Lecture 23: Intensional Logic

CS 181O
Spring 2016
Kim Bruce

Midterm

- Lambda calculus
  - Define semantics using lambda calculus
- Propositional & predicate logic
  - Syntax & Semantics
  - Natural language
- Intensional/Modal logic
- Parsing
- Extend programs

Midterm

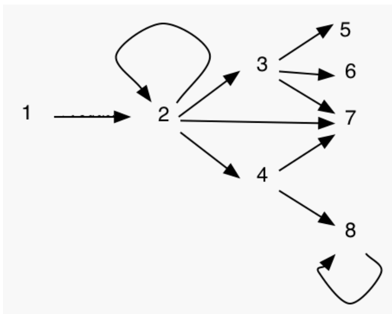
- Pick up sometime Wednesday (10 a.m - 5 p.m.)
- Due 24 hours after pick up

Models for Intensional Logic

- A (Kripke) model M consists of
  - a non-empty set W of contexts,
  - a binary relation R on W, the accessibility relation
  - A valuation function V which assigns a truth value $V_w(p)$ to every proposition letter p in each context w.
- Contexts referred to as possible worlds
- Combination of W,R called a "frame"

Contexts & Accessibility

- Accessibility relation:
  - R = {(v1, v2), (v2, v2), (v2, v3), (v2, v4), (v2, v7), (v3, v5), (v3, v6), (v3, v7), (v4, v7), (v4, v8), (v8, v8)}
  - or



Truth in Intensional Propositional Logic

- Let M be model with W as set of possible worlds, R as accessibility relation, and V as valuation, then $V_{M,w}(\phi)$, the truth value of ϕ in w given M is defined as follows:
- $V_{M,w}(p) = V_w(p)$ for all proposition letters p.
- $V_{M,w}(\neg\phi)$ = true iff $V_w(\phi)$ = false.
- $V_{M,w}(\phi \to \psi)$ = true iff
  $V_{M,w}(\phi)$ = false or $V_{M,w}(\psi)$ = true.
- ...
- $V_{M,w}(O\phi)$ = true iff $\forall w' \in W$ s.t. $\langle w,w' \rangle \in R$,
  $V_{M,w'}(\phi)$ = true.

Types

- Let s be type for set of worlds, with e and t as before. s can only be used as a domain of functions.
- The set of all types, T, is the smallest set such that
  - $e, t \in T$
  - if $a, b \in T$ then so is $a \to b$
  - if $a \in T$, then so is $s \to a$

See model in EAI.hs

- IBool = World → Bool
- IEntity = World → Entity
- iSent :: Sent → IBool
- iNP :: NP → (IEntity → IBool) → IBool
- iVP :: VP → (IEntity → IBool)
- iCN :: CN → (IEntity → IBool)
- iDet :: DET → (IEntity → IBool) →
  (IEntity → IBool) → IBool

Example

- SnowWhite laughed.
- iSent (Sent SnowWhite Laughed)
  ⇒ iNP SnowWhite (iVP Laughed)
  ⇒ (λ p. p iSnowWhite)(λx, w. iLaugh w (x w))
  ⇒ (λ x, w. iLaugh w (x w)) (iSnowWhite)
  ⇒ λw.((iLaugh w) (ISnowWhite w))
- *Determine truth once know which world*

Modeling Intension

- Book's approach is over-simplified!
  - Do not model accessibility relation over worlds!
  - Plenty of room for improvement

Using Intentions

- What is a fake?
  - How can we use possible worlds to make sense of it?

Adjectives

- iAdj :: ADJ → (IEntity → IBool) →
  (IEntity → IBool)
- *p is (IEntity → IBool), x is IEntity, i is World* in

  ```
  iADJ Fake = \ p x i ->
          not (p x i) && any (\ j -> p x j) worlds
  ```
- iAdj Fake Princess = \ x i ->
  not (Princess x i) && any (\ j -> Princess x j) worlds

Evaluating

- Is SnowWhite a fake princess in world 1?
- iRCN (RCN3 Fake Princess) iSnowWhite W1
- = iADJ Fake (iCN Princess) iSnowWhite W1
- = iADJ Fake (\x i -> iPrincess i (x i)) iSnowWhite W1
- = not ((iPrincess W1) (iSnowWhite W1)) &&
  any (\j → (iPrincess j) (iSnowWhite j)

Attitude Verbs

- Wants, Hopes, Believes, ...
- Necessarily - *true in all worlds*
- Possibly - *true in some world*

Intensionalization

- Take an extensional type and convert to corresponding intensional type

	extensional type	*intensional type*
sentence	t	s → t
definite description	e	s → e
noun	e → t	(s → e) → (s → t)
transitive verb	e → (e → t)	(s → e) → ((s → e) → (s → t))

Intensionalization(1)

- The intensional counterpart of an extensional type τ is the type $i_1(\tau)$, where i_1 is a mapping that replaces each occurrence of an atomic type by its intensional counterpart, i.e. replaces type e by type s → e and type t by s → t.
  - Ex: e → (e → t) replaced by (s → e) → ((s → e) → (s → t))

Intensionalization(2)

- The intensional counterpart of an extensional type τ is $s \to i_2(\tau)$, where i_2 is the following mapping:
  - $i_2(e) = e$
    $i_2(t) = t$
    $i_2(\tau \to \tau') = (s \to \tau) \to \tau'$
  - Ex: $e \to (e \to t)$ replaced by $s \to ((s \to e) \to ((s \to e) \to t))$
  - Same as previous if swap argument order

Intensionalization

- Book introduces operations: $^{\cap}$ and $^{\cup}$ to raise and lower meanings
  - translate to intensionalized world

Meanings

- Kripke: Names are rigid identifiers — meaning same in all worlds:
- Reference of other expressions varies
  - E.g., the Nobel prize winner can be different
- Therefore truth can vary in different worlds
  - Jane won the Nobel prize this year.

Time

- Treat times as possible worlds.
  - Captures time-dependent meanings
  - E.g. the president of the United States, the first person to enter class today.

Time-dependent Types

- Intensionalize as before:
  - Let i = domain of time instants

	extensional type	*intensional type*
sentence	t	$i \to t$
definite description	e	$i \to e$
noun	$e \to t$	$(i \to e) \to (i \to t)$
transitive verb	$e \to (e \to t)$	$(i \to e) \to ((i \to e) \to (i \to t))$

Quantifying over time

- Former
  - Example: former president of Pomona
    - $\lambda t \lambda x.\neg((\text{presPomona } t)\ (x\ t)) \wedge$
      $\exists t'.(t' < t) \wedge ((\text{presPomona } t')\ (x\ t'))$
- (Have to add < to our language.)

Semantics

- N ::= ADJ CN
- iN = (iADJ ADJ) (iN CN)
- iADJ Former = $\lambda P \lambda x \lambda t.\ (\neg((P\ t)\ (x\ t)) \wedge$
  $\exists t'.(t' < t) \wedge ((P\ t')\ (x\ t')))$

Past Tense

- Meaning of laughed (past tense):
  - $\lambda x \lambda t.\exists t'.(t' < t) \wedge ((\text{laugh } t')\ (x\ t')) :: ie \to it$
- Alternatively can write operators to be applied to VP:
  - PAST = $\lambda P \lambda x \lambda t.\exists t'.(t'<t) \wedge ((P t')(x t')) :: (ie \to it) \to (ie \to it)$
  - FUTURE = $\lambda P \lambda x \lambda t.\exists t'.(t' > t) \wedge ((P\ t')\ (x\ t')) :: (ie \to it)$
    $\to (ie \to it)$

Raising and Lowering

- Text talks about operations to take from extensional to intensional world and back.

Questions?