First Order Predicate Logic

Semantics

Wolfgang Schreiner and Wolfgang Windsteiger
Wolfgang.(Schreiner|Windsteiger)@risc.jku.at

Research Institute for Symbolic Computation (RISC)
Johannes Kepler University (JKU), Linz, Austria
http://www.risc.jku.at

Semantics

The meaning of a predicate logic formula depends on the following entities.

- A non-empty domain D
  - The universe about which the formula talks.

    $D = \mathbb{N}$.
- An interpretation I of all function and predicate symbols
  - Constants: For every constant c, $I(c)$ denotes an element of D, i.e., $I(c) \in D$.
  - Functions: For every function symbol f with arity n, $I(f)$ denotes an n-ary function on D, i.e., $I(f) : D^n \to D$.
  - Predicates: For every predicate symbol p with arity n, $I(p)$ denotes an n-ary predicate (relation) on D, i.e., $I(p) \subseteq D^n$.

$$I = [0 \mapsto \mathit{zero}, + \mapsto \mathit{add}, < \mapsto \mathit{less\text{-}than}, \ldots]$$

- An assignment $a : \mathit{Var} \to D$
  - A function that maps every (free) variable x to a value $a(x)$ in D.

    $a = [x \mapsto 1, y \mapsto 0, z \mapsto 3, \ldots]$

The pair $M = (D, I)$ is also called a *structure*.

The Informal Semantics of Terms and Formulas

- The meaning of a term is an object in the domain.
  - Meaning of a variable is given by the assignment.
  - Meaning of a constant is given by the interpretation.
  - Meaning of $f(t_1, \ldots, t_n)$ is determined by applying the interpretation of f to the meaning of the t_i.
- The meaning of a formula is true or false.
  - Meaning of $\top$ is true, meaning of $\bot$ is false.
  - Meaning of $p(t_1, \ldots, t_n)$ is determined by applying the interpretation of p to the meaning of the t_i.
    - Special case with fixed interpretation: meaning of $t_1 = t_2$ is true, iff the meanings of t_1 and t_2 are identical.
  - Meaning of logical connectives is determined by applying the truth tables to the meaning of the constituent subformulas.
  - Meaning of $\forall x : F$ is true iff the meaning of F is true for all possible assignments for the free variable x.
  - Meaning of $\exists x : F$ is true iff the meaning of F is true for at least one assignment for the free variable x.

The Formal Semantics of Terms

$$D, I, a \longrightarrow \boxed{[\![\, t \,]\!]} \longrightarrow d \in D$$

- Term semantics $[\![\, t \,]\!]^{D,I}_{a} \in D$
  - Given D, I, a, the semantics of term t is a value in D.
  - This value is defined by structural induction on t.

    $$t \ ::= \ x \mid c \mid f(t_1, \ldots, t_n)$$
- $[\![\, x \,]\!]^{D,I}_{a} := a(x)$
  - The semantics of a variable is the value given by the assignment.
- $[\![\, c \,]\!]^{D,I}_{a} := I(c)$
  - The semantics of a constant is the value given by the interpretation.
- $[\![\, f(t_1, \ldots, t_n) \,]\!]^{D,I}_{a} := I(f)([\![\, t_1 \,]\!]^{D,I}_{a}, \ldots, [\![\, t_n \,]\!]^{D,I}_{a})$
  - The semantics of a function application is the result of the interpretation of the function symbol applied to the values of the argument terms.

The recursive definition of a function evaluating a term.

Example

$$
\begin{aligned}
&D = \mathbb{N} = \{zero, one, two, three, \ldots\}\\
&a = [x \mapsto one, y \mapsto two, \ldots]\\
&I = [0 \mapsto zero, + \mapsto add, \ldots]
\end{aligned}
$$

$$
\begin{aligned}
[\![x+(y+0)]\!]_a^{D,I} &= add([\![x]\!]_a^{D,I}, [\![y+0]\!]_a^{D,I})\\
&= add(a(x), [\![y+0]\!]_a^{D,I})\\
&= add(one, [\![y+0]\!]_a^{D,I})\\
&= add(one, add([\![y]\!]_a^{D,I}, [\![0]\!]_a^{D,I}))\\
&= add(one, add(a(y), I(0)))\\
&= add(one, add(two, zero))\\
&= add(one, two)\\
&= three
\end{aligned}
$$

The meaning of the term with the "usual" interpretation.

Example

$$D = \mathcal{P}(\mathbb{N}) = \{\emptyset, \{zero\}, \{one\}, \{two\}, \ldots, \{zero, one\}, \ldots\}$$
$$a = [x \mapsto \{one\}, y \mapsto \{two\}, \ldots]$$
$$I = [0 \mapsto \emptyset, + \mapsto union, \ldots]$$

$$\begin{aligned}
[\![x + (y + 0)]\!]_a^{D,I} &= union([\![x]\!]_a^{D,I}, [\![y + 0]\!]_a^{D,I}) \\
&= union(a(x), [\![y + 0]\!]_a^{D,I}) \\
&= union(\{one\}, [\![y + 0]\!]_a^{D,I}) \\
&= union(\{one\}, union([\![y]\!]_a^{D,I}, [\![0]\!]_a^{D,I})) \\
&= union(\{one\}, union(a(y), I(0)) \\
&= union(\{one\}, union(\{two\}, emptyset)) \\
&= union(\{one\}, \{two\}) \\
&= \{one, two\}
\end{aligned}$$

The meaning of the term with another interpretation.

The Formal Semantics of Formulas

$$D, I, a \longrightarrow \boxed{[\![F]\!]} \longrightarrow \mathit{true}, \mathit{false}$$

- Formula semantics $[\![F]\!]^{D,I}_a \in \{\mathit{true}, \mathit{false}\}$
  - Given D, I, a, the semantics of term T is a truth value.
  - This value is defined by structural induction on F.

$$\begin{aligned} F \ :=\ & \top \mid \bot \mid p(t_1, \ldots, t_n) \\ & \mid (\neg F) \mid (F_1 \wedge F_2) \mid (F_1 \vee F_2) \mid (F_1 \rightarrow F_2) \mid (F_1 \leftrightarrow F_2) \\ & \mid (\forall x : F) \mid (\exists x : F) \mid \ldots \end{aligned}$$

- $[\![\top]\!]^{D,I}_a := \mathit{true}, [\![\bot]\!]^{D,I}_a := \mathit{false}$
- $[\![p(t_1, \ldots, t_n)]\!]^{D,I}_a := I(p)([\![t_1]\!]^{D,I}_a, \ldots, [\![t_n]\!]^{D,I}_a)$
  - The semantics of a atomic formula is the result of the interpretation of the predicate symbol applied to the values of the argument terms.

And now for the non-atomic formulas.

The Semantics of Propositional Formulas

- $[\![\neg F]\!]^{D,I}_{a} := \begin{cases} \mathit{true} & \text{if } [\![F]\!]^{D,I}_{a} = \mathit{false} \\ \mathit{false} & \text{else} \end{cases}$
- $[\![F_1 \wedge F_2]\!]^{D,I}_{a} := \begin{cases} \mathit{true} & \text{if } [\![F_1]\!]^{D,I}_{a} = [\![F_2]\!]^{D,I}_{a} = \mathit{true} \\ \mathit{false} & \text{else} \end{cases}$
- $[\![F_1 \vee F_2]\!]^{D,I}_{a} := \begin{cases} \mathit{false} & \text{if } [\![F_1]\!]^{D,I}_{a} = [\![F_2]\!]^{D,I}_{a} = \mathit{false} \\ \mathit{true} & \text{else} \end{cases}$
- $[\![F_1 \rightarrow F_2]\!]^{D,I}_{a} := \begin{cases} \mathit{false} & \text{if } [\![F_1]\!]^{D,I}_{a} = \mathit{true} \text{ and } [\![F_2]\!]^{D,I}_{a} = \mathit{false} \\ \mathit{true} & \text{else} \end{cases}$
- $[\![F_1 \leftrightarrow F_2]\!]^{D,I}_{a} := \begin{cases} \mathit{true} & \text{if } [\![F_1]\!]^{D,I}_{a} = [\![F_2]\!]^{D,I}_{a} \\ \mathit{false} & \text{else} \end{cases}$

The semantics coincides here with that of propositional logic.

The Semantics of Quantified Formulas

- $[\![\forall x : F]\!]^{D,I}_{a} := \begin{cases} \mathit{true} & \text{if } [\![F]\!]^{D,I}_{a[x \mapsto d]} = \mathit{true} \text{ for all } d \in D \\ \mathit{false} & \text{else} \end{cases}$
  - Formula is true, if body F is true for every value of the domain assigned to x.
- $[\![\exists x : F]\!]^{D,I}_{a} := \begin{cases} \mathit{true} & \text{if } [\![F]\!]^{D,I}_{a[x \mapsto d]} = \mathit{true} \text{ for some } d \in D \\ \mathit{false} & \text{else} \end{cases}$
  - Formula is true, if body F is true for at least one value of the domain assigned to x.

$$a[x \mapsto d](y) = \begin{cases} d & \mathit{if}\ x = y \\ a(y) & \mathit{else} \end{cases}$$

The core of the semantics.

Example

$D = \mathbb{N}_3 = \{zero, one, two\}$

$a = [x \mapsto one, y \mapsto two, z \mapsto two, \ldots]$, $I = [0 \mapsto zero, + \mapsto add, \ldots]$

$[\![\forall x : \exists y : x + y = z]\!]^{D,I}_{a} = true$

- $[\![\exists y : x + y = z]\!]^{D,I}_{a[x \mapsto zero]} = true$
  - $[\![x + y = z]\!]^{D,I}_{a[x \mapsto zero, y \mapsto zero]} = false$
  - $[\![x + y = z]\!]^{D,I}_{a[x \mapsto zero, y \mapsto one]} = false$
  - $[\![x + y = z]\!]^{D,I}_{a[x \mapsto zero, y \mapsto two]} = \underline{true}$
- $[\![\exists y : x + y = z]\!]^{D,I}_{a[x \mapsto one]} = true$
  - $[\![x + y = z]\!]^{D,I}_{a[x \mapsto one, y \mapsto zero]} = false$
  - $[\![x + y = z]\!]^{D,I}_{a[x \mapsto one, y \mapsto one]} = \underline{true}$
  - $[\![x + y = z]\!]^{D,I}_{a[x \mapsto one, y \mapsto two]} = false$
- $[\![\exists y : x + y = z]\!]^{D,I}_{a[x \mapsto two]} = true$
  - $[\![x + y = z]\!]^{D,I}_{a[x \mapsto two, y \mapsto zero]} = \underline{true}$
  - $[\![x + y = z]\!]^{D,I}_{a[x \mapsto two, y \mapsto one]} = false$
  - $[\![x + y = z]\!]^{D,I}_{a[x \mapsto two, y \mapsto two]} = false$

The systematic investigation of respectively search for assignments.

Nested Quantifiers

When quantifiers of different type are nested, the order matters.

Example

Domain: natural numbers.

$$\forall x : \exists y : x < y \quad \rightsquigarrow \quad \textit{true}$$

(Why? For the assignment $[x \mapsto \bar{x}]$ for x take $[y \mapsto \bar{x} + 1]$ as the assignment for y. The meaning of $x < y$ is then $\bar{x} < \bar{x} + 1$, which is true no matter what $\bar{x}$ is.)

$$\exists y : \forall x : x < y \quad \rightsquigarrow \quad \textit{false}$$

(Why? Assume it was true, i.e. there is an assignment $[y \mapsto \bar{y}]$ for y such that $x < y$ is true for all assignments for x. But take $[x \mapsto \bar{y}]$ as the assignment for x. The meaning of $x < y$ is then $\bar{y} < \bar{y}$, which is false, hence the original assumption must not be made, thus the meaning of the formula must be false.)

Semantics: Examples

- $\forall n : R(n, n)$
  - Domain: natural numbers.
  - R is interpreted as the divisibility relation on natural numbers.
  - Every natural number is divisible by itself. $\rightsquigarrow$ true
- $\forall n : R(n, n)$
  - Domain: real numbers.
  - R is interpreted as the less-than relation on real numbers.
  - Every real number is less than itself. $\rightsquigarrow$ false
- $\exists x : R(a, x) \wedge R(x, b)$
  - Domain: real numbers.
  - R is interpreted as the less-than relation on real numbers.
  - There is a real number x such that $a < x$ and $x < b$. $\rightsquigarrow$???
  - Assignment $[a \mapsto 5, b \mapsto 6]$: There is an assignment for x such that $5 < x$ and $x < 6$. $\rightsquigarrow$ true, e.g. $[x \mapsto 5.5]$
  - Assignment $[a \mapsto 7, b \mapsto 6]$: There is an assignment for x such that $7 < x$ and $x < 6$. $\rightsquigarrow$ false, why?

Semantics Convention

- The meaning of "=", logical connectives, and quantifiers is defined by above rules.
- The meaning of all other symbols depends on the interpretation which can be chosen as desired and must be given explicitly.
  - It is in principle possible to express "a divides the sum of b and c" by

    $$a \subseteq (b * c)$$

    using the interpretation

    $$[\subseteq \mapsto \text{ the divisibility relation}, \; * \mapsto \text{ the addition function}].$$

Convention: if the interpretation is not given explicitly, then a "standard interpretation" is assumed.

Semantic Notions

Let F denote a formula, M a structure, a an assignment.

- F is satisfiable, if $[\![F]\!]^M_a = \mathit{true}$ for some M and a.

  Example: $p(0,x)$ *is satisfiable;* $q(x) \land \neg q(x)$ *is not.*

- M is a model of F (short: $M \models F$), if $[\![F]\!]^M_a = \mathit{true}$ for all a.

  Example: $(\mathbb{N}, [0 \mapsto \mathit{zero}, p \mapsto \mathit{less\text{-}equal}]) \models p(0,x)$

- F is valid (short: $\models F$), if $M \models F$ for all M.

  Example: $\models p(x) \land (p(x) \rightarrow q(x)) \rightarrow q(x)$

  - If $\models F$, then F is true independent of the interpretation and the assignment, i.e., F is a "fact".

- Lemmas:
  - F is satisfiable, if $\neg F$ is not valid.
  - F is valid, if $\neg F$ is not satisfiable.

Logical Consequence

- F is a logical consequence of formula set Γ (short: $\Gamma \models F$), if for every structure M and assignment a, the following is true:

  If $[\![\, G \,]\!]^M_a = \mathit{true}$ *for every* G *in* Γ, *then also* $[\![\, F \,]\!]^M_a = \mathit{true}$.

  $\{p(x), p(x) \rightarrow q(x)\} \models q(x)$

  - If $\emptyset \models F$, then $\models F$, i.e., F is a "fact".
- F_2 is a logical consequence of formula F_1, if $\{F_1\} \models F_2$.
  - F_2 "follows from" F_1 in every structure and assignment.
- Lemma: $(\{G_1, \ldots, G_n\} \models F)$ is true if and only if the formula $(G_1 \wedge \ldots \wedge G_n \rightarrow F)$ is valid.

Logical consequence can be reduced to the validity of an implication.

Logical Equivalence

- Definition: two formulas F_1 and F_2 are logically equivalent (short: $F_1 \Leftrightarrow F_2$), if $F_1 \models F_2$ and $F_2 \models F_1$.
  - F_1 and F_2 mean the same, regardless of structure and assignment.
  - Every formula can always be substituted by an equivalent one.
- Lemma: if $F \Leftrightarrow F'$ and $G \Leftrightarrow G'$, then

$$\neg F \Leftrightarrow \neg F'$$
$$F \land G \Leftrightarrow F' \land G'$$
$$F \lor G \Leftrightarrow F' \lor G'$$
$$F \rightarrow G \Leftrightarrow F' \rightarrow G'$$
$$F \leftrightarrow G \Leftrightarrow F' \leftrightarrow G'$$
$$\forall x : F \Leftrightarrow \forall x : F'$$
$$\exists x : F \Leftrightarrow \exists x : F'$$

- Lemma: $(F_1 \Leftrightarrow F_2)$ is true if and only if formula $(F_1 \leftrightarrow F_2)$ is valid.

Logical equivalence can be reduced to the validity of an equivalence.

Equivalent Formulas

In addition to equivalences for connectives (see propositional logic):

$$
\begin{array}{lcl}
\neg(\forall x : F) & \Leftrightarrow & \exists x : \neg F \quad \text{(De-Morgan)} \\
\neg(\exists x : F) & \Leftrightarrow & \forall x : \neg F \quad \text{(De-Morgan)} \\
\forall x : (F_1 \wedge F_2) & \Leftrightarrow & (\forall x : F_1) \wedge (\forall x : F_2) \\
\exists x : (F_1 \vee F_2) & \Leftrightarrow & (\exists x : F_1) \vee (\exists x : F_2) \\
\forall x : (F_1 \vee F_2) & \Leftrightarrow & F_1 \vee (\forall x : F_2), \text{ if } x \text{ does not occur free in } F_1 \\
\exists x : (F_1 \wedge F_2) & \Leftrightarrow & F_1 \wedge (\exists x : F_2), \text{ if } x \text{ does not occur free in } F_1
\end{array}
$$

For a finite domain $\{v_1, \ldots, v_n\}$:

$$
\begin{array}{lcl}
\forall x : F & \Leftrightarrow & F[v_1/x] \wedge \ldots \wedge F[v_n/x] \\
\exists x : F & \Leftrightarrow & F[v_1/x] \vee \ldots \vee F[v_n/x]
\end{array}
$$

$E[t/x]$: the expression E with every free occurrence of x substituted by the term t. ($\leadsto$ E has the same meaning for x as $E[t/x]$ has for t.)

Some Formula Transformations

- Push negations from the outside to the inside.

$$\begin{aligned}
&\neg(\forall x : p(x) \rightarrow \exists y : q(x,y)) \\
\Leftrightarrow\ &\exists x : \neg(p(x) \rightarrow \exists y : q(x,y)) \\
\Leftrightarrow\ &\exists x : \neg((\neg p(x)) \vee \exists y : q(x,y)) \\
\Leftrightarrow\ &\exists x : ((\neg\neg p(x)) \wedge \neg\exists y : q(x,y)) \\
\Leftrightarrow\ &\exists x : (p(x) \wedge \neg\exists y : q(x,y)) \\
\Leftrightarrow\ &\exists x : (p(x) \wedge \forall y : \neg q(x,y))
\end{aligned}$$

- Reduce the scope of quantifiers.

$$\begin{aligned}
&\forall x, y : (p(x) \rightarrow q(x,y)) \\
\Leftrightarrow\ &\forall x, y : (\neg p(x) \vee q(x,y)) \\
\Leftrightarrow\ &\forall x : (\neg p(x) \vee \forall y : q(x,y)) \\
\Leftrightarrow\ &\forall x : (p(x) \rightarrow \forall y : q(x,y))
\end{aligned}$$

- Replace finite quantifications.

$$\begin{aligned}
&\forall x \in \{0,1\} : p(x) \\
\Leftrightarrow\ &p(0) \wedge p(1)
\end{aligned}$$