CS 511, Fall 2020, Lecture Slides 12

Syntax of Predicate Logic (aka First-Order Logic)

Assaf Kfoury

Sept 24, 2020

from English reasoning to formal reasoning:

for all x, **if** x is a bird **then** x has wings

for all x, **if** x has wings **then** x can fly

Coco is a bird

Coco has wings

Coco's *mother* can fly

from English reasoning to formal reasoning:

for all x, **if** x is a bird **then** x has wings	$\forall x\ (B(x) \ \rightarrow\ W(x))$
for all x, **if** x has wings **then** x can fly	$\forall x\ (W(x) \ \rightarrow\ F(x))$
Coco is a bird	$B(\mathbf{C})$
Coco has wings	$W(\mathbf{C})$
Coco's *mother* can fly	$F(m(\mathbf{C}))$

from English reasoning to formal reasoning:

for all x, **if** x is a bird **then** x has wings	$\forall x\ (B(x) \ \rightarrow\ W(x))$
for all x, **if** x has wings **then** x can fly	$\forall x\ (W(x) \ \rightarrow\ F(x))$
Coco is a bird	$B(\mathbf{C})$
Coco has wings	$W(\mathbf{C})$
Coco's *mother* can fly	$F(m(\mathbf{C}))$
it is **not** the case that **for all** x ...	$\neg(\forall x\ (B(x) \ \rightarrow\ W(x)))$
there exists an x such that ...	$\exists x\ (B(x) \ \wedge\ \neg W(x))$

WFF's of predicate logic

- *vocabulary* (a.k.a. *similarity type*, a.k.a. *signature*):

- *terms*:

- *well-formed formulas*:

WFF's of predicate logic

- *vocabulary* (a.k.a. *similarity type*, a.k.a. *signature*):
  - set $\mathcal{P}$ of **predicate** symbols, each of arity $n \geqslant 0$
  - set $\mathcal{F}$ of **function** symbols, each of arity $n \geqslant 1$
  - set $\mathcal{C}$ of **constant** symbols, (a.k.a. functions of arity $= 0$)
- *terms*:
  - a variable x is a term
  - a constant $c \in \mathcal{C}$ is a term
  - if $t_1, \ldots, t_n$ are terms and $f \in \mathcal{F}$ is n-ary, $f(t_1, \ldots, t_n)$ is a term
  - as a BNF definition: $t \ ::= \ x \mid c \mid f(t, \ldots, t)$
- *well-formed formulas*:

$$\varphi ::= P(t_1, \ldots, t_n) \mid (t_1 \approx t_2) \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi) \mid (\forall x\, \varphi) \mid (\exists x\, \varphi)$$

WFF's of predicate logic

- *vocabulary* (a.k.a. *similarity type*, a.k.a. *signature*):
  - set $\mathcal{P}$ of **predicate** symbols, each of arity $n \geqslant 0$
  - set $\mathcal{F}$ of **function** symbols, each of arity $n \geqslant 1$
  - set $\mathcal{C}$ of **constant** symbols, (a.k.a. functions of arity $= 0$)
- *terms*:
  - a variable x is a term
  - a constant $c \in \mathcal{C}$ is a term
  - if $t_1, \ldots, t_n$ are terms and $f \in \mathcal{F}$ is n-ary, $f(t_1, \ldots, t_n)$ is a term
  - as a BNF definition: $t \ ::= \ x \mid c \mid f(t, \ldots, t)$
- *well-formed formulas*:

$$\varphi ::= P(t_1, \ldots, t_n) \mid (t_1 \approx t_2) \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \to \varphi) \mid (\forall x\, \varphi) \mid (\exists x\, \varphi)$$

- are all WFF's of propositional logic WFF's of predicate logic ?

free and **bound** variables

- a variable x may occur **free** or **bound** in a WFF φ
- if x is bound in φ, then there are $\geqslant 0$ **bound** occurrences of x and $\geqslant 1$ **binding** occurrences of x in φ
- a **binding** occurrence of x is of the form "$\forall x$" or "$\exists x$"
- if a binding occurrence of x occurs as $(\mathbf{Q}\, x\, \varphi)$ where $\mathbf{Q} \in \{\forall, \exists\}$, then φ is the **scope** of the binding occurrence
- scopes of two binding occurrences "$\mathbf{Q}\, x$" and "$\mathbf{Q}'\, x'$" may be

  disjoint: $\quad \cdots\ (\mathbf{Q}\, x \underbrace{\cdots\ \cdots}) \ \cdots\ (\mathbf{Q}'\, x' \underbrace{\cdots\ \cdots}) \ \cdots$

  or **nested**: $\quad \cdots\ (\mathbf{Q}\, x \underbrace{\cdots\ (\mathbf{Q}'\, x' \underbrace{\cdots\ \cdots})\ \cdots}) \ \cdots$

  but cannot **overlap**

free and **bound** variables (continued)

- the set of **free variables** in terms t and WFF's φ:

$$\mathsf{FV}(t) = \begin{cases} \varnothing & \text{if } t = c \\ \{x\} & \text{if } t = x \\ \mathsf{FV}(t_1) \cup \cdots \cup \mathsf{FV}(t_n) & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

$$\mathsf{FV}(\varphi) = \begin{cases} \mathsf{FV}(t_1) \cup \cdots \cup \mathsf{FV}(t_n) & \text{if } \varphi = P(t_1, \ldots, t_n) \\ \mathsf{FV}(t_1) \cup \mathsf{FV}(t_2) & \text{if } \varphi = (t_1 \approx t_2) \\ \mathsf{FV}(\varphi') & \text{if } \varphi = \neg\varphi' \\ \mathsf{FV}(\varphi_1) \cup \mathsf{FV}(\varphi_2) & \text{if } \varphi = (\varphi_1 \star \varphi_2),\ \star \in \{\wedge, \vee, \to\} \\ \mathsf{FV}(\varphi') - \{x\} & \text{if } \varphi = (\mathbf{Q}x\ \varphi') \text{ and } \mathbf{Q} \in \{\forall, \exists\} \end{cases}$$

free and **bound** variables (continued)

- the set of **free variables** in terms t and WFF's φ:

$$\mathsf{FV}(t) = \begin{cases} \varnothing & \text{if } t = c \\ \{x\} & \text{if } t = x \\ \mathsf{FV}(t_1) \cup \cdots \cup \mathsf{FV}(t_n) & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

$$\mathsf{FV}(\varphi) = \begin{cases} \mathsf{FV}(t_1) \cup \cdots \cup \mathsf{FV}(t_n) & \text{if } \varphi = P(t_1, \ldots, t_n) \\ \mathsf{FV}(t_1) \cup \mathsf{FV}(t_2) & \text{if } \varphi = (t_1 \approx t_2) \\ \mathsf{FV}(\varphi') & \text{if } \varphi = \neg\varphi' \\ \mathsf{FV}(\varphi_1) \cup \mathsf{FV}(\varphi_2) & \text{if } \varphi = (\varphi_1 \star \varphi_2), \star \in \{\wedge, \vee, \to\} \\ \mathsf{FV}(\varphi') - \{x\} & \text{if } \varphi = (\mathbf{Q}x\ \varphi') \text{ and } \mathbf{Q} \in \{\forall, \exists\} \end{cases}$$

- **assumption**: every variable x has $\leqslant 1$ binding occurrence in any WFF (is this realistic?)

  this assumption is not essential, but without it, a variable may occur both free and bound in the same WFF.

free and **bound** variables (continued)

- φ is **closed** iff $\mathrm{FV}(\varphi) = \varnothing$
- how to satisfy the following **assumption**:
  every variable x has $\leqslant 1$ binding occurrence in any WFF?
- consider a WFF φ (**not** satisfying the **assumption**), say:

$$\varphi = \quad \cdots \Big(\mathbf{Q}_1\, x\, (\cdots\ x\ \cdots)\Big) \cdots \Big(\mathbf{Q}_2\, x\, (\cdots\ x\ \cdots)\Big) \cdots$$

  where $\mathbf{Q}_1, \mathbf{Q}_2 \in \{\forall, \exists\}$

- is φ equivalent to:

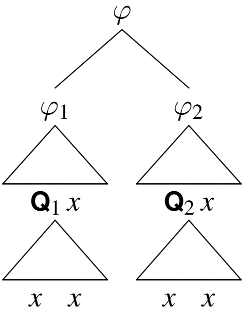
$$\varphi' = \quad \cdots \Big(\mathbf{Q}_1\, x\, (\cdots\ x\ \cdots)\Big) \cdots \Big(\mathbf{Q}_2\ x'\ (\cdots\ x'\ \cdots)\ \cdots\Big)\ \textbf{??}$$

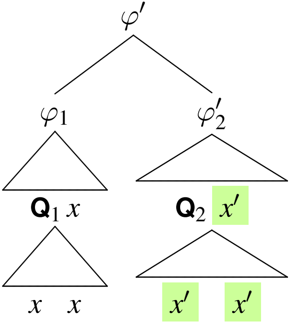
- yes, φ and φ' are equivalent

  Exercise: define the algorithm to transform φ into φ'

free and **bound** variables (continued)

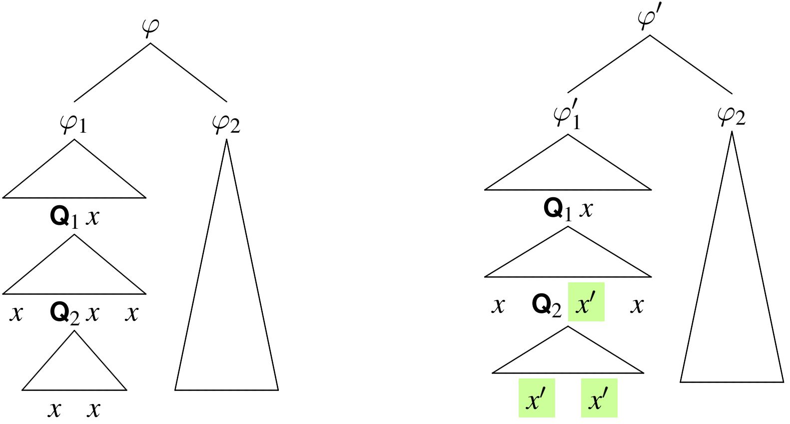
renaming binding occurrences "$\mathbf{Q}_1\, x$" and "$\mathbf{Q}_2\, x$" in **disjoint** scopes





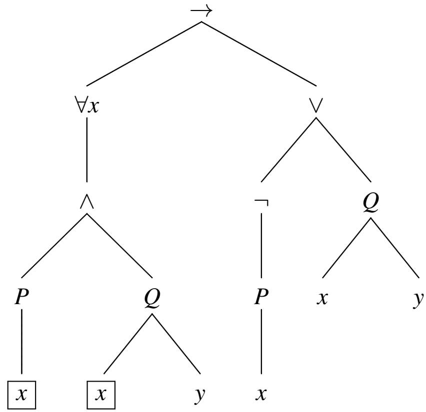
free and **bound** variables (continued)

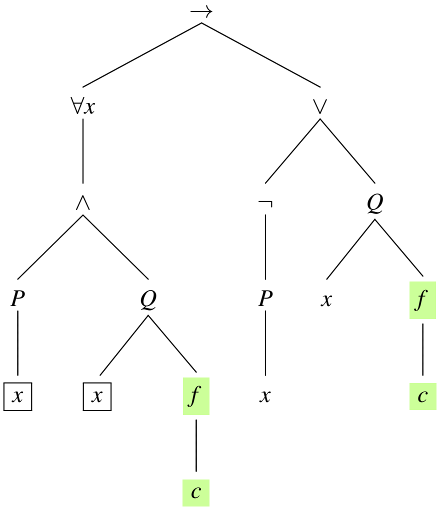
renaming binding occurrences "$\mathbf{Q}_1\, x$" and "$\mathbf{Q}_2\, x$" in **nested** scopes



substitution examples in $\varphi = (\forall x\ (P(x) \wedge Q(x,y))) \to (\neg P(x) \vee Q(x,y))$

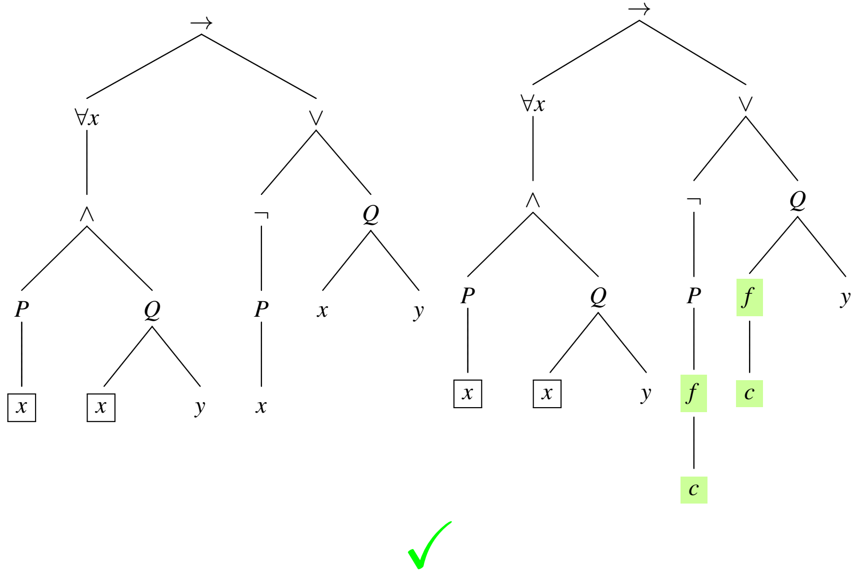
substitute $f(c)$ for y in φ: $\varphi[f(c)/y]$ (also written $\varphi[y := f(c)]$)





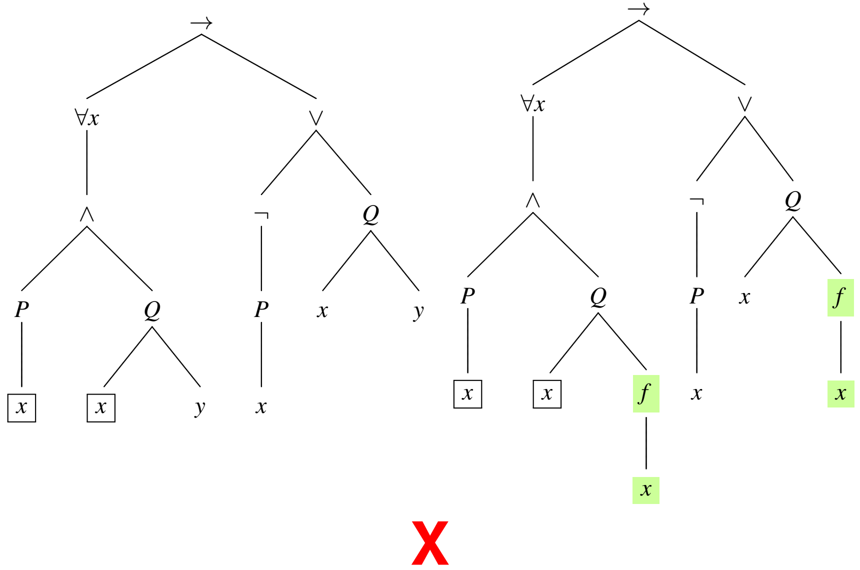
substitution examples in $\varphi = (\forall x\ (P(x) \wedge Q(x,y))) \to (\neg P(x) \vee Q(x,y))$

substitute $f(c)$ for x in φ: $\varphi[f(c)/x]$



substitution examples in $\varphi = (\forall x\ (P(x) \wedge Q(x,y))) \to (\neg P(x) \vee Q(x,y))$

substitute $f(x)$ for y in φ: $\varphi[f(x)/y]$



formal definition of substitution

- given: term t, WFF φ, variable x, term u

$$t[u/x] = \begin{cases} c & \text{if } t = c \\ u & \text{if } t = x \\ y & \text{if } t = y \text{ and } y \neq x \\ f(t_1[u/x], \ldots, t_n[u/x]) & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

$$\varphi[u/x] = \begin{cases} P(t_1[u/x], \ldots, t_n[u/x]) & \text{if } \varphi = P(t_1, \ldots, t_n) \\ (t_1[u/x] \approx t_2[u/x]) & \text{if } \varphi = (t_2 \approx t_2) \\ \neg(\varphi'[u/x]) & \text{if } \varphi = \neg\varphi' \\ \varphi_1[u/x] \star \varphi_2[u/x] & \text{if } \varphi = \varphi_1 \star \varphi_2 \text{ and} \\ & \star \in \{\wedge, \vee, \to\} \\ \mathbf{Q}y\,(\varphi'[u/x]) & \text{if } \varphi = \mathbf{Q}y\,\varphi', \\ & \mathbf{Q} \in \{\forall, \exists\},\ x \neq y, \text{ and} \\ & u \text{ is } \textbf{substitutable} \text{ for } x \text{ in } \varphi \\ \mathbf{Q}y\,\varphi' & \text{if } \varphi = \mathbf{Q}y\,\varphi', \\ & \mathbf{Q} \in \{\forall, \exists\},\ x = y \end{cases}$$

(THIS PAGE INTENTIONALLY LEFT BLANK)