Chapter 3: Sentences and Determiner Phrases

In this chapter we will begin to develop rules for assigning truth conditions to sentences of English. After setting up the outlines of the approach we will take, this chapter will focus on Determiner Phrases and sentential conjunction.

3.1 The Syntax

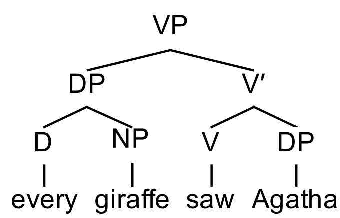
As discussed in the introduction, our aim is a grammar that assigns truth conditions to sentences based on the meanings of the parts of the sentence. For this to work, one needs to know what the parts of sentences are and how they are combined. So we first need to fix ideas about syntax.

We will be adopting the view that the grammar of a natural language generates phrase structures, representable as trees, and then movement transformations can apply to these to create new phrase structures. We will take for granted some means by which morphemes are combined into words; for example, '*see*+Past' becomes '*saw*'.

Many of our assumptions about syntax are motivated by Lasnik's (1995) "The forms of sentences". We review some of them now.

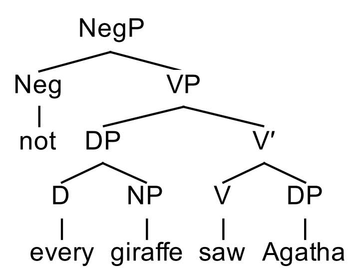
3.2 Phrase Structures for this Chapter

We'll be adopting the idea that sentential subjects are generated as specifiers in a VP, giving us structures like the one below:



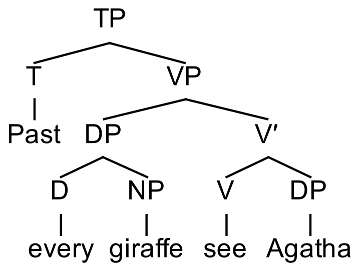
Tree 1: deep structure of '*Every giraffe saw Agatha'*

This VP is the basis for a sentence. It can be extended with various functional categories. For example, there can be a negation above the VP:

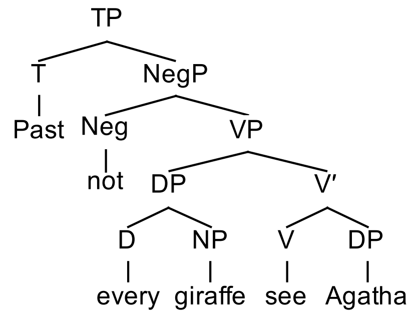


Tree 2: deep structure of '*Every giraffe didn't see Agatha'*

Another possibility is to extend a VP or a NegP into a Tense Phrase (TP):

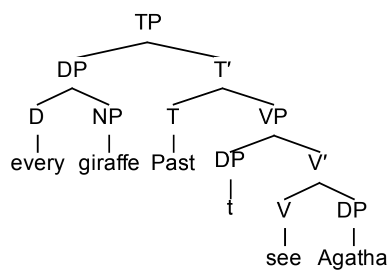


Tree 3: deep structure with tense of ‘*Every giraffe saw Agatha’*



Tree 4: deep structure with tense of ‘*Every giraffe did not see Agatha’*

These trees will be the input to rules of interpretation. However, the trees are subjected to further syntactic and morphological rules that produce the surface forms (notice that what is given in the tree does not match what we actually pronounce!). In Tree 3, the subject will raise to the specifier of TP, giving us:



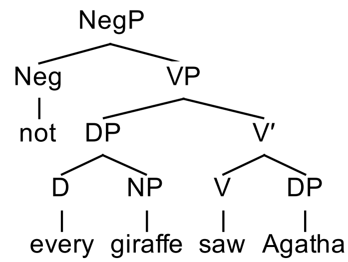
Tree 5: Subject Raising

We’ll refer to this as “Subject Raising”. The tense will lower onto the verb and combine with it to form *saw*. We’ll call that “Tense Lowering”. The result is the sentence: ‘*Every giraffe saw Agatha’.*

Subject Raising will also apply in Tree 4. However, the negation will block Tense Lowering. This forces the insertion of the dummy verb *do* (often called an ‘auxiliary’) which then combines with the Past, giving us: ‘*Every giraffe didn’t see Agatha.*’ This insertion operation is called “*Do*-Support”.

It is important to keep these last transformations (Subject Raising, Tense Lowering and *Do* Support) in mind, because our data, the expressions that we have intuitions about, are **surface structures**. To test our theory, we need to know what surface string corresponds to which syntactic structure. However, for the purposes of interpretation, we don’t need to go further than the structures that precede these transformations. We will call the structures that precede those transformations **deep structures**.

In a subsequent chapter, we will discuss the semantics of the tenses. Until then, our deep structures will be VPs and NegPs and we'll assume that tense is on the verb from the start. This means, for example, that in developing the semantics for the sentence '*Every giraffe didn't see Agatha*' we begin with the deep structure given in Tree 2, repeated below:



Tree 2: deep structure of '*Every giraffe didn't see Agatha'*

3.3 Direct and Indirect Interpretation

There are two different ways of assigning truth conditions to sentences of natural language. The first way is called **direct interpretation**. On that method, meanings are assigned to words of the language and then semantic rules govern how those meanings are combined. On this method, one may apply techniques like those we used in the previous chapter to interpret our symbolic language. There is another method, which is called **indirect interpretation**. On that method, a natural language such as English is translated into a symbolic language and then the sentences of the symbolic language are assigned truth conditions. For example, to interpret the sentence '*Bruce sings*' we first appeal to **rules of translation** that get us to the sentence 'Sb'. Then our semantic rules for the symbolic language assign truth conditions to 'Sb' and we count those conditions as the truth conditions for '*Bruce sings*'. There is a diagram on the next page that illustrates this process. The method is called indirect interpretation because the truth conditions are assigned to sentences of natural language indirectly, via an intermediate translation language. In this textbook, we will be doing indirect interpretation.

Simple sentences like '*Bruce sings*' are useful to illustrate what indirect interpretation is. But the potential advantages of doing indirect interpretation rather than direct interpretation are only realized once we start to analyze more complex sentences. So for now, you should understand how truth conditions get assigned to sentences of English on the indirect method, without necessarily seeing why one would choose this method.

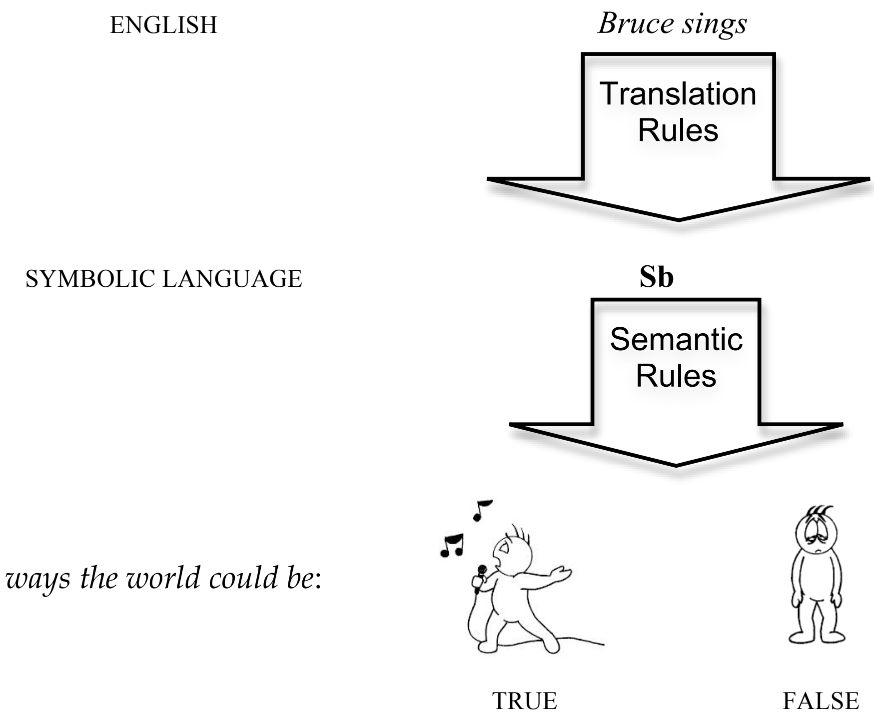


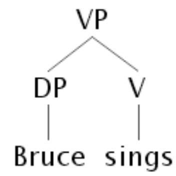
Figure 1: *Indirect Interpretation*

Above we said that 'Sb' might serve as the translation for '*Bruce sings*' on the indirect approach. In symbolic logic there is an arbitrary connection between the form of a predicate or individual constant and its meaning. One lexicon may assign to 'S' the set of all individuals who sing and another lexicon might assign it the set of all rabbits. This degree of freedom is not useful for the purpose of assigning meanings to natural language. When we translate '*sings*' we want to be sure that the translation picks out the singers and not the rabbits. Likewise, when we translate '*Bruce*' we want the translation to be a constant that gets assigned Bruce and not Agatha.

To guarantee these results we make two stipulations about our translation language. First, we'll expand the set of symbols so their <u>form</u> corresponds to English words. In particular, we'll take words with primes in bolded font to be expressions of the symbolic language. So now, **sings**, **rabbit** and **blue** are all one-place predicates of the symbolic language. **loves**, **contains** and **sees** are two-place predicates. And **bruce**, **agatha**, **jack** are individual constants. The second thing we will do is identify a lexicon, which we'll call *E*, which assigns to predicates and constants the extensions of the corresponding English words. For example:

E(**sings**) = {z | z sings}	*E*(**blue**) = {z | z is blue}
E(**rabbit**) = {x | x is a rabbit}	*E*(**bruce**) = Bruce
E(**loves**) = the loving relation.	*E*(**agatha**) = Agatha.

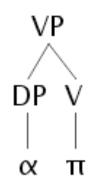
Now instead of 'Sb', we want our rules of translation to take us from '*Bruce sings*' to the atomic symbolic sentence '**sings(bruce)**'. Since the rules of translation are part of the system that assigns meanings, we must make sure that those rules are sensitive to syntax, for the reasons emphasized in the beginning of the course. That means we will be translating syntactic structures like:



To do that, we'll want a rule that is general, one that applies to any sentence with this structure. In order to state a general rule, we need metavariables that range over expressions of English like '*Bruce*' and '*sings*'. We'll use α and β to range over names and we will use π to range over predicates of English (transitive and intransitive verbs, adjectives, nouns, prepositions).[1] Finally, we adopt the convention that just as the logical symbol '**sings**' is the translation of the English word '*sings*', so the bolded $\boldsymbol{\pi}$ stands for the translation of π, bolded $\boldsymbol{\alpha}$ for the translation of α and bolded $\boldsymbol{\beta}$ for β. Here then is the rule for simple intransitive clauses. Note that it refers to the translation as the "logical form":

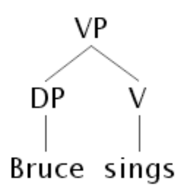
INTRANSITIVE

Anything of the form



has the logical form: $\boldsymbol{\pi}(\boldsymbol{\alpha})$

Instantiating this rule, we get the structure below:



has the logical form '**sings(bruce)**'. And now, by **ATOMIC-1** we can use that to get the truth conditions of the English sentence:

(1) '**sings**(**bruce**)' is true with respect to E *iff* E(**bruce**) $\in$ E(**sings**)

[1] In our trees we have category labels above lexical items. It's the lexical item with its label that 'matches' the metavariable.

Given how the lexicon *E* was defined above, we have that:

(2) ‘**sings**’(**bruce**)’ is true with respect to *E iff* Bruce sings.

We’ve now made a connection between the VP structure above and truth conditions. So we have a system of rules that altogether assigns truth conditions to English sentences. Here’s the diagram from above updated with our new symbols in the translation language:

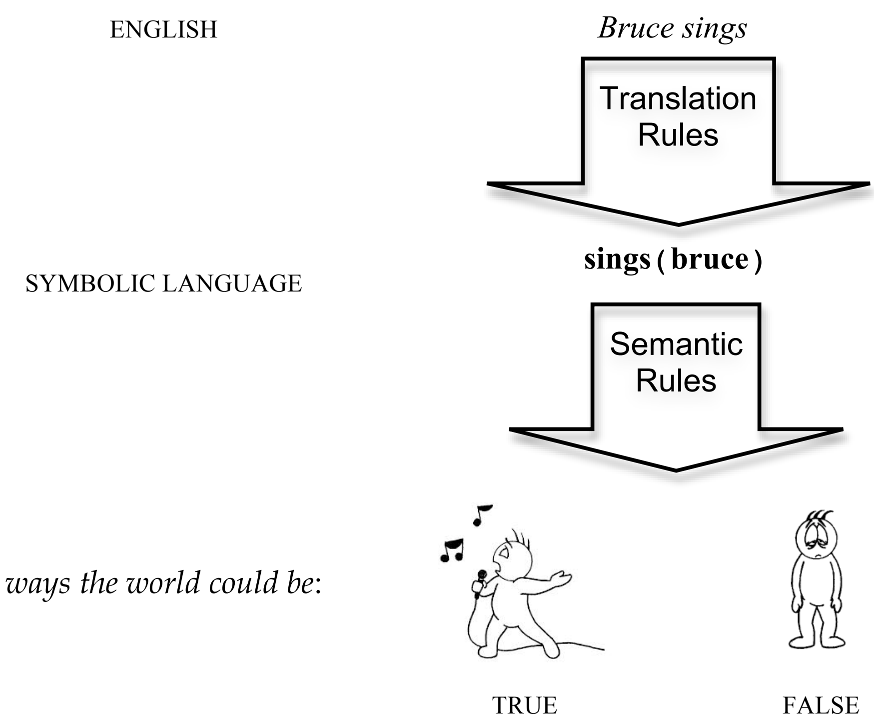
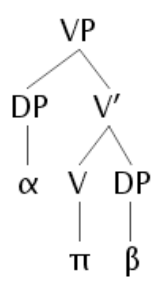


Figure 3
Indirect Interpretation (with updated symbols)

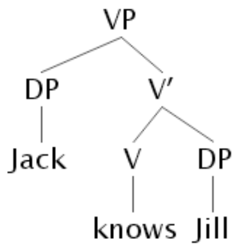
To interpret simple transitive clauses, we’ll adopt this rule:

TRANSITIVE
Anything of the form



has the logical form: $\boldsymbol{\pi}(\boldsymbol{\alpha}, \boldsymbol{\beta})$.

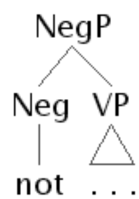
By that rule, the structure below:



has the logical form: **knows**(**jack**, **jill**), which means that that tree has the truth conditions that are met *iff* Jack bears the know-relation to Jill.

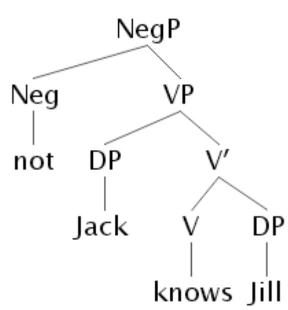
Finally, we'll introduce a translation rule for negation.

NEGATION
Anything of the form



has the logical form: $\neg\,\phi$
where ϕ is the logical form of the VP

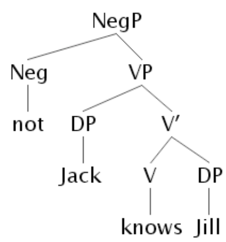
Putting together our last two rules, we have that:



has the logical form: ¬**knows**(**jack**, **jill**). And its logical form is true with respect to *E iff* it is not the case that Jack knows Jill. So those are the truth conditions we assign to the deep structure tree above. And the surface structure associated with that tree gets us to the sentence '*Jack doesn't know Jill*'. So overall, our grammar says that '*Jack doesn't know Jill*' is true *iff* it is not the case that Jack knows Jill.

We are calling the symbolic logic sentence associated with a sentence of English its logical form. In keeping with common parlance, we will call the syntactic structure to which the logical form is assigned **the LF**. For the sentences discussed in this section, the LFs were just the deep structure, the structure that is generated for the sentence prior to Subject Raising, Tense Lowering or *Do*-Support. In the next section, we will be introduced to a transformation that will apply in the course of producing an LF.

Here's a summary for a particular sentence of the terminology introduced:

	$\neg$ **knows**(**jack**, **jill**)	*Jack doesn't know Jill*
LF	logical form	surface string

EXERCISE A

For each sentence below, give its LF structure and its logical form.

(i) Jack doesn't dance.
(ii) Agatha owns Fido.

3.4 DPs, Scope, and Logical Form

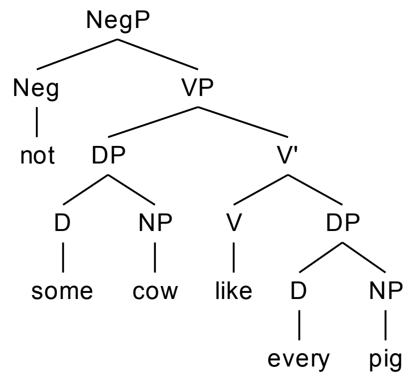
We turn now to the interpretation of sentences with quantificational DPs, such as:

(3) Some cow doesn't like every pig.

This sentence is ambiguous. Some of its readings are captured in the paraphrases below:

It is not the case that there is some cow such that it likes every pig.
Every pig is such that there is some cow that doesn't like it.
There is some cow such that: it is not the case that it likes every pig.
There is some cow such that: it dislikes every pig.

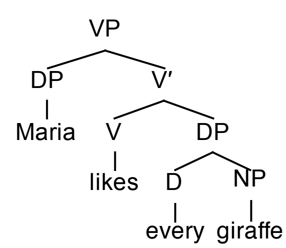
The fact that (3) has several readings means that we need a mechanism that associates the string '*some cow doesn't like every pig*' with different sets of truth conditions. As demonstrated in the previous section, the logical forms we assign are associated by rule with a syntactic structure. And yet, although (3) has multiple meanings, given our syntactic assumptions to this point, there is only one phrase structure for the sentence:



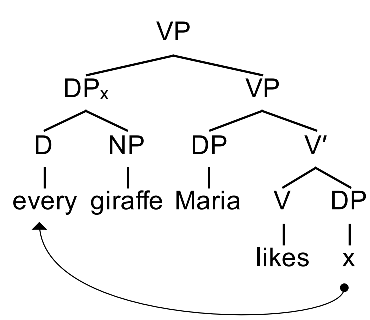
Observe that the various paraphrases above differ by the relative order of the quantifiers and negation. This leads to the idea that the English sentence comes to have multiple meanings because the surface form in fact corresponds to more than one syntactic structure and these different syntactic structures differ by the relative order of quantifiers and negation. This idea can be implemented by means of a movement transformation called "Quantifier Raising" or simply "QR".

> **Quantifier Raising (QR)**: A DP consisting of a determiner and an NP may be adjoined to a node that dominates it. The place it is moved from is filled by a variable, and the moved DP is indexed with that variable.

If we apply QR to the object DP in the phrase structure below:

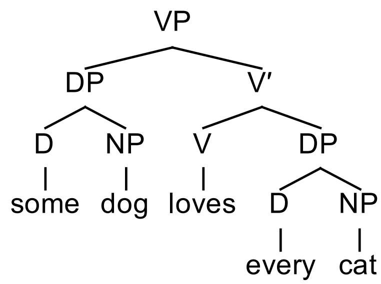


we get:

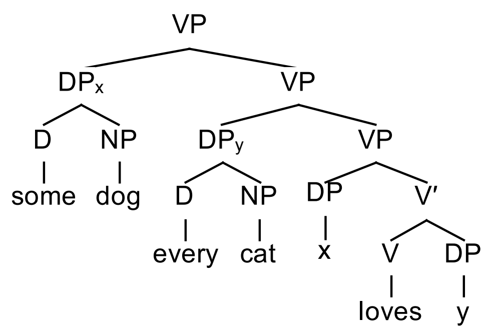


The variable that is left behind is called a "trace". Traces are often written using the letter 't' with a subscript, such as 't_3', but we will use letters such as 'x', 'y', 'z', to simplify the production of logical forms.

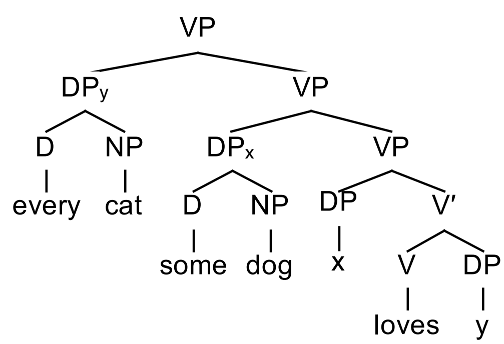
The structure below has two complex DPs, each of which can raise by QR.



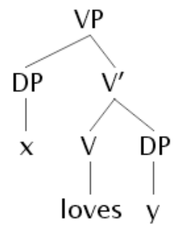
If the object *every cat* is raised, followed by the raising of the subject '*some dog'*, we get:



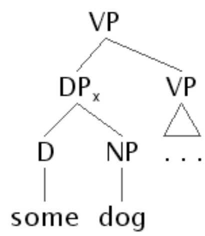
If we raise the subject first and then the object we get:



In order to translate these trees into sentence of the symbolic language, there are two tasks before us. We'll need a rule that allows for DPs like '*every cat*'. And we'll also need rules that interpret VPs that have variables in them. This second task is easy to take care of by allowing our metavariables α and β to range over variables, and by taking the translation associated with a variable to be the variable itself. In that case, according to the TRANSITIVE rule given earlier, '**loves**(x, y)' is the logical form of:



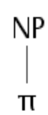
Next we turn to the translation of structures that include quantificational DPs. Suppose we have an adjoined DP in the following configuration:



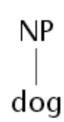
and suppose that the logical form associated with the lower VP is ψ. Then we want the logical form of the whole structure to be:

$$\mathbf{some}_x\{\mathbf{dog}(x)\}\ \psi$$

To achieve that, we'll introduce two rules, one that gives the translation for the NP and the other that gives the translation of the entire VP. Here goes:

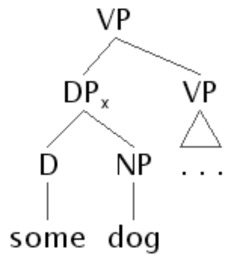
> **NON-BRANCHING NP**
>
> Anything of the form:
>
> 
>
>
> has the logical form: $\pi(u)$
>
> where 'u' corresponds to the variable subscripted on the DP that immediately contains NP.

That means that in the tree above, '**dog(**x)' is the logical form of:



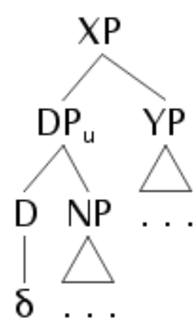
In the structure above, the [$_{NP}$ *dog*] is contained in the DP which has a subscripted 'x'. That is why in the logical form, the predicate **dog** is followed by '(x)'. The rule says 'immediately contained' to exclude looking higher to a DP that might also contain the NP (e.g. in [DP the man who owns [DP some dog]]).

Next, we need a rule that will interpret a VP with a DP that has undergone QR, like the one above repeated here:



Key features of the tree are: the choice of determiner, the two phrases that combine with the determiner, NP, VP and the index on the determiner. All of these features will have to figure in our rule. Since the rule is meant to be general, we'll use the metavariable '*u*' to stand for any variable and we'll use 'δ' (delta) as a metavariable over determiners and the bolded version for their translation. In general, we take a determiner to translate as the corresponding generalized quantifier. Finally, in the tree above, the node above DP happens to be VP. But that won't always be the case. It all depends on where the DP raises and adjoins. So the rule will use 'XP' and 'YP' to mean phrases of any category. Here is the rule:

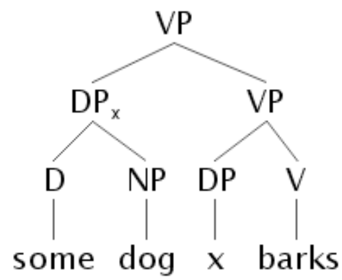
BRANCHING DP:
Anything of the form:

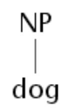
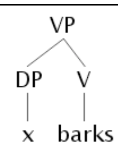


has the logical form: $\boldsymbol{\delta}_u\{\phi\}\ \psi$

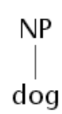
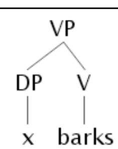
ϕ is the logical form assigned to the NP
ψ is the logical form assigned to the YP

Let's consider how we instantiate this rule with this tree:



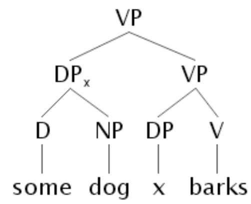
Expression in the tree	What it instantiates in the rule
‘*some*’	δ
	
x	u
	

From our previous rules **NON-BRANCHING NP** and **INTRANSITIVE** and we know that:

The logical form of:	is:
	**dog**(x)
	**barks**(x)

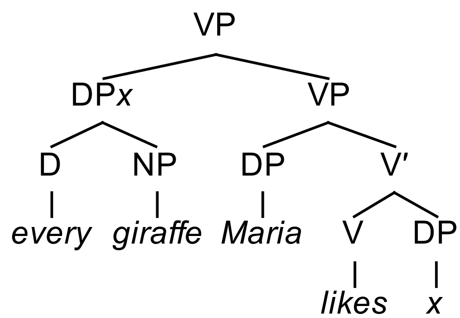
Putting this together, our BRANCHING DP rule tells us that:

The logical form of: is:



some$_x$\{**dog**(x)\} **barks(**x)

Consider now the LF structure generated for ‘Maria likes every giraffe’:



Given the TRANSITIVES rule, the logical form of the lower VP is:

likes(**maria**, x)

Given the NON-BRANCHING NP rule, the logical form of the NP is:

giraffe(x)

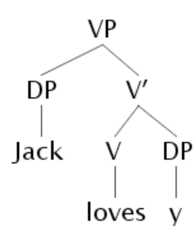
and given the BRANCHING DP rule, the logical form of the whole sentence is:

every$_x$\{**giraffe**(x)\} **likes(maria,** x)

That logical form is true with respect to the lexicon *E iff* for every object o such that o is a giraffe, Maria likes o. So those are the truth conditions we assign to ‘*Maria likes every giraffe*’.

EXERCISE B

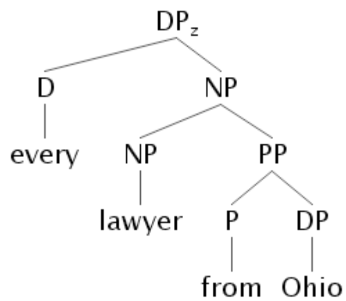
Provide the logical form for the tree:



Which rule or rules of translation did you use to arrive at that logical form?

EXERCISE C

There are two NP nodes in the tree below. The Non-Branching NP rule only applies to one of them. Provide the logical form assigned to that NP.



EXERCISE D

You may have noticed that our rules of interpretation, proposed in the previous chapter, don't apply to $\mathbf{every}_x\{\mathbf{giraffe}(x)\}\ \mathbf{likes}(\mathbf{maria}, x)$.

i. Explain why not
ii. Devise a fragment of English that would interpret the logical form above.
iii. Update your fragment with all the rules discussed in the previous chapter.

✪ IMPORTANT ✪

For iii, check that your updated fragment still makes the correct predictions for $\mathbf{every}_x\{\mathbf{giraffe}(x)\}\ \text{likes}(\mathbf{maria}, x)$ and all other logical forms discussed in this chapter.

Earlier, we arrived at two LF-structures for the sentence '*some dog loves every cat*'. Verify that the rules presented so far associate the logical forms below with those two structures in (4) and (5):

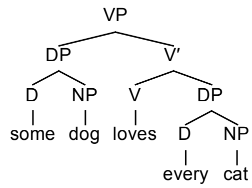
(4) $\mathbf{some}_x\{\mathbf{dog}(x)\}\ \mathbf{every}_y\{\mathbf{cat}(y)\}\ \mathbf{loves}(x, y)$

(5) $\mathbf{every}_y\{\mathbf{cat}(y)\}\ \mathbf{some}_x\{\mathbf{dog}(x)\}\ \mathbf{loves}(x, y)$

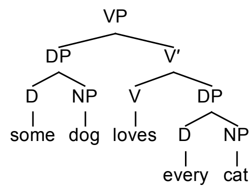
These two logical forms are assigned distinct truth conditions relative to the lexicon *E*. The first requires that there be at least one dog that loves every cat. The second requires that for every cat there be at least one dog that loves that cat.

Here is a summary of how we account for the ambiguity in the English sentence '*some dog loves every cat*'.

Deep Structure:



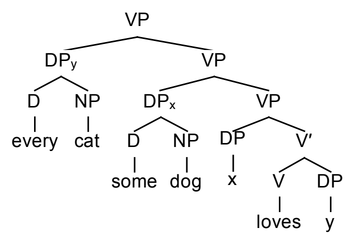
Deep Structure is transformed into Surface Structure. In this case nothing interesting happens:



Surface Structure determines the string: '*some dog loves every cat*'.

Turning now to interpretation:

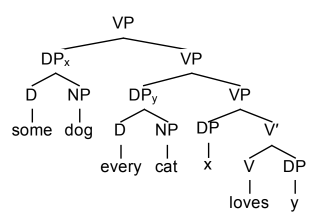
Deep Structure leads to an LF via QR:



That LF translates into the logical form:

(6) **every**$_y${**cat**(y)} **some**$_x${**dog**(x)} **loves**(x, y)

Deep Structure leads to an LF via QR:



That LF translates into the logical form:

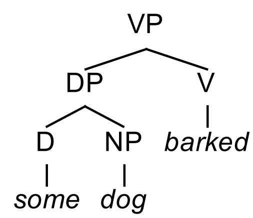
(7) **some**$_x${**dog**(x)} **every**$_y${**cat**(y)} **loves**(x, y)

The net result is one surface structure associated with two distinct logical forms.

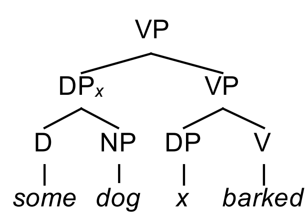
EXERCISE E

1. Describe a situation in which '*some dog loves every cat*' is true on one reading and false on the other.
2. Provide a deep structure for the sentence '*Jack doesn't know every girl*'.
3. The top node in your deep structure in question 2 should be labeled NegP. Create an LF from your deep structure, by applying QR to the DP '*every girl*' adjoining it at the top to NegP.
4. Your LF in question 3 should include within it a subtree whose top node is VP. Provide the logical form assigned to the VP by the TRANSITIVES rule.
5. The translation rule NEGATION applies to a part of your tree in question 4. What logical form does it assign?
6. What logical form is assigned by the BRANCHING DP rule to the LF you supplied in question 3?

The rule of QR is optional. A DP is not required to move, nor is there any particular node to which it must move. However, it turns out that every DP that is neither a name nor a variable must QR at some point. Consider the following structure:



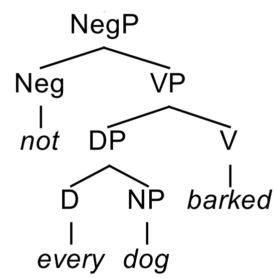
None of the rules for associating phrase structures with logical forms will apply here. We have rules for simple VP's containing just names and variables and they don't apply. And we have a rule for complex DPs but it relies on having a variable subscript on the DP. If we stop at this point, we make the incorrect prediction that '*some dog barked*' has no truth conditions! But we don't need to stop. We can perform QR, adjoining the subject to the VP:



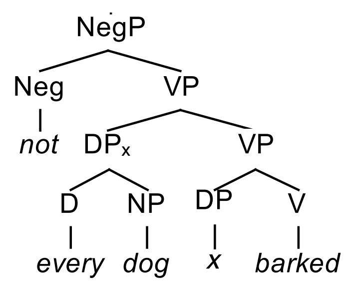
And now our rules lead to the logical form: **some**$_x$ {**dog**(x)} **barked**(x).

Above we showed how the current system correctly predicts that a sentence with two complex quantifiers can have two different logical forms. In other words, we accounted for one type of **scopal ambiguity**. Now, let's see how ambiguity arises when a given structure contains a quantificational DP and negation.

Let's begin with the phrase structure for '*every dog didn't bark*' before Subject Raising and *Do*-Support:



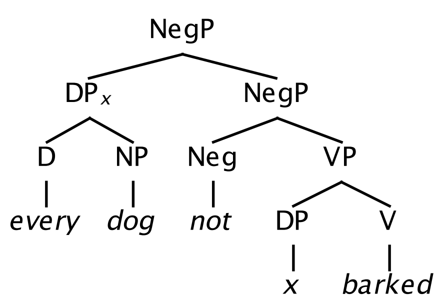
Let's adjoin the DP to the VP like we did above for '*some dog barked*':



Using the NEGATION rule from the previous section and the rules used for the previous example we get:

(8) ¬ $\mathbf{every}_{x}\{\mathbf{dog}(x)\}$ $\mathbf{barked}(x)$.

Next, we start from the original phrase structure (pre-QR) and this time we raise the DP to adjoin to NegP:



This gives rises to a different logical form:

(9) $\mathbf{every}_{x}\{\mathbf{dog}(x)\}\neg\ \mathbf{barked}(x)$.

EXERCISE F:
For the sentences in (a)-(c) below:
(i) Provide the deep structure.
(ii) Provide a structure that results from applying QR in the deep structure you provided in (i).
(iii) Provide the logical form that our rules associate with the structure you gave in (ii).

(a) Maria doesn't like every cat.
(b) No cat likes Maria.
(c) Some cat didn't scratch Fido, (some squirrel did).

If there is more than one way to complete the task in (ii) and these lead to truth conditionally distinct logical forms – indicate that by writing next to your syntactic structure "there is another truth conditionally distinct logical form". If you have an intuition about which reading is more salient for you, indicate that as well (a reading is more salient if it comes to mind more naturally, as opposed to requiring some work to get it).

✪ NOTE ✪
When doing (i)-(iii) for the sentence in (c), do not include the part in parenthesis. Also remember that we are assuming that there is no NegP in the structure of (b).

The rule of QR is optional and unconstrained. This means that at present, according to our rules, whenever an English sentence contains two or more quantificational DP's it is ambiguous. There are several ways in which this picture is incomplete. First, there is an interesting body of work showing that movement operations are constrained syntactically. Since QR is a movement operation, these constraints translate into constraints on possible readings of sentences containing quantifiers. Next, there are lexical items that favor one scope over the other. For example, the symbolic language sentence in (10) is best translated as '*Fido didn't see any cat'* using the expression '*any*', which is called a **negative polarity item** because it likes to be in the scope of negation.

(10) $\neg$ $\mathbf{atlst1}_x\{\mathbf{cat}(x)\}$ $\mathbf{see}(\mathbf{fido}, x)$

If instead of '*any*' we use '*some*', i.e. *Fido didn't see some cat,* the preferred interpretation is:

(11) $\mathbf{atlst1}_x\{\mathbf{cat}(x)\}$ $\neg$ $\mathbf{see}(\mathbf{fido}, x)$

In general, when *some* is used with a negation, it tends to have wide scope over it. '*each*' is another determiner that forms DPs that have a preference for wide-scope.

Another constraint on scope relations has to do with ellipsis. Consider the sentence:

(12) A dog chased every cat, and a donkey did too.

This surface form is generally thought to be produced by VP ellipsis from:

(13) A dog chased every cat, and a donkey chased every cat too.

Since this is a conjunction of two sentences, each of which contains two DP's, it is theoretically four ways ambiguous because there are two possible scope orderings within each conjunct. However, the sentence with ellipsis does not have these four readings; it has only two. It can be read with '*a dog*' and '*a donkey*' each taking scope over the direct object, or each taking scope under the direct object; mixed readings are not possible. This is apparently because ellipsis signals identity of interpretation between what is elided and what remains on the surface to tell you what is elided.

We've just reviewed some syntactic and lexical factors that constrain or limit scopal relations. But even when there are no grammatical constraints on scope, speakers often do not perceive ambiguity. The ambiguity is often resolved without noticing alternatives. What resolves it? One consideration is that there is a natural tendency for the quantifier phrase that occurs first in the surface structure to be interpreted as having widest scope, and if this makes sense of what the person is saying we interpret it that way and ignore other possibilities. Second, we tend to interpret people as saying things that are plausible. In most circumstances a sentence like:

(14) Some boy married every girl.

would be so implausible if we gave the subject phrase widest scope that this is not a salient option. So in this case, conditions of plausibility tend to overrule the principle of giving the subject widest scope.

These factors, surface order and plausibility, push towards a particular disambiguation. And since the disambiguation is automatic, we often don't perceive the ambiguity in the first place.

The following figure comes from page 79 of *Thinking, Fast and Slow* by Daniel Kahneman:

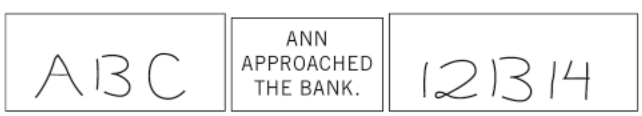


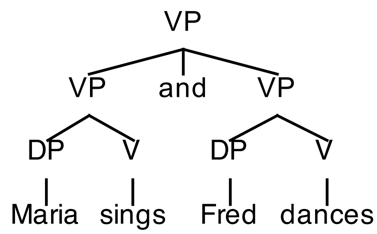
Figure 6

Before continuing, say aloud what you see in the figure.

More likely than not, when you looked at the figure and when you read aloud, you treated the first box as consisting of letters, the second of which was 'B'. And you saw the third box as containing numbers, the middle two being '13'. In fact, the same symbol was used in the middle of the two boxes. That symbol is ambiguous between 'B' and '13'. This experiment illustrates how context leads us to automatically disambiguate, in such a way that we are unaware of the ambiguity. The same thing can happen with scopal ambiguities.

3.5 Coordinating Conjunction

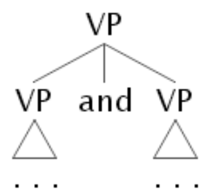
The conjunction '*and*' connects many types of expressions. We will focus here on its use as a sentence conjoiner. We'll also assume a simple syntax according to which a VP or a projection of VP can expand into 3 daughters, two VPs and a conjunction in the middle:



On the next page is a proposal for assigning logical forms to conjoined VPs:

AND

Anything of the form:



has the logical form: (ϕ & ψ)

where 'ϕ' is the logical form of the lefthand VP and 'ψ' is the logical form of the righthand VP.

So, given the rule above, and two applications of the transitive rule, the logical form associated with (15) would be (16):

(15) Maria likes Fred, and every teacher likes Maria

(16) (**likes**(**maria**, **fred**) & **every**$_x${**teacher**(x)} **likes**(x, **maria**))

EXERCISE G

1. The logical form above is in fact assigned to an LF structure. Provide that LF.
2. Provide the LF, the logical form and the truth conditions for the sentence '*Jack arrived and Jill cried*'.
3. Provide the LF, the logical form and the truth conditions for the sentence '*Jill cried and Jack arrived*'.
4. Review the **AND** rule, used to interpret symbolic logic sentences containing '&'. Is it possible to find a situation in which the truth conditions assigned to '*Jack arrived and Jill cried*' are met but the truth conditions assigned to '*Jill cried and Jack arrived*' are not met? If so, describe such a situation.

3.6 Narrative Discourse and the Semantics of '*and*'

When we propose a logical form for a sentence of English, we are stating a hypothesis about the perceived meaning and usage of the sentence. But this can be a fairly complicated matter to assess. In this section we will discuss one of these factors, illustrating it with respect to the proper analysis of '*and*'.

We consider first a well-known challenge to the accuracy of our analysis of '*and*'. It is often said that conjunctions with English '*and*' usually mean the same as conjunctions with symbolic logic '&', but that in some contexts an English conjunction has a meaning best captured using the phrase '*and then*'. The famous illustration is the contrast between (17) and (18) below:

(17) They got married and they had a baby.
(18) They had a baby and they got married.

Clearly, these sentences could be used to communicate the claim that two events happened *in a certain order*, a different order for each sentence, hence the contrast between them. There is nothing about order in the semantics of conjunctions with '&' so one might suggest that this shows that '*and*' conjunctions shouldn't be translated with '&' and that in general '*and'* conjunctions convey temporal succession. One problem with this idea is that '*and'* isn't always synonymous with *'and then'.* For example, if I utter (19) it does not seem like I am communicating the idea that she walked in and then *some time after that* he was sitting there. Instead, the tendency is to understand that he was sitting there when she walked in. So the data is a bit complicated.

(19) She walked in and Jack was sitting there.

A key difference between the baby examples (17)-(18) and the walking in example in (19) has to do with the kind of eventuality described in the second sentence. In (17) and in (18) the second sentence describes an event that occurs – a wedding or a birth. In (19), the second sentence describes a state that Jack is in. This contrast illustrates a general rule we follow in interpreting stories. The phenomenon is called ***narrative progression*** and the rule is stated below:

Narrative Progression
In the course of a narrative, when an event is described, it is usually understood to follow in time a previously mentioned event. When a state is described, it is usually understood to hold at the time of a previously mentioned event.

In (17), first a marriage is described and then a birth is described. The rule of Narrative Progression leads us to the understanding that the birth followed the marriage. In (18), first a birth is described and then a marriage is described. The rule of Narrative Progression leads us to the understanding that the marriage followed the birth. In (19), an entering is described and then a state of sitting is described. In this case, the rule of Narrative Progression leads us to the understanding that the sitting state is understood to hold throughout the time of the entering. For the event sentences, Narrative Progression amounts to the idea that the order in which events are described follows the order in which they occurred.

The following discourse serves as further illustration of Narrative Progression. As you read it, consider how you perceive the order of events:

1. *She walked in.*
2. *He was sitting there.*
3. *She looked around the room.*
4. *It was dark and gloomy.*
5. *There was nothing to lighten the mood.*
6. *He got up saying "I'm glad you're here."*
7. *She replied "I'm glad I could come."*
8. *Her reply didn't sound sincere to him.*
9. *It didn't sound sincere to her either.*

Each of these sentences tells us what is going on at a certain past time, sometimes called a "*reference time*". Most people interpret the reference times as advancing from earlier to later as the narrative progresses. But not every sentence in the narrative triggers such an advance. Normally, sentences 3, 6, and 7 are interpreted as advancing the reference time, and sentences 2, 4, 5, 8, and 9 are interpreted as leaving it unchanged. In other words, if someone were to claim that this story were a completely true one, then one would expect there to be four successive past times such that

at time 1, she walks in
at time 1, he is sitting there

at time 2, she looks around the room
at time 2, it is dark and gloomy
at time 2, there is nothing to lighten the mood

at time 3, he gets up and says "I'm glad you're here"

at time 4, she replies "I'm glad I could come"
at time 4, it doesn't sound sincere to him
at time 4, it doesn't sound sincere to her

The sentences that advance the reference time, 3, 6 and 7 describe events and the ones that don't advance the reference time, 2,4,5,8 and 9 describe states. The Narrative Progression rule makes reference to 'a previously mentioned event'. In our implementation, we understood the events to follow, and the states to hold, at the time of the <u>immediately</u> previously mentioned events. The rule doesn't require this and in more elaborate discourses it may not work this way. Also, above we said that the state of being dark held at the time of looking around the room (time 2). Most likely it held also at the time of entering (time 1). States can last a while and the rule only requires that they at least held at the time of one previously mentioned event.

Once we recognize the phenomenon of Narrative Progression, we have a new way to understand the contrast between (17) and (18):

(17) They got married and they had a baby
(18) They had a baby and they got married

In both sentences, two events are reported. In both cases, the events are understood to have occurred in the order in which they are reported. This can be attributed to the Narrative Progression rule. In that case, our semantics of conjunction can remain as originally proposed in the previous section. This way of viewing things also explains why the *and-then* effect is absent in (19):

(19) She walked in and Jack was sitting there.

The second conjunct of (19) is stative so by the rule of Narrative Progression, it should not introduce a later reference time.

This analysis fits into a picture of language according to which there are rules of grammar that assign truth conditions to sentences of the language. These come under the narrow heading of semantics. On top of that, there are conventions governing the use of language which may allow a speaker to convey more information than is captured by truth conditions alone. The rule of Narrative Progression is one such example. It governs how we interpret sequencing of sentences. These conventions come under the heading of ***pragmatics***.

Separating the data in this way, as opposed to revising our truth conditions for conjunction, also helps to explain the 'softness' of the effect. A speaker can explicitly cancel the effect. So it is felicitous to say:

(20) He rented an apartment and he found a job, but not necessarily in that order.

Here the speaker is removing the effect of a pragmatic rule with the *but*-clause. There is a strong contrast between this and an attempt by the speaker to deny or question an entailment. This sounds contradictory:

(21) He rented an apartment and he found a job, though I'm not sure if he found a job.

We began this section with the observation that a logical form for a sentence of English is a hypothesis about the perceived meaning of the sentence and that such a hypothesis can be a fairly complicated matter to assess. Our discussion of '*and*' has revealed an important source of complication: pragmatics. When considering what to include in our account of truth conditions, we need to be alert to the possibility of a pragmatic contribution that needs to be separated out before capturing the truth conditions. A convincing hypothesis about the pragmatics will lead us to adopt a simpler semantics and hopefully a more correct account of our intuitions about meaning.

EXERCISE H

Consider the following bit of narrative:

a. The concert was fantastic yesterday.
b. John played the bass.
c. Susan played the saxophone.
d. At the end of the concert, there was a standing ovation.

It is natural to understand the bass playing and the saxophone playing as simultaneous – which is not what is expected by our Narrative Progression rule. Formulate a hypothesis about what distinguishes this case from the ones covered in the text. Write an amended version of the rule of Narrative Progression that covers this case as well.

EXERCISE I

Devise a short narrative, and see if you can tell which parts advance the reference time and which do not. For each pair of adjacent sentences, see what happens when they are conjoined with '*and*'; does this make any difference to your understanding of the narrative? How does this bear on the question of what '*and*' means?

3.6 Summary

- **Indirect interpretation**

In this chapter, we set out a general format for interpreting sentences of English. We adopted the method of indirect interpretation. This means that we have rules that map English into a symbolic language. The sentences of that language are assigned truth conditions. So we indirectly assign to an English sentence the truth conditions of its symbolic logic translation.

- **Quantificational DPs**

We introduced the rule of QR along with rules for interpreting simple sentences with DPs in them and a rule for negation. The grammar now includes an account of scopal ambiguity.

- **Conjunction**

We have a way to interpret sentential conjunctions. The interpretation we gave treats conjunctions of the form 'ϕ *and* ψ' as truth conditionally equivalent to 'ψ *and* ϕ'. Intuitions to the contrary were explained in terms of the Narrative Progression rule.

You should be able to:

(i) Assign a deep structure, LF and logical form to sentences of English formed using: negation, transitive and intransitive verbs, conjunction and one or more DPs formed from a name or a determiner and a noun. Here are examples of such sentences:

Jack knows Jill
Jack doesn't know Jill.
Every farmer knows Jack.
At least one politician knows Jack and Jack admires every politician.

(ii) Show how our grammar assigns two truth conditionally distinct logical forms to sentences such as:

Every passenger didn't run.
At least one triangle touches every circle.

3.7 List of rules

Metavariables:

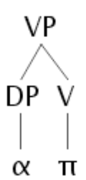
π	predicates (1-place and 2-place)
α, β,	variables and individual constants
u, δ, X, Y	variables
ϕ, ψ	sentences / formulas
M	lexicons

Syntactic Rule:

Quantifier Raising (QR): A DP consisting of a determiner and an NP may be adjoined to a node that dominates it. The place it is moved from is indicated by a variable, and the moved DP is indexed with that variable.

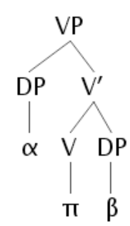
Rules of Translation:

INTRANSITIVE
Anything of the form



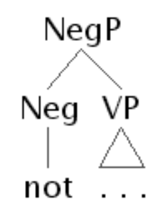
has the logical form:
$\boldsymbol{\pi(\alpha)}$

TRANSITIVE
Anything of the form



has the logical form: $\boldsymbol{\pi(\alpha, \beta)}$

NEGATION
Anything of the form



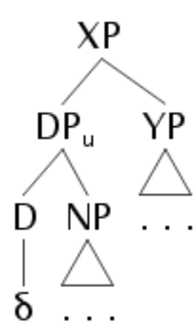
has the logical form: $\neg\phi$ where ϕ is the logical form of the VP

NON-BRANCHING NP
Anything of the form:

has the logical form:
$\boldsymbol{\pi}$(u)

where ‘u’ corresponds to the variable subscripted on the DP that immediately contains NP.

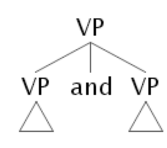
BRANCHING DP:
Anything of the form:



has the logical form: $\boldsymbol{\delta}_u\{\varphi\}\ \psi$

ϕ is the logical form associated with the NP
ψ is the logical form associated with YP

AND
Anything of the form:



has the logical form: (ϕ & ψ)

‘ϕ’ is the logical form of the lefthand VP

‘ψ’ is the logical form of the righthand VP.

Semantic Rules:

ATOMIC-1
'$\pi(\alpha)$' is true with respect to M *iff* $M(\alpha) \in M(\pi)$

ATOMIC-2
'$\pi(\alpha,\beta)$' is true with respect to M *iff* $M(\alpha)$ bears the relation $M(\pi)$ to $M(\beta)$

AND
'$(\phi$ & $\psi)$' is true with respect to M *iff* 'ϕ' is true with respect to M and 'ψ' is true with respect to M

NOT
'$\neg\phi$' is true with respect to M *iff* it is not the case that: 'ϕ' is true with respect to M

EXIST
'$\exists u\phi$' is true with respect to M *iff* there is at least one object o such that, 'ϕ' is true with respect to M+<u,o>

UNIVERSAL
'$\forall u\phi$' is true with respect to M *iff* every object o is such that 'ϕ' is true with respect to M+<u,o>

EVERY
'**every**$_u\{\phi\}$ ψ' is true with respect to M *iff* for every object o such that 'ϕ' is true with respect to M+<u,o> , 'ψ' is also true with respect to M+<u,o>

NO
'**no**$_u\{\phi\}$ ψ' is true with respect to M *iff* there is no object o such that 'ϕ' is true with respect to M+<u,o> and 'ψ' is also true with respect to M+<u,o>

THE
'**the**$_u\{\phi\}$ ψ' is true with respect to M *iff* there is exactly one object o such that 'ϕ' is true with respect to M+<u,o>, and 'ψ' is also true with respect to M+<u,o>

SOME
'**some**$_u\{\phi\}$ ψ' is true with respect to M *iff* for some object o such that 'ϕ' is true with respect to M+<u,o>, 'ψ' is also true with respect to M+<u,o>

ATLST1
'**atlst1**$_u\{\phi\}$ ψ' is true with respect to M *iff* there is at least one object o such that 'ϕ' is true with respect to M+<u,o> and 'ψ' is also true with respect to M+<u,o>

Pragmatic Rule:

Narrative Progression
In the course of a narrative, when an event is described, it is usually understood to follow in time a previously mentioned event. When a state is described, it is usually understood to hold at the time of a previously mentioned event.

Rules of Inference:

REPLACE [Rule of Inference]

If two statements ϕ and ψ are logically equivalent (ϕ *iff* ψ), then from a statement including ϕ, we can infer the statement that results from replacing ϕ with ψ.

If two expressions E1 and E2 name the same entity, then from a statement that includes E1, we can infer the statement that results from replacing E1 with E2.

INSTANTIATE [Rule of Inference]

From a rule stated in terms of metavariables, infer the result of substituting the metavariables with expressions of the right kind – taking care to substitute all occurrences of a given variable with the same expression.