Foundations of Semantics I:
Truth-conditions, entailment and logic

Rick Nouwen | R.W.F.Nouwen@uu.nl

February/March 2011, Gent

1 Meaning

There can be no simple answer to the question of what the meaning of a sentence is. The word *meaning* has many different senses. This is why in linguistics we tend to be very explicit about the kinds of meanings we are studying. Here are two examples:

(1) What is the meaning of a natural language expression?
   a. *that to which it refers*
   b. *that which the speaker intends to communicate with it*

The first answer to the question, (1-a), is what is normally associated with the study of natural language semantics. Central to semantics is the relationship between a sentence and the world. So, the meaning of *John* is the individual we refer to by the word 'John' and the meaning of *John hates Bill* is what needs to be the case in order for this sentence to be true.

One could see the answer in (1-b) as typical to pragmatic inquiry. Beyond mere reference, pragmatic meanings are about the *use* of a sentence in a particular context. For instance, consider the following context. Someone has just asked *Will John invite Bill to the party?* If in this context, I answer *John hates Bill*, it is likely that I intend this sentence to be understood as a negative answer to the question whether John will invite Bill to the party.

There are interesting ways in which pragmatic meaning appears to depend on semantic meaning, and there are many phenomena that straddle the semantics/pragmatics divide. It is therefore important to be very precise about what kind of meaning one focuses on. Central to this course is the referential theory of meaning that underlies what is often called *formal*, or *truth-conditional*, or *model-theoretic semantics*.

2 Truth-conditions

Apart from the referential nature of meaning, one crucial assumption in formal semantics concerns what it means to know the (semantic) meaning of a sentence. Consider, (2).

(2) Rick has a 50 cent coin in his wallet.

To know (semantically) what (2) means is to be able to distinguish a situation in which (2) is true, from one in which (2) is false. You clearly know how you would go about this: all you need to do is look in my wallet and sift through the coins on the lookout for a 50c coin. Consequently, you know what (2) means, even though you don't know whether or not it is true.

> *To know the meaning of S is to know when S is true; that is, to know the conditions that make it true: its* truth-conditions.

Crucially, (2) tells us something very specific about the world we live in, namely that if you were to look in my wallet you would find a 50c coin. This information is very specific in the sense that it leaves a lot of other stuff open: whether there is more than one coin in my wallet; whether there are any other coins in my wallet; what my wallet looks like; etc. In other words, there exists an infinity of situations that make (2) true, but all these situations have one thing in common. This one thing is what the meaning of (2) corresponds to.

If you haven't looked in my wallet yet, and I assert (2) and you believe me, then you will have gained information. Before accepting (2), you did not know whether the world you lived in was one in which I have a 50c coin in my wallet. Afterwards, you did know (or at least believed so). This is what meanings in the relevant sense do: they convey information about the world.[1]

Formal or truth-conditional semantics is sometimes called *model-theoretic* semantics. The idea is that a sentence is true or false only with respect to a particular way things are, a particular model of what is reality. In some state of affairs, the sentence is true, and in some others it will be false. Such alternative state of affairs are often called *a possible world*. Imagine that apart from the world we live in, there are many other possible worlds. Truth-values are relative to such *possible worlds*. For instance:

(3) World A: Obama is president of the US, Rick is a semanticist, Rome is the capital of Italy, Rick has a 50 cent coin in his wallet, etc.

(4) World B: Obama is president of the US, Rick is a semanticist, Rome is the capital of Italy, Rick doesn't have a 50 cent coin in his wallet, etc.

(5) World C: Hilary Clinton is president of the US, Rick is a baker, Rome is the capital of Italy, Rick has a 50 cent coin in his wallet.

(6) World D: etc.

The idea is that all the facts that are true describe a unique possible world, namely *the actual one*. At the same time, there are many alternatives to these actual matters of fact. Semantic sentence meanings can steer us towards finding out which possible world is actual. This is because semantic sentence meanings, i.e. truth-conditions, are particular ways of distinguishing different possible worlds. One knows the meaning of *Rick has a 50 cent coin in his wallet* if and only if one can distinguish worlds / models in which it is true from worlds in which it is false.

3 Entailment

One of the prime sources of data for the study of semantics are entailments. You can use (intuitions about) entailments to establish whether two (declarative) sentences are semantically independent, semantically related or semantically identical.

Entailment — *Sentence S entails sentence S′ if and only if whenever S is true, S′ is true too*

In (7), you find an example of an entailment, indicated with ⇒.

(7) a. John owns a blue sweater.
    b. ⇒ John owns a sweater.

[1] Contrast this kind of informational view on meaning to the following use of the verb *to mean*, which is not the primary sense of meaning that we're after:

(i) Rick has a 50c coin in his wallet. This *means* that Rick can get a shopping trolley.

The denial of something true is false, and so, by the definition of entailment, we come to expect that a sentence together with the denial of one of its entailments forms a *contradiction*. Contradictions cannot be uttered felicitously.

(8) #John owns a blue sweater, but he does not own a sweater.

3.1. Entailment relations and truth-conditions

Given the notion of entailment, there are three kinds of meaning relations that may exist between two sentences.

(9)

either	S⇒S′	or	S′⇒S	truth-conditionally related
neither	S⇒S′	nor	S′⇒S	truth-conditionally unrelated
both	S⇒S′	and	S′⇒S	truth-conditionally equivalent

The sentences in (7) are truth-conditionally related. That is, *John owns a blue sweater* entails *John owns a sweater*, but not vice versa. Consequently, we cannot find a situation in which the former is true, but the latter false, while we can find a situation in which it is true that John owns a sweater, but false that he owns a blue one. (Just take a situation in which John's sweater is red.)

Two truth-conditionally equivalent sentences, i.e. two sentences that entail one-another, have exactly the same semantic meaning. This means that there are no situations in which their truth value differs: when one of the two sentences is true, then both of them are true; when one of the two sentences is false, then both of them are false.

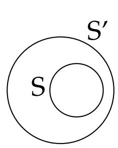
For instance, (10-a) both entails and is entailed by (10-b). This suggests that the dative alternation in English has no semantic import.

(10) a. John gave Mary an apple.
b. John gave an apple to Mary.

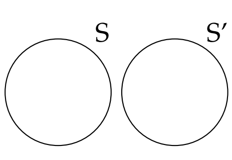
In summary, the relation between entailments and truth-conditions is a tight one. Above we said that S entail S′ if whenever S is true, S′ is true too. An alternative way of saying the same thing is to make use of the notion of possible world:

> **Entailment —** *Sentence S entails sentence S′ if and only if S′ is true in all possible worlds in which S is true*

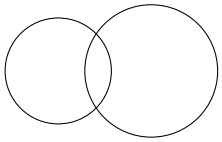
We can depict truth-conditions by sketching the worlds in which sentences are true by means of Venn diagrams.

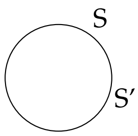
(11) 

S entails S′

(12) 

S and S′ are contradictory

(13)  no entailment relation

(14) 

equivalence

Exercise— How would you show that the two sentences in (15) are *not* truth-conditionally equivalent?

(15) a. All students didn't do their homework.
b. Not all students did their homework.

Do the same for (16):

(16) a. The two girls lifted a piano.
b. Both girls lifted a piano.

Exercise: What are the entailment relations between the following (a.) and (b.) sentences?

(17) a. All famous boxers are rich.
b. All boxers are rich.

(18) a. John does not own a blue sweater.
b. John does not own a sweater.

Exercise — We can test a semantic theory, by testing the predicted entailments. Say, our theory gives the following truth-conditions for the sentence S="John did not see a unicorn": S is true only in situations in which there exists a unicorn that John did not see. Show that these truth-conditions make a wrong prediction.

Exercise — Discuss the following sentences with respect to entailments.

(19) a. John was born in London or in Manchester.
b. John was born in London or John was born in Manchester.
c. Everybody in my class was born in London or in Manchester.
d. Everybody in my class was born in London or everybody in my class was born in Manchester.

Exercise — Discuss the meaning of *cold*, *warm* and *hot*, given the following intuition that (20-b) is a contradiction, but (20-a) is not.

(20) a. The soup is warm. In fact, it is hot.
b. The soup is warm. #In fact, it is cold.

3.2. Entailment and embedding

Entailments are sensitive to their syntactic environment.

(21) a. John owns a blue sweater. ⇒ John owns a sweater.
b. Surprisingly, John owns a blue sweater. ⇒ John owns a sweater.
c. John doesn't own a blue sweater. ⇏ John owns a sweater.
d. Does John own a blue sweater? ⇏ John owns a sweater.
e. If John owns a blue sweater, he will be considered cool. ⇏ John owns a sweater.
f. Mary thinks that John owns a blue sweater ⇏ John owns a sweater.

This is an important property, since it allows us to distinguish entailments from a second kind of inference, a presupposition.

(22) a. John's daughter is 12 years old.
b. ⇒ John has a daughter.

The relation between (22-a) and (22-b) is not one of entailment, because the inference remains in different environments.

(23) a. Is John's daughter 12 years old?
b. It's not the case that John's daughter is 12 years old.
c. If John's daughter is 12 years old, then she could become friends with Sue.

4 Writing down truth-conditions

There is an immediate problem with using truth-conditions for the meaning of sentences. How will we write them down? One option is to use paraphrases. For instance, the meaning of (24-a) is the paraphrase of truth-conditions in (24-b).

(24) a. John owns a blue sweater.
b. *There exists a blue sweater such that the individual named John owns this sweater.*

Using paraphrases has an important drawback. By using paraphrases, the language we use to state our truth-conditions in is the same as the language we are studying. We are thereby basically postponing semantic analysis, for the meaning of the paraphrase is part of what our semantic theory needs to explain. The biggest risk in this is in ambiguity. If a paraphrase is ambiguous, then it is unsuitable for representing the truth-conditions of a sentence, for it would fail to determine a single set of conditions. Consider, for instance, (25).

(25) The four boys ate three apples.

We could say that (25) is true if and only if the group of individuals that is the reference of *the four boys* have the property that is the reference of *ate three apples*. This, unfortunately, is a semantic stale mate, for (25) is very interesting from a semantic point of view, something which is completely obscured by using the language in which the original sentence was stated as the language for presenting its truth-conditions.

What makes (25) interesting is that it is ambiguous. On one reading, it says that there were three apples and that these were eaten by the boys. On another reading, however, the boys each have the property of eating three apples. While in the first reading, three apples were eaten, in the second twelve apples were eaten.

We could distinguish between the two readings by using more elaborate paraphrases, such as:

(26) a. There are three apples and the four boys ate these three apples
b. For each of the four boys there are three apples such that the boy in question ate these three apples

What these complex paraphrases underline is the complexity of reaching an accurate statement of truth-conditions. As stated before, the paraphrase needs to be completely unambiguous. The problem now is what the relation is between the original sentence (25) and the two paraphrases in (26). Ideally, we would want a systematic and completely predictive system that derives paraphrases for sentences. This is another reason why we want to use a formal language as our metalanguage. Because our theory of interpretation ought to account for the productive nature of meaning, in our mapping from object to meta-language we must be able to systematically ensure that the meta-language expression is unambiguous, we cannot do this on a case-by-case basis.

The common way to write down truth-conditions is therefore to choose a formal language as the metalanguage, and to choose a formal language of which we know exactly the correspondence to truth-conditions. Usually, we use a language called *predicate logic*. So, instead of the paraphrase in (27-b), we will use the predicate logical sentence in (27-c) to represent the truth-conditions of (27-a).

(27) a. Rick has a 50 cent coin in his wallet.
b. *there is a 50 cent coin which is contained in Rick's wallet*
c. $\exists x[\text{50-cent-coin}(x) \wedge \text{in}(\text{Rick's-wallet}, x)]$

We will work towards a detailed understanding of predicate logic below. Before we do so, it is important to understand what role the logic is going to play. A good theory of semantics will offer a system for translating a natural language sentence into a predicate logical sentence. The logic is such that we have full knowledge of what the meaning relations are between sentences in the logical language. In other words, we know exactly which logical sentences entail which other logical sentences. It is the task of the semantic theory to ensure that whenever there is an entailment relation between two natural language sentences, there is also an entailment relation between the two corresponding logical sentences, and vice versa.. Schematically,

(28)
$$\begin{array}{ccccl} S & \Rightarrow & S' & & \text{(natural language)} \\ \downarrow & & \downarrow & & \text{(semantic interpretation)} \\ \varphi & \models & \psi & & \text{(logical language)} \end{array}$$

Conversely, because we have full knowledge of the entailment relations within the logical language, by mapping natural language sentences to logical ones, a semantic theory predicts what entailment relations we should observe with respect to natural language sentences.

5 Predicate Logic

5.1. Connectives

Before we have a look at predicate logic, we have to introduce truth-conditional connectives. Such connectives have a rough correspondence to expressions in natural languages that combine or manipulate clauses. Think of coordinating expressions like *and* and *or*, which combine two truth-conditional structures to create a new structure with truth-conditional meaning. In the logical language these expressions are called connectives. Our first step to defining predicate logic will be to look at the meaning of these connectives.

Assume that in the logic we have a means of expressing *propositions*. Propositions are expressions that have a truth-value. Let's say that φ and ψ stand for such propositions, then we can define the following more complex expressions:

(29) a. $\varphi \wedge \psi$: conjunction — *'φ and ψ'*
b. $\varphi \vee \psi$: disjunction — *'φ or ψ'*
c. $\varphi \rightarrow \psi$: implication — *'if φ then ψ'*
d. $\neg\varphi$ (and $\neg\psi$): negation — *'it's not the case that φ'*

These connectives are interpreted according to how they manipulate the truth-value of the proposition(s) they combine with. For instance, for a disjunction to be true, one or more of the disjunct propositions need to be true. For a conjunction to be true, both conjuncted propositions need to be true. We can represent these truth-conditions for complex sentences in so-called *truth tables*. In such tables, but in logic in general, it is customary to refer to the truth value *true* as 1 and the value *false* as 0. Here is the interpretation of the four connectives in (29):

(30) negation

φ	$\neg\varphi$
1	0
0	1

conjunction

φ	ψ	$\varphi \wedge \psi$
1	0	0
0	1	0
1	1	1
0	0	0

disjunction

φ	ψ	$\varphi \vee \psi$
1	0	1
0	1	1
1	1	1
0	0	0

material implication

φ	ψ	$\varphi \rightarrow \psi$
1	0	0
0	1	1
1	1	1
0	0	1

Up to a certain extent, these tables will be intuitive. Most people can convince themselves that the semantics of $\neg$ and $\wedge$ has a correspondence to the meaning of *not* and *and* respectively. The meaning of $\vee$ and $\rightarrow$, however, is a different matter. It is clearly debatable whether $\vee$ is suitable to model the meaning of *or* and whether $\rightarrow$ is suitable to model the meaning of conditionals. While it is interesting to compare the connectives as defined in (30) to the meaning of potential natural language counterparts, it is important to understand that the semantics in (30) is not meant to be a semantics of natural language. It is merely a way of defining a formal language. It is up to semantic theory to study in detail the relation between the particular semantics in (30) and our observations about natural language meaning. For instance, you might believe that (31) is false if John ate both the pizza and the linguini.

(31) John had the pizza or he had the linguini.

In that case, $\vee$ does not model this use of *or*, but we need something more complex to describe natural language disjunction. If φ is the proposition that John had pizza and ψ is the proposition that he had linguini, then (31) could be thought to express (32).

(32) $(\varphi \vee \psi) \wedge \neg(\varphi \wedge \psi)$

5.2. The building blocks of predicate logic

We now know how to combine propositions into complex propositions. However, we don't yet know where the simplex propositions themselves come from. Here are some examples of basic propositions in predicate logic:

(33) a. $asleep(j)$
b. $hate(j, m)$
c. $P(x)$
d. $asleep(y)$
e. $hate(x, m)$
f. $give(j, x, m)$

The vocabulary of a predicate-logical language makes two distinctions:

1. the distinction between terms and predicates;
2. the distinction between variables and constants

In predicate logic, simple propositions are formed by predicate-argument combinations. For example, if P is a predicate and a an argument, then $P(a)$ is a proposition. Expressions that can be arguments of predicates are called *terms*. What $P(a)$ expresses is the proposition that is true if a has the property expressed by P and false otherwise. In other words, predicates express properties and terms express individual that may or may not have certain properties. For instance, with (33-a) we might be expressing that the individual j (call him John) has the property *asleep*. So, $asleep(j)$ expresses the proposition that is true if and only if John is asleep.

Some predicates take more than one term as argument, as in (33-b), which could be used to express the truth-conditions that go with *John hates Mary*.

Some terms refer to specific individuals. For instance, we can use j to stand for the individual called John, m for Mary, etc. Such terms are called *individual constants*. Not all terms are constants, however. Sometimes we need to express terms without a specific reference. Such terms are called *individual variables*. We need variables to make statements that are not about specific individuals, but about individuals in general. For instance, we could assert that there exists an individual which has both the property of being a boy and the property of being lazy. The way to do this is to use a variable and then to quantify over this variable.

(34) $\exists x[boy(x) \wedge lazy(x)]$

This says that for some value for x, it is true that this value is both a boy and lazy. For instance, if it is true that $boy(j) \wedge lazy(j)$, then (34) will be true, because we have found a way of "filling in" the variable x that makes the scope of the quantifier (everything between the brackets [and]) true, namely with j.

Note that, by itself, a variable has no real meaning. Variables are only meaningful in context. In (34), $boy(x)$ is meaningful because it is in the scope of the existential quantifier $\exists$ that quantifies over x.

Predicate logic has two modes of quantification: besides the existential quantifier $\exists$, there is the universal quantifier $\forall$. The formula in (35) expresses that all possible ways of filling in x will make $beautiful(x)$ true. In other words, everything is beautiful.

(35) $\forall x[beautiful(x)]$

Universal quantification is truly universal. For a universally quantified statement to be true, its scope has to be true of all entities in the world. So, (35) is an extremely strong statement. It does not just say that *everybody* is beautiful, it says that *everything* is. So how do we express something weaker like *every flower is beautiful* in predicate logic? We do this as follows:

(36) $\forall x[flower(x) \rightarrow beautiful(x)]$

This says that for any x, if this x is a flower, then it is also beautiful, which is the same as saying that every flower is beautiful. It is important to understand how this works. The quantifier in (36) quantifies over all entities in the world. For (36) to be true, its scope has to be true for all possible values for x. So, it shouldn't matter whether we substitute John for x, or a rose, or a daffodil, or a giraffe, all these values should make the scope of the quantifier true. Note however, that if we substitute John for x, we require the following to be true:

(37) $flower(j) \rightarrow beautiful(j)$

Since John is not a flower, (37) is trivially true. (If you don't understand why, have another look at the truth-table for implication in (30).) In fact, the only interesting cases for (36) are cases in which the value for x is a flower. All other cases make the scope of $\forall x$ vacuously true. The only way to falsify (36) is to find something that is a flower, but which is not beautiful.

Two notes on variables: The distinction between variables and constants is not only relevant with respect to terms. Predicates too can be variable or constant. For instance, (38) says that there exists some property that Mary has, but John lacks.

(38) $\exists P[P(m) \wedge \neg P(j)]$

Finally, the choice of a variable name is in principle immaterial. That is, there is no difference in meaning between (39-a) and (39-b), while there is (of course) a difference in meaning between (39-b) and (39-c) (assuming j and m are constants).

(39) a. $\exists x[boy(x) \wedge hate(x, m)]$
b. $\exists y[boy(y) \wedge hate(y, m)]$
c. $\exists y[boy(y) \wedge hate(y, j)]$

One should be careful not to use the same variable twice, however. For instance, (40-a) is not the same as (40-b).

(40) a. $\exists x \exists y[boy(x) \wedge girl(y) \wedge hate(x, y)]$
b. $\exists x \exists x[boy(x) \wedge girl(x) \wedge hate(x, x)]$

5.3. Some examples of predicate logic

(41)

John is asleep	$asleep(j)$
John is asleep and Bill is too	$asleep(j) \wedge asleep(b)$
John is asleep but Bill isn't	$asleep(j) \wedge \neg asleep(b)$
John is asleep or he is ill	$asleep(j) \vee ill(j)$
Some boy is asleep	$\exists x[boy(x) \wedge asleep(x)]$
If John is asleep, he is lazy	$asleep(j) \rightarrow lazy(j)$
Every boy is asleep	$\forall x[boy(x) \rightarrow asleep(x)]$
John hates Mary	$hate(j, m)$
Every boy hates Mary	$\forall x[boy(x) \rightarrow hate(x, m)]$

The examples in (41) are to get an idea of the kind of formulae predicate logic allows, and to which natural language sentences they roughly correspond. (Roughly, because we will only be able to properly link natural language to predicate logic once we introduce a system of compositional interpretation.)

Predicate logic comes with quite a bit of terminology. Here's a dissection of a couple of sentences in predicate logic with some important notions spelled out.

(42) $\forall x[boy(x) \rightarrow hate(x, m)]$

boy: unary predicate constant (a predicate that combines with a single argument only)
hate: binary predicate constant (a predicate that needs to combine with two arguments)
m: individual constant (a term with specific reference)
x: individual variable (a term lacking specific reference)
m, x: terms (arguments for predicates)
$\forall$: universal quantifier
$\rightarrow$: a connective, in particular so-called material implication
$boy(x) \rightarrow hate(x, m)$: the scope of $\forall x$

(43) $\exists x \exists y[boy(x) \wedge hate(x, y)]$

boy, *hate*: predicate constants
x, y: individual variables
$\wedge$: a connective, namely conjunction
$\exists$: existential quantifier
$\exists y[boy(x) \wedge hate(x, y)]$: the scope of $\exists x$
$boy(x) \wedge hate(x, y)$: the scope of $\exists y$

5.4. How to read predicate logic

(44)

$\exists x[\ldots]$	there exists an x such that ...
$asleep(x)$	x is asleep
$boy(x)$	x is a boy
$hate(j, m)$	John hates Mary
$\forall x[\ldots]$	for every x...
$\forall x[\varphi \rightarrow \psi]$	for every x such that φ it is the case that ψ
$\forall x[boy(x) \rightarrow sleeps(x)]$	for every x such that x is a boy, it's the case that x is asleep
	= every x that is a boy is asleep
	= every boy is asleep

Exercise— Read and try to understand the following formulae:

(45) a. $\forall x[student(x) \wedge \neg lazy(x)]$
b. $\forall x[student(x) \rightarrow \exists y[professor(y) \wedge hate(x, y)]]$
c. $\exists y[professor(y) \wedge \forall x[student(x) \rightarrow hate(x, y)]]$

6 Interpreting predicate logic

6.1. Predicates and Set theory

A *set* is a formal object that lumps together a collection of other formal objects. Say, there exist three objects a, b, and c, then there also exists the set $\{a, b, c\}$, the set containing all these objects. But more things are possible:

(46) a. $\{a, b\}$ the set containing a and b
b. $\{a\}$ the set containing just a
c. $\{\{a, b\}, a\}$ the set containing a and containing the set containing a and b

The entities in a set are called the *elements* of a set. Note:

(47) a. $\{a\}$ is not the same as a
b. $\{a, b, c\}$ is not the same as $\{\{a, b, c\}\}$

(To get the idea: the bag containing your purse is not the same thing as your purse. Note also that $\{a, a\}$ does not make sense. You can only put your purse into your bag once. However, this analogy does not go far: $\{a, \{a\}\}$ is a set, since it contains two different entities.)

Set theory defines a powerful toolbox of how to reason about sets and their contents. For now, we only need one notion of set theory, namely that of being an *element* of a set. (More on set theory below, though).

(48) $a \in A$: "a is an element of A"

Using $\in$, we can produce truth-conditional statements about sets and their contents. For instance, $a \in \{b, g, a, c\}$ is true, but $a \in \{b, g, c, r\}$ is false.

The relevance of set theory to predicate logic is as follows. Predicates in predicate logic are interpreted as sets. For instance, the predicate $friendly$ could be interpreted as the set of all friendly entities. Similarly, the predicate *boy* would correspond to the set of boys. We use the following notation.

(49) a. $[\![friendly]\!]^w = \{\text{John, Bill, Mary, Sue}, \dots\}$
b. $[\![boy]\!]^w = \{\text{John, Harry}, \dots\}$

Ignoring the superscript w for now, the rationale behind this is that nouns like *boy*, adjectives like *friendly*, but also intransitive verbs like *smoke* express properties (the properties of being a boy, being friendly and being a smoker, respectively). Properties divide the domain of entities; i.e. they distinguish two kinds of entities: those that have the property and those that do not. This is exactly what a set does too. The set $\{a, b, c\}$ divides the domain in entities that are included in the set (a, b and c) and entities that are not (all the rest). In sum, in logic, we do not distinguish between adjectives, nouns and intransitive verbs: they all express sets and, in predicate logic, they all correspond to predicates. Given this relation between predicates and sets, the interpretation of propositions that are made up of a predicate-argument combination is straightforward:

(50) $[\![P(a)]\!]^w = 1$ if and only if $[\![a]\!]^w \in [\![P]\!]^w$

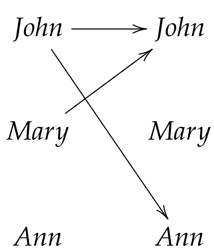
Now *asleep*(j) is true if and only if the individual j refers to is a member of the set *asleep* refers to, which is set theory speak of saying that this individual is to have the property of being asleep.

The brackets $[\![\ ]\!]$ map predicate logical expressions to entities and sets, but they only do so with respect to a parameter w. This parameter indicates the possible world with respect to which interpretation takes place. Worlds differ in all kinds of ways, for instance, with respect to who is asleep and who is awake. So, in one possible world, say w_3 above, John and Bill are both asleep, but in another (say, w_4) Bill could be awake, and in yet another (say, w_{351}) John and Bill are both awake. To know the meaning of *John is asleep* is to know for which parameters w the value of *asleep*(j) is 1.

6.2. Binary (and ternary) predicates

If unary predicate correspond to sets, what do binary predicates correspond to? The answer is that they also correspond to sets, but to sets of a certain kind, namely sets of *ordered pairs*. Assume that in the world we are considering – call it w_4 – there are only three entities, *John*, *Mary* and *Ann*, and that in this world John hates Ann, John hates himself, Mary hates John, and no-one else hates no-one else. Schematically, the hate relation in this world is as follows:

(51)



This relation corresponds to the following set:

(52) $[\![hate]\!]^{w_4} = \{\langle John, John\rangle, \langle Mary, John\rangle, \langle John, Ann\rangle\}$

The elements in this set are ordered pairs. This means that (53) is a crucially different relation from (52).

(53) $\{\langle John, John\rangle, \langle John, Mary\rangle, \langle Ann, John\rangle\}$

In interpreting the binary predicates of predicate logic, we normally use the following convention about the ordered pairs : the order of the entities in the ordered pairs reflects the syntactic argument hierarchy *subject* > *object*. Moreover, in predicate logical propositions, we assume this order as well. So *hate*(j, m) expresses that John hates Mary, not that Mary hates John. The interpretation of such propositions goes as follows:

(54) $[\![R(x, y)]\!]^w = 1$ if and only if $\langle [\![x]\!]^w, [\![y]\!]^w \rangle \in [\![R]\!]^w$

Ternary predicates correspond to sets of triples.

6.3. Interpreting quantification

For completeness, here is the way quantification is interpreted in predicate logic.

Say that φ is well-formed statement in predicate logic. We write $\varphi[v/d]$ for the proposition that results from substituting every occurrence of v in φ by d. For example, $boy(x) \wedge girl(y)[y/m]$ equals $boy(x) \wedge girl(m)$.

(55) $[\![\forall x[\varphi]]\!]^w = 1$ if and only if for every value d: $[\![\varphi[x/d]]\!]^w$=1

(56) $[\![\exists x[\varphi]]\!]^w = 1$ if and only if for some value d: $[\![\varphi[x/d]]\!]^w$=1

7 Entailment in predicate logic

Given the interpretation of predicate logical expressions, we can derive *all* entailment relations between sentences of the logical language. For instance, we know that (57-a) entails (57-b) (indicated by the logical notation for entailment $\models$).

(57) a. $boy(j) \wedge lazy(j)$
b. $\models lazy(j)$

How do we know this? The truth table for $\wedge$ says that a conjunction is true only if both conjuncts are true. So, in all cases in which (57-a) is true, both $boy(j)$ and $lazy(j)$ will be true, hence the entailment to (57-b). Another example is the following: (58-a) entails (58-b).

(58) a. $\exists x[sweater(x) \wedge blue(x) \wedge own(j, x)]$
b. $\models \exists x[sweater(x) \wedge own(j, x)]$

This entailment holds since (58-a) is true only if we can find some object, call it d, such that the conjunction $sweater(d) \wedge blue(d) \wedge own(j, d)$ is true. Given the semantics of $\wedge$ this means that in those cases $sweater(d) \wedge own(j, d)$ is true. Consequently, there exists a value for x such that $sweater(x) \wedge own(j, x)$ (namely whatever d was), and so $\exists x[sweater(x) \wedge own(j, x)]$ is true.

8 An example of using predicate logic: scope

In this section, we will encounter a few examples of how predicate logic can be used to be precise about the meaning of natural language examples. In particular, we will look at how we can use predicate logic to distinguish between different readings due to quantifier scope.

In logic, the scope of operators and connectives is unambiguously given by a formula. For instance, in (59-a) the universal quantifier is in the scope of negation, whereas in (59-b) the scopal relation between negation and quantifier is reversed.

(59) a. $\neg\forall x[glitters(x) \rightarrow gold(x)]$
b. $\forall x[glitters(x) \rightarrow \neg gold(x)]$

The form in (59-a) corresponds to the English sentence (60). It says that glittering objects are not automatically made of gold. The form in (59-b) corresponds to the Dutch sentence (61). It says something completely different, namely that glittering objects are not made of gold. That is, (61) expresses that no glittering object is made of gold.[2]

[2]By the way, the Dutch example in (61) *can* express (59-a), but only with a very specific intonation pattern.

(60) *All that glitters is not gold.*

not > all
$\neg\forall x[glitter(x) \rightarrow gold(x)]$

(61) *Alles wat glinstert is niet van goud.*
All what glitters is not of gold
'*Nothing that glitters is gold*'
all > not
$\forall x[glitter(x) \rightarrow \neg gold(x)]$

What these examples show is that scope relations are not trivial. Part of the task of semantic analysis is to understand how scopal relations are expressed in a language. In other words, we would like to know how given a specific syntactic configuration a specific compositional interpretation comes about, in particular how the observed scopal relations are realised.

Give predicate logical representations for the meanings of the following examples.

(62) a. John didn't snore.
b. It is not the case that John snored.

(63) If a relative of mine dies, I will inherit a fortune

(64) What do you think is the scopal relation in the most natural interpretation of the following sentences? (The examples in (d-f) are from a study in VanLehn, K. (1978). *Determining the scope of English quantifiers*. Cambridge, MA: M.I.T. Artificial Intelligence laboratory technical report 483. M. S. Dissertation.) Can you give a predicate logical representation of the interpretations?

a. A letter informed me that every student of mine complained.
b. A letter was sent informing every student about their result.
c. A letter informed the dean about each student's results.
d. At the conference yesterday, I managed to talk to a guy who is representing each raw rubber producer in Brazil
e. At the conference yesterday, I managed to talk to a guy representing each raw rubber producer in Brazil
f. At the conference yesterday, I managed to talk to a representative from each raw rubber producer in Brazil

9 Further reading

Hodges (2001) is a good and accessible introduction to elementary logic. Chapter 3 and 4 of de Swart (1998) contain a detailed and gentle introduction to predicate logic. For a somewhat less accessible but very rigorous overview of set theory, logic and other mathematical aspects of formal semantics, see various chapters in Partee et al. (1993).

Interestingly, Heim and Kratzer (1998) does not contain any predicate logic. Heim and Kratzer sidestep predicate logic by doing everything in set theory. More on this in the next handout.

References

de Swart, H. (1998). *Introduction to Natural Language Semantics*. CSLI Publications.

Heim, I. and A. Kratzer (1998). *Semantics in Generative Grammar*. Blackwell Publishing.

Hodges, W. (2001). *Logic: an introduction to elementary logic* (Second Edition ed.). Penguin Books Ltd.

Partee, B., A. ter Meulen, and R. Wall (1993). *Mathematical methods in linguistics*. Kluwer Academic Publishers.