Logic

The $\alpha$ and $\omega$ in science.

- Basis of mathematics.
- Also of engineering.
  - Particularly useful for verification (model-checking = checking a model against a logical formula).
  - But also used in other domains, e.g.: Prolog, Datalog, UML OCL (Object Constraint Language), ...

A myriad of logics:

- Propositional logic
- First-order logic
- Temporal logic
- ...
What is logic?

Logic = Syntax + Semantics + Proofs

Proofs
  • Manual, or
  • Automated: Proofs = Computations

Example:
  • Syntax: boolean formulas
  • Semantics: boolean functions
  • Proofs: is a formula satisfiable? valid (a tautology)?
    ▶ E.g., for boolean logic: an NP-complete problem (a representative for many combinatorial problems).

BOOLEAN LOGIC
(a.k.a. Propositional Logic or Propositional Calculus)
Syntax

Symbols:
- Constants: “false” and “true”, or 0, 1, or ⊥, ⊤
- Variable symbols (atomic propositions): p, q, ..., x, y, ...
- Boolean connectives: ∧ (and), ∨ (or), ¬ (not), → (implies), ≡ or ↔ (is equivalent to)
- Parentheses ( ): used to make syntax unambiguous

Expressions (formulas):

\[ \phi ::= 0 \mid 1 \mid p \mid q \mid \ldots \mid x \mid y \mid \ldots \]
\[ \mid \phi_1 \land \phi_2 \mid \phi_1 \lor \phi_2 \]
\[ \mid \neg \phi' \]
\[ \mid \phi_1 \rightarrow \phi_2 \mid \phi_1 \equiv \phi_2 \]

Syntax

Examples:

\[ x \lor \neg x \]
\[ x \rightarrow y \rightarrow z \text{ (ambiguous)} \]
\[ x \rightarrow (y \rightarrow z) \]
\[ (x \rightarrow y) \rightarrow z \]
\[ (p \rightarrow q) \leftrightarrow (0 \lor \neg p \lor q) \]

¬ usually brings stronger, so \( \neg p \lor q \) means \((\neg p) \lor q\).

Similarly, \( p \land q \lor r \) usually means \((p \land q) \lor r\),
\( p \land q \rightarrow a \lor b \) usually means \((q \land q) \rightarrow (a \lor b)\),
etc.

When unsure, better use parentheses!
Alternative syntax

- ⇒ instead of →, but in modern logic notation, ⇒ is used for semantical entailment, as in “formula $\phi$ entails formula $\phi'$, or $\phi \Rightarrow \phi'$, meaning that $\phi'$ is true when $\phi$ is true”
- ⇔ instead of ↔
- + instead of $\lor$
- $\cdot$ instead of $\land$ (often omitted altogether)
- $\overline{x}$ instead of $\neg x$

E.g.,

$$xy + \overline{z}$$

instead of

$$(x \land y) \lor (\neg z)$$

Semantics

The **meaning** of logical formulas.

E.g., what is the semantics of a boolean formula such as $p \rightarrow q$?

“If $p$, then $q$”, of course.

So, why do we even need to talk about semantics?
Semantics

What is the meaning of a boolean formula?

Different views (all equivalent):

- A “truth table”.
- A boolean function.
- A set containing the “solutions” ("models") of the formula.

Why not consider the syntax itself to be the semantics?

Semantics

Formula:

$$x \land (y \lor z)$$

Truth table:

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

An equivalent formula (different syntax, same semantics):

$$(x \land y) \lor (x \land z)$$
Semantics

Boolean function: a function $f : \mathbb{B}^n \rightarrow \mathbb{B}^m$, where $\mathbb{B} = \{0, 1\}$.

Formula:

$$x \land (y \lor z)$$

defines\(^1\) the boolean function: $f : \mathbb{B}^3 \rightarrow \mathbb{B}$ such that:

$$f(0, 0, 0) = 0$$
$$f(0, 0, 1) = 0$$

... 

---

\(^1\)assuming an order on the variables: (1) $x$, (2) $y$, (3) $z$.

Semantics

A formula $\phi : x \land (y \lor z)$ defines\(^2\) a subset $\llbracket \phi \rrbracket \subseteq \mathbb{B}^3$:

$$\llbracket \phi \rrbracket = \{(1, 0, 1), (1, 1, 0), (1, 1, 1)\}$$

This is the set of “solutions”: all assignments to $x, y, z$ which make the formula true.

To be independent from an implicit order on variables, we can also view $\llbracket \phi \rrbracket$ as a set of minterms:

$$\llbracket \phi \rrbracket = \{x\overline{y}z, x\overline{y}\overline{z}, xyz\}$$

We can also view $\llbracket \phi \rrbracket$ as a set of sets of atomic propositions:

$$\llbracket \phi \rrbracket = \{\{x, z\}, \{x, y\}, \{x, y, z\}\}$$

What is the type of $\llbracket \phi \rrbracket$ in this last case?

$\llbracket \phi \rrbracket \subseteq \mathbb{B}^P = 2^P$ where $P$ is the set of atomic propositions (\(=\) formula variables).

\(^2\)assuming an order on the variables: (1) $x$, (2) $y$, (3) $z$. 

---

Stavros Tripakis (UC Berkeley)  
EE 144/244, Fall 2015  
Basic Concepts of Logic 11 / 34
Semantics: satisfaction relation

Satisfaction relation:

\[ a \models \phi \]

means \( a \) is a “solution” (or model) of \( \phi \) (or “\( a \) satisfies \( \phi \)”).

So

\[ a \models \phi \quad \text{iff} \quad a \in \left[ \phi \right] \]

Semantics: satisfiability, validity

A formula \( \phi \) is **satisfiable** if \( \left[ \phi \right] \) is non-empty, i.e., if there exists \( a \models \phi \).

A formula \( \phi \) is **valid** (a **tautology**) if for all \( a \), \( a \models \phi \), i.e., if \( \left[ \phi \right] = 2^P \).
Limitations of propositional logic

>All humans are mortal.

How to write it in propositional logic?

We can associate one proposition $p_i$ for every human $i$, with the meaning “human $i$ is mortal”, and then state:

$$p_1 \land p_2 \land \cdots \land p_{7000000000}$$

But even this is not enough, since we also want to talk about future generations.
Expressing this in (first-order) predicate logic

$$\forall x : H(x) \to M(x)$$

$x$: variable

$H, M$: predicates (functions that return “true” or “false”)

$H(x)$: “$x$ is human”.

$M(x)$: “$x$ is mortal”.

$\forall$: “for all” quantifier.

First-Order Predicate Logic (FOL) – Syntax

Terms:

$$t ::= x \mid c \mid f(t_1, ..., t_n)$$

where $x$ is any variable symbol, $c$ is any constant symbol, and $f$ is any function symbol of some arity $n$.

Formulas:

$$\phi ::= P(t_1, ..., t_n)$$

$$\mid (\phi \land \phi) \mid (\phi \lor \phi) \mid (\neg \phi) \mid \cdots$$

$$\mid (\forall x : \phi) \mid (\exists x : \phi)$$

where $P$ is any predicate symbol of some arity $n$, and $t_i$ are terms.

---

$^3$ constants can also be seen as functions of arity 0
FOL – Syntax

Example:

$$\forall x : x > 0 \rightarrow x + 1 > 0$$

or, more pedantically:

$$\forall x : > (x, 0) \rightarrow > (+ (x, 1), 0)$$

- 0, 1: constants
- $x$: variable symbol
- $+$: function symbol of arity 2
- $>$: predicate symbol of arity 2

FOL – Syntax

Note:

- This is also a syntactically well-formed formula:
  $$x > 0 \rightarrow x + 1 > 0$$

- so is this:
  $$\forall x : x > y$$

- or this:
  $$\forall x : 2z > f(y)$$
Parse Tree of Formula

Formula: \( \forall x : x > 0 \rightarrow x + 1 > 0 \)

Parse tree:

Free and Bound Variables

Formula: \( \forall x : x > y \)

Parse tree:

\( y \) is free in the formula: no ancestor of the leaf node \( y \) is a node of the form \( \forall y \) or \( \exists y \).

\( x \) is bound in the formula: has ancestor \( \forall x \).
Scope of Variables

Formula: \((\forall x : x = x \land \exists x : P(x)) \land x > 0\)

Parse tree:

Renaming

Formula: \((\forall x : x = x \land \exists x : P(x)) \land x > 0 \leadsto (\forall y : y = y \land \exists z : P(z)) \land x > 0\)

Parse tree:
FOL – Semantics

In propositional logic, a “solution” (model) of a formula was simply an assignment of truth values to the propositional variables. E.g.,

\[ (p := 1, q := 0) \models p \lor q \]

What are the “solutions” (models) of predicate logic formulas?

\[ ??? \models \forall x : P(x) \rightarrow \exists y : Q(x, y) \]

Cannot give meaning to the formula without first giving meaning to \( P, Q \).

FOL – Semantics

Let \( \mathcal{P} \) and \( \mathcal{F} \) be the sets of predicate and function symbols (for simplicity \( \mathcal{F} \) also includes the constants).

A model \( \mathcal{M} \) for the pair \((\mathcal{P}, \mathcal{F})\) consists of the following:

- A non-empty set \( \mathcal{U} \), the universe of concrete values.
- For each 0-arity symbol \( c \in \mathcal{F} \), a concrete value \( c_{\mathcal{M}} \in \mathcal{U} \).
- For each \( f \in \mathcal{F} \) with arity \( n \), a function \( f_{\mathcal{M}} : \mathcal{U}^n \rightarrow \mathcal{U} \).
- For each \( P \in \mathcal{P} \) with arity \( n \), a set \( P_{\mathcal{M}} \subseteq \mathcal{U}^n \).

Note:

- \( c, f, P \) are just symbols (syntactic objects).
- \( c_{\mathcal{M}}, f_{\mathcal{M}}, P_{\mathcal{M}} \) are semantical objects (values, functions, sets).
FOL – Semantics

Example:

\[ \forall x : P(x) \rightarrow \exists y : Q(x, y) \]

Let \( M \) be such that

- \( U = \mathbb{N} \): the set of naturals.
- \( P_M = \{0, 2, \ldots\} \): the set of even naturals.
- \( Q_M = \{(0, 1), (1, 2), (2, 3), \ldots\} \): the set of pairs \((n, n + 1)\), for \( n \in \mathbb{N} \).

Then the statement above is true.

Of course, it could have been written “more clearly” (for a human):

\[ \forall x : \text{Even}(x) \rightarrow \exists y : y = x + 1 \]

... but a computer (or a person who does not speak English) is equally clueless as to what \( P \) or \( \text{Even} \) means ...
FOL – Semantics

What is the meaning of $\forall x : x > y$ ?

Undefined if we know nothing about the value of $y$.

We need one more thing: environments (or “look-up tables” for variables).

Environment:

$$l : \text{VariableSymbols} \rightarrow \mathcal{U}$$

assigns a concrete value to every variable symbol.

Notation:

$$l[x \sim a]$$

is a new environment $l'$ such that $l'(x) = a$ and $l'(y) = l(y)$ for any other variable $y$.

FOL – Semantics: Giving concrete values to terms

Once we have $\mathcal{M}$ and $l$, every term evaluates to a concrete value in $\mathcal{U}$.

Example:

$$\mathcal{M} : \mathcal{U} = \mathbb{N}, "0" = 0, "1" = 1, ..., + = \text{addition function},$$

$$l : x \sim 2, y \sim 1$$

<table>
<thead>
<tr>
<th>term $t$</th>
<th>value $\mathcal{M}_l(t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x + 1$</td>
<td>3</td>
</tr>
<tr>
<td>$x \cdot y$</td>
<td>2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

For a term $t$, we denote this value by $\mathcal{M}_l(t)$. 

---

Stavros Tripakis (UC Berkeley) EE 144/244, Fall 2015 Basic Concepts of Logic 30 / 34
Finally we can define the satisfaction relation for first-order predicate logic ($\mathcal{M}$: model, $l$: environment, $\phi$: formula):

$$\mathcal{M}, l \models \phi$$

- $\mathcal{M}, l \models P(t_1, \ldots, t_n)$ iff $(\mathcal{M}_l(t_1), \ldots, \mathcal{M}_l(t_n)) \in P_M$
- $\mathcal{M}, l \models \phi_1 \land \phi_2$ iff $\mathcal{M}, l \models \phi_1$ and $\mathcal{M}, l \models \phi_2$
- $\mathcal{M}, l \models \neg \phi$ iff $\mathcal{M}, l \not\models \phi$
- $\mathcal{M}, l \models \forall x : \phi$ iff for all $a \in \mathcal{U}$ : $\mathcal{M}, l[x \sim a] \models \phi$ holds
- $\mathcal{M}, l \models \exists x : \phi$ iff for some $a \in \mathcal{U}$ : $\mathcal{M}, l[x \sim a] \models \phi$ holds

A FOL formula $\phi$ is **satisfiable** if there exist $\mathcal{M}, l$ such that $\mathcal{M}, l \models \phi$ holds.

A formula $\phi$ is **valid** (a tautology) if for all $\mathcal{M}, l$, it holds $\mathcal{M}, l \models \phi$. 

FOL – Semantics: Satisfiability, Validity

Examples:

1. \( \forall x : P(x) \rightarrow P(x) \)
   Valid.

2. \( x \geq 0 \land f(x) \geq 0 \land y \geq 0 \land f(y) \geq 0 \land x \neq y \)
   Satisfiable.
   Example model: \( U = \mathbb{N}, x \mapsto 0, y \mapsto 1, f(.) \mapsto 0, \neq \) is the “not equal to” relation on \( \mathbb{N} \): \( \neq \mapsto \{(0,1),(0,2),\ldots,(1,0),(1,2),\ldots\} \).

3. \( x + 2 = y \land f(\text{read(write}(A,x,3),y-2)) \neq f(y-x+1) \)
   Satisfiable with a non-standard interpretation of +, − or read, write.
   Unsatisfiable with the standard interpretation of those symbols (theories of arithmetic and arrays). Why?

Bibliography

