x, y, z) and the names max, claire, scruffy, and carl.

1. List all of the atomic sentences that can be expressed in the first language. (Some of these may say weird things like GaveScruffy(claire, claire), but don't worry about that.)
2. How many atomic sentences can be expressed in the second language? (Count all of them, including odd ones like Gave(scruffy, scruffy, scruffy).)
3. How many names and binary predicates would a language like the first need in order to say everything you can say in the second?

Table 1.2: Names and predicates for a language.

English	FOL	Comment
Names:		
Max	max	
Claire	claire	
Folly	folly	The name of a certain dog.
Carl	carl	The name of another dog.
Scruffy	scruffy	The name of a certain cat.
Pris	pris	The name of another cat.
2 pm, Jan 2, 2011	2:00	The name of a time.
2:01 pm, Jan 2, 2011	2:01	One minute later.
⋮	⋮	Similarly for other times.
Predicates:		
x is a pet	Pet(x)	
x is a person	Person(x)	
x is a student	Student(x)	
x is at home	Home(x)	
x is happy	Happy(x)	
t is earlier than t′	$t < t'$	Earlier-than for times.
x was hungry at time t	Hungry(x, t)	
x was angry at time t	Angry(x, t)	
x owned y at time t	Owned(x, y, t)	
x gave y to z at t	Gave(x, y, z, t)	
x fed y at time t	Fed(x, y, t)	

1.9 We will be giving a number of problems that use the symbols explained in Table 1.2. Start a new sentence file in Tarski's World and translate the following into FOL, using the names and predicates listed in the table. (You can switch to the Pets language in the Sentence toolbar. When you type, make sure they appear exactly as in the table; for example, use 2:00, not 2:00 pm or 2 pm.) All references to times are assumed to be to times on January 2, 2011.

1. *Claire owned Folly at 2 pm.*
2. *Claire gave Pris to Max at 2:05 pm.*
3. *Max is a student.*
4. *Claire fed Carl at 2 pm.*
5. *Folly belonged to Max at 3:05 pm.*
6. *2:00 pm is earlier than 2:05 pm.*

Name and submit your file in the usual way.

1.10 ✎ Translate the following into natural sounding, colloquial English, consulting Table 1.2.

1. Owned(max, scruffy, 2:00)
2. Fed(max, scruffy, 2:30)
3. Gave(max, scruffy, claire, 3:00)
4. 2:00 < 2:00

1.11 ✎⋆ For each sentence in the following list, suggest a translation into an atomic sentence of FOL. In addition to giving the translation, explain what kinds of objects your names refer to and the intended meaning of the predicate you use.

1. *Max shook hands with Claire.*
2. *Max shook hands with Claire yesterday.*
3. *AIDS is less contagious than influenza.*
4. *Spain is between France and Portugal in size.*
5. *Misery loves company.*

Section 1.5

Function symbols

function symbols

Some first-order languages have, in addition to names and predicates, other expressions that can appear in atomic sentences. These expressions are called *function symbols*. Function symbols allow us to form name-like terms from names and other name-like terms. They allow us to express, using atomic sentences, complex claims that could not be perspicuously expressed using just names and predicates. Some English examples will help clarify this.

English has many sorts of noun phrases, expressions that can be combined with a verb phrase to get a sentence. Besides names like *Max* and *Claire*, other noun phrases include expressions like *Max's father*, *Claire's mother*, *Every girl who knows Max*, *No boy who knows Claire*, *Someone* and so forth. Each of these combines with a singular verb phrase such as *likes unbuttered popcorn* to make a sentence. But notice that the sentences that result have very different logical properties. For example,

Claire's mother likes unbuttered popcorn

implies that someone likes unbuttered popcorn, while

No boy who knows Claire likes unbuttered popcorn

does not.

terms

Since these noun phrases have such different logical properties, they are treated differently in FOL. Those that intuitively refer to an individual are