COL:750

Foundations of Automatic Verification

Instructor: Priyanka Golia

Course Webpage

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

Imagine a smart home with multiple devices (lights, fans, thermostats) spread across different rooms (kitchen, bedroom, living room). A control system needs to ensure certain rules are satisfied, such as:

1. All lights should be off when no one is in the room.
2. If the temperature is above 30°C, the fan should turn on.

Assume: m many person, n many lights.

Imagine a smart home with multiple devices (lights, fans, thermostats) spread across different rooms (kitchen, bedroom, living room). A control system needs to ensure certain rules are satisfied, such as:

1. All lights should be off when no one is in the room.
2. If the temperature is above 30°C, the fan should turn on.

$P = \{p_1, \ldots, p_m\}, L = \{L_1, \ldots, L_n\}$

Assume: m many person, n many lights.

Let p_i represents that i^{th} person is in the room, and L_j represents that j^{th} light is on.

$\neg(p_1 \vee p_2 \vee \ldots \vee p_m) \rightarrow (\neg L_1 \wedge \neg L_2 \wedge \ldots \wedge \neg L_n)$

Clauses n many, each clause has m+1 variables.

$\equiv ((p_1 \vee p_2 \vee \ldots \vee p_m) \vee \neg L_1) \wedge ((p_1 \vee p_2 \vee \ldots \vee p_m) \vee \neg L_2) \wedge \ldots \wedge ((p_1 \vee p_2 \vee \ldots \vee p_m) \vee \neg L_n)$

$P = \{p_1, \ldots, p_m\}, L = \{L_1, \ldots, L_n\}$

Assume: m many person, n many lights.

Let p_i represents that i^{th} person is in the room, and L_j represents that j^{th} light is on.

$$\neg(p_1 \vee p_2 \vee \ldots \vee p_m) \to (\neg L_1 \wedge \neg L_2 \wedge \ldots \wedge \neg L_n)$$

Clauses n many, each clause has m+1 variables.

$$\equiv ((p_1 \vee p_2 \vee \ldots \vee p_m) \vee \neg L_1) \wedge ((p_1 \vee p_2 \vee \ldots \vee p_m) \vee \neg L_2) \wedge \ldots \wedge ((p_1 \vee p_2 \vee \ldots \vee p_m) \vee \neg L_n)$$

Repetition: writing separate formulas for each room.
As the number of rooms increases, the formula grows linearly.

No generalization: We cannot express the general rule "For any room, if no one is present, the light should be off" without enumerating each case.

First Order Logic (FOL)

FOL is a logical system for reasoning about properties of objects.

Predicates — describes properties of objects.

Functions — maps objects to one another.

Quantifiers — to reason about multiple objects

First Order Logic (FOL): Objects

"John is happy" as P
“Mary is happy” as Q

Propositional variables don’t provide any structure about what the proposition refers to or relationships between entities — how P and Q are related ?

Objects: It represent entities in a domain of discourse (things we want to reason about), such as people, numbers, or physical objects.

Objects are: John, and Marry.
Happy(John) — property “happy” is applied to John.
Happy(Mary) — property “happy” is applied to Mary.
Likes(Mary,John): "Mary likes John."

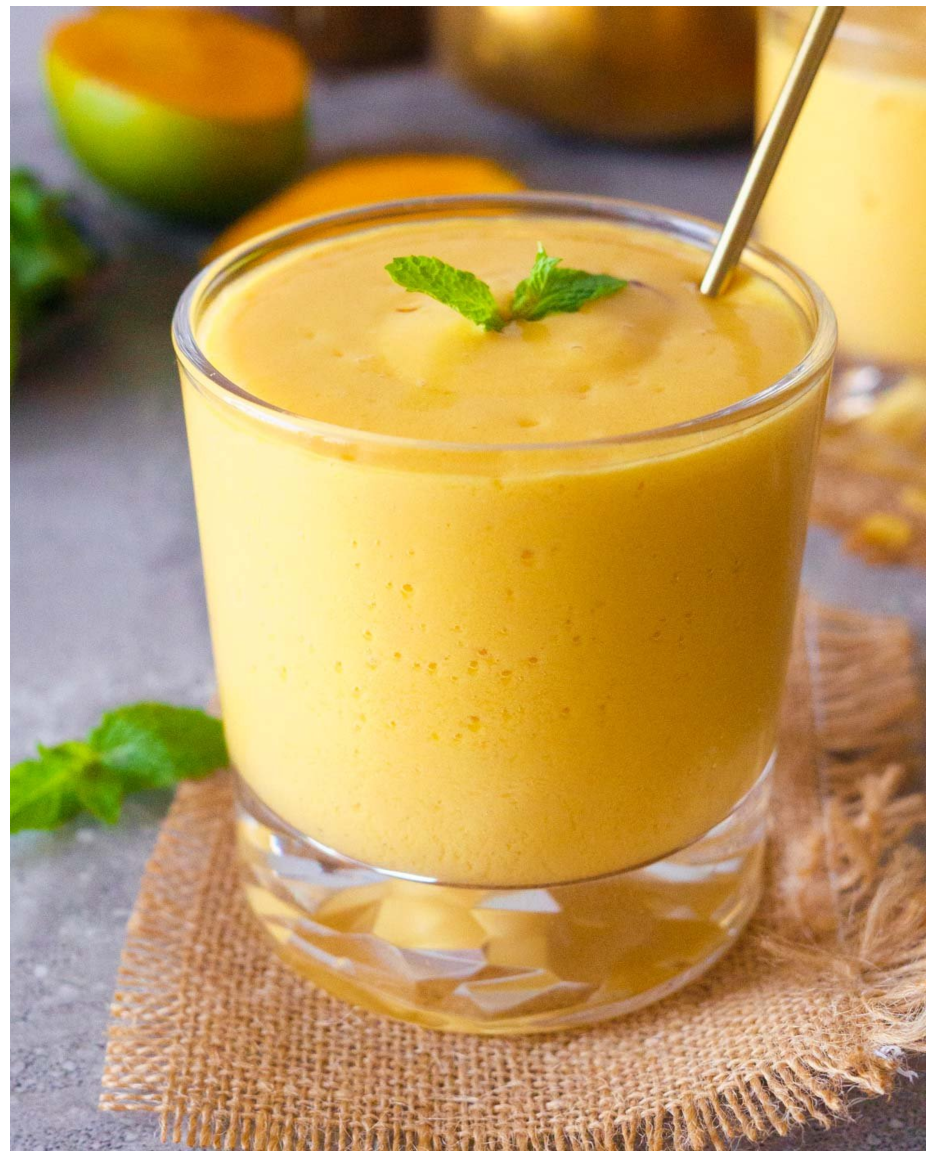
Objects allow FOL to express relationships, properties, and reasoning about entities.

First Order Logic (FOL): Predicates

$$Likes(You, Yogurt) \wedge Likes(You, Mango) \rightarrow Likes(You, MangoLassi)\ .$$

Objects: { You, Yogurt, Mango, MangoLassi}.

Predicates: $Likes(Obj_1, Obj_2) \mapsto \{0,1\}$



Predicates takes objects as an arguments and evaluate to True or False.

Predicates — describes properties of objects.
Happy(John)
Cute(John)

First Order Logic (FOL): Functions

FavoriteMovieOf(You) ≠ FavoriteMovieOf(Date) ∧
StarOf(FavoriteMovieOf(You)) = StarOf(FavoriteMovieOf(Date))

Functions take objects as an argument and return objects associated with it.

Medianof(x,y,z), +(x,y), Wife(John).

As with predicates, functions can take in any number of arguments, but always return a single object.

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

First Order Logic (FOL): Quantifiers

There is a number which is both prime and even.

Variables: x.
Predicates: Even(x), Prime(x)

$$\exists x(Even(x) \wedge Prime(x))$$

There is someone who is taller than I am and weighs more than I do.

Objects: me, Variable: x
Predicates: Taller(x,me), WeighsMore(x,me)

$$\exists x Taller(x, me) \wedge WeighMore(x, me)$$

Existential Quantifier (∃): Expresses the existence of at least one element for which a statement is true.

First Order Logic (FOL): Quantifiers

For every number x, adding 0 to results in x itself.

Variable: x

Function: $+(x,0)$

Predicate: $=(x, +(x,0))$

$\forall x = (x, +(x,0))$

For all even numbers x, x is divisible by 2.

Variable: x

Function: $mod(x,2)$

Predicate: $even(x)$, $=(mod(x,2),0)$

$\forall x \ (even(x) \rightarrow =(mod(x,2),0))$

Universal Quantifier (∀): Expresses generalization across all elements.

First Order Logic (FOL): Quantifiers

Scope of Quantifiers: refers to the part of the formula where the quantifier applies to the variable it introduces.

Bound Variable: A variable is bound if it lies within the scope of a quantifier.

Free Variable: A variable is free if it is not within the scope of any quantifier.

$\forall x\ P(x) \to Q(y)$. x is bounded and y is free

Nested Quantifiers: When quantifiers are nested, the scope of the inner quantifier is restricted by the outer quantifier.

$\forall x((\exists y P(x, y)) \to Q(x))$

Scope of $\forall x$ is entire formula.

Scope of $\exists y$ is limited to $P(x, y)$

First Order Logic (FOL): Quantifiers

When multiple quantifiers share overlapping scopes, their interactions can lead to significant differences in meaning.

$$\forall x \exists y P(x, y)$$

For every x, there exists a y such that P(x, y).

Each person can know a different language, as long as they know at least one language.

$$\exists y \forall x P(x, y)$$

There exists a y, for all x such that P(x, y).

There is a single language that everyone knows.

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 -> $\leftrightarrow \mid \wedge \mid \vee \mid \rightarrow$

Quantifier -> $\forall \mid \exists$

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.

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

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

All Birds cannot Fly.

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

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

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.

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

Course Webpage

Thanks!