FIRST-ORDER 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

JOHANNES KEPLER UNIVERSITY LINZ

The Semantics of First-Order Logic

In first-order logic, the semantics (meaning) depends on a structure and an assignment.

- A structure (D, I) consists of a domain D and an interpretation I on D:
  - A domain is a non-empty collection of objects (e.g., a set $D \neq \emptyset$).
    - The "universe" about which a first-order logic formula talks.
  - An interpretation maps every constant and function/predicate symbol to its meaning:
    - Constant $c \in \mathcal{C}$: $I(c)$ is an object in D ($I(c) \in D$).
    - Function symbol $f \in \mathcal{F}$ of arity n: $I(f)$ is an n-ary function on D ($I(f) \colon D^n \to D$).
    - Predicate symbol $p \in \mathcal{P}$ of arity n: $I(p)$ is an n-ary predicate/relation on D ($I(p) \subseteq D^n$).
- An assignment a maps every variable to its meaning:
  - Variable $v \in \mathcal{V}$: $a(v)$ is an object in D ($a(v) \in D$).

$$D = \mathbb{N}$$

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

$$a = [x \mapsto \text{one}, y \mapsto \text{zero}, z \mapsto \text{three}, \ldots]$$

Informal Semantics

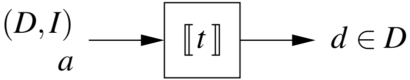
Terms The meaning of a term is an object in D.

- The meaning of a variable v is the object assigned to it by a, i.e., $a(v)$.
- The meaning of a constant c is its interpretation in I, i.e., $I(c)$.
- The meaning of a function application $f(t_1,\ldots,t_n)$ is the result of applying its interpretation $I(f)$ to the meanings of $t_1,\ldots,t_n$.

Formulas The meaning of a formula is "true" or "false".

- The meaning of an atomic formula $p(t_1,\ldots,t_n)$ is the result of applying its interpretation $I(p)$ to the meanings of $t_1,\ldots,t_n$.
  - An equality $t_1 = t_2$ is "true", if t_1 has the same meaning as t_2.
- The meaning of the propositional constructions is as already known.
- $(\forall x\colon F)$ is true if F is true *for all* possible objects assigned to x in a.
- $(\exists x\colon F)$ is true if F is true *for some* possible object assigned to x in a.

The Formal Semantics of Terms



- Term semantics $[\![t]\!]_a^{D,I} \in D$
  - Given structure (D,I) and assignment a, the semantics of term t is an object in D.
$$t \ ::= \ v \mid c \mid f(t_1, \ldots, t_n)$$
  - The meaning of a variable is the value given by the assignment:
$$[\![v]\!]_a^{D,I} := a(v)$$
  - The meaning of a constant is the value given by the interpretation:
$$[\![c]\!]_a^{D,I} := I(c)$$
  - The meaning of a function application is the result of the interpretation of the function symbol applied to the values of the argument terms:
$$[\![f(t_1, \ldots, t_n)]\!]_a^{D,I} := I(f)([\![t_1]\!]_a^{D,I}, \ldots, [\![t_n]\!]_a^{D,I})$$

The recursive definition of a function evaluating a term.

Example

$$D = \mathbb{N} = \{\mathsf{zero}, \mathsf{one}, \mathsf{two}, \mathsf{three}, \ldots\}$$

$$I = [0 \mapsto \mathsf{zero}, + \mapsto \mathsf{add}, \ldots]$$

$$a = [x \mapsto \mathsf{one}, y \mapsto \mathsf{two}, \ldots]$$

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

The meaning of the term with the “usual” interpretation.

Example

$$D = \mathcal{P}(\mathbb{N}) = \{\emptyset, \{\mathsf{zero}\}, \{\mathsf{one}\}, \{\mathsf{two}\}, \ldots, \{\mathsf{zero}, \mathsf{one}\}, \ldots\}$$

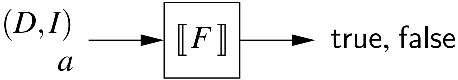
$$I = [0 \mapsto \emptyset, + \mapsto \mathsf{union}, \ldots]$$

$$a = [x \mapsto \{\mathsf{one}\}, y \mapsto \{\mathsf{two}\}, \ldots]$$

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

The meaning of the term with another interpretation.

The Formal Semantics of Formulas



- Formula semantics $[\![F]\!]_a^{D,I} \in \{\text{true}, \text{false}\}$
  - Given structure (D, I) and assignment a, the semantics of formula F is a truth value.

$$F ::= p(t_1, \ldots, t_n) \mid \top \mid \bot \mid \ldots \mid (\forall v\colon F) \mid (\exists v\colon F)$$

  - The meaning of an atomic formula is the result of the interpretation of the predicate symbol applied to the values of the argument terms (fixed interpretation of equality).

$$[\![p(t_1, \ldots, t_n)]\!]_a^{D,I} := I(p)\big([\![t_1]\!]_a^{D,I}, \ldots, [\![t_n]\!]_a^{D,I}\big)$$

$$[\![t_1 = t_2]\!]_a^{D,I} := \begin{cases} \text{true} & \text{if } [\![t_1]\!]_a^{D,I} = [\![t_2]\!]_a^{D,I} \\ \text{false} & \text{else} \end{cases}$$

  - The meaning of the logical constants:

$$[\![\top]\!]_a^{D,I} := \text{true} \qquad [\![\bot]\!]_a^{D,I} := \text{false}$$

The meaning of the basic formulas.

The Semantics of Propositional Formulas

- The meaning of the logical connectives:

$$[\![\neg F]\!]_a^{D,I} := \begin{cases} \text{true} & \text{if } [\![F]\!]_a^{D,I} = \text{false} \\ \text{false} & \text{else} \end{cases}$$

$$[\![F_1 \land F_2]\!]_a^{D,I} := \begin{cases} \text{true} & \text{if } [\![F_1]\!]_a^{D,I} = [\![F_2]\!]_a^{D,I} = \text{true} \\ \text{false} & \text{else} \end{cases}$$

$$[\![F_1 \lor F_2]\!]_a^{D,I} := \begin{cases} \text{false} & \text{if } [\![F_1]\!]_a^{D,I} = [\![F_2]\!]_a^{D,I} = \text{false} \\ \text{true} & \text{else} \end{cases}$$

$$[\![F_1 \rightarrow F_2]\!]_a^{D,I} := \begin{cases} \text{false} & \text{if } [\![F_1]\!]_a^{D,I} = \text{true and } [\![F_2]\!]_a^{D,I} = \text{false} \\ \text{true} & \text{else} \end{cases}$$

$$[\![F_1 \leftrightarrow F_2]\!]_a^{D,I} := \begin{cases} \text{true} & \text{if } [\![F_1]\!]_a^{D,I} = [\![F_2]\!]_a^{D,I} \\ \text{false} & \text{else} \end{cases}$$

An embedding of the semantics of propositional logic into first-order logic.

The Semantics of Quantified Formulas

- $(\forall x\colon F)$ is true, if F is true for every possible object d assigned to variable x:

$$[\![\forall x\colon F]\!]_a^{D,I} := \begin{cases} \text{true} & \text{if } [\![F]\!]_{a[x\mapsto d]}^{D,I} = \text{true for all } d \text{ in } D \\ \text{false} & \text{else} \end{cases}$$

- $(\exists x\colon F)$ is true, if F is true for at least one possible object d assigned to variable x:

$$[\![\exists x\colon F]\!]_a^{D,I} := \begin{cases} \text{true} & \text{if } [\![F]\!]_{a[x\mapsto d]}^{D,I} = \text{true for some } d \text{ in } D \\ \text{false} & \text{else} \end{cases}$$

- Assignment a updated by the assignment of object d to variable x:

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

The core of the semantics of first-order logic.

Example

$$D = \mathbb{N}_3 = \{\text{zero}, \text{one}, \text{two}\} \quad I = [0 \mapsto \text{zero}, + \mapsto \text{add}, \ldots] \quad a = [x \mapsto \text{one}, y \mapsto \text{two}, z \mapsto \text{two}, \ldots]$$

$[\![\forall x\colon \exists y\colon x+y=z]\!]_a^{D,I} = ?$

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

$[\![\forall x\colon \exists y\colon x+y=z]\!]_a^{D,I} = \text{true}$.

Semantics: Structures and Assignments

- $\forall n\colon R(n,n)$
  - The domain of natural numbers with R interpreted as the divisibility relation.
  - "Every natural number is divisible by itself": true (for every assignment).
- $\forall n\colon R(n,n)$
  - The domain of natural numbers with R interpreted as the less-than relation.
  - "Every natural number is less than itself": false (for every assignment).
- $\exists x\colon R(y,x) \wedge R(x,z)$
  - The domain of natural numbers with R interpreted as the less-than relation.
  - "There exists a natural number x with $y < x$ and $x < z$".
  - Assignment $[y \mapsto 2, z \mapsto 4]$: true (there is the value $x = 3$ with $2 < x$ and $x < 4$).
  - Assignment $[y \mapsto 2, z \mapsto 3]$: false (there is no value for x with $2 < x$ and $x < 3$).

The truth value of a formula depends on the structure and the assignment.

Semantics: Nested Quantifiers

Consider the domain of natural numbers with the usual interpretation of $<$.

- $(\forall x\colon \exists y\colon x < y)$: true.
  - "For every natural number x there exists some y such that x is less than y".
  - For every natural number x, there is indeed such a y, namely $y := x + 1$.
- $(\exists y\colon \forall x\colon x < y)$: false
  - "There exists some natural number y such that every x is less than y."
  - We assume that the formula is true and derive a contradiction. Because of the assumption, there exists some natural number y such that $(\forall x\colon x < y)$ is true. But then, since $x < y$ is true for every value of x, it is also true for $x := y$. Thus $y < y$ is true, which we know to be false.

The order of nested quantifiers matters.

Semantic Notions: Satisfiability and Validity

Let F denote a formula, $M = (D, I)$ a structure, a an assignment.

Satisfiability Formula F is satisfiable, if there exists some structure M and assignment a such that $[\![F]\!]_a^M = \text{true}$.

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

Model Structure M is a model of formula F, written as $M \models F$, if for every assignment a, we have $[\![F]\!]_a^M = \text{true}$.

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

Validity Formula F is valid, written as $\models F$, if every structure M is a model of F, i.e., for every structure M we have $M \models F$.

- Example: $\models p(x) \land (p(x) \to q(x)) \to q(x)$

Semantic Notions: Logical Consequence and Equivalence

Logical Consequence Formula F_2 is a logical consequence of formula F_1, written as $F_1 \models F_2$, if for every structure M and assignment a, the following is true:
If $[\![F_1]\!]_a^M = \text{true}$, then also $[\![F_2]\!]_a^M = \text{true}$.

- Example: $p(x) \wedge (p(x) \rightarrow q(x)) \models q(x)$

Logical Consequence Generalized Formula F is a logical consequence of formulas $F_1, \ldots, F_n$, written $F_1, \ldots, F_n \models F$, if for every M and a the following is true:
If for every formula F_i we have $[\![F_i]\!]_a^M = \text{true}$, then $[\![F]\!]_a^M = \text{true}$.

- Example: $p(x), q(x) \models p(x) \wedge q(x)$

Logical Equivalence Formulas F_1 and F_2 are logically equivalent, written as $F_1 \Leftrightarrow F_2$, if and only if F_1 is a logical consequence of F_2 and vice versa, i.e., $F_1 \models F_2$ and $F_2 \models F_1$.

- Example: $p(x) \rightarrow q(x) \Leftrightarrow \neg p(x) \vee q(x)$

Semantic Notions: Propositions

Satisfiability and Validity

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

Logical Consequence and Equivalence

- Formula F_2 is a logical consequence of formula F_1 (i.e., $F_1 \models F_2$) if and only if the formula $(F_1 \rightarrow F_2)$ is valid.
- Formula F is a logical consequence of formulas $F_1, \ldots, F_n$ (i.e., $F_1, \ldots, F_n \models F$) if and only if the formula $(F_1 \wedge \ldots \wedge F_n \rightarrow F)$ is valid.
- Formula F_1 and formula F_2 are logically equivalent (i.e., $F_1 \Leftrightarrow F_2$) if and only if the formula $(F_1 \leftrightarrow F_2)$ is valid.

Logical consequence/equivalence reduced to validity of an implication/equivalence.

Logical Equivalence: Formula Substitutions

Assume $F \Leftrightarrow F'$ and $G \Leftrightarrow G'$. Then we have the following equivalences:

$$\neg F \Leftrightarrow \neg F'$$

$$F \wedge G \Leftrightarrow F' \wedge G'$$

$$F \vee G \Leftrightarrow F' \vee G'$$

$$F \rightarrow G \Leftrightarrow F' \rightarrow G'$$

$$F \leftrightarrow G \Leftrightarrow F' \leftrightarrow G'$$

$$\forall x\colon F \Leftrightarrow \forall x\colon F'$$

$$\exists x\colon F \Leftrightarrow \exists x\colon F'$$

Logically equivalent formulas can be subsituted in any context.

Logical Equivalence: Rules

In addition to the logical equivalences for connectives in propositional logic:

$$\neg\forall x\colon F \Leftrightarrow \exists x\colon \neg F \qquad \text{(De Morgan's Law)}$$

$$\neg\exists x\colon F \Leftrightarrow \forall x\colon \neg F \qquad \text{(De Morgan's Law)}$$

$$\forall x\colon (F_1 \wedge F_2) \Leftrightarrow (\forall x\colon F_1) \wedge (\forall x\colon F_2)$$

$$\exists x\colon (F_1 \vee F_2) \Leftrightarrow (\exists x\colon F_1) \vee (\exists x\colon F_2)$$

$$\forall x\colon (F_1 \vee F_2) \Leftrightarrow F_1 \vee (\forall x\colon F_2) \qquad \text{if } x \text{ is not free in } F_1$$

$$\exists x\colon (F_1 \wedge F_2) \Leftrightarrow F_1 \wedge (\exists x\colon F_2) \qquad \text{if } x \text{ is not free in } F_1$$

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

$$\forall x\colon F \Leftrightarrow F[v_1/x] \wedge \ldots \wedge F[v_n/x]$$

$$\exists x\colon F \Leftrightarrow F[v_1/x] \vee \ldots \vee F[v_n/x]$$

Logical Equivalence: Examples

- Push negations from the outside to the inside:

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

- Reduce the scope of quantifiers:

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

- Replace quantification in a finite domain $D = \{0, 1, 2\}$:

$$\forall x\colon p(x) \Leftrightarrow p(0) \wedge p(1) \wedge p(2)$$