CS228 Logic for Computer Science 2022

Lecture 14: Handling first-order logic

Instructor: Ashutosh Gupta

IITB, India

Compile date: 2022-08-07

Topic 14.1

Supporting definitions

Clubbing similar quantifiers

If we have a chain of same quantifier then we write the quantifier once followed by the list of variables.

Example 14.1

- $\forall z, x. \exists y. G(x, y, z) = \forall z.(\forall x.(\exists y. G(x, y, z)))$
- $\exists z, x, y. G(x, y, z) = \exists z.(\exists x.(\exists y. G(x, y, z)))$

Subterm and subformulas

Definition 14.1

A term t is subterm of term t', if t is a substring of t'.

Exercise 14.1

- *Is $f(x)$ a subterm of $g(f(x), y)$?*
- *Is c a subterm of c?*
- *x is a subterm of $f(x)$*

Definition 14.2

A formula F is subformula of formula F', if F is a substring of F'.

Example 14.2

- *$G(x, y, z)$ is a subformula of $\forall z, x.\exists y.G(x, y, z)$*
- *$P(c)$ is a subformula of $P(c)$*
- *$\exists y.G(x, y, z)$ is a subformula of $\forall z, x.\exists y.G(x, y, z)$*

Closed terms and quantifier free

Definition 14.3

*A closed term is a term without any variable. Let $\hat{T}_{\mathbf{S}}$ be the set of closed **S**-terms.*

Sometimes closed terms are also referred as ground terms.

Example 14.3

Let $\mathbf{F} = \{f/1, c/0\}$. $f(c)$ is a closed term, and $f(x)$ is not, where x is a variable.

Exercise 14.2

Which of the following terms are closed with respect to $\mathbf{F} = \{f/1, g/2, c/0\}$?

- $g(c, y)$
- x
- c
- $f(g(c, c))$

Quantifier-free

Definition 14.4

A formula F is quantifier-free if there are no quantifiers in F.

Example 14.4

$H(c)$ *is a quantifier-free formula and* $\forall x.H(x)$ *is not a quantifier-free formula.*

Exercise 14.3

For signature $(\{f/1, c/0\}, \{H/1\})$, *which of the following are quantifier-free?*

- $\forall x.H(y)$
- $H(y) \lor \bot$
- $f(c)$
- $H(f(c))$

Free variables

Definition 14.5

A variable $x \in \text{Vars}$ *is* *free* *in formula* F *if*

- $F \in A_{\mathbf{S}}$: x *occurs in* F,
- $F = \neg G$: x *is free in* G,
- $F = G \circ H$: x *is free in* G *or* H, *for some binary operator* $\circ$, *and*
- $F = \exists y.G$ *or* $F = \forall y.G$: x *is free in* G *and* $x \neq y$.

Let $FV(F)$ *denote the set of free variables in* F.

Exercise 14.4

Is x *free?*

- $H(x)$
- $H(y)$
- $\forall x.H(x)$
- $x = y \Rightarrow \exists x.G(x)$

Sentence

Definition 14.6

In $\forall x.(G)$*, we say the quantifier* $\forall x$ *has scope* G *and bounds* x.
In $\exists x.(G)$*, we say the quantifier* $\exists x$ *has scope* G *and bounds* x.

Definition 14.7

A formula F *is a sentence if it has no free variable.*

Exercise 14.5

Which of the following formulas are sentence(s)?

- $H(x)$
- $\forall x.H(x)$
- $x = y \Rightarrow \exists x.G(x)$
- $\forall x.\exists y.\ x = y \Rightarrow \exists x.G(x)$

Topic 14.2

Understanding FOL semantics

No free variables

Definition 14.8

Let t be a closed term. $m(t) \triangleq m^{\nu}(t)$ *for any ν.*

If F is a sentence, ν has no influence in the satisfaction relation.(why?)

For sentence F, we say

- F is *true* in m if $m \models F$
- Otherwise, F is *false* in m.

Why nonempty domain?

We are required to have nonempty domain in the model. Why?

Example 14.5

Consider formula $\forall x.(H(x) \wedge \neg H(x))$.

Should any model satisfy the formula? *Nooooooooo..*

But, if we allow $m = \{\emptyset; H_m = \emptyset\}$ *then*

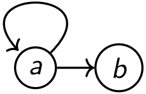
$$m \models \forall x.(H(x) \wedge \neg H(x)).$$

Due to this counter-intuitive behavior, the empty domain is disallowed.

Example: graph models

Example 14.6

Consider $\mathbf{S} = (\{\}, \{E/2\})$ *and* $m = (\{a, b\}; \{(a, a), (a, b)\})$*.*
m may be viewed as the following graph.



$m, \{x \to a\} \models E(x, x) \land \exists y.(E(x, y) \land \neg E(y, y))$

Exercise 14.6

Give another model and assignment that satisfy the above formula

Example : counting

Example 14.7

Let $\mathbf{S} = (\{\}, \{E/2\})$*. The following sentence is false in all the models with one element domain*

$$\forall x.\neg E(x,x) \land \exists x \exists y. E(x,y)$$

Exercise: counting

Exercise 14.7

Give a sentence that is true only in the models with more than two elements

Exercise 14.8 (important)

a. Give a sentence that is true only in infinite models
b. Do only finite models satisfy the negation of the sentence in (a)? If not, give an example of infinite model.

Exercise 14.9

a. Give a sentence that is true only in models with less than or equal to two element domains.
b. Can you answer (a) without using $=$?

A limit: Impossibility of expressing finite

Theorem 14.1

No FOL sentence can express that all satisfying models are finite.

Commentary: Proof of the above is not part of this course.

Topic 14.3

Substitution

Substitution

In first-order logic, we have terms and formulas. We need a more elaborate notion of substitution for terms.

Definition 14.9

A substitution σ is a map from Vars $\rightarrow T_{\mathbf{S}}$*. We will write $t\sigma$ to denote $\sigma(t)$.*

Definition 14.10

We say σ has finite support if only finite variables do not map to themselves. σ with finite support is denoted by $[t_1/x_1, .., t_n/x_n]$ or $\{x_1 \mapsto t_1, .., x_n \mapsto t_n\}$.

We may write a formula as $F(x_1, \dots, x_k)$, where variables $x_1, \dots, x_k$ play a special role in F.

$$\text{Let } F(t_1, \dots, t_n) \text{ be } F[t_1/x_1, \dots, t_n/x_n].$$

Commentary: We have seen substitution in propositional logic. Now in FOL, substitution is not a simple matter. We were replacing a formula by another. Now we need another kind of substitution that replaces terms.

Substitution on terms

Definition 14.11

For $t \in T_{\mathbf{S}}$*, let the following naturally define* $t\sigma$ *as extension of* σ*.*

- $c\sigma \triangleq c$
- $(f(t_1, \dots, t_n))\sigma \triangleq f(t_1\sigma, \dots, t_n\sigma)$

Example 14.8

Consider $\sigma = \{x \mapsto f(x, y), y \mapsto f(y, x)\}$

- $x\sigma = f(x, y)$
- $f(x, y)\sigma = f(f(x, y), f(y, x))$
- $(f(x, y)\sigma)\sigma = ?$
- $f(x, g(y))\{x \mapsto g(z), z \mapsto y\} = f(g(z), g(y))$

Substitution on atoms

We further extend the substitution σ to atoms.

Definition 14.12

For $F \in A_{\mathbf{S}}$, $F\sigma$ is defined as follows.

- $\top\sigma \triangleq \top$
- $\bot\sigma \triangleq \bot$
- $P(t_1, \dots, t_n)\sigma \triangleq P(t_1\sigma, \dots, t_n\sigma)$
- $(t_1 = t_2)\sigma \triangleq t_1\sigma = t_2\sigma$

Substitution projection

Sometimes, we may need to remove variable x from the support of σ.

Definition 14.13

Let $\sigma_x = \sigma[x \mapsto x]$.

Example 14.9

Consider $\sigma = \{x \mapsto f(x, y), y \mapsto f(y, x)\}$. $\sigma_x = \{y \mapsto f(y, x)\}$

Commentary: The need of the definition will be clear soon.

Substitution in formulas (Incorrect)

Now we extend the substitution σ to all the formulas.

Definition 14.14

For $F \in P_{\mathbf{S}}$, $F\sigma$ is defined as follows.

- $(\neg G)\sigma \triangleq \neg(G\sigma)$
- $(G \circ H)\sigma \triangleq (G\sigma) \circ (H\sigma)$ *for some binary operator* $\circ$
- $(\forall x.G)\sigma \triangleq \forall x.(G\sigma_x)$
- $(\exists x.G)\sigma \triangleq \exists x.(G\sigma_x)$

Example 14.10

- $(P(x) \Rightarrow \forall x.Q(x))\{x \mapsto y\} = (P(y) \Rightarrow \forall x.Q(x))$
- $(\exists y.\ x \neq y)\{x \mapsto z\} = (\exists y.\ z \neq y)$
- $(\exists y.\ x \neq y)\{x \mapsto y\} = (\exists y.\ y \neq y)$ ☹ *Undesirable!!!*

Some substitutions should be disallowed.

Commentary: Example: The following is not desirable. $\{x \mapsto 0, y \mapsto 0\} \models (\exists y.\ x \neq y)$. $\{x \mapsto 0, y \mapsto 0\} \not\models (\exists y.\ x \neq y)\{x \mapsto y\}$
The above näive definition of the substitution in formulas appears to be incorrect. In the next slide, we present the accepted definition. Please note that that the substitution is a syntactic operation. It does not provide any semantic guarantee. One could consider the above definition correct, but it would be a useless definition. The definition in the next slide is correct because it is useful.

Substitution in formulas(Correct)

Definition 14.15

σ is suitable with respect to formula G and variable x if for all $y \neq x$, if $y \in FV(G)$ then x does not occur in $y\sigma$.

Now we correctly extend the substitution σ to all formulas.

Definition 14.16

For $F \in \mathrm{P}_{\mathbf{S}}$, $F\sigma$ is defined as follows.

- $(\neg G)\sigma \triangleq \neg(G\sigma)$
- $(G \circ H)\sigma \triangleq (G\sigma) \circ (H\sigma)$ *for some binary operator* $\circ$
- $(\forall x.G)\sigma \triangleq \forall x.(G\sigma_x)$, *where σ is suitable with respect to G and x*
- $(\exists x.G)\sigma \triangleq \exists x.(G\sigma_x)$, *where σ is suitable with respect to G and x*

It is not a true restriction.
We will see later.

Commentary: We want the following natural property, which is only true it the substitution is defined.

For a variable z, a term t, and a formula F, if $m^\nu(z) = m^\nu(t)$, then $m, \nu \models F$ iff $m, \nu \models F\{z \mapsto t\}$.

Composition

Definition 14.17

Let σ_1 and σ_2 be substitutions. The composition $\sigma_1\sigma_2$ of the substitutions is defined as follows.

$$\text{For each } x \in \text{Vars}, \; x(\sigma_1\sigma_2) \triangleq (x\sigma_1)\sigma_2.$$

Example 14.11

- $\sigma_1 = \{x \mapsto f(x,y)\}$ *and* $\sigma_2 = \{y \mapsto c\}$. $\sigma_1\sigma_2 = \; \{x \mapsto f(x,c), y \mapsto c\}$.
- $\sigma_1 = \{x \mapsto y\}$ *and* $\sigma_2 = \{y \mapsto x\}$. $\sigma_1\sigma_2 = \; \{x \mapsto x, y \mapsto x\} = \{y \mapsto x\}$.

Exercise 14.10

Show $\sigma_1(\sigma_2\sigma_3) = (\sigma_1\sigma_2)\sigma_3$, *i.e., substitution is associative.*

Commentary: Type check composition definition. Convince yourself that composition is well-defined.
Solution for exercise: Consider variable x. $(x\sigma_1)(\sigma_2\sigma_3) = ((x\sigma_1)\sigma_2)\sigma_3 = (x(\sigma_1\sigma_2))\sigma_3$

Composition works on terms and atoms

Commentary: Why do we need this theorem? In the definition, the composition is defined only for variables but not for arbitrary terms. We need to show that the definition extends for any term.

Theorem 14.2

For each $t \in T_{\mathbf{S}}$, $t(\sigma_1\sigma_2) = (t\sigma_1)\sigma_2$

Proof.

Proved by trivial structural induction. □

Theorem 14.3

For each $F \in A_{\mathbf{S}}$, $F(\sigma_1\sigma_2) = (F\sigma_1)\sigma_2$

Proof.

Proved by trivial structural induction. □

Substitution composition on formulas

Theorem 14.4

if $F\sigma_1$ and $(F\sigma_1)\sigma_2$ are defined then $(F\sigma_1)\sigma_2 = F(\sigma_1\sigma_2)$

Proof.

We prove it by induction. Non-quantifier cases are simple structural induction.

Assume $F = \forall x.G$

Since $F\sigma_1$ is defined, $G\sigma_{1x}$ is defined. Since $(F\sigma_1)\sigma_2$ is defined, $(G\sigma_{1x})\sigma_{2x}$ is defined (why?).

By induction hypothesis, $(G\sigma_{1x})\sigma_{2x} = G(\sigma_{1x}\sigma_{2x})$

claim: $G(\sigma_{1x}\sigma_{2x}) = G(\sigma_1\sigma_2)_x$

Choose $y \in FV(G)$ and $y \neq x$

$$y(\sigma_{1x}\sigma_{2x}) \underbrace{=((y\sigma_{1x})\sigma_{2x})}_{\text{Def. substitution}} \underbrace{=((y\sigma_1)\sigma_{2x})}_{y\neq x} \underbrace{=((y\sigma_1)\sigma_2)}_{x\notin FV(y\sigma_1)\text{(why?)}} = y(\sigma_1\sigma_2) = y(\sigma_1\sigma_2)_x$$

$$(\forall x.G\sigma_1)\sigma_2 = (\forall x.G(\sigma_{1x}\sigma_{2x})) = (\forall x.G(\sigma_1\sigma_2)_x) = F(\sigma_1\sigma_2)$$ □

Commentary: The substitution notation may be new to you. Please follow the argument for each step.

Topic 14.4

Problems

Properties of FOL

Exercise 14.11

If $x, y \notin Vars(F(z))$*, then* $\forall x.F(x) \Leftrightarrow \forall y.F(y)$

Exercise 14.12

Let us suppose x *does not occur in formula* G*. Show that the following formulas are valid.*

- $\exists x.G \Leftrightarrow G$
- $\forall x.G \Leftrightarrow G$
- $(\forall x.F(x) \lor G) \Leftrightarrow \forall x.(F(x) \lor G)$
- $(\forall x.F(x) \land G) \Leftrightarrow \forall x.(F(x) \land G)$
- $(\exists x.F(x) \lor G) \Leftrightarrow \exists x.(F(x) \lor G)$
- $(\exists x.F(x) \land G) \Leftrightarrow \exists x.(F(x) \land G)$

Encode mod k

Exercise 14.13

Give an FOL sentence that encodes that there are n elements in any satisfying model, such that $n \bmod k = 0$ for a given k.

Unique quantifier

Exercise 14.14
We could consider enriching the language by the addition of a new quantifier. The formula $\exists! x.F$ *(read "there exists a unique* x *such that* F*") is to be satisfied in model* m *and assignment* ν *iff there is one and only one* $d \in D_m$ *such that* $m, \nu[x \to d] \models F$*. Show that this apparent enrichment does not increase expressive power of FOL.*

Exercise: equality propagation

Exercise 14.15

Which of the following equivalences are correct?

- $\exists x, x'. (x' = x \wedge F(x, x')) \equiv \exists x. F(x, x)$
- $\exists x, x'. (x' = x \Rightarrow F(x, x')) \equiv \exists x. F(x, x)$
- $\forall x, x'. (x' = x \wedge F(x, x')) \equiv \forall x. F(x, x)$
- $\forall x, x'. (x' = x \Rightarrow F(x, x')) \equiv \forall x. F(x, x)$

Topic 14.5

Extra slides: not-so-useful definitions

Bounded variables

Definition 14.18

*A variable $x \in \text{Vars}$ is **bounded** in formula F if*

- $F = \neg G$: *x is bounded in G,*
- $F = G \circ H$: *x is bounded in G or H, for some binary operator $\circ$, and*
- $F = \exists y.G$ *or* $F = \forall y.G$: *x is bounded in G or x is equal to y.*

Let $bnd(F)$ denote the set of bounded variables in F.

Exercise 14.16

Is x bounded?

- $H(x)$
- $H(y)$
- $\forall x.H(x)$
- $x = y \Rightarrow \exists x.G(x)$

End of Lecture 14