Surface Reasoning

Lecture 3: A Natural Logic Proof System

Thomas Icard

June 18-22, 2012

- Overview
- Back to Types
- Review of Lambek Calculus
- The Order Calculus OC
- The Proof System
- Examples
- References

- The basic idea of Zamansky et al. (2006) is to build a natural logic proof system directly on top of the Lambek Calculus. The resulting system avoids the kind of algorithm Sánchez-Valencia and van Eijk propose. Monotonicity reasoning is instead carried out by using the η and β rules of λ-calculus, together with monotonicity proof rules.

- The nice thing about this paper is that everything is made completely explicit. Each step of the reasoning is fully formalized.

- While the system is probably not useful for practical applications, the resulting logic is interesting and attractive. Lots of open questions about this system, and related systems, remain.

- The setup in this paper is slightly different from the version of Lambek Calculus we had before. We take the Lambek Calculus with λ-terms, but instead of marking categories with monotonicity information we will be marking types, and keeping track of types by subscripting them on λ-terms.

- The categories are there only for syntax. Once an expression is parsed, the category label will be dropped.

Definition (Marked Types)

The set $\mathcal{T}^+ \supseteq \mathcal{T}$ of marked types is given by:

- The basic types e and t are in $\mathcal{T}^+$.
- If $\sigma, \tau \in \mathcal{T}^+$, then: $\sigma \rightarrow \tau \in \mathcal{T}^+$, $\sigma \xrightarrow{+} \tau \in \mathcal{T}^+$, and $\sigma \xrightarrow{-} \tau \in \mathcal{T}^+$.

- We will sometimes write $\sigma \xrightarrow{\bullet} \tau$ instead of $\sigma \rightarrow \tau$, so that we can write down our set of markings as $M = \{+, -, \bullet\}$.

- Type domains are given as usual, with the exception that a term of type $\sigma \xrightarrow{+} \tau$ is always interpreted as a *monotone* function in $D_\tau^{D_\sigma}$, and a term of type $\sigma \xrightarrow{-} \tau$ is interpreted as an *antitone* function.

- Since terms of types $\sigma \rightarrow \tau$, $\sigma \xrightarrow{+} \tau$, and $\sigma \xrightarrow{-} \tau$ are all interpreted in the same domain, it may sometimes be useful to compare them. (Recall that $\leq$ relations are inherited all the way up the hierarchy.) This motivates the following definition

Definition (Unmarked Types)

If $\tau \in \mathcal{T}^+$, let τ° be the result of erasing all $+$'s and $-$'s. That is,

- If τ is a basic type, $\tau^\circ = \tau$.
- If τ is either $\sigma \rightarrow \rho$, $\sigma \xrightarrow{+} \rho$, or $\sigma \xrightarrow{-} \rho$, then $\tau^\circ = \sigma^\circ \rightarrow \rho^\circ$.

- If $\tau^\circ = \sigma^\circ$ we will write $\tau \equiv \sigma$. Hence $\mathcal{T} \cong \mathcal{T}^+_{\equiv}$.

- As a side note (which will come up again in a later lecture), since we are restricting the interpretations of terms with marked functional types, we must ensure that arguments to functions are appropriate, which is no longer guaranteed by matching categories. (In fact, if we were in LP the categories would be superfluous altogether.)

- We can define a partial order $\preceq$ on $\equiv$-equivalence classes as follows:
  - $\tau \preceq \tau$ for all τ ;
  - $\sigma \xrightarrow{m} \tau \preceq \sigma \rightarrow \tau$, for $m \in \{+, -\}$.
  - If $\tau \preceq \tau'$ and $\sigma \preceq \sigma'$, then $\sigma' \xrightarrow{m} \tau \preceq \sigma \xrightarrow{m} \tau'$, for any $m \in \{+, -, \bullet\}$.

- $\tau \preceq \sigma$ can be read,

  Anything of type τ can also be interpreted as of type σ.

- We require that $\sigma \preceq \sigma'$ in order for $t_{\sigma' \xrightarrow{m} \tau}$ to take u_σ as argument.

- In our lexicon, instead of, e.g.:

  every, $\lambda x.\lambda y.\forall z(x(z) \rightarrow y(z)) : (s \backslash iv)/n$

  or

  every, $(s \backslash iv^{+})/n^{-}$

  we will write

  every$_{(e \rightarrow t)\overset{-}{\rightarrow}((e \rightarrow t)\overset{+}{\rightarrow}t)}$: $(s \backslash iv)/n$.

- That is, we keep only as much of the meaning as we will need for monotonicity reasoning.

- Throughout let us abbreviate $e \rightarrow t$ as p, the 'predicate type'. With that our lexical entry becomes:

  every$_{p\overset{-}{\rightarrow}(p\overset{+}{\rightarrow}t)}$: $(s \backslash iv)/n$.

- Recall the (associative) Lambek Calculus L.

$$(\mathrm{Ax})\ \frac{}{x_\tau : A \blacktriangleright x_\tau : A}$$

$$(/E)\ \frac{\Delta \blacktriangleright t_{\sigma' \xrightarrow{m} \tau} : A/B \qquad \Gamma \blacktriangleright u_\sigma : B}{\Delta \circ \Gamma \blacktriangleright t(u)_\tau : A}$$

$$(\backslash E)\ \frac{\Gamma \blacktriangleright u_\sigma : B \qquad \Delta \blacktriangleright t_{\sigma' \xrightarrow{m} \tau} : B\backslash A}{\Gamma \circ \Delta \blacktriangleright t(u)_\tau : A}$$

$$(/I)\ \frac{\Delta \circ x_\tau : A \blacktriangleright t_\sigma : B}{\Delta \blacktriangleright (\lambda x.t)_{\sigma\to\tau} : B/A}$$

$$(\backslash I)\ \frac{x_\tau : A \circ \Delta \blacktriangleright t_\sigma : B}{\Delta \blacktriangleright (\lambda x.t)_{\sigma\to\tau} : B\backslash A}$$

- In all of the above, $\textsc{type}(A) = \tau^\circ$, $\textsc{type}(B) = \sigma^\circ$, and $\sigma \preceq \sigma'$.

- Again, we shall write $\mathsf{L} \vdash \Gamma \blacktriangleright t : A$.

- Given a grammar $\mathcal{G} = \langle \Sigma, \mathsf{CAT}, \mathsf{LEX} \rangle$, the language of $\mathcal{G}$ is:

$$\mathcal{L}[\mathcal{G}] = \{\mathbf{w} \mid \exists\, \Gamma \in \ \mathsf{LEX}, \text{ such that } \mathsf{L} \ \vdash \Gamma \blacktriangleright \mathbf{w} : s\}.$$

- We can also specify the language of $\mathcal{G}$ for other types τ. Where CAT^τ is the set of categories A for which $(\textsc{type}(A))^\circ = \tau$, then:

$$\mathcal{L}^\tau[\mathcal{G}] = \{\mathbf{w} \mid \exists\, \Gamma \in \ \mathsf{LEX}, A \in \mathsf{CAT}^\tau, \text{ s.t. } \mathsf{L} \vdash \Gamma \blacktriangleright \mathbf{w} : A\}.$$

  So in particular, $\mathcal{L}[\mathcal{G}] = \mathcal{L}^t[\mathcal{G}]$.

- The central contribution of Zamansky et al. (2006) is an *order-calculus* over the languages given by the Lambek Calculus.
- For each undecorated type τ, we define an ordering $\leq_\tau$ between terms whose undecorated type is τ, i.e. $\leq_\tau \subseteq \mathcal{L}^\tau(\mathcal{G}) \times \mathcal{L}^\tau(\mathcal{G})$.
- The calculus OC is given by the axioms and rules on the next page.

Axioms and Rules of OC

$$\text{(refl)}\ \frac{}{t_\tau \leq_{\tau^\circ} t_\tau} \qquad \text{(trans)}\ \frac{t \leq_{\tau^\circ} s \qquad s \leq_{\tau^\circ} u}{t \leq_{\tau^\circ} u}$$

$$\text{(mono)}\ \frac{u \leq_{\sigma^\circ} v}{t_{\sigma \overset{+}{\to} \tau}(u) \leq_{\tau^\circ} t_{\sigma \overset{+}{\to} \tau}(v)} \qquad \text{(anti)}\ \frac{v \leq_{\sigma^\circ} u}{t_{\sigma \overset{-}{\to} \tau}(u) \leq_{\tau^\circ} t_{\sigma \overset{-}{\to} \tau}(v)}$$

$$\text{(repl)}\ \frac{t \leq_{(\sigma \to \tau)^\circ} s \qquad u \equiv_{\sigma^\circ} v}{t(u) \leq_{\tau^\circ} s(v)} \qquad \text{(abstr)}\ \frac{t \leq_{\tau^\circ} s}{\lambda x_\sigma.t \ \leq_{(\sigma \to \tau)^\circ} \ \lambda x_\sigma.s}$$

$$(\beta)\ \frac{}{(t[x_\sigma / u_\sigma])_\tau \equiv_{\tau^\circ} (\lambda x_\sigma.t)(u_\sigma)} \qquad (\eta)\ \frac{}{t_{\sigma \overset{m}{\to} \tau} \ \equiv_{(\sigma \to \tau)^\circ} \ \lambda x_\sigma t(x_\sigma)}$$

- If there is a proof of $t \leq_\tau s$ in OC, we write $\vdash_{\mathsf{OC}} t \leq_\tau s$.

- Using the standard interpretation of λ-terms in models $\mathcal{M}$ under assignments g, it is possible to show Soundness:

Theorem (Zamansky)

If $\vdash_{\mathsf{OC}} t \leq_\tau s$, *then for all* $\mathcal{M}$ *and* g, $[\![t]\!]_{\mathcal{M},g} \leq [\![s]\!]_{\mathcal{M},g}$.

- Completeness seems to be open.

A Natural Logic Proof System

- The last step is to define a proof system over words and other expressions of category s, type t, using a special expression $\top$. This element will play the role of constant *true*.

- Given expressions t_1, ..., t_n of category s, let us say,

$$t_1, ..., t_n \vdash t,$$

just in case,

$$\vdash_{OC} \top \leq t_1, ..., \vdash_{OC} \top \leq t_n \text{ together imply } \vdash_{OC} \top \leq t.$$

- That is, whenever t_1, ..., t_n are true, so is t.

- Soundness of this calculus follows from soundness of OC.

- Let the lexicon be as follows:
  - Theodore, np
  - candidate, n
  - every, $(s/iv)/n$
  - some, $(s/iv)/n$
  - obsequious, (n/n)
  - deferential, (n/n)
  - broccoli, np
  - likes, tv
  - adores tv
  - runs, iv
  - who, $(n\backslash n)/iv$
  - excessively, $(n/n)/(n/n)$
- In this setting, categories are not marked, types are. Some of the interesting cases of type markings are as follows:
  - every, $p \xrightarrow{-} (p \xrightarrow{+} t)$
  - some, $p \xrightarrow{+} (p \xrightarrow{+} t)$
  - who, $p \xrightarrow{+} (p \to p)$

  The other types are given by the type assignment to categories.
- We can also add various meaning postulates as order axioms:
  - adores $\leq_{e\to(p)}$ likes
  - λx_p.excessively(deferential(x)) $\leq_{p\to p}$ obsequious
  - ...

Example 1

Every candidate who Theodore likes runs $\vdash$ Every candidate who Theodore adores runs.

$$\dfrac{\textsf{adores} \leq_{e \to p} \textsf{likes}}{\textsf{adores}(x_e) \leq_p \textsf{likes}(x_e)}\ (\text{repl})$$

$$\dfrac{\textsf{adores}(x_e) \leq_p \textsf{likes}(x_e)}{\textsf{Theodore}_{p \overset{+}{\to} t}(\textsf{adores}(x)) \leq_t \textsf{Theodore}_{p \overset{+}{\to} t}(\textsf{likes}(x))}\ (\text{mono})$$

$$\dfrac{\textsf{Theodore}_{p \overset{+}{\to} t}(\textsf{adores}(x)) \leq_t \textsf{Theodore}_{p \overset{+}{\to} t}(\textsf{likes}(x))}{\lambda x.\textsf{Theodore}(\textsf{adores}(x)) \leq_p \lambda x.\textsf{Theodore}(\textsf{likes}(x))}\ (\text{abstr})$$

$$\dfrac{\lambda x.\textsf{Theodore}(\textsf{adores}(x)) \leq_p \lambda x.\textsf{Theodore}(\textsf{likes}(x))}{\textsf{who}(\lambda x.\textsf{Theodore}(\textsf{adores}(x))) \leq_{p \to p} \textsf{who}(\lambda x.\textsf{Theodore}(\textsf{likes}(x)))}\ (\text{mono})$$

$$\dfrac{\textsf{who}(\lambda x.\textsf{Theodore}(\textsf{adores}(x))) \leq_{p \to p} \textsf{who}(\lambda x.\textsf{Theodore}(\textsf{likes}(x)))}{[\textsf{candidate}]\textsf{who}(\lambda x.\textsf{Theodore}(\textsf{adores}(x))) \leq_p [\textsf{candidate}]\textsf{who}(\lambda x.\textsf{Theodore}(\textsf{likes}(x)))}\ (\text{repl})$$

$$\dfrac{[\textsf{candidate}]\textsf{who}(\lambda x.\textsf{Theodore}(\textsf{adores}(x))) \leq_p [\textsf{candidate}]\textsf{who}(\lambda x.\textsf{Theodore}(\textsf{likes}(x)))}{\textsf{every}([\textsf{candidate}]\textsf{who}(\lambda x.\textsf{Theodore}(\textsf{likes}(x)))) \leq_{p \to t} \textsf{every}([\textsf{candidate}]\textsf{who}(\lambda x.\textsf{Theodore}(\textsf{adores}(x))))}\ (\text{anti})$$

$$\dfrac{\textsf{every}([\textsf{candidate}]\textsf{who}(\lambda x.\textsf{Theodore}(\textsf{likes}(x)))) \leq_{p \to t} \textsf{every}([\textsf{candidate}]\textsf{who}(\lambda x.\textsf{Theodore}(\textsf{adores}(x))))}{(\textsf{every}([\textsf{candidate}]\textsf{who}(\lambda x.\textsf{Theodore}(\textsf{likes}(x)))))\textsf{runs} \leq_t (\textsf{every}([\textsf{candidate}]\textsf{who}(\lambda x.\textsf{Theodore}(\textsf{adores}(x)))))\textsf{runs}}\ (\text{repl})$$

(N.B. When we type raise 'Theodore', showing that L $\vdash$ Theodore: e ► Theodore: $p \to t$, the resulting function will always be a monotone function; $[\![\textsf{Theodore}_{p \to t}]\!]$ will denote the evaluation functional on the individual, necessarily monotonic.)

Example 2

Some excessively deferential candidate likes broccoli $\vdash$ Some obsequious candidate likes broccoli.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\text{excessively(deferential(candidate))} \equiv \lambda x.\text{excessively(deferential}(x))\text{(candidate)}}\,(\beta) \quad \dfrac{\lambda x.\text{excessively(deferential}(x)) \leq \text{obsequious}}{\lambda x.\text{excessively(deferential}(x))\text{(candidate)} \leq \text{obsequious(candidate)}}\,(\text{repl})}{\text{excessively(deferential(candidate))} \leq \text{obsequious(candidate)}}\,(\text{trans})}{\text{some(excessively(deferential(candidate)))} \leq \text{some(obsequious(candidate))}}\,(\text{mono})}{\text{some(excessively(deferential(candidate)))(likes broccoli)} \leq \text{some(obsequious(candidate))(likes broccoli)}}\,(\text{repl})$$

Further topics and issues

- The paper by Zamasky et al. shows that their proof system derives everything that is derivable in the Sánchez-Valencia / van Eijk version of Monotonicity Calculus.

- The paper considers additional markings, not just $+$ and $-$. For example, for modifier types $\tau \rightarrow \tau$, they focus on *restricted modifiers* t for which $t(u) \leq_\tau u$. This has a special rule, where R stands for restrictive:

$$\text{(rmod)}\ \frac{}{t_{\sigma_1 \overset{R}{\rightarrow} \tau}(u_{\sigma_2}) \leq_{\tau^\circ} u_{\sigma_2}}$$

- They also consider markings and corresponding rules for l.u.b. and g.l.b. operators of type $\tau \rightarrow (\tau \rightarrow \tau)$, such as 'or' and 'and'.

- As was already mentioned, a number of interesting (meta-)logical questions remain: How powerful is the proof system? Is it complete? Is the theory of standard models over this language decidable?

A. Zamansky, N. Francez, and Y. Winter. 'A 'Natural Logic' Inference System Using the Lambek Calculus', *Journal of Logic, Language, and Information*, 15(3): 273-295, 2006.