Predicate Logic (Natural Deduction Proofs)
20 November 2024 21:08

Substitution is defined recursively on terms and formulas:
$P[x\backslash t]$ substitute all the free occurrences of x in P with t.

Substitution

Substitution is defined recursively on terms and formulas:
$P[x\backslash t]$ substitute all the free occurrences of x in P with t.

1st attempt (WRONG)

$$
\begin{array}{lcl}
x[x\backslash t] & = & t \\
x[y\backslash t] & = & x \\
(f(t_1,\dots,t_n))[x\backslash t] & = & f(t_1[x\backslash t],\dots,t_n[x\backslash t]) \\
(p(t_1,\dots,t_n))[x\backslash t] & = & p(t_1[x\backslash t],\dots,t_n[x\backslash t]) \\
\hline
(\neg P)[x\backslash t] & = & \neg P[x\backslash t] \\
(P_1 \land P_2)[x\backslash t] & = & P_1[x\backslash t] \land P_2[x\backslash t] \\
(P_1 \lor P_2)[x\backslash t] & = & P_1[x\backslash t] \lor P_2[x\backslash t] \\
(P_1 \to P_2)[x\backslash t] & = & P_1[x\backslash t] \to P_2[x\backslash t] \\
\hline
(\forall x.P)[x\backslash t] & = & \forall x.P \\
(\exists x.P)[x\backslash t] & = & \exists x.P \\
(\forall y.P)[x\backslash t] & = & \forall y.P[x\backslash t] \\
(\exists y.P)[x\backslash t] & = & \exists y.P[x\backslash t]
\end{array}
$$

Why is this wrong? $(\forall y.y > x)[x\backslash y]$ would return $\forall y.y > y$, where the free y is now bound! The free y got **captured**! The red occurrences of y stand for different variables than the green ones.

Substitution is defined recursively on terms and formulas:
$P[x\backslash t]$ substitute all the free occurrences of x in P with t.

2nd attempt (CORRECT)

$$
\begin{array}{lcl}
x[x\backslash t] & = & t \\
x[y\backslash t] & = & x \\
(f(t_1,\dots,t_n))[x\backslash t] & = & f(t_1[x\backslash t],\dots,t_n[x\backslash t]) \\
(p(t_1,\dots,t_n))[x\backslash t] & = & p(t_1[x\backslash t],\dots,t_n[x\backslash t]) \\
\hline
(\neg P)[x\backslash t] & = & \neg P[x\backslash t] \\
(P_1 \land P_2)[x\backslash t] & = & P_1[x\backslash t] \land P_2[x\backslash t] \\
(P_1 \lor P_2)[x\backslash t] & = & P_1[x\backslash t] \lor P_2[x\backslash t] \\
(P_1 \to P_2)[x\backslash t] & = & P_1[x\backslash t] \to P_2[x\backslash t] \\
\hline
(\forall x.P)[x\backslash t] & = & \forall x.P \\
(\exists x.P)[x\backslash t] & = & \exists x.P \\
(\forall y.P)[x\backslash t] & = & \forall y.P[x\backslash t],\ \text{if } y \notin \mathtt{fv}(t) \\
(\exists y.P)[x\backslash t] & = & \exists y.P[x\backslash t],\ \text{if } y \notin \mathtt{fv}(t)
\end{array}
$$

The additional **conditions** ensure that **free variables do not get captured**.

These conditions can always be met by silently renaming bound variables before substituting.

Rules for Substitution:

1. **Free Variable Substitution:**
   - When substituting a free variable, make sure it doesn't conflict with any bound variable in the expression.
   - Example: If $P = \forall y.(y < x \land odd(y))$ and you want to substitute x with 2, it becomes $\forall y.(y < 2 \land odd(y))$.
2. **Bound Variable Substitution:**
   - When substituting a bound variable, ensure the substitution doesn't change the meaning of the expression.
   - Example: If $P = \forall x.(x < y \land odd(x))$ and you want to substitute x with z, it becomes $\forall z.(z < y \land odd(z))$.

Suppose:

- $P = x > y$ (a predicate where x and y are free variables).
- Substitute $t = z + 1$ for x in $\forall y.P$.

Correct Substitution:

Here, $\forall y.(x > y)[x \setminus (z+1)]$ becomes: $\forall y.(z + 1 > y)$.

This is valid as long as y (the quantified variable) is not free in $t = z + 1$. No conflict or variable capture happens because $y \notin \mathrm{fv}(z+1)$.

Similarly for $(\exists y.P)[x \setminus t]$:

If $P = x > y$ and we substitute $t = z + 1$ for x, the result is: $\exists y.(z + 1 > y)$, as long as $y \notin \mathrm{fv}(t)$. Again, this avoids any accidental binding or capture.

By following the rule $y \notin \mathrm{fv}(t)$, we ensure no unintended interactions between the quantifier and the free variables in t.

What if $y \in \mathrm{fv}(t)$?

If $t = y + 1$ (where y is already free in t), the substitution could lead to incorrect binding: $\forall y.(y + 1 > y)$, where the free y from t gets unintentionally bound by the quantifier $\forall y$. This violates the rule $y \notin \mathrm{fv}(t)$ to prevent such variable capture.

Free & Bound Variables

Free variables and **Bound** variables:

Bound variables:

- Consider the formula $\forall x.\mathrm{even}(x) \lor \mathrm{odd}(x)$
  Here the variable x is **bound** by the quantifier $\forall$
- $\forall x.\mathrm{even}(x) \lor \mathrm{odd}(x)$ is considered the same as $\forall y.\mathrm{even}(y) \lor \mathrm{odd}(y)$
  Renaming a **bound** variable **doesn't** change the meaning!

Free variables:

- Consider the formula $\forall y.x \leqslant y$
- y is a **bound** variable and x is a **free** variable
- variables are **free** if they are not bound
- $\forall y.x \leqslant y$ is the **same** as $\forall z.x \leqslant z$
- $\forall y.x \leqslant y$ is **not the same** as $\forall y.w \leqslant y$
- Renaming a free variable changes the meaning!

Free & Bound Variables

The **scope** of a quantified formula of the form $\forall x.P$ or $\exists x.P$ is P. The quantifier are said to **bind** x.

Bound variables: a variable x occurs bound in a formula, if it occurs in the scope of a quantifier quantifying x

Free variables: a variable x occurs free in a formula, if it does not occur in the scope of a quantifier quantifying x

The set of variables occurring free/bound in a terms and formulas is recursively computed as follows:

$$
\begin{array}{lcl}
\mathtt{fv}(x) & = & \{x\} \\
\mathtt{fv}(f(t_1,\dots,t_n)) & = & \mathtt{fv}(t_1) \cup \dots \cup \mathtt{fv}(t_n) \\
\mathtt{fv}(p(t_1,\dots,t_n)) & = & \mathtt{fv}(t_1) \cup \dots \cup \mathtt{fv}(t_n) \\
\hline
\mathtt{fv}(\neg P) & = & \mathtt{fv}(P) \\
\mathtt{fv}(P_1 \land P_2) & = & \mathtt{fv}(P_1) \cup \mathtt{fv}(P_2) \\
\mathtt{fv}(P_1 \lor P_2) & = & \mathtt{fv}(P_1) \cup \mathtt{fv}(P_2) \\
\mathtt{fv}(P_1 \to P_2) & = & \mathtt{fv}(P_1) \cup \mathtt{fv}(P_2) \\
\hline
\mathtt{fv}(\forall x.P) & = & \mathtt{fv}(P) \setminus \{x\} \\
\mathtt{fv}(\exists x.P) & = & \mathtt{fv}(P) \setminus \{x\}
\end{array}
\qquad
\begin{array}{lcl}
\mathtt{bv}(p(t_1,\dots,t_n)) & = & \emptyset \\
\hline
\mathtt{bv}(\neg P) & = & \mathtt{bv}(P) \\
\mathtt{bv}(P_1 \land P_2) & = & \mathtt{bv}(P_1) \cup \mathtt{bv}(P_2) \\
\mathtt{bv}(P_1 \lor P_2) & = & \mathtt{bv}(P_1) \cup \mathtt{bv}(P_2) \\
\mathtt{bv}(P_1 \to P_2) & = & \mathtt{bv}(P_1) \cup \mathtt{bv}(P_2) \\
\hline
\mathtt{bv}(\forall x.P) & = & \mathtt{bv}(P) \cup \{x\} \\
\mathtt{bv}(\exists x.P) & = & \mathtt{bv}(P) \cup \{x\}
\end{array}
$$

What are the free variables of the following formulas

- $P_1 = (\mathrm{odd}(x) \land \exists y.y < x \land \mathrm{odd}(y))$
  $\mathtt{fv}(P_1) = \{x\}$
- $P_2 = (\mathrm{odd}(x) \land x > y \land \exists y.y < x \land \mathrm{odd}(y))$
  $\mathtt{fv}(P_2) = \{x, y\}$
- $P_3 = (\forall x.\mathrm{odd}(x) \land x > y \land \exists y.y < x \land \mathrm{odd}(y))$
  $\mathtt{fv}(P_3) = \{y\}$

Note: In $(\mathrm{odd}(x) \land x > y \land \exists y.y < x \land \mathrm{odd}(y))$ the green occurrence of y is **not** the same variable as the red occurrence of y.

The formula $(\mathrm{odd}(x) \land x > y \land \exists y.y < x \land \mathrm{odd}(y))$ is considered the same as $(\mathrm{odd}(x) \land x > y \land \exists z.z < x \land \mathrm{odd}(z))$

Here odd(y) is bound by ∃, which is why changing the variable name doesn't affect it

If you don't get this (^) you're a fucking idiot

Inference rules for ∀ and ∃?

Propositional logic: Each connective has at least 2 inference rules

- At least 1 for introduction
- At least 1 for elimination

Introduction and elimination rules for ∀ and ∃?

If you don't get this ( ) you're a fucking idiot

Inference rules for ∀ and ∃?

Propositional logic: Each connective has at least 2 inference rules

- At least 1 for introduction
- At least 1 for elimination

Introduction and elimination rules for ∀ and ∃?

$$\frac{?}{\forall y.P}\ [\forall I] \qquad \frac{\forall x.P}{?}\ [\forall E]$$

$$\frac{?}{\exists y.P}\ [\exists I] \qquad \frac{\exists x.P}{?}\ [\exists E]$$

WARNING ⚠

Trickier than inference rules from propositional logic!

We need to be careful with free and bound variables!

$$\frac{?}{\exists y.P}\ [\exists I]$$

Inference Rule for "exists introduction"

$$\frac{?}{\exists x.P}\ [\exists I]$$

When can we conclude P is true for some x?

If we have proved predicate P for an element of the domain

$$\frac{P[x\backslash t]}{\exists x.P}\ [\exists I]$$

Condition: $\mathrm{fv}(t)$ must not clash with $\mathrm{bv}(P)$

Example: Consider the predicate $P = (\forall y.y = x)$

- Without the substitution conditions $P[x\backslash y]$ would be true
- We could then deduce $\exists x.\forall y.y = x$, i.e., numbers are all equal to each other — obviously incorrect!
- The substitution conditions prevents such captures
- $[\exists I]$'s condition guarantees that the substitution conditions hold

$$\frac{\forall x.P}{?}\ [\forall E]$$

Inference Rule for "for all elimination" – 1st attempt

$$\frac{\forall x.P}{?}\ [\forall E]$$

What can we conclude from the fact that P is true for all x?

Predicate P is true for all elements x of the domain

- For any element of the domain t, we can deduce that P is true where x is replaced by t is true
- This "replacing" operation is a **substitution** operation as seen in lecture 2.
- However, we now have to be careful with free/bound variables.

Inference Rule for "for all elimination" – 2nd attempt

The correct rule is:

$$\frac{\forall x.P}{P[x\backslash t]}\ [\forall E]$$

Condition: $\mathrm{fv}(t)$ must not clash with any bound variables of P

Example: consider the formula $\forall x.\exists y.y > x$

- True over domain of natural numbers
- P is $\exists y.y > x$
- Let t be y
- This condition guarantees that we can do the substitution

1st method

- Substituting x with y without renaming bound variables would give the wrong answer (see previous slide)
- Therefore, we first rename bound variables that clash with $\mathrm{fv}(t)$, i.e., with y: $\exists z.z > x$
- Then, we substitute: $\exists z.z > y$

2nd method

$$\frac{?}{\forall y.P}\ [\forall I]$$

Inference Rule for "for all introduction"

$$\frac{?}{\forall x.P}\ [\forall I]$$

When can we conclude P is true for all x?

If we have proved P for a "**general/representative/typical**" variable

$$\frac{P[x\backslash y]}{\forall x.P}\ [\forall I]$$

Condition: y must not be free in any not-yet-discharged hypothesis or in $\forall x.P$

What could go wrong without this condition?

Otherwise, given the assumption $y > 2$, we could derive $\forall x.x > 2$, which is clearly wrong.

- Otherwise, given the assumption $x > 2$, we could derive $\forall x.x > 2$, which is clearly wrong.
- We could also derive $\forall x.\forall y.x > 0 \rightarrow y > 0$, which is also clearly wrong.

However, it is okay for the variable to appear in an assumption that is discharged above the $[\forall I]$ rule:

$$\cfrac{\cfrac{\overline{x > 2}^{\,1}}{x > 2 \rightarrow x > 2}\ 1\ [\rightarrow I]}{\forall x.x > 2 \rightarrow x > 2}\ [\forall I]$$

$$\frac{\exists x.P}{?}\ [\exists E]$$

Inference Rule for "exists

$$\frac{\exists x.P}{?}\ [\exists E]$$

What can we conclude from the fact that

We know that it holds about some element

but we do not know which

$$\frac{\exists x.P \qquad \begin{array}{c}\overline{P[x\backslash y]}^{\,1}\\ \vdots \\ Q\end{array}}{Q}\ 1\ [\exists E]$$

Condition: y must not be free in Q or in

hypotheses or in $\exists x.P$

This rule is similar to OR-elimination!

Examples:

A simple proof

Prove that $(\forall z.p(z)) \rightarrow \forall x.p(x) \vee q(x)$

We use backward reasoning

$$\cfrac{\cfrac{\cfrac{\cfrac{\overline{\forall z.p(z)}^{\,1}}{p(y)}\ [\forall E]}{p(y) \vee q(y)}\ [\vee I_L]}{\forall x.p(x) \vee q(x)}\ [\forall I]}{(\forall z.p(z)) \rightarrow \forall x.p(x) \vee q(x)}\ 1\ [\rightarrow I]$$

Conditions:

- y does not occur free in not-yet-discharged hypotheses or in $\forall x.p(x) \vee q(x)$
- y does not clash with bound variables in $p(z)$

A simple proof

More generally, we can prove:

$$\cfrac{\cfrac{\cfrac{\cfrac{\overline{\forall z.P}^{\,1}}{P[x\backslash y]}\ [\forall E]}{P[x\backslash y] \vee Q[x\backslash y]}\ [\vee I_L]}{\forall x.P \vee Q}\ [\forall I]}{(\forall z.P) \rightarrow \forall x.P \vee Q}\ 1\ [\rightarrow I]$$

We assume that y does not occur in P or Q

Summary of all 4 inference rules:

elimination"

P is true for some x?
t of the domain,

not-yet-discharged

All four inference rules in one slide

$$\frac{P[x\backslash y]}{\forall x.P}\ [\forall I]$$

Condition: y must not be free in any not-yet-discharged hypothesis or in $\forall x.P$

$$\frac{\forall x.P}{P[x\backslash t]}\ [\forall E]$$

Condition: $\texttt{fv}(t)$ must not clash with $\texttt{bv}(P)$

$$\frac{P[x\backslash t]}{\exists x.P}\ [\exists I]$$

Condition: $\texttt{fv}(t)$ must not clash with $\texttt{bv}(P)$

$$\cfrac{\exists x.P \qquad \begin{array}{c}\overline{P[x\backslash y]}^{\,1} \\ \vdots \\ Q\end{array}}{Q}\ 1\ [\exists E]$$

Condition: y must not be free in Q or in not-yet-discharged hypotheses or in $\exists x.P$