Computational Logic

Recall of First-Order Logic

Damiano Zanardini

UPM European Master in Computational Logic (EMCL)
School of Computer Science
Technical University of Madrid
damiano@fi.upm.es

Academic Year 2008/2009

Semantics of a first-order language

Interpretations

An *interpretation* $\mathcal{I}$ is a pair (D, I), where $D \neq \emptyset$ is a set (the *domain* of the universe) and I maps symbols to individuals or functions

- constants: $I(a) = d \in D$
- variables: $I(x) = d \in D$
- functions: $I(f/n) = \mathcal{F} : D^n \mapsto D$
  - $I(f(t_1, ..., t_n)) = \mathcal{F}(I(t_1), ..., I(t_n)) = \mathcal{F}(d_1, ..., d_n) \in D$
- predicates: $I(p/n) = \mathcal{P} : D^n \mapsto \{\mathbf{t}, \mathbf{f}\}$
  - $I(p(t_1, ..., t_n)) = \mathcal{P}(I(t_1), ..., I(t_n)) = \mathcal{P}(d_1, ..., d_n) \in \{\mathbf{t}, \mathbf{f}\}$

- an interpretation assigns an element of D to any term, and a *truth value* to any predicate applied to terms
- $\mathcal{P}$ is an n-ary *relation* $\mathcal{R}$: $\mathcal{P}(d_1, ..., d_n) = \mathbf{t}$ iff $\langle d_1, ..., d_n \rangle \in \mathcal{R}$

Semantics of a first-order language

Evaluation of a formula

Assigning a truth value to a formula, according to:

- The chosen interpretation of constants, functions and predicates
- The rules for evaluation (see also *truth tables*)

$$
\begin{array}{lll}
I(\neg F) = \mathbf{t} & \text{iff} & I(F) = \mathbf{f} \\
I(F \wedge G) = \mathbf{t} & \text{iff} & I(F) = I(G) = \mathbf{t} \\
I(F \vee G) = \mathbf{f} & \text{iff} & I(F) = I(G) = \mathbf{f} \\
I(F \rightarrow G) = \mathbf{f} & \text{iff} & I(F) = \mathbf{t} \text{ and } I(G) = \mathbf{f} \\
I(F \leftrightarrow G) = \mathbf{t} & \text{iff} & I(F) = I(G) \\
I(\forall x F(x)) = \mathbf{t} & \text{iff} & I(F(x/c)) = \mathbf{t}\ ^{*} \text{ for every constant } c \\
I(\exists x F(x)) = \mathbf{t} & \text{iff} & I(F(x/c)) = \mathbf{t}\ ^{*} \text{ for at least one constant } c
\end{array}
$$

* it is required that every element d of D is denoted by at least one constant

I (instead of $\mathcal{I}$) will often denote an interpretation when D is clear

Example (propositional): $(p \rightarrow q) \wedge (q \rightarrow r) \rightarrow r$

- first interpretation: $I'(p) = \mathbf{f}$, $I'(q) = \mathbf{f}$, $I'(r) = \mathbf{t}$

$$
\begin{array}{ccccccccc}
(\mathbf{f} & \rightarrow & \mathbf{f}) & \wedge & (\mathbf{f} & \rightarrow & \mathbf{t}) & \rightarrow & \mathbf{t} \\
 & \mathbf{t} & & \wedge & & \mathbf{t} & & \rightarrow & \mathbf{t} \\
 & & & \mathbf{t} & & & & \rightarrow & \mathbf{t} \\
 & & & & & \mathbf{t} & & &
\end{array}
$$

- second interpretation: $I''(p) = \mathbf{f}$, $I''(q) = \mathbf{f}$, $I''(r) = \mathbf{f}$

$$
\begin{array}{ccccccccc}
(\mathbf{f} & \rightarrow & \mathbf{f}) & \wedge & (\mathbf{f} & \rightarrow & \mathbf{f}) & \rightarrow & \mathbf{f} \\
 & \mathbf{t} & & \wedge & & \mathbf{t} & & \rightarrow & \mathbf{f} \\
 & & & \mathbf{t} & & & & \rightarrow & \mathbf{f} \\
 & & & & & \mathbf{f} & & &
\end{array}
$$

Example (propositional): $(p \rightarrow q) \wedge (q \rightarrow r) \rightarrow r$

- this example only needs truth tables, for all possible interpretations

p	q	r	$p \rightarrow q$	$q \rightarrow r$	$(p \rightarrow q) \wedge (q \rightarrow r)$	$(p \rightarrow q) \wedge (q \rightarrow r) \rightarrow r$
t	**t**	**t**	**t**	**t**	**t**	**t**
t	**t**	**f**	**t**	**f**	**f**	**t**
t	**f**	**t**	**f**	**t**	**f**	**t**
t	**f**	**f**	**f**	**t**	**f**	**t**
f	**t**	**t**	**t**	**t**	**t**	**t**
f	**t**	**f**	**t**	**f**	**f**	**t**
f	**f**	**t**	**t**	**t**	**t**	**t**
f	**f**	**f**	**t**	**t**	**t**	**f**

Semantics of a first-order language

Example (first-order): $\forall x(m(a,x) \wedge p(x)) \rightarrow \forall y q(s(y))$

- first interpretation: $D = \{0, 1, 2, 3, ..\}$
  - $I(a) = 0$
  - $I(s(x)) = \mathcal{S}(I(x)) =$ the successor of $I(x)$
  - $p(x)$ means that x is even
  - $q(x)$ means that x is odd
  - $m(x, y)$ means that $x < y$
- $\mathcal{I}$ evaluates the formula to **t** (try it!)

Semantics of a first-order language

Example (first-order): $\forall x(m(a,x) \wedge p(x)) \rightarrow \forall y q(s(y))$

- second interpretation: $D = \{$  ,  ,  $\}$, $I(a) =$ 

x	$s(x)$
	
	
	

x	$p(x)$
	**t**
	**t**
	**t**

x	$q(x)$
	**t**
	**f**
	**t**

m			
	**t**	**t**	**t**
	**f**	**f**	**t**
	**t**	**f**	**f**

- and this evaluates to **f** (**t** $\rightarrow$ **f** $=$ **f**)

Semantics of a first-order language

Satisfiable formulæ

An interpretation $\mathcal{I} = (D, I)$ *satisfies* a formula F on D iff $I(F) = \mathbf{t}$ (also written $\mathcal{I}(F) = \mathbf{t}$). In this case, $\mathcal{I}$ is a *model* of F

- F is *satisfiable* (written $SAT(F)$) iff it has at least one model
- F is *unsatisfiable* (written $UNSAT(F)$) iff it has no models
  - that is, all interpretations are *countermodels*
- F is *valid* (written $VAL(F)$) iff every interpretation is a model
  - this is denoted by $\models F$, and amounts to say $UNSAT(\neg F)$

With a set of formulæ $\{F_1, .., F_n\}$:

- (D, I) satisfies $\{F_1, .., F_n\}$ iff $I(F_i) = \mathbf{t}$ on D for every i
- $\{F_1, .., F_n\}$ is satisfiable iff there is such an interpretation

Example: $\forall x(m(a, x) \wedge p(x)) \rightarrow \forall y q(s(y))$

This formula is satisfiable but not valid

Logical consequence

Logical consequence

Given a set of formulæ $\Gamma = \{F_1, .., F_n\}$ and a formula G over the same language, G is a *logical consequence* of Γ (written $\Gamma \models G$) iff every interpretation satisfying Γ also satisfies G, or, equivalently, there is no interpretation which satisfies Γ but not G

Important (in some sense, it is a matter of convenience)

$$\{F_1, .., F_n\} \models G \qquad \text{iff} \qquad \models (F_1 \wedge .. \wedge F_n) \rightarrow G$$

To decide $\Gamma \models G$ can be very hard

We have to take all models of Γ and verify that they all satisfy G, or find a counterexample

Logical consequence

Example: $\{p \rightarrow (q \rightarrow r),\ p \wedge q\} \models r$

Equivalent to $\models ((p \rightarrow (q \rightarrow r)) \wedge (p \wedge q)) \rightarrow r$

p	q	r	$p \rightarrow (q \rightarrow r)$	$p \wedge q$	$(p \rightarrow (q \rightarrow r)) \wedge (p \wedge q)$	G
t	**t**	**t**	**t**	**t**	**t**	**t**
t	**t**	**f**	**f**	**t**	**f**	**t**
t	**f**	**t**	**t**	**f**	**f**	**t**
t	**f**	**f**	**t**	**f**	**f**	**t**
f	**t**	**t**	**t**	**f**	**f**	**t**
f	**t**	**f**	**t**	**f**	**f**	**t**
f	**f**	**t**	**t**	**f**	**f**	**t**
f	**f**	**f**	**t**	**f**	**f**	**t**

Logical consequence

Example: $\{p \rightarrow (q \rightarrow r),\ p \wedge q\} \models \neg r$

Equivalent to $\models ((p \rightarrow (q \rightarrow r)) \wedge (p \wedge q)) \rightarrow \neg r$

p	q	r	$p \rightarrow (q \rightarrow r)$	$p \wedge q$	$(p \rightarrow (q \rightarrow r)) \wedge (p \wedge q)$	G
t	**t**	**t**	**t**	**t**	**t**	**f**
t	**t**	**f**	**f**	**t**	**f**	**t**
t	**f**	**t**	**t**	**f**	**f**	**t**
t	**f**	**f**	**t**	**f**	**f**	**t**
f	**t**	**t**	**t**	**f**	**f**	**t**
f	**t**	**f**	**t**	**f**	**f**	**t**
f	**f**	**t**	**t**	**f**	**f**	**t**
f	**f**	**f**	**t**	**f**	**f**	**t**

Logical consequence

Example: $\{\exists x p(x),\ \exists x q(x)\} \models \exists x(p(x) \wedge q(x))$

- $D = \{1, 2, 3, 4, ..\}$
- $p(x)$: x is even
- $q(x)$: x is odd

It is easy to see that this interpretation makes both premises true (indeed, there exist even numbers and there exist odd numbers), but does not satisfy the conclusion (no numbers are both even and odd)

$$I(\exists x p(x)) = \mathbf{t} \quad I(\exists x q(x)) = \mathbf{t} \quad I(\exists x(p(x) \wedge q(x))) = \mathbf{f}$$

Therefore, the deduction is incorrect

Example: $\{\exists x p(x),\ \exists x q(x)\} \models \exists x(p(x) \vee q(x))$

This is, of course, a valid deduction

Syntax vs. semantics

Formal systems

A *proof formal system* consists of:

- a formal language (alphabet and rules for building formulæ)
- a set of *logical axioms* (i.e., valid formulæ, which do not require proof)
- a set of *inference rules* for proving new formulæ
- a definition of proof

Theories

A *theory* T is a formal system extended with a set Γ of *non-logical axioms* (i.e., formulæ taken for granted) $\quad \rightsquigarrow \quad T[\Gamma]$

- if $\Gamma = \emptyset$, then T is the *basic theory* of the formal system

Syntax vs. semantics

Proofs

A *proof* of a formula G in a theory $T[\Gamma]$ (written $T[\Gamma] \vdash G$) is a *finite* sequence of formulæ such that each formula of the sequence is either

- a logical or non-logical axiom of the theory; or
- the result of applying an inference rule to previous formulæ in the sequence

and G is the last formula in the sequence

Theorems

A *theorem* is a formula for which there is at least one proof

Arguments

An argument with premises $A_1, .., A_n$ and conclusion B is *logically correct* in a formal system if $T[A_1, .., A_n] \vdash B$

Syntax vs. semantics

Proof example: $T[p \rightarrow (q \rightarrow r),\ p \wedge q] \vdash r$

❶	$p \rightarrow (q \rightarrow r)$	first premise
❷	$\neg p \vee (\neg q \vee r)$	interdefinition of $\rightarrow$, $\neg$ and $\vee$ on ❶
❸	$(\neg p \vee \neg q) \vee r$	associativity on ❷
❹	$\neg(p \wedge q) \vee r$	De Morgan on ❸
❺	$p \wedge q \rightarrow r$	interdefinition of $\rightarrow$, $\neg$ and $\vee$ on ❹
❻	$p \wedge q$	second premise
❼	r	modus ponens on ❺, ❻

Another approach to prove validity

Instead of looking at all the possible models of a formula, we exploited our knowledge of logical rules

- we also say that $((p \rightarrow (q \rightarrow r)) \wedge (p \wedge q)) \rightarrow r$ is a *tautology*

Syntax vs. semantics

Theorem (Deduction)

$T[F_1, .., F_n] \vdash G$ *iff* $T \vdash (F_1 \wedge .. \wedge F_n) \rightarrow G$

- *question: why do we use both forms (premises and implication)?*

Theorem (Validity)

Every theorem of T is logically valid: if $T \vdash G$ then $\models G$

- *this happens if the formal system is consistent!* $\rightsquigarrow$ *Gödel*

Theorem (Completeness)

In a first-order theory T, all valid formulæ are theorems of T: if $\models G$ then $T \vdash G$

- *the rest of the course will be basically about finding such theorems*

Syntax vs. semantics

Theorem (Deduction, equivalent form)

$T[F_1, .., F_n] \vdash G \quad$ *iff* $\quad VAL((F_1 \wedge .. \wedge F_n) \rightarrow G)$

- *question: why do we use both forms (premises and implication)?*

Theorem (Validity)

Every theorem of T is logically valid: if $T \vdash G$ then $\models G$

- *this happens if the formal system is consistent!* $\rightsquigarrow$ *Gödel*

Theorem (Completeness)

In a first-order theory T, all valid formulæ are theorems of T: if $\models G$ then $T \vdash G$

- *the rest of the course will be basically about finding such theorems*