• Consider the following arguments

Every A is a B
Every B is a C
Every A is a C

Some A is a B
No B is a C
Some A is not a C

• Arguments of this general kind are known as syllogisms
• The study of logic began with the syllogism!
Consider the very simple grammar (note that the only verb is the copula *is*):

**Syntax**

\[ IP/\varphi(\psi) \rightarrow NP/\varphi, \ I'/\psi \]
\[ I'/\varphi \rightarrow \text{is a } \ N'/\varphi \]
\[ I'/\lambda p\lambda x[\neg p(x)](\varphi) \rightarrow \text{is not a } \ N'/\varphi \]
\[ NP/\varphi \rightarrow \text{PropN}/\varphi \]
\[ NP/\varphi(\psi) \rightarrow \text{Det}/\varphi, \ N'/\psi \]
\[ N'/\varphi \rightarrow \text{N}/\varphi \]

**Lexicon**

\[ \text{Det}/\lambda p\lambda q[\exists x(p(x) \land q(x))] \rightarrow \text{some} \]
\[ \text{Det}/\lambda p\lambda q[\forall x(p(x) \rightarrow q(x))] \rightarrow \text{every} \]
\[ \text{Det}/\lambda p\lambda q[\forall x(p(x) \rightarrow \neg q(x))] \rightarrow \text{no.} \]
\[ \text{N}/\lambda x[\text{man}(x)] \rightarrow \text{man} \]
\[ \text{N}/\lambda x[\text{mortal}(x)] \rightarrow \text{mortal} \]

...
This grammar yields correct semantics for the relevant fragment of English:

Every $A$ is a $B$ \quad $\forall x (a(x) \rightarrow b(x))$
Every $A$ is not a $B$ \quad $\forall x (a(x) \rightarrow \neg b(x))$
Some $A$ is a $B$ \quad $\exists x (a(x) \land b(x))$
Some $A$ is not a $B$ \quad $\exists x (a(x) \land \neg b(x))$
$S$ is a $B$ \quad $b(s)$
$S$ is not a $B$ \quad $\neg b(s)$
• Consider the clause forms of these sentences

\[
\begin{align*}
\forall x (a(x) \rightarrow b(x)) & \quad \neg a(x) \lor b(x) \\
\forall x (a(x) \rightarrow \neg b(x)) & \quad \neg a(x) \lor \neg b(x) \\
\exists x (a(x) \land b(x)) & \quad a(s), \ b(s) \\
\exists x (a(x) \land \neg b(x)) & \quad a(s), \ \neg b(s) \\
b(s) & \quad b(s) \\
\neg b(s) & \quad \neg b(s)
\end{align*}
\]

• There are no function-symbols, and they have at most two literals each.
• Resolving (or factoring) clauses with at most two literals produces only clauses with at most two literals:

\[
\begin{array}{c}
\neg p(x) \lor q(x) \quad \neg q(x') \lor \neg r(x') \\
\hline
\neg p(x) \lor \neg r(x)
\end{array}
\]

• Resolving (or factoring) clauses with no function-symbols introduces no new terms or predicates.
• Suppose we have a set $E$ of sentences in our syllogistic fragment, comprising $n$ words. The translation to first-order logic and thence to clause form will yield at most $n$ clauses over a signature of at most $n$ predicates and at most $n$ individual constants, with each clause being function-free and having at most two literals.

• There are (up to renaming of variables) at most $(2n(n + 1))^2$ such clauses.

• So a resolution theorem-prover working on these clauses will stop in time bounded by a polynomial function of $n$. 
• Let the fragment of English defined above be called $\mathcal{E}_0$.

• Then we have

  **Theorem:** The problem of determining the satisfiability of a set of sentences in $\mathcal{E}_0$ can be decided in polynomial time.

• Erm, well, *actually*, the complexity of the validity problem for syllogisms is NLogSpace-complete . . .
• Consider the following argument:

  Every philosopher who is not a stoic is a cynic
  Every stoic is a man
  Every cynic is a man

  Therefore, every philosopher is a man
We can capture this argument by adding the following grammar rules:

**Syntax**

\[
N' / \varphi(\psi) \rightarrow N/\psi, \ CP/\varphi \\
CP/\varphi(\psi) \rightarrow \text{CSpec}/\varphi, \ C' / \psi \\
C'/\lambda t[\varphi] \rightarrow C, \ IP_t/\varphi \\
NP/\varphi \rightarrow \text{RelPro}/\varphi \\
\text{CSpec} \rightarrow 
\]

**Lexicon**

\[
\text{RelPro}/\lambda Q \lambda P \lambda x[P(x) \land Q(x)] \rightarrow \text{who, which}. \\
C \rightarrow 
\]

In addition, we have the usual Wh-movement rule.
This grammar yields phrase-structures of the following form:
• It is able to handle the above argument:

\[ \forall x (\text{philosopher}(x) \land \neg \text{stoic}(x) \rightarrow \text{cynic}(x)) \]
\[ \forall x (\text{stoic}(x) \rightarrow \text{man}(x)) \]
\[ \forall x (\text{cynic}(x) \rightarrow \text{man}(x)) \]
\[ \forall x (\text{philosopher}(x) \rightarrow \text{man}(x)) \]
• Let the fragment of English defined above be called $E_1$.
• Then we have

  **Theorem:** The problem of determining the satisfiability of a set of sentences in $E_1$ is NP-complete.
• What about non-copula verbs?

Every horse is an animal

Every head which some horse has is a head which some animal has.
We can capture this argument by adding the following grammar rules:

**Syntax**

\[ I'/\varphi \rightarrow VP/\varphi \]

\[ VP/\varphi(\psi) \rightarrow V/\varphi, NP/\psi \]

**Lexicon**

\[ V/\lambda s \lambda x[s(\lambda y[have(x, y)])] \rightarrow have \]

.....
These rules translate the above argument as follows:

\[ \forall x (\text{horse}(x) \to \text{animal}(x)) \]

\[ \forall x (\text{head}(x) \land \exists y (\text{horse}(y) \land \text{has}(y, x)) \to \text{head}(x) \land \exists y (\text{animal}(y) \land \text{has}(y, x))). \]
• Let the fragment of English defined above be called $\mathcal{E}_2$.
• Then we have

Theorem: The problem of determining the satisfiability of a set of sentences in $\mathcal{E}_2$ is EXPTIME-complete.
• Consider the following argument:
  
  Every artist despises some bureaucrat
  Every bureaucrat admires every artist who despises him
  Every artist despises some bureaucrat who admires him
• We can capture this argument by adding the following grammar rules:

**Syntax**

NP → Reflexive  
NP → Pronoun  
I' → NegP, VP

**Lexicon**

Reflexive → itself (him/herself)  
Pronoun → it (he/she/him/her)  
NegP → does not

• Erm, we omit the formal semantics this time . . .
... though they can be provided, and translate the above argument as follows:

\[
\forall x (\text{artist}(x) \rightarrow \\
\exists y (\text{bureaucrat}(y) \land \text{despises}(x, y)))
\]

\[
\forall x (\text{bureaucrat}(x) \rightarrow \\
\forall y (\text{artist}(y) \land \text{despises}(y, x) \rightarrow \text{admires}(x, y)))
\]

\[
\forall x (\text{artist}(x) \rightarrow \\
\exists y (\text{bureaucrat}(y) \land \text{admires}(y, x) \land \text{despises}(x, y))).
\]
• We have to be careful about anaphoric ambiguities:

Every artist who employs a caretaker despises every bureaucrat who admires him

• This translates to

\[ \forall x_1 (\text{artist}(x_1) \land \exists x_2 (\text{caretaker}(x_2) \land \text{employ}(x_1, x_2)) \rightarrow \forall x_3 (\text{bureaucrat}(x_3) \land \text{admire}(x_3, x_1) \rightarrow \text{despise}(x_1, x_3))). \]

and

\[ \forall x_1 \forall x_2 (\text{artist}(x_1) \land \text{caretaker}(x_2) \land \text{employ}(x_1, x_2) \rightarrow \forall x_3 (\text{bureaucrat}(x_3) \land \text{admire}(x_3, x_2) \rightarrow \text{despise}(x_1, x_3))). \]
We can see what is going on better by looking at the sentence structure:

```
IP
   /      \
  NP    I'
  /  \
Det N'  VP
 / \
Every N CP
   /\  |
   \ C Spec
      /\  |
     RelPro C I'
       /\  |
      t NP I'
       \|
        V NP
          /|
         V NP
            |
            |
employs a caretaker
```

```
I'
 / \
NP   V
  / \
Despises Det N CP
  / \
Every N Spec C I'
  / \
Bureaucrat RelPro C I'
   / \
   t NP I'
    \|
     V NP
      |
      |
admires him/himself
```
• Let the fragment of English defined above in which pronouns take their closest antecedents be called $\mathcal{E}_3$.

• Then we have

  **Theorem:** The problem of determining the satisfiability of a set of sentences in $\mathcal{E}_3$ is NEXPTIME-complete.
• Let the fragment of English defined above, but with pronouns allowed to take any (grammatically acceptable) antecedents be called $\mathcal{E}_4$.

• Then we have

  **Theorem:** The problem of determining the satisfiability of a set of sentences in $\mathcal{E}_4$ is undecidable.
• Adding numbers to Cop
  At least thirteen artists are beekeepers
  At most three beekeepers are carpenters
  At most four dentists are not carpenters
  At least six artists are not dentists.

• This gives us the fragment Cop+Num.
• Adding numbers to Cop+TV

At most one artist admires at most seven beekeepers
At most two carpenters admire at most eight dentists
At most three artists admire at least seven electricians
At most four beekeepers are not electricians
At most five dentists are not electricians
At most one beekeeper is a dentist
At most six artists are carpenters

• This gives us the fragment Cop+TV+Num.
Time to cut a long story short:

<table>
<thead>
<tr>
<th>Fragment</th>
<th>Computational complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cop</td>
<td>PTIME</td>
</tr>
<tr>
<td>Cop+Rel</td>
<td>NP-complete</td>
</tr>
<tr>
<td>Cop +TV+DTV</td>
<td>PTIME</td>
</tr>
<tr>
<td>Cop+Rel+TV</td>
<td>EXPTIME-complete</td>
</tr>
<tr>
<td>Cop+Rel+TV+DTV</td>
<td>NEXPTIME-complete</td>
</tr>
<tr>
<td>Cop+Rel+TV+RA</td>
<td>NEXPTIME-complete</td>
</tr>
<tr>
<td>Cop+Rel+TV+GA</td>
<td>undecidable</td>
</tr>
<tr>
<td>Cop+Rel+TV+DVT+RA</td>
<td>NEXPTIME-complete</td>
</tr>
<tr>
<td>Cop+Num</td>
<td>NP-complete</td>
</tr>
<tr>
<td>Cop+TV+Num</td>
<td>NEXPTIME-complete</td>
</tr>
</tbody>
</table>
• In this course, we have covered the basic techniques needed to compute the meanings of natural language sentences, and to perform logical inferences with the meanings thus obtained.

• Topic breakdown
  - Prolog
  - first-order logic
  - how Prolog fits into logic: the SATCHMO theorem-prover
  - natural language syntax: transformational grammar
  - parsing: encoding transformational grammars using DCGs
  - higher-order logic ($\lambda$-calculus)
  - Montague semantics: using dcgs to compute sentence meanings
  - resolution theorem-proving and decidable fragments.

• Good luck in the last lab and in the exam!