Semantics and Pragmatics of NLP
DRT: Constructing LFs and Presuppositions

Alex Lascarides

School of Informatics
University of Edinburgh
Outline

1. Constructing DRSs for Discourse
2. Pronouns and Presuppositions
Add $\lambda$ and @ operators and a merge operator $\oplus$.

Use these operators to build representations *compositionally*, but the pronouns aren’t resolved at this stage, so

Then we resolve the underspecified condition given by the pronoun, according to certain heuristics.
The General Picture

mary(x)  y  car(y), own(x,y)
Context

Current sentence syntax and λs

alex(x)  y  car(y), own(x,y)

Got with ⊕

z is accessible; y is not

z=x, unhappy(z)
DRS1 ⊕ DRS2 = DRS3, where:

1. DRS3’s discourse referents is the set union of DRS1’s and DRS2’s discourse referents.

2. DRS3’s conditions is the set union DRS1’s and DRS2’s conditions.

\[
\begin{align*}
\text{DRS1:} & \\
 & x \\
 & \text{john(x)} \\
 & y \\
 & \neg \text{car(y), own(x,y)}
\end{align*}
\]

\[
\begin{align*}
\text{DRS2:} & \\
 & z \\
 & \text{z=?}, \text{unhappy(z)}
\end{align*}
\]

\[
\begin{align*}
\text{DRS3:} & \\
 & x, z \\
 & \text{john(x)} \\
 & y \\
 & \neg \text{car(y), own(x,y)} \\
 & \text{z=?}, \text{unhappy(z)}
\end{align*}
\]
Lexical Items: Nouns and Intransitive Verbs

boxer: \( \lambda y \) \( \text{boxer}(y) \)

woman: \( \lambda y \) \( \text{woman}(y) \)

dances: \( \lambda y \) \( \text{dance}(y) \)

Do pronouns later, since they’re different from what we had before...
Determiners and Proper Names

\[ a: \lambda P \lambda Q z \quad \lambda \lambda Q \oplus P@z \equiv Q@z \]

\[ \text{every: } \lambda P \lambda Q z \quad \lambda \lambda Q \oplus P@z \implies Q@z \]

\[ \text{Mia: } \lambda P x \quad \lambda mia(x) \oplus P@x \]

Will change proper names a bit later…
Every woman dances (S)

\[
\begin{array}{c}
z \\
\text{woman}(z) \\
\end{array} \quad \Rightarrow 
\begin{array}{c}
dance(z) \\
\end{array}
\]

Every woman (NP)

\[
\lambda Q\ z \\
\text{woman}(z) \Rightarrow Q@z
\]

dances (VP)

\[
\lambda y \\
dance(y)
\]

every (DET)

\[
\lambda P\lambda Q\ z \quad \oplus P@z \Rightarrow Q@z
\]

woman (N)

\[
\lambda x \\
\text{woman}(x)
\]

dances (IV)

\[
\lambda y \\
dance(y)
\]
DRSs in NLTK

\[
\text{DRS}([], [(\text{man } x)] \Rightarrow \text{DRS}([y], [(\text{bicycle } y), (\text{owns } y x)])]
\]

- `toFol()` : Converts DRSs to FoL.
- `draw()` : Draws a DRS in ‘box’ notation (currently works only for Windows).
- NLTK grammar adapts lambda abstracts so that their bodies are DRSs rather than FoL expressions.
Presuppositions

- Are a way of conveying information as if it’s taken for granted;
- Are different from entailments because they survive under negation:
  
  \[
  \text{John loves his wife} \quad \rightarrow \quad \text{John loves someone} \\
  \quad \rightarrow \quad \text{John has a wife.}
  \]

  \[
  \text{John doesn’t love his wife} \quad \not\rightarrow \quad \text{John loves someone} \\
  \quad \rightarrow \quad \text{John has a wife.}
  \]

- Behave a bit like pronouns; *anaphora*...
Presupposition Triggers

Presuppositions are triggered by certain words and phrases:
- *the, manage, her, regret, know, again*, proper names, possessive marker, . . .
- comparatives: John is a better linguist than Bill
- it-clefts: It was Fred who ate the beans

To **Test** whether you’re dealing with a presupposition:
- Negate the sentence or stick a modality (e.g., *might*) in it. Does the inference survive? If so, it’s a presupposition.
The Projection Problem

When there’s a presupposition trigger in a complex sentence, is the (potential) presupposition it triggers a presupposition of the whole sentence?

(1)  

a. If baldness is hereditary, John’s son is bald. 
   yes; presupposition semantically outscopes conditional

b. If John has a son, then John’s son is bald. 
   no; presupposition doesn’t semantically outscope conditional
Presuppositions as Anaphora

Indefinite Antecedents

(2)  
   a. Theo has a little rabbit, and his rabbit is grey.
   b. Theo has a little rabbit, and it is grey.

(3)  
   a. If Theo has a rabbit, his rabbit is grey.
   b. If Theo has a rabbit, it is grey.

Presupposition ‘cancelled’.

Conjecture:

- Presupposition cancellation like binding anaphora.
Presuppositions are Anaphora with Semantic Content
Van der Sandt

- *she*: female
  - *His wife*: she’s married, female, human, adult,...

- Presupposition binds to antecedent if it can:

  (4) If John has a wife, then *his wife* will be happy.

- Otherwise it’s *accommodated*:
  - The presupposition is *added* to the context.
  - The process of binding and accommodating determines the semantic scope of the presupposition and so solves the Projection Problem.
The Details of the Story

Three tasks:

1. Identify presupposition triggers in the lexicon; and
2. Indicate what they presuppose (separating it from the rest of their content, since presuppositions are interpreted differently);
3. Implement the process of binding and accommodation for presuppositions
Tasks 1 and 2

Triggers (Task 1):
- *the*, possessive constructions, proper names, …

DRS-representation (Task 2):
- Extend the DRS language with an $\alpha$ operator.
- This separates DRSs representing presupposed information from DRSs which aren’t presupposed.

\[
\text{the waitress: } \lambda P. \bigoplus P@x \alpha x \rightarrow \text{waitress}(x)
\]
Representing More Presupposition triggers (including pronouns!)

Mia: $\lambda P \oplus P@x\alpha \quad mia(x)$

he: $\lambda P \oplus P@x\alpha \quad male(x)$

his: $\lambda P \lambda Q \oplus P@x\alpha((x \quad own(y,x) \oplus Q@x)\alpha \quad male(y))$
A Clearer Notation: $\alpha$-bits to double-lined boxes

Mia: $\lambda P \; mia(x)$

he: $\lambda P \; male(x)$

his: $\lambda P \; \lambda Q \; own(y,x)$

$\oplus P@x$

$\oplus Q@x$

$\oplus P@x$
Constructing DRSs

Pronouns and Presuppositions

DRS Construction

The waitress smiles (S)

The waitress (NP)  smiles (VP)

The (DET)  waitress (N)

Alex Lascarides

SPNLP: Presuppositions
The Presupposition Resolution Algorithm

1. Create a DRS for the input sentence with all presuppositions marked with $\alpha$. Merge this DRS with the DRS for the discourse so far (using $\oplus$). Go to step 2.

2. Traverse the DRS, and on encountering an $\alpha$-marked DRS try to:
   1. link the presupposed information to an accessible antecedent with the same content. Go to step 2.
   2. otherwise, accommodate it in the highest accessible site, subject to it being *consistent* and *informative*. Go to step 2.
   3. otherwise, return *presupposition failure*.

   otherwise, go to step 3.

3. Reduce any merges appearing in the DRS.
After adding the presupposed material, the resulting DRS must be *satisfiable*.

(5) John hasn’t got a wife. He loves his wife.  \(\text{no!}\)
(6) John hasn’t got a mistress. He loves his wife.  \(\text{yes!}\)
Adding the presupposed material should not render any of the *asserted* material redundant.

\[(7)\] Either there is no bathroom or the bathroom is in a funny place.

Note binding isn’t possible (because x isn’t accessible)
Constructing DRSs
Pronouns and Presuppositions

Accommodating the bathroom

- **Global accommodation** gives \( p \land (\neg p \lor q) \), which is equivalent to \( p \land q \), and so violates informativeness.
- **Local accommodation** gives \( \neg p \lor (p \land q) \), and this satisfies informativeness.

\[
\neg x \quad \lor \\
\text{bathroom}(x) \\
\quad \quad \quad \\
\lor \\
y \\
\quad \quad \\
\text{bathroom}(y) \quad \text{funny-place}(y)
\]
There is no accessible $y$ and $\text{waitress}(y)$, so it can’t be bound.
Therefore, it must be added.
There’s only one accessible site.
Adding the presupposition to this site is consistent and informative.
And so it’s added there.
(1) a. If baldness is hereditary, then John’s son is bald.

b. If John has a son, then John’s son is bald.

\[
\begin{array}{c}
\text{a'} \\
\text{If baldness is hereditary, then John’s son is bald.} \\
\text{b'} \\
\text{If John has a son, then John’s son is bald.}
\end{array}
\]
If baldness is hereditary, then John’s son is bald

If baldness is hereditary, then John’s son is bald
If John has a son, then John's son is bald.

```
\[\begin{array}{c}
\text{X} \\
\text{john(x)} \\
\hline
\text{w} \\
\text{son(w), has(x,w)} \\
\text{x} \\
\text{john(x)} \\
\hline
\text{y} \\
\text{son(y), has(z,y)} \\
\text{z} \\
\text{john(z)} \\
\hline
\text{bald(y)} \\
\text{son(y), has(z,y)} \\
\text{w} \\
\text{son(w), has(x,w)} \\
\\end{array}\]

\Rightarrow

\[\begin{array}{c}
\text{X} \\
\text{john(x)} \\
\hline
\text{w} \\
\text{son(w), has(x,w)} \\
\text{\Rightarrow} \\
\text{bald(y)} \\
\text{son(y), has(z,y)} \\
\text{z} \\
\text{john(z)} \\
\text{\Rightarrow} \\
\text{w} \\
\text{son(w), has(x,w)} \\
\text{\Rightarrow} \\
\text{bald(w)} \\
\end{array}\]
```
DRT is an elegant framework for representing the content of discourse, because it handles inter-sentential anaphoric dependencies, and in particular it provides an elegant solution to the projection problem. But right now we’ve ignored *pragmatics*: DRT still only uses *linguistic* information to compute meaning. Non-linguistic information also influences interpretation! We’ll examine pragmatics for the rest of the course.