A Note on Negation in Categorial Grammar

H.Wansing 2006

Éricles – Giacomo - Warrick

16^{th} October 2020

Recapitulation

Warrick In case you forgot (or were sleeping during the) the presentation of Categorial Grammars, the **goal** is to obtain a **system to allow the syntesis and analysis of sentences/formulas**.

1. types such as "$(n\backslash s)$", "n";
2. Has ways of constructing new types and a grammar for "well-formedness";
3. A (finite set) of symbols (such as "poor" and "John"), to which are matched said types.

Overview

1. *Negation* is a contentions notion, in a sense (but we'll not get into this here)
2. Buszkowski added axioms for a kind of negation in categorial grammars
3. Wansing presents a different kind, which leads to motivation of connexive logic

Negative Information

To allow for the expression that "'sleeps John' is an invalid sentence" (it's not just "not valid", it's *invalid*, one could assign it the type "$\neg(s\backslash n)$"
There are many nice connections to algebra, and even category theory (Lambek calculus was inspired on that), but we won't be touching upon (no time).

The Negation Normal Form

Observation 3.4

For every type symbol x, x' is in NNF and $\vdash_S x \Leftrightarrow x'$, for $S \in \{\mathbf{NL}^\neg, \mathbf{L}^\neg\}$.

Definition 3.1: type symbol

1. atomic type symbols $x, y, w, \ldots$ are type symbols;
2. if X and Y are type symbols, also $(X * Y)$ is a type symbol, for $* \in \{\backslash, /, x\}$
3. if X is a type symbol, also $\neg X$ is a type symbol $(X \neq Y \times Z)$;

Negation Normal Form

Define a function $'$ such that:

$$
\begin{aligned}
x' &= x \quad \text{(x atomic)} \\
(\neg x)' &= \neg x \quad \text{(x atomic)} \\
(\neg\neg X)' &= X' \\
(X * Y)' &= (X' * Y') \\
(\neg(Y/X))' &= ((\neg Y)'/X') \\
(\neg(X\backslash Y))' &= (X'\backslash(\neg Y)')
\end{aligned}
$$

Proof:

A. x is a type symbol $\rightarrow$ x' is its negation normal form;

B. x is a type symbol $\rightarrow \vdash_S x \Leftrightarrow x'$

Proof:

A. x is a type symbol $\rightarrow$ x' is its negation normal form;

B. x is a type symbol $\rightarrow$ $\vdash_S x \Leftrightarrow x'$

A.

Proof is straightforward by induction on the complexity of type symbols.

(note: $\neg(X * Y)$ is not a valid type symbol here).

B. ($\Rightarrow$ *case*)

By induction on the complexity of x:

1. x is atomic: $x' = x$ and by $(id) \vdash x \Rightarrow x'$;

B. ($\Rightarrow$ *case*)

By induction on the complexity of x:

1. x is atomic: $x' = x$ and by $(id) \vdash x \Rightarrow x'$;
2. $x = (y \times w)$: by IH $\vdash y \Leftrightarrow y'$ and $\vdash w \Leftrightarrow w'$

$$\dfrac{\dfrac{y \Rightarrow y' \qquad w \Rightarrow w'}{y, w \Rightarrow (y' \times x')}\,(\rightarrow \times)}{(y \times w) \Rightarrow (y' \times x')}\,(\times \rightarrow)$$

3. $x = (y/w)$: same IH

$$\dfrac{\dfrac{w' \Rightarrow w \qquad y \Rightarrow y'}{(y/w), w' \Rightarrow y'}\,(/ \rightarrow)}{y/w \Rightarrow y'/w'}\,(\rightarrow /)$$

4. $x = (y\backslash w)$: dual.

5 $x = \neg y$: by IH $\vdash y \Leftrightarrow y'$
We can't deduce $\neg y \Leftrightarrow (\neg y)'$ directly. We need to look at y:

5 $x = \neg y$: by IH $\vdash y \Leftrightarrow y'$
We can't deduce $\neg y \Leftrightarrow (\neg y)'$ directly. We need to look at y:

- y atomic then $(\neg y)' = \neg y$ and $\vdash \neg y \Rightarrow \neg y$ by (id);
- $y = \neg w$. We want a proof of $\neg\neg w \Leftrightarrow (\neg\neg w)'$. By IH we know that $\vdash w \Leftrightarrow w'$

$$\dfrac{\dfrac{w \Rightarrow w'}{\neg\neg w \Rightarrow w'}\,(\neg\neg \rightarrow)}{\neg\neg w \Rightarrow \neg\neg w'}\,(\rightarrow \neg\neg)$$

5 $x = \neg y$: by IH $\vdash y \Leftrightarrow y'$
We can't deduce $\neg y \Leftrightarrow (\neg y)'$ directly. We need to look at y:

- y atomic then $(\neg y)' = \neg y$ and $\vdash \neg y \Rightarrow \neg y$ by (id);
- $y = \neg w$. We want a proof of $\neg\neg w \Leftrightarrow (\neg\neg w)'$. By IH we know that $\vdash w \Leftrightarrow w'$

$$\dfrac{\dfrac{w \Rightarrow w'}{\neg\neg w \Rightarrow w'}\,(\neg\neg \rightarrow)}{\neg\neg w \Rightarrow \neg\neg w'}\,(\rightarrow \neg\neg)$$

- $y = w/z$. We want a proof of $\neg(w/z) \Rightarrow ((\neg w')/z')$ since $(\neg(w/z))' = ((\neg w')/z')$ By IH we can assume

$$z \Leftrightarrow z' \qquad \neg w \Leftrightarrow (\neg w)'$$

$$\dfrac{\dfrac{z' \Rightarrow z \qquad \neg w \Rightarrow (\neg w)'}{\neg(w/z), z' \Rightarrow (\neg w)'}\,(\neg/ \rightarrow)}{\neg(w/z) \Rightarrow ((\neg w)'/z')}\,(\rightarrow /)$$

Definition 3.1: type symbol

1. Atomic type symbols $x, y, w, \ldots$ are type symbols;
2. if X and Y are type symbols, also $(X * Y)$ is a type symbol, for $* \in \{\backslash, /, x\}$
3. if X is a type symbol, also $\neg X$ is a type symbol $(X \neq Y \times Z)$;
4. nothing else is a type symbol.

```
data tSymb : Set where
  base : Nat → tSymb
  ~ : tSymb → tSymb
  _\\_ : tSymb → tSymb → tSymb
  _//_ : tSymb → tSymb → tSymb
```

Definition 3.2: categorial entailment

$$\frac{}{x \Rightarrow x}\ (id)$$

$$\frac{x, X \Rightarrow y}{X \Rightarrow (x \backslash y)}\ (\rightarrow \backslash)$$

$$\frac{X \Rightarrow x \qquad Y, y, Y' \Rightarrow z}{Y, X, (x \backslash y), Y' \Rightarrow z}\ (\backslash \rightarrow)$$

```
Ctx : Set
Ctx = List tSymb

data _=>_ : Ctx → tSymb → Set where
  id-axiom : (x : tSymb) → [ x ] => x
  \\r : (Γ : Ctx )(x y : tSymb)
        → ( x , Γ) => y
        → Γ => ( x \\ y )
  \\l : (Δ Δ' Γ : Ctx )(x y z : tSymb)
        → (Δ ++ [ y ] ++ Δ') => z
        → Γ => x
        ----------------------------
        → (Δ ++ Γ ++ [ x \\ y ] ++ Δ')
          => z
```