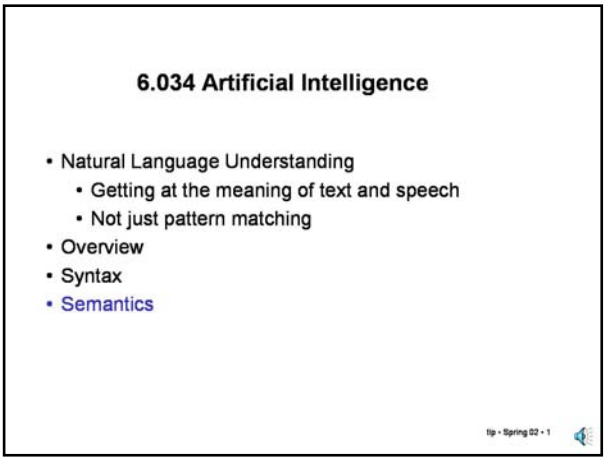
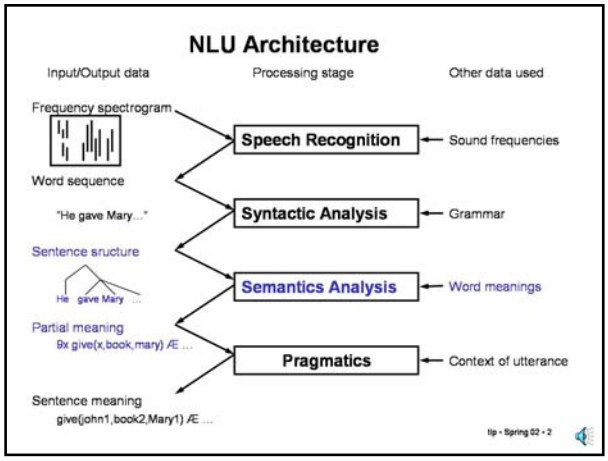
6.034 Artificial Intelligence. Copyright © 2004 by Massachusetts Institute of Technology.

6.034 Notes: Section 12.3

Slide 12.3.1

Now, we move to consider the semantics phase of processing natural language.





Slide 12.3.2

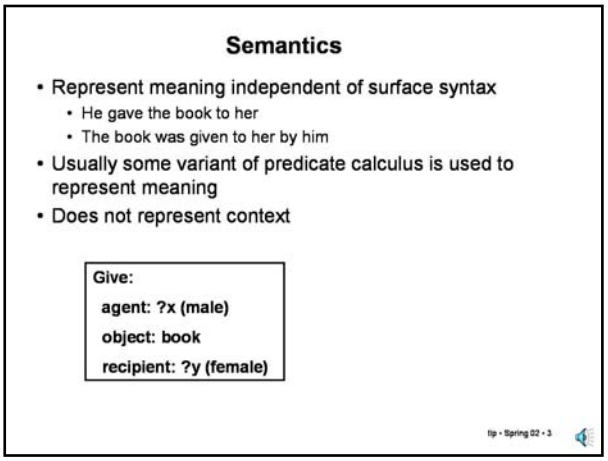
Recall that our goal is to take in the parse trees produced by syntactic analysis and produce a meaning representation.

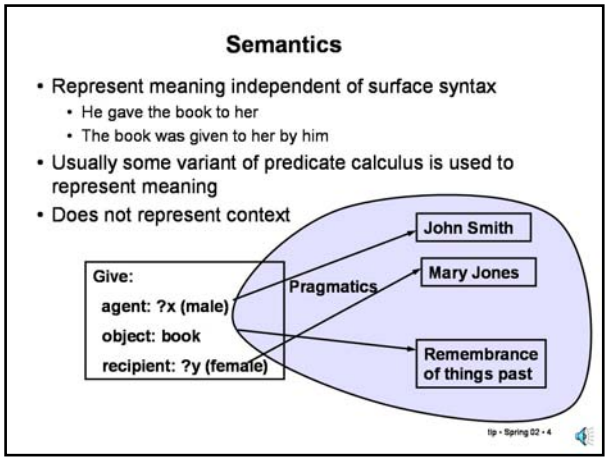
Slide 12.3.3

We want semantics to produce a representation that is somewhat independent of syntax. So, for example, we would like the equivalent active and passive voice versions of a sentence to produce equivalent semantics representations.

We will assume that the meaning representation is some variant of first order predicate logic. We will specify what type of variant later.

We have limited the scope of the role of semantics by ruling out context. So, for example, given the sentence "He gave her the book", we will be happy with indicating that some male gave the book to some female, without identifying who these people might be.





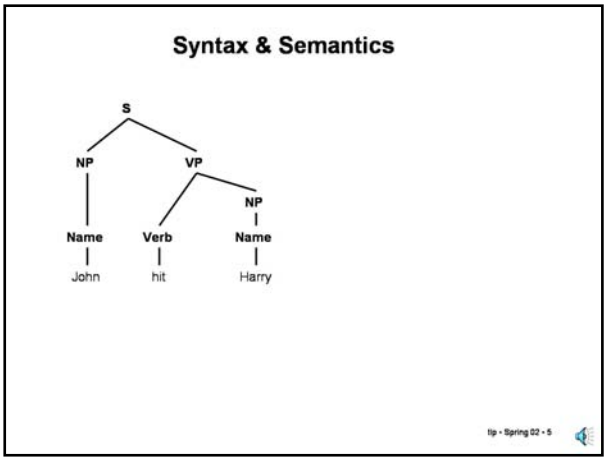
Slide 12.3.4

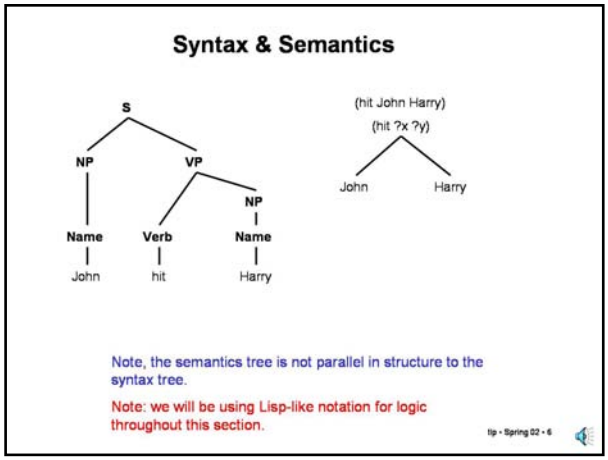
Part of the role of pragmatics, the next phase of processing, is to try to make those connections.

6.034 Artificial Intelligence. Copyright © 2004 by Massachusetts Institute of Technology.

Slide 12.3.5

So, let's consider a very simple sentence "John hit Harry". We have here the simple parse tree. What should we expect the semantic representation to be?





Slide 12.3.6

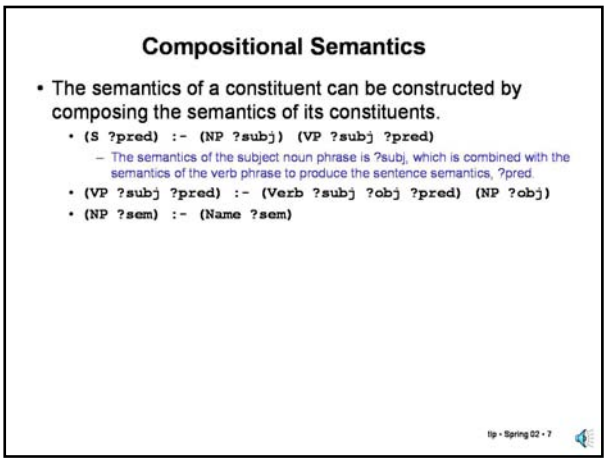
In this simple case, we might want something like this, where hit is a predicate and John and Harry are constant terms in the logical language. The key thing to notice is that even for this simple sentence the semantic structure produced is not perfectly parallel to the syntactic structure.

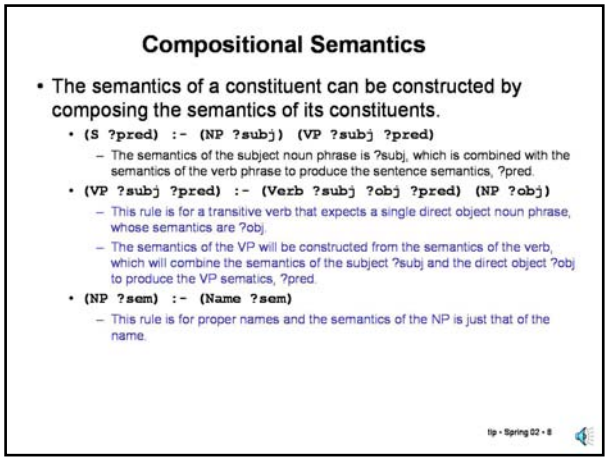
In this interpretation, the meaning of the verb is the center of the semantics. The meaning representation of the subject NP is embedded in the meaning representation of the verb phrase. This suggests that producing the semantics will not be a trivial variant of the parse tree. So, let's see how we can achieve this.

Slide 12.3.7

Our guiding principle will be that the semantics of a constituent can be constructed by composing the semantics of its constituents. However, the composition will be a bit subtle and we will be using feature values to carry it out.

Let's look at the sentence rule. We will be exploiting the "two way" matching properties of unification strongly here. This rule says that the meaning of the sentence is picked up from the meaning of the VP, since the second argument of the VP is the same as the semantics of the sentence as a whole. We already saw this in our simple example, so it comes as no surprise. Note also that the semantics of the subject NP is passed as the first argument of the VP (by using the same variable name).





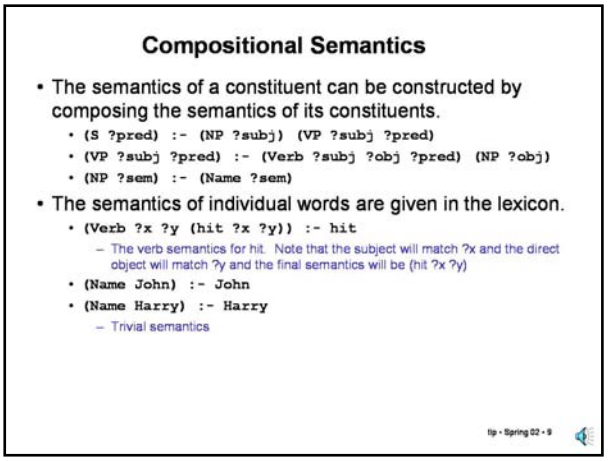
Slide 12.3.8

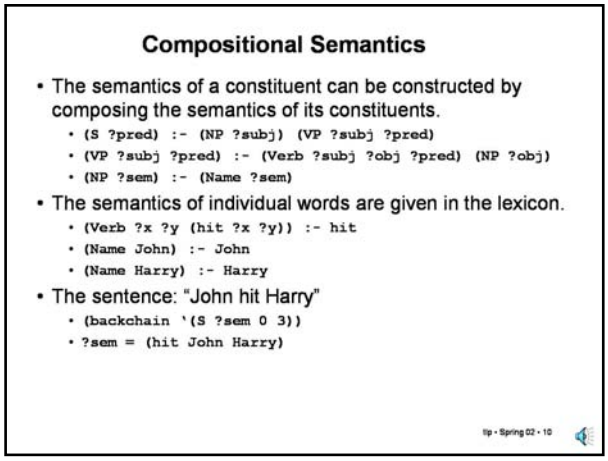
The VP has two arguments, the semantics of the subject NP (which will be an input) and the resulting semantics of the VP. In the VP rule, we see that the result semantics is coming from the Verb, which is combining the semantics of the subject and the object NPs to produce the result for the VP (and ultimately the sentence).

6.034 Artificial Intelligence. Copyright © 2004 by Massachusetts Institute of Technology.

Slide 12.3.9

Let's look at the rule for a particular Verb. Note that the first two arguments are simply variables which are then included in the expression for the verb semantics, the predicate hit with two arguments (the subject and the object).



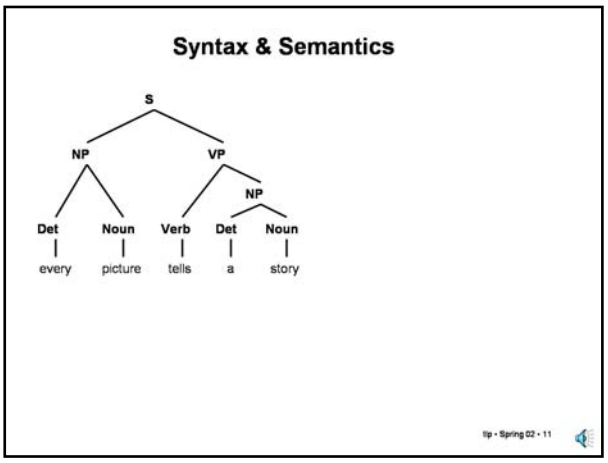


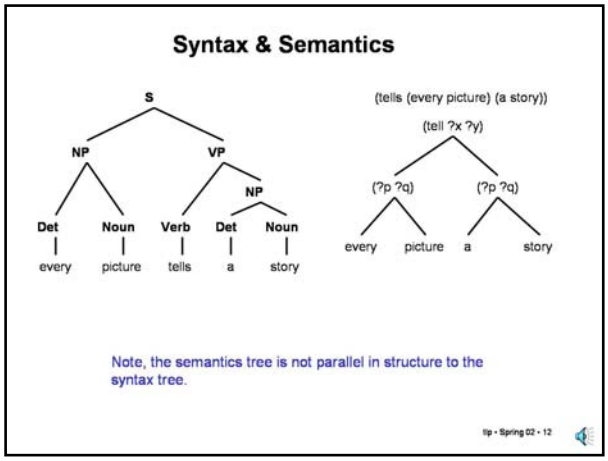
Slide 12.3.10

We can pull this altogether by simply calling backchain with the goal pattern for a successful parse. We will want to retrieve the value of the binding for ?sem, which is the semantics for the sentence.

Slide 12.3.11

Let's look at a somewhat more complex example - "Every picture tells a story". Here is the syntactic analysis.





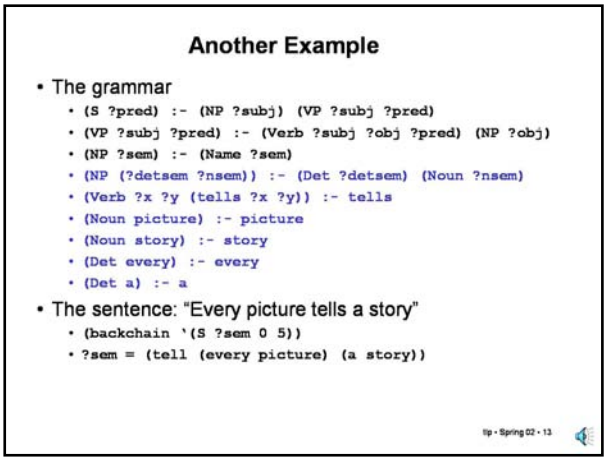
Slide 12.3.12

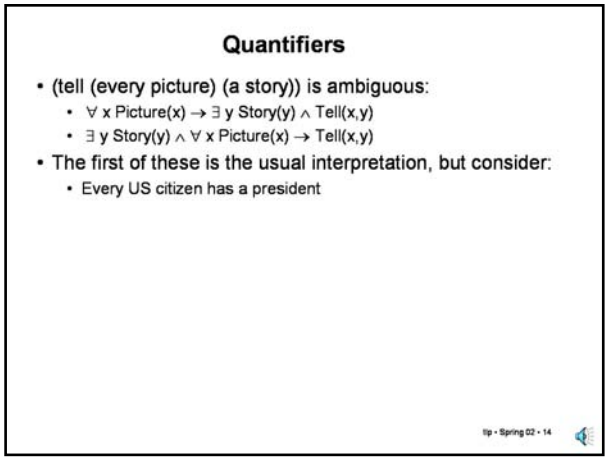
This is one possible semantic analysis. Note that it follows the pattern of our earlier example. The top-level predicate is derived from the verb and it includes as arguments the semantics of the subject and direct object.

6.034 Artificial Intelligence. Copyright © 2004 by Massachusetts Institute of Technology.

Slide 12.3.13

The only innovation in this grammar, besides the new words is a simple semantics for a noun phrase formed from a Determiner and a Noun - just placing them in a list. We can interpret the result as a quantifier operating on a predicate. But, what does this mean? It's certainly not legal logic notation.



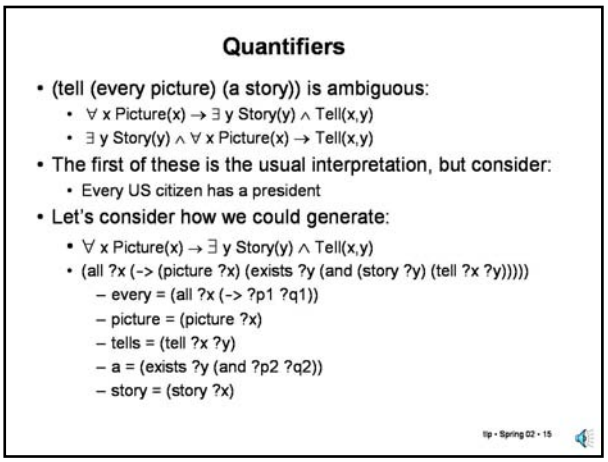


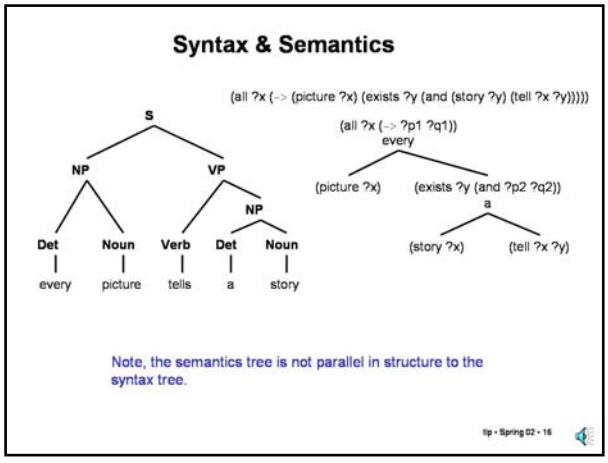
Slide 12.3.14

Furthermore, even if we are generous and consider this a legal quantified expression, then it's ambiguous - in the usual sense that "Every man loves a woman" is ambitious. That is, is there one story per picture or do all the pictures tell the same story.

Slide 12.3.15

Let's pick one of the interpretations and see how we could generate it. At the heart of this attempt is a definition of the meaning of the determiners "every" and "a", which now become patterns for universally and existentially quantified statements. Note also that the nouns become patterns for predicate expressions.





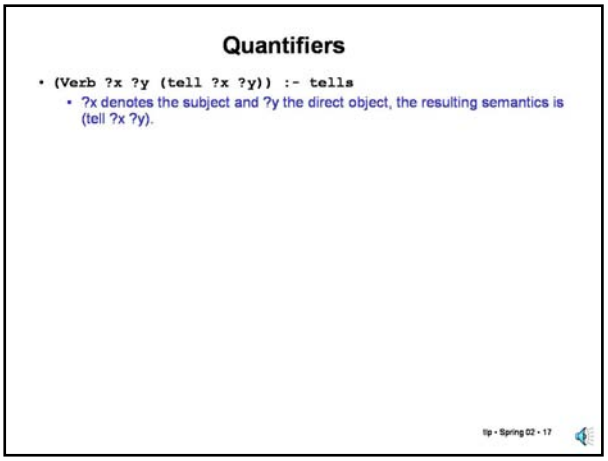
Slide 12.3.16

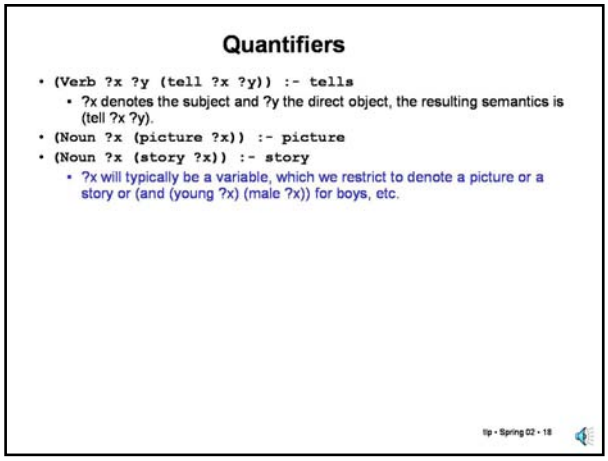
Our target semantic representation is shown here. Note that by requiring the semantics to be a legal logical sentence, we've had to switch the key role from the verb to the determiner. That is, the top node in the sentence semantics comes from the determiner, not the verb. The semantics of the verb is fairly deeply nested in the final semantics - but it still needs to combine the semantics of the subject and direct object NPs. Note, however, that it is incorporating them by using the quantified variable introduced by the determiners of the subject and object NPs.

6.034 Artificial Intelligence. Copyright © 2004 by Massachusetts Institute of Technology.

Slide 12.3.17

Let's start with the definitions of the words. Here's the definition for the verb "tells". We have seen this before. It combines the semantics of the subject NP (bound to ?x) and the semantics of the object NP (bound to ?y) with the predicate representing the verb to produce the VP semantics.



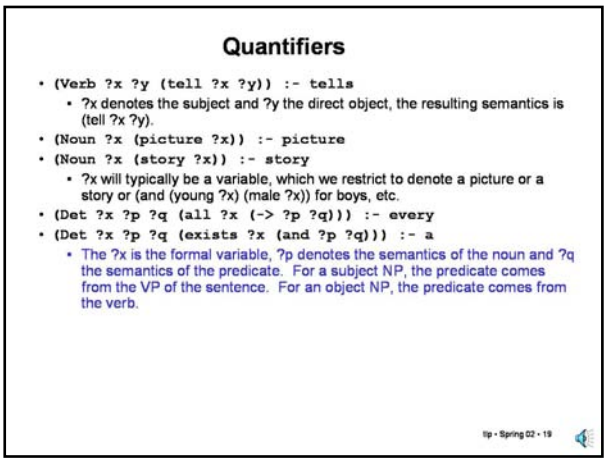


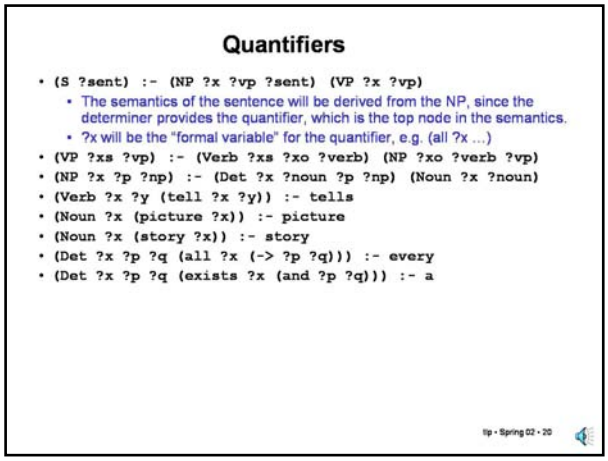
Slide 12.3.18

The nouns will be denoted by one of the quantified variables introduced by the quantifiers. The noun places a restriction on the entities that the variable can refer to. In this definition, the quantified variable will be bound to ?x and incorporated into the predicate representing the noun.

Slide 12.3.19

Finally, the determiners are represented by quantified formulas that combine the semantics derived from the noun with the semantics of the VP (for a subject NP) or of the Verb (for an object NP).





Slide 12.3.20

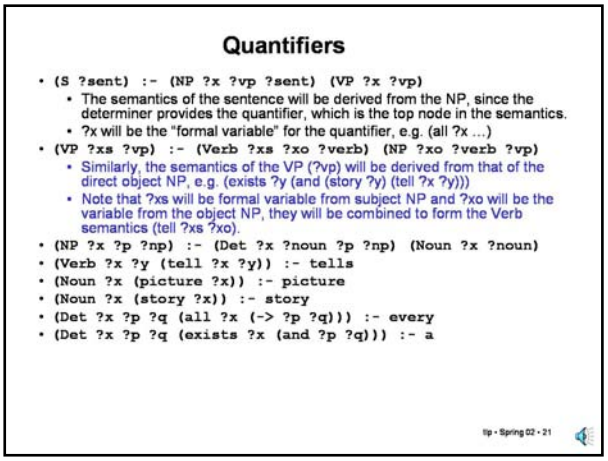
The new sentence (S) rule reflects the difference in where the top-level semantics is being assembled. Before, we passed the semantics of the subject NP into the VP, now we go the other way. The semantics of the VP is an argument to the subject NP.

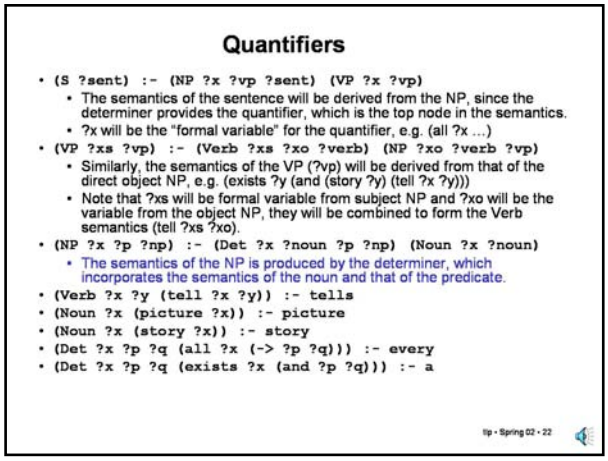
Note that the variable ?x here will not be bound to anything, it is the variable that will be used as the quantified variable by the determiner's semantics.

6.034 Artificial Intelligence. Copyright © 2004 by Massachusetts Institute of Technology.

Slide 12.3.21

The VP rule is analogous. The semantics of the Verb will combine a reference to the subject and object semantics (through their corresponding quantified variables) and the resulting semantics of the Verb will be combined into the semantics of the NP (which will ultimately be derived from the semantics of the determiner).





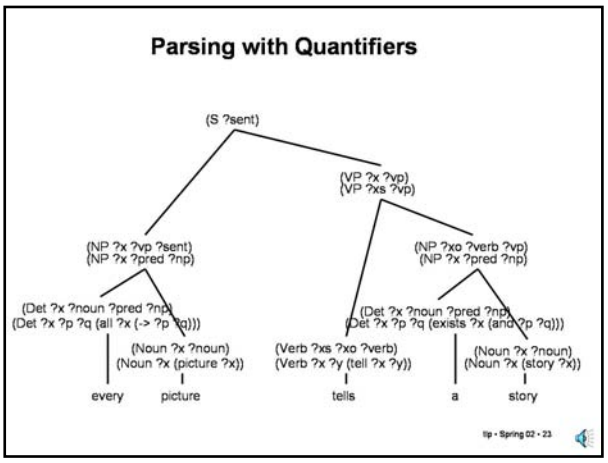
Slide 12.3.22

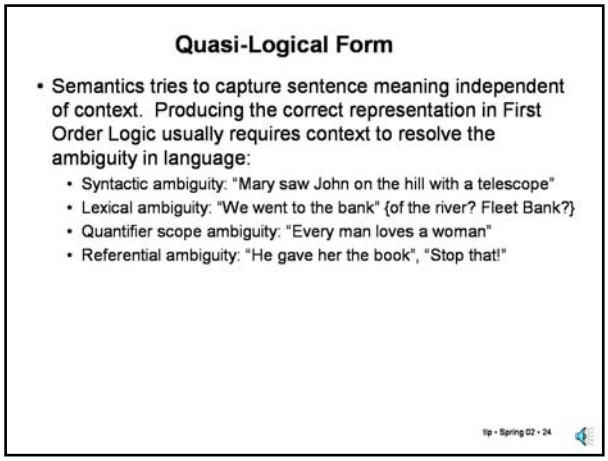
The NP rule in fact takes ?p, which will be the semantics of the Verb phrase and combine them with the semantics of the noun in the semantics of the Determiner.

Slide 12.3.23

Here we see how the parse works out. You have to follow the bindings carefully to see how it all works out.

What is remarkable about this is that we were able to map from a set of words to a first-order logic representation (which does not appear to be very similar) with a relatively compact grammar and with quite generic mechanisms.





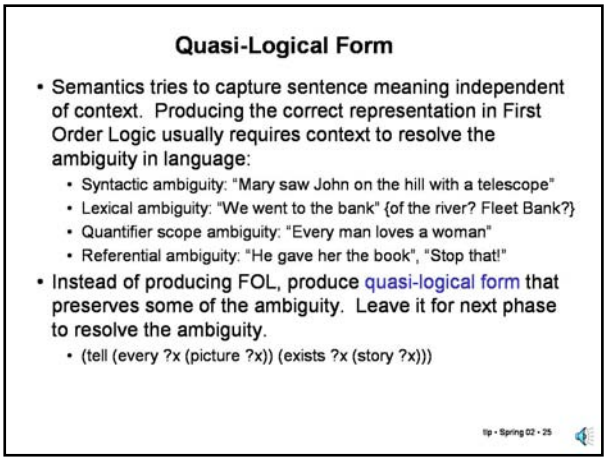
Slide 12.3.24

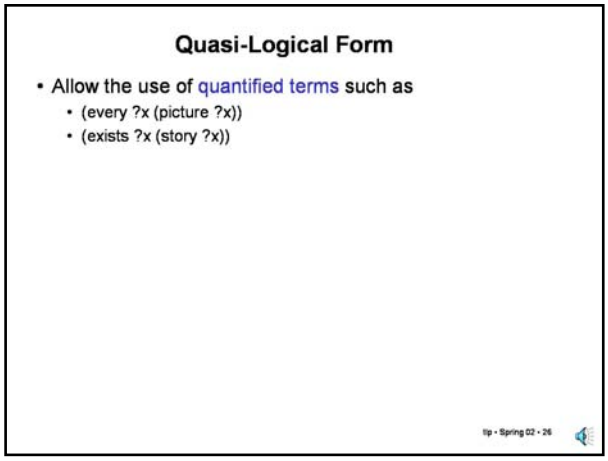
The quantified expression we produced in the previous example is unambiguous, as required to be able to write an expression in first order logic. However, natural language is far from unambiguous. We have seen examples of syntactic ambiguity, lexical and attachment ambiguity in particular, plus there are many examples of semantic ambiguity, for example, ambiguity in quantifier scope and ambiguity on who or what pronouns refer to are examples.

6.034 Artificial Intelligence. Copyright © 2004 by Massachusetts Institute of Technology.

Slide 12.3.25

One common approach to semantics is to have it produce a representation that is not quite the usual logical notation, sometimes called **quasi-logical form**, that preserves some of the ambiguity in the input, leaving it to the pragmatics phase to resolve the ambiguities employing contextual information.



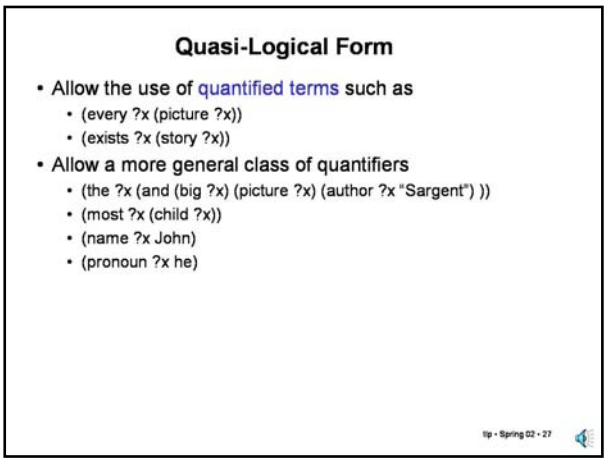


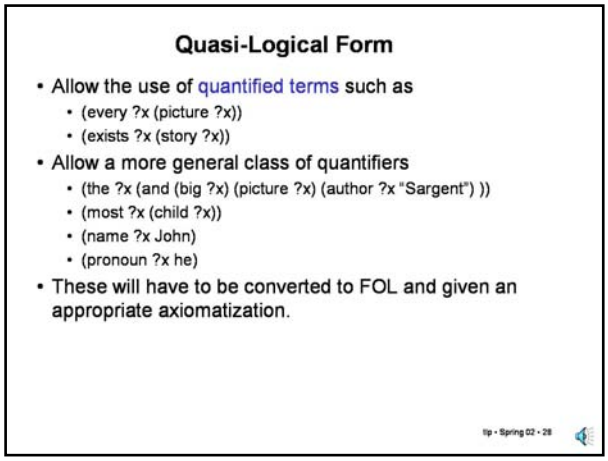
Slide 12.3.26

One common aspect of quasi-logical notation is the use of **quantified terms**. These terms indicate the nature of the intended quantification but do not specify the scope of the quantifier in the sentence and thus preserves the ambiguity in the natural language. Note that we are treating these quantified expressions as terms, and using them as arguments to functions and predicates - which is not legal FOL.

Slide 12.3.27

In quasi-logical notation, one also typically extends the range of available quantifiers to correspond more closely to the range of determiners available in natural language. One important case, is the determiner "the", which indicates a unique descriptor.





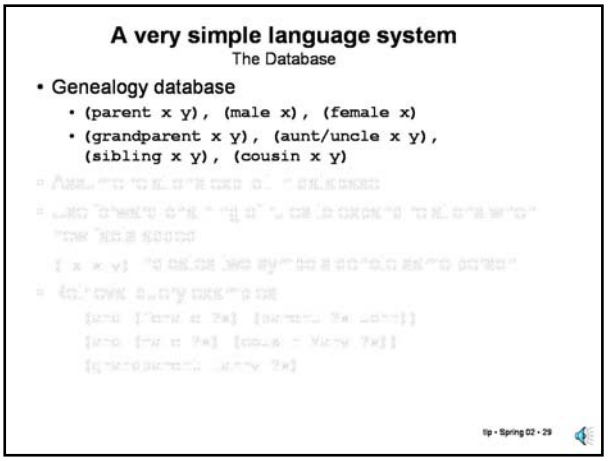
Slide 12.3.28

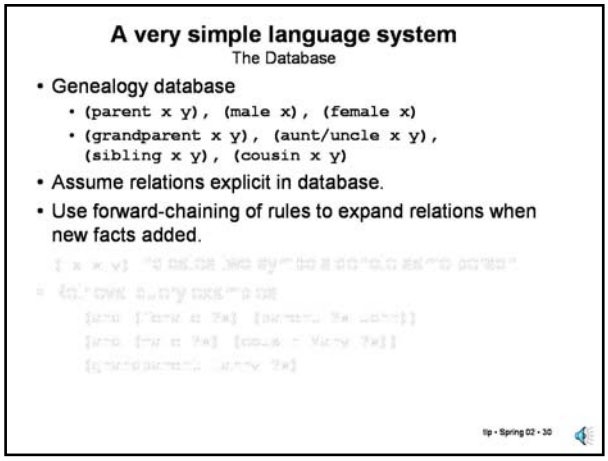
These quantified terms and generalized quantifiers will require conversion to standard FOL together with a careful axiomatization of their intended meaning before the resulting semantics can be used for inference.

6.034 Artificial Intelligence. Copyright © 2004 by Massachusetts Institute of Technology.

Slide 12.3.29

Let's illustrate how the type of language processing we have been discussing here could be used to build an extremely simple database system. We'll assume that we want to deal with a simple genealogy domain. We will have facts in our database describing the family relationships between some set of people. We will not restrict ourselves to just the minimal set of facts, such as parent, male and female, we will also keep derived relationships such as grandparent and cousin.



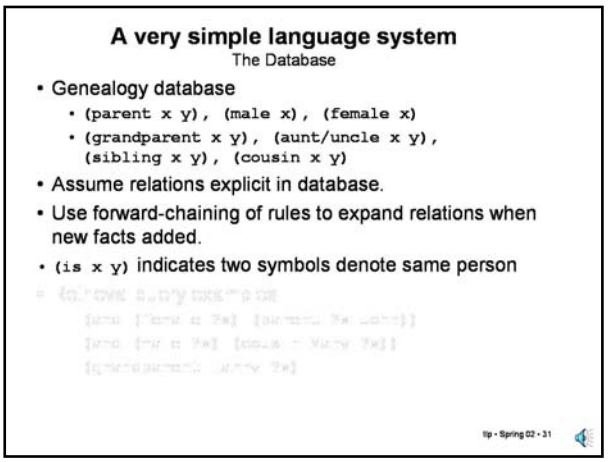


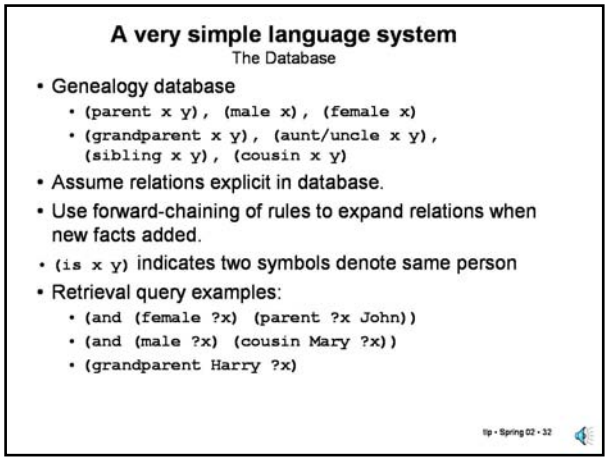
Slide 12.3.30

In fact, we will assume that all the relationships between people we know about are explicitely in the database. We can accomplish them by running a set of Prolog-like rules in forward chaining fashion whenever a new fact is added. We do this, rather than do deduction at retrieval time because of issues of equality, which we will discuss momentarily.

Slide 12.3.31

We will also allow assertions of the form (is x y) which indicate that two symbols denote the same person. We will assume that the forward chaining rules will propagate this equality to all the relevant facts. That is, we substitute equals for equals in each predicate, explicitely. This is not efficient, but it is simple.





Slide 12.3.32

We can now do very simple retrieval from this database of facts using our backchaining algorithm. We initialize the goal stack in backchaining with the query. If the query is a conjunction, we initialize the stack with all the conjuncts.

6.034 Artificial Intelligence. Copyright © 2004 by Massachusetts Institute of Technology.

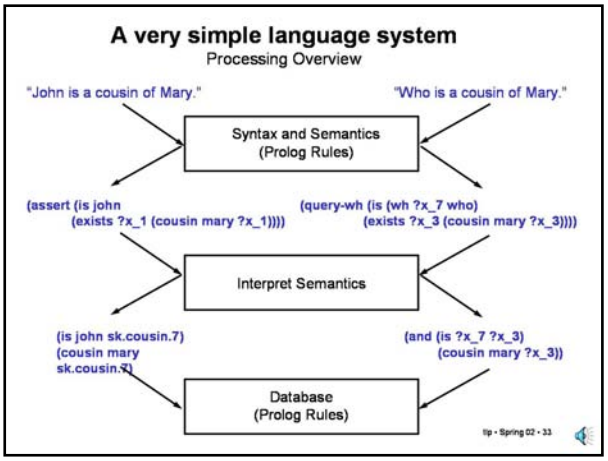
Slide 12.3.33

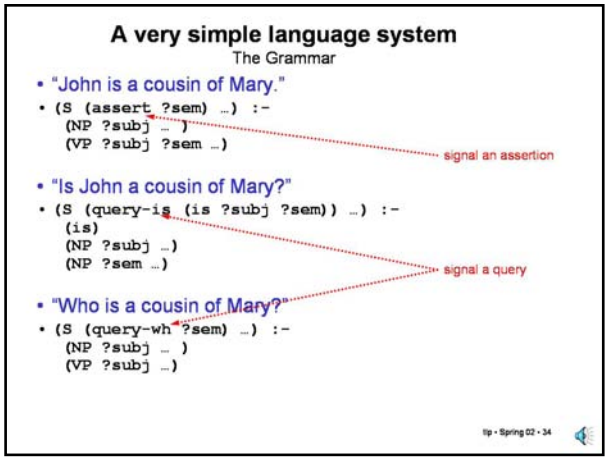
Here we see a brief overview of the processing that we will do to interact with the genealogy database.

We will be able to accept declarative sentences, such as "John is a cousin of Mary". These sentences will be processed by a grammar to obtain a semantic representation. This representation will then be interpreted as a set of facts to be added to the database.

We can also ask questions, such as "Who is a cousin of Mary". Our grammar will produce a semantic representation. The semantics of this type of sentence is converted into a database query and passed to the database.

Let's look in more detail at the steps of this process.





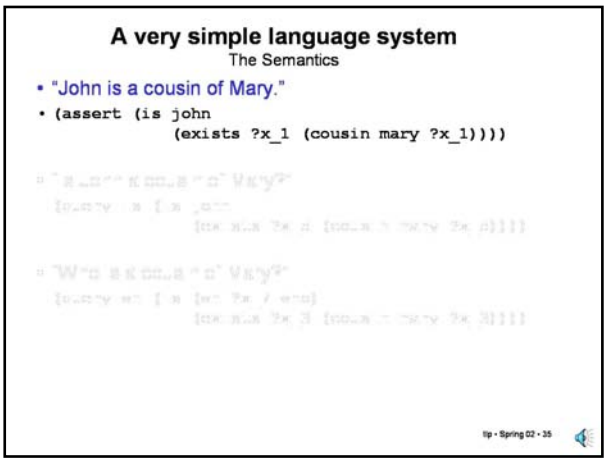
Slide 12.3.34

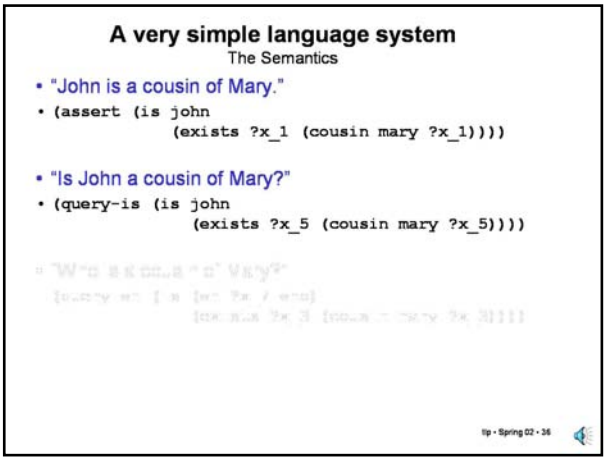
We will need a grammar built along the lines we have been discussing. One of the things the grammar does is classify the sentences into declarative sentences, such as "John is a cousin of Mary", which will cause us to assert a fact in our database, and questions, such as, "Is John a cousin of Mary" or "Who is a cousin of Mary", which will cause us to query the database.

Slide 12.3.35

Here we see one possible semantics for the declarative sentence "John is a cousin of Mary". The operation `assert` indicates the action to be taken. The body is in quasi-logical form; the quantified term `(exists ?x_1 (cousin mary ?x_1))` is basically telling us there exists a person that is in the cousin relationship to Mary. The outermost `is` assertion is saying that John denotes that person. This is basically interpreting this quasi-logical form as:

```
] x . (is John x) ^ (cousin Mary x)
```





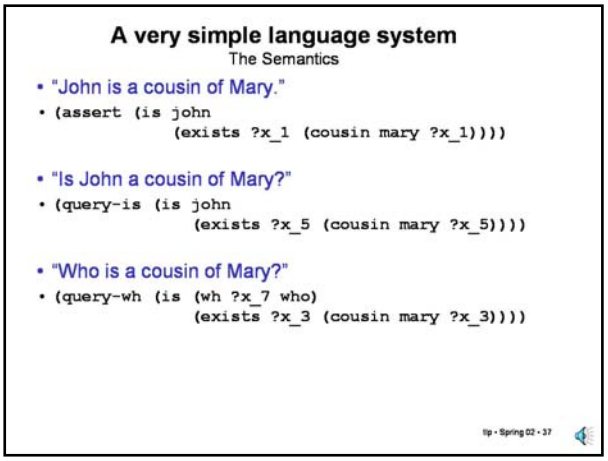
Slide 12.3.36

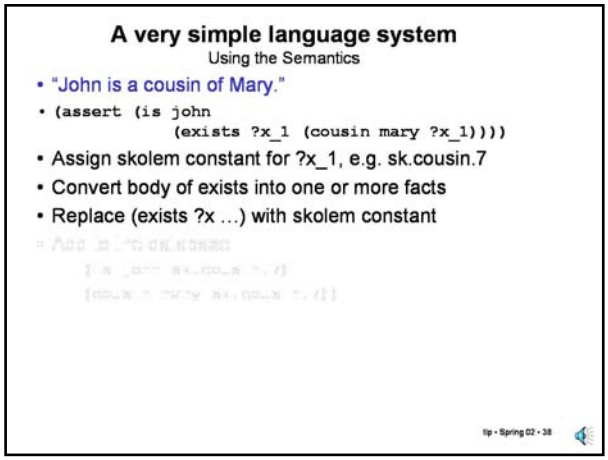
The semantics of the question "Is John a cousin of Mary?" is essentially identical to that of the declarative form, but it is prefixed by a query operation rather than an assertion. So, we would want to use this to query the database rather than for asserting new facts.

6.034 Artificial Intelligence. Copyright © 2004 by Massachusetts Institute of Technology.

Slide 12.3.37

We can also have a question such as "Who is a cousin of Mary", which is similar except that John is replaced by a term indicating that we are interested in determining the value of this term.





Slide 12.3.38

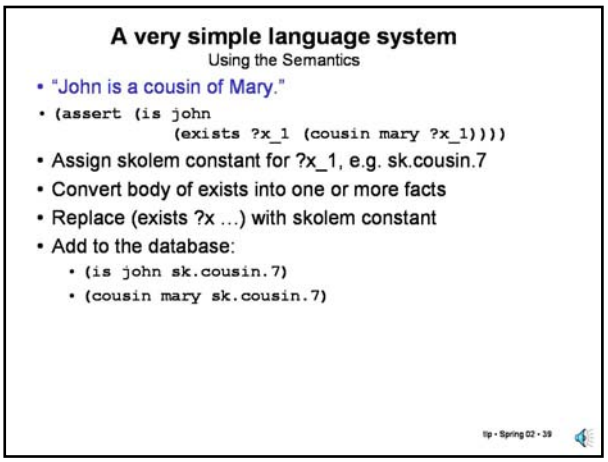
Given the semantics, we have to actually decide how to add new facts and do the retrieval. Here we show an extremely simple approach that operates for these very simple types of queries (note that we are only using existentially quantified terms).

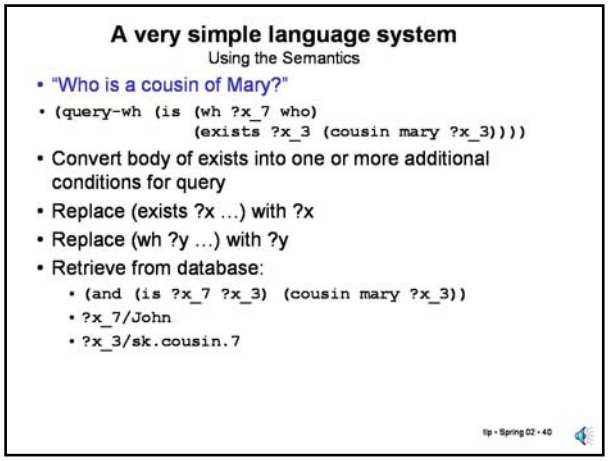
We are basically going to turn the assertion into a list of ground facts to add to the database. We will do this by skolemizing. Since we have only existentially quantified variables, this will eliminate all variables.

We replace the quantified terms with the corresponding skolem constant and we convert the body of the quantified term into a set of facts that describe the constant.

Slide 12.3.39

In this example, we get two new facts. One is from the outer `is` assertion which tells us that John denotes the same person as the skolem constant. The second fact comes from the body of the quantified term which tells us some properties of the person denote by the skolem constant.





Slide 12.3.40

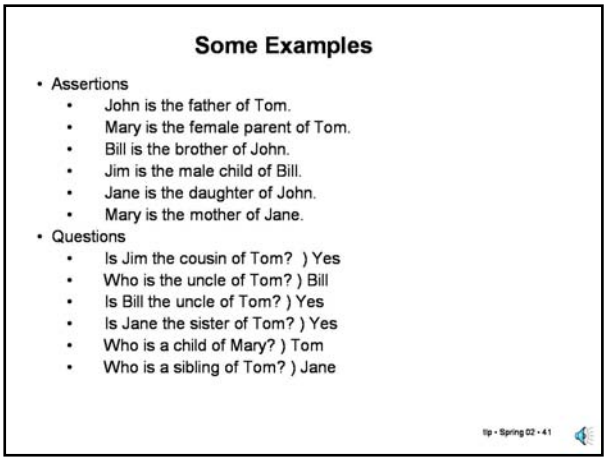
We process the question in a similar way except that instead of using skolem constants we keep the variables, since we want those to match the constants in the database. When we perform the query, ? x_7 is bound to John as expected. In general, there may be multiple matches for the query, some may be skolem constants and some may be people names. We would want to return the specific names whenever possible.

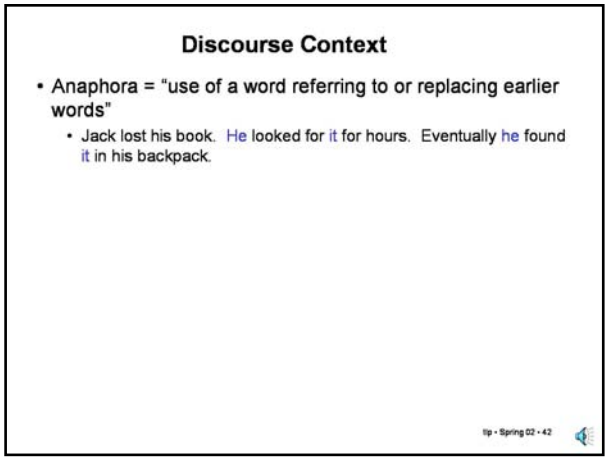
6.034 Artificial Intelligence. Copyright © 2004 by Massachusetts Institute of Technology.

Slide 12.3.41

Here are some examples that show that this approach can be used to do a little inference above and beyond what is explicitely stated. Note that the assertions do not mention cousin, uncle, sister or sibling relations, those are inferred. So, we are going beyond what an Internet search engine can do, that is, pattern match on the presence of particular words.

This example has been extremely simple but hopefully it illustrates the flavor of how such a system may be built using the tools we have been developing and what could be done with such a system.





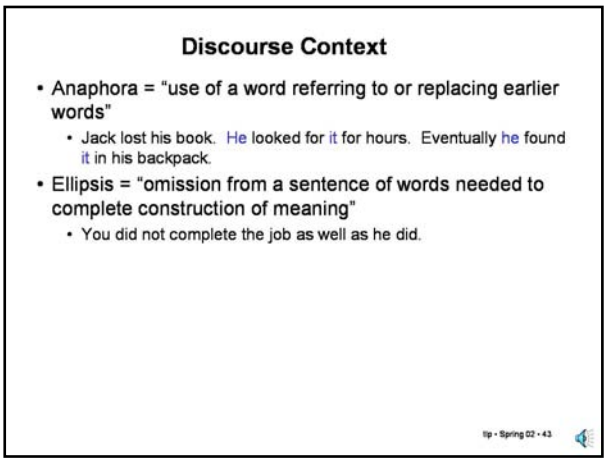
Slide 12.3.42

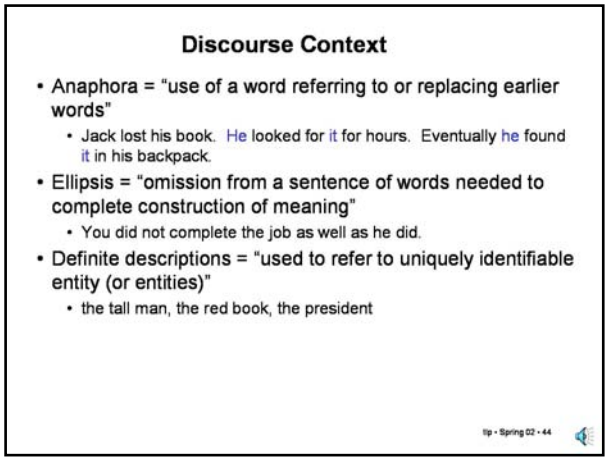
At this point, we'll touch briefly on a set of phenomena that are beyond the scope of pure semantics because they start dealing with the issue of context.

One general class of language phenomena is called **anaphora**. this includes pronoun use, where a word is used to refer to other words appearing either elsewhere in the sentence or in another sentence.

Slide 12.3.43

Another phenomenon is called **ellipsis**, when words or phrases are missing and need to be filled in from context. In this example, the phrase "complete the job" is missing from the enf of the second conjoined sentence.





Slide 12.3.44

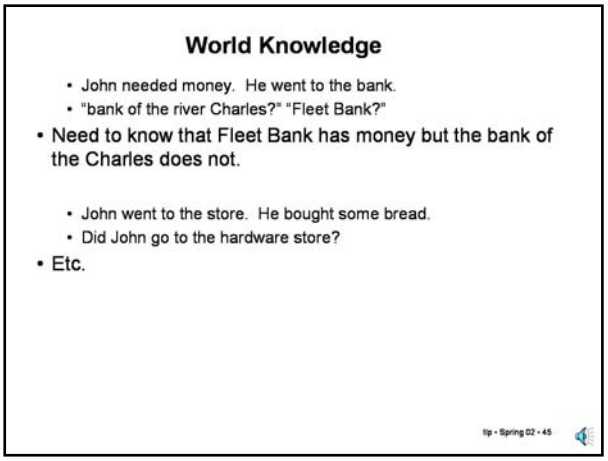
Another important mechanism in language is the use of **definite descriptions**, signaled by the determiner "the". The intent is that the listener be able to identify an entity previously mentioned or expected to be known.

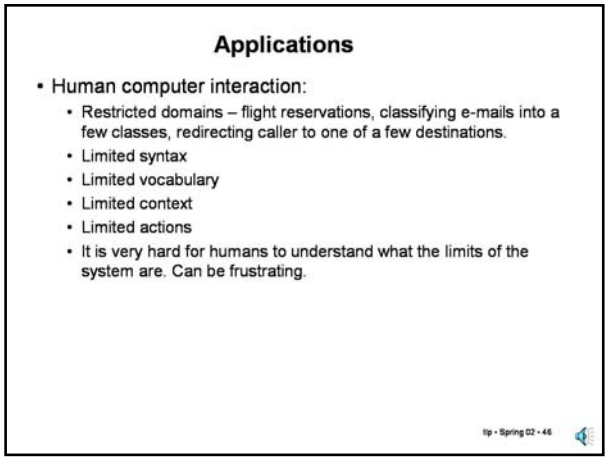
All of these are linguistic mechanisms for incorporating context and require that a language understanding system that is engaged in an interaction with a human keep a context and be able to identify entities and actions in context based on the clues in the sentence. This is an area of active research and some systems with competence in this area have been built.

6.034 Artificial Intelligence. Copyright © 2004 by Massachusetts Institute of Technology.

Slide 12.3.45

Even beyond conversational context, understanding human language requires access to the whole range of human knowledge. Even when speaking with a child, one assumes a great deal of "common sense" knowledge that computers are, as yet, sorely lacking in. The problem of language understanding at this point merges with the general problem of knowledge representation and use.





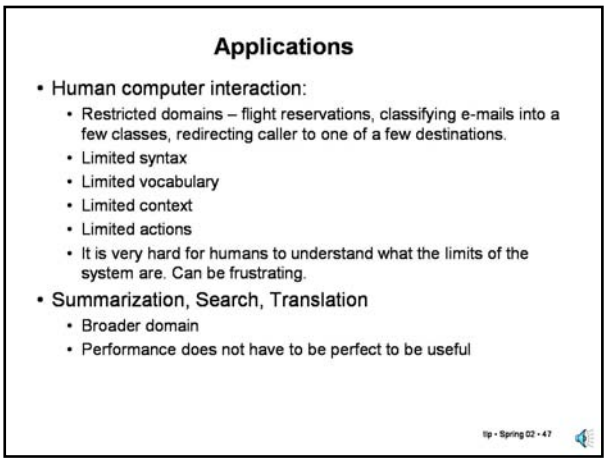
Slide 12.3.46

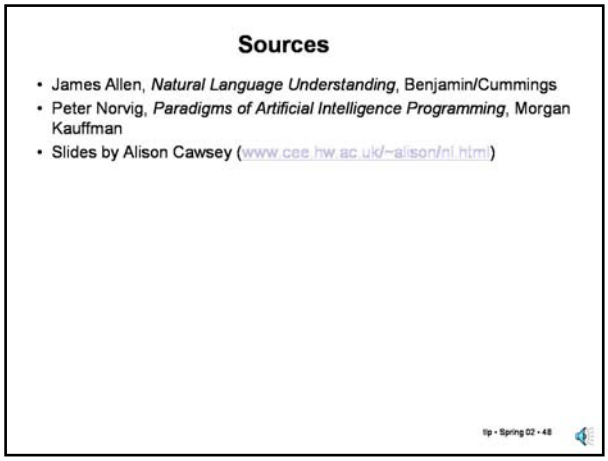
Real applications of natural language technology for human computer interfaces require a very limited scope so that the computer can get by with limited language skills and can have enough knowledge about the domain to be useful. However, it is difficult to keep people completely within the language and knowledge boundaries of the system. This is why the use of natural language interfaces is still limited.

Slide 12.3.47

There is, however, a rapidly increasing use of limited language processing in tasks that don't involve direct interaction with a human but do require some level of understanding of language. These tasks are characterized by situations where there is value in even limited capabilities, e.g. doing the first draft of a translation or a building a quick summary of a much longer news article.

I expect to see an explosion of applications of natural language technologies in the near future.





Slide 12.3.48

Here are some sources that were used in the preparation of these slides and which can serve as additional reading material.