Semantics and Pragmatics of NLP
DRT: Constructing LFs and Presuppositions

Alex Lascarides

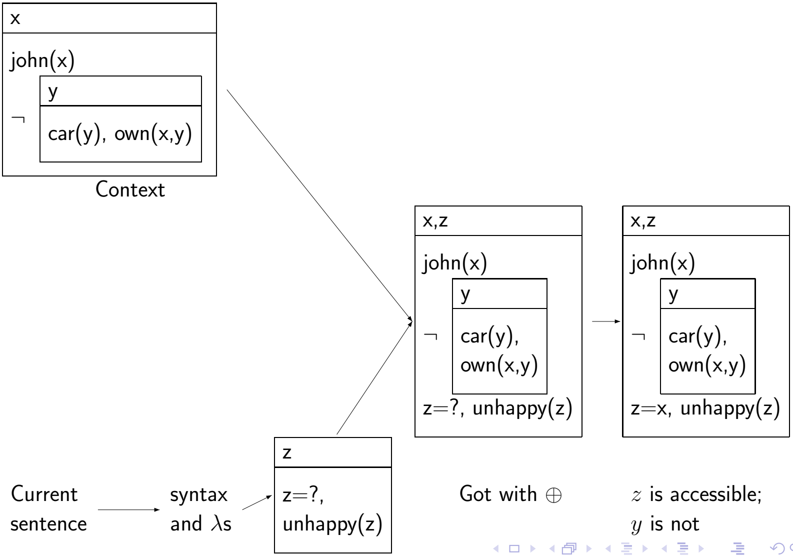
School of Informatics
University of Edinburgh

Outline

Building DRSs with Lambdas: λ-DRT

- Add λ 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.

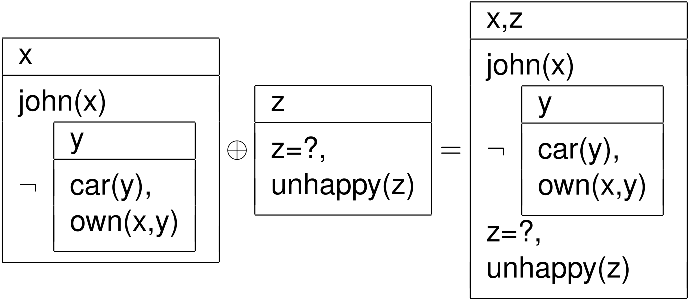
The General Picture



Merging

DRS1 $\oplus$ 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.



Lexical Items: Nouns and Intransitive Verbs

boxer: $\lambda y \begin{array}{|l|}\hline \\ \hline \text{boxer(y)} \\ \hline\end{array}$

woman: $\lambda y \begin{array}{|l|}\hline \\ \hline \text{woman(y)} \\ \hline\end{array}$

dances: $\lambda y \begin{array}{|l|}\hline \\ \hline \text{dance(y)} \\ \hline\end{array}$

Do pronouns later, since they're different from what we had before…

Determiners and Proper Names

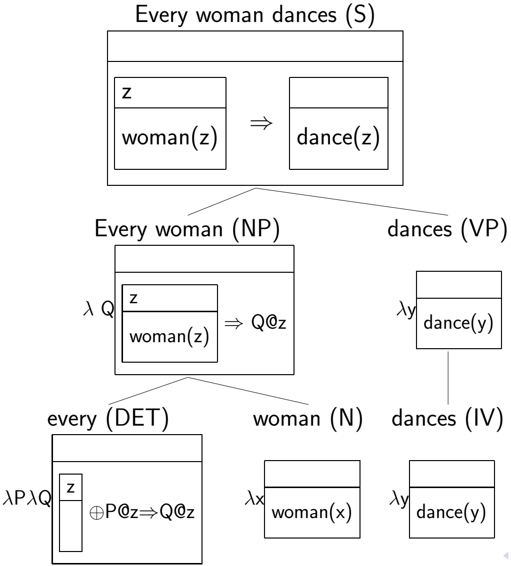
$$\text{a:} \quad \lambda P \lambda Q \begin{array}{|c|}\hline z \\ \hline \\ \\ \hline \end{array} \oplus P@z \oplus Q@z$$

$$\text{every:} \quad \lambda P \lambda Q \begin{array}{|c|}\hline \\ \hline \begin{array}{|c|}\hline z \\ \hline \\ \\ \hline \end{array} \oplus P@z \Rightarrow Q@z \\ \hline \end{array}$$

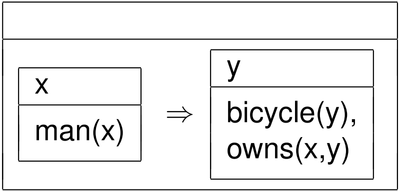
$$\text{Mia:} \quad \lambda P \begin{array}{|c|}\hline x \\ \hline \text{mia}(x) \\ \hline \end{array} \oplus P@x$$

Will change proper names a bit later…

DRS Construction



DRSs in NLTK




```
DRS([],[(DRS([x],[(man x)]) implies
     DRS([y],[(bicycle y),(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.

More on Anaphora

Presuppositions

- Are a way of conveying information as if it's taken for granted;
- Are different from entailments because they survive under negation:

$$\begin{array}{rl} \text{John loves his wife} & \rightarrow \text{John loves someone} \\ & \rightarrow \text{John has a wife.} \\ \text{John doesn't love his wife} & \not\rightarrow \text{John loves someone} \\ & \rightarrow \text{John has a wife.} \end{array}$$

- 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 <u>his wife</u> 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 α operator.
- This separates DRSs representing presupposed information from DRSs which aren't presupposed.

the waitress: $\lambda P \begin{array}{|c|}\hline \quad \\ \hline \quad \\ \hline\end{array} \oplus P@x \ \alpha \begin{array}{|l|}\hline x \\ \hline \text{waitress}(x) \\ \hline\end{array}$

Representing More Presupposition triggers (including pronouns!)

Mia: $\lambda P \begin{array}{|c|}\hline \\ \hline \\ \hline\end{array} \oplus P@x \, \alpha \begin{array}{|l|}\hline x \\ \hline mia(x) \\ \hline\end{array}$

he: $\lambda P \begin{array}{|c|}\hline \\ \hline \\ \hline\end{array} \oplus P@x \, \alpha \begin{array}{|l|}\hline x \\ \hline male(x) \\ \hline\end{array}$

his: $\lambda P \lambda Q \begin{array}{|c|}\hline \\ \hline \\ \hline\end{array} \oplus P@x \, \alpha ((\begin{array}{|l|}\hline x \\ \hline own(y,x) \\ \hline\end{array} \oplus Q@x) \alpha \begin{array}{|l|}\hline y \\ \hline male(y) \\ \hline\end{array})$

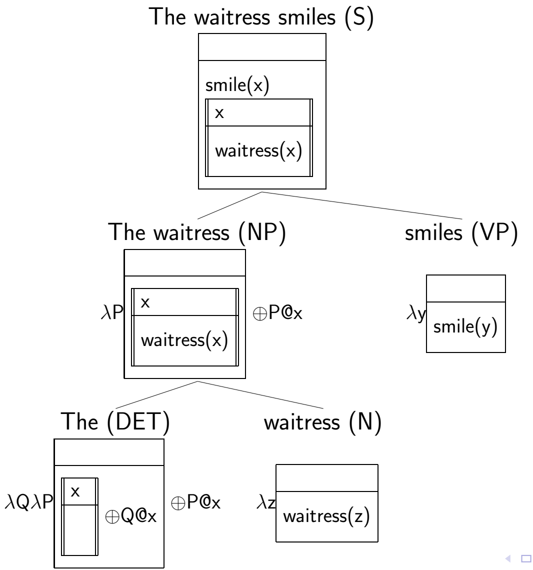
A Clearer Notation: α-bits to double-lined boxes

$$\text{Mia: } \lambda P \begin{array}{|c|}\hline \quad \\ \hline \begin{array}{||l||}\hline x \\ \hline \text{mia}(x) \\ \hline \end{array} \\ \hline \end{array} \oplus P@x$$

$$\text{he: } \lambda P \begin{array}{|c|}\hline \quad \\ \hline \begin{array}{||l||}\hline x \\ \hline \text{male}(x) \\ \hline \end{array} \\ \hline \end{array} \oplus P@x$$

$$\text{his: } \lambda P \lambda Q \begin{array}{|c|}\hline \quad \\ \hline \begin{array}{||l||}\hline x \\ \hline \text{own}(y,x) \\ \begin{array}{||l||}\hline y \\ \hline \text{male}(y) \\ \hline \end{array} \\ \hline \end{array} \oplus Q@x \\ \hline \end{array} \oplus P@x$$

DRS Construction



The Presupposition Resolution Algorithm

1. Create a DRS for the input sentence with all presuppositions marked with α. 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 α-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.

Consistency

- After adding the presupposed material, the resulting DRS must be *satisfiable*.

(5)	John hasn't got a wife. He loves his wife.	*no!*
(6)	John hasn't got a mistress. He loves his wife.	*yes!*

Informativeness

- 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.

global site

$\neg$ [x | bathroom(x)] $\vee$ [*local site* funny-place(y) [y | bathroom(y)]]

Note binding isn't possible (because x isn't accessible)

Accommodating the bathroom

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

$$
\begin{array}{|c|}
\hline
\\
\hline
\begin{array}{|c|}
\hline
\\
\hline
\neg \begin{array}{|l|}
\hline
x \\
\hline
\text{bathroom(x)} \\
\hline
\end{array} \\
\hline
\end{array}
\vee
\begin{array}{|l|}
\hline
y \\
\hline
\text{bathroom(y)} \\
\text{funny-place(y)} \\
\hline
\end{array} \\
\hline
\end{array}
$$

Back to *The waitress smiles*

smile(x) x | waitress(x)

- There is no accessible y and 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.

x
waitress(x), smile(x)

Conditionals

(1) a. If baldness is hereditary, then John's son is bald.

a′

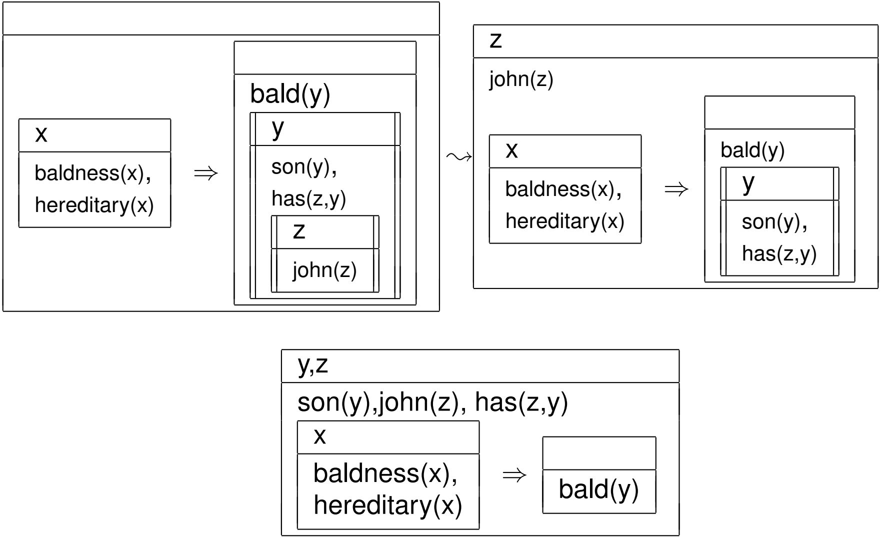
[ ]
- [x | baldness(x), hered(x)] $\Rightarrow$ [| bald(y), [y | son(y), has(z,y), [z | john(z)]]]

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

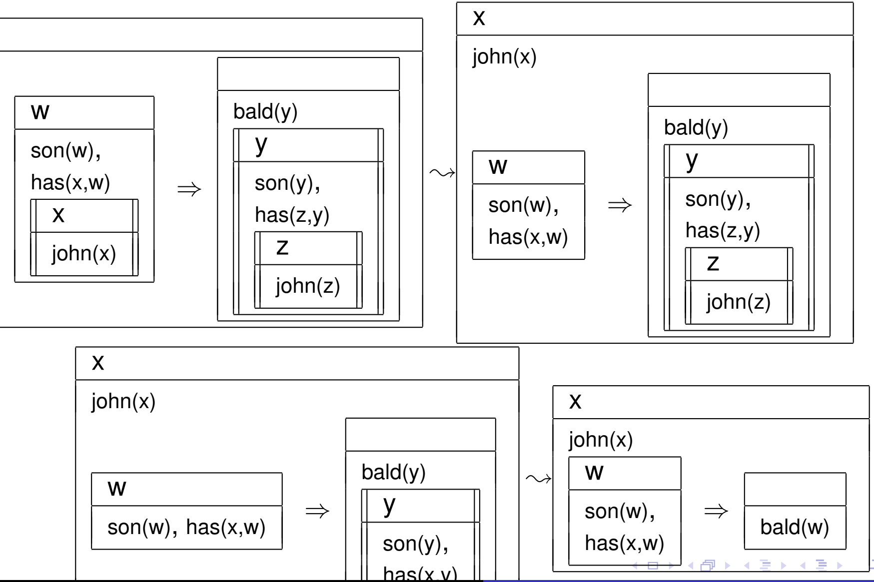
b′

[ ]
- [w | son(w), has(x,w), [x | john(x)]] $\Rightarrow$ [| bald(y), [y | son(y), has(z,y), [z | john(z)]]]

If baldness is hereditary, then John's son is bald



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



Conclusion

- 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.