First Order Logic (FOL)

Testing, Quality Assurance, and Maintenance
Winter 2019

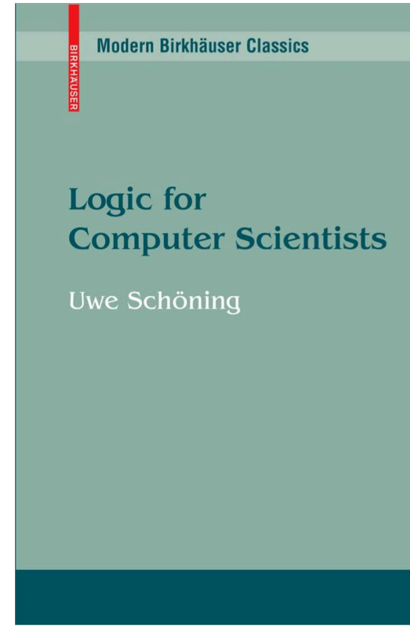
Prof. Arie Gurfinkel

based on slides by Prof. Ruzica Piskac, Nikolaj Bjorner, and others

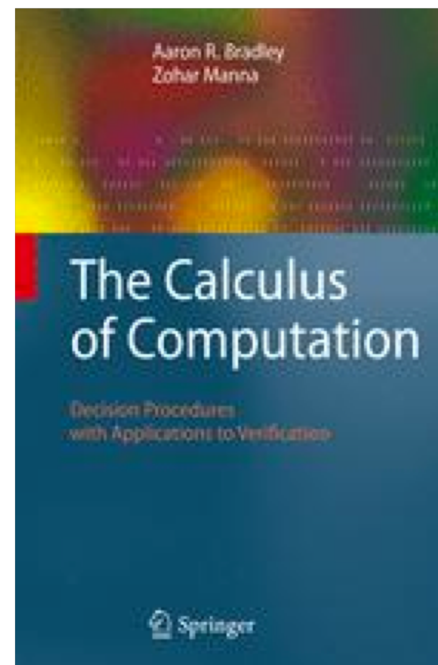
UNIVERSITY OF WATERLOO

References

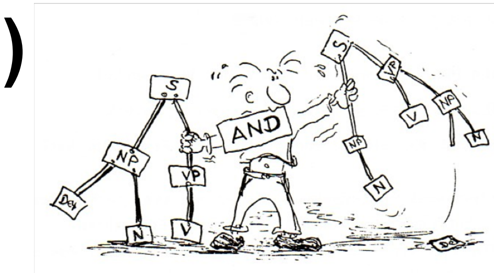
- Chpater 2 of Logic for Computer Scientists
  http://www.springerlink.com/content/978-0-8176-4762-9/



- Chapters 2 and 3 of Calculus of Computation
  https://link.springer.com/book/10.1007/978-3-540-74113-8

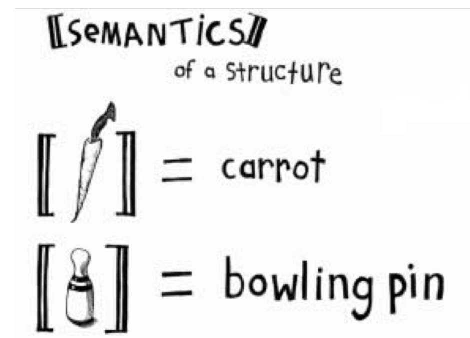


Syntax and Semantics (Again)



Syntax

- MW: the way in which linguistic elements (such as words) are put together to form constituents (such as phrases or clauses)
- Determines and restricts how things are written



Semantics

- MW: the study of meanings
- Determines how syntax is interpreted to give meaning

The language of First Order Logic

Functions , Variables, Predicates

- *f, g,…* *x, y, z, …* P, Q, =, <, …

Atomic formulas, Literals

- $P(x,f(y))$, $\neg Q(y,z)$

Quantifier free formulas

- $P(f(a), b) \wedge c = g(d)$

Formulas, sentences

- $\forall x . \forall y . [P(x, f(x)) \vee g(y,x) = h(y)]$

Language: Signatures

A *signature* Σ is a finite set of:

- Function symbols:

$$\Sigma_F = \{\ f,\ g,\ +,\ \ldots\ \}$$

- Predicate symbols:

$$\Sigma_P = \{\ P,\ Q, =,\ \text{true},\ \text{false},\ \ldots\ \}$$

- And an *arity* function:

$$\Sigma \to N$$

Function symbols with arity 0 are *constants*

- notation: $f_{/2}$ means a symbol with arity 2

A countable set *V* of *variables*

- disjoint from Σ

Language: Terms

The set of *terms* $T(\Sigma_F, V)$ is the smallest set formed by the syntax rules:

- $t \in T \quad ::= \quad v \qquad\qquad v \in V$

  $\qquad\qquad\qquad | \quad f(t_1, \ldots, t_n) \qquad f \in \Sigma_F, t_1, \ldots, t_n \in T$

Ground terms are given by $T(\Sigma_F, \varnothing)$

- a term is *ground* if it contains no variables

Language: Atomic Formulas

$a \in Atoms \quad ::= P(t_1, \ldots, t_n)$

$P \in \Sigma_P \; t_1, \ldots, t_n \in T$

An atom is *ground* if $t_1, \ldots, t_n \in T(\Sigma_F, \emptyset)$

- ground atom contains no variables

Literals are atoms and negation of atoms:

$l \in Literals \quad ::= a \mid \neg a \qquad\qquad a \in Atoms$

UNIVERSITY OF WATERLOO

Language: Quantifier free formulas

The set QFF(Σ,V) of *quantifier free formulas* is the smallest set such that:

$\varphi \in$ QFF ::=	$a \in Atoms$	*atoms*
	$\mid \neg \varphi$	*negations*
	$\mid \varphi \leftrightarrow \varphi'$	*bi-implications*
	$\mid \varphi \wedge \varphi'$	*conjunction*
	$\mid \varphi \vee \varphi'$	*disjunction*
	$\mid \varphi \rightarrow \varphi'$	*implication*

UNIVERSITY OF WATERLOO

Language: Formulas

The set of *first-order formulas* are obtained by adding the formation rules:

$$
\begin{array}{llll}
\varphi ::= & & \ldots & \\
& | & \forall x \,.\, \varphi & \textit{universal quant.} \\
& | & \exists x \,.\, \varphi & \textit{existential quant.}
\end{array}
$$

Free (occurrences) of *variables* in a formula are theose not bound by a quantifier.

A sentence is a first-order formula with no free variables.

Dreadbury Mansion Mystery

Someone who lived in Dreadbury Mansion killed Aunt Agatha. Agatha, the Butler and Charles were the only people who lived in Dreadbury Mansion. A killer always hates his victim, and is never richer than his victim. Charles hates no one that aunt Agatha hates. Agatha hates everyone except the butler. The butler hates everyone not richer than Aunt Agatha. The butler also hates everyone Agatha hates. No one hates everyone. Agatha is not the butler.

Who killed Aunt Agatha?

Dreadbury Mansion Mystery

Someone who lived in Dreadbury Mansion killed Aunt Agatha. Agatha, the Butler and Charles were the only people who lived in Dreadbury Mansion. A killer always hates his victim, and is never richer than his victim. Charles hates no one that aunt Agatha hates. Agatha hates everyone except the Butler. The Butler hates everyone not richer than Aunt Agatha. The Butler also hates everyone Agatha hates. No one hates everyone. Agatha is not the Butler.

Who killed Aunt Agatha?

Constants are blue

Predicates are purple

Dreadbury Mansion Mystery

$killed/_2, hates/_2, richer/_2, a/_0, b/_0, c/_0$

$$\exists x \cdot killed(x, a) \tag{1}$$

$$\forall x \cdot \forall y \cdot killed(x, y) \implies (hates(x, y) \land \neg richer(x, y)) \tag{2}$$

$$\forall x \cdot hates(a, x) \implies \neg hates(c, x) \tag{3}$$

$$hates(a, a) \land hates(a, c) \tag{4}$$

$$\forall x \cdot \neg richer(x, a) \implies hates(b, x) \tag{5}$$

$$\forall x \cdot hates(a, x) \implies hates(b, x) \tag{6}$$

$$\forall x \cdot \exists y \cdot \neg hates(x, y) \tag{7}$$

$$a \neq b \tag{8}$$

Solving Dreadbury Mansion in SMT

```
(declare-datatypes () ((Mansion (Agatha) (Butler) (Charles))))
(declare-fun killed (Mansion Mansion) Bool)
(declare-fun hates (Mansion Mansion) Bool)
(declare-fun richer (Mansion Mansion) Bool)
(assert (exists ((x Mansion)) (killed x Agatha)))
(assert (forall ((x Mansion) (y Mansion))
   (=> (killed x y) (hates x y))))
(assert (forall ((x Mansion) (y Mansion))
   (=> (killed x y) (not (richer x y)))))
(assert (forall ((x Mansion))
   (=> (hates Agatha x) (not (hates Charles x)))))
(assert (hates Agatha Agatha))
(assert (hates Agatha Charles))
(assert (forall ((x Mansion))
   (=> (not (richer x Agatha)) (hates Butler x))))
(assert (forall ((x Mansion))
   (=> (hates Agatha x) (hates Butler x))))
(assert (forall ((x Mansion)) (
   exists ((y Mansion)) (not (hates x y)))))

(check-sat)
(get-model)
```

Models (Semantics)

A model *M* is defined as:

- Domain *S;* non-empty set of elements; often called the *universe*
- Interpretation, $f^M : S^n \rightarrow S$ for each $f \in \Sigma_F$ with arity(f) = n
- Interpretation $P^M \subseteq S^n$ for each $P \in \Sigma_P$ with arity(P) = n
- Assignment $x^M \in S$ for every variable $x \in V$

A formula φ is true in a model *M* if it evaluates to true under the given interpretations over the domain *S.*

M is a *model* for a set of sentences T if all sentences of T are true in M.

Models (Semantics)

A term t in a model M is interpreted as:

- Variable $x \in V$ is interpreted as x^M
- $f(t_1, \ldots, t_n)$ is interpreted as $f^M(a_1, \ldots, a_n)$,
  - where a_i is the current interpretation of t_i

$P(t_1, \ldots, t_n)$ atom is *true* in a model M if and only if

- $(a_1, \ldots, a_n) \in P^M$, where
- a_i is the current interpretation of t_i

Models (Semantics)

A formula φ is true in a model M if:

- $M \models \neg \varphi$ iff $M \not\models \varphi$ (i.e., M is not a model for φ)
- $M \models \varphi \leftrightarrow \varphi'$ iff $M \models \varphi$ is equivalent to $M \models \varphi'$
- $M \models \varphi \wedge \varphi'$ iff $M \models \varphi$ and $M \models \varphi'$
- $M \models \varphi \vee \varphi'$ iff $M \models \varphi$ or $M \models \varphi'$
- $M \models \varphi \rightarrow \varphi'$ iff *if* $M \models \varphi$ then $M \models \varphi'$
- $M \models \forall x. \varphi$ iff for all $s \in S$, $M[x:=s] \models \varphi$
- $M \models \exists x. \varphi$ iff exists $s \in S$, $M[x:=s] \models \varphi$

Interpretation Example

$$
\begin{aligned}
\Sigma &= \{0, +, <\}, \text{ and } M \text{ such that } |M| = \{a, b, c\} \\
M(0) &= a, \\
M(+) &= \{\langle a, a \mapsto a\rangle, \langle a, b \mapsto b\rangle, \langle a, c \mapsto c\rangle, \langle b, a \mapsto b\rangle, \langle b, b \mapsto c\rangle, \\
&\quad \langle b, c \mapsto a\rangle, \langle c, a \mapsto c\rangle, \langle c, b \mapsto a\rangle, \langle c, c \mapsto b\rangle\} \\
M(<) &= \{\langle a, b\rangle, \langle a, c\rangle, \langle b, c\rangle\}
\end{aligned}
$$

If $M(x) = a, M(y) = b, M(z) = c$, then

$M[\![+(+(x, y), z)]\!] =$

$M(+)(M(+)(M(x), M(y)), M(z)) = M(+)(M(+)(a, b), c) =$

$M(+)(b, c) = a$

Interpretation Example

$$\begin{array}{lcl}
\Sigma & = & \{0, +, <\}, \text{ and } M \text{ such that } |M| = \{a, b, c\} \\
M(0) & = & a, \\
M(+) & = & \{\langle a, a \mapsto a\rangle, \langle a, b \mapsto b\rangle, \langle a, c \mapsto c\rangle, \langle b, a \mapsto b\rangle, \langle b, b \mapsto c\rangle, \\
 & & \langle b, c \mapsto a\rangle, \langle c, a \mapsto c\rangle, \langle c, b \mapsto a\rangle, \langle c, c \mapsto b\rangle\} \\
M(<) & = & \{\langle a, b\rangle, \langle a, c\rangle, \langle b, c\rangle\}
\end{array}$$

$$M \models (\forall x : (\exists y : +(x, y) = 0))$$

$$M \not\models (\forall x : (\exists y : x < y))$$

$$M \models (\forall x : (\exists y : +(x, y) = x))$$

Dreadbury Mansion Mystery

$killed/_2, hates/_2, richer/_2, a/_0, b/_0, c/_0$

$$\exists x \cdot killed(x, a) \tag{1}$$

$$\forall x \cdot \forall y \cdot killed(x, y) \implies (hates(x, y) \land \neg richer(x, y)) \tag{2}$$

$$\forall x \cdot hates(a, x) \implies \neg hates(c, x) \tag{3}$$

$$hates(a, a) \land hates(a, c) \tag{4}$$

$$\forall x \cdot \neg richer(x, a) \implies hates(b, x) \tag{5}$$

$$\forall x \cdot hates(a, x) \implies hates(b, x) \tag{6}$$

$$\forall x \cdot \exists y \cdot \neg hates(x, y) \tag{7}$$

$$a \neq b \tag{8}$$

UNIVERSITY OF WATERLOO

Dreadbury Mansion Mystery: Model

$killed/_2, hates/_2, richer/_2, a/_0, b/_0, c/_0$

$$S = \{a, b, c\}$$

$$M(a) = a \qquad M(b) = b$$

$$M(c) = c \qquad M(killed) = \{(a, a)\}$$

$$M(richer) = \{(b, a)\}$$

$$M(hates) = \{(a, a), (a, c)(b, a), (b, c)\}$$

UNIVERSITY OF WATERLOO

Semantics: Exercise

Drinker's paradox:

There is someone in the pub such that, if he is drinking, everyone in the pub is drinking.

- $\exists x.\ (D(x) \rightarrow \forall y.\ D(y))$

Is this logical formula valid?

Or unsatisfiable?

Or satisfiable but not valid?

Inference Rules for First Order Logic

We write $\vdash A$ when A can be inferred from basic axioms

We write $B \vdash A$ when A can be inferred from B

Natural deduction style rules

Notation: A[*a*/*x*] means A with variable *x* replaced by term *a*

$$\frac{A \qquad B}{A \wedge B} \qquad \frac{A}{A \vee B} \qquad \frac{B}{A \vee B} \qquad \frac{A \Rightarrow B \qquad A}{B}$$

$$\frac{A[e/x]}{\exists\, x.\, A} \qquad \frac{\forall\, x.\, A}{A[e/x]} \qquad \frac{A[a/x]}{\forall\, x.\, A}\ \ a \text{ is fresh}$$

$$\frac{A \vdash B}{A \Rightarrow B} \qquad \frac{\vdash \exists\, x.\, A \qquad A[a/x] \vdash B}{\vdash B}\ \ a \text{ is fresh}$$

UNIVERSITY OF WATERLOO

Theories

A (first-order) *theory T (over signature Σ) is a set of (deductively closed) sentences (over Σ and V)* - *axioms*

Let *DC*(Γ) be the deductive closure of a set of sentences Γ.

- For every theory T, *DC*(T) = T

A theory T is *constistent* if *false* $\notin$ *T*

A theory captures the intendent interpretation of the functions and predicates in the signature

- e.g., '+' is a plus, '0' is number 0, etc.

We can view a (first-order) theory *T* as the class of all *models* of *T* (due to completeness of first-order logic).

Theory of Equality T_E

Signature: $\Sigma_E = \{ =, a, b, c, \ldots, f, g, h, \ldots, P, Q, R, \ldots. \}$

$=$, a binary predicate, interpreted by axioms

all constant, function, and predicate symbols.

Axioms:

1. $\forall x \,.\, x = x$ (reflexivity)
2. $\forall x, y \,.\, x = y \rightarrow y = x$ (symmetry)
3. $\forall x, y, z \,.\, x = y \wedge y = z \rightarrow x = z$ (transitivity)

Theory of Equality T_E

Signature: $\Sigma_E = \{ =, a, b, c, \ldots, f, g, h, \ldots, P, Q, R, \ldots. \}$

=, a binary predicate, interpreted by axioms

all constant, function, and predicate symbols.

Axioms:

4. for each positive integer *n* and *n*-ary function symbol *f*,

$\forall x_1, \ldots, x_n, y_1, \ldots, y_n . \bigwedge_i x_i = y_i \rightarrow f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n)$ (congruence)

5. for each positive integer *n* and *n*-ary predicate symbol *P*

$\forall x_1, \ldots, x_n, y_1, \ldots, y_n . \bigwedge_i x_i = y_i \rightarrow (P(x_1, \ldots, x_n) \leftrightarrow P(y_1, \ldots, y_n))$ (equivalence)

Theory of Peano Arithmetic (Natural Number)

Signature: $\Sigma_{PA} = \{ 0, 1, + , *, = \}$

Axioms of T_{PA} : axioms for theory of equality, T_E , plus:

1. $\forall x.\ \neg (x+ 1 = 0)$ (zero)
2. $\forall x, y.\ x + 1 = y + 1 \rightarrow x = y$ (successor)
3. $F[0] \land (\forall x.F[x] \rightarrow F[x+ 1]) \rightarrow \forall x.F[x]$ (induction)
4. $\forall x.\ x + 0 = x$ (plus zero)
5. $\forall x, y.\ x + (y + 1) = (x + y) + 1$ (plus successor)
6. $\forall x.\ x * 0 = 0$ (times zero)
7. $\forall x, y.\ x * (y + 1) = x * y + x$ (times successor)

Note that induction (#3) is an axiom schema

- one such axiom is added for each predicate F in the signature

Peano arithmetic is undecidable!

Theory of Presburger Arithmetic

Signature: $\Sigma_{PA} = \{ 0, 1, + , = \}$

Axioms of T_{PA} : axioms for theory of equality, T_E , plus:

1. $\forall x.\ \neg\ (x + 1 = 0)$ (zero)
2. $\forall x, y.\ x + 1 = y + 1 \rightarrow x = y$ (successor)
3. $F[0] \wedge (\forall x.F[x] \rightarrow F[x + 1]) \rightarrow \forall x.F[x]$ (induction)
4. $\forall x.\ x + 0 = x$ (plus zero)
5. $\forall x, y.\ x + (y + 1) = (x + y) + 1$ (plus successor)

Note that induction (#3) is an axiom schema

- one such axiom is added for each predicate F in the signature

Can extend the signature to allow multiplication by a numeric constant

Presburger arithmetic is decidable

- linear integer programming (ILP)

McCarthy theory of Arrays T_A

Signature: Σ_A = { read, write, = }

read(a, i) is a binary function:

- reads an array a at the index i
- alternative notations:
  - (select a i), and a[i]

write(a, i, v) is a ternary function:

- writes a value v to the index i of array a
- alternative notations:
  - (store a i v) , a[i:=v]
- side-effect free – results in new array, does not modify *a*

Axioms of T_A

Array congruence

- $\forall a , i, j . i = j \rightarrow$ read (a, i) = read (a, j)

Read-Over-Write 1

- $\forall a , v, i, j. i = j \rightarrow$ read (write (a, i, v), j) = v

Read-Over-Write 2

- $\forall a,v, i, j. i \neq j \rightarrow$read (write (a, i, v), j) = read (a, j)

Extensionality

- a=b $\leftrightarrow \forall i$. read(a, i) = read(b, i)

UNIVERSITY OF WATERLOO

T-Satisfiability

A formula $\varphi(x)$ is T-satisfiable in a theory T if there is a model of $DC(T \cup \exists x.\varphi(x))$.

That is, there is a model M for T in which $\varphi(x)$ evaluates to true.

Notation:

$$M \vDash_T \varphi(x)$$

where, $DC(V)$ stands for deductive closure of V

T-Validity

A formula $\varphi(x)$ is T-*valid* in a theory T if $\forall x. \varphi(x) \in T$

That is, $\forall x. \varphi(x)$ evaluates to *true* in every model M of T

T-validity:

$$\vDash_{\mathrm{T}} \varphi(x)$$

UNIVERSITY OF WATERLOO

Fragment of a Theory

Fragment of a theory *T* is a syntactically restricted subset of formulae of the theory

Example:

- Quantifier-free fragment of theory T is the set of formulae without quantifiers that are valid in T

Often decidable fragments for undecidable theories

Theory *T* is *decidable* if *T*-validity is decidable for every formula *F* of *T*

- There is an algorithm that always terminates with "yes" if *F* is *T*-valid, and "no" if *F* is *T*-unsatisfiable

Exercises (1/2)

Find a model for $P(f(x,y)) \Rightarrow P(g(x,y,x))$

Write an axiom that will restrict that every model has to have exactly three different elements.

Write a FOL formula stating that *i* is the position of the minimal element of an integer array *A*

Write a FOL formula stating that *v* is the minimal element of an integer array *A*

Exercises (1/2)

Find a model for P(f(x,y)) ⇒ P(g(x,y,x))

Write an axiom that will restrict that every model has to have exactly three different elements.

$$(\exists x, y, z \cdot x \neq y \wedge x \neq z \wedge y \neq z) \wedge (\forall a_0, a_1, a_2, a_3 \cdot \bigvee_{0 \leq i < j \leq 3} a_i = a_j)$$

Write a FOL formula stating that *i* is the position of the minimal element of an integer array *A*

$$isIntArray(A) \wedge isInt(i) \wedge 0 \leq i < len(A)$$

$$\forall j \cdot 0 \leq j < len(A) \wedge i \neq j \implies A[i] \leq A[j]$$

Write a FOL formula stating that *v* is the minimal element of an integer array *A*

$$isIntArray(A) \wedge isInt(v)$$

$$\exists i \cdot 0 \leq i < len(A) \wedge A[i] = v$$

$$\forall i \cdot 0 \leq i < len(A) \implies A[i] \leq v$$

UNIVERSITY OF WATERLOO

Exercises (2/2)

Show whether the following sentence is valid or not

$$(\exists x \cdot P(x) \lor Q(x)) \iff (\exists x \cdot P(x)) \lor (\exists x \cdot Q(x))$$

Show whether the following FOL sentence is valid or not

$$(\exists x \cdot P(x) \land Q(x)) \iff (\exists x \cdot P(x)) \land (\exists x \cdot Q(x))$$

Exercises (2/2)

Show whether the following sentence is valid or not

$$(\exists x \cdot P(x) \vee Q(x)) \iff (\exists x \cdot P(x)) \vee (\exists x \cdot Q(x))$$

- Valid. Prove by contradiction that every model M of the LHS is a model of the RHS and vice versa.

Show whether the following FOL sentence is valid or not

$$(\exists x \cdot P(x) \wedge Q(x)) \iff (\exists x \cdot P(x)) \wedge (\exists x \cdot Q(x))$$

- Not valid. Prove by constructing a model M of the RHS that is not a model of the LHS. For example, S = {0,1}, M(P) = { 0 }, and M(Q) = { 1 }

Completeness, Compactness, Incompleteness

Gödel Completeness Theorem of FOL

- any (first-order) formula that is true in *all* models of a theory, must be logically deducible from that theory, and vice versa (every formula that is deducible from a theory is true in *all* models of that theory)

Corollary: Compactness Theorem

- A FOL theory G is SAT iff every finite subset G' of G is SAT
- A set G of FOL sentences is UNSAT iff exists a finite subset G' of G that is UNSAT

Incompleteness of FOL Theories

- A theory is *consistent* if it is impossible to prove both *p* and *~p* for any sentence *p* in the signature of the theory
- A theory is *complete* if for every sentence *p* it includes either *p* or *~p*
- There are FOL theories that are consistent but incomplete

https://terrytao.wordpress.com/2009/04/10/the-completeness-and-compactness-theorems-of-first-order-logic/

https://plato.stanford.edu/entries/goedel-incompleteness/