Lecture 29: Discourse Representation Theory

CS 181O
Spring 2016
Kim Bruce

Some slides based on those of Christina Unger

What is context?

- Context is sequence of entities w/constraints
  - Manage as a stack: $c_1, c_2, \ldots, c_n$
  - Context extension pushes new item on stack
- Context transitions are functions that convert context to new context, represented as characteristic function:
  - $\lambda c\ \lambda c'$. body, where body returns true or false
  - Type $[c] \rightarrow [c] \rightarrow t$

Operations on contexts

- If c is context, let c^x represent context where add x to context c.
- Operations:
  - $\exists = \lambda c\ \lambda c'.\ \exists x.(c^\wedge x = c')$
    - $\exists\ c\ c'$ is true iff c' is extension of c.
  - Let ϕ, ψ represent context transitions
    - $\boldsymbol{\phi\ ;\ \psi} = \lambda c\ \lambda c'.\ \exists c''.(\phi\ c\ c'' \wedge \psi\ c''\ c')$

Operations on Contexts

- Example: Operator for "a" or "some"
  - Let P, Q :: $N \rightarrow [e] \rightarrow [e] \rightarrow t$
    - Call the type of this K
    - Pi applies P to the ith discourse reference
    - Interpret some P are Q
  - $\lambda P\ \lambda Q\ \lambda c\ \ (\exists\ ;\ Pi\ ;\ Qi)\ c$ where $i = |c|$
    - Inserts new referent and asserts P, Q true of it

Expressing Negation

- $\neg\,\phi = \lambda c\ \lambda c'\ (c = c' \wedge \neg\exists c''.\ \phi\ c\ c'')$
  - Notice output context not include anything new
- $\phi \Rightarrow \psi = \lambda c\ \lambda c'\ (c = c' \wedge \forall c_2.\ (\phi\ c\ c_2 \rightarrow \psi\ c_2\ c_3)$
  - Again output not include anything new
- Interpret "all" as
  - $\lambda P\ \lambda Q\ \lambda c\ \ (\exists\ ;\ Pi \Rightarrow Qi)\ c$ where $i = |c|$
  - *or equivalently* $\lambda P\ \lambda Q\ \lambda c\ \ \neg(\exists\ ;\ Pi\ ;\ \neg\ Qi)\ c$ where $i = |c|$

More determiners

- Interpret "no" as
  - $\lambda P\ \lambda Q\ \lambda c\ \ \neg(\exists\ ;\ Pi\ ;\ Qi)\ c$ where $i = |c|$

Use Continuations

- Let ϕ be context transition: $[c] \rightarrow [c] \rightarrow t$
  - Let P be property of output contexts
- Define $G = \lambda\phi\ \lambda c\ \lambda P.\ \exists\ c'.(\phi\ c\ c' \wedge P\ c')$
- Combine continuized contexts by
  - $\lambda\Phi\ \lambda\Psi\ \lambda c\ \lambda P.\ \Phi c\ (\lambda c.\Psi\ c\ P)$

Example

- [[A man slept]]:
  $G(\lambda c\ \lambda c'.\ \exists x.\ (man(x) \wedge slept(x) \wedge c\hat{\ }x = c')$
  $= \lambda c\ \lambda P.\ \exists\ c'.(\exists x.\ man(x) \wedge slept(x)$
  $\wedge\ c\hat{\ }x = c' \wedge P\ c')$
  $= \lambda c\ \lambda P.(\exists x.\ man(x) \wedge slept(x) \wedge P(c\hat{\ }x))$
- [[A woman slept]]:
  $\lambda c\ \lambda P.(\exists x.\ woman(x) \wedge slept(x) \wedge P(c\hat{\ }x))$

Example

- [[A man slept]]:
  $\lambda c\ \lambda P.(\exists x.\ man(x) \wedge slept(x) \wedge P(c\hat{\ }x))$
- [[A woman slept]]:
  $\lambda c\ \lambda P.(\exists x.\ woman(x) \wedge slept(x) \wedge P(c\hat{\ }x))$
- Use combine function:
  - $\lambda c\ \lambda P.\ (\lambda c\ \lambda P.\ (\exists x.\ man(x) \wedge slept(x) \wedge P(c\hat{\ }x))c$
    $\lambda c\ (\lambda c\ \lambda P.\ (\exists x.\ woman(x) \wedge slept(x) \wedge P(c\hat{\ }x)))cP) =$
    $\lambda c\ \lambda P.\ (\lambda P.\ (\exists x.\ man(x) \wedge slept(x) \wedge P(c\hat{\ }x))$
    $\lambda c\ (\exists y.\ woman(y) \wedge slept(y) \wedge P(c\hat{\ }y))) =$
    $\lambda c\ \lambda P.\ (\exists x.\ man(x) \wedge slept(x) \wedge$
    $(\exists y.\ woman(y) \wedge slept(y) \wedge P(c\hat{\ }x\hat{\ }y)))$

Representing in Haskell

- See DRAC.hs
- Need to raise unary and binary relations to context change operations:
  - Let A be unary: $A^\circ = \lambda j\ \lambda c\ \lambda c'(c = c' \wedge Ac[j])$
  - Let B be binary: $B^\circ = \lambda j\ \lambda j'\ \lambda c\ \lambda c'(c = c' \wedge Bc[j]c[j'])$
- Context = [Entity]
- Prop = [Context]
- Trans = Context → Prop

Salience

- Anaphoric reference changes incrementally
- Determining reference of pronoun depends on:
  - morphological and lexical factors
    - gender, number
  - syntactic properties of sentence that contains pronoun
  - information contained in previous discourse
  - background information (common ground) shared by speaker and hearer.

Salience

- Syntactic properties of sentence that contains pronoun
  - Subject more salient than an object
  - Mary kicked the ball to Sarah. She liked to play soccer.
  - Sarah was kicked the ball by Mary. She liked to play soccer.

Question

- What overall strategy should we use:
  - Get accessible discourse referents
  - Filter them according to constraints
  - Rank the remaining ones wrt preferences
- At which stage does pronoun resolution apply?

How to represent salience?

- Consider reordering order of referents after a sentence.
  - Use an index to find actual referent (e.g. in sentence)
  - But search through referents in an order that may be re-ordered by each succeeding sentence.
  - Let (i)c indicate moving item i to front of list
  - Let d:c indicate adding new element d at front of list

Redefine basic operations

- Let $\exists = \lambda c\ \lambda c'. \exists x.(x{:}c = c')$
- Use to redefine determiners
- $[[\text{a girl}]] = \lambda Q.\ \lambda c.\ \lambda c'.\ \exists x.\ (girl(x) \wedge Qi(x{:}c)c')$ where $i = |c|$

Logical Inference

Where we are

- We saw how to construct first-order logical representations for natural language sentences. What can we use them for?
  - Model checking: Check whether a formula is true w.r.t. a model of the world.
  - Model building: Check whether a formula is satisfiable, i.e. whether there is a model in which the formula is true.
  - Inference: Check what follows from a given set of formulas.

Motivation

- Knowing the meaning of a sentence also means knowing what follows from it.
  - Not all robots are intelligent.
⇒ Some robots are not intelligent.
  - Either Turing or Church invented the lambda calculus ⇏ Turing invented the lambda calculus.

Inferences Critical

- Inferences, often using world knowledge, play a big role in understanding utterances.
  - John ate the pudding with a fork.
John ate the pudding with vanilla flavor.
  - A: Would you like to come to the Keith Jarrett concert?
B: I hate Jazz!

Application: Question Answering

- Was Erdös married?
  - Apart from his family and old friends, Paul Erdös had no interest in a relationship which was not founded in shared intellectual curiosity and therefore he remained a bachelor until his death.
- Did United win the Champions League?
  - United failed to progress beyond the group stages of the Champions League and trailed in the Premiership title race, sparking rumours over its future.