Computational Semantics and Pragmatics

Autumn 2012

Raquel Fernández
Institute for Logic, Language & Computation
University of Amsterdam

Plan for Today

Wrapping up our first topic: computational formal semantics

- the Functional Programming approach
- other approaches and available resources

What's next?

- Lexical semantics

Computational Formal Semantics

Modelling the semantics of a fragment of Natural Language *à la* Montague:

- Meaning representations: formal language, e.g. FOL
  Why?
  - ∗ unambiguous
  - ∗ model-theoretical interpretation
  - ∗ reasoning
- Principle of compositionality
  - ∗ meaning of non-sentential components: lambda expressions
  - ∗ semantic composition: functional application
- Syntax-driven semantic composition
  - ∗ input: a parse tree (given a grammar and a parser)
  - ∗ output: a logical formula
    - ▸ whose truth can be evaluated on a model (of a situation/the world)
    - ▸ which can be used for reasoning

⇒ HW#2

Computational Semantics with FP

Haskell is particularly well suited to implement compositional formal semantics:

- based on the typed lambda calculus (like Montague grammar)
- FP implementations are very close to formal definitions
- declarative

Other Approaches

Other implemented approaches to computational formal semantics:

- Patrick Blackburn & Johan Bos
  http://www.blackburnbos.org
  Representation and Inference for a Natural Language
  - Prolog implementation

- Steven Bird, Ewan Klein, and Edward Loper
  http://nltk.org
  Natural Language Processing with Python
  - Python implementation
  - Chapter 10: Analyzing the Meaning of Sentences

An advantage of these approaches is that they come together with extra NLP tools and resources over and above semantic processing.

C&C Tools

Johan Bos' approach to computational semantics has been included into the so-called C&C Tools by James Currant and Stephen Clark http://svn.ask.it.usyd.edu.au/trac/candc

$\hookrightarrow$ efficient and robust tools for large-scale NLP tasks

With regards to semantic composition, the key ingredients are:

- CCG: Combinatory Categorial Grammar
- Boxer: DRT-based semantic analyzer

Johan Bos (2008) Wide-Coverage Semantic Analysis with Boxer. In: J. Bos, R. Delmonte (eds): *Semantics in Text Processing. STEP 2008 Conference Proc.*, pp.277–286, Research in Computational Semantics, College Publications.

Basics of CCG

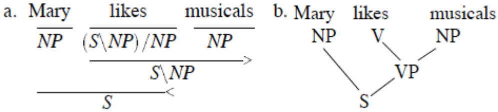
The CCG site http://groups.inf.ed.ac.uk/ccg/

Grammatical entities are associated with either a basic or a functional category that specifies the type and directionality of its arguments and the type of the result.

- "result leftmost" notation: α/β and $\alpha\backslash\beta$ represent functions from β into α, where the slash determines that the argument β is respectively to the right (/) or to the left (\) of the functor.

$$(S\backslash NP)/NP$$

- *Forward application* $(>)$
  $X/Y \;\; Y \Rightarrow X$
- *Backward application* $(<)$
  $Y \;\; X\backslash Y \Rightarrow X$



Basics of CCG

(9) Marcel proved completeness

$$\dfrac{\dfrac{\text{Marcel}}{NP_{3sm} : marcel'} \quad \dfrac{\dfrac{\text{proved}}{(S\backslash NP_{3s})/NP : \lambda x \lambda y.prove'xy} \quad \dfrac{\text{completeness}}{NP : completeness'}}{S\backslash NP_{3s} : \lambda y.prove'completeness'y}>}{S : prove'completeness'marcel'}<$$

The derivation yields the category S with a compositional interpretation, equivalent under a convention of left associativity to (10a):

(10) a. $(prove'completeness')marcel'$ b.

From Steedman & Baldridge (2003)

Boxer as part of C&C tools

- The input to Boxer is a CCG derivation generated by the C&C parser:
  - the C&C parser is a statistical CCG parser trainned on the CCGbank, a translation of the Penn Treebank into CCG derivations.
- Its output is a Discourse Representation Structure (DRS), equivalent to a FOL formula.

Basics of DRT

DRSs are conventionally represented as *boxes*. They distinguish two types of information:

- discourse referents
- conditions (properties of the referents); we can build complex conditions with the logical symbols $\neg$, $\vee$, and $\Rightarrow$ operating on DRSs.

Vincent offers Mia a drink.

x y z
x=VINCENT
y=MIA
DRINK(z)

It is blue.

x y z v
x=VINCENT
y=MIA
DRINK(z)
BLUE(v)
v=z

Kamp, H.& Reyle, U. (1993) *From Discourse to Logic*. Kluwer, Dordrecht.

A derivation with Boxer

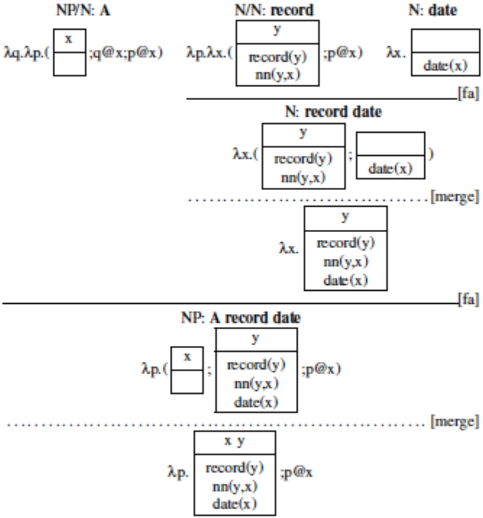


Figure 1: Derivation with λ-DRSs, including β-conversion, for "A record date". Combinatory rules are indicated by solid lines, semantic rules by dotted lines.

Summing Up

Computational formal semantics:

- automatic computation of logic-based meaning representations
- syntax-driven semantic composition
- choice of syntactic grammar / parser and meaning representation formalism
- compatible with probabilistic grammars / parsers

Most important available implementations and resources:

- Functional programming (can Eijck & Unger)
  http://www.computational-semantics.eu/
- Imperative programming (NLTK)
  http://nltk.org/
- Logic programming (Blackburn & Bos; Boxer)
  http://www.blackburnbos.org
  http://svn.ask.it.usyd.edu.au/trac/candc

What's Next

Lexical semantics: words meaning

If you don't know what **WordNet** it, find out by Friday:

- what kind of words (part of speech) are included in WordNet?
- how is WordNet organised: what are *synsets* and what semantic relation are covered?