Title	Chapter 3 : Predicates and arguments : Introduction to semantics for non-native speakers of English
Sub Title	
Author	Tancredi, Christopher
Publisher	慶應義塾大学言語文化研究所
Publication year	2023
Jtitle	慶應義塾大学言語文化研究所紀要 (Reports of the Keio Institute of Cultural and Linguistic Studies). No.54 (2023. 3) ,p.287- 301
JaLC DOI	10.14991/005.00000054-0287
Abstract	
Notes	研究ノート
Genre	Departmental Bulletin Paper
URL	https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=AN00069467-00000054-0287

Reports of the Keio Institute of Cultural and Linguistic Studies 54 (2023), 287–301

Chapter 3
Predicates and Arguments
Introduction to Semantics for Non-native Speakers of English

Christopher Tancredi

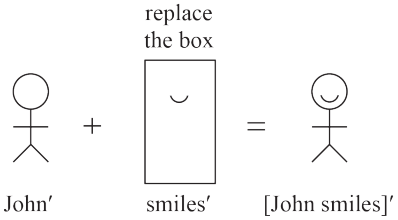
In Chapter 2, in order to analyze *and*, *or*, and *if*, we did not need to know anything about the structure of the sentences they combine with. It was enough to know that those sentences have propositions as their meanings. The meanings of *and*, *or*, and *if* cared only about the truth-values of the sentences they connect. They did not care about how those truth-values were determined. To give a full semantics of English, however, it is important to show how the truth-values are determined. In this chapter we will start to do so.

All sentences are made up of words, and every word adds something to the sentence it occurs in. However, what kind of thing a word adds depends on the kind of word it is. We can show this with the sentence *John smiles*. Knowing the meaning of this sentence allows you to picture a situation in which it is true. Of course, you have to know who John is in order to get a complete picture. But even just knowing that John is a person allows you to get a pretty good picture.

Now consider the two words that make up that sentence. The word *John* is a name. If you know who it names, you can easily picture that person. Even if you do not know who it names, you can still imagine some person that the name picks out. The word *smiles* is different. You know, of course, what it looks like when someone smiles. But try to picture *smiles*. Not *John smiles* or *someone smiles*, but just *smiles* alone. You will notice that it can't be done. The verb *smiles* does not have a meaning that we can picture.

So, what kind of a meaning does *smiles* have? We can think of its meaning like

instructions for how to make a picture. The instructions are like a picture with something missing. They tell you to take what you are given and add it to the picture so that it has a smile. If you are given a person, you end up with a picture of that person smiling.



Syntactically, expressions like *smiles* that act like instructions in this way are **predicates**. Their meanings are **properties**. The predicate *smiles* needs to combine with a subject like *John* to make a complete sentence. The subject is called a **syntactic argument** of *smiles*. The property denoted by *smiles* needs to be given an individual like john in order to make a complete picture. This individual is a **semantic argument** of *smiles*.

3.1 Predicate Logic

In Chapter 2, propositional logic helped us give clear, testable meanings for *and*, *or*, *if* and *not*. It allowed us to analyze complex propositions that combine logical operators with basic propositions. However, it cannot analyze basic propositions. For that we turn to **predicate logic**.

Analyzing an English sentence using predicate logic involves two steps. First, we translate the sentence into a **formula** of predicate logic. Second, we give rules for determining the **denotation** of the predicate logic formula. Our reason for using two steps is the same as before. We can define the expressions of our logic however we want to. We can then test different definitions to see which one gives us the best analysis of the English expressions they translate.

The basic expressions of predicate logic are predicates and **terms**. Terms include names of **individuals**, where an individual can be a person, an animal, or an object like a flower or table. Individuals are also sometimes called **entities**. In simple cases, the expressions of English have obvious translations in logic. The sentence *John*

smiles, for example, translates as:

SMILE (john)

Here, the English predicate *smiles* translates directly as the logical predicate SMILE, and the English name *John* translates as the logical term john.

Logical predicates can take any number of terms. A predicate that takes n terms is an *n*-place predicate. A basic formula in predicate logic is made up of an *n*-place predicate followed by a list of *n* terms:

PREDICATE ($term_1$, $term_2$, ..., $term_n$)

Notice that the logical predicate is written in capitals, and the terms are not capitalized. This makes it easier to distinguish between expressions of English and their translations in logic.

We said in Chapter 1 that sentences have truth-values as their denotation. If John smiles, the sentence *John smiles* denotes *True*. If John does not smile, the sentence denotes *False*. We also said that a 1-place predicate denotes a set, and that a name denotes the individual it names. So how do we combine the denotation of a 1-place predicate with the denotation of a name to get the denotation of a sentence?

Consider our sentence *John smiles* and its predicate logic translation, SMILE (john). SMILE denotes the set of individuals who smile. Suppose that there are 5 individuals in the world: *a*, *b*, *c*, *d*, and *j*. These individuals are named *Alice*, *Betty*, *Charles*, *David*, and *John*. Suppose that of these people, *a* and *b* smile but *c*, *d*, and *j* do not. Then the denotation of SMILE, written SMILE′, is the following:

SMILE′ = {*a*,*b*}

The denotation of john, the predicate logic translation of the name *John*, is the individual *j*.

john′ = *j*

The task is then to determine a truth value from these two pieces. We cannot do this by putting the two denotations together into a list. Whatever the list $\{a,b\}, j$ is, it is not a truth value. Rather, we need a rule. The rule we use is this:

> A formula of the form PREDICATE (term) is True if the denotation of the term **is a member of** the denotation of the PREDICATE. It is False otherwise.

In the case of *John smiles*, this rule tells us the following:

> SMILE (john) is True if j is a member of $\{a,b\}$, and it is False otherwise.

The set $\{a,b\}$ has two members, a and b. Since j is not a member of this set, SMILE (john) is False.

If all predicates were 1-place predicates, we could stop here. However, many English predicates take more than one argument:

> 1-place predicates: *laugh*, *fall*, *amazing*, *happy*, *nurse*, *dog*, *out*
> 2-place predicates: *see*, *kiss*, *say*, *from*, *with*, *brother*, *picture*, *proud*
> 3-place predicates: *give*, *tell*, *say*, *send*, *teach*, *introduce*

It is natural to translate these English predicates into predicate logic predicates with the same number of terms. This gives us predicate logic formulas like the following:

> LAUGH (term)
> SEE (term_1, term_2)
> GIVE (term_1, term_2, term_3)

The analysis we gave for 1-place predicates does not work for 2- and 3-place predicates. 2-and 3-place predicates do not describe properties of individuals. They rather describe **relations** among individuals. A property is a set of individuals. Intuitively, the individuals in the set are the individuals that have the property in question. Relations cannot be sets of individuals. A set of individuals will not tell you which members were seers and which members were seen, for example. Nor will it

tell you how to match the seers with the individuals they see.

To describe relations, we use **tuples**. A tuple is an ordering of individuals. An **ordered pair** <*a*,*b*> is an ordering of two individuals, also called a 2-tuple. An **ordered triple** <*a*,*b*,*c*> is an ordering of three individuals, also called a 3-tuple. Just like SMILE is a 1-place predicate that denotes a set of individuals, we define SEE as a 2-place predicate that denotes a set of pairs of individuals. The pairs we want are all pairs of the form <*x*,*y*> such that *x* sees *y*. In a tuple, the order matters. The ordered pair <*a*,*b*> is different from the ordered pair <*b*,*a*>. If *a* sees *b* but *b* does not see *a*, then <*a*,*b*> will be in the denotation of SEE but <*b*,*a*> will not. This makes ordered pairs different from sets. The set {*a*,*b*} is the same as the set {*b*,*a*}. In a set, the only thing that matters is what members it has.

To illustrate, suppose that Alice sees Betty and Charles, Betty sees Charles and David, and no one else sees anyone else. Then the denotation of SEE is the following:

{<*a*,*b*>, <*a*,*c*>, <*b*,*c*>, <*b*,*d*>}

To use this denotation, we need a new rule for interpreting formulas. Consider a formula of the form PREDICATE ($term_1$, ..., $term_n$). For each term of the form $term_i$, take t_i to be its denotation. Then our new rule is the following.

> A formula of the form PREDICATE ($term_1$, ..., $term_n$) is True if the n-tuple <t_1, ..., t_n> is a member of the denotation of the PREDICATE. It is False otherwise.

We can use this new rule to determine whether SEE (betty, alice) is True or False in the situation above. This rule gives us the following:

> SEE (betty, alice) is True if <*b*,*a*> is a member of the set {<*a*,*b*>, <*a*,*c*>, <*b*,*c*>, <*b*,*d*>}. It is False otherwise.

Since <*b*,*a*> is not a member of {<*a*,*b*>, <*a*,*c*>, <*b*,*c*>, <*b*,*d*>}, SEE (betty, alice) is False.

3.2 The Semantics of English Predicates

To interpret an English expression, we translate it into logic and interpret the logic. We just saw how to interpret logical names and predicates. We have not yet seen how to translate English expressions into logic, though.

To give a translation of an English sentence, we need to make some decisions. Consider the sentence *Alice sees Betty*. This sentence describes Alice as the one seeing, and Betty as the one who is seen. How do we represent this relation in logic? In principle, there are at least the following two options.

> SEE_1 (alice, betty): the denotation of SEE_1 is the set of pairs <*x*,*y*> such that *x* sees *y*
> SEE_2 (betty, alice): the denotation of SEE_2 is the set of pairs <*x*,*y*> such that *y* sees *x*

The only difference between SEE_1 and SEE_2 is the ordering of their arguments. For SEE_1, the pairs are ordered <seer, seen>. For SEE_2, they are ordered <seen, seer>. Which one should we use as the translation of *Alice sees Betty*?

There is no right answer to this question. Since both translations identify Alice as the seer and Betty as the one seen, both are ok. If we wanted to, we could even include both options in our semantics and nothing would go wrong. In practice, however, we make a choice. The standard choice is SEE_1. The reason for the choice is because the order of the arguments for SEE_1 matches the order of the arguments for English *see*. More generally, we translate English sentences into logical formulas where the order of terms matches the normal order of English arguments given below:

> Subject < Direct Object < Indirect Object < Complement

In addition to ordering of arguments, we also have to make a decision about the number of arguments the logical predicate should have. Consider the following sentences:

> John eats.
> John eats the cake.

Can we translate these sentences as follows?

EAT (john)
EAT (john, the cake)

If we do, then the English predicate *eats* is given the same translation in both cases, namely as EAT. However, the logical predicate EAT is then ambiguous: it has two interpretations. Under one interpretation EAT takes one argument -- the one who does the eating. Under another interpretation EAT takes two arguments -- the one who does the eating and the thing eaten.

There is no law saying that logical expressions cannot be ambiguous. However, in general, semanticists prefer a logic that is unambiguous, and this book will use an unambiguous logic. This makes it easier to see where there is ambiguity in English. With an unambiguous logic, if the English word *eat* has two different translations into logic, it is ambiguous. If it has only one translation into logic, then it is unambiguous. So, is the English word eat ambiguous? Consider two ways of analyzing it.

Option 1:		
John eats	is translated as	$\underline{\text{EAT}}_1$ (john)
John eats the cake	is translated as	$\underline{\text{EAT}}_2$ (john, the cake)

Option 2:		
John eats	is translated as	For some *x*, EAT (john, *x*)
John eats the cake	is translated as	EAT (john, the cake)

Under the first option, English *eat* is ambiguous. It can be interpreted as $\underline{\text{EAT}}_1$ or as $\underline{\text{EAT}}_2$, where $\underline{\text{EAT}}_1$ takes one argument, and $\underline{\text{EAT}}_2$ takes two arguments. Under the second option, English *eat* is unambiguous. It is always translated as $\underline{\text{EAT}}$, which always takes two arguments. For the sentence *John eats*, which has a subject but no object, the second argument is like a silent *something*. The analysis of this silent *something* has two parts. *For some x* is placed before $\underline{\text{EAT}}$, and the second argument of $\underline{\text{EAT}}$ is filled by the variable *x*. The formula is true just in case for some individual *x*, the pair <*j*,*x*> is in the denotation of $\underline{\text{EAT}}$.

How can we choose between these two options? Observe that intuitively, *John eats the cake* entails *John eats*. If *John eats the cake* is true, it follows that *John eats* is true too. Do the two options predict this relation? In the case of Option 1, the answer is a clear *no*. The predicates $\underline{\text{EAT}_1}$ and $\underline{\text{EAT}_2}$ are different basic predicates. Though they look similar, they are as different as LAUGH and SEE. Suppose John sees the cake. Then SEE (john, the cake) is true. Does it follow that John laughs? That is, does it follow that LAUGH (john) is true? Clearly not. The reason why not is also clear: SEE and LAUGH are unrelated. If $\underline{\text{EAT}_1}$ and $\underline{\text{EAT}_2}$ are different basic predicates, then they too are unrelated. If $\underline{\text{EAT}_2}$ (john, the cake) is true, then, nothing follows about the truth or falsity of $\underline{\text{EAT}_1}$ (john). Option 1 does not explain our intuition about entailment.

What about Option 2? Does it predict that *John eats* follows from *John eats the cake*? Yes, it does. We said that For some x, EAT (john, x) is true just in case for some individual *x*, the pair <*j*,*x*> is in the denotation of EAT. The formula EAT (john, the cake) is true just in case the pair <*j*,*c*> is in the denotation of EAT, where *c* is the denotation of *the cake*. It follows from this that whenever EAT (john, the cake) is true, For some x, EAT (john, x) is also true. Option 2 thus predicts that *John eats the cake* entails *John eats*, as desired. This makes Option 2 a better option than Option 1.

3.2.1 Obligatory Expressions and Optional Expressions

The sentences *John eats* and *John eats the cake* are both acceptable sentences of English. One way to describe this fact is to say that the **object** of the verb *eat* is **optional**. It does not have to appear in order to make an acceptable English sentence using the word *eat*. Other expressions are **obligatory**: they have to be there. We can see this by comparing *eat* with *devour*. *Devour* means to eat quickly. However, **John devours* is not an acceptable English sentence. Unlike with *eat*, the object of *devour* is obligatory.

How do we explain the fact that the object of *eat* is optional while the object of *devour* is obligatory? If we adopt Option 1 for our analysis of *eat*, the explanation is easy: *eat* is ambiguous but *devour* is not. Option 1, however, did not explain our intuitions about entailment. If we adopt Option 2 for our analysis of *eat*, the explanation is harder. We need a reason why *John devours* cannot be analyzed like *John eats*.

Conceptually, both *eat* and *devour* require something that is consumed. We cannot picture someone eating without their eating something, and similarly for devouring. This supports Option 2 above. On Option 2, there is only one translation for *eat*, namely as EAT, and EAT requires two arguments. However, it also means that the optional/obligatory difference between *eat* and *devour* cannot be encoded in the differences between EAT and DEVOUR, since DEVOUR requires two arguments as well. That is, whether the object of a verb is obligatory or optional cannot be determined by whether an object argument is conceptually necessary.

If the optional/obligatory difference between *eat* and *devour* is not encoded in EAT and DEVOUR, where does it come from? The best explanation is that it comes from the syntax. Syntactically, *devour* requires an object, but *eat* does not. Since this is an introduction to semantics and not to syntax, we will not ask how the syntax makes this distinction. If it does so, however, then syntactic requirements and semantic requirements do not have to match. We will say that *eat* can take either one or two syntactic arguments, but always takes two semantic arguments. *Devour*, in contrast, always takes two syntactic arguments and two semantic arguments.

If the number of syntactic arguments of a predicate can be different from the number of its semantic arguments, how can we know how many semantic arguments a predicate has? Before we said that John smiles can be represented as SMILE (john). We just saw that a predicate can have more semantic arguments than syntactic arguments, though. How can we decide the correct number, then?

A way suggested by the analyses of *eat* and *devour* is to ask what is conceptually necessary. Can you conceptually picture smiling without also picturing something or someone who smiles? If so then SMILE does not need a semantic argument. If not, it does. In this case our intuitions tell us that SMILE needs a semantic argument. Let us put this idea into a proposal.

> **Proposal**
> The semantic arguments of a predicate are those arguments that are required in order to make a proposition that can be conceptualized.

According to this proposal, john is a semantic argument of SMILE. Taking john away

from SMILE (john) gives us something that we can no longer picture to ourselves.

If we take the above proposal seriously, there are more semantic arguments than we have seen so far. We cannot imagine John smiling, after all, without imagining him smiling at some time and in some place. According to the proposal, then, SMILE has at least three arguments: a smiler, a time, and a place, or **location**. We represent these additional arguments the same way we represented the optional argument of *ate* above. In particular, we analyze *John smiles* as follows:

For some time t and for some location l, SMILE (john, t, l)

Though this is the official analysis of *John smiles*, we will still sometimes use SMILE (john) instead, for example when we do not care about the time or location of the smiling.

3.2.2 Referring to times and places

If SMILE has semantic time and place arguments, we expect to be able to say something about them. Indeed we can. We can say *John smiled in the park yesterday*, for example. This tells us that the place of John's smiling is somewhere in the park, and that the time of the smiling was sometime yesterday. How should we translate this sentence into logic? One option is to translate *yesterday* and *in the park* as semantic arguments, or terms, just like *John*. Under this option, ignoring the past tense of *smiled*, the logical translation of the sentence becomes:

SMILE (john, yesterday, in the park)

However, this option faces a major challenge. A sentence can contain many expressions describing the time or place of an event. For example, we can say *John smiled yesterday in the park in the afternoon near the fountain*. Here there are two expressions that describe the time and two that describe the place. If each of these expressions is translated as a term, then we need a version of SMILE, SMILE_5 for example, that takes five arguments rather than three:

SMILE_5 (john, yesterday, in the park, in the afternoon, near the fountain)

But SMILE$_5$ is a different predicate from SMILE. This means that we do not expect a formula using SMILE to be entailed by a formula using SMILE$_5$. We observe, however, that the longer sentence does entail the shorter one. If John smiled yesterday in the park in the afternoon near the fountain, it follows that John smiled yesterday in the park. Translating all expressions of time and place as arguments of SMILE$_5$ does not explain this entailment. We need another option.

A second option is to take *smiled* to be unambiguous, always being translated as SMILE, but to not translate *yesterday*, *in the park*, *in the afternoon* or *near the fountain* as terms. Under this option, we translate them as formulas that add information about the time t and the place p arguments of SMILE:

> For some time t and for some place p, SMILE (john, t, p) & YESTERDAY (t) & IN (p, the park)

> For some time t and for some place p, SMILE (john, t, p) & YESTERDAY (t) & IN (p, the park) & IN (t, the afternoon) & NEAR (p, the fountain)

Here, it is clear that the longer formula entails the shorter one. This follows from the fact that in logic, *p & q* entails *p*. This second option is therefore better than the first option, since it predicts the observed entailments that the first option does not.

3.2.3 Prepositions

English prepositions play many roles. Sometimes they describe where something is, like when you say *John is in*/ *behind*/ *above*/ *near the car*. Here, the prepositions are 2-place predicates of **location**. They describe how the location of John relates to the location of the car at a fixed time. Sometimes they describe the **path** something took, like when you say *John went into*/ *toward*/ *through*/ *over the town*. Here again the prepositions are 2-place predicates. This time, though, they do not describe John's single location at a fixed time. Rather, they describe his change of location through time. Most prepositions can be used in both of these ways. Whether they describe a location or a path depends on the main predicate.

As seen in the examples below, sentences can contain many preposition phrases describing places or paths. (We ignore times.)

John stood on the chair, near the door, behind Mary.
For some **location** l, STAND (john, l), & ON (l, the chair) & NEAR (l, the door) & BEHIND (l, mary)

John sent Mary down the hill, through the tunnel, to the store.
For some **path** p, SEND (john, mary, p), & DOWN (p, the hill) & THROUGH (p, the tunnel) & TO (p, the store)

STAND (john, l) says that John stands at location l. The conjoined formulas give more information about l: that l is on the chair, that l is near the door, and that l is behind Mary. SEND (john, mary, p) says that John sent Mary along path p. The conjoined formulas again give more information about p: that p goes down the hill, that p goes through the tunnel, and that p goes to the store.

A third use of prepositions is for making **phrasal verbs**. These are expressions made from a verb and a preposition. The meaning of a phrasal verb is not composed from the meanings of its parts. To *look up* a word in a dictionary does not involve looking in an upward direction. *Look up* in this sense is a phrasal verb. To *go over* something can be to move from one side of something to another by traveling above it. It can also be to review something. The first meaning is a compositional meaning. It combines the meaning of *go* with the meaning of *over*. The second one is not compositional. It uses *go over* as a phrasal verb. We translate phrasal verbs into logic as single predicates. The translations of *look up* and of *go over* as phrasal verbs are LOOK-UP and GO-OVER. Though these look like they are made of parts, in the logic they are treated as independent predicates. LOOK-UP is as different from LOOK and UP as CHECK is. Similarly, GO-OVER is as different from GO and OVER as REVIEW is. The two translations of *go over* are shown below:

John goes over the plans.
phrasal verb translation: GO-OVER (john, the plans)
compositional translation: For some path p, GO (john, p) & OVER (p, the plans)

The first translation means that John reviews the plans. The second means that John moves along a path that goes above the plans.

3.2.4 Sentences as arguments:

All of the examples we have looked at so far have been simple sentences. Their predicate logic translations have only individual-denoting expressions as terms. English also has complex sentences that contain, or **embed**, other sentences. To translate these into logic, we allow formulas to be terms as well. This is shown below with the predicates *believe* and *show*.

John believes that Tokyo is south of Hong Kong
BELIEVE (john, SOUTH (tokyo, hong kong))

That John answers the questions shows that John is smart
SHOW (ANSWER (john, the questions), SMART (john))

The translation of these sentences into logic is usually easy. As we will see in Chapter 9, however, interpreting these logical formulas can be difficult. Because of these difficulties, we will avoid using examples with embedded sentences until then.

3.3 Set Theory

In predicate logic, predicates denote sets. Until now we have described the sets we need using English. It is more common, however, to describe sets using set theory. Set theory is a branch of mathematics. As with logic, we only introduce parts of set theory that are useful for doing semantics.

A set is a collection of objects. The objects in a set can be everyday objects like pens and people. They can be expressions of a language like the English name *John*, or the predicate SEE of predicate logic. They can also be other sets. In fact, anything at all can be a member of a set.

Semantics only uses **uniform** sets. These are sets whose members are all the same kind of thing. The denotation of (the simple version of) SMILE, for example, is a set of individuals, while the denotation of SEE is a set of pairs of individuals. These sets look something like this:

SMILE: {*a*, *b*, *c*, ...}

SEE: $\{\langle a,b\rangle, \langle a,c\rangle, \langle b,c\rangle, ...\}$

Set theory allows us to mix different kinds of things in a set. We can form sets like the following, for example:

$\{a, \langle a,b\rangle, \langle b,b,c\rangle, \{a,b\}\}$

While this is a well-formed set in set theory, however, we have no use for such non-uniform sets in semantics.

Individuals are different from ordered pairs of individuals. They are also different from sets of individuals. The individual named Alice, *a*, is different from the set that contains that individual, $\{a\}$. The individual and the set are clearly related, though. The individual *a* is a **member** of the set $\{a\}$. It is also a member of the set $\{a, c, d\}$, and of many other sets. We use the symbol '$\in$' to show this membership relation:

Set theory	**English description**
$a \in \{a\}$	*a* is a member of the set containing *a*.
$a \in \{a, c, d\}$	*a* is a member of the set containing *a*, *c* and *d*.
$\langle a,c\rangle \in \{\langle a,b\rangle, \langle a,c\rangle, \langle b,c\rangle\}$	The ordered pair $\langle a,c\rangle$ is a member of the set containing the ordered pairs $\langle a,b\rangle$, $\langle a,c\rangle$, and $\langle b,c\rangle$.

A set is defined entirely by its members. Something is either in a set or it is not. It cannot be in the set twice. $\{a, b, c\}$ is a well-formed set. $\{a, b, a\}$ is not, because *a* occurs twice. Also, unlike ordered pairs and n-tuples, in a set the order of the members does not matter. The following sets are all identical:

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

Very often, we do not know the exact members of a set that we want to use. We might want to talk about the set of all kings even though we do not know who all the kings are. In those cases, we can describe that set as we just have, using English. We can also describe that set by giving a condition for membership:

Set theory	**English description**
$\{x: x \in \text{KING}'\}$	The set of all x such that x is a member of the denotation of KING

This way of describing a set makes use of a **variable**, here the variable x. You can think of a variable as a place holder. In this case, it is a place holder for an individual. To know if an individual belongs in the set, you put the individual in place of x. If you end up with something that is true, then that individual is in the set. If you end up with something false, that individual is not in the set. King John of England, for example, is in the set because the denotation of KING contains King John of England as a member. I am not in the set because the denotation of KING does not contain me as a member.

Semanticists often use a mix of set theory and English or of set theory and logic to describe sets:

Mixed set theory and English	**English description**
$\{x: x$ is a king$\}$	The set of all x such that x is a king

Mixed set theory and logic	**English description**
$\{x: \text{KING}(x)\}$	The set of all x such that x is a king

These are informal ways of picking out sets. In $\underline{x \text{ is a king}}$, *is a king* is an expression of English. In order to combine this expression with something in place of x, that something has to be an expression of English as well, such as a name, not an individual. This makes it look like *{x: x is a king}* should be a set of English expressions. Similarly, in $\underline{\text{KING}(x)}$, $\underline{\text{KING}}$ is a logical predicate. This means that x has to play the role of a logical term. This makes it look like *{x: KING(x)}* should be a set of logical terms. In practice, however, semanticists use these informal descriptions of sets to pick out sets of individuals, not sets of English names or of logical terms.