Logic for AI

Master 1 IFI

Andrea G. B. Tettamanzi
Nice Sophia Antipolis University
Computer Science Department
andrea.tettamanzi@unice.fr

Session 2

Predicate Logic

Agenda

- From Propositional to Predicate Logic
- Relational Logic (= Predicate Logic without Functions)
  - Syntax
  - Semantics
- Predicate Logic with Functions
  - Syntax
  - Herbrand Semantics (→ Herbrand Logic)
  - Tarskian Semantics (→ First-Order Logic)
- Deductive systems will be treated in the following lectures

From Propositional to Predicate Logic

- Propositional Logic
  - Premises:
    - If Jack knows Jill, then Jill knows Jack.
    - Jack knows Jill.
  - Conclusion:
    - Is it the case that Jill knows Jack?
- What about
  - Premises:
    - If one person knows another, then the second person knows the first.
    - Jack knows Jill
  - Conclusion:

Relational Logic

- We need to introduce new features
  - Variables
  - Quantifiers

- Sample sentence

$$\forall x \forall y (\mathsf{knows}(x, y) \Rightarrow \mathsf{knows}(y, x))$$

Relational Logic: Syntax

- Object constants (individuals): Joe, Nice, France, 0, 2345, 3.1415
- Relation constants (predicates): knows, loves, same
- Predicates have an arity:
  - Unary – 1 argument
  - Binary – 2 arguments
  - Ternary – 3 arguments
  - *n*-ary – *n* arguments
- Signature:
  - Set of object constants
  - Set of predicates together with a specification of their arity
- Variables: *x*, *y*, *z*, etc.

Terms and Sentences

- A term is a either a variable or an object constant
- Terms represent objects
- Terms are analogous to noun phrases in natural language
- Sentences
  - Relational sentences (atoms, ≈ simple propositions):
    - A predicate of arity *n* applied to *n* terms
  - Logical sentences (≈ complex propositions):
    - Combinations of sentences using logical operators
  - Quantified sentences:
    - Sentences with quantified variables

Definition (Sentence)

Sentence ::=

- A relation constant with arity *n* applied to *n* terms.
- $(\neg \varphi)$ where φ is a sentence.
- $(\varphi \vee \psi)$, where φ and ψ are sentences.
- $(\varphi \wedge \psi)$, where φ and ψ are sentences.
- $(\varphi \Rightarrow \psi)$, where φ and ψ are sentences.
- $(\varphi \Leftrightarrow \psi)$, where φ and ψ are sentences.
- $(\forall x\ \varphi)$, where φ is a sentence.
- $(\exists x\ \varphi)$, where φ is a sentence.

Only expressions produced by the above rules are sentences.

Some Nomenclature

- An **atom** is a relation constant with arity *n* applied to *n* terms
- A **literal** is either an atom or the negation of an atom
- A **ground** sentence has no variables or quantifiers
- A variable is **bound** if and only if it lies within the scope of a quantifier of that variable
- A variable is **free** if it is not bound
- A **closed** sentence has no free variables
- An **open** sentence does have free variables
- We treat free variables in an open sentence as being implicitly universally quantified.

Relational Logic: Semantics

- The classical semantics of relational logic is based on seminal work by Polish logician Alfred Tarski in the 1930s
- Tarski, "Pojęcie prawdy w językach nauk dedukcyjnych" 1933, → "Der Wahrheitsbegriff in den formalisierten Sprachen", 1935
- We will refer to his approach as “Tarskian semantics”

- An alternative approach stems from the work by French mathematician Jacques Herbrand (who died at age 23 while mountain-climbing on the Alps)
- We will refer to this approach as “Herbrand semantics”

Herbrand Base

- The **Herbrand base** for a Relational language is the set of all ground relational sentences that can be formed from the vocabulary of the language
- Example:
  - Object constants: a, b
  - Unary predicate: P
  - Binary predicate: R
  - Herbrand base: { P(a), P(b), R(a,a), R(a,b), R(b,a), R(b,b) }

Interpretation

- An interpretation is a mapping from the Herbrand base (i.e., the ground atoms) to the truth values {F, T}.
- We will use 1 as a synonym for T and 0 as a synonym for F.

$$\mathcal{I} : H \to \{0, 1\}$$

Example: let

$$H = \{P(a), P(b), R(a,a), R(a,b), R(b,a), R(b,b)\}$$

$$\begin{array}{ccc} P(a)^{\mathcal{I}} = 1 & P(b)^{\mathcal{I}} = 0 & R(a,a)^{\mathcal{I}} = 1 \\ R(a,b)^{\mathcal{I}} = 0 & R(b,a)^{\mathcal{I}} = 1 & R(b,b)^{\mathcal{I}} = 0 \end{array}$$

Equivalent view: interpretation as a subset of the Herbrand base

Sentential Interpretation

- A sentential interpretation is an extension of an interpretation mapping every sentence to the truth values 0 or 1.

$$\begin{aligned} P(a)^{\mathcal{I}} &= 1 \\ P(b)^{\mathcal{I}} &= 0 \end{aligned} \qquad\qquad \begin{aligned} (P(a) \vee P(b))^{\mathcal{I}} &= 1 \\ (\neg P(b) \wedge P(a))^{\mathcal{I}} &= 1 \end{aligned}$$

- Each interpretation is extended to a sentential interpretation based on the type of sentence.

Logical Sentences

$(\neg\phi)^{\mathcal{I}} = 1$	if and only if	$\phi^{\mathcal{I}} = 0$
$(\phi \wedge \psi)^{\mathcal{I}} = 1$	if and only if	$\phi^{\mathcal{I}} = 1$ and $\psi^{\mathcal{I}} = 1$
$(\phi \vee \psi)^{\mathcal{I}} = 1$	if and only if	$\phi^{\mathcal{I}} = 1$ or $\psi^{\mathcal{I}} = 1$
$(\phi \Rightarrow \psi)^{\mathcal{I}} = 1$	if and only if	$\phi^{\mathcal{I}} = 0$ or $\psi^{\mathcal{I}} = 1$
$(\phi \Leftrightarrow \psi)^{\mathcal{I}} = 1$	if and only if	$\phi^{\mathcal{I}} = \psi^{\mathcal{I}}$

Quantified Sentences

- A universally quantified sentence is true under an interpretation if and only if every instance of the scope of the quantified sentence is true under that interpretation.
- An existentially quantified sentence is true under an interpretation if and only if some instance of the scope of the quantified sentence is true under that interpretation.
- An interpretation satisfies a sentence with free variables if and only if it satisfies every instance of that sentence.

Alternative (Equivalent) View

- An interpretation (or model) M is a subset of H
- $\models_M P(t_1,\ldots,t_n)$ if and only if $P(t_1,\ldots,t_n) \in M$.
- $\models_M \neg\psi$ if and only if $\not\models_M \psi$.
- $\models_M \varphi \wedge \psi$ if and only if $\models_M \varphi$ and $\models_M \psi$.
- $\models_M \varphi \vee \psi$ if and only if $\models_M \varphi$ or $\models_M \psi$.
- $\models_M \varphi \Rightarrow \psi$ if and only if $\not\models_M \varphi$ or $\models_M \psi$.
- $\models_M \varphi \Leftarrow \psi$ if and only if $\models_M \psi \Rightarrow \varphi$.
- $\models_M \varphi \Leftrightarrow \psi$ if and only if either $\models_M \varphi\wedge\psi$ or $\models_M \neg\varphi \wedge \neg\psi$.
- $\models_M \forall x.\varphi(x)$ if and only if $\models_M \varphi(t)$ for all ground terms t.
- $\models_M \exists x.\varphi(x)$ if and only if $\models_M \varphi(t)$ for some ground term t.

Quantificational Tautologies

Common Quantifier Reversal:

$$\forall x \forall y P(x,y) \Leftrightarrow \forall y \forall x P(x,y)$$
$$\exists x \exists y P(x,y) \Leftrightarrow \exists y \exists x P(x,y)$$

Existential Distribution:

$$\exists x \forall y P(x,y) \Leftrightarrow \forall y \exists x P(x,y)$$

Negation Distribution:

$$\neg \forall x P(x) \Leftrightarrow \exists x \neg P(x)$$

(Herbrand) Entailment

- Let Δ be a set of closed sentences and V a vocabulary that is a superset of the vocabulary of Δ.
- Let φ be a closed sentence.
- Δ entails φ with respect to vocabulary V if and only if every Herbrand model for V that satisfies Δ also satisfies φ.

  $\Delta \models \varphi$ wrt V if and only if $\models_M \Delta$ implies $\models_M \varphi$,

  where M is a Herbrand model for V
- If no vocabulary is named in satisfaction or entailment, it is assumed the minimal vocabulary is used, i.e. the vocabulary that includes just the constants in the sentences given.

Mapping Relational Sentences to Propositional Sentences

There is a simple procedure for mapping Relational Logic sentences to equivalent Propositional Logic sentences.

1) Convert to Prenex form.
2) Compute the grounding.
3) Consider each ground atom as a propositional constant.

Since satisfiability and entailment in Propositional Logic are decidable, then satisfiability and entailment in Relational Logic are decidable too.

Prenex Form

- A sentence is in prenex form if and only if
  1) it is closed and
  2) all of the quantifiers are outside of all logical operators.

- Sentence in Prenex Form:
  - $\forall x.\exists y.\forall z.(p(x,y) \lor q(z))$
- Sentences not in Prenex Form:
  - $\forall x.\exists y.p(x,y) \lor \exists y.q(y)$
  - $\forall x.(p(x,y) \lor q(x))$

Conversion to Prenex Form

1) Rename duplicate variables.
   - $\forall y.p(x,y) \vee \exists y.q(y) \rightarrow \forall y.p(x,y) \vee \exists z.q(z)$
2) Distribute logical operators over quantifiers.
   - $\forall y.p(x,y) \vee \exists z.q(z) \rightarrow \forall y.\exists z.(p(x,y) \vee q(z))$
3) Quantify any free variables.
   - $\forall y.\exists z.(p(x,y) \vee q(z)) \rightarrow \forall x.\forall y.\exists z.(p(x,y) \vee q(z))$

Substitution

- A **substitution** is a finite set of the form

$$\{t_1/v_1, \ldots, t_n/v_n\}$$

  where
  - Every v_i is a variable
  - Every t_i is a term different from v_i
  - All variables v_i are different
- When $t_1, \ldots, t_n$ are ground terms, the substitution is called a **ground substitution**.
- We denote by $\varphi[\theta]$ the application of substitution θ to sentence φ

Grounding

- Replace each universally quantified sentence with the set of its instances

$$\forall x \phi \rightsquigarrow \{\phi[c/x] \mid c \text{ is an object constant}\}$$

- Replace each existentially quantified sentence with the disjunction of its instances

$$\exists x \phi \rightsquigarrow \bigvee_{c} \phi[c/x]$$

- Until all sentences are quantifier-free (and, therefore, ground)

Compactness

- A logic is **compact** if and only if every unsatisfiable set of sentences (including infinite sets) has a finite subset that is unsatisfiable.
- Propositional Logic is compact.
- Given our mapping, we know that Relational Logic must also be compact.

Predicate Logic (with Functions)

- The syntax is an extension of the syntax of Relational Logic
- Function constants with their arity: f(.), g(., .), etc.
- The definition of a term becomes:
  - A variable
  - An object constant
  - A function constant with arity *n* applied to *n* terms.
- Only expressions produced by the the above rules are terms.

- As a result, the set of terms will be infinite, even though the vocabulary of the language is finite.

Avoiding Constants and Functions

- It is possible to entirely avoid function symbols and constant symbols, rewriting them via predicate symbols in an appropriate way.
- For example, instead of using a constant symbol 0 one may use a predicate $0(x)$, interpreted as $x = 0$, and replace every predicate such as $P(0, y)$ with $\forall x(0(x) \Rightarrow P(x, y))$.
- A function such as $f(x_1, x_2, \ldots, x_n)$ will similarly be replaced by a predicate $F(x_1, x_2, \ldots, x_n, y)$ interpreted as $y = f(x_1, x_2, \ldots, x_n)$.
- This change comes at a cost: additional axioms must be added to the theory at hand, so that interpretations of the predicate symbols used have the correct semantics.

Herbrand Semantics

- Same definition as for the Relational Logic but...
- In the presence of functions, the Herbrand base is infinite!
- However, every interpretation (or model) M, as subset of H, is a finite set of atoms.

Undecidability

- Satisfiability and logical entailment for Herbrand Logic are undecidable.
- Proof sketch:
  - We can reduce a problem that is generally accepted to be non-semidecidable to a question of satisfiability / logical entailment in Herbrand Logic
  - If Herbrand logic were semidecidable, then such question would be semidecidable as well
  - Since it is known not to be semidecidable, then Herbrand Logic must not be semidecidable either.

Tarskian Semantics

- Herbrand logic differs from first-order logic solely in the structures it considers to be models.
- The semantics of a given set of sentences is defined to be the set of Herbrand models that satisfy it, for a given vocabulary.
- In Tarskian semantics, we map all the elements of the language to the element of an (external) domain D
- A first-order model M consists of a domain D and a mapping $\cdot^M$ such that
  - For each *n*-ary predicate *P* an *n*-ary relation P^M over D
  - For each *n*-ary function constant *f* an *n*-ary function f^M over D
  - For each object constant *c* an element c^M from D

First-Order Model

- In a model M, a variable assignment is a mapping of all the variables in the vocabulary to elements in D.
- Given an arbitrary model and a variable assignment for that model, every term in the language is assigned an element in that model's universe:
  - Let v be a variable assignment and M a first-order model
  - e_v maps a term to an element of D.
    - For variable x, $e_v(x) = v(x)$
    - For object constant c, $e_v(c) = c^M$
    - For terms $t_1, \ldots, t_n$, $e_v(f(t_1, \ldots, t_n)) = f^M(e_v(t_1),\ldots,e_v(t_n))$

First-Order Satisfaction

Given M and v, $\models_M \varphi$ is defined as follows:

- $\models_M P(t_1,\ldots,t_n)[v]$ if and only if $\langle e_v(t_1),\ldots,e_v(t_n)\rangle \in P^M$
- $\models_M \neg\, \psi[v]$ if and only if $\not\models_M \psi[v]$
- $\models_M (\varphi \wedge \psi)[v]$ if and only if $\models_M \varphi[v]$ and $\models_M \psi[v]$
- $\models_M (\varphi \vee \psi)[v]$ if and only if $\models_M \varphi[v]$ or $\models_M \psi[v]$
- $\models_M (\varphi \Rightarrow \psi)[v]$ if and only if $\not\models_M \varphi[v]$ or $\models_M \psi[v]$
- $\models_M (\varphi \Leftrightarrow \psi)[v]$ if and only if either $\models_M (\varphi\wedge\psi)[v]$ or $\models_M (\neg\varphi \wedge \neg\psi)[v]$
- $\models_M \forall x.\varphi[v]$ if and only if for every $d \in D$ $\models_M \varphi[v][d/x]$
- $\models_M \exists x.\varphi[v]$ if and only if for some $d \in D$ $\models_M \varphi[v][d/x]$

Comparison of the Two Semantics

- Given vocabulary $\{P(\cdot), a, b\}$,
- Sentence $P(a)$:
  - Has exactly 2 Herbrand models:
    - $\{ P(a) \}$
    - $\{ P(a), P(b) \}$
  - Has infinitely many First-Order models:
    - $D = \{1\}$, $P^M = \{\langle 1 \rangle\}$, $a^M = 1$, $b^M = 1$,
    - $D = \{1, 2, 3, \ldots\}$, $P^M = \{\langle 17 \rangle, \langle 63 \rangle\}$, $a^M = 17$, $b^M = 51$,
    - D = Reals, $P^M = \{\langle 3.14159... \rangle, \langle 17.0 \rangle\}$,
      $a^M = 3.14159...$, $b^M = 0.33333...$
    - …

Comparison of the Two Semantics

- Given vocabulary $\{P(\cdot), a\}$,
- Sentences $P(a)$, $\exists x.\neg P(x)$:
  - Are Herbrand-unsatisfiable
  - Are always satisfiable in First-Order Logic:
    - $D = \{1, 2\}$, $P^M = \{\langle 1 \rangle\}$, $a^M = 1$
    - …
- We have to extend the vocabulary to $\{P(\cdot), a, b\}$ for them to be Herbrand-satisfiable:
    - $M = \{ P(a) \}$

Skolem Standard Form

We can obtain the Skolem Standard form of a sentence by applying the following procedure:

1) Transform the sentence into prenex normal form
2) Transform the matrix of the prenex normal form into CNF
3) Eliminate the existential quantifiers in the prefix by using Skolem functions:
    1) For each quantifier $\exists x$ in the prefix, let m be the number of $\forall$'s preceding it;
    2) Replace every occurrence of x in the matrix with the term $s_x(x_1, \ldots, x_m)$, where s_x is a new function constant of arity m and $x_1, \ldots, x_m$ are the universally quantified variables occurring before x in the prefix.

Semantic Trees

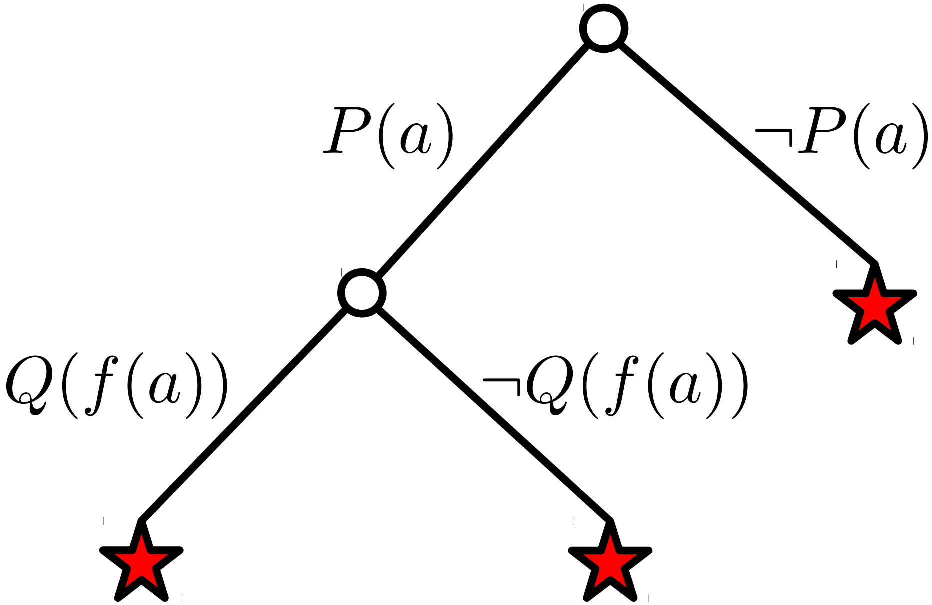
- Checking the Herbrand-satisfiability of a set of clauses (obtained from the matrix of a Skolem Standard Form) can be done by constructing a semantic tree
- Given a set S of clauses, a **semantic tree** for S is a tree where each edge is labeled with a finite set of literals of atoms of S in such a way that
  - The disjunction of all the labels of the outgoing edges of a node is a tautology
  - The labels on the path from the root to node N constitute a partial interpretation I(N).
- A semantic tree is **complete** iff for every leaf N, I(N) contains either A or ¬A for every atom in S

Semantic Trees (continued)

- A node N is a **failure node** if I(N) falsifies some ground instance of a clause in S, but I(N’) does not falsify any ground instance of a clause in S for every ancestor N’ of N.
- A semantic tree is **closed** if and only if every branch terminates at a failure node
- A node N of a closed semantic tree is an **inference node** if all the children of N are failure nodes.

Example

$$\{P(x), \neg P(x) \lor Q(f(x)), \neg Q(f(a))\}$$



Herbrand's Theorem

A set S of clauses is unsatisfiable if and only if corresponding to every complete semantic tree of S, there is a finite closed semantic tree

Proof:

- $[\Rightarrow]$: Suppose that S is unsatisfiable. Then for every path in a complete semantic tree of S, there must be a failure node at a finite depth.
- $[\Leftarrow]$: If corresponding to every complete semantic tree of S there is a finite closed semantic tree, then every branch contains a failure node. This means that every interpretation falsifies S. Hence S is unsatisfiable.

Thank you for your attention