EE 244: Fundamental Algorithms for System Modeling, Analysis, and Optimization
Fall 2016

A Logic Primer

Stavros Tripakis
University of California, Berkeley

Logic

The α and ω 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).
  - Software tools (SAT solvers) routinely solve such problems today, even with tens of thousands of variables or more.

BOOLEAN LOGIC

(a.k.a. Propositional Logic or Propositional Calculus)

Syntax

Symbols:

- Constants: "false" and "true", or $0, 1$, or $\bot, \top$
- Variable symbols (*atomic propositions*): $p, q, ..., x, y, ...$
- Boolean connectives: $\wedge$ (and), $\vee$ (or), $\neg$ (not), $\rightarrow$ (implies), $\equiv$ or $\leftrightarrow$ (is equivalent to)
- Parentheses ( ): used to make syntax unambiguous

Expressions (formulas):

$$
\begin{aligned}
\phi \quad ::= \quad & 0 \mid 1 \mid p \mid q \mid ... \mid x \mid y \mid ... \\
& \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \\
& \mid \neg \phi' \\
& \mid \phi_1 \rightarrow \phi_2 \mid \phi_1 \equiv \phi_2
\end{aligned}
$$

Syntax

Examples:

$$x \vee \neg x$$

$$x \rightarrow y \rightarrow z \text{ (ambiguous)}$$

$$x \rightarrow (y \rightarrow z)$$

$$(x \rightarrow y) \rightarrow z$$

$$(p \rightarrow q) \leftrightarrow (0 \vee \neg p \vee q)$$

$\neg$ usually bings stronger, so $\neg p \vee q$ means $(\neg p) \vee q$.

Similarly, $p \wedge q \vee r$ usually means $(p \wedge q) \vee r$,
$p \wedge q \rightarrow a \vee b$ usually means $(q \wedge q) \rightarrow (a \vee b)$,
etc.

To be sure, better use parentheses!

Alternative syntax

- $\Rightarrow$ instead of $\rightarrow$, but in modern logic notation, $\Rightarrow$ is used for semantical entailment, as in "formula ϕ entails formula ϕ', or $\phi \Rightarrow \phi'$, meaning that ϕ' is true when ϕ is true"
- $\Leftrightarrow$ instead of $\leftrightarrow$
- $+$ instead of $\vee$
- $\cdot$ instead of $\wedge$ (often omitted altogether)
- $\overline{x}$ instead of $\neg x$

E.g.,

$$xy + \overline{z}$$

instead of

$$(x \wedge y) \vee (\neg z)$$

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:

$$x \wedge (y \vee z)$$

Truth table:

x	y	z	result
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

An equivalent formula (different syntax, same semantics):

$$(x \wedge y) \vee (x \wedge z)$$

Semantics

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

Formula:

$$x \wedge (y \vee z)$$

defines[1] the boolean function: $f : \mathbb{B}^3 \to \mathbb{B}$ such that:

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

[1]assuming an order on the variables: (1) x, (2) y, (3) z.

Semantics

A formula $\phi : x \wedge (y \vee z)$ defines[2] a subset $[\![\phi]\!] \subseteq \mathbb{B}^3$:

$$[\![\phi]\!] = \{(1,0,1), (1,1,0), (1,1,1)\}$$

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 $[\![\phi]\!]$ as a set of *minterms*:

$$[\![\phi]\!] = \{x\overline{y}z, xy\overline{z}, xyz\}$$

We can also view $[\![\phi]\!]$ as a set of *sets of atomic propositions*:

$$[\![\phi]\!] = \{\{x,z\}, \{x,y\}, \{x,y,z\}\}$$

What is the type of $[\![\phi]\!]$ in this last case?
$[\![\phi]\!] \subseteq \mathbb{B}^P = 2^P$ where P is the set of atomic propositions ($=$ formula variables).

[2]assuming an order on the variables: (1) x, (2) y, (3) z.

Semantics: satisfaction relation

Satisfaction relation:

$$a \models \phi$$

means a is a "solution" (or model) of ϕ (or "a *satisfies* ϕ").

So

$$a \models \phi \qquad \text{iff} \qquad a \in [\![\phi]\!]$$

Semantics: satisfiability, validity

A formula ϕ is *satisfiable* if $[\![\phi]\!]$ is non-empty, i.e., if there exists $a \models \phi$.

A formula ϕ is *valid* (a *tautology*) if for all a, $a \models \phi$, i.e., if $[\![\phi]\!] = 2^P$.

PREDICATE LOGIC

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:

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

But even this is not enough, since we also want to talk about future generations.

Expressing this in (first-order) predicate logic

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

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:

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

where x is any variable symbol, c is any constant symbol,[3] and f is any function symbol of some arity n.

Formulas:

$$\begin{array}{rcl} \phi & ::= & P(t_1, ..., t_n) \\ & & \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid (\neg \phi) \mid \cdots \\ & & \mid (\forall x : \phi) \mid (\exists x : \phi) \end{array}$$

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:

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

or, more pedantically:

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

- $0, 1$: 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:

$$x > 0 \to x + 1 > 0$$

- so is this:

$$\forall x : x > y$$

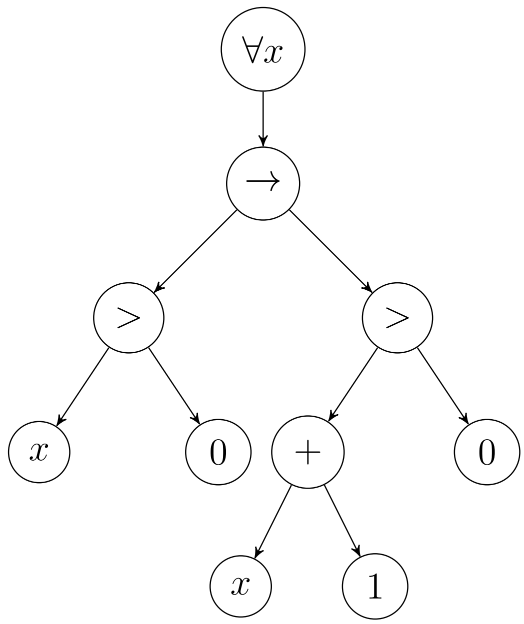
- or this:

$$\forall x : 2z > f(y)$$

Parse Tree of Formula

Formula: $\forall x : x > 0 \to x + 1 > 0$

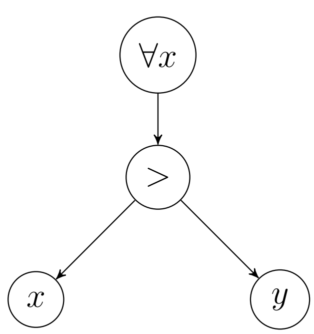
Parse tree:



Free and Bound Variables

Formula: $\forall x : x > y$

Parse tree:



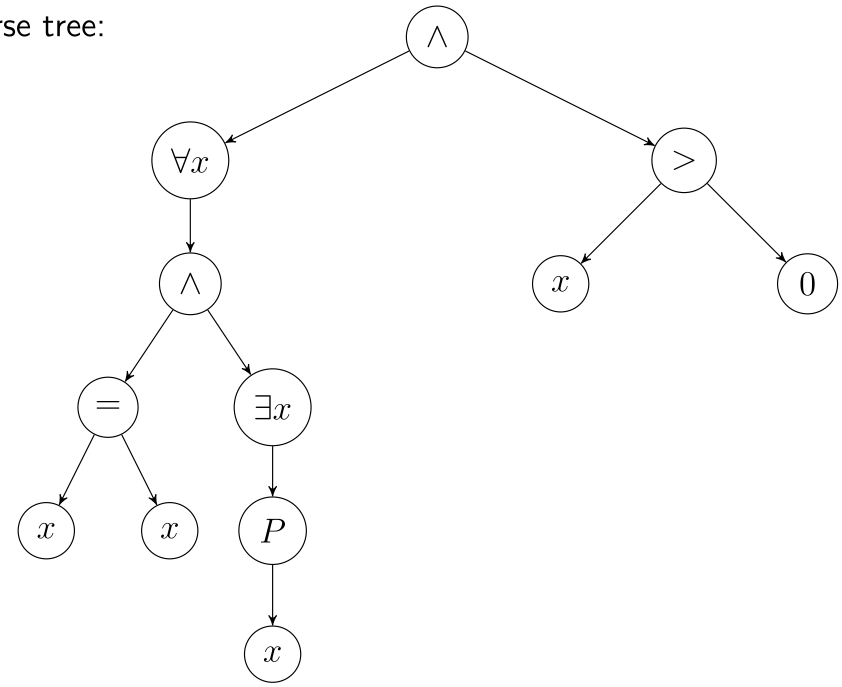
y is *free* in the formula: no ancestor of the leaf node y is a node of the form $\forall y$ or $\exists y$.

x is *bound* in the formula: has ancestor $\forall x$.

Scope of Variables

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

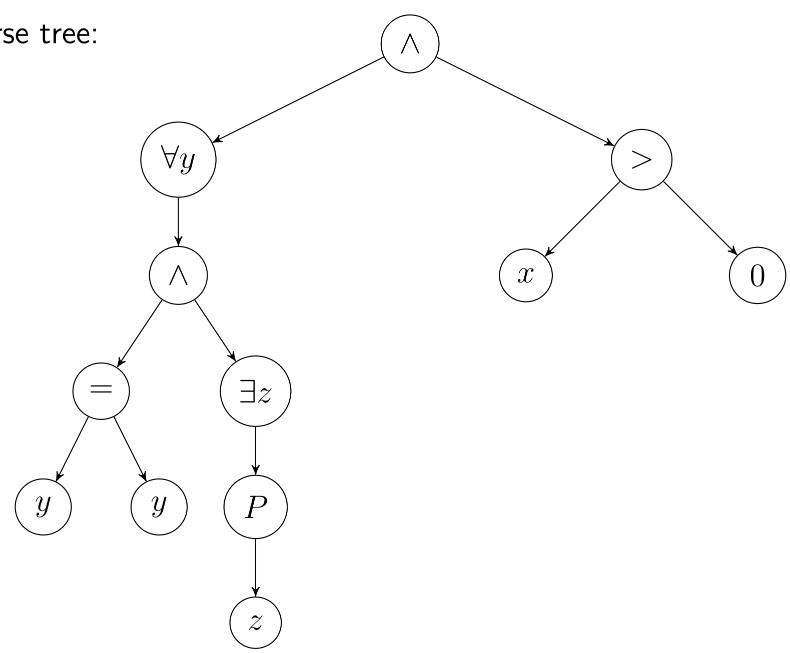
Parse tree:



Renaming

Formula: $(\forall x : x = x \wedge \exists x : P(x)) \wedge x > 0 \rightsquigarrow (\forall y : y = y \wedge \exists z : P(z)) \wedge x > 0$

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.,

$$\underbrace{(p := 1, q := 0)}_{\textit{model}} \models \underbrace{p \lor q}_{\textit{formula}}$$

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

$$\underbrace{???}_{\textit{model}} \models \underbrace{\forall x : P(x) \to \exists y : Q(x, y)}_{\textit{formula}}$$

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 $f_{\mathcal{M}} : \mathcal{U}^n \to \mathcal{U}$.
- 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 $\mathcal{M}$ be such that

- $\mathcal{U} = \mathbb{N}$: the set of naturals.
- $P_{\mathcal{M}} = \{0, 2, ...\}$: the set of even naturals.
- $Q_{\mathcal{M}} = \{(0, 1), (1, 2), (2, 3), ...\}$: 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):

$$\forall x : \textit{Even}(x) \rightarrow \exists y : y = x + 1$$

... but a computer (or a person who does not speak English) is equally clueless as to what P or *Even* means ...

FOL – Semantics

Example:

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

Let $\mathcal{M}'$ be another model such that

- $\mathcal{U} = \mathbb{N}$: the set of naturals.
- $P_{\mathcal{M}'} = \{0, 2, ...\}$: the set of even naturals.
- $Q_{\mathcal{M}'} = \{(1, 0), (3, 1), (5, 2), ...\}$: the set of pairs $(2n + 1, n)$, for $n \in \mathbb{N}$.

Then the statement above is false.

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:

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

assigns a concrete value to every variable symbol.

Notation:

$$l[x \rightsquigarrow a]$$

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" $= 1$, ..., $+ =$ addition function,
...

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

term t	value $\mathcal{M}_l(t)$
$x + 1$	3
$x \cdot y$	2
...	

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

FOL – Semantics

Finally we can define the satisfaction relation for first-order predicate logic ($\mathcal{M}$: model, l: environment, ϕ: formula):

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

$\mathcal{M}, l \models P(t_1, ..., t_n)$	iff	$\big(\mathcal{M}_l(t_1), ..., \mathcal{M}_l(t_n)\big) \in P_{\mathcal{M}}$
$\mathcal{M}, l \models \phi_1 \wedge \phi_2$	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 $a \in \mathcal{U} : \mathcal{M}, l[x \rightsquigarrow a] \models \phi$ holds
$\mathcal{M}, l \models \exists x : \phi$	iff	for some $a \in \mathcal{U} : \mathcal{M}, l[x \rightsquigarrow a] \models \phi$ holds

FOL – Semantics: Satisfiability, Validity

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

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

FOL – Semantics: Satisfiability, Validity

Examples:

1. $$\forall x : P(x) \to P(x)$$

   Valid.

2. $$x \geq 0 \wedge f(x) \geq 0 \wedge y \geq 0 \wedge f(y) \geq 0 \wedge x \neq y$$

   Satisfiable.

   Example model: $\mathcal{U} = \mathbb{N}$, $x \mapsto 0$, $y \mapsto 1$, $f(\_) \mapsto 0$, $\neq$ is the "not equal to" relation on $\mathbb{N}$: $\neq \mapsto \{(0,1),(0,2),...,(1,0),(1,2),...\}$.

3. $$x + 2 = y \wedge f(\texttt{read}(\texttt{write}(A, x, 3), y - 2)) \neq f(y - x + 1)$$

   Satisfiable with a non-standard interpretation of $+, -$ or $\texttt{read}, \texttt{write}$.

   Unsatisfiable with the standard interpretation of those symbols (theories of arithmetic and arrays). Why?

Bibliography

Biere, A., Heule, M., Van Maaren, H., and Walsh, T. (2009).
Handbook of Satisfiability.
IOS Press.

Huth, M. and Ryan, M. (2004).
Logic in Computer Science: Modelling and Reasoning about Systems.
Cambridge University Press.

Tourlakis, G. (2008).
Mathematical Logic.
Wiley.