First-order logic
Second part:
Interpretation of a formula

Benjamin Wack

Université Grenoble Alpes

March 2022
A few examples

Formalize in first-order logic:

- Some people love each other.
  \[ \exists x \exists y (\ell(x, y) \land \ell(y, x)) \]

- If two people are in love, then they are spouses.
  \[ \forall x \forall y (\ell(x, y) \land \ell(y, x) \Rightarrow s(x) = y \land s(y) = x) \]

- No one can love two distinct persons.
  \[ \forall x \forall y (\ell(x, y) \Rightarrow \forall z (\ell(x, z) \Rightarrow y = z)) \]
  \[ \forall x \forall y \forall z (\ell(x, y) \land \ell(x, z) \Rightarrow y = z) \]
Overview

Truth value of formulae

Finite interpretation by expansion (continued)

Interpretation and substitution

Important equivalences

Conclusion
Overview

Truth value of formulae

Finite interpretation by expansion (continued)

Interpretation and substitution

Important equivalences

Conclusion
Reminder: Interpretation and state

Definition 4.3.16
An interpretation $I$ over a signature $\Sigma$ is defined by:

- a non-empty domain $D$
- every symbol $s^{gn}$ is mapped to its value as follows:
  - (constant) $s^I_{f0}$ is an element of $D$
  - (function) $s^I_{fn}$ is a function from $D^n \rightarrow D$
  - (propositional variable) $s^I_{r0}$ is either 0 or 1
  - (relation) $s^I_{rn}$ is a set of $n$-uples in $D$

Definition 4.3.21
A state $e$ maps each variable to an element in the domain $D$. 
Remark 4.3.24

- For a formula with free variables, we need an assignment \((I, e)\) with a state \(e\).
- For a formula with no free variables, simply give an interpretation \(I\) of the symbols of the formula.

Indeed, \((I, e)\) and \((I, e')\) will yield the same value for any formula: thus, we will identify \((I, e)\) and \(I\).
Terms

Definition 4.3.25 Evaluation

The evaluation of a term \( t \) is inductively defined as:

1. if \( t \) is a variable, then \( \llbracket t \rrbracket_{(l,e)} = e(t) \),
2. if \( t \) is a constant, then \( \llbracket t \rrbracket_{(l,e)} = t^f_0 \),
3. if \( t = s(t_1, \ldots, t_n) \) where \( s \) is a function symbol, then \( \llbracket t \rrbracket_{(l,e)} = s^f_n(\llbracket t_1 \rrbracket_{(l,e)}, \ldots, \llbracket t_n \rrbracket_{(l,e)}) \)
Example 4.3.26

Let the signature be $a^{f_0}, f^{f_2}, g^{f_2}$.
Let $I$ be the interpretation of domain $\mathbb{N}$ which maps:

- $a$ to the integer 1;
- $f$ to the product;
- $g$ to the sum.

Let $e$ be the state such that $e(x) = 2$ and $e(y) = 3$.
Let us compute $\llbracket f(x, g(y, a)) \rrbracket_{(I,e)}$.

\[
\llbracket f(x, g(y, a)) \rrbracket_{(I,e)} = \llbracket x \rrbracket_{(I,e)} \times \llbracket g(y, a) \rrbracket_{(I,e)} \\
= \llbracket x \rrbracket_{(I,e)} \times (\llbracket y \rrbracket_{(I,e)} + \llbracket a \rrbracket_{(I,e)}) \\
= e(x) \times (e(y) + 1) \\
= 2 \times (3 + 1) = 8
\]
Definition 4.3.27 Truth value of an atomic formula

The truth value of an atomic formula is given by the following inductive rules:

1. $[\top](I,e) = 1$ and $[\bot](I,e) = 0$.

2. Let $s$ be a propositional variable, $[s](I,e) = s^r_I$

3. Let $A = s(t_1, \ldots, t_n)$ where $s$ is a relation symbol:
   - if $([t_1](I,e), \ldots, [t_n](I,e)) \in s^r_{fn}$ then $[A](I,e) = 1$
   - otherwise $[A](I,e) = 0$
Example 4.3.19

Let us consider the following signature:

- $Anne^0$, $Bernard^0$ and $Claude^0$: constants
- $\ell^{r^2}$: a binary relation (we read $\ell(x, y)$ as “$x$ loves $y$”)
- $s^{f^1}$: a unary function (we read $s(x)$ as the spouse of $x$).

A possible interpretation over this signature is the interpretation $I$ of domain $D = \{0, 1, 2\}$ where:

- $Anne_i^0 = 0$, $Bernard_i^0 = 1$, and $Claude_i^0 = 2$.
- $\ell_i^{r^2} = \{(0, 1), (1, 0), (2, 0)\}$.
- $s_i^{f^1}$ is a function from $D$ to $D$ defined as

<table>
<thead>
<tr>
<th>$x$</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i^{f^1}(x)$</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>
Example 4.3.29

We obtain:

- $[\ell(\text{Anne}, \text{Bernard})]_I =$
  
  $\text{true}$ since $(\llbracket \text{Anne} \rrbracket_I, \llbracket \text{Bernard} \rrbracket_I) = (0, 1) \in \ell^{r_2}_I$.

- $[\ell(\text{Anne}, \text{Claude})]_I =$
  
  $\text{false}$ since $(\llbracket \text{Anne} \rrbracket_I, \llbracket \text{Claude} \rrbracket_I) = (0, 2) \notin \ell^{r_2}_I$. 
Example 4.3.29

Let $e$ be the state $x = 0, y = 2$. We have:

- $\llbracket (x, s(x)) \rrbracket_{(I,e)} = \text{true}$ since $(\llbracket x \rrbracket_{(I,e)}, \llbracket s(x) \rrbracket_{(I,e)}) = (0, s_{I}^{f_1}(0)) = (0, 1) \in \ell_{I}^{r_2}$.

- $\llbracket (y, s(y)) \rrbracket_{(I,e)} = \text{false}$ since $(\llbracket y \rrbracket_{(I,e)}, \llbracket s(y) \rrbracket_{(I,e)}) = (2, s_{I}^{f_1}(2)) = (2, 2) \not\in \ell_{I}^{r_2}$.

Here, we have used \textit{true} and \textit{false} instead of the truth values 0 and 1 in order to distinguish them from the elements 0 and 1 of the domain (beware of the ambiguity, use the context).
Example 4.3.29

We have:

- $[(Anne = Bernard)]_I = false$, since $(\llbracket Anne \rrbracket_I, \llbracket Bernard \rrbracket_I) = (0, 1)$ and $(0, 1) \notin r^2_I$.

- $[(s(Anne) = Anne)]_I = false$, since $(\llbracket s(Anne) \rrbracket_I, \llbracket Anne \rrbracket_I) = (s^f_1(0), 0) = (1, 0)$.

- $[(s(s(Anne)) = Anne)]_I = true$, since $(\llbracket s(s(Anne)) \rrbracket_I, \llbracket Anne \rrbracket_I) = (s^f_1(s^f_1(0)), 0) = (0, 0)$ and $(0, 0) \in r^2_I$. 
Truth value of a formula 4.3.30

1. Propositional connectives have the same meaning as in propositional logic.

2. Let $e[x = d]$ be the state that is identical to $e$, except for $x$.

$$\forall x B(I, e) = \min_{d \in D} [B](I, e[x = d]) = \prod_{d \in D} [B](I, e[x = d]),$$

* i.e. it is true if and only if $[B](I, f) = 1$ for every state $f$ identical to $e$, except for $x$.

3. 

$$\exists x B(I, e) = \max_{d \in D} [B](I, e[x = d]) = \sum_{d \in D} [B](I, e[x = d]),$$

* i.e. it is true if there is a state $f$ identical to $e$, except for $x$, such that $[B](I, f) = 1$. 
Example 4.3.32

Let us use the interpretation $I$ given in example 4.3.19.
(Reminder $D = \{0, 1, 2\}$)

$\exists x \ell(x, x) \ |$ 

\[
= \text{max}\{\ell(0, 0), \ell(1, 1), \ell(2, 2)\} = \text{false}
\]

\[
= \ell(0, 0) + \ell(1, 1) + \ell(2, 2) = \text{false} + \text{false} + \text{false} = \text{false}.
\]

$\forall x \exists y \ell(x, y) \ |$ 

\[
= \text{min}\{\text{max}\{\ell(0, 0), \ell(0, 1), \ell(0, 2)\},\]
\[
\text{max}\{\ell(1, 0), \ell(1, 1), \ell(1, 2)\},\]
\[
\text{max}\{\ell(2, 0), \ell(2, 1), \ell(2, 2)\}\}\}
\]

\[
= \text{min}\{\text{max}\{\text{false, true, false}\}, \text{max}\{\text{true, false, false}\},\]
\[
\text{max}\{\text{true, false, false}\}\}
\]

\[
= \text{min}\{\text{true, true, true} = \text{true}.
\]
Example 4.3.32

\[ \exists y \forall x \ell(x, y)] \models \]

\[ = [\ell(0, 0)] \models [\ell(1, 0)] \models [\ell(2, 0)] \models + [\ell(0, 1)] \models [\ell(1, 1)] \models [\ell(2, 1)] \models \\
+ [\ell(0, 2)] \models [\ell(1, 2)] \models [\ell(2, 2)] \models \\
= \text{false} \cdot \text{true} \cdot \text{true} + \text{true} \cdot \text{false} \cdot \text{false} + \text{false} \cdot \text{false} \cdot \text{false} \\
= \text{false} + \text{false} + \text{false} = \text{false}. \]

Remark 4.3.33

The formulae \( \forall x \exists y \ell(x, y) \) and \( \exists y \forall x \ell(x, y) \) do not have the same value. Exchanging a \( \exists \) and a \( \forall \) does not preserve the truth value of a formula.
Model, validity, consequence, equivalence

Defined as in propositional logic but...

What’s needed to evaluate a formula

- **In propositional logic**: an assignment $V \rightarrow \{0, 1\}$
- **In first-order logic**: $(I, e)$ where
  - $I$ is a symbol interpretation
  - $e$ a variable state.

... we use an interpretation instead of an assignment.

The truth value of a formula only depends on

- the state of its free variables
- and the interpretation of its symbols.
Overview

Truth value of formulae

Finite interpretation by expansion (continued)

Interpretation and substitution

Important equivalences

Conclusion
Reminders about finite expansions

We look for models with $n$ elements by reduction to the propositional case.

**Base case:** a formula with no function symbol and no constant, except integers less than $n$.

**Building the $n$-elements model**

1. eliminate the quantifiers by expansion over a domain of $n$ elements,
2. replace equalities with their value
3. search for a propositional assignment of atomic formulae which is a model of the formula.
Property of the $n$-expansion

Theorem 4.3.41

Let $A$ be a formula containing only integers $< n$. Let $B$ be the $n$-expansion of $A$. Any interpretation over the domain $\{0, \ldots, n-1\}$ assigns the same value to $A$ and $B$.

Proof: by induction on the height of formulae.
Assignment VS interpretation

Let $A$ be a formula:
- closed,
- with no quantifier,
- with no equality nor function symbol,
- with no constant except the integers less than $n$.

Let $P$ be the set of atomic formulae in $A$ (except $\top$ and $\bot$).

**Theorem 4.3.42**

For any propositional assignment $\nu : P \rightarrow \{\text{false, true}\}$ there exists an interpretation $I$ of $A$ such that $[A]_I = [A]_\nu$.

**Theorem 4.3.44**

For any interpretation $I$ there exists an assignment $\nu : P \rightarrow \{\text{false, true}\}$ such that $[A]_I = [A]_\nu$. 
Example 4.3.43

Let \( \nu \) be the assignment defined by \([p(0)]_\nu = true\) and \([p(1)]_\nu = false\).

\( \nu \) gives the value \( false \) to the formula \((p(0) + p(1)) \Rightarrow (p(0) \cdot p(1))\).

The interpretation \( I \) defined by \( p_I = \{0\} \) gives the same value to the same formulae.

This example shows that \( \nu \) and \( I \) are two analogous ways of presenting an interpretation.
Correctness of the method

<table>
<thead>
<tr>
<th>$A$</th>
<th>$\rightarrow$</th>
<th>$B$</th>
<th>$\rightarrow$</th>
<th>$C$</th>
<th>$\rightarrow$</th>
<th>$C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1st order)</td>
<td>$\equiv_n$</td>
<td>(without $\forall \exists$)</td>
<td>$\equiv$</td>
<td>(without const.)</td>
<td>$\equiv$</td>
<td>(propos.)</td>
</tr>
</tbody>
</table>

- $[A]_I = [B]_I$ for any $I$ over a domain of $n$ elements
- $B \equiv C$ by construction (hence $[B]_I = [C]_I$ for any $I$)
- For any $\nu$ there is an $I$ such that $[C]_I = [C]_\nu$.
- For any $I$ there is a $\nu$ such that $[C]_I = [C]_\nu$.

Thus $A$ has a model $I$ over a domain of $n$ elements if and only if $C$ has a model $\nu$ (and we can find $I$ from $\nu$ if need be).
Finding a finite model of a closed formula \textbf{with} a function symbol

Let $A$ be a closed formula which can contain integers of value less than $n$.

**Procedure**

- Replace $A$ by its expansion
- Enumerate the choices of symbol values, by propagating as much as possible each of the realized choices.

Similar to \texttt{DPLL} algorithm.
Example 4.3.46: $A = \exists y P(y) \Rightarrow P(a)$

Look for a counter-model with 2 elements.

2-expansion of $A$

$$P(0) + P(1) \Rightarrow P(a)$$

Find the values of $P(0)$, $P(1)$, $a$. We (arbitrarily) choose $a = 0$.

$$P(0) + P(1) \Rightarrow P(0)$$

$P(0) \mapsto false$, $P(1) \mapsto true$ is a propositional counter-model, we deduce an interpretation $I$ such that $P_I = \{1\}$.

A counter-model is $I$ over domain $\{0, 1\}$ such that $P_I = \{1\}$ and $a_I = 0$. 
Example 4.3.47: $P(a), \forall x(P(x) \Rightarrow P(f(x))), \neg P(f(b))$

1. 2-expansion:
   
   $F = \{ P(a), (P(0) \Rightarrow P(f(0))), (P(1) \Rightarrow P(f(1))), \neg P(f(b)) \}$. 

2. Find values for $P(0), P(1), a, b, f(0)$ and $f(1)$ which provide a model of $F$.

3. Let us choose $a = 0$
   
   - From $P(a) = true$ and $a = 0$, we deduce: $P(0) = true$
   - From $P(0) = true$ and $(P(0) \Rightarrow P(f(0))) = true$, we deduce: $P(f(0)) = true$
   - From $P(f(b)) = false$ and $P(f(0)) = true$, we deduce $f(0) \neq f(b)$ therefore $b \neq 0$, hence: $b = 1$ and $P(f(1)) = false$.
   - From $P(f(1)) = false$ and $P(0) = true$, we deduce $f(1) \neq 0$ hence: $f(1) = 1$ and $P(1) = false$
   - From $P(f(0)) = true$ and $P(1) = false$, we deduce: $f(0) = 0$

4. Model: $a = 0, b = 1, P = \{0\}, f(0) = 0, f(1) = 1$
William McCune (1953-2011)

- Author of several automated reasoning systems: Otter, Prover9, Mace4

**MACE**

- expansion of first-order formulas
- efficient algorithms such as DPLL


- 1996: Proof of the Robbins conjecture using the automated theorem prover EQP
  - 8 days of computation on a 66 MHz processor, 30 Mo of memory
  - production of a proof witness by Otter, in turn checked by a third program

(Undecided conjecture since 1933)
Overview

Truth value of formulae

Finite interpretation by expansion (continued)

Interpretation and substitution

Important equivalences

Conclusion
Substitution at the **propositional** level

Recall that substituting a **propositional** variable in a valid formula yields another valid formula. This extends to first-order logic.

**Example:**

Let $\sigma(p) = \forall x \ q(x)$. $p \lor \neg p$ is valid, the same holds for

$$\sigma(p \lor \neg p) = \forall x \ q(x) \lor \neg \forall x \ q(x)$$

The **replacement** principle extends to first-order logic as well since:

For any formulae $A$ and $B$ and any variable $x$:

- $(A \iff B) \models (\forall x A \iff \forall x B)$
- $(A \iff B) \models (\exists x A \iff \exists x B)$
Instantiation of a variable in a term

Definition 4.3.34

$A < x ::= t >$ is the formula obtained by replacing in $A$ every free occurrence of $x$ with the term $t$.

Example 4.3.35

Let $A$ be the formula $(\forall x P(x) \lor Q(x))$, the formula $A < x ::= b >$ is $(\forall x P(x) \lor Q(b))$ since only the bold occurrence of $x$ is free.

But one cannot substitute any variable with anything:

Example 4.3.37

Let $A$ be the formula $\exists y p(x, y)$.

$A < x ::= y > = \exists y p(y, y)$ (capture phenomenon)
Capture changes the meaning of a formula

Example 4.3.37

Let $p$ be a binary relation interpreted over $\{0, 1\}$ as $p_I = \{(0, 1)\}$. Let $e$ be a state where $y = 0$.

- $[A < x := y >]_{(I,e)} = $ \[
\exists y p(y, y)_{(I,e)} = [p(0, 0)]_{(I,e)} + [p(1, 1)]_{(I,e)} = false + false = false.
\]

- Let $d = 0$.
  In the assignment $(I, e[x = d])$, we have $x = 0$.
  Hence $[A]_{(I, e[x=d])} = $ \[
\exists y p(x, y)_{(I,e[x=d])} = [p(0, 0)]_{(I,e)} + [p(0, 1)]_{(I,e)} = false + true = true.
\]

Thus, $[A < x := y >]_{(I,e)} \neq [A]_{(I, e[x=d])}$, for $d = [y]_{(I,e)}$. 
Precautions for the instantiation of a variable in a term

Solution: notion of a term $t$ free for a variable

Definition 4.3.34

2. The term $t$ is free for $x$ in $A$ if the variables of $t$ are not bound in the free occurrences of $x$.

Example 4.3.35

- The term $f(z)$ is free for $x$ in formula $\exists y \, p(x, y)$.
- On the opposite, the terms $y$ or $g(y)$ are not free for $x$ in this formula.
- By definition, the term $x$ is free for $x$ in any formula.
Properties

Theorem 4.3.36
Let $A$ be a formula and $t$ a free term for the variable $x$ in $A$. For any assignment $(I,e)$ we have
$$[A < x := t >]_{(I,e)} = [A]_{(I,e[x=d])} \quad \text{where } d = [t]_{(I,e)}.$$

Corollary 4.3.38
Let $A$ be a formula and $t$ a free term for $x$ in $A$. The formulae $\forall x A \Rightarrow A < x := t >$ and $A < x := t > \Rightarrow \exists x A$ are valid.
Overview

Truth value of formulae

Finite interpretation by expansion (continued)

Interpretation and substitution

Important equivalences

Conclusion
Relation between $\forall$ and $\exists$

Lemma 4.4.1
Let $A$ be a formula and $x$ be a variable.

1. $\neg \forall x A \equiv \exists x \neg A$
2. $\forall x A \equiv \neg \exists x \neg A$
3. $\neg \exists x A \equiv \forall x \neg A$
4. $\exists x A \equiv \neg \forall x \neg A$

Let us prove the first two equivalences, the other are in exercise 78
Proof of \( \neg \forall x A \equiv \exists x \neg A \)

Let us evaluate \([\neg \forall x A](l,e)\):

\[
\begin{align*}
    & \neg \forall x A(l,e) \\
    & = 1 - \forall x A(l,e) \\
    & = 1 - \min_{d \in D} [A](l,e[x=d]) \\
    & = \max_{d \in D} \left(1 - [A](l,e[x=d])\right) \\
    & = \max_{d \in D} [\neg A](l,e[x=d]) \\
    & = [\exists x \neg A](l,e)
\end{align*}
\]

Proof of \( \forall x A \equiv \neg \exists x \neg A \):

Let us evaluate \( \forall x A \):

\[
\begin{align*}
    & \forall x A \\
    & \equiv \neg \neg \forall x A \\
    & \equiv \neg \exists x \neg A \quad \text{(see above)}
\end{align*}
\]
Moving quantifiers

Let $x, y$ be two variables and $A, B$ be two formulae.

1. $\forall x \forall y A \equiv \forall y \forall x A$
2. $\exists x \exists y A \equiv \exists y \exists x A$
3. $\forall x (A \land B) \equiv (\forall x A \land \forall x B)$
4. $\exists x (A \lor B) \equiv (\exists x A \lor \exists x B)$

5. Let $Q$ be a quantifier and let $\circ$ be a connective among $\land, \lor$.
   If $x$ is not a free variable of $A$ then:
   5.1 $Qx A \equiv A$,
   5.2 $Qx (A \circ B) \equiv A \circ QxB$
Example 4.4.2

Let us eliminate useless quantifiers from these two formulae:

\[
\forall x \exists x P(x) \equiv \\
\exists x P(x)
\]

\[
\forall x (\exists x P(x) \lor Q(x)) \equiv \\
\exists x P(x) \lor \forall x Q(x)
\]
Renaming of bound variables (1/3)

Theorem 4.4.3

Let $Q$ be a quantifier. If $y$ does not occur in $Qx A$ then:

$Qx A \equiv Qy A \quad \langle x := y \rangle.$

Example 4.4.4

- $\forall x \ p(x, z) \equiv \forall y \ p(y, z)$
- $\forall x \ p(x, z) \not\equiv \forall z \ p(z, z)$
Overview

Truth value of formulae

Finite interpretation by expansion (continued)

Interpretation and substitution

Important equivalences

Conclusion
Today

- To evaluate a formula = to choose an interpretation for its **symbols** and a state for its **variables**
- Method for finding (counter-)**model** by finite interpretation and expansion
- Important equivalences about quantifiers (beware, no usable notion of normal form)
Next time

- Skolemisation
- Semi-algorithm to prove that a formula is unsatisfiable.

Homework

Every man is mortal.
Socrates is a man.
Hence Socrates is mortal.

- Look for a counter-model using 1-expansion then 2-expansion.
- What is your conclusion?