The raison d’être of logic is the study of inference in language.

However, modern logic was developed in connection with the foundations of mathematics.

So we have a mismatch, leading to
— neglect of language in the first place
— use of first-order logic and no other tools

First-order logic is both too big and too small:
— cannot handle many interesting phenomena
— is undecidable
**Program**

Show that significant parts of natural language inference can be carried out in **decidable** logical systems.

Whenever possible, to obtain **complete axiomatizations**, because the resulting logical systems are likely to be interesting.

To be completely mathematical and hence to work using all tools and to make connections to fields like **complexity theory**, **(finite) model theory**, **decidable fragments of first-order logic**, and **algebraic logic**.
Natural Logic: parallel studies
I won’t have much to say on these, but you can ask me about them

- History of logic: reconstruction of original ideas
- Philosophy of language: proof-theoretic semantics
- Philosophy of logic: why variables?
- Cognitive science: models of human reasoning
- Linguistic semantics: Are deep structures necessary, or can we just use surface forms? And is a complete logic a semantics?
- Computational linguistics/artificial intelligence: many precursors
**Syntax:** Start with a collection of unary atoms (for nouns). Then the sentences are the expressions

\[ \text{All } p \text{ are } q \]

**Semantics:** A model \( \mathcal{M} \) is a set \( M \), together with an interpretation \([^p] \subseteq M\) for each noun \( p \).

\[ \mathcal{M} \models \text{All } p \text{ are } q \iff [p] \subseteq [q] \]

**Proof system is based on the following rules:**

\[
\frac{\text{All } p \text{ are } p \quad \text{All } p \text{ are } n \quad \text{All } n \text{ are } q}{\text{All } p \text{ are } q}
\]
If $\Gamma$ is a set of sentences, we write $\mathcal{M} \models \Gamma$ if for all $\varphi \in \Gamma$, $\mathcal{M} \models \varphi$.

$\Gamma \models \varphi$ means that every $\mathcal{M} \models \Gamma$ also has $\mathcal{M} \models \varphi$.

A proof tree over $\Gamma$ is a finite tree $T$ whose nodes are labeled with sentences, and each node is either an element of $\Gamma$, or comes from its parent(s) by an application of one of the rules.

$\Gamma \vdash \varphi$ means that there is a proof tree $T$ for over $\Gamma$ whose root is labeled $\varphi$. 

The simplest completeness theorem in logic

If \( \Gamma \models All\ p\ are\ q \), then \( \Gamma \vdash All\ p\ are\ q \)

Suppose that \( \Gamma \models All\ p\ are\ q \).

Build a model \( \mathcal{M} \), taking \( M \) to be the set of variables.

Define \( u \leq v \) to mean that \( \Gamma \vdash All\ u\ are\ v \).
The semantics is \( \llbracket u \rrbracket = \downarrow u \).
Then \( \mathcal{M} \models \Gamma \).
Hence for the \( p \) and \( q \) in our statement, \( \llbracket p \rrbracket \subseteq \llbracket q \rrbracket \).

But by reflexivity, \( p \in \llbracket p \rrbracket \).
And so \( p \in \llbracket q \rrbracket \); this means that \( p \leq q \).

But this is exactly what we want:
\( \Gamma \vdash All\ p\ are\ q \).
Syntax: All \( p \) are \( q \), Some \( p \) are \( q \)

Semantics: A model \( \mathcal{M} \) is a set \( M \), and for each noun \( p \) we have an interpretation \( [p] \subseteq M \).

\[
\begin{align*}
\mathcal{M} \models \text{All } p \text{ are } q & \quad \text{iff} \quad [p] \subseteq [q] \\
\mathcal{M} \models \text{Some } p \text{ are } q & \quad \text{iff} \quad [p] \cap [q] \neq \emptyset
\end{align*}
\]

Proof system:

- \( \text{All } p \text{ are } p \)
- \( \text{All } p \text{ are } n \)
- \( \text{All } n \text{ are } q \)
- \( \text{All } p \text{ are } q \)
- \( \text{All } p \text{ are } q \)
- \( \text{All } p \text{ are } q \)
- \( \text{All } q \text{ are } n \)
- \( \text{Some } p \text{ are } q \)
- \( \text{Some } q \text{ are } p \)
- \( \text{Some } p \text{ are } p \)
- \( \text{Some } q \text{ are } p \)
- \( \text{Some } p \text{ are } q \)
- \( \text{Some } p \text{ are } q \)
- \( \text{Some } p \text{ are } n \)
If there is an $n$, and if all $n$ are $p$ and also $q$, then some $p$ are $q$.

Some $n$ are $n$, All $n$ are $p$, All $n$ are $q$ $\vdash$ Some $p$ are $q$.

The proof tree is

\[
\begin{array}{c}
\text{All } n \text{ are } p \\
\text{Some } n \text{ are } n \\
\hline
\text{Some } n \text{ are } p \\
\text{All } n \text{ are } q \\
\text{Some } p \text{ are } n \\
\hline
\text{Some } p \text{ are } q
\end{array}
\]
Beyond first-order logic: cardinality

Read $\exists \geq (X, Y)$ as “there are at least as many $X$s as $Y$s”.

\[
\begin{align*}
\text{All } Y \text{ are } X & \quad \exists \geq (X, Y) \quad \exists \geq (Y, Z) \\
\exists \geq (X, Y) & \quad \exists \geq (X, Z)
\end{align*}
\]

\[
\begin{align*}
\text{All } Y \text{ are } X & \quad \exists \geq (Y, X) \\
\exists \geq (Y, X) & \quad \text{All } X \text{ are } Y
\end{align*}
\]

\[
\begin{align*}
\text{Some } Y \text{ are } Y & \quad \exists \geq (X, Y) \\
\exists \geq (X, Y) & \quad \text{No } Y \text{ are } Y
\end{align*}
\]

Some $X$ are $X$ 

The point here is that by working with a weak basic system, we can go beyond the expressive power of first-order logic.
Let us add complemented atoms $\overline{p}$ on top of the language of **All** and **Some**, with interpretation via set complement: $[\overline{p}] = M \setminus [p]$.

So we have

$$
\begin{align*}
S & \quad \begin{cases}
\text{All } p \text{ are } q \\
\text{Some } p \text{ are } q \\
\text{All } p \text{ are } \overline{q} \equiv \text{No } p \text{ are } q \\
\text{Some } p \text{ are } \overline{q} \equiv \text{Some } p \text{ aren’t } q \\
\text{Some non-}p \text{ are non-}q
\end{cases} \\
S^\dagger & \quad \begin{cases}
\end{cases}
\end{align*}
$$
The logical system for $S^+$

- **All p are p**
  - **Some p are q**
  - **Some p are p**

- **All p are n**
  - **All q are q**
  - **All p are q**

- **All n are p**
  - **Some n are q**
  - **Some p are q**

- **All q are $\neg q$**
  - **All q are p**
  - **Zero**

- **All p are $\neg q$**
  - **All q are $\neg p$**
  - **Antitome**

- **Some p are $\neg p$**
  - **$\phi$**
  - **Ex falso quodlibet**

- **All q are q**
  - **All p are q**
  - **One**
The system uses

\[
\frac{\text{Some } p \text{ are } \overline{p}}{\varphi} \quad \text{Ex falso quodlibet}
\]

and this is prima facie weaker than \textit{reductio ad absurdum}.

One of the logical issues in this work is to determine exactly where various principles are needed.
Completeness via representation of orthoposets

**Definition**

An orthoposet is a tuple \((P, \leq, 0, \prime)\) such that

- **POSET** \(\leq\) is a reflexive, transitive, and antisymmetric relation on the set \(P\).
- **ZERO** \(0 \leq p\) for all \(p \in P\).
- **ANTITONE** If \(x \leq y\), then \(y' \leq x'\).
- **INVOLUTIVE** \(x'' = x\).
- **INCONSISTENCY** If \(x \leq y\) and \(x \leq y'\), then \(x = 0\).

**A Key Point**

Orthoposets need not have a meet or join operation.
Ortho Posets: Two Examples

Example
For all sets $X$ we have an orthoposet $(\mathcal{P}(X), \subseteq, \emptyset, ',)$, where $a' = X \setminus a$ for all subsets $a$ of $X$.

Example

$$(x')' = x, \ 0' = 1, \ 1' = 0.$$
Orthopoosets: two examples

**Example**

For all sets $X$ we have an orthoposet $(\mathcal{P}(X), \subseteq, \emptyset, ')$, where $a' = X \setminus a$ for all subsets $a$ of $X$.

**Example**

\[ \begin{array}{c}
1 \\
p \\
\quad p' \\
\quad q \\
\quad q' \\
0 \\
\end{array} \]

$(x')' = x$, $0' = 1$, $1' = 0$.

**The idea**

Boolean algebra \[\frac{\text{propositional logic}}{\text{logic of All, Some and '}}\] = orthopooset

The details concerning completeness are somewhat different, and the whole thing would take about 10 minutes.
We have discussed these

\[ S \]

\[ S^\dagger \]

\[ S^\geq \]

\[ \text{monadic FOL} \]

\[ \text{2 variable fragment} \]

\[ \dagger \text{ adds full } N\text{-negation} \]
How about verbs?

We have discussed these relational syllogistic next.

† adds full N-negation
The next language uses “see” or $r$ as variables for transitive verbs.

<table>
<thead>
<tr>
<th>All $p$ are $q$</th>
<th>All $p$ aren’t $q$ $\equiv$ No $p$ are $q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Some $p$ are $q$</td>
<td>Some $p$ aren’t $q$</td>
</tr>
<tr>
<td>All $p$ see all $q$</td>
<td>All $p$ don’t see all $q$ $\equiv$ No $p$ sees any $q$</td>
</tr>
<tr>
<td>All $p$ see some $q$</td>
<td>All $p$ don’t see some $q$ $\equiv$ No $p$ sees all $q$</td>
</tr>
<tr>
<td>Some $p$ see all $q$</td>
<td>Some $p$ don’t see any $q$</td>
</tr>
<tr>
<td>Some $p$ see some $q$</td>
<td>Some $p$ don’t see some $q$</td>
</tr>
</tbody>
</table>

The interpretation is the natural one, using the subject wide scope readings in the ambiguous cases.

This is $\mathcal{R}$.
(The first system of its kind was Nishihara, Morita, Iwata 1990.)

The language $\mathcal{R}^\dagger$ has complemented atoms $\overline{p}$ on top of $\mathcal{R}$. 
Towards the syntax for $\mathcal{R}$

joint work with Ian Pratt-Hartmann

All p are q $\forall(p, q)$
Some p are q $\exists(p, q)$

All p r all q $\forall(p, \forall(q, r))$
All p r some q $\forall(p, \exists(q, r))$
Some p r all q $\exists(p, \forall(q, r))$
Some p r some q $\exists(p, \exists(q, r))$
No p are q $\forall(p, \neg q)$
Some p aren’t q $\exists(p, \neg q)$
All p don’t r all q $\equiv$
No p r any q $\forall(p, \forall(q, \neg r))$
All p don’t r some q $\equiv$
No p r all q $\forall(p, \exists(q, \neg r))$
Some p don’t r any q $\exists(p, \forall(q, \neg r))$
Some p don’t r some q $\exists(p, \exists(q, \neg r))$
Towards the syntax for $R$

Joint work with Ian Pratt-Hartmann

- All $p$ are $q$: $\forall(p, q)$
- Some $p$ are $q$: $\exists(p, q)$

- All $p$ $r$ all $q$: $\forall(p, \forall(q, r))$
- All $p$ $r$ some $q$: $\forall(p, \exists(q, r))$
- Some $p$ $r$ all $q$: $\exists(p, \forall(q, r))$
- Some $p$ $r$ some $q$: $\exists(p, \exists(q, r))$
- No $p$ are $q$: $\forall(p, \overline{q})$
- Some $p$ aren’t $q$: $\exists(p, \overline{q})$
- No $p$ $r$ any $q$: $\forall(p, \forall(q, \overline{r}))$
- No $p$ $r$ all $q$: $\forall(p, \exists(q, \overline{r}))$
- Some $p$ don’t $r$ any $q$: $\exists(p, \forall(q, \overline{r}))$
- Some $p$ don’t $r$ some $q$: $\exists(p, \exists(q, \overline{r}))$

Set terms $c$

<table>
<thead>
<tr>
<th>$p$</th>
<th>positive</th>
<th>$\forall(p, r)$</th>
<th>$\exists(p, r)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\overline{p}$</td>
<td>negative</td>
<td>$\exists(p, \overline{r})$</td>
<td>$\forall(p, \overline{r})$</td>
</tr>
</tbody>
</table>
\( \forall(p, r) \) those who \( r \) all \( p \)

\( \exists(p, r) \) those who \( r \) some \( p \)

\( \forall(p, \bar{r}) \) those who fail-to-\( r \) all \( p \approx \) those who \( r \) no \( p \)

\( \exists(p, \bar{r}) \) those who fail-to-\( r \) some \( p \approx \) those who don’t \( r \) some \( p \)
Towards the syntax for $\mathcal{R}$

<table>
<thead>
<tr>
<th></th>
<th>( \forall(p, q) )</th>
<th>( \exists(p, q) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>All p are q</td>
<td>( \forall(p, q) )</td>
<td></td>
</tr>
<tr>
<td>Some p are q</td>
<td>( \exists(p, q) )</td>
<td></td>
</tr>
<tr>
<td>All p r all q</td>
<td>( \forall(p, \forall(q, r)) )</td>
<td></td>
</tr>
<tr>
<td>All p r some q</td>
<td>( \forall(p, \exists(q, r)) )</td>
<td></td>
</tr>
<tr>
<td>Some p r all q</td>
<td>( \exists(p, \forall(q, r)) )</td>
<td></td>
</tr>
<tr>
<td>Some p r some q</td>
<td>( \exists(p, \exists(q, r)) )</td>
<td></td>
</tr>
<tr>
<td>No p are q</td>
<td>( \forall(p, \overline{q}) )</td>
<td></td>
</tr>
<tr>
<td>Some p aren’t q</td>
<td>( \exists(p, \overline{q}) )</td>
<td></td>
</tr>
<tr>
<td>No p sees any q</td>
<td>( \forall(p, \forall(q, \overline{r})) )</td>
<td></td>
</tr>
<tr>
<td>No p sees all q</td>
<td>( \forall(p, \exists(q, \overline{r})) )</td>
<td></td>
</tr>
<tr>
<td>Some p don’t r any q</td>
<td>( \exists(p, \forall(q, \overline{r})) )</td>
<td></td>
</tr>
<tr>
<td>Some p don’t r some q</td>
<td>( \exists(p, \exists(q, \overline{r})) )</td>
<td></td>
</tr>
</tbody>
</table>

simplifies to

\[ \forall(p, c) \quad \exists(p, c) \]

set terms $c$

<table>
<thead>
<tr>
<th></th>
<th>positive</th>
<th>negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p )</td>
<td>( \forall(p, r) )</td>
<td>( \exists(p, r) )</td>
</tr>
<tr>
<td>( \overline{p} )</td>
<td>( \exists(p, \overline{r}) )</td>
<td>( \forall(p, \overline{r}) )</td>
</tr>
</tbody>
</table>
We start with one collection of unary atoms (for nouns) and another of binary atoms (for transitive verbs).

<table>
<thead>
<tr>
<th>expression</th>
<th>variables</th>
<th>syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>unary atom</td>
<td>$p, q$</td>
<td>$p$</td>
</tr>
<tr>
<td>binary atom</td>
<td>$r$</td>
<td>$p$</td>
</tr>
<tr>
<td>positive set term</td>
<td>$c^+$</td>
<td>$p$</td>
</tr>
<tr>
<td>set term</td>
<td>$c, d$</td>
<td>$p$</td>
</tr>
<tr>
<td>$\mathcal{R}$ sentence</td>
<td>$\varphi$</td>
<td>$\forall(p, c)$</td>
</tr>
<tr>
<td>$\mathcal{R}^\dagger$ sentence</td>
<td>$\varphi$</td>
<td>$\forall(p, c)$</td>
</tr>
</tbody>
</table>
We need one last concept, syntactic negation:

<table>
<thead>
<tr>
<th>expression</th>
<th>syntax</th>
<th>negation</th>
</tr>
</thead>
<tbody>
<tr>
<td>positive set term $c$</td>
<td>$p$</td>
<td>$\bar{p}$</td>
</tr>
<tr>
<td></td>
<td>$\bar{p}$</td>
<td>$p$</td>
</tr>
<tr>
<td></td>
<td>$\exists(p, r)$</td>
<td>$\forall(p, \bar{r})$</td>
</tr>
<tr>
<td></td>
<td>$\forall(p, r)$</td>
<td>$\exists(p, \bar{r})$</td>
</tr>
<tr>
<td></td>
<td>$\exists(p, \bar{r})$</td>
<td>$\forall(p, r)$</td>
</tr>
<tr>
<td></td>
<td>$\forall(p, \bar{r})$</td>
<td>$\exists(p, r)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\mathcal{R}$ sentence $\varphi$</th>
<th>$\forall(p, c)$</th>
<th>$\exists(p, \bar{c})$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\exists(p, c)$</td>
<td>$\forall(p, \bar{c})$</td>
</tr>
</tbody>
</table>

Note that $\bar{p} = p$, $\bar{c} = c$ and $\bar{\varphi} = \varphi$. 
Theorem

There are no finite syllogistic logical systems which are sound and complete for $\mathcal{R}$.

However, there is a logical system (presented below) which uses reductio ad absurdum

\[
\begin{align*}
&\vdash \phi \\
&\vdots \\
&\exists(p, \overline{p}) \\
&\overline{\phi} \quad \text{RAA}
\end{align*}
\]

and which is complete.
There are no finite syllogistic logical systems which are sound and complete for $R$.

However, there is a logical system (presented below) which uses reductio ad absurdum

$$\begin{align*}
\exists(p, \overline{p}) \\
\implies(p, \overline{p}) \quad \text{RAA}
\end{align*}$$

and which is complete.

There are no finite, sound and complete syllogistic logical systems for $R^\dagger$, even ones which allow RAA.
The Aristotle Boundary

\[ \text{Aristotle} \quad \xrightarrow{\text{Church-Turing}} \quad \text{FOL} \quad \xrightarrow{\text{relational syllogistic}} \quad \text{FO}^2 \quad \xrightarrow{\dagger \text{ adds full } N\text{-negation}} \]
Relational syllogistic logic

$p$ and $q$ range over unary atoms, $c$ over set terms, and $t$ over binary atoms or their negations.

\[
\begin{align*}
\exists(p, q) \quad \forall(q, c) & \quad \exists(p, c) \\
\forall(p, q) \quad \exists(p, c) & \quad \exists(q, c) \\
\forall(q, \bar{c}) \quad \exists(p, c) & \quad \exists(p, \bar{c}) \\
\forall(p, \forall(n, t)) \quad \exists(q, n) & \quad \forall(p, \exists(q, t)) \\
\forall(p, \exists(q, t)) \quad \forall(q, n) & \quad \exists(p, \exists(n, t)) \\
\forall(p, \exists(q, t)) \quad \forall(q, n) & \quad \exists(p, \bar{p}) \quad \varphi \quad \text{RAA}
\end{align*}
\]
Most are monotonicty principles

\[ \exists(p^{\uparrow}, q^{\uparrow}) \quad \forall(p^{\downarrow}, q^{\uparrow}) \]
\[ \exists(p^{\uparrow}, \forall(q^{\downarrow}, t)) \quad \exists(p^{\uparrow}, \exists(q^{\uparrow}, t)) \]
\[ \forall(p^{\downarrow}, \forall(q^{\downarrow}, t)) \quad \forall(p^{\downarrow}, \exists(q^{\uparrow}, t)) \]

Plus also

\[ \forall(p, p) \quad \exists(p, c) \quad \forall(p, \bar{p}) \quad \exists(p, \exists(q, t)) \]
\[ \exists(p, p) \quad \forall(p, c) \quad \forall(q, q) \]

\[ \forall(q, \bar{c}) \quad \exists(p, c) \quad (\star) \quad \forall(p, \forall(n, t)) \quad \exists(q, n) \]
\[ \exists(p, \bar{q}) \quad \forall(p, \exists(q, t)) \]
\[ \forall(p, \exists(q, t)) \]

Of these, \((\star)\) is the most interesting.
Most are monotonicity principles

\[ \exists (p^\uparrow, q^\uparrow) \quad \forall (p^\downarrow, q^\uparrow) \]
\[ \exists (p^\uparrow, \forall (q^\downarrow, t)) \quad \exists (p^\uparrow, \exists (q^\uparrow, t)) \]
\[ \forall (p^\downarrow, \forall (q^\downarrow, t)) \quad \forall (p^\downarrow, \exists (q^\uparrow, t)) \]

I should mention that I had a hard time with this talk in deciding whether to only talk about monotonicity and its relation to categorial grammar, generalized quantifiers, and other areas.

Relevant papers:

van Benthem (2007) and earlier
Sanchez Valencia (1991)
van Eijck (2007)
Zamansky, Francez, and Winter (2006)
What do you think? Sound or unsound?

\[ \text{All } X \text{ see all } Y, \text{ All } X \text{ see some } Z, \text{ All } Z \text{ see some } Y \]
\[ \models \text{ All } X \text{ see some } Y \]
What do you think? Sound or unsound?

$$\forall X \text{ see all } Y, \forall X \text{ see some } Z, \forall Z \text{ see some } Y$$

$$\vdash \forall X \text{ see some } Y$$

The conclusion does indeed follow:
take cases as to whether or not there are \(Z\).

We should have a formal proof.
Example of a proof in this system

\[
\begin{align*}
    & \text{All } X \text{ see all } Y, \text{ All } X \text{ see some } Z, \text{ All } Z \text{ see some } Y \\
    \equiv & \quad \text{All } X \text{ see some } Y \\
    \text{Some } X \text{ see no } Y \quad \text{Some } X \text{ see some } Z \\
    \text{Some } X \text{ are } X \quad \text{All } X \text{ see some } Z \\
    \text{Some } Z \text{ see some } Z \quad \text{All } Z \text{ see some } Y \\
    \text{Some } Z \text{ are } Z \\
    \text{Some } Z \text{ see some } Y \\
    \text{Some } Y \text{ are } Y \\
    \text{All } X \text{ see some } Y \quad \text{All } X \text{ see all } Y \\
    \text{Some } X \text{ aren't } X \\
\end{align*}
\]
Some X see no Y

Some X are X, All X see some Z

Some X see some Z

Some Z are Z, All Z see some Y

Some Z see some Y

Some Y are Y, All X see all Y

All X see some Y

All X see all Y, All X see some Z, All Z see some Y ⊢ All X see some Y
Again, $\mathcal{R}$ has no pure syllogistic proof system. But it has an \textit{indirect} system (one using RAA).

With a lot more work, one can show that $\mathcal{R}^\dagger$ doesn’t even have an indirect system!

The arguments are reminiscent of arguments in \textit{finite model theory}, but without the boolean connectives there are many differences.
Next: relative clauses

\[ \text{FOL} \]

\[ \text{FO}^2 \]

\[ \mathcal{R}^{\dagger*} \]

\[ \mathcal{R}^{\dagger} \]

\[ \mathcal{R} \]

\[ S^{\dagger} \]

\[ S \]

† adds full $N$-negation

Add relative clauses = relativized quantifiers
Inference with relative clauses

What do you think about this one?

All skunks are mammals

All who fear all who respect all skunks fear all who respect all mammals
Inference with relative clauses

It follows, using an interesting antitonicity principle:

\[
\begin{align*}
    \text{All skunks are mammals} \\
    \text{All who respect all mammals respect all skunks}
\end{align*}
\]
Inference with relative clauses

It follows, using an interesting antitonicity principle:

- All skunks are mammals
- All who respect all mammals respect all skunks
- All who fear all who respect all skunks fear all who respect all mammals
$\mathcal{R}^*$ allows sentential subjects to be noun phrases containing subject relative clauses.

- who $r$ all p
- who don’t $r$ all p
- who $r$ some p
- who don’t $r$ any p

---

<table>
<thead>
<tr>
<th>expression</th>
<th>syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{R}^*$ sentence</td>
<td>$\forall(d^+, c) \mid \exists(d^+, c)$</td>
</tr>
<tr>
<td>$\mathcal{R}^{\dagger*}$ sentence</td>
<td>$\forall(d, c) \mid \exists(d, c)$</td>
</tr>
</tbody>
</table>

$d^+$ is a positive set term, and $c$ is an arbitrary set term.
Syllogistic logic for $\mathcal{R}^*$

\[
\begin{align*}
\forall(p, q) & \quad \forall(\forall(q, r), \forall(p, r)) \\
\forall(p, q) & \quad \forall(\exists(p, r), \exists(q, r)) \\
\exists(p, q) & \quad \forall(\forall(p, r), \exists(q, r))
\end{align*}
\]

These rules are based on McAllester and Givan (1992).

The remaining rules for $\mathcal{R}^*$ are generalizations of the $\mathcal{R}$ rules to the bigger syntax.
In a variant of this language which admits iterated relative clauses, we would just have

\[ \forall(s, m) \vdash \forall(\forall(s, r), f), \forall(\forall(m, r), f), \]

\[
\frac{\forall(s, m) \\
\forall(\forall(m, r), \forall(s, r)) \\
\forall(\forall(s, r), f), \forall(\forall(m, r), f))}
\]

\[ \frac{\forall(\forall(s, r), f), \forall(\forall(m, r), f))}
\]
\(\mathcal{R}^\dagger\) and \(\mathcal{R}^{\dagger*}\) lie beyond the Aristotle boundary, due to full negation on nouns.

It is possible to formulate a logical system with a restricted notion of variables, prove completeness, and yet stay inside the Turing boundary.

It’s a fairly involved definition, so I’ve hidden the details to slides after the end of the talk.

Instead, I’ll show examples.
Example of a proof in the system

From all keys are old items,
infer everyone who owns a key owns an old item
Example of a proof in the system

From all keys are old items, infer everyone who owns a key owns an old item

1. $\forall (key, old\text{-}item)$  
   hyp

2. $\exists (key, own)(x)$  
   hyp

3. $key(y)$  
   $\exists E, 2$

4. $own(x, y)$  
   $\exists E, 2$

5. $old\text{-}item(y)$  
   $\forall E, 1, 3$

6. $\exists (old\text{-}item, own)(x)$  
   $\exists I, 4, 5$

7. $\forall (\exists (key, own), \exists (old\text{-}item, own))$  
   $\forall I, 1–6$
1. John is a man
   Hyp

2. Any woman is a mystery to any man
   Hyp

3. Jane
   Jane is a woman
   Hyp

4. Any woman is a mystery to any man
   R, 2

5. Jane is a mystery to any man
   Any Elim, 4

6. John is a man
   R, 1

7. Jane is a mystery to John
   Any Elim, 6

8. Any woman is a mystery to John
   Any intro, 3, 7
For these logics, one can prove completeness by a Henkin-style argument.

The easiest way to prove decidability would be via the
- finite model property: use filtration from modal logic
- embedding into $FO^2$
- embedding into boolean modal logic (better complexity)
- results on resolution in Pratt-Hartmann 2004 (better complexity)

Also, there is a lower bound using $K + universal modality$.

The upshot: the validity problem is complete for exponential time.
**Next: Comparative Adjectives**

*Used for inferences involving phrases like* **Bigger than some kitten**

Grädel, Otto, Rosen 1999

† adds full *N*-negation

* adds relative clauses

*tr* adds comparatives, requiring transitivity
Comparative adjectives

Every giraffe is taller than every gnu
Some gnu is taller than every lion
Some lion is taller than some zebra
Every giraffe is taller than some zebra

We extend \( R^* \) to a language \( R^*(tr) \) by taking a set \( A \) of comparative adjective phrases in the base.

In the semantics, we would require of a model that for \( a \in A \), \([a] \) must be a transitive relation. (At the end of the talk we’ll see irreflexivity.)
Comparative adjectives

Every giraffe is taller than every gnu
Some gnu is taller than every lion
Some lion is taller than some zebra
Every giraffe is taller than some zebra

\[
\forall(p, \exists(q, r)) \Rightarrow \exists(p, \forall(q, r)) \Rightarrow \forall(p, \forall(q, r))
\]

\[
\forall(p, \exists(q, r)) \Rightarrow \exists(p, \forall(q, r)) \Rightarrow \forall(p, \forall(q, r))
\]
Comparative adjectives

Every giraffe is taller than every gnu
Some gnu is taller than every lion
Some lion is taller than some zebra
Every giraffe is taller than some zebra

\[ \forall \text{giraffe}, \forall (\text{gnu, taller}) \quad \exists (\text{gnu, } \forall (\text{lion, taller})) \]

\[ \forall (\text{giraffe, } \forall (\text{lion, taller})) \quad \exists (\text{lion, } \exists (\text{zebra, taller})) \]

\[ \forall (\text{giraffe, } \exists (\text{zebra, taller})) \]
We begin with the logical system for $\mathcal{R}^{\dagger*}$, and then we add a rule:

$$\frac{a(x, y) \quad a(y, z)}{a(x, z)} \quad \text{trans}$$

This rule is added for all $a \in A$, and all $x, y, z$.

This gives a language $\mathcal{R}^{\dagger*}(tr)$. 
Every sweet fruit is bigger than every kumquat

Every fruit bigger than some sweet fruit is bigger than every kumquat

\[
\begin{align*}
\exists (sw, bigger) & \quad \forall (kq, bigger) (y) \\
\forall (kq, bigger) & \quad \forall (kq, bigger) (y) \\
\forall (kq, bigger) & \quad \forall (kq, bigger) (x) \\
\forall (\exists (sw, bigger), \forall (kq, bigger)) & \quad \forall (\exists (sw, bigger), \forall (kq, bigger))
\end{align*}
\]
We want to account for inferences such as

Frege's favorite food was sushi
Frege ate sushi at least once
We want to account for inferences such as

\[
\text{Frege's favorite food was sushi} \\
\text{Frege ate sushi at least once}
\]

The hypothesis and conclusion would be rendered in some logical system or other. There would be a background theory (\(\approx\) common sense), and then the inference would be modeled either as a semantic fact:

\[
\text{Common sense} + \text{Frege's favorite food was sushi} \models \text{Frege ate sushi at least once}
\]

or a via a formal deduction:

\[
\text{Common sense} + \text{Frege's favorite food was sushi} \vdash \text{Frege ate sushi at least once}
\]

Either way, it’s all in one and the same language.
Transitivity should not be treated as a meaning postulate, since even stating it would seem to render the logic undecidable.

Instead, it is a proof rule:

\[
\begin{array}{c}
a(x, y) \\
a(y, z)
\end{array}
\Rightarrow
\begin{array}{c}
a(x, z)
\end{array} \quad \text{trans}
\]

(I have not proved that one can’t formulate a decidable logic which can directly express transitivity using variables and also cover the sentences we’ve seen. But there are results that suggest it.)
Next: relational converses

used for inferences relating bigger and smaller

\[ FOL \]

\[ FO^2 + \text{trans} \]

\[ FO^2 \]

\[ \mathcal{R}^{\dagger}(tr) \]

\[ \mathcal{R}^{\dagger} \]

\[ \mathcal{R}^* \]

\[ S^{\dagger} \]

\[ S \]

\[ \mathcal{R}^{\dagger*}(tr, opp) \]

\[ \mathcal{R}^{\dagger*} \]

\[ \mathcal{R}^*(tr) \]

\[ \mathcal{R}^* \]

\[ \mathcal{R}^*(tr, opp) \]

\[ \dagger \text{ adds full } N\text{-negation} \]

\[ * \text{ adds relative clauses} \]

\[ opp \text{ adds opposites of comparative adjectives} \]
Converses of transitive relations

On top of all the other syllogistic systems we have seen

∀(p, ∀(q, t))
∀(q, ∀(p, t⁻¹))

∃(∃(p, r⁻¹), ∃(q, r))
∃(p, ∃(q, r))

∀(p, ∃(q, r))
∀(∃(p, r⁻¹), ∃(n, r))

∀(p, ∃(q, r))
∀(∃(p, r⁻¹), ∃(n, r))

∀(p, ∃(q, r))
∀(∃(p, r⁻¹), ∀(n, r))

∀(p, ∀(n, r))

(scope): if some p is bigger than all q,
then all q are smaller than some p or other.

(⋆): if every dog is bigger than some hedgehog,
and everything smaller than some dog is bigger than some cat,
then every dog is bigger than some cat.
first-order logic

$FO^2 + "R \text{ is trans}"$

2 variable FO logic

† adds full $N$-negation

$R^{\dagger}(tr) + \text{opposites}$

$R^{\dagger} + (\text{transitive})$

$R + \text{relative clauses}$

$S + \text{full } N\text{-negation}$

$R = \text{relational syllogistic}$

$S^{\geq} \text{ adds } |p| \geq |q|$

$S: \text{ all/some/no } p \text{ are } q$
Complexity

(Mostly) Best possible results on the validity problem

- **FOL**
- **FO^2 + trans**
- **FO^2**
- **Aristotle**

- **Church-Turing**
- **\( R^+(tr) \)**
- **\( R^+(tr, opp) \)**
- **\( R^* \)**
- **\( S^+ \)**

- **\( BML(tr) \)**
- **\( R^+(tr) \)**
- **\( R^+ \)**

- **undecidable**
  - Church 1936
  - Grädel, Otto, Rosen 1999

- in co-NEXPTIME
  - EXPTIME
  - Lutz & Sattler 2001
  - Grädel, Kolaitis, Vardi '97
  - EXPTIME
  - Pratt-Hartmann 2004

- Co-NEXPTIME
  - lower bounds also open

- **Co-NP**
  - McAllester & Givan 1992

- **NLOGSPACE**
### Complexity sketches

*Again, joint with Ian Pratt-Hartmann*

<table>
<thead>
<tr>
<th>$S$</th>
<th>NLOGSPACE</th>
<th>lower bound via reachability problem for directed graphs</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S^+$</td>
<td>NLOGSPACE</td>
<td>upper bound via 2SAT</td>
</tr>
<tr>
<td>$R$</td>
<td>NLOGSPACE</td>
<td>upper bound takes special work based on the proof system</td>
</tr>
<tr>
<td>$R^+$</td>
<td>EXPTIME</td>
<td>lower bound via $K^U$, Hemaspaandra 1996</td>
</tr>
<tr>
<td>$R^{*+}$</td>
<td>EXPTIME</td>
<td>upper bound by Pratt-Hartmann 2004</td>
</tr>
<tr>
<td>$BML(tr)$</td>
<td>EXPTIME</td>
<td>Boolean modal logic on transitive models</td>
</tr>
<tr>
<td>$R^*$</td>
<td>Co-NPTIME</td>
<td>essentially in McAllester and Givan 1992</td>
</tr>
<tr>
<td>$FO^2$</td>
<td>NEXPTIME</td>
<td>Grädel, Kolaitis, and Vardi 1997</td>
</tr>
</tbody>
</table>
The finite model property: **Yes**↓ and **No**↑

Aristotle

Church-Turing

\[ \text{FO}^2 + \text{trans} \]

\[ \forall (p, \exists (p, r)) + \exists p \]

filtration of a Henkin model

Mortimer 1975

\[ \text{irr} \] means that comparative adjectives must have irreflexive interpretations.
Some $p$ are $q$
Some $p$ are not $q$
Some $q$ are not $p$
Every $p$ is smaller than some $q$
Every $q$ is smaller than some $p$

The relation in the model is the transitive closure of the arrows.
NATURAL LOGIC: WHAT I HOPE TO HAVE GOTTEN ACROSS

PROGRAM

Show that significant parts of natural language inference can be carried out in **decidable** logical systems.

Whenever possible, to obtain **complete axiomatizations**, because the resulting logical systems are likely to be interesting.

To be completely mathematical and hence to work using all tools and to make connections to fields like **complexity theory**, **(finite) model theory**, **decidable fragments of first-order logic**, and **algebraic logic**.
We must ask whether a complete proof system is a semantics.

We should not be afraid of doing logic beyond logic.

Joining the perspectives of semantics, complexity theory, proof theory, cognitive science, and computational linguistics should allow us to ask interesting questions and answer them.
Details on the proof system for $\mathcal{R}^+\ast$

<table>
<thead>
<tr>
<th>Expression</th>
<th>Variables</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>unary atom</td>
<td>$p, q$</td>
<td></td>
</tr>
<tr>
<td>binary atom</td>
<td>$s$</td>
<td></td>
</tr>
<tr>
<td>constant</td>
<td>$j, k$</td>
<td></td>
</tr>
<tr>
<td>unary literal</td>
<td>$l$</td>
<td>$p \mid \bar{p}$</td>
</tr>
<tr>
<td>binary literal</td>
<td>$r$</td>
<td>$s \mid \bar{s}$</td>
</tr>
<tr>
<td>set term</td>
<td>$b, c, d$</td>
<td>$l \mid \exists(c, r) \mid \forall(c, r)$</td>
</tr>
<tr>
<td>sentence</td>
<td>$\varphi, \psi$</td>
<td>$\forall(c, d) \mid \exists(c, d) \mid c(j) \mid r(j, k)$</td>
</tr>
</tbody>
</table>

Think of the constants as proper names: John, Mary, etc. the unary atoms as predicates like boys or girls, the binary atoms by transitive verbs such as likes and sees.
Recursion allows us to embed set terms, and so we have set terms like
\[ \exists (\forall (\forall (b, s), h), a) \]
which may be taken to symbolize
a verb phrase such as
admires someone who hates everyone who does not see any boy.

We should note that the relative clauses which can be obtained in this way are all “subject relatives”, never “object relatives”.

The language is too poor to express predicates like
\[ \lambda x. \text{all boys see } x. \]
Proof system: general sentences

General sentences in this fragment are what usually are called formulas.
We prefer to change the standard terminology to make the point that here, sentences are not built from formulas by quantification. Sentences in our sense do not have variable occurrences. But general sentences do allow variables.

<table>
<thead>
<tr>
<th>Expression</th>
<th>Variables</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>individual variable</td>
<td>$x, y$</td>
<td>$x \mid j$</td>
</tr>
<tr>
<td>individual term</td>
<td>$t, u$</td>
<td>$\varphi \mid c(t) \mid r(t, u) \mid \bot$</td>
</tr>
<tr>
<td>general sentence</td>
<td>$\alpha$</td>
<td></td>
</tr>
</tbody>
</table>

It will turn out that for this fragment, only two variables are needed.
We don't need general sentences of the form $r(j, x)$ or $r(x, j)$. 
Proof system: half of the rules

\[
\begin{align*}
\frac{c(t) \quad \forall(c, d)}{d(t)} & \quad \forall E \\
\frac{c(t) \quad d(t)}{\exists(c, d)} & \quad \exists I \\
\frac{c(u) \quad \forall(c, r)(t)}{r(t, u)} & \quad \forall E \\
\frac{r(t, u) \quad c(u)}{\exists(c, r)(t)} & \quad \exists I
\end{align*}
\]
Proof system: the second half of the rules

\[
\frac{\exists(c, d)}{\alpha} \exists E
\]

\[
\frac{\exists(c, r)(t)}{\alpha} \exists E
\]

\[
\frac{\alpha}{\alpha} \perp I
\]

\[
\frac{\alpha}{\alpha} \perp I
\]

\[
\frac{\alpha}{\alpha} \perp I
\]

\[
\frac{\alpha}{\alpha} \perp I
\]

\[
\frac{\alpha}{\alpha} \perp I
\]
Proof system: side conditions

In (\(\forall I\)), \(x\) must not occur free in any uncanceled hypothesis.

In (\(\exists E\)), the variable \(x\) must not occur free in the conclusion \(\alpha\) or in any uncanceled hypothesis in the subderivation of \(\alpha\).

In contrast to usual first-order natural deduction systems, there are no side conditions on the rules (\(\forall E\)) and (\(\exists I\)).