THREE FLAVORS OF NATURAL LOGIC:
EXTENDED SYLLOGISMS,
LOGICS WITH VARIABLES,
AND REASONING WITH POLARITIES

Larry Moss, Indiana University

CSLI Workshop on
Natural Logic, Proof Theory, and Computational Semantics
April 2011
Program

Re-think semantics based on computational linguistics.

Re-work the relation of logic and language, starting with inference.

**First step:** 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**, **proof theory**, **decidable fragments of first-order logic**, and **algebraic logic**.

But these are all the **first step**, and they hardly touch upon the real goals.
Johan van Benthem recalled the history of his pioneering work on natural logic in the 1980’s and 90’s.

The proposed ingredients of a logical system to satisfy his goals would consist of several “modules”:

(A) Monotonicity Reasoning, i.e., Predicate Replacement,
(B) Conservativity, i.e., Predicate Restriction, and also
(C) Algebraic Laws for inferential features of specific lexical items.

This talk is also concerned with three modules, but working a little differently.
Logics of the first two flavors

- First-order logic
  - \( FO^2 + "R \text{ is trans}" \)
- Two variable FO logic
  - \( RC^{\dagger}(tr, \text{opp}) \)
  - \( RC^{\dagger}(tr) \)
  - \( RC(tr) \)
  - \( RC^{\dagger} \)
  - \( \mathcal{R} \)
  - \( \mathcal{S} \)

\( \dagger \) adds full \( N \)-negation

- \( RC(tr) \) + opposites
- \( RC + (\text{transitive}) \) comparative adjs
- \( R + \text{relative clauses} \)
- \( S + \text{full} \ N \)-negation
- \( R = \text{relational syllogistic} \)
- \( S \gtrless \text{adds } |p| \geq |q| \)
- \( S: \text{all/some/no } p \text{ are } q \)
Purely syllogistic systems appear to be the simplest ones, but looks can be deceiving.
**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 \( \llbracket p \rrbracket \subseteq M \) for each noun \( p \).

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

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

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

\[
\frac{\text{All } p \text{ are } n \quad \text{All } n \text{ are } q}{\text{All } p \text{ are } q}
\]
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 $\mathcal{T}$ 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 $\mathcal{T}$ for over $\Gamma$ whose root is labeled $\varphi$. 
The simplest completeness theorem in logic

If $\Gamma \models \text{All } p \text{ are } q$, then $\Gamma \vdash \text{All } p \text{ are } q$

Suppose that $\Gamma \models \text{All } p \text{ are } q$.

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

Define $u \leq v$ to mean that $\Gamma \vdash \text{All } u \text{ are } v$.

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: $\Gamma \vdash \text{All } p \text{ are } q$. 
Syntactic Logic of All and Some

**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 $\langle [p] \rangle \subseteq M$.

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

$\mathcal{M} \models \text{Some } p \text{ are } q \iff \langle [p] \rangle \cap \langle [q] \rangle \neq \emptyset$

**Proof system:**

\[
\begin{array}{ccc}
\text{All } p \text{ are } p & \rightarrow & \text{All } p \text{ are } p \\
\text{All } p \text{ are } n & \rightarrow & \text{All } n \text{ are } q \\
\text{All } p \text{ are } q & \rightarrow & \text{All } p \text{ are } q
\end{array}
\]

\[
\begin{array}{ccc}
\text{Some } p \text{ are } q & \rightarrow & \text{Some } q \text{ are } p \\
\text{Some } p \text{ are } q & \rightarrow & \text{Some } p \text{ are } p \\
\text{All } q \text{ are } n & \rightarrow & \text{Some } p \text{ are } q
\end{array}
\]

\[
\begin{array}{ccc}
\text{All } q \text{ are } n & \rightarrow & \text{Some } p \text{ are } q \\
\text{All } q \text{ are } n & \rightarrow & \text{Some } p \text{ are } n
\end{array}
\]
**Example of a derivation**

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 $q$ ⊢ Some $p$ are $q$.

The proof tree is

\[
\begin{align*}
\text{All } n \text{ are } p & \quad \text{Some } n \text{ are } n \\
\text{Some } n \text{ are } p & \\
\text{All } n \text{ are } q & \quad \text{Some } p \text{ are } n \\
\text{Some } p \text{ are } q
\end{align*}
\]
Beyond first-order logic: cardinality

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

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

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

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

The point here is that by working with a weak basic system, we can go beyond the expressive power of first-order logic.
The languages $S$ and $S^\dagger$ add noun-level negation.

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

$$S = \left\{ \begin{array}{l}
    \text{All } p \text{ are } q \\
    \text{Some } p \text{ are } q \\
    \text{All } p \text{ are } \overline{q} \equiv \text{No } p \text{ are } q \\
    \text{Some } p \text{ are } \overline{q} \equiv \text{Some } p \text{ aren't } q \\
    \text{Some non}-p \text{ are non}-q
    \end{array} \right\}$$

$$S^\dagger$$
A syllogistic system for $S^\dagger$

But what is a syllogistic system?

<table>
<thead>
<tr>
<th>All $p$ are $p$</th>
<th>Some $p$ are $q$</th>
<th>Some $p$ are $q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Some $p$ are $p$</td>
<td>Some $p$ are $p$</td>
<td>Some $q$ are $p$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>All $p$ are $n$</th>
<th>All $n$ are $q$</th>
<th>All $n$ are $p$</th>
<th>Some $n$ are $q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>All $p$ are $q$</td>
<td>All $p$ are $q$</td>
<td>Some $p$ are $q$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>All $q$ are $\overline{q}$</th>
<th>Zero</th>
</tr>
</thead>
<tbody>
<tr>
<td>All $q$ are $p$</td>
<td>All $q$ are $p$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Antitone</th>
</tr>
</thead>
<tbody>
<tr>
<td>All $p$ are $\overline{q}$</td>
</tr>
<tr>
<td>All $q$ are $\overline{p}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ex falsa quodlibet</th>
</tr>
</thead>
</table>
Tautologies of classical propositional logic.

- All $p''$ are $p$
- All $p$ are $p''$
- All $p$ are $n$ \land (All $n$ are q) \rightarrow All $p$ are q
- All $p$ are $n$ \rightarrow All $n'$ are $p'$
- All $p$ are $p'$ \rightarrow All $p$ are q

Actually, this axiomatizes the boolean sentential closure of the syllogistic logic.

Incidentally, an implementation of the logic was provided by Rocha and Meseguer 2007 based on Dijkstra and Scholten 1990.
The system uses

\[
\begin{align*}
\text{Some } p & \text{ are } \neg p \\
\hline
\varphi & \text{ Ex falso quodlibet}
\end{align*}
\]

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.
Adding transitive verbs

The next language uses “see” or $r$ as variables for transitive verbs.

$\begin{align*}
\text{All } p & \text{ are } q \\
\text{Some } p & \text{ are } q \\
\text{All } p & \text{ see all } q \\
\text{All } p & \text{ see some } q \\
\text{Some } p & \text{ see all } q \\
\text{Some } p & \text{ see some } q
\end{align*}$

$\begin{align*}
\text{All } p & \text{ aren’t } q \equiv \text{ No } p \text{ are } q \\
\text{Some } p & \text{ aren’t } q \\
\text{All } p & \text{ don’t see all } q \equiv \text{ No } p \text{ sees any } q \\
\text{All } p & \text{ don’t see some } q \equiv \text{ No } p \text{ sees all } q \\
\text{Some } p & \text{ don’t see any } q \\
\text{Some } p & \text{ don’t see some } q
\end{align*}$

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 $\mathcal{R}^+$ has complemented atoms $\bar{p}$ on top of $\mathcal{R}$.
Adding transitive verbs

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 \[\exists(p, \neg q)\]
No p r any q \[\forall(p, \forall(q, \neg r))\]
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 \[\exists(p, \exists(q, \neg r))\]
### Adding transitive verbs

| 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 | $\exists(p, \exists(q, \overline r))$ |

<table>
<thead>
<tr>
<th>set terms c</th>
<th>positive</th>
<th>negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>$\forall(p, r)$</td>
<td>$\exists(p, \overline r)$</td>
</tr>
<tr>
<td>$\forall(p, r)$</td>
<td>$\exists(p, \overline r)$</td>
<td>$\forall(p, \overline r)$</td>
</tr>
</tbody>
</table>
### Reading the Set Terms

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \forall (p, r) )</td>
<td>those who ( r ) all ( p )</td>
</tr>
<tr>
<td>( \exists (p, r) )</td>
<td>those who ( r ) some ( p )</td>
</tr>
<tr>
<td>( \forall (p, \bar{r}) )</td>
<td>those who fail-to-( r ) all ( p ) ( \approx ) those who ( r ) no ( p )</td>
</tr>
<tr>
<td>( \exists (p, \bar{r}) )</td>
<td>those who fail-to-( r ) some ( p ) ( \approx ) those who don’t ( r ) some ( p )</td>
</tr>
</tbody>
</table>
Towards the syntax for $R$

- 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 $p$ sees any $q$: $\forall(p, \forall(q, \overline{r}))$
- No $p$ sees 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></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></td>
</tr>
<tr>
<td>binary atom</td>
<td>$r$</td>
<td></td>
</tr>
<tr>
<td>positive set term</td>
<td>$c^+$</td>
<td>$p \mid \exists (p, r) \mid \forall (p, r)$</td>
</tr>
<tr>
<td>set term</td>
<td>$c, d$</td>
<td>$p \mid \exists (p, r) \mid \forall (p, r)$</td>
</tr>
<tr>
<td>$\mathcal{R}$ sentence</td>
<td>$\varphi$</td>
<td>$\forall (p, c) \mid \exists (p, c)$</td>
</tr>
<tr>
<td>$\mathcal{R}^\dagger$ sentence</td>
<td>$\varphi$</td>
<td>$\forall (p, c) \mid \exists (p, c) \mid \forall (\bar{p}, c) \mid \exists (\bar{p}, c)$</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>$\bar{p}$</td>
<td>$p$</td>
<td>$\bar{p}$</td>
</tr>
<tr>
<td>$\exists(p, r)$</td>
<td>$\forall(p, \bar{r})$</td>
<td>$\forall(p, r)$</td>
</tr>
<tr>
<td>$\forall(p, r)$</td>
<td>$\exists(p, \bar{r})$</td>
<td>$\exists(p, r)$</td>
</tr>
<tr>
<td>$\exists(p, \bar{r})$</td>
<td>$\forall(p, r)$</td>
<td>$\forall(p, r)$</td>
</tr>
<tr>
<td>$\forall(p, \bar{r})$</td>
<td>$\exists(p, r)$</td>
<td>$\exists(p, r)$</td>
</tr>
<tr>
<td>$\mathcal{R}$ sentence $\varphi$</td>
<td>$\forall(p, c)$</td>
<td>$\exists(p, \bar{c})$</td>
</tr>
<tr>
<td>$\exists(p, c)$</td>
<td>$\forall(p, \bar{c})$</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

\[
\begin{align*}
[\varphi] \\
\vdots \\
\exists (p, \overline{p}) \\
\frac{\varphi}{\overline{\varphi}} \quad \text{RAA}
\end{align*}
\]

and which is complete.
**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

\[
\begin{align*}
\exists (p, \neg p) \\
\overline{\varphi} \\
\text{RAA}
\end{align*}
\]

and which is complete.

**Theorem**

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

† adds full $N$-negation

relational syllogistic
Relational syllogistic logic

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

\[
\begin{align*}
\exists(p, q) & \quad \forall(q, c) \\
\implies & \quad \exists(p, c) \\
\forall(p, q) & \quad \exists(p, c) \\
\implies & \quad \exists(q, c) \\
\forall(q, \bar{c}) & \quad \exists(p, c) \\
\implies & \quad \exists(p, \bar{q}) \\
\forall(p, \forall(n, t)) & \quad \exists(q, n) \\
\implies & \quad \forall(p, \exists(q, t)) \\
\forall(p, \exists(q, t)) & \quad \exists(q, n) \\
\implies & \quad \exists(p, \exists(n, t)) \\
\forall(p, \exists(q, t)) & \quad \forall(q, n) \\
\implies & \quad \forall(p, \exists(n, t)) \\
\forall(p, \exists(q, t)) & \quad \forall(q, n) \\
\implies & \quad \forall(p, \exists(n, t)) \\
& \vdots
\end{align*}
\]

$[\varphi]$
Example of a proof in the system for $R^+$

What do you think? Sound or unsound?

\[
\begin{align*}
\text{All } X \text{ see all } Y, \text{ All } X \text{ see some } Z, \text{ All } Z \text{ see some } Y \\
\models \text{ All } X \text{ see some } Y
\end{align*}
\]
Example of a proof in the system for $R^\dagger$

What do you think? Sound or unsound?

\[\text{All } X \text{ see all } Y, \text{ All } X \text{ see some } Z, \text{ All } Z \text{ see some } Y\]

\[\vdash \text{ All } X \text{ see some } Y\]

The conclusion does indeed follow:

take cases as to whether or not there are $X$.

We should have a formal proof.
All X see all Y, All X see some Z, All Z see some Y
\[ \vdash \text{All X see some Y} \]

Some X see no Y

Some X are X \[ \text{All X see some Z} \]

Some X see some Z

Some Z are Z \[ \text{All Z see some Y} \]

Some Z see some Y

Some Y are Y \[ \text{All X see all Y} \]

All X see some Y

Some X see no Y

Some X aren’t X
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
Next: Relative Clauses

- Aristotle
  - $S^\uparrow$
  - $R^\uparrow$
  - $RC$
  - $RC^\dagger$

- $FO^2$
- $FOL$
- Church-Turing

- $\dagger$ adds full $N$-negation
- Add relative clauses
  - = relativized quantifiers
What do you think about these?

\[
\text{All skunks are mammals} \\
\text{All who fear all who respect all skunks fear all who respect all mammals}
\]

\[
\text{All skunks are mammals} \\
\text{All who fear all who respect some skunks fear all who respect some mammals}
\]

\[
\text{All skunks are mammals} \\
\text{Some who fear all who respect some skunks fear some who respect some mammals}
\]
\( \text{RC} \) allows sentential subjects to be noun phrases containing subject relative clauses.

\[
\begin{align*}
\text{who r all p} & \quad \text{who r some p} \\
\text{who don’t r all p} & \quad \text{who don’t r any p}
\end{align*}
\]

<table>
<thead>
<tr>
<th>expression</th>
<th>syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{RC} ) sentence</td>
<td>( \forall(d^+, c) \mid \exists(d^+, c) )</td>
</tr>
<tr>
<td>( \text{RC}^\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.
The main rules are

\[
\frac{\forall(p, q)}{\forall(\forall(q, r), \forall(p, r))} \quad \frac{\forall(p, q)}{\forall(\exists(p, r), \exists(q, r))} \quad \frac{\exists(p, q)}{\forall(\forall(p, r), \exists(q, r))}
\]

These rules are based on McAllester and Givan (1992).
In a variant of this language which admits iterated relative clauses, we would just have

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

\[ \forall(s, m) \]
\[ \forall(\forall(m, r), \forall(s, r)) \]
\[ \forall(\forall(s, r), f), \forall(\forall(m, r), f)) \]
kissing involves touching

\[
\begin{align*}
\text{All skunks are mammals} \\
\text{All who fear all who touch all skunks fear all who kiss all skunks}
\end{align*}
\]

The point is that we incorporate the constraint into the proof theory, not as a meaning postulate.
Suppose that $r \Rightarrow s$

\[
\begin{align*}
\forall(d, \forall(c, r)) & \quad \forall(d, \exists(c, r)) & \quad \exists(d, \forall(c, r)) & \quad \exists(d, \exists(c, r)) \\
\forall(d, \forall(c, s)) & \quad \forall(d, \exists(c, s)) & \quad \exists(d, \forall(c, s)) & \quad \exists(d, \exists(c, s)) \\
\forall(\exists(c, r), \exists(c, s)) & \quad \forall(\forall(c, r), \forall(c, s))
\end{align*}
\]

We again have completeness in the relevant sense.
Next: Comparative Adjectives

Used for inferences involving phrases like BIGGER THAN SOME KITTEN.
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 $\mathcal{RC}$ to a language $\mathcal{RC}(tr)$ 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. (We could also require irreflexivity.)
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

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

∀(gir, ∀(gnu, taller))  ∃(gnu, ∀(lion, taller))
∀(gir, ∀(lion, taller))    ∃(lion, ∃(zebra, taller))
∀(giraffe, ∃(zebra, taller))
Next: Relational Converse

Used for inferences relating bigger and smaller

\[ FOL \]

\[ FO^2 + trans \]

\[ Church-Turing \]

\[ RC^{\dagger}(tr, opp) \]

\[ RC^{\dagger}(tr) \]

\[ RC \]

\[ RC(tr, opp) \]

\[ RC(tr) \]

\[ RC \]

\[ S^{\dagger} \]

\[ S \]

\[ S^{\dagger} \]

\[ S \]

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

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

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

On top of all the other syllogistic systems we have seen

\[ \forall(p, \forall(q, t)) \quad \frac{\exists(p, \forall(q, t))}{\forall(q, \exists(p, t^{-1}))} \quad (\text{scope}) \quad \forall(p, \exists(q, r^{-1})) \quad \frac{\forall(p, \forall(q, r^{-1}))}{\forall(q, \forall(p, r^{-1}))} \]

\[ \exists(\exists(p, r^{-1}), \exists(q, r)) \quad \frac{\exists(\forall(p, r), \forall(q, r^{-1}))}{\forall(p, \forall(q, r^{-1}))} \quad \exists(\forall(p, r), \exists(q, r^{-1})) \quad \frac{\exists(q, \forall(p, r^{-1}))}{\forall(q, \forall(p, r^{-1}))} \]

\[ \forall(p, \exists(q, r)) \quad \frac{\forall(\exists(p, r^{-1}), \exists(n, r))}{\forall(p, \exists(n, r))} \quad (\star) \quad \forall(p, \exists(q, r)) \quad \frac{\forall(\exists(p, r^{-1}), \forall(n, r))}{\forall(p, \forall(n, r))} \]

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

\text{(\star): 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.}
Logic beyond the Aristotle boundary

So far in this talk, all of the systems have been syllogistic to one degree or another.

$\mathcal{R}^\dagger$ and $\mathcal{RC}^\dagger$ 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 Church-Turing boundary.
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. $\forall(key, old-item)$ hyp
2. $\exists(key, own)(x)$ hyp
3. key(y) $\exists E, 2$
4. own(x, y) $\exists E, 2$
5. old-item(y) $\forall E, 1, 3$
6. $\exists(old-item, own)(x)$ $\exists I, 4, 5$
7. $\forall(\exists(key, own), \exists(old-item, own))$ $\forall I, 1–6$
1. John is a man (Hyp)
2. Any woman is a mystery to any man (Hyp)
3. 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)
Details on the proof system for $\mathcal{RC}^+$

<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>$p \mid \bar{p}$</td>
</tr>
<tr>
<td>binary atom</td>
<td>$s$</td>
<td>$s \mid \bar{s}$</td>
</tr>
<tr>
<td>constant</td>
<td>$j, k$</td>
<td></td>
</tr>
<tr>
<td>unary literal</td>
<td>$l$</td>
<td>$l \mid \exists (c, r) \mid \forall (c, r)$</td>
</tr>
<tr>
<td>binary literal</td>
<td>$r$</td>
<td>$r \mid \exists (c, k) \mid \forall (c, k)$</td>
</tr>
<tr>
<td>set term</td>
<td>$b, c, d$</td>
<td>$b \mid \forall (c, d) \mid \exists (c, d) \mid c(j) \mid r(j, k)$</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

\[ \exists (\forall (b, s), h), a) \]

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>(x \mid j)</td>
</tr>
<tr>
<td>general sentence</td>
<td>(\alpha)</td>
<td>(\varphi \mid c(t) \mid r(t, u) \mid \bot)</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

\[
\frac{c(t)}{d(t)} \quad \forall E
\]

\[
\frac{c(t)}{d(t)} \quad \exists I
\]

\[
\frac{c(u)}{r(t, u)} \quad \forall E
\]

\[
\frac{r(t, u)}{\exists (c, r)(t)} \quad \exists I
\]
**Proof System: The Second Half of the Rules**

\[
\begin{align*}
\forall (c, d) & \quad \forall I \\
\exists (c, d) & \quad \exists E \\
\perp & \quad \perp I
\end{align*}
\]
Proof system: side conditions

\[
\begin{align*}
[c(x)] & \vdash d(x) \quad \forall I \\
\forall(c, d) & \quad \forall I \\
\exists(c, d) & \quad \exists E \\
\alpha & \quad \alpha
\end{align*}
\]

\[
\begin{align*}
[c(x)] & \vdash r(t, x) \quad \forall I \\
\forall(c, r(t)) & \quad \forall I \\
\exists(c, r(t)) & \quad \exists E \\
\alpha & \quad \alpha
\end{align*}
\]

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 \((\exists I)\).
A WORD ON
COMPLETENESS/DECIDABILITY/COMPLEXITY OF THE
LOGICS FOR $\mathcal{R}^\dagger$ AND $\mathcal{RC}^\dagger$

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 $\text{FO}^2$
- 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.
We begin with the logical system for $\mathcal{RC}^\dagger$, 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 $\mathcal{RC}^\dagger(tr)$. 

**Adding Transitivity to $\mathcal{RC}^\dagger$**
Example of the transitivity rule

Every sweet fruit is bigger than every kumquat

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

1. $\exists I_{sw}, bigger(I, x)\implies bigger(I, y)$
2. $\forall I_{kq}, bigger(I, z)\implies bigger(I, y)$
3. $\forall I, bigger(I, y)\implies bigger(I, z)$

\[
\begin{align*}
\text{Transitivity Rule:} & \hspace{10mm} \frac{\text{big} (x, y) \quad \text{big} (y, z)}{\text{big} (x, z)} \\
\text{Universal Elimination:} & \hspace{10mm} \frac{\text{big} (sw, y) \quad \forall (sw, \forall (kq, \text{big}))}{\text{big} (y, z)}
\end{align*}
\]
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:

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

(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.)
first-order logic

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

2 variable FO logic

† adds full $N$-negation

$RC(tr) + \text{opposites}$
$RC + \text{(transitive)}$
  comparative adjs
$R + \text{relative clauses}$
$S + \text{full } N\text{-negation}$
$R = \text{relational syllogistic}$
$S^\geq \text{ adds } |p| \geq |q|$
$S: \text{ all/some/no } p\text{ are } q$
**Complexity**

*(mostly) best possible results on the validity problem*

- **undecidable**
  - Church 1936
  - Grädel, Otto, Rosen 1999
  - in co-NEXPTIME
  - EXPTIME
  - Lutz & Sattler 2001

- **Co-NEXPTIME**
  - Grädel, Kolaitis, Vardi '97
  - EXPTIME
  - Pratt-Hartmann 2004
  - lower bounds also open

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

- **NLOGSPACE**
### Complexity Sketches

*Again, joint with Ian Pratt-Hartmann*

<table>
<thead>
<tr>
<th>Complexity Class</th>
<th>Time Complexity</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S)</td>
<td>NLOGSPACE</td>
<td>lower bound via reachability problem for directed graphs</td>
</tr>
<tr>
<td>(S^\dagger)</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^\dagger)</td>
<td>EXPTIME</td>
<td>lower bound via (K^U), Hemaspaandra 1996</td>
</tr>
<tr>
<td>(R^{*\dagger})</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 Lutz and Sattler 2001</td>
</tr>
<tr>
<td>(RC)</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: \textbf{Yes}↓ and \textbf{No}↑

- Filtration of a Henkin model
- Mortimer 1975

Irr means that comparative adjectives must have irreflexive interpretations.
\[ \forall(p, \exists(p, r)) + \exists p \]
The last flavor in this talk: monotonicity reasoning

The modest contribution here is to comment on

van Benthem 1986, 1991
Sánchez Valencia 1991
Bernardi 2002

and in a sense to rehabilitate

Dowty 1994

The specific result could be of interest to people working in RTE following

MacCartney and Manning 2009
Nairn, Condoravdi, and Karttunen 2006
Christodoulopoulos 2008
Take categorical grammar a la Ajdukiewicz-Bar Hillel-Lambek and interpret the syntactic types not in sets but in preorders, adding the ability to use opposite of a preorder as well.

Examples of typed constants:

\[
\begin{align*}
\text{every}^+ & : (\lnot \text{pr}, (\text{pr}, \text{t})) & \text{every}^- & : (\text{pr}, (\lnot \text{pr}, \lnot \text{t})) \\
\text{some}^+ & : (\text{pr}, (\text{pr}, \text{t})) & \text{some}^- & : (\lnot \text{pr}, (\lnot \text{pr}, \lnot \text{t})) \\
\text{no}^+ & : (\lnot \text{pr}, (\lnot \text{pr}, \lnot \text{t})) & \text{no}^- & : (\text{pr}, (\text{pr}, \lnot \text{t})) \\
\text{any}^+ & : (\lnot \text{pr}, (\text{pr}, \text{t})) & \text{any}^- & : (\lnot \text{pr}, (\lnot \text{pr}, \lnot \text{t})) \\
\end{align*}
\]

Every binary atom \( r \) gives four type constants:

\[
\begin{align*}
\text{r}_1^+ & : ((\text{pr}, \text{t}), \text{pr}) & \text{r}_2^+ & : ((\lnot \text{pr}, \text{t}), \text{pr}) \\
\text{r}_1^- & : ((\lnot \text{pr}, \lnot \text{t}), \lnot \text{pr}) & \text{r}_2^- & : ((\text{pr}, \lnot \text{t}), \lnot \text{pr}) \\
\end{align*}
\]
Sets do not come with a built in order.

To really discuss upward/downward monotonicity, we need to move from sets to preorders: $\mathbb{P} = (P, \leq)$, where $\leq$ is reflexive and transitive.
Sets do not come with a built in order.

To really discuss upward/downward monotonicity, we need to move from sets to preorders: $\mathcal{P} = (P, \leq)$, where $\leq$ is reflexive and transitive.

**A function $f : \mathcal{P} \to \mathcal{Q}$ is**

- **monotone** if $p \leq q$ in $\mathcal{P}$ implies $f(p) \leq f(q)$ in $\mathcal{Q}$.
- **antitone** if $p \leq q$ in $\mathcal{P}$ implies $f(q) \leq f(p)$ in $\mathcal{Q}$.
Sets do not come with a built in order.

To really discuss upward/downward monotonicity, we need to move from sets to preorders: \( \mathbb{P} = (P, \leq) \), where \( \leq \) is reflexive and transitive.

---

**A function** \( f : \mathbb{P} \to \mathbb{Q} \) **is**

- monotone if \( p \leq q \) in \( \mathbb{P} \) implies \( f(p) \leq f(q) \) in \( \mathbb{Q} \).
- antitone if \( p \leq q \) in \( \mathbb{P} \) implies \( f(q) \leq f(p) \) in \( \mathbb{Q} \).

---

**From now on, all functions are monotone**

- \( \mathbb{Q} \) is \( (Q, \geq) \): it’s \( \mathbb{Q} \) upside-down.

- \( -(-\mathbb{Q}) = \mathbb{Q} \).

An antitone \( f : \mathbb{P} \to \mathbb{Q} \) is exactly a montone \( f : \mathbb{P} \to -\mathbb{Q} \).
What is a monotone function?

Sets do not come with a built in order.

To really discuss upward/downward monotonicity, we need to move from sets to preorders: $\mathbb{P} = (P, \leq)$, where $\leq$ is reflexive and transitive.

Let $[\mathbb{P}, \mathbb{Q}]$ be the monotone function preorder.

$[\mathbb{P}, -\mathbb{Q}] = -[\mathbb{P}, \mathbb{Q}]$

This means that any lexical items typed as $\mathbb{P} \rightarrow -\mathbb{Q}$ could just as well be typed as $-\mathbb{P} \rightarrow \mathbb{Q}$.

However, the orders $[\mathbb{P}, -\mathbb{Q}]$ and $[-\mathbb{P}, \mathbb{Q}]$ are opposites.
We begin with a set $\mathcal{I}_0$ of **basic types**: for simplicity $pr$ and $t$.

We then form a set $\mathcal{I}_1$ of **types** as follows:

$\mathcal{I}_0 \subseteq \mathcal{I}_1$.

If $\sigma, \tau \in \mathcal{I}_1$, then also $(\sigma, \tau) \in \mathcal{I}_1$.

If $\sigma \in \mathcal{I}_1$, then also $-\sigma \in \mathcal{I}_1$.

Let $\equiv$ be the smallest equivalence relation on $\mathcal{I}_1$ such that the following hold:

1. $-(-\sigma) \equiv \sigma$.
2. $-(\sigma, \tau) \equiv (-\sigma, -\tau)$.
3. If $\sigma \equiv \sigma'$, then also $-\sigma \equiv -\sigma'$.
4. If $\sigma \equiv \sigma'$ and $\tau \equiv \tau'$, then $(\sigma, \tau) \equiv (\sigma', \tau')$.

**The set of types**

$\mathcal{I} = \mathcal{I}_1/\equiv$. 
Proposals use preorders for the semantic spaces lor the semantics of our higher-order language \( \mathcal{L} \) we use models \( \mathcal{M} \) of the following form. 

\( \mathcal{M} \) consists of an assignment of preorders \( \sigma \mapsto \mathbb{P}_\sigma \) on \( \mathcal{I}_0 \), together with some data which we shall mention shortly. 

Before this, extend the assignment \( \sigma \mapsto \mathbb{P}_\sigma \) to \( \mathcal{I}_1 \) by

\[
\mathbb{P}(\sigma, \tau) = [\mathbb{P}_\sigma, \mathbb{P}_\tau] \quad \text{monotone function space}
\]

\[
\mathbb{P}_{-\sigma} = -\mathbb{P}_\sigma \quad \text{opposite preorder}
\]

If \( \sigma \equiv \tau \), then \( \mathbb{P}_\sigma = \mathbb{P}_\tau \).

So we have \( \mathbb{P}_\sigma \) for \( \sigma \in \mathcal{I} \). 

We use \( P_\sigma \) to denote the set underlying the preorder \( \mathbb{P}_\sigma \).

The rest of the structure of a model \( \mathcal{M} \) consists of an assignment \( \llbracket c \rrbracket \in P_\sigma \) for each constant \( c : \sigma \), and also a typed map \( f \); this is just a map which to a typed variable \( v : \sigma \) gives some \( f(v) \in P_\sigma \).
Some semantic interpretations in a universe $\mathbf{X}$

$\mathcal{2}$ is true $< \text{false}$. $\mathbf{X}$ is the flat preorder on a set $\mathbf{X}$.

$[[\mathbf{X}, \mathcal{2}], 2]$ is in one-to-one correspondence with the set of subsets of $\mathbf{X}$.

- $\text{every} \in [\neg [[\mathbf{X}, \mathcal{2}], [[\mathbf{X}, \mathcal{2}], 2]]$
- $\text{some} \in [[\mathbf{X}, \mathcal{2}], [[\mathbf{X}, \mathcal{2}], 2]]$
- $\text{no} \in [\neg [[\mathbf{X}, \mathcal{2}], [\neg [[\mathbf{X}, \mathcal{2}], 2]]$

in the standard way:

\[
\text{every}(p)(q) = \begin{cases} 
\text{true} & \text{if } p \leq q \\
\text{false} & \text{otherwise}
\end{cases}
\]

\[
\text{some}(p)(q) = \neg \text{every}(p)(\neg \cdot q)
\]

\[
\text{no}(p)(q) = \neg \text{some}(p)(q)
\]

It follows that

- $\text{every} \in [[\mathbf{X}, \mathcal{2}], [\neg [[\mathbf{X}, \mathcal{2}], -2]]$
- $\text{some} \in [\neg [[\mathbf{X}, \mathcal{2}], [\neg [[\mathbf{X}, \mathcal{2}], -2]]$
- $\text{no} \in [[\mathbf{X}, \mathcal{2}], [[\mathbf{X}, \mathcal{2}], -2]]$
chase\_1^\rightarrow : ((-pr, -t), -pr) \quad \text{every}^- : (pr, (-pr, -t)) \quad \text{cat}^+ : pr

\text{chase}_1^- (\text{every}^- (\text{cat}^+)) : -pr

\text{some}^+ (\text{dog}^+) (\text{chase}_1^+(\text{every}^+ (\text{cat}^-))) : t

\text{some}^+ (\text{dog}^+) (\text{chase}_2^+(\text{no}^+ (\text{cat}^-))) : t

\text{no}^+ (\text{dog}^-) (\text{chase}_2^- (\text{no}^+ (\text{cat}^+))) : t

It can be proved easily that the +, − signs automatically indicate the polarity.
\[
\begin{align*}
\text{any}^- & : (-pr, (-pr, -t)) \\
\text{cat}^- & : -pr \\
\text{see}^-_2 & : ((-pr, -t), -pr) \\
\text{any}^-(\text{cat}^-) & : (-pr, -t) \\
\text{every}^+ & : (-pr, (pr, t)) \\
\text{see}^-_2(\text{any}^-(\text{cat}^-)) & : -pr \\
\text{every}^+(\text{see}^-_2(\text{any}^-(\text{cat}^-))) & : (pr, t) \\
\text{runs}^+ & : \\
\text{every}^+(\text{see}^-_2(\text{any}^-(\text{cat}^-))(\text{runs}^+)) & : t
\end{align*}
\]

Note that any\(^+\) and any\(^-\) should not have the same interpretation!!

\[
\text{any}^- = \text{some} \quad \text{any}^+ = \text{every}
\]
\[
\begin{align*}
&t : \sigma \leq t : \sigma \\
&u : \sigma \leq v : \sigma \quad t : (\sigma, \tau) \\
&t(u) : \tau \leq t(v) : \tau \\
&t : \sigma \leq u : \sigma \quad u : \sigma \leq v : \sigma \\
&t : \sigma \leq v : \sigma \\
&u : (\sigma, \tau) \leq v : (\sigma, \tau) \\
&t : \sigma \quad u(t) : \tau \leq v(t) : \tau
\end{align*}
\]

But it’s open to blend this with the other flavors.

See also Zamansky, Francez and Winter, 2006.
Dessert: questions to spark a discussion

- What are the actual limits of this kind of work?
- How can we go beyond classical model-theoretic validity?
- Is a (complete) logic a semantics?
- How important is the requirement that natural language inference be modeled in a decidable system?
- Philosophy of language: proof-theoretic semantics
- Can we find psycholinguistic/Turker evidence of the change in “flavor”?
- Does the fact that extended syllogistic rules look so weird suggest that the whole enterprise is off track?
- Linguistic semantics:
  Are deep structures necessary, or can we just use surface forms?
- Is there any conceptual value to the complexity results?