Natural Logic: Proof Systems for Reasoning in Natural Language

Larry Moss
Indiana University

EuroProofNet
September 18, 2025

Program

Show that significant parts of what people
when they carry out "inference" in natural language
can be done automatically, using surface forms as much as possible.

To re-work logic in light of computational semantics.

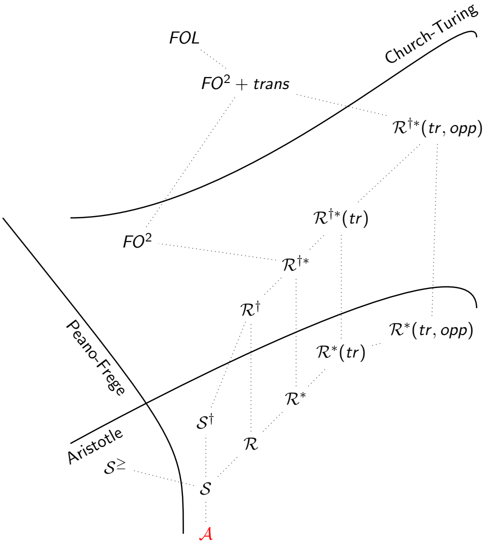
The logical systems that one obtains in this way, should all be decidable, in contrast to standard systems of logic.

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, algebraic logic.
and proof theory,

Today's talk

- Extended syllogistic logics
- Monotonicity calculi
- How does it all work in practice?

The Map



first-order logic

FO^2 + "R is trans"

2 variable FO logic

† adds full N-negation

$\mathcal{R}^*(tr)$ + opposites

$\mathcal{R}^*$ + (transitive) comparative adjs

$\mathcal{R}$ + relative clauses

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

$\mathcal{R}$ = relational syllogistic

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

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

$\mathcal{A}$: all p are q

Syntax: Start with a collection of nouns.
Then the sentences are the expressions

$$\textit{All } p \textit{ are } q$$

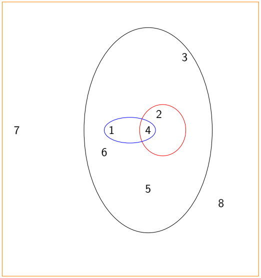
Semantics: A model $\mathcal{M}$ is a set M,
together with an interpretation $[\![p]\!] \subseteq M$ for each noun p.

$$\mathcal{M} \models \textit{All } p \textit{ are } q \qquad \text{iff} \qquad [\![p]\!] \subseteq [\![q]\!]$$

The semantics is trivial, as it should be

Even so, there is an issue with existential import

Let $M = \{1, 2, 3, 4, 5, 6, 7, 8\}$.
Let $[\![a]\!] = \{1, 2, 3, 4, 5, 6\}$.
Let $[\![x]\!] = \{1, 4\}$.
Let $[\![y]\!] = \{2, 4\}$.



$\mathcal{M} \models$ All x are a
$\mathcal{M} \not\models$ All a are x
$\mathcal{M} \not\models$ All y are x
$\mathcal{M} \models$ All y are a
$\mathcal{M} \models$ All a are a

If Γ 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 $\mathcal{M} \models \varphi$.

All of this is semantic.

The rules are

$$\frac{}{\textit{All } p \textit{ are } p} \qquad\qquad \frac{\textit{All } p \textit{ are } n \quad \textit{All } n \textit{ are } q}{\textit{All } p \textit{ are } q}$$

A proof tree over Γ is a finite tree $\mathcal{T}$
whose nodes are labeled with sentences,
and each node is either an element of Γ,
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 Γ
whose root is labeled φ.

Let Γ be the set

$$\{\textit{All } a \textit{ are } b, \textit{All } q \textit{ are } a, \textit{All } b \textit{ are } d, \textit{All } c \textit{ are } d, \textit{All } a \textit{ are } q\}$$

Let φ be *All* q *are* d.

Here is a proof tree showing that $\Gamma \vdash \varphi$:

$$\dfrac{\textit{All } q \textit{ are } a \qquad \dfrac{\textit{All } a \textit{ are } b \quad \textit{All } b \textit{ are } d}{\textit{All } a \textit{ are } d}\,\text{BARBARA}}{\textit{All } q \textit{ are } d}\,\text{BARBARA}$$

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 $[\![u]\!] = \downarrow u$.
Then $\mathcal{M} \models \Gamma$.
Hence for the p and q in our statement, $[\![p]\!] \subseteq [\![q]\!]$.

But by reflexivity, $p \in [\![p]\!]$.
And so $p \in [\![q]\!]$; this means that $p \leq q$.

But this is exactly what we want:
$\Gamma \vdash$ *All p are q*.

Syllogistic 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 $[\![p]\!] \subseteq M$.

$$\begin{array}{lll} \mathcal{M} \models \textit{All p are q} & \text{iff} & [\![p]\!] \subseteq [\![q]\!] \\ \mathcal{M} \models \textit{Some p are q} & \text{iff} & [\![p]\!] \cap [\![q]\!] \neq \emptyset \end{array}$$

Proof system:

$$\frac{}{\textit{All p are p}} \qquad \frac{\textit{All p are n} \quad \textit{All n are q}}{\textit{All p are q}}$$

$$\frac{\textit{Some p are q}}{\textit{Some q are p}} \qquad \frac{\textit{Some p are q}}{\textit{Some p are p}} \qquad \frac{\textit{All q are n} \quad \textit{Some p are q}}{\textit{Some p are n}}$$

Example

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

$$\dfrac{\textit{All n are q} \qquad \dfrac{\dfrac{\textit{All n are p} \quad \textit{Some n are n}}{\textit{Some n are p}}}{\textit{Some p are n}}}{\textit{Some p are q}}$$

The languages $\mathcal{S}$ and $\mathcal{S}^\dagger$ 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

$$
\left.\begin{array}{l}
\mathcal{S} \quad \left\{ \begin{array}{l}
\textit{All } p \textit{ are } q \\
\textit{Some } p \textit{ are } q \\
\textit{All } p \textit{ are } \overline{q} \equiv \textit{No } p \textit{ are } q \\
\textit{Some } p \textit{ are } \overline{q} \equiv \textit{Some } p \textit{ aren't } q
\end{array} \right. \\
\\
\qquad\quad \textit{Some non-}p\textit{ are non-}q
\end{array}\right\} \quad \mathcal{S}^\dagger
$$

$$\frac{}{\textit{All } p \textit{ are } p} \qquad \frac{\textit{Some } p \textit{ are } q}{\textit{Some } p \textit{ are } p} \qquad \frac{\textit{Some } p \textit{ are } q}{\textit{Some } q \textit{ are } p}$$

$$\frac{\textit{All } p \textit{ are } n \quad \textit{All } n \textit{ are } q}{\textit{All } p \textit{ are } q} \qquad \frac{\textit{All } n \textit{ are } p \quad \textit{Some } n \textit{ are } q}{\textit{Some } p \textit{ are } q}$$

$$\frac{\textit{All } q \textit{ are } \overline{q}}{\textit{All } q \textit{ are } p}\ \textit{Zero} \qquad \frac{\textit{All } \overline{q} \textit{ are } q}{\textit{All } p \textit{ are } q}\ \textit{One}$$

$$\frac{\textit{All } p \textit{ are } \overline{q}}{\textit{All } q \textit{ are } \overline{p}}\ \textit{Antitone} \qquad \frac{\textit{Some } p \textit{ are } \overline{p}}{\varphi}\ \textit{Ex falso quodlibet}$$

The system uses

$$\frac{\textit{Some } p \textit{ are } \overline{p}}{\varphi}\ \textit{Ex falso quodlibet}$$

and this is prima facie weaker than reductio ad absurdum.

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

What do you think about this one?

$$\frac{\textit{All skunks are mammals}}{\textit{All who fear all who respect all skunks fear all who respect all mammals}}$$

It follows, using an interesting antitonicity principle:

$$\frac{\textit{All skunks are mammals}}{\textit{All who respect all mammals respect all skunks}}$$

We start with two sets:

- a set of nouns.
- a set of verbs.

We make terms as follows:

- If x is a noun, then x is a term.
- If r is verb and x is a term, then r all x is a term.

We make sentences as follows:

- If x and y are terms, then

  All x y

  is a sentence.

Let's say

- $\mathbf{P} = \{\text{dogs, cats, birds, ants, } \ldots\}$
- $\mathbf{R} = \{\text{see, like, hate, fear, respect, } \ldots\}$

Here are some terms of $\mathcal{A}(RC)$:

- dogs
- see all dogs
- respect all (see all dogs)
- love all (respect all (see all dogs))

Note that there are infinitely many terms, and terms may occur in terms.

Let's say

- $\mathbf{P} = \{\text{dogs, cats, birds, ants, } \ldots\}$
- $\mathbf{R} = \{\text{see, like, hate, fear, respect, } \ldots\}$

Here are some terms of $\mathcal{A}(RC)$:

- dogs
- see all dogs
- respect all (see all dogs)
  read as respect all who see all dogs
- love all (respect all (see all dogs))
  read as love all who respect all who see all dogs

Note that there are infinitely many terms, and terms may occur in terms.

We read these in English using relative clauses.

We make proof trees using the following rules

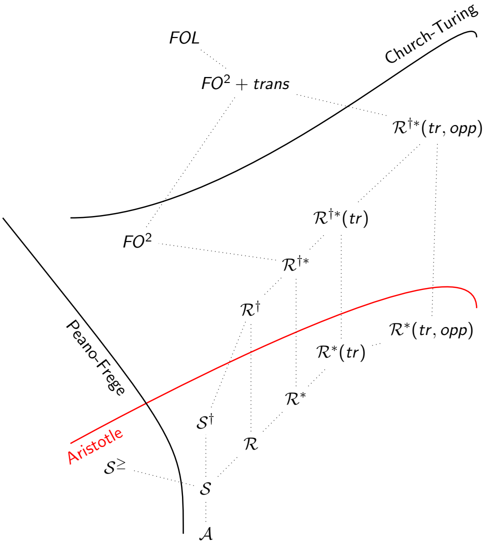
$$\frac{}{\mathsf{All}\ x\ x}\ \textsc{Axiom} \qquad \frac{\mathsf{All}\ x\ y \quad \mathsf{All}\ y\ z}{\mathsf{All}\ x\ z}\ \textsc{Barbara}$$

$$\frac{\mathsf{All}\ y\ x}{\mathsf{All}\ (r\ \mathsf{all}\ x)\ (r\ \mathsf{all}\ y)}\ \textsc{Anti}$$

Note that we are using this with x, y, and z as terms, not only as unary variables.

Example

$$\dfrac{\dfrac{\textsf{All skunks mammals}}{\textsf{All (love all mammals) (love all skunks)}}\ \textsc{anti}}{\textsf{All (hate all (love all skunks)) (hate all (love all mammals))}}\ \textsc{anti}$$



first-order logic

FO^2 + "R is trans"

2 variable FO logic

† adds full N-negation

$\mathcal{R}^*(tr)$ + opposites

$\mathcal{R}^*$ + (transitive) comparative adjs

$\mathcal{R}$ + relative clauses

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

$\mathcal{R}$ = relational syllogistic

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

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

$\mathcal{A}$: all p are q

$\mathcal{R}^{\dagger}$ 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

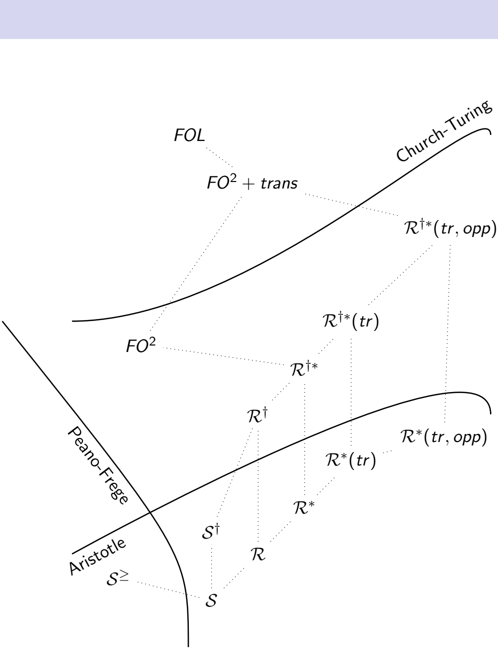
$$
\dfrac{[\exists(key, own)(x)]^2 \qquad \dfrac{[own(x,y)]^1 \qquad \dfrac{[key(y)]^1 \quad \forall(key, old\text{--}item)}{old\text{--}item(y)}\ \forall E}{\exists(old\text{--}item, own)(x)}\ \exists I}{\dfrac{\exists(old\text{--}item, own)(x)}{\forall(\exists(key, own), \exists(old\text{--}item, own))}\ \forall I^2}\ \exists E^1
$$

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	$\quad \exists(key, own)(x)$	hyp
3	$\quad key(y)$	$\exists E$, 2
4	$\quad own(x, y)$	$\exists E$, 2
5	$\quad old\text{–}item(y)$	$\forall E$, 1, 3
6	$\quad \exists(old\text{–}item, own)(x)$	$\exists I$, 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



first-order logic

FO^2 + "R is trans"

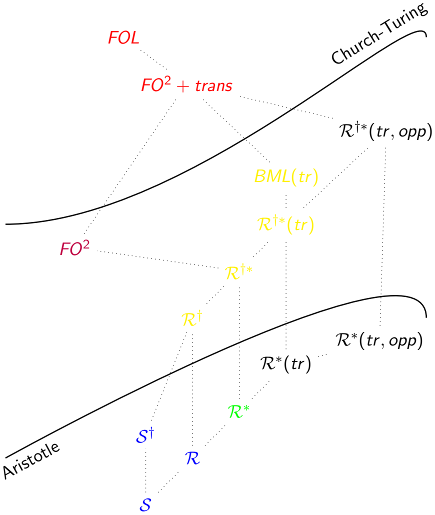
2 variable FO logic

† adds full N-negation

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

Other work that I'm not talking about today

- A lot more on relational syllogistic logics, including connections to some modal logics (with Alex Kruckman).

- Logic for reasoning about the sizes of sets,

$$\frac{\text{most } Y \textit{ are } X}{\text{most } X \textit{ are } Y}\ ? \qquad \frac{}{\text{most } X \textit{ are } X}\ ? \qquad \frac{\text{all} X \textit{ are } Y}{\text{most } X \textit{ are } Y}\ ?$$

$$\frac{\text{most } X \textit{ are } Y \quad \text{most } Y \textit{ are } Z}{\text{most } X \textit{ are } Z}\ ? \qquad \frac{\text{most } X \textit{ are } Y \quad \text{all } Y \textit{ are } Z}{\text{most } X \textit{ are } Z}\ ?$$

  (with Tri Lai, Joerg Endrullis, Paul Pedersen, and others)

- Logics for definite descriptions and names, including connections to free logic.

- Logics in the natural logic area that do have $\sqcap$ and $\sqcup$, including connections to description logics, and to lattice theory. (with Antonio Badia, just completed)

This is an entry for a United States National Science Foundation contest on mathematics outreach for the general public.

general rules of CCG (a few missing)

$$\frac{Y \quad X\backslash Y}{X}\,< \qquad \frac{X/Y \quad Y}{X}\,> \qquad \frac{Y}{X/(X\backslash Y)}\,\mathrm{T}$$

$$\frac{X}{Y\backslash(Y/X)}\,\mathrm{T} \qquad \frac{X/Y \quad Y/Z}{X/Z}\,\mathrm{B} \qquad \frac{Y\backslash Z \quad X\backslash Y}{X\backslash Z}\,\mathrm{B}$$

A tiny lexicon

word	category
every	NP/N
cat	N
that	$(\mathrm{N}\backslash\mathrm{N})/(\mathrm{S}/\mathrm{NP})$

word	category
Fido	NP
chased	$(\mathrm{S}\backslash\mathrm{NP})/\mathrm{NP}$
ran	$\mathrm{S}\backslash\mathrm{NP}$

$$\dfrac{\dfrac{\text{every} : \mathrm{NP}/\mathrm{N} \quad \dfrac{\text{cat} : \mathrm{N} \quad \dfrac{\text{that} : (\mathrm{N}\backslash\mathrm{N})/(\mathrm{S}/\mathrm{NP}) \quad \dfrac{\dfrac{\text{F} : \mathrm{NP}}{\text{F} : \mathrm{S}/(\mathrm{S}\backslash\mathrm{NP})}\,\mathrm{T} \quad \text{ch} : (\mathrm{S}\backslash\mathrm{NP})/\mathrm{NP}}{\text{Fido chased} : \mathrm{S}/\mathrm{NP}}\,\mathrm{B}}{\text{that Fido chased} : \mathrm{N}\backslash\mathrm{N}}\,>}{\text{cat that Fido chased} : \mathrm{N}}\,<}{\text{every cat that Fido chased} : \mathrm{NP}}\,> \quad \text{ran} : \mathrm{S}\backslash\mathrm{NP}}{\text{every cat that Fido chased ran} : \mathrm{S}}\,<$$

The syntax tree is given to us by a CCG parser:

$$
\dfrac{\dfrac{\textsf{every} : \mathrm{NP/N} \qquad \dfrac{\textsf{cat} : \mathrm{N} \qquad \dfrac{\textsf{that} : (\mathrm{N}\backslash\mathrm{N})/(\mathrm{S/NP}) \qquad \dfrac{\dfrac{\textsf{F} : \mathrm{NP}}{\textsf{F} : \mathrm{S}/(\mathrm{S}\backslash\mathrm{NP})}\,{}^{\mathrm{T}} \qquad \textsf{ch} : (\mathrm{S}\backslash\mathrm{NP})/\mathrm{NP}}{\textsf{Fido chased} : \mathrm{S/NP}}\,{}_{\mathrm{B}}}{\textsf{that Fido chased} : \mathrm{N}\backslash\mathrm{N}}\,{}_{>}}{\textsf{cat that Fido chased} : \mathrm{N}}\,{}_{<}}{\textsf{every cat that Fido chased} : \mathrm{NP}}\,{}_{>} \qquad \textsf{ran} : \mathrm{S}\backslash\mathrm{NP}}{\textsf{every cat that Fido chased ran} : \mathrm{S}}\,{}_{<}
$$

This tree has a semantics which is suggested below:

$$
\dfrac{\dfrac{\textsf{every} : \mathrm{N} \to \mathrm{NP} \qquad \dfrac{\textsf{cat} : \mathrm{N} \qquad \dfrac{\textsf{that} : (\mathrm{NP} \to \mathrm{S}) \to (\mathrm{N} \to \mathrm{N}) \qquad \dfrac{\dfrac{\textsf{F} : \mathrm{NP}}{\textsf{F} : (\mathrm{NP} \to \mathrm{S}) \to \mathrm{S}}\,{}^{\mathrm{T}} \qquad \textsf{ch} : \mathrm{NP} \to (\mathrm{NP} \to \mathrm{S})}{\textsf{Fido ch} : \mathrm{NP} \to \mathrm{S}}\,{}_{\mathrm{B}}}{\textsf{that Fido chased} : \mathrm{N} \to \mathrm{N}}\,{}_{>}}{\textsf{cat that Fido chased} : \mathrm{N}}\,{}_{<}}{\textsf{every cat that Fido chased} : \mathrm{NP}}\,{}_{>} \qquad \textsf{ran} : \mathrm{NP}}{\textsf{every cat that Fido chased ran} : \mathrm{S}}
$$

“The structure of every sentence is a lesson in logic.”

John Stuart Mill (1867)

Saving on notation by writing W for $\mathrm{NP}^{+} \xrightarrow{+} t$:

$$
\dfrac{
  \dfrac{\text{every}^{\uparrow} : \mathrm{N} \xrightarrow{-} \mathrm{NP}^{+}
  \qquad
  \dfrac{\text{cat}^{\downarrow} : \mathrm{N}
    \qquad
    \dfrac{\text{that}^{\downarrow} : W \xrightarrow{+} (\mathrm{N} \xrightarrow{+} \mathrm{N})
      \qquad
      \dfrac{
        \dfrac{\dfrac{\text{F}^{\downarrow} : e}{\text{F}^{\downarrow} : \mathrm{NP}^{+}}\,{\scriptstyle \mathrm{J}}}{\text{F}^{\downarrow} : W \xrightarrow{+} t}\,{\scriptstyle \mathrm{T}}
        \quad
        \dfrac{\text{ch}^{\downarrow} : e \xrightarrow{+} W}{\text{ch}^{\downarrow} : \mathrm{NP}^{+} \xrightarrow{+} W}\,{\scriptstyle \mathrm{J}}
      }{\text{Fido ch}^{\downarrow} : W}\,{\scriptstyle \mathrm{B}}
    }{\text{that Fido chased}^{\downarrow} : \mathrm{N} \xrightarrow{+} \mathrm{N}}\,{\scriptstyle >}
  }{\text{cat that Fido chased}^{\downarrow} : \mathrm{N}}\,{\scriptstyle <}
  }{\text{every cat that Fido chased}^{\uparrow} : \mathrm{NP}^{+}}\,{\scriptstyle >}
  \qquad
  \dfrac{\text{ran}^{\uparrow} : e \xrightarrow{+} t}{\text{ran}^{\uparrow} : W}\,{\scriptstyle \mathrm{J}}
}{\text{every cat that Fido chased ran}^{\uparrow} : t}\,{\scriptstyle <}
$$

The arrows *could* be determined just by parsing from our rules,
but since we want to use the parse given to us by a parser,
we aim for an algorithm that polarizes an upolarized CCG tree.

I am omitting discussion of our actual algorithm.
You could take it to be constraint satisfaction,
but it's possible to be much more direct.

$$
\dfrac{\mathsf{F}^{\uparrow} : e \qquad \dfrac{\dfrac{\dfrac{\mathsf{ch}^{=} : e \xrightarrow{+} pr}{\mathsf{ch}^{=} : np \xrightarrow{+} pr}\,\mathrm{K}}{\mathsf{ch}^{\uparrow} : np \xrightarrow{+} pr}\,\mathrm{W} \qquad \dfrac{\dfrac{\begin{array}{c}\vdots\\ \text{some cat}^{\uparrow} : np^{+}\end{array}}{\text{some cat}^{\uparrow} : np}\,\mathrm{M} \qquad \dfrac{\text{and} : np \xrightarrow{+} (np \xrightarrow{+} np) \qquad \dfrac{\begin{array}{c}\vdots\\ \text{no arm}^{\downarrow} : np^{-}\end{array}}{\text{no arm}^{\downarrow} : np}\,\mathrm{M}}{\text{and no arm} : np \xrightarrow{+} np}}{\text{some cat and no armadillo}^{\uparrow} : np}\,<}{\text{chased some cat and no armadillo}^{\uparrow} : pr}\,>}{\text{Fido chased some cat and no armadillo}^{\uparrow} : t}\,<
$$

We use (M) twice in order conjoin some cat and no armadillo.

Semantic rules again, with hints about what they mean

Rules

$$\frac{u^{d} : x \xrightarrow{m} y \quad v^{md} : x}{(uv)^{d} : y} > \qquad \frac{u^{md} : x}{(\mathrm{T}u)^{d} : (x \xrightarrow{m} y) \xrightarrow{+} y}\ \mathrm{T} \qquad \frac{u^{md} : x \xrightarrow{n} y \quad v^{md} : y \xrightarrow{n} z}{(\mathrm{B}uv)^{d} : (x \xrightarrow{mn} z)}\ \mathrm{B}$$

$$\frac{u^{md} : e \to b}{(r_m u)^{d} : np^{m} \xrightarrow{+} b}\ \mathrm{K} \qquad \frac{u^{d} : x \xrightarrow{m} y}{u^{d} : x \xrightarrow{\cdot} y}\ \mathrm{M} \qquad \frac{u^{=} : x \xrightarrow{m} y}{u^{d} : x \xrightarrow{m} y}\ \mathrm{W}$$

The $>$ in the application rule is function application.

The T in the type-raising rule is the Montague lift.

The B in the type-raising rule is function composition, backwards.

The r_m in the K rule is from our refinement of the Justification Theorem.

In the M rule, we have a trivial inclusion.

The W rule is trivial.

What our polarized trees mean, on a semantic level

Example (a polarized syntax tree)

$$\dfrac{\dfrac{\text{some}^{\uparrow} : pr \xrightarrow{+} np^{+} \quad \text{dog}^{\uparrow} : pr}{\text{some dog}^{\uparrow} : pr \xrightarrow{+} t}\,> \qquad \dfrac{\dfrac{\dfrac{\text{chased}^{\downarrow} : e \xrightarrow{+} pr}{\text{chased}^{\uparrow} : np^{-} \xrightarrow{+} pr}\,\mathrm{K} \qquad \dfrac{\text{no}^{\uparrow} : pr \xrightarrow{-} np^{-} \quad \text{cat}^{\downarrow} : pr}{\text{no cat}^{\uparrow} : np^{-}}\,>}{\text{chased no cat}^{\uparrow} : pr}\,>}{\text{some dog chased no cat}^{\uparrow} : t}\,>$$

Example (Abstract the words and move from syntax to semantics)

$$\dfrac{\dfrac{v^{\uparrow} : pr \xrightarrow{+} np^{+} \quad w^{\uparrow} : pr}{vw^{\uparrow} : pr \xrightarrow{+} t}\,> \qquad \dfrac{\dfrac{\dfrac{x^{\downarrow} : e \xrightarrow{+} pr}{r\_x^{\uparrow} : np^{-} \xrightarrow{+} pr}\,\mathrm{K} \qquad \dfrac{y^{\uparrow} : pr \xrightarrow{-} np^{-} \quad z^{\downarrow} : pr}{yz^{\uparrow} : np^{-}}\,>}{(r\_x)(yz)^{\uparrow} : pr}\,>}{\tau = (vw)((r\_x)(yz))^{\uparrow} : t}\,>$$

Note that the semantic term τ on the bottom is a combinator term.

The polarity arrows on the leaves mean that in every model,

$$[\![\tau]\!] : \mathbb{P}_{pr \xrightarrow{+} np^{+}} \times \mathbb{P}_{pr} \times (\mathbb{P}_{e \xrightarrow{+} pr})^{op} \times \mathbb{P}_{pr \xrightarrow{-} np^{-}} \times (\mathbb{P}_{pr})^{op} \quad \xrightarrow{+} \quad \mathbb{P}_{t}$$

Monotonicity + Natural Logic at work: the FRaCaS dataset

entail, _contradict_ or _neural_?

P: A schoolgirl with a black bag is on a crowded train

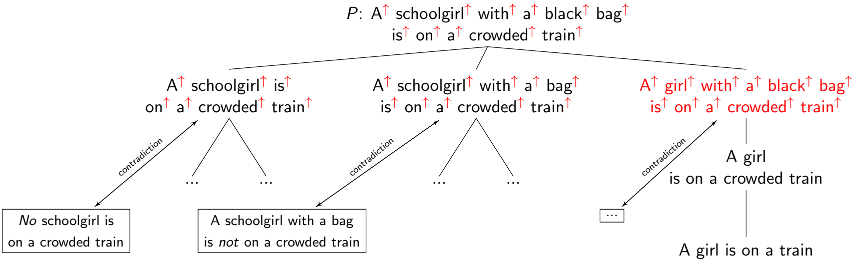
H: No schoolgirl is on a crowded train

entail, _contradict_ or _neural_?

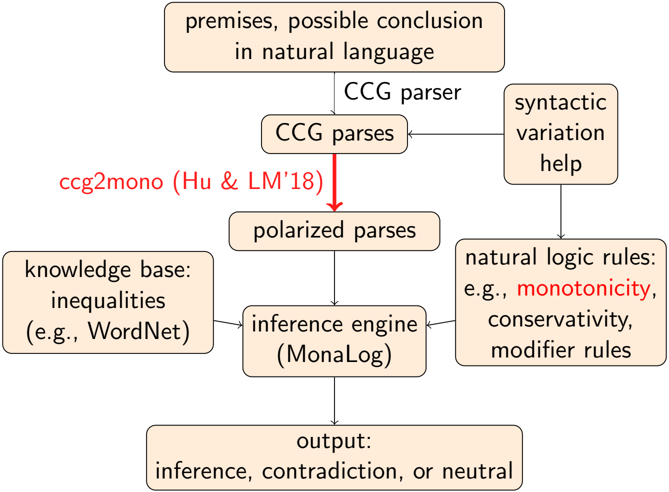
P: A schoolgirl with a black bag is on a crowded train

H: A girl is on a train

How the algorithm works, roughly



How arrow tagging fits in to our natural logic inference (NLI) system MonaLog



Results on the SICK dataset

system	P	R	acc.
majority baseline	–	–	56.36
Natural-logic-based: MonaLog‡			
MonaLog + pass2act	89.42	72.18	80.25†
MonaLog + $\exists$ transformation	89.43	71.53	79.11†
MonaLog + all rewrite help	83.75	70.66	77.19
MonaLog + all rewrite help	89.91	74.23	**81.66**†
Hybrid: MonaLog + BERT	83.09	85.46	85.38
Hybrid: MonaLog + BERT	85.65	87.33	**85.95**†
ML/DL-based systems			
BERT (base, uncased)	86.81	85.37	**86.74**
BERT (base, uncased)	84.62	84.27	85.00†
Yin & Schütze'17	–	–	**87.1**
Beltagy et al '16	–	–	85.1
Other logic-based systems			
Bjerva et al '14	93.6	60.6	81.6
Abzianidze'17	97.95	58.11	81.35
Martínez-Gomez et al '17	97.04	63.64	83.13
Yanaka et al '18	84.2	77.3	84.3

† = running on a corrected version of the SICK dataset.

‡ = P / R for MonaLog averaged across three labels.

Using a different parsing system

Two talented undergraduate students

Zeming Chen and Qiyue Gao

built a polarity-tagging system using universal dependency parses rather than CCG parses.

+ More people can use it.

+ It performs better than ccg2Mono,
  partly because the parses are better,
  and partly because ccg2Mono misses some arrows,
  such as on attitude verbs: John refused to dance$^{\downarrow}$.

− This form of grammar doesn't have
  a connection to formal semantics,
  so one can't really prove soundness results
  the way we can with ccg2Mono.

Using a different parsing system

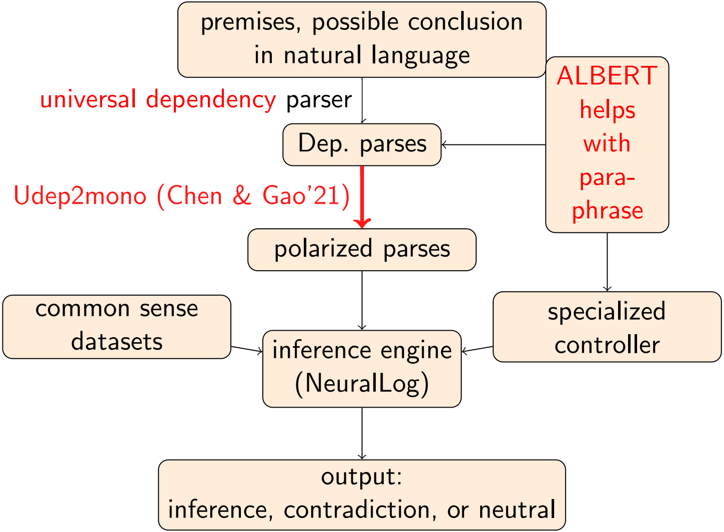
Two talented undergraduate students

Zeming Chen and Qiyue Gao

built a polarity-tagging system using universal dependency parses rather than CCG parses.

It's called Udep2mono and you can get it at `https://github.com/eric11eca/Udep2Mono`.

A hybrid system: NeuralLog



What my collaborators and I have been doing

Thanks to

Hai Hu, Thomas Icard, Kyle Richardson, Zeming Chen, Qiyue Gao
Valeria de Paiva, Katerina Kalouli

- We extended monotonicity from vanilla CG to CCG.
- We have an order-enriched version of the typed lambda calculus.
- We have a running system that can polarize input sentences.
- We built MonaLog and NeuralLog to solve a large NLI dataset.
- The output of these systems is "correct by construction" unlike what you find with ChatGPT.
- We can generate high-quality sentence pairs, helpful to a ML model.
- We have hybridized logic and machine learning and currently have the
  currently-best system for inference on the SICK dataset.
- We have re-annotated SICK by hand and have a deep study of NLI annotation.
- We are looking at mathematics, especially connecting computational linguistics to mathematical AI