Monadic Second Order Logic (MSO)

mercredi 4 novembre 2015 09:53

First order logic

Logic formulas are defined on a signature

$$\langle \underbrace{f_1, f_2, \dots, f_n}_{\text{functions operators constants}}, \underbrace{P_1, \dots, P_m}_{\text{predicates}} \rangle$$

predicates ← Say "yes" or "no" about their inputs

A signature is a syntactique notion, function f_i has no special meaning it just represents any function

Ex: $\forall x \forall y \quad x + y = y + x$ means that $+$ is commutative, whatever "+" is

The only information about symbols in the signature is their arities : the number of argument to the function/operator/predicate

Ex: $\langle +, -, \cos, \leq \rangle$

- $\leq$: binary predicate (no link with order)
- $\cos$: unary function (no link with cosine)
- $-$: unary operator (no link with opposite)
- $+$: binary operator (no link with addition)

A constant is a 0-ary function

Note that an n-ary function can be represented by a (n+1)-ary predicate:

The n-ary function f is represented by the (n+1)-ary predicate F defined by:

$$F(x_1, \dots, x_n, y) \text{ if and only if } f(x_1, \dots, x_n) = y$$

Ex: if cos is represented by Cos, the fact it is surjective is expressed by $\forall x\ \exists y\quad Cos(y, x)$

Remark that not all predicate represent a function; it must verifies that

$$\forall x_1, \dots, x_n\ \exists!\, y\ \ F(x_1, \dots, x_n, y)$$

"there exists a unique ..." see below

Building first order formulas

Terms are:
- variables x
- constants i
- functions/operators applied to terms $\cos(x)+i$

Atomic formula are:
- true or false
- 0-ary predicates
- Predicates applied to terms

Formulas are:
- $\varphi \wedge \psi$, $\varphi \vee \psi$, $\neg\varphi$, $\varphi \to \psi$, $\varphi \longleftrightarrow \psi$
- $\forall x\ \varphi \quad \exists x\ \varphi$
- an atomic formula

Defining new "functions" (#define in logic)

When you have that a formula $\forall x_1 \dots, x_n\ \exists y\ \varphi$ is true/proved, you can introduced a new functionnal symbol (not in the signature; not to be added to the signature) to denote the y, unique or not, given by the formula from $x_1 \dots, x_n$

↳ It is a shortcut to be replaced by an existential quantification

-

Ex: $\forall x \; \exists y \quad x+y=0$ allows to define a unary operator $-$

The formula $\forall x \forall y \quad -(x+y) = (-x)+(-y)$ is in fact

$$\forall x \forall y \; \exists a_1 \exists a_2 \exists a_3 \quad a_1+x=0 \wedge a_2+y=0 \wedge a_3+(x+y)=0 \wedge a_3 = a_1+a_2$$

Second order logic

Second order logic allows to quantify on predicates (thus on functions or sets)

Ex: a formula that states that there exists an order relation

first order variables

$$\exists R \; \forall x \; R(x,x) \wedge \forall x \forall y \; R(x,y) \wedge R(y,x) \Rightarrow x=y \wedge \forall x \forall y \forall z \; R(x,y) \wedge R(y,z) \Rightarrow R(x,z)$$

second order variable

a formula that states that any stable set is full or empty:

$$\forall S \; \forall x \; S(x) \Rightarrow S(f(x)) \Rightarrow \forall x \; \neg S(x) \vee \forall x \; S(x)$$

↳ here $S(x)$ means x is in the set represented by predicate S

a formula which states that only the identity commutes with f:

$$\underbrace{\forall F \; \forall x \; \exists! y \; F(x,y)}_{\text{"for all function } F\text{"}} \Rightarrow \forall x \; \exists a \exists b \; \overbrace{F(x,a)}^{a=F(x)} \wedge \overbrace{F(f(x),b)}^{b=F(f(x))} \wedge f(a)=b \Rightarrow \forall x \; F(x,x)$$

As this last formula is a bit tedious to read, we write, by abuse of notations:

$$\forall E\ \forall x\ x \in E \Rightarrow f(x) \in E \implies \forall x\ x \in E \lor \forall x\ x \notin E$$

$$\forall g,\ \forall x\ g(f(x)) = f(g(x)) \Rightarrow \forall x\ g(x) = x$$

Remarks

"such that" has a different meaning for $\forall$ and $\exists$.

$\forall x$ such that $x > 1$, we have $\frac{1}{x} < x$

is

$$\forall x \quad x > 1 \implies \frac{1}{x} < x$$

while

$\exists x$ such that $x < 1$ where $\frac{1}{x} < x$

is

$$\exists x \quad x < 1 \quad \land \quad \frac{1}{x} < x$$

Monadic second order logic is the restriction of second order logic to unary predicates: one can only quantify on sets and elements

Ex: $\forall E\ \forall x\ x \in E \Rightarrow f(x) \in E \implies \forall x\ x \in E \lor \forall x\ x \notin E$

Definability / decidability

A structure is a tuple $\langle E, \sigma \rangle$ where E is the domain of the structure (a set or a class) and σ a list of predicates, functions, operators or constants defined on E:

- constants are elements of E
- predicates, operators and function range over E: if f is a binary function, for all x and y of E, $f(x,y)$ has a value in E

The elements of σ seen as syntactic objects form the signature of the structure. For a structure, any close formula on this signature is either true or false

The theory of a structure $\mathcal{S}$ is the set of all the true (first order) formulas φ (without free variables) on the signature of the structure: $Th(\mathcal{S}) = \{\varphi \mid \mathcal{S} \models \varphi\}$

The [monadic] second order theory of a structure is the set of all the [monadic] second order true formulas (without free variables) on the signature of the structure

A theory is decidable if there exists an algorithm which tell if a formula is true or not

<u>Theorem</u> - the theory of $\langle \mathbb{N}, +, < \rangle$ is decidable

- the theory of $\langle \mathbb{N}, +, \times \rangle$ is undecidable
- the theory of $\langle \mathbb{R}, +, <, 0 \rangle$ is decidable
- the MSO theory of $\langle \mathbb{N}, S \rangle$ is decidable $(S : x \mapsto x+1)$

A n-ary predicate P (function, set) is [X] definable in a structure if there exists a [X] formula φ with $x_1, \dots, x_n$ as only free variables such that for all $a_1, \dots, a_n \in E$, $P(a_1, \dots, a_n)$ is true if and only if the formula φ where x_i is replaced by a_i is true

Ex: $\leq$ is definable in $\langle \mathbb{N}, + \rangle$:

$x \leq y$ defined by $\exists n \quad x + n = y$

$<$ is definable in $\langle \mathbb{N}, + \rangle$:

$x < y$ defined by $\neg\, y \leq x \quad (\neg \exists n \; y + n = x)$

$\leq$ is MSO definable in $\langle \mathbb{N}, S \rangle \quad (S(x) = x+1)$

$x \leq y$ defined by

$$\forall E \; (x \in E \wedge \forall h \; h \in E \Rightarrow S(h) \in E) \Rightarrow y \in E$$

"y belongs to all set containing x and stable under S"

ω-languages and formulas

Let $\Sigma = \{a_0, \dots, a_{|\Sigma|-1}\}$ be a finite alphabet and u an ω-word

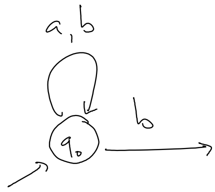
We define, for $a \in \Sigma$, P_a^u the set

$$P_a^u = \{ n \in \mathbb{N},\ u_n = a \}$$

An ω-language $\mathcal{L}$ is [X]-definable in a structure $\mathcal{S} = \langle \mathbb{N}, \sigma \rangle$ if the predicate P defined by $P(P_{a_0}^u, \dots, P_{a_{|\Sigma|-1}}^u) \Leftrightarrow u \in \mathcal{L}$ is [X]-definable in $\mathcal{S}$.

Theorem : an ω-language is MSO-definable in $\langle \mathbb{N}, \leq \rangle$ if and only if it is ω-regular

Ex (if) :



$$\exists Q_0\ Q_1 \quad \begin{cases} 0 \in Q_0 \wedge \forall i\ \exists j\ \ j \geq i \wedge j \in Q_1 \\ \forall i\ (i \in Q_0 \vee i \in Q_1) \wedge \neg (i \in Q_0 \wedge i \in Q_1) \\ \forall i\ \ i \in Q_0 \wedge i \in P_a \Rightarrow S(i) \in Q_0 \\ \cancel{\wedge\ i \in Q_0 \wedge i \in P_b \Rightarrow S(i) \in Q_0 \vee S(i) \in Q_1} \\ \wedge \neg (i \in Q_1 \wedge i \in P_a) \cancel{\Rightarrow \text{false}} \\ \wedge\ i \in Q_1 \wedge i \in P_b \Rightarrow S(i) \in Q_1 \end{cases}$$

Remark: we need only existential second order quantifications

Application : automatic structures.

Suppose $\langle E, \sigma \rangle$ is a structure where:

- there is an injection $J : E \to \Sigma^\omega$
- For all n-ary predicate $P \in \sigma$, $J(P)$ is an

ω - regular language on alphabet Σ^n where

$$\mathcal{I}(P) = \{ (u_0^{(0)}, \dots, u_0^{(n-1)}) \dots (u_k^{(0)} \dots u_k^{(n-1)}) \dots \mid$$

$$u^{(0)} = \mathcal{I}(e_0)$$

$$\vdots$$

$$^{(n-1)} = \mathcal{I}(e_{n-1})$$

$$P(e_0, \dots, e_{n-1}) \text{ is true} \}$$

- $\mathcal{I}(E)$ is an ω-regular language

Then, $\langle E, \sigma \rangle$ has a decidable theory