CS 357: Advanced Topics in Formal Methods
Fall 2019

Lecture 6

Aleksandar Zeljić
(materials by Clark Barrett)
Stanford University

First-Order Logic: Motivation

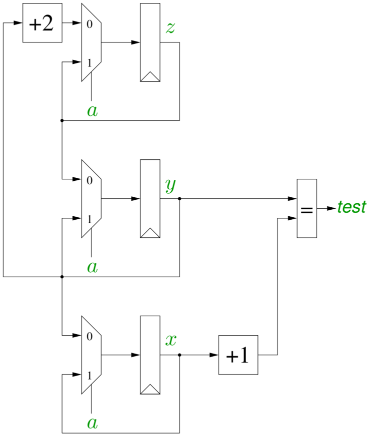
Propositional logic is not powerful enough for many applications.

For example, propositional logic cannot reason about natural numbers directly.

In general, to reason about infinite domains or to express properties which are more abstract, a more expressive logic is required.

First-order logic is the most common logic of choice for handling tasks that require more power than that offered by propositional logic.

Example



Example

2-bit counter property specification:
$z1 \leftrightarrow \neg x1 \wedge z0 \leftrightarrow x0 \wedge y1 \leftrightarrow (x1 \oplus x0) \wedge y0 \leftrightarrow \neg x0$

n-bit counter specification requires a formula of size $O(n)$.

Using first-order logic, we can express the specification using a formula whose size is constant for all n:

$z = x +_{[2^n]} 2 \wedge y = x +_{[2^n]} 1$

Here, the intended meaning is that variables x, y, and z range over the set $[0..2^n - 1]$ and $+_{[2^n]}$ indicates addition modulo 2^n.

When using first-order logic, part of our task is to specify the meaning of the symbols we are using.

First-Order Logic: Syntax

As with propositional logic, expressions in first-order logic are made up of sequences of symbols.

Symbols are divided into *logical symbols* and *non-logical symbols* or *parameters*.

Logical Symbols

- Parentheses: (,)
- Propositional connectives: $\rightarrow$, $\neg$
- Variables: v_1, v_2, $\ldots$
- Universal quantifier: $\forall$

Parameters

- Equality symbol (optional): $=$
- Predicate symbols: e.g. $p(x)$, $x > y$
- Constant symbols: e.g. 0, $John$, π
- Function symbols: e.g. $f(x)$, $x + y$, $x +_{[2]} y$

First-Order Logic: Syntax

Abbreviations

- Other propositional connectives: $\vee$, $\wedge$, $\leftrightarrow$
- Existential quantifier: $\exists x\, p(x) \Leftrightarrow \neg\forall x\, \neg p(x)$

Each predicate and function symbol has an associated *arity*, a natural number indicating how many arguments it takes.

Equality is a special predicate symbol of arity 2.

Constant symbols can also be thought of as functions of arity 0.

A *first-order language* must first specify its parameters.

First-Order Languages: Examples

Propositional Logic

- Equality: *no*
- Predicate symbols: A_1, A_2, . . .
- Constant symbols: *none*
- Function symbols: *none*

Set Theory

- Equality: *yes*
- Predicate symbols: $\in$
- Constant symbols: $\emptyset$
- Function symbols: *none*

First-Order Languages: Examples

Elementary Number Theory

- Equality: *yes*
- Predicate symbols: $<$
- Constant symbols: 0
- Function symbols: S (successor), $+$, $\times$, *exp*

First-Order Logic: Terms

The first important concept on the way to defining well-formed formulas is that of *terms*.

For each function symbol f of arity n, we define a term-building operation $\mathcal{F}_f$:

$$\mathcal{F}_f(\alpha_1, \ldots, \alpha_n) = f\alpha_1, \ldots, \alpha_n$$

Note that we are using prefix notation to avoid ambiguity.

The set of *terms* is the set of expressions generated from the constant symbols and variables by the $\mathcal{F}_f$ operations.

Terms are expressions which name *objects*.

Theorem

The set of terms is freely generated from the set of variables and constant symbols by the $\mathcal{F}_f$ operations.

First-Order Logic: Formulas

Atomic Formulas

An *atomic formula* is an expression of the form: $Pt_1, \ldots, t_n$ where P is a predicate symbol of arity n and t_1,...,t_n are terms.

If the language includes the equality symbol, we consider the equality symbol as a predicate of arity 2.

Formulas

We define the following formula-building operations:

- $\mathcal{E}_{\neg}(\alpha) = (\neg\alpha)$
- $\mathcal{E}_{\rightarrow}(\alpha, \beta) = (\alpha \rightarrow \beta)$
- $\mathcal{Q}_i(\alpha) = \forall\, v_i\; \alpha$

The set of *well-formed formulas* (*wffs* or just *formulas*) is the set of expressions generated from the atomic formulas by the operations $\mathcal{E}_{\neg}$, $\mathcal{E}_{\rightarrow}$, and $\mathcal{Q}_i$ $i = 1, 2, \ldots$.

This set is also freely generated.

Formula Examples

In the language of elementary number theory introduced above, which of the following are terms?

1. v_6

Formula Examples

In the language of elementary number theory introduced above, which of the following are terms?

1. v_6 *yes*
2. $v_2 + v_3$

Formula Examples

In the language of elementary number theory introduced above, which of the following are terms?

1. v_6 *yes*
2. $v_2 + v_3$ *no*
3. $+v_2v_3$

Formula Examples

In the language of elementary number theory introduced above, which of the following are terms?

1. v_6 *yes*
2. $v_2 + v_3$ *no*
3. $+v_2v_3$ *yes*

atomic formulas?

1. $= exp + v_1 0 v_2 S v_3$

Formula Examples

In the language of elementary number theory introduced above, which of the following are terms?

1. v_6 *yes*
2. $v_2 + v_3$ *no*
3. $+v_2 v_3$ *yes*

atomic formulas?

1. $= exp + v_1 0 v_2 S v_3$ *yes*: $(v_1 + 0)^{v_2} = S(v_3)$
2. $\neg = v_2 v_3$

Formula Examples

In the language of elementary number theory introduced above, which of the following are terms?

1. v_6 *yes*
2. $v_2 + v_3$ *no*
3. $+v_2 v_3$ *yes*

atomic formulas?

1. $= exp + v_1 0 v_2 S v_3$ *yes*: $(v_1 + 0)^{v_2} = S(v_3)$
2. $\neg = v_2 v_3$ *no*

well-formed formulas?

1. $\neg = v_2 v_3$

Formula Examples

In the language of elementary number theory introduced above, which of the following are terms?

1. v_6 *yes*
2. $v_2 + v_3$ *no*
3. $+v_2 v_3$ *yes*

atomic formulas?

1. $= exp + v_1 0 v_2 S v_3$ *yes*: $(v_1 + 0)^{v_2} = S(v_3)$
2. $\neg = v_2 v_3$ *no*

well-formed formulas?

1. $\neg = v_2 v_3$ *no*
2. $(\neg = v_2 v_3)$

Formula Examples

In the language of elementary number theory introduced above, which of the following are terms?

1. v_6 *yes*
2. $v_2 + v_3$ *no*
3. $+v_2 v_3$ *yes*

atomic formulas?

1. $= exp + v_1 0 v_2 S v_3$ *yes*: $(v_1 + 0)^{v_2} = S(v_3)$
2. $\neg = v_2 v_3$ *no*

well-formed formulas?

1. $\neg = v_2 v_3$ *no*
2. $(\neg = v_2 v_3)$ *yes*: $v_2 \neq v_3$
3. $\times 0 v_1$

Formula Examples

In the language of elementary number theory introduced above, which of the following are terms?

1. v_6 *yes*
2. $v_2 + v_3$ *no*
3. $+v_2v_3$ *yes*

atomic formulas?

1. $= exp + v_1 0 v_2 S v_3$ *yes*: $(v_1 + 0)^{v_2} = S(v_3)$
2. $\neg = v_2v_3$ *no*

well-formed formulas?

1. $\neg = v_2v_3$ *no*
2. $(\neg = v_2v_3)$ *yes*: $v_2 \neq v_3$
3. $\times 0 v_1$ *no*
4. $\forall v_1 \; = \times 0 v_1 v_1$

Formula Examples

In the language of elementary number theory introduced above, which of the following are terms?

1. v_6 *yes*
2. $v_2 + v_3$ *no*
3. $+v_2 v_3$ *yes*

atomic formulas?

1. $= exp + v_1 0 v_2 S v_3$ *yes*: $(v_1 + 0)^{v_2} = S(v_3)$
2. $\neg = v_2 v_3$ *no*

well-formed formulas?

1. $\neg = v_2 v_3$ *no*
2. $(\neg = v_2 v_3)$ *yes*: $v_2 \neq v_3$
3. $\times 0 v_1$ *no*
4. $\forall v_1 = \times 0 v_1 v_1$ *yes*: $\forall v_1 \,(0 \times v_1 = v_1)$

Free and Bound Variables

We define by recursion what it means for a variable x to *occur free* in a *wff* α:

- If α is an atomic formula, then x occurs free in α iff x occurs in α.
- x occurs free in $(\neg\alpha)$ iff x occurs free in α.
- x occurs free in $(\alpha \to \beta)$ iff x occurs free in α or in β.
- x occurs free in $\forall v_i\, \alpha$ iff x occurs free in α and $x \neq v_i$.

To make this definition precise, we would need to define a recursive function and make use of the recursion theorem and the fact that *wffs* are freely generated.

If $\forall v_i$ appears in α, then v_i is said to be *bound* in α.

> *Note that a variable can both occur free and be bound in α. Because this can be confusing, we typically require the set of free and bound variables to be disjoint.*

If no variable occurs free in a *wff* α, then α is a *sentence*.

First-Order Logic: Semantics

In propositional logic, the truth of a formula was determined by a *truth assignment* over the propositional symbols.

In first-order logic, we use a *model* (also known as a *structure*) to determine the truth of a formula.

A *signature* is a set of non-logical symbols (predicates, constants, and functions). Given a signature Σ, a model M of Σ consists of the following:

1. A nonempty set called the *domain* of M, written $dom(M)$. Elements of the domain are called elements of the model M.
2. A mapping from each constant c in Σ to an element c^M of M.
3. A mapping from each n-ary function symbol f in Σ to f^M, an n-ary function from $[dom(M)]^n$ to $dom(M)$.
4. A mapping from each n-ary predicate symbol p in Σ to $p^M \subseteq [dom(M)]^n$, an n-ary relation on the set $dom(M)$.

Example

Consider the signature corresponding to the language of set theory which has a single predicate symbol $\in$ and a single constant symbol $\emptyset$.

A possible model M for this signature has $dom(M) = \mathcal{N}$, the set of natural numbers, $\in^M = <$, and $\emptyset^M = 0$.

Now consider the sentence $\exists x \forall y \neg y \in x$.

Example

Consider the signature corresponding to the language of set theory which has a single predicate symbol $\in$ and a single constant symbol $\emptyset$.

A possible model M for this signature has $dom(M) = \mathcal{N}$, the set of natural numbers, $\in^M = <$, and $\emptyset^M = 0$.

Now consider the sentence $\exists x \forall y \neg y \in x$.

What does this sentence mean in this model?

Example

Consider the signature corresponding to the language of set theory which has a single predicate symbol $\in$ and a single constant symbol $\emptyset$.

A possible model M for this signature has $dom(M) = \mathcal{N}$, the set of natural numbers, $\in^M = <$, and $\emptyset^M = 0$.

Now consider the sentence $\exists x \forall y \neg y \in x$.

What does this sentence mean in this model?

The translation of the sentence in the model M is that there is a natural number x such that no other natural number is smaller than x.

Example

Consider the signature corresponding to the language of set theory which has a single predicate symbol $\in$ and a single constant symbol $\emptyset$.

A possible model M for this signature has $dom(M) = \mathcal{N}$, the set of natural numbers, $\in^M = <$, and $\emptyset^M = 0$.

Now consider the sentence $\exists x \forall y \neg y \in x$.

What does this sentence mean in this model?

The translation of the sentence in the model M is that there is a natural number x such that no other natural number is smaller than x.

Is this sentence true in the model?

Example

Consider the signature corresponding to the language of set theory which has a single predicate symbol $\in$ and a single constant symbol $\emptyset$.

A possible model M for this signature has $dom(M) = \mathcal{N}$, the set of natural numbers, $\in^M = <$, and $\emptyset^M = 0$.

Now consider the sentence $\exists x \forall y \neg y \in x$.

What does this sentence mean in this model?

The translation of the sentence in the model M is that there is a natural number x such that no other natural number is smaller than x.

Is this sentence true in the model?

Since 0 has this property, the sentence is true in this model.

First-Order Logic: Semantics

We will often use a shorthand when discussing both signatures and models. The signature shorthand lists each symbol in the signature.

The model shorthand lists the domain and the interpretation of each symbol of the signature.

The signature for set theory can thus be described as $(\in, \emptyset)$, and the above model as $(\mathcal{N}, <, 0)$.

First-Order Logic: Semantics

Given a model M, a *variable assignment* s is a function which assigns to each variable an element of M.

Given a wff ϕ, we say that M *satisfies* ϕ with s and write $\models_M \phi[s]$ if ϕ is true in the model M with variable assignment s.

To define this formally, we first define the extension $\overline{s} : T \to dom(M)$, a function from the set T of all terms into the domain of M:

1. For each variable x, $\overline{s}(x) = s(x)$.
2. For each constant symbol c, $\overline{s}(c) = c^M$.
3. If $t_1, \ldots, t_n$ are terms and f is an n-ary function symbol, then $\overline{s}(ft_1, \ldots, t_n) = f^M(\overline{s}(t_1), \ldots, \overline{s}(t_n))$.

The existence of a unique such extension $\overline{s}$ follows from the recursion theorem and the fact that the terms are freely generated.

Note that $\overline{s}$ depends on both s and M.

First-Order Logic: Semantics

Atomic Formulas

1. $\models_M = t_1 t_2[s]$ iff $\overline{s}(t_1) = \overline{s}(t_2)$.
2. For an n-ary predicate symbol P,
   $\models_M Pt_1, \ldots, t_n[s]$ iff $\langle \overline{s}(t_1), \ldots, \overline{s}(t_n) \rangle \in P^M$.

Other wffs

1. $\models_M (\neg \phi)[s]$ iff $\not\models_M \phi[s]$.
2. $\models_M (\phi \rightarrow \psi)[s]$ iff $\not\models_M \phi[s]$ or $\models_M \psi[s]$.
3. $\models_M \forall x \, \phi[s]$ iff $\models_M \phi[s(x|d)]$ for every $d \in dom(M)$.

$s(x|d)$ signifies the function which is the same as s everywhere except at x where its value is d.

Again, the well-formedness of this definition depends on the recursion theorem and the fact that *wffs* are freely generated.

Logical Definitions

Suppose Σ is a signature. A *Σ-formula* is a well-formed formula whose non-logical symbols are contained in Σ.

Let Γ be a set of Σ-formulas. We write $\models_M \Gamma[s]$ to signify that $\models_M \phi[s]$ for every $\phi \in \Gamma$.

If Γ is a set of Σ-formulas and ϕ is a Σ-formula, then Γ *logically implies* ϕ, written $\Gamma \models \phi$, iff for every model M of Σ and every variable assignment s, if $\models_M \Gamma[s]$ then $\models_M \phi[s]$.

We write $\psi \models \phi$ as an abbreviation for $\{\psi\} \models \phi$.

ψ and ϕ are *logically equivalent* (written $\psi \models\!\!\!=\!\!\!\mid \phi$) iff $\psi \models \phi$ and $\phi \models \psi$.

A Σ-formula ϕ is *valid*, written $\models \phi$ iff $\emptyset \models \phi$ (i.e. $\models_M \phi[s]$ for every M and s).