COL:750/7250

Foundations of Automatic Verification

Instructor: Priyanka Golia

Course Webpage

https://priyanka-golia.github.io/teaching/COL-750-COL7250/index.html

	Operate On	**And Produce**
Connectives ($\leftrightarrow$, $\rightarrow$, $\wedge$, . .)	Propositions	A Proposition
Predicates	Objects	A Proposition
Functions	Objects	An Object

First Order Logic (FOL): Syntax

Well-Formed Formula (wff) of FOL are composed of six types of symbols (not including Parenthesis).

1. Constant symbols — representing objects.
2. Functions symbols — functions from pre-specified number of objects to an object.
3. Predicate symbols — more like specify properties to objects. Have specified arity. Zero arity predicate symbols are treated as propositional symbols.
4. Variable symbols — will be used to quantify over objects.
5. Universal and existential quantifiers — will be used to indicate the type of quantification.
6. Logical connectives and negation.

First Order Logic (FOL): Syntax

Formula -> Atomic Formula
| Formula Connective Formula
| Quantifier Variable Formula
| ¬ Formula
| (Formula)

Connective -> $\leftrightarrow$ | $\wedge$ | $\vee$ | $\rightarrow$
Quantifier -> $\forall$ | $\exists$

Atomic Formula -> $P(T_1, \ldots, T_n)$ where
$P \in Predicates$, T_i are Terms, n is arity.

Term -> c, where $c \in$ CONST.
| v, where $v \in VAR$
| $F(T_1, \ldots, T_n)$, where $F \in$ Functions, T_i are Terms,
n is arity of F.

First Order Logic (FOL): Syntax

Is it a WFF?

$TallerThan(John, Fatherof(John)) \wedge TallerThan(Fatherof(Fatherof(John)), John)$.

Yes, notice, Term is recursive.

Term -> c, where $c \in$ CONST.

| v, where $v \in VAR$

| $F(T_1, \ldots, T_n)$, where $F \in$ Functions, T_i are Terms,

n is arity of F.

First Order Logic (FOL): Additional Terminology

Ground Terms — Terms without variables. Refers to Objects. John, Fatherof(John)

Ground Formulas — Formulas without variables.

$$TallerThan(John, Fatherof(John)) \wedge TallerThan(Fatherof(Fatherof(John)), John) .$$

Closed Formulas — formulas in which all variables are associated with quantifier.

$$\forall x \; Number(x) \rightarrow Number(+(x,1))$$

$\forall x \; GreaterThan(x, y) \rightarrow LessThan(y, x)$ Y is not associated with quantifier.

Free variables — variables in a formula that don't have any quantifier. Typically free variables are treated as being implicitly universally quantified variables.

First Order Logic (FOL): Additional Terminology

All Birds can Fly.

$$\forall x\ (Bird(x) \rightarrow Fly(x))$$

Not all Birds can Fly.

$$\neg(\forall x\ (Bird(x) \rightarrow Fly(x)))$$

$$\equiv \exists x\ (Bird(x) \land \neg Fly(x))$$

All Birds cannot Fly.

$$\forall x\ (Bird(x) \rightarrow \neg Fly(x))$$

$$\equiv \neg(\exists x\ (Bird(x) \land Fly(x)))$$

First Order Logic (FOL): Semantics

Models of FOL!

Model of FOL is a tuple $< D, I >$

D — non-empty domain of objects (set of objects, finite, infinite, uncountable)
I — Interpretation function.

Interpretation — assign a meaning.

If c is a constant symbol then $I(c)$ is an object in D.

Defined for all inputs:
Single output per input

If f is a function symbol of arity n, then $I(f)$ is a total function from $D^n \mapsto D$

If p is a predicate symbol of arity n, then $I(p)$ is a subset of D^n. If a tuple $O = < o_1, \ldots, o_n > \in I(p)$, then we say that p is True for tuple O.

First Order Logic (FOL): Semantics

D = {BOB, JOHN, NULL} Bob is taller than John.
John is father of Bob.

If c is a constant symbol then $I(c)$ is an object in D. $I(Bob) = BOB$

If f is a function symbol of arity n, then $I(f)$ is a total function from $D^n \mapsto D$

$I(FatherOf)(BOB) = JOHN$ $I(FatherOf)(JOHN) = NULL$. $I(FatherOf)(NULL) = NULL$.

If p is a predicate symbol of arity n, then $I(p)$ is a subset of D^n. If a tuple $O = < o_1, \ldots, o_n > \in I(p)$, then we say that p is True for tuple O.

$$I(TallenThan) = \{ < BOB, JOHN > \}$$

First Order Logic (FOL): Semantics

How do we handle variables?

Given a model $M = < D, I >$ and a variable x, and object $o \in D$,
Extended Model $M[x \rightarrow o]$ as a model that is identical to M, except that I is extended to interpret x as o.

$$\exists x \; TallerThan(x, FatherOf(x))$$

If we can find an object o in D such that following is True:

$$TallerThan(x, FatherOf(x))^{M[x \rightarrow o]}$$

First Order Logic (FOL): Semantics

$$F = TallerThan(x, FatherOf(x))$$

$$D = \{BOB, JOHN, NULL)$$

$$I(Bob) = \{BOB\},\ I(John) = \{JOHN\},\ I(NULL) = \{NULL\}$$

$$I(FatherOf)(BOB) = \{JOHN\},\ I(FatherOf)(JOHN) = \{NULL\},\ I(FatherOf)(NULL) = \{NULL\}$$

$$I(TallerThan) = < BOB, JOHN >$$

Is F True, with respect to M<D,I>, where variable assignment σ = <John>?

First Order Logic (FOL): Semantics

How do we define the meaning of terms and formulas relative to a given model $M = < D, I >$

Notation: Interpretation of a string(terms/formula) F relative to a model M, and an assignment σ by $F^{M,\sigma}$

Interpreting Terms:

If t is a constant or a variable, then we have:

$t^{M,\sigma} = I(t)$ $\quad$ $x^{M,\sigma}$= I(John) = JOHN.

If t is a function $f(t_1, \ldots, t_n)$, then we have:

$$t^{M,\sigma} = I(f)(t_1^{M,\sigma}, \ldots, t_n^{M,\sigma})$$

$$FatherOf(x)^{M,\sigma} = I(FatherOf)(x^{M,\sigma})$$

$$FatherOf(x)^{M,\sigma} = I(FatherOf)(JOHN)$$

$$FatherOf(x)^{M,\sigma} = NULL$$

First Order Logic (FOL): Semantics

$x^{M,\sigma}$ = I(John) = JOHN. $FatherOf(x)^{M,\sigma} = NULL$

Interpreting Formulas:

1. Atomic Formulas F of the form $p(t_1, \ldots, t_m)$

$$F^{M,\sigma} = \begin{cases} \text{True if } < t_1^{M,\sigma}, \ldots, t_n^{M,\sigma} > \in I(p) \\ \text{False otherwise.} \end{cases}$$

$$TallerThan^{F,\sigma} = < JOHN, NULL >$$

$TallerThan^{M,\sigma} \notin I(TallerThan)$, $F^{M,\sigma}$ is False.

First Order Logic (FOL): Semantics

Interpreting Formulas:

1. Atomic Formulas F of the form $p(t_1, \ldots, t_n)$

$$F^{M,\sigma} = \begin{cases} \text{True if } < t_1^{M,\sigma}, \ldots, t_n^{M,\sigma} > \in I(p) \\ \text{False otherwise.} \end{cases}$$

2. If F is of the form $F_1 \ o \ F_2$ where o is logical connective:

$$F^{M,\sigma} = F_1^{M,\sigma} \ o \ F_2^{M,\sigma}$$

3. If F is of the form $\neg F_1$:

$$F^{M,\sigma} = \neg F_1^{M,\sigma}$$

First Order Logic (FOL): Semantics

4. If F is of the form $\exists x F_1$

$$F^{M,\sigma} = \begin{cases} \text{True if there exists an } o \in D \text{ such that } F_1^{M,\sigma[x \to o]} \text{ is True} \\ \text{False otherwise.} \end{cases}$$

5. If F is of the form $\forall x F_1$

$$F^{M,\sigma} = \begin{cases} \text{True if for all } o \in D,\ F_1^{M[x \to o]} \text{ is True} \\ \text{False otherwise.} \end{cases}$$

First Order Logic (FOL): Semantics

$$F = \exists x\; TallerThan(x, FatherOf(x))$$

We need to find a model M such that following is True:

$$[\exists x\; TallerThan(x, FatherOf(x))]^M$$

This is true iff we can find an object o in D such that:

$$TallerThan(x, FatherOf(x))^{M[x \to o]}$$

BOB is such an object.

How about $F = \forall x\; TallerThan(x, FatherOf(x))$?

First Order Logic (FOL): Semantics

$$F = \forall x \; TallerThan(x, FatherOf(x))$$

We need to find a model M such that following is True:

$$[\forall x \; TallerThan(x, FatherOf(x))]^M$$

This is true iff for all objects o in D the following is True:

$$TallerThan(x, FatherOf(x))^{M[x \to o]}$$

We saw that $TallerThan(x, FatherOf(x))^{M[x \to JOHN]}$ is False.

$F = \forall x \; TallerThan(x, FatherOf(x))$ is False.

First Order Logic (FOL): Semantics

Assignment: For a domain D is a function $\sigma : X \mapsto D$

Where X is set of variables of formula

Given M = (D,I) and given an assignment σ, satisfaction relation $M, \sigma \models F$ is follows:

$M, \sigma \models \top$

$M, \sigma \not\models \bot$

$M, \sigma \models P(t_1, \ldots, t_n)$ — iff $I(P)((t_1^M, \ldots, t_n^M)^\sigma) = 1$

$M, \sigma \models \neg F$ — iff $M, \sigma \not\models F$

$M, \sigma \models F \wedge G$ — iff $M, \sigma \models F$ and $M, \sigma \models G$

$M, \sigma \models F \vee G$ — iff $M, \sigma \models F$ or $M, \sigma \models G$

$M, \sigma \models F \rightarrow G$ — iff $M, \sigma \not\models F$ or $M, \sigma \models G$

$M, \sigma \models \forall x F$ — iff $M, \sigma[x \mapsto a] \models F$ for all $a \in D$

$M, \sigma \models \exists x F$ — iff $M, \sigma[x \mapsto a] \models F$ for some $a \in D$

First Order Logic (FOL): analogy with Propositional Logic

Truth table in propositional logic is similar to Model $M = < D, I >$ in FOL

Truth table consists of various truth assignments (σ) and to check if $\sigma \models F$, we need to check if $F(\sigma) = 1$ in truth table. Similarly in FOL, we need to check if $I^{M,\sigma}$ is 1 or not!

Given a formula, the truth table is fixed, however in FOL, model M depends on the Domain. We can have $M_1 = < D_{real}, I >$, $M_2 = < D_{int}, I >$, $\ldots, \ldots$

First Order Logic (FOL): Validity and Satisfiability

When $M, \sigma \models F$, we say that M satisfies F with σ

A formula F is

Valid — iff $M, \sigma \models F$ holds for all models M and assignments σ.

Satisfiable — iff there is some model M, and some assignment σ such that $M, \sigma \models F$

Unsatisfiable — iff it is not satisfiable

True — F is called True in M, iff some assignment σ in M, $M, \sigma \models F$

First Order Logic (FOL): Validity and Satisfiability

$\forall x(x = x)$ Valid.

$\exists x(x \neq x)$ Unsatisfiable.

$\exists x P(x)$ Depends on given M, σ . Suppose in a given M, I(P) is empty.
Then, in that M, Formula is False.
but, it may happen that there exists another M, under which it might be True.

First Order Logic (FOL): Validity and Satisfiability

Decidability — a solution to a decision problem is an algorithm that takes problem as input, and always terminates, producing a correct "yes" or "no" output

Valid — iff $M, \sigma \models F$ holds for all models M and assignments σ.

Satisfiable — iff there is some model M, and some assignment σ such that $M, \sigma \models F$

The decision problem of validity of FOL is **undecidable** (given any FOL formula F)

The decision problem of of FOL is **undecidable** (given any FOL formula F)

First Order Logic (FOL): Equivalent Formulas

F and G are called equivalent to each other if and only if:

For each model and assignment (M, σ), if $M, \sigma \models F$, then $M, \sigma \models G$ (notation $F \models G$)

and for each model and assignment (M', σ') if $M', \sigma' \models G$, then $M', \sigma' \models F$ (notation $G \models F$)

Exercise: Is $\neg \forall x P(x) \equiv \exists x \neg P(x)$

Intro to SMT: Satisfiability Modulo Theory

FOL: grammar for a rational abstract thinking

FOL: Doesn't have a knowledge of any specific matter.

Theory = Subject Knowledge + FOL

Model M <D = set of natural numbers>

— we can consider only theory of natural numbers.

— we also consider the set of valid sentences over natural numbers.

For example: $\forall x \; x + 1 \neq 0$

Intro to SMT: Satisfiability Modulo Theory

Theory = Subject Knowledge + FOL

Model M <D = set of natural numbers>

— we can consider only theory of natural numbers.

— we also consider the set of valid sentences over natural numbers.

For example: $\forall x \; x + 1 \neq 0$

A theory T is a set of sentences closed under implications

If $T \to F$, then $F \in T$

Intro to SMT: Satisfiability Modulo Theory

Is F = $\exists x, x > 0$ satisfiable? Valid ? In FOL?

Yes, it is satisfiable!

$M :< D = \mathbb{N}, I >$ F is satisfiable.

No, it is not valid, $M :< D = \mathbb{Z}^{-}, I >$

A formula F is T-satisfiable if there is model M such that $M \vDash T \cup F$.

We write T -satisfiability as $M \vDash_T F$.

T: set of true sentences in arithmetic over natural numbers.

Is $T \cup F$ satisfiable ?, we need to restrict our domain to set of natural numbers, and assume the knowledge of natural number arithmetic like $\forall x\ x > 0, \forall x\ x + 1 \neq 0$

Yes, it is satisfiable!

$M \vDash_T F$

Intro to SMT: Satisfiability Modulo Theory

T: set of true sentences in arithmetic over natural numbers.

Is $T \cup F$ satisfiable ?, we need to restrict our domain to set of natural numbers, and assume the knowledge of natural number arithmetic like $\forall x\ x > 0, \forall x\ x + 1 \neq 0$

Is F = $\exists x, x > 0$ T-satisfiable?

Yes, it is T-satisfiable!

$M \vDash_T F$

Also, $T \vDash F$

A formula F is T-valid if $T \vDash F$. We write T -validity as $\vDash_T F$

Intro to SMT: Satisfiability Modulo Theory

Is F = $\exists x, x < 0$ satisfiable? Valid ? In FOL?

Yes, it is satisfiable!

$M :< D = \mathbb{Z}, I >$ F is satisfiable.

No, it is not valid, $M :< D = \mathbb{N}, I >$

T: set of true sentences in arithmetic over natural numbers.

Is F = $\exists x, x < 0$ T-satisfiable? T-Valid ?

No, it is unsatisfiable, $\not\models T_{\mathbb{N}} \cup F$

https://smt-lib.org/logics.shtml