Reasoning about contexts in Henkin models

Chung-chieh Shan, Rutgers
with Chris Barker, NYU

Lambda Calculus and Formal Grammar
March 24, 2008
Contexts are syntactic functions

Alice saw [everyone].
everyone \circ (\lambda x. \text{Alice saw } x)

Alice saw [him].
him \circ (\lambda x. \text{Alice saw } x)
Contexts are syntactic functions

The [same] critic saw [every movie]. (Barker)

\[
\text{every movie} \circ (\text{same} \circ (\lambda x. \lambda y. \text{The } x \text{ critic saw } y))
\]

Americans [on average] have [2.3] children. (Kennedy & Stanley)

\[
2.3 \circ (\text{on average} \circ (\lambda x. \lambda y. \text{Americans } x \text{ have } y \text{ children}))
\]
Contexts are syntactic functions

But not ‘exotic’ functions such as

\[ \lambda x. \begin{cases} 
  & \text{if } x = \text{Bob} \\
  & \text{then Alice saw } x \\
  & \text{else } x \text{ slept} 
\end{cases} \]

(cf. Henkin models for higher-order logic)
Contexts are syntactic functions

But not ‘exotic’ functions such as

\[ \lambda x. \begin{align*}
& \text{if } x = \text{Bob} \\
& \text{then Alice saw } x \\
& \text{else } x \text{ slept}
\end{align*} \]

(cf. Henkin models for higher-order logic)

This talk

From proof theory \( \lambda \)-terms and their \( \beta \)-equivalence

To model theory relational models of the \( \lambda \)-calculus

van Benthem 1999: Relating modal logic (categorial grammar) and type theory (abstract categorial grammar)
NL proofs

Connectives: \ /  Punctuation: ●

\[
\begin{align*}
\text{DP} \vdash \text{DP} & \quad \quad \text{S} \vdash \text{S} \\
\text{DP} \bullet \text{DP} \setminus \text{S} & \vdash \text{S} & \quad \quad \\
\text{DP} \bullet (\text{DP} \setminus \text{S}) & \vdash \text{S} \quad & \quad \text{DP} \vdash \text{DP} \\
\text{Alice} & \quad \text{saw} & \quad \text{Bob}
\end{align*}
\]
NL proofs

Connectives: \ / Punctuation: ⋅

\[
\begin{align*}
\text{DP} & \vdash \text{DP} & \text{S} & \vdash \text{S} \\
\text{DP} \cdot \text{DP}/\text{S} & \vdash \text{S} & \text{DP} & \vdash \text{DP} \\
\text{DP} \cdot ((\text{DP}/\text{S})/\text{DP} \cdot \text{DP}) & \vdash \text{S}
\end{align*}
\]

Alice \hspace{1cm} \text{saw} \hspace{1cm} Bob

\[
\begin{align*}
\Gamma & \vdash X & \Sigma[Y] & \vdash Z \\
\Sigma[\Gamma \cdot X/Y] & \vdash Z \\
\Sigma[Y] & \vdash Z & \Gamma & \vdash X \\
\Sigma[Y/X \cdot \Gamma] & \vdash Z
\end{align*}
\]

\[
\begin{align*}
X \cdot \Gamma & \vdash Y \\
\Gamma & \vdash X/Y \\
\Gamma & \vdash Y/X
\end{align*}
\]
NL models

A frame consists of
- a set of points \( \mathcal{P} \) and
- a ternary accessibility relation \( R_\bullet \subseteq \mathcal{P}^3 \).

A model consists of a frame and a valuation \( \models \) that relates points \( p, q, r \) to the structures and formulas they satisfy.
NL models

A frame consists of

- a set of points $\mathcal{P}$ and
- a ternary accessibility relation $R_\bullet \subseteq \mathcal{P}^3$.

A model consists of a frame and a valuation $\models$ that relates points $p, q, r$ to the structures and formulas they satisfy.

\[
r \models DP \bullet ((DP \setminus S)/DP \bullet DP)
\]
\[
\iff \exists p. \exists q. R_\bullet (p, q, r) \land (p \models DP) \land (q \models (DP \setminus S)/DP \bullet DP)
\]
\[
r \models (DP \setminus S)/DP \bullet DP
\]
\[
\iff \exists p. \exists q. R_\bullet (p, q, r) \land (p \models (DP \setminus S)/DP) \land (q \models DP)
\]
\[
p \models (DP \setminus S)/DP
\]
\[
\iff \forall q. \forall r. R_\bullet (p, q, r) \rightarrow (q \models DP) \rightarrow (r \models DP \setminus S)
\]
\[
q \models DP \setminus S
\]
\[
\iff \forall p. \forall r. R_\bullet (p, q, r) \rightarrow (p \models DP) \rightarrow (r \models S)
\]
NL soundness and completeness

\[ \Gamma \vdash X \iff \text{In any model, at any point } p, \text{ if } p \not\models \Gamma \text{ then } p \not\models X. \]

The canonical completeness proof constructs a canonical model, in which points are structures.
\textbf{NL}_\lambda \text{ proofs}

Connectives: \(\setminus / \setminus / \setminus/\)

Punctuation: \(\bullet \circ x \lambda x\) (linear)

\[
\begin{align*}
\text{DP} \bullet ((\text{DP}\setminus \text{S})/\text{DP} \bullet \text{DP}) & \vdash \text{S} \\
\text{DP} \circ \lambda x. (\text{DP} \bullet ((\text{DP}\setminus \text{S})/\text{DP} \bullet x)) & \vdash \text{S} \\
\lambda x. (\text{DP} \bullet ((\text{DP}\setminus \text{S})/\text{DP} \bullet x)) & \vdash \text{DP}\setminus \text{S} \\
\text{S}\setminus (\text{DP}\setminus \text{S}) \circ \lambda x. (\text{DP} \bullet ((\text{DP}\setminus \text{S})/\text{DP} \bullet x)) & \vdash \text{S} \\
\text{DP} \bullet ((\text{DP}\setminus \text{S})/\text{DP} \bullet \text{S}\setminus (\text{DP}\setminus \text{S})) & \vdash \text{S}
\end{align*}
\]

\begin{itemize}
\item Alice
\item saw
\item everyone
\end{itemize}
For binding, [same], and [on average], the context $\Gamma[\ ]$ may contain $\lambda$. 
**NLλ models**

A *frame* consists of

- a set of points $\mathcal{P}$,
- a ternary accessibility relation $\mathcal{R} \circ \subseteq \mathcal{P}^3$, and
- a ternary accessibility relation $\mathcal{R} \bullet \subseteq \mathcal{P}^3$ (not a function), such that there are ‘enough functions’ (more on that later).
**NL\_\lambda** models: satisfaction

How to define satisfaction?
Points in structures are convenient (hybridization).

\[ p \models q \]
\[ \iff p = q \]

\[ q \models \lambda x. (\text{DP} \bullet ((\text{DP}\setminus S)/\text{DP} \bullet x)) \]
\[ \iff \forall p. \forall r. R_\circ (p, q, r) \iff (r \models \text{DP} \bullet ((\text{DP}\setminus S)/\text{DP} \bullet p)) \]
**NL_λ models: satisfaction**

How to define satisfaction?
Points in structures are convenient (hybridization).

\[ p \vdash q \]

\[ \iff \quad p = q \]

\[ q \vdash \lambda x. (DP \cdot ((DP \setminus S)/DP \cdot x)) \]

\[ \iff \quad \forall p. \forall r. R_\circ (p, q, r) \leftrightarrow (r \vdash DP \cdot ((DP \setminus S)/DP \cdot p)) \]

But what if no point satisfies DP or no point satisfies \((DP \setminus S)/DP\)?

\[
\begin{align*}
\text{DP} \circ \lambda y. \lambda x. (y \cdot ((DP \setminus S)/DP \cdot x)) & \vdash X \\
\frac{\lambda x. (DP \cdot ((DP \setminus S)/DP \cdot x)) \vdash X}{(DP \setminus S)/DP \circ \lambda z. \lambda x. (DP \cdot (z \cdot x)) \vdash X}
\end{align*}
\]
NL_\lambda_ models: satisfaction

The definition of satisfaction for \( \lambda x. \Gamma[x] \) quantifies over the maximal substructures of \( \Gamma[x] \) that do not contain \( x \).

\[
p \models q \quad \iff \quad p = q
\]

\[
q \models \lambda x. (\text{DP} \bullet ((\text{DP} \setminus \text{S})/\text{DP} \bullet x))
\]

\[
\iff \exists s. \exists t. (s \models \text{DP}) \land (t \models (\text{DP} \setminus \text{S})/\text{DP}) \\
\land \forall p. \forall r. R_\circ (p, q, r) \leftrightarrow (r \models s \bullet (t \bullet p))
\]

Each \( \lambda \)-abstraction shape is like a jumbo product connective.

\[
\text{DP} \circ \lambda y. \lambda x. (y \bullet ((\text{DP} \setminus \text{S})/\text{DP} \bullet x)) \vdash X
\]

\[
\beta
\]

\[
\lambda x. (\text{DP} \bullet ((\text{DP} \setminus \text{S})/\text{DP} \bullet x)) \vdash X
\]

\[
\beta
\]

\[
(\text{DP} \setminus \text{S})/\text{DP} \circ \lambda z. \lambda x. (\text{DP} \bullet (z \bullet x)) \vdash X
\]
We require of the frame that there be ‘enough functions’:

- There must be some point $q$ such that $q \models \lambda x. x$.
- For any points $s, t$, there must be some point $q$ such that $q \models \lambda x. s \bullet (t \bullet x)$.
- And so on, for each $\lambda$-abstraction shape.

Or in computational terms: we can always build a closure.
\[ \Gamma \models X \quad \iff \quad \text{In any model, at any point } p, \text{ if } p \models \Gamma \text{ then } p \models X. \]

The canonical completeness proof constructs a canonical model, in which points are \( \beta \)-equivalence classes of structures.
$\text{NL}_\lambda$ conservatativity over NL

An NL sequent that is provable in $\text{NL}_\lambda$ is already provable in NL.

Extend any NL model to an $\text{NL}_\lambda$ model whose points are $\beta$-equivalence classes of structures whose maximal substructures that do not contain variables are the old points.

What keeps the old points separate in the new model is the confluence of the $\lambda$-calculus!

Domain theory for syntax?
Summary

Relational models of the $\lambda$-calculus

- are natural to define;
- capture the meanings of contexts as syntactic functions;
- should perhaps be equipped with kind structure.