Foundations of Artificial Intelligence

9. Predicate Logic

Syntax and Semantics, Reduction to Propositional Logic

Joschka Boedecker and Wolfram Burgard and
Frank Hutter and Bernhard Nebel and Michael Tangermann

Albert-Ludwigs-Universität Freiburg

June 5, 2019

Motivation

We can already do a lot with propositional logic. It is, however, annoying that there is no structure in the atomic propositions.

Example:

"All blocks are red"
"There is a block A"
It should follow that "A is red"

But propositional logic cannot handle this.

Idea: We introduce individual variables, predicates, functions, ...

$\rightarrow$ First-Order Predicate Logic (PL1)

Contents

Lecture Overview

The Alphabet of First-Order Predicate Logic

Symbols:

- Operators: $\neg$, $\vee$, $\wedge$, $\forall$, $\exists$, $=$
- Variables: x, x_1, x_2, $\ldots$, x', x'', $\ldots$, y, $\ldots$, z, $\ldots$
- Brackets: $()$, $[]$, $\big(\big)$, $\big[\big]$
- Function symbols (e.g., $weight()$, $color()$)
- Predicate symbols (e.g., $Block()$, $Red()$)
- Predicate and function symbols have an arity (number of arguments).
  0-ary predicate = propositional logic atoms: $P, Q, R, \ldots$
  0-ary function = constants: $a, b, c, \ldots$
- We assume a countable set of predicates and functions of any arity.
- "$=$" is usually not considered a predicate, but a logical symbol

The Grammar of First-Order Predicate Logic (1)

Terms (represent objects):

1. Every variable is a term.
2. If $t_1, t_2, \ldots, t_n$ are terms and f is an n-ary function, then

   $$f(t_1, t_2, \ldots, t_n)$$

   is also a term.

   Terms without variables: ground terms.

Atomic Formulae (represent statements about objects)

1. If $t_1, t_2, \ldots, t_n$ are terms and P is an n-ary predicate, then $P(t_1, t_2, \ldots, t_n)$ is an atomic formula.
2. If t_1 and t_2 are terms, then $t_1 = t_2$ is an atomic formula.

   Atomic formulae without variables: ground atoms.

The Grammar of First-Order Predicate Logic (2)

Formulae:

1. Every atomic formula is a formula.
2. If φ and ψ are formulae and x is a variable, then

   $$\neg\varphi,\ \varphi \land \psi,\ \varphi \lor \psi,\ \varphi \Rightarrow \psi,\ \varphi \Leftrightarrow \psi,\ \exists x\varphi \text{ and } \forall x\varphi$$

   are also formulae.
   $\forall$, $\exists$ are as strongly binding as $\neg$.

Propositional logic is part of the PL1 language:

1. Atomic formulae: only 0-ary predicates
2. Neither variables nor quantifiers.

Alternative Notation

Here	Elsewhere
$\neg\varphi$	$\sim\varphi \quad \overline{\varphi}$
$\varphi \land \psi$	$\varphi \& \psi \quad \varphi \bullet \psi \quad \varphi, \psi$
$\varphi \lor \psi$	$\varphi \vert \psi \quad \varphi ; \psi \quad \varphi + \psi$
$\varphi \Rightarrow \psi$	$\varphi \rightarrow \psi \quad \varphi \supset \psi$
$\varphi \Leftrightarrow \psi$	$\varphi \leftrightarrow \psi \quad \varphi \equiv \psi$
$\forall x \varphi$	$(\forall x)\varphi \land x\varphi$
$\exists x \varphi$	$(\exists x)\varphi \lor x\varphi$

Meaning of PL1-Formulae

Our example: $\forall x[Block(x) \Rightarrow Red(x)]$, $Block(a)$

For all objects x: If x is a block, then x is red and a is a block.

Generally:

- Terms are interpreted as objects.
- Universally-quantified variables denote all objects in the universe.
- Existentially-quantified variables represent one of the objects in the universe (made true by the quantified expression).
- Predicates represent subsets of the universe.

Similar to propositional logic, we define interpretations, satisfiability, models, validity, . . .

Semantics of PL1-Logic

Interpretation: $I = \langle D, \bullet^I \rangle$ where D is an arbitrary, non-empty set and $\bullet^I$ is a function that

- maps n-ary function symbols to functions over D:
  $f^I \in [D^n \mapsto D]$
- maps individual constants to elements of D:
  $a^I \in D$
- maps n-ary predicate symbols to relations over D:
  $P^I \subseteq D^n$

Interpretation of ground terms:

$$(f(t_1, \ldots, t_n))^I = f^I(t_1^I, \ldots, t_n^I)$$

Satisfaction of ground atoms $P(t_1, \ldots, t_n)$:

$$I \models P(t_1, \ldots, t_n) \text{ iff } \langle t_1^I, \ldots, t_n^I \rangle \in P^I$$

Example (1)

$$
\begin{aligned}
D &= \{d_1, \ldots, d_n \mid n > 1\} \\
a^I &= d_1 \\
b^I &= d_2 \\
c^I &= \ldots \\
Block^I &= \{d_1\} \\
Red^I &= D \\
I &\models Red(b) \\
I &\not\models Block(b)
\end{aligned}
$$

Example (2)

$$
\begin{aligned}
D &= \{1, 2, 3, \ldots\} \\
1^I &= 1 \\
2^I &= 2 \\
&\ldots \\
Even^I &= \{2, 4, 6, \ldots\} \\
succ^I &= \{(1 \mapsto 2), (2 \mapsto 3), \ldots\} \\
I &\models Even(2) \\
I &\not\models Even(succ(2))
\end{aligned}
$$

Semantics of PL1: Variable Assignment

Set of all variables V. Function $\alpha : V \mapsto D$

Notation: $\alpha[x/d]$ is the same as α apart from point x.

For $x : \alpha[x/d](x) = d$.

Interpretation of terms under I, α:

$$x^{I,\alpha} = \alpha(x)$$
$$a^{I,\alpha} = a^I$$
$$(f(t_1, \ldots, t_n))^{I,\alpha} = f^I(t_1^{I,\alpha}, \ldots, t_n^{I,\alpha})$$

Satisfaction of atomic formulae:

$$I, \alpha \models P(t_1, \ldots, t_n) \text{ iff } \langle t_1^{I,\alpha}, \ldots, t_n^{I,\alpha} \rangle \in P^I$$

Example

$$Block^I = \{d_1\}$$
$$Red^I = D$$

$$\alpha = \{(x \mapsto d_1), (y \mapsto d_2)\}$$
$$I, \alpha \models Red(x)$$
$$I, \alpha[y/d_1] \models Block(y)$$

Semantics of PL1: Satisfiability

A formula φ is satisfied by an interpretation I and a variable assignment α, i.e., $I, \alpha \models \varphi$:

$$I, \alpha \models \top$$
$$I, \alpha \not\models \bot$$
$$I, \alpha \models \neg\varphi \text{ iff } I, \alpha \not\models \varphi$$
$$\ldots$$

and all other propositional rules as well as

$I, \alpha \models P(t_1, \ldots, t_n)$	iff	$\langle t_1^{I,\alpha}, \ldots, t_n^{I,\alpha} \rangle \in P^I$
$I, \alpha \models \forall x \varphi$	iff	for all $d \in D$, $I, \alpha[x/d] \models \varphi$
$I, \alpha \models \exists x \varphi$	iff	there exists a $d \in D$ with $I, \alpha[x/d] \models \varphi$

Example

$$
\begin{aligned}
D &= \{d_1, \ldots, d_n \mid n > 1\} \\
a^I &= d_1 \\
b^I &= d_2 \\
Block^I &= \{d_1\} \\
Red^I &= D \\
\alpha &= \{(x \mapsto d_1), (y \mapsto d_2)\}
\end{aligned}
$$

Questions:

1. $I, \alpha \models Block(b) \vee \neg Block(b)$?
2. $I, \alpha \models Block(x) \Rightarrow (Block(x) \vee \neg Block(y))$?
3. $I, \alpha \models Block(a) \wedge Block(b)$?
4. $I, \alpha \models \forall x(Block(x) \Rightarrow Red(x))$?

Free and Bound Variables

$$\forall x \big[R(\boxed{y}, \boxed{z}) \wedge \exists y \big((\neg P(y,x) \vee R(y, \boxed{z}) \big) \big]$$

The boxed appearances of y and z are free. All other appearances of x,y,z are bound.

Formulae with no free variables are called closed formulae or sentences. We form theories from closed formulae.

Note: With closed formulae, the concepts *logical equivalence, satisfiability, and implication, etc.* are not dependent on the variable assignment α (i.e., we can always ignore all variable assignments).

With closed formulae, α can be left out on the left side of the model relationship symbol:

$$I \models \varphi$$

Terminology

An interpretation I is called a model of φ under α if

$$I, \alpha \models \varphi$$

A PL1 formula φ can, as in propositional logic, be satisfiable, unsatisfiable, falsifiable, or valid.

Analogously, two formulae are logically equivalent ($\varphi \equiv \psi$) if for all I, α:

$$I, \alpha \models \varphi \text{ iff } I, \alpha \models \psi$$

Note: $P(x) \not\equiv P(y)$!

Logical Implication is also analogous to propositional logic.

Question: How can we define derivation?

Lecture Overview

Derivation in PL1: Possible Approaches

- We now know the semantics of PL1. How can we do inference in PL1?
- One way: Normalization + Skolemization + Resolution with Unification
- Alternative: Reduction to propostional logic by instantiation based on the so-called Herbrand Universe (all possible terms) $\leadsto$ infinite propositional theories
- It turns out that logical implication in PL1 is undecidable!
- Simple way for special case: If the number of objects is finite, instantiate all variables by possible objects (in fact, often used in AI systems, e.g. planning or ASP)

Finite Universes

- Let us assume that we only want to talk about a finite number of objects.
- Domain closure axiom (DCA):

$$\forall x[x = c_1 \vee x = c_2 \vee \ldots \vee x = c_n]$$

- Often one also assumes that different names denote different objects (unique name assumption/axiom or UNA):

$$\bigwedge_{i \neq j}[c_i \neq c_j]$$

→ Only important when counting or using $\neq$ or $=$ as a predicate.

- Elimate quantification by instantiating all variables with all possible values.

Instantiation

- Notation: if φ is a formula, then $\varphi[x/a]$ is the formula with all free occurences of x replaced by a.
- Universally quantified formulas are replaced by a conjunction of formulas with the variable instantiated to all possible values (from DCA):
  $$\forall x \varphi \rightsquigarrow \bigwedge_i \varphi[x/c_i]$$
- Existentially quantified variables are replaced by a disjunction of formulas with the variable instantiated to all possible values (from DCA):
  $$\exists x \varphi \rightsquigarrow \bigvee_i \varphi[x/c_i]$$
- Note: does blow up the formulas exponentially in the arity of the predicates!

Example

$$
\begin{array}{ll}
\forall x & (Block(x) \Rightarrow Red(x)) \\
\forall x & (x = a \vee x = b \vee x = c) \\
\leadsto & \\
 & (Block(a) \Rightarrow Red(a)) \wedge \\
 & (Block(b) \Rightarrow Red(b)) \wedge \\
 & (Block(c) \Rightarrow Red(c))
\end{array}
$$

Lecture Overview

Summary

- PL1 makes it possible to structure statements, thereby giving us considerably more expressive power than propositional logic.
- Logical implication in PL1 is undecidable.
- If we only reason over a finite universe, PL1 can be reduced to propositional logic over finite theories (but the reduction is exponential in the arity of the predicates).