2.1 Syntax of Predicate Logic

Dienstag, 25. April 2017 08:30

2. Basics of Predicate Logic

2.1. Syntax of Predicate Logic

Signature: Alphabet for Predicate Logic

Def 2.1.1 (Signature)

A signature (Σ, Δ) is a pair with $\Sigma = \bigcup_{n \in \mathbb{N}} \Sigma_n$ and $\Delta = \bigcup_{n \in \mathbb{N}} \Delta_n$, where all Σ_n and Δ_n are pairwise disjoint. Every $f \in \Sigma_n$ is a function symbol of arity n, every $p \in \Delta_n$ is a predicate symbol of arity n. We always require $\Sigma_0 \neq \emptyset$.

Σ_0: the set of constants

Ex. 2.1.2

Example program uses the following signature (Σ, Δ):

$\Sigma = \Sigma_0 \cup \Sigma_3$, $\Delta = \Delta_1 \cup \Delta_2$

See slide 9

Terms are the "objects" of pred. logic.

Def 2.1.3 (Terms)

Let (Σ, Δ) be a signature, let $\mathcal{V}$ be a set of

Let (Σ, Δ) be a signature, let $\mathcal{V}$ be a set of variables with $\mathcal{V} \cap \Sigma = \emptyset$. Then $\mathcal{T}(\Sigma, \mathcal{V})$ is the set of <u>terms</u> (over Σ and $\mathcal{V}$). Here, $\mathcal{T}(\Sigma, \mathcal{V})$ is the smallest set with:

- $\mathcal{V} \subseteq \mathcal{T}(\Sigma, \mathcal{V})$ and
- $f(t_1, \dots, t_n) \in \mathcal{T}(\Sigma, \mathcal{V})$ if $f \in \Sigma_n$ and $t_1, \dots, t_n \in \mathcal{T}(\Sigma, \mathcal{V})$ for some $n \in \mathbb{N}$.

$\mathcal{T}(\Sigma)$ stands for $\mathcal{T}(\Sigma, \emptyset)$, i.e., the set of <u>ground terms</u>. For any term t, $\mathcal{V}(t)$ is the set of all variables in t.

<u>Ex 2.14</u> We use the signature of Ex. 2.12. If $\mathcal{V} = \{X, Y, Z, Grandma, Mom, \dots\}$, then we have the following terms in $\mathcal{T}(\Sigma, \mathcal{V})$:

X, Y, $monika$, 5,

$date(Mom, monika, 5)$,

$date(25, 4, 2017)$,

$date(date(X, Y, Z), monika, 7)$, ...

ground terms

<u>Formulas</u> represent statements about terms.

<u>Def 2.15</u> (Formulas)

Def 2.15 (Formulas)

Let (Σ, Δ) be a signature, let $\mathcal{V}$ be a set of variables.
The set of <u>atomic formulas</u> over (Σ, Δ) and $\mathcal{V}$ is defined as:

$$At(\Sigma, \Delta, \mathcal{V}) = \{ p(t_1, \ldots, t_n) \mid p \in \Delta_n \text{ for some } n, \; t_1, \ldots, t_n \in \mathcal{T}(\Sigma, \mathcal{V}) \}.$$

$\mathcal{F}(\Sigma, \Delta, \mathcal{V})$ is the set of <u>formulas</u> over (Σ, Δ) and $\mathcal{V}$.
Here, $\mathcal{F}(\Sigma, \Delta, \mathcal{V})$ is the smallest set with:

- $At(\Sigma, \Delta, \mathcal{V}) \subseteq \mathcal{F}(\Sigma, \Delta, \mathcal{V})$
- if $\varphi \in \mathcal{F}(\Sigma, \Delta, \mathcal{V})$, then $\underbrace{\neg}_{\text{"not"}} \varphi \in \mathcal{F}(\Sigma, \Delta, \mathcal{V})$
- if $\varphi_1, \varphi_2 \in \mathcal{F}(\Sigma, \Delta, \mathcal{V})$, then
  $(\varphi_1 \underbrace{\wedge}_{\text{"and"}} \varphi_2), (\varphi_1 \underbrace{\vee}_{\text{"or"}} \varphi_2), (\varphi_1 \underbrace{\rightarrow}_{\text{"implies"}} \varphi_2), (\varphi_1 \underbrace{\leftrightarrow}_{\text{"is equivalent"}} \varphi_2) \in \mathcal{F}(\Sigma, \Delta, \mathcal{V})$
- if $X \in \mathcal{V}$, $\varphi \in \mathcal{F}(\Sigma, \Delta, \mathcal{V})$ then
  $(\underbrace{\forall}_{\text{"for all"}} X \; \varphi), (\underbrace{\exists}_{\text{"there exists"}} X \; \varphi) \in \mathcal{F}(\Sigma, \Delta, \mathcal{V})$

universal quantifier — existential quantifier

For a formula φ, $\mathcal{V}(\varphi)$ is the set of variables in φ.

A variable X is <u>free</u> in a formula φ if

- φ is an atomic formula and $X \in \mathcal{V}(\varphi)$ or
- $\varphi = \neg \varphi_1$ and X is free in φ_1 or
- $\varphi = (\varphi_1 \cdot \varphi_2)$ with $\cdot \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ and X is free in φ_1 or in φ_2 or
- $\varphi = (Q\ Y\ \varphi_1)$ with $Q \in \{\forall, \exists\}$, X is free in φ_1, and $X \neq Y$

A formula is <u>closed</u> if it has no free variables.
A formula is <u>quantifier-free</u> if it does not contain $\forall$ or $\exists$.

Notation:

- We omit brackets if this does not create confusion: $\forall X\ \mathit{female}(X)$
- We write variables with upper-case letters and function + pred. symbols with lower-case letters.

<u>Ex. 2.1.6</u> Formulas over the signature of Ex. 2.1.2

$\mathit{female}(\mathit{monika})$ $\in At(\Sigma, \Delta, \mathcal{V})$

← X is free

$female(monika) \in At(\Sigma, \Delta, \mathcal{V})$

$motherOf(X, susanne) \in At(\Sigma, \Delta, \mathcal{V})$ ← X is free

$born(monika, date(25, 4, 2017)) \in At(\Sigma, \Delta, \mathcal{V})$

$\forall W\, (married(gerd, W) \wedge motherOf(W, C)) \in \mathcal{F}(\Sigma, \Delta, \mathcal{V})$ ← C is free

$married(gerd, W) \wedge \neg(\forall W\ motherOf(W, C)) \in \mathcal{F}(\Sigma, \Delta, \mathcal{V})$

↑ W and C are free

We abbreviate $\forall X_1 (\ldots (\forall X_n\ \varphi) \ldots)$ by

$\forall X_1, \ldots, X_n\ \ \varphi$ and

$\exists X_1 (\ldots (\exists X_n\ \varphi) \ldots)$ by

$\exists X_1, \ldots, X_n\ \ \varphi$

<u>Ex 2.1.7</u> Every logic program can be translated into a set of formulas. All variables in the prog. are universally quantified.

See slide 10

<u>Def 2.1.8</u> (Substitution)

A mapping $\sigma : \mathcal{V} \to \mathcal{T}(\Sigma, \mathcal{V})$ is a substitution if $\sigma(X) \neq X$ for finitely many $X \in \mathcal{V}$.

$DOM(\sigma) = \{X \in \mathcal{V} \mid \sigma(X) \neq X\}$ is the <u>domain</u> of σ.

$RANGE(\sigma) = \{\sigma(X) \mid X \in DOM(\sigma)\}$ is the <u>range</u> of σ.

Since $DOM(\sigma)$ is finite, a substitution σ can be represented as a finite set of pairs
$\{ X/\sigma(X) \mid X \in DOM(\sigma) \}$.

A subst. σ is a <u>ground substitution</u> if

$$\mathcal{V}(\sigma(X)) = \emptyset \quad \text{for all } X \in DOM(\sigma).$$

Ex: $\sigma = \{ X/monika, Y/date(X,Y,Z) \}$
is not a ground subst.
$DOM(\sigma) = \{X, Y\}$
$RANGE(\sigma) = \{monika, date(X,Y,Z)\}$
$\sigma(Y) = date(X,Y,Z)$
$\sigma(Z) = Z$

A subst. σ is a <u>variable renaming</u> if
σ is injective and $\sigma(X) \in \mathcal{V}$ for all $X \in \mathcal{V}$.

Ex: $\sigma = \{ X/Y, Y/Z \}$ no variable renaming
$\sigma(X) = Y$, $\sigma(Y) = Z$, $\sigma(Z) = Z$
not injective

But $\sigma' = \{X/Y, Y/Z, Z/X\}$ is a var. renaming

Substitutions are also applied to terms, i.e.,
$\sigma : \mathcal{T}(\Sigma, \mathcal{V}) \to \mathcal{T}(\Sigma, \mathcal{V})$ by defining

$$\sigma(f(t_1,\ldots,t_n)) = f(\sigma(t_1),\ldots,\sigma(t_n))$$

Ex: $\sigma = \{X/monika,\ Y/gerd\}$

$$\sigma(date(X,Y,Z)) = date(monika, gerd, Z)$$

Substitutions can also be applied to formulas:

- $\sigma(p(t_1,\ldots,t_n)) = p(\sigma(t_1),\ldots,\sigma(t_n))$
- $\sigma(\neg\varphi) = \neg\,\sigma(\varphi)$
- $\sigma(\varphi_1 \cdot \varphi_2) = \sigma(\varphi_1) \cdot \sigma(\varphi_2)$ for $\cdot \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$
- $\sigma(QX\ \varphi) = QX\ \sigma(\varphi)$ for $Q \in \{\forall, \exists\}$, if $X \notin DOM(\sigma) \cup \mathcal{V}(RANGE(\sigma))$

It should not matter whether we write $\forall X\ female(X)$ or $\forall Y\ female(Y)$

- $\sigma(QX\ \varphi) = QX'\ \sigma(\delta(\varphi))$ for $Q \in \{\forall, \exists\}$, $X \in DOM(\sigma) \cup \mathcal{V}(RANGE(\sigma))$.

Here, X' is a fresh variable with $X' \notin DOM(\sigma) \cup \mathcal{V}(RANGE(\sigma)) \cup \mathcal{V}(\varphi)$ and $\delta = \{X/X'\}$.

An instance $\sigma(t)$ of a term t (resp. $\sigma(\varphi)$ of a formula φ) is a <u>ground instance</u> if

$\mathcal{V}(\sigma(t)) = \emptyset$ (resp. $\mathcal{V}(\sigma(\varphi)) = \emptyset$)

$$\vartheta(\sigma(t)) = \emptyset \quad (\text{resp. } \vartheta(\overline{\sigma}(\varphi)) = \emptyset).$$

Ex. 2.1.9

$$\sigma = \{X/date(X,Y,Z),\ Y/monika,\ Z/date(Z,Z,Z)\}.$$

Then:

$$\sigma(date(X,Y,Z)) = date(date(X,Y,Z),\ monika,\ date(Z,Z,Z))$$

$$\sigma(\underbrace{\forall Y\ married(X,Y)}_{\forall Y'\ married(X,Y')}) = \forall Y'\ married(date(X,Y,Z),\ Y')$$