Artificial Intelligence

5. First-Order Logic

Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
Institute of Economics and Information Systems
& Institute of Computer Science
University of Hildesheim
http://www.ismll.uni-hildesheim.de

Artificial Intelligence

1. Introduction

2. Syntax

3. Semantics

4. Example

5. Conclusion

Lars Schmidt-Thieme, Information Systems and Machine Learning Lab (ISMLL), University of Hildesheim, Germany,
Course on Artificial Intelligence, summer term 2008 1/19

What is first-order logic?

Think about expressing these phrases in propositional logic:

$A :=$ "Socrates is human."
$B :=$ "All humans are mortal."
$C :=$ "Thus, Socrates is mortal."

How can we see that A, B, C are related?

First-order logic is richer than propositional logic:

$H(a)$
$\forall x H(x) \rightarrow M(x)$
$M(a)$

where a stands for "Socrates", H for "is human", and M for "is mortal".

Artificial Intelligence / 1. Introduction

What is first-order logic?

$H(a)$
$\forall x H(x) \rightarrow M(x)$
$M(a)$

So what do we have here?

- x is a **variable.** Variables denote arbitrary elements (objects) of an underlying set.
- a is a **constant**. Constants denote specific elements of an underlying set.
- H and M are *unary* **relations.**
- $\forall$ is the **all quantifier**. It is read "for all".
- We can also use the connectives we already know from propositional logic.

In first-order logic, there are also relations with other arities, as well as *n-ary* **functions**. In addition to the all quantifier, there is the **existential quantifier**, read "there exists".

1. Introduction

2. Syntax

3. Semantics

4. Example

5. Conclusion

Syntax: Symbols

- Let $\{f, g, h, \ldots, f_1, f_2, \ldots\}$ be the set of **function symbols**. Every function symbol has a given arity. Sometimes we write f^n to denote that f has arity n.
- Let $\{a, b, c, \ldots, a_1, a_2, \ldots\}$ be the set of **constant symbols**. Constant symbols can be seen as 0-ary function symbols.
- $\{P, R, S, \ldots, P_1, P_2, \ldots\}$ be the set of **relation symbols**. Every relation symbol (predicate) has a given arity. Sometimes we write P^n to denote that P has arity n.
- $\{x, y, z, x_1, x_2, \ldots\}$ be the set of **variable symbols**.

Syntax: Terms

A **term** is a logical expression that refers to an object.

(T1) Every variable or constant symbol is a term.

(T2) If f is an n-ary function symbol and $t_1, \ldots, t_n$ are terms, then $f(t_1, \ldots, t_n)$ is also a term.

Examples:

- a is a term, b as well.
- $f(a)$ is a term if f is unary.
- $f^3(a, x)$ is not a term.
- $P(x)$ and $P(x) \vee Q(x)$ are not terms.
- $f^1(f(f(a)))$ is a term.

More meaningful names for the symbols:

- $aristotle, socrates, kallias$
- $succ(root)$
- $Likes(zeno, hockey)$, $Likes(steffen, soccer) \wedge Likes(steffen, hockey)$
- $succ(succ(succ(0)))$

Syntax: Formulas

An **atomic formula** has the form $t_1 = t_2$ or $R(t_1, \ldots t_n)$ is an n-ary relation symbol and $t_1, \ldots, t_n$ are terms.

(F0) Every atomic formula is a **formula**.

(F1) If ϕ is a formula then so is $(\neg\phi)$.

(F2) If ϕ and ψ are formulas then so is $(\phi \wedge \psi)$.

(F3) If ϕ is a formula, then so is $(\exists x \phi)$ for any variable x.

We define $\vee$, $\rightarrow$, and $\leftrightarrow$ the same way as in propositional logic.
For any formula ϕ, $(\forall x \phi)$ and $(\neg\exists x \neg\phi)$ are interchangeable.

Unnecessary brackets can be left out as in propositional logic.
Precedence: $\neg, \exists, \forall, \wedge, \vee, \rightarrow, \leftrightarrow$

Examples:

- $P(x)$ and $P(x) \vee Q(x)$ are formulas if P and Q are unary.
- $succ(succ(succ(0))) = 3$ is a formula.
- $\forall y P(x, y)$ is a formula and $x(P(z)\exists)$ is not.

Syntax: Subformulas

Let ϕ be a formula of first-order logic. We inductively define what it means for θ to be a **subformula** of ϕ as follows:

- If ϕ is atomic, then θ is a subformula of ϕ if and only if $\theta = \phi$.
- If ϕ has the form $\neg\psi$, then θ is a subformula of ϕ if and only if $\theta = \phi$ or θ is a subformula of ψ.
- If ϕ has the form $\psi_1 \wedge \psi_2$, then θ is a subformula of ϕ if and only if $\theta = \phi$ or θ is a subformula of ψ_1, or θ is a subformula of ψ_2.
- If ϕ has the form $\exists x\psi$, then θ is a subformula of ϕ if and only if $\theta = \phi$ or θ is a subformula of ψ.

Syntax: Free variables

The **free variables** of a formula are those variables occurring in it that are not quantified.

Example: In $\forall y R(x, y)$, x is free, but y is **bound** by $\forall y$.

For any first-order formula ϕ, let $free(\phi)$ denote the set of free variables of ϕ. We define $free(\phi)$ inductively as follows:

- If ϕ is atomic, then $free(\phi)$ is the set of all variables occurring in ϕ,
- if $\phi = \neg\psi$, then $free(\phi) = free(\psi)$,
- if $\phi = \psi \wedge \theta$, then $free(\phi) = free(\psi) \cup free(\theta)$, and
- if $\phi = \exists x\psi$, then $free(\phi) = free(\psi) - \{x\}$.

How would you define the set of bound variables of ϕ, $bnd(\phi)$?

A **sentence** of first-order logic is a formula having no free variables.

1. Introduction

2. Syntax

3. Semantics

4. Example

5. Conclusion

Semantics: Vocabularies, structures and interpretations

A **vocabulary** is a set of function, relation, and constant symbols.

Let $\mathcal{V}$ be a vocabulary. A $\mathcal{V}$**-structure** $M = (U, I)$ consists of a nonempty underlying set U (the **universe**) along with an interpretation I of $\mathcal{V}$. An **interpretation** I of $\mathcal{V}$ assigns:

– an element of U to each constant symbol in $\mathcal{V}$,

– a function from U^n to U to each n-ary function in $\mathcal{V}$, and

– a subset of U^n to each n-ary relation in $\mathcal{V}$.

Examples:

– $\mathcal{V} = \{f^1, R^2, c\}$, $\mathbf{Z} = (\mathbb{Z}, I_{\mathbf{Z}})$
The universe is the set of integers $\mathbb{Z}$.
$I_{\mathbf{Z}}$ could interpret $f(x)$ as x^2, $R(x,y)$ as $x < y$, and c as 3.

– $\mathcal{V} = \{f^1, R^2, c\}$, $\mathbf{N} = (\mathbb{N}, I_{\mathbf{N}})$
The universe is the set of natural numbers $\mathbb{N}$.
$I_{\mathbf{N}}$ could interpret $f(x)$ as $x + 1$, $R(x,y)$ as $x < y$, and c as 0.

Semantics: $\mathcal{V}$-formulas and $\mathcal{V}$-sentences

Let $\mathcal{V}$ be a vocabulary. A **$\mathcal{V}$-formula** is formula in which every function, relation, and constant symbol is in $\mathcal{V}$. A **$\mathcal{V}$-sentence** is a $\mathcal{V}$-formula that is a sentence.

If M is a $\mathcal{V}$-structure, then each $\mathcal{V}$-sentence ϕ is either true or false in M. If ϕ is true in M, then we say M **models** ϕ and write $M \models \phi$.

Example: $\mathcal{V}_{ar} = \{+, \cdot, 0, 1\}$ is the vocabulary of arithmetic. Then $\mathbf{R} = (\mathbb{R}, I_\mathbf{R})$ is an $\mathcal{V}_{ar}$-structure if $I_\mathbf{R}$ is a interpretation of $\mathcal{V}_{ar}$.
$\mathbf{R} \models \forall x \exists y (1 + x \cdot x = y)$
What about $\forall y \exists x (1 + x \cdot x = y)$?

Semantics: The value of terms

We define the value $V_M(t) \in U$ of a term t inductively as

- $V_M(t) = I_M(t)$, if t is a constant symbol, and
- $V_M(t) = I_M(f)(V_M(t_1), \ldots, V_M(t_n))$, if $t = f^n(t_1, \ldots, t_n)$.

Example: $\mathcal{V} = \{f^1, R^2, c\}$, $\mathbf{N} = (\mathbb{N}, I_\mathbf{N})$, interpretation $I_\mathbf{N}$ as before
What is the value of the term $t = f(f(c))$?

$$\begin{aligned} V_\mathbf{N}(f(f(c))) &= I_\mathbf{N}(f)(V_\mathbf{N}(f(c))) \\ &= I_\mathbf{N}(f)(I_\mathbf{N}(f)(V_\mathbf{N}(c))) \\ &= I_\mathbf{N}(f)(I_\mathbf{N}(f)(I_\mathbf{N}(c))) \\ &= I_\mathbf{N}(f)(I_\mathbf{N}(f)(0)) \\ &= I_\mathbf{N}(f)(1) \\ &= 2 \end{aligned}$$

Semantics: Vocabulary/structure expansions and reducts

An **expansion** of a vocabulary $\mathcal{V}$ is a vocabulary containing $\mathcal{V}$ as a subset.

A structure M' is an expansion of the $\mathcal{V}$-structure M if M' has the same universe and interprets the symbols of $\mathcal{V}$ in the same way as M.

If M' is an expansion of M, then we say that M is a **reduct** of M'.

Examples:
The $\{+, -, \cdot, <, 0, 1\}$-structure $M' = (\mathbb{R}, I')$ is an expansion of the $\mathcal{V}_{ar}$-structure $M = (\mathbb{R}, I)$ if both I' and I interpret the symbols $+, \cdot, 0$, and 1 in the usual way.

A $\{+, -, \cdot, <, 0, 1\}$-structure $M'' = (\mathbb{Q}, I'')$ cannot be an expansion of M.

Any structure is an expansion of itself.

Semantics: The value of formulas

We define $M \models \phi$ by induction:

- $M \models t_1 = t_2$ if and only if $V(t_1) = V(t_2)$,
- $M \models R^n(t_1, \ldots, t_n)$ iff. $(V_M(t_1), \ldots, V_M(t_n)) \in I_M(R^n)$,
- $M \models \neg\phi$ iff. M does not model ϕ,
- $M \models \phi_1 \wedge \phi_2$ iff. both $M \models \phi_1$ and $M \models \phi_2$, and
- $M \models \exists x \phi(x)$ iff. $M_C \models \phi(c)$ for some constant $c \in \mathcal{V}(M)$.

$\mathcal{V}(M) = \mathcal{V} \cup \{c_m | m \in U_M\}$

$M_C = (U_M, I_C)$ is the expansion of $M = (U_M, I)$ to a $\mathcal{V}(M)$-structure where I_C interprets each c_m as the element m.

1. Introduction

2. Syntax

3. Semantics

4. Example

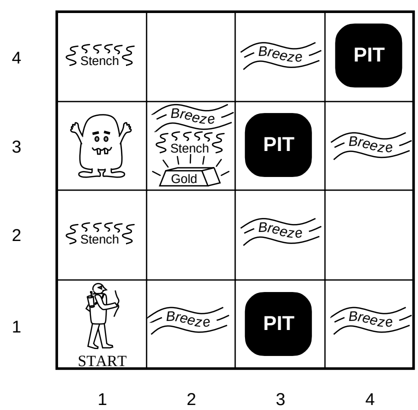
5. Conclusion

Back to the Silly Example

Toy example by Gregory Yob (1975), adapted by our textbook.

– 4×4 grid, tiles numbered (1,1) to (4,4),
– the agent starts in (1,1),
– the beast Wumpus sits at a random tile, unknown to the agent,
– a pile of gold sits at another random tile, unknown to the agent,
– some pits are located at random tiles, unknown to the agent.

– if the agent enters the tile of the Wumpus, he will be eaten,
– if the agent enters a pit, he will be trapped,



Encoding in propositional logic

64 variables:

$P_{x,y}$ tile x, y contains a pit ($x, y = 1, \ldots, 4$).
$W_{x,y}$ tile x, y contains the Wumpus ($x, y = 1, \ldots, 4$).
$B_{x,y}$ tile x, y contains a breeze ($x, y = 1, \ldots, 4$).
$S_{x,y}$ tile x, y contains stench ($x, y = 1, \ldots, 4$).

start is save: (2 formulas)

$$\neg P_{1,1}, \quad \neg W_{1,1}$$

how breeze arises: (16 formulas)

$$B_{x,y} \leftrightarrow P_{x-1,y} \vee P_{x+1,y} \vee P_{x,y-1} \vee P_{x,y+1}, \quad x, y = 1, \ldots, 4$$

how stench arises: (16 formulas)

$$S_{x,y} \leftrightarrow W_{x-1,y} \vee W_{x+1,y} \vee W_{x,y-1} \vee W_{x,y+1}, \quad x, y = 1, \ldots, 4$$

there is exactly one Wumpus: (121 formulas)

$$W_{1,1} \vee W_{1,2} \vee \ldots \vee W_{4,4}$$
$$\neg W_{x,y} \vee \neg W_{x',y'}, \quad x, y, x', y' = 1, \ldots, 4, x \neq x' \text{ or } y \neq y'$$

Encoding in first logic (1/2)

Vocabulary:

– constants $1, 2, 3, 4$
– binary relations symbols P, W, B, S

Meaning of the predicates:

$P(x, y)$ tile x, y contains a pit.
$W(x, y)$ tile x, y contains the Wumpus.
$B(x, y)$ tile x, y contains a breeze.
$S(x, y)$ tile x, y contains stench.

Encoding in first logic (2/2)

start is save:

$$\neg P(1,1) \wedge \neg W(1,1)$$

how breeze arises:

$$\forall x \forall y B(x,y) \leftrightarrow P(x-1,y) \vee P(x+1,y) \vee P(x,y-1) \vee P(x,y+1)$$

there is exactly one Wumpus:

$$W(x,y) \rightarrow \forall x' \forall y' W(x',y') \rightarrow (x = x' \wedge y = y')$$

Further possibilities: Encode actions as functions, encode time steps.

1. Introduction

2. Syntax

3. Semantics

4. Example

5. Conclusion

Summary

We introduced **first-order logic**, a representation language far more powerful than propositional logic.

– Knowledge representation should be declarative, compositional, expressive, context-independent, and unambiguous.
– **Constant symbols** name objects, **relation symbols** (predicates) name properties and relations, and **function symbols** name functions. Complex **terms** apply function symbols to terms to name an object.
– Given a $\mathcal{V}$-structure, the truth of a formula is determined.
– An **atomic formula** consists of a relation symbol applied to one or more terms; it is true iff. the relation named by the predicate holds between the objects named by the terms. **Complex formulas** use connectives just like propositional logic. **Quantifiers** allow the expression of general rules.

Outlook

Next lesson: **Inference** in first-order predicate logic.

Which other kind of logics exist?

– Temporal logic: $G\phi \rightarrow X\phi$
– Description logic: $C \subseteq D$
– Modal logic: $\Box p \rightarrow \Box\Box p$
– Higher-order predicate logic: $\forall P \forall x \forall y P(x, y) \wedge P(y, x) \rightarrow S(P)$
– Typed/intuitionistic/default/relevance logics
– . . .

(not covered in this course)

Literature

– Shawn Hedman: *A First Course in Logic*

– Heinz-Dieter Ebbinghaus, Jörg Flum, Wolfgang Thomas: *Einführung in die mathematische Logik*

– Uwe Schöning: *Logik für Informatiker*

– Stuart Russell, Peter Norvig: *Artificial Intelligence. A Modern Approach*