1 SPNLP 2008: Lambda Terms, Quantifiers, Satisfaction

Contents

1 Outline **1**

2 Typed Lambda Calculus **1**

3 First Order Logic **2**

4 Truth and Satisfaction **3**

2 Typed Lambda Calculus

Transitive Verbs as Functions

We looked at replacing n-ary relations with functions. How does this work with transitive verbs?

- Version 1: chase of type $<$IND, IND $>\rightarrow$ BOOL
- Version 2: chase of type IND $\rightarrow$ (IND $\rightarrow$ BOOL)

Advantages of Version 2 (called a *curryed function*):

- Makes the syntax more uniform.
- Fits better with compositional semantics (discussed later)

Lambda

Lambdas talk about missing information, and where it is.

- The λ binds a variable.
- The positions of a λ-bound variable in the formula mark where information is 'missing'.
- Replacing these variables with values fills in the missing information.

Example:

- $\lambda x.(\mathsf{man}\, x)$ — λ-abstract
- $(\lambda x.(\mathsf{man}\, x)\, \mathsf{john})$ — application
- $(\mathsf{man}\, \mathsf{john})$ — β-reduction/function application.

NB. I'm using `ind` to replace the `term` I used in the last lecture.

Types

- IND and BOOL are *basic types*.
- If σ, τ are types, then so is $(\sigma \to \tau)$. Brackets are omitted if no ambiguity.
- For types τ, we have variables **Var**(τ), constants **Con**(τ).
- Since we are doing first order logic, we will later restrict variables to **Var**(IND), but allow constants of any type.

Terms in Typed Lambda Calculus

We define terms **Term**(τ) of type τ:

- **Var**(τ) $\subseteq$ **Term**(τ).
- **Con**(τ) $\subseteq$ **Term**(τ).
- If $\alpha \in \mathbf{Term}(\sigma \to \tau)$ and $\beta \in \mathbf{Term}(\sigma)$ then $(\alpha\,\beta) \in \mathbf{Term}(\tau)$ (function application).
- If $x \in \mathbf{Var}(\sigma)$ and $\alpha \in \mathbf{Term}(\rho)$, then $\lambda x.\alpha \in \mathbf{Term}(\tau)$, where $\tau = (\sigma \to \rho)$

3 First Order Logic

Extending to a First Order Language

1. Variables i.e., **Var**(IND): $x, y, z, \ldots, x_0, x_1, x_2, \ldots$
2. Boolean connectives:

$\neg$	BOOL $\to$ BOOL	(negation)
$\wedge$	BOOL $\to$ (BOOL $\to$ BOOL)	(and)
$\vee$	BOOL $\to$ (BOOL $\to$ BOOL)	(or)
$\to$	BOOL $\to$ (BOOL $\to$ BOOL)	(if…then)

3. Quantifiers: $\forall$ (all)
   $\exists$ (some)
4. Equality: $=$ $\quad \tau \to (\tau \to$ BOOL)
5. Punctuation: brackets and period

Quantifier Syntax

- If $\phi \in$ **Term**(BOOL), and $x \in$ **Var**(IND), then $\forall x.\phi$ and $\exists x.\phi \in$ **Term**(BOOL).
- $x \in$ **Var**(IND) is called an *individual variable*.

Syntactic conventions:

- Instead of writing $((= \alpha)\beta)$, $((\wedge\phi)\psi)$, etc., we write $(= \alpha = \beta)$, $(\phi \wedge \psi)$, etc.
- Instead of writing e.g., ((chase fido) john), we sometimes write (chase fido john).
- **NB** this is equivalent to *chase(john, fido)* on a relational approach.

Some Examples

1. $\exists x.(\mathsf{love}\, x\, \mathsf{kim})$ *Kim loves someone*
2. $(\neg\exists x.(\mathsf{love}\, x\, \mathsf{kim}))$ *Kim doesn't love anyone*
3. $\forall x.((\mathsf{robber}\, x) \rightarrow \exists y.((\mathsf{customer}\, y) \wedge (\mathsf{love}\, y\, x)))$ *All robbers love a (perhaps different) customer*
4. $\exists y.((\mathsf{customer}\, y) \wedge \forall x.((\mathsf{robber}\, x) \rightarrow (\mathsf{love}\, y\, x)))$ *All robbers love the same customer*

Free and Bound Variables

$$((\mathsf{customer}\, x) \vee \forall x.((\mathsf{robber}\, x) \rightarrow \exists y.(\mathsf{person}\, y)))$$

- First occurrence of x is *free*;
- Second occurrence of x is *bound*; Occurrence of y is *bound*.
- Free variable $\approx$ pronouns.
  - *She loves Fido*
- Context needed to interpret *she*; Something in addition to models so far needed to interpret free variables.

A WFF (**Term**(BOOL)) with no free variables is a *(closed) sentence*. FOL sentences $\subset$ WFFs.

Formulas with free variables are sometimes called *open*.

4 Truth and Satisfaction

Interpreting FOL Sentences

Task:

- Compute whether a sentence is *true* or *false* with respect to a model.
  - Is the sentence an accurate description of the situation?

Strategy: *Compositionality!* Use recursion, but:

- Subformula of $\forall x.(\mathsf{robber}\, x)$ is $(\mathsf{robber}\, x)$ and this is not a sentence! So…

Satisfaction: $[\![\phi]\!]^{M,g} = 1$

Model M and *variable assignment* g satisfy the WFF ϕ.

- g defined for all individual variables, i.e., $x \in \mathbf{Var}(\text{IND})$;
- $g(x) \in D$.
- If α is an atomic term ($\in \mathbf{Con}(\tau) \cup \mathbf{Var}(\tau)$), then

$$i^g_V(\alpha) = \begin{cases} g(\alpha) & \text{if } \alpha \text{ is a variable} \\ V(\alpha) & \text{if } \alpha \text{ is a constant} \end{cases}$$

- $[\![\exists x.\phi]\!]^{M,g} = 1$ iff $[\![\phi]\!]^{M,g[u/x]} = 1$ for some $u \in D$
- $g[u/x](x) = u$
- 'Try to find some value u for x that makes ϕ true'

Value of a term in a model

Where $M = \langle D, V \rangle$:

$[\![\alpha]\!]^{M,g} = i^g_V(\alpha)$	if	α is atomic
$[\![(\alpha\,\beta)]\!]^{M,g}$	$=$	$[\![\alpha]\!]^{M,g}([\![\beta]\!]^{M,g})$
$[\![(\lambda x.\alpha)]\!]^{M,g}$	$=$	that function h such that for any u $\in D, h(u) = [\![\alpha]\!]^{M,g[u/x]}$
$[\![\alpha_1 = \alpha_2]\!]^{M,g} = 1$	iff	$[\![\alpha_1]\!]^{M,g} = [\![\alpha_2]\!]^{M,g}$
$[\![\neg\phi]\!]^{M,g} = 1$	iff	$[\![\phi]\!]^{M,g} = 0$
$[\![\phi \wedge \psi]\!]^{M,g} = 1$	iff	$[\![\phi]\!]^{M,g} = 1$ and $[\![\psi]\!]^{M,g} = 1$
$[\![\phi \vee \psi]\!]^{M,g}$	iff	$[\![\phi]\!]^{M,g} = 1$ or $[\![\psi]\!]^{M,g} = 1$
$[\![\phi \rightarrow \psi]\!]^{M,g} = 1$	iff	$[\![\phi]\!]^{M,g} = 0$ or $[\![\psi]\!]^{M,g} = 1$
$[\![\exists x.\phi]\!]^{M,g} = 1$	iff	$[\![\phi]\!]^{M,g[u/x]} = 1$ for some $u \in D$
$[\![\forall x.\phi]\!]^{M,g} = 1$	iff	$[\![\phi]\!]^{M,g[u/x]} = 1$ for every $u \in D$.

Truth (in terms of Satisfaction)

If $\phi \in$ **Term**(BOOL), we often write $M, g \models \phi$ instead of $[\![\phi]\!]^{M,g'} = 1$.

It doesn't matter which g you use for sentences, so:

Truth: A *sentence* ϕ is true in a model M (written $M \models \phi$) iff for *any* g, $M, g \models \phi$

Validity: A sentence ϕ is valid (written $\models \phi$) iff for any M, $M \models \phi$

Entailment: $\phi_1, \ldots, \phi_n \models \psi$ iff if $M, g \models \phi_i$ for all i, $1 \leq i \leq n$, then $M, g \models \psi$