# Mathematical and Logical Foundations of Computer Science

# Lecture 13 - Predicate Logic (Natural Deduction Proofs – Continued)

Vincent Rahli

(some slides were adapted from Rajesh Chitnis' slides)

University of Birmingham

# Where are we?

- Symbolic logic
- Propositional logic
- **Predicate logic**

# Today

- Natural Deduction proofs for Predicate Logic
- side conditions

**Further reading**:

- Chapter 8 of
  http://leanprover.github.io/logic_and_proof/

# Recap: Syntax

The syntax of predicate logic is defined by the following grammar:

$$
\begin{array}{lcl}
t & ::= & x \mid f(t, \ldots, t) \\
P & ::= & p(t, \ldots, t) \mid \neg P \mid P \wedge P \mid P \vee P \mid P \rightarrow P \mid \forall x.P \mid \exists x.P
\end{array}
$$

where:

- $x$ ranges over variables
- $f$ ranges over function symbols
- $f(t_1, \ldots, t_n)$ is a well-formed term only if $f$ has arity $n$
- $p$ ranges over predicate symbols
- $p(t_1, \ldots, t_n)$ is a well-formed formula only if $p$ has arity $n$

The pair of a collection of function symbols, and a collection of predicate symbols, along with their arities, is called a **signature**.

The scope of a quantifier extends as far right as possible. E.g., $P \wedge \forall x.p(x) \vee q(x)$ is read as $P \wedge \forall x.(p(x) \vee q(x))$

# Recap: Substitution

Substitution is defined recursively on terms and formulas:

$P[x\backslash t]$ substitute all the free occurrences of $x$ in $P$ with $t$.

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

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

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

# Recap: $\forall$ & $\exists$ elimination and introduction rules

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

**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**: $\mathtt{fv}(t)$ must not clash with $\mathtt{bv}(P)$

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

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

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

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

# Inference Rule for "for all elimination"

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

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

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

# Inference Rule for "for all elimination"

**More precisely**: Assume that from $\forall x.\exists y.y > x$, we want to derive a number greater than $y$.

We would use the following rule:

$$\frac{\forall x.\exists y.y > x}{(\exists y.y > x)[x\backslash y]}\ [\forall E]$$

However, without renaming the bound $y$, $P[x\backslash y]$ is undefined

Therefore, we rename the bound variable just before performing the substitution:

$$\frac{\forall x.\exists y.y > x}{\exists z.z > y}\ [\forall E]$$

# Inference Rule for "for all introduction"

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

We conclude $P$ is true for all $x$ if we have proved $P$ for a "**general/representative/typical**" variable

**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 $x > 2$, 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.

# Inference Rule for “for all introduction”

**More precisely**: without this condition we would be able to derive

$$\dfrac{\dfrac{\overline{x > 2}^{\;1}}{\forall x. x > 2}\ \cancel{[\forall I]}}{x > 2 \rightarrow \forall x. x > 2}\ 1\ [\rightarrow I]$$

WARNING ⚠ Note that this is **not a correct use** of the $[\forall I]$ rule because $x$ is free in $x > 2$, which is not-yet-discharged when the $[\forall I]$ rule is applied

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

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

# Inference Rule for "for all introduction"

How can we make checking this condition more tractable?

Going backward, we must ensure such variables

- are not free in the hypotheses we have introduced and discharged at the time $[\forall I]$ is used,
- are not free in the universally quantified formula.

We record those hypotheses in a **context** as follows:

$$\dfrac{\dfrac{y > 2}{\forall x. x > 2}\ [\forall I]}{x > 2 \rightarrow \forall x. x > 2}\ 1\ [\rightarrow I]$$

**Context**:

- 1: $x > 2$

We cannot pick $x$ as it occurs in our **context**
We must pick a "fresh" variable not free in the **context** or in $\forall x. x > 2$
We cannot finish this proof now

# Inference Rule for "for all introduction"

Prove $\forall x. x > 2 \rightarrow x > 2$ backward using contexts

Here is a proof:

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

Context:

- 1: $x > 2$

We can pick any variable we want as the context is empty and our conclusion does not have any free variables

# Inference Rule for “for all introduction”

What could happen if we could pick a variable free in the conclusion?

If we could pick a variable free in the conclusion, we could derive:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{x>0}\;1}{x>0\to x>0}\;1\;[\to I]}{\forall y.x>0\to y>0}\;\cancel{[\forall I]}}{\forall x.\forall y.x>0\to y>0}\;[\forall I]$$

WARNING Note that this is **not a correct use** of the $[\forall I]$ rule because $x$ is free the conclusion $\forall y.x>0\to y>0$

# Inference Rule for “for all introduction”

The rule’s condition forces us to pick a **different** variable:

$$\dfrac{\dfrac{\dfrac{\overline{y > 0}}{x > 0 \to y > 0}\ {}^{1}\ [\to I]}{\forall y.x > 0 \to y > 0}\ [\forall I]}{\forall x.\forall y.x > 0 \to y > 0}\ [\forall I]$$

We cannot finish this proof now

# Inference Rule for "exists introduction"

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

We conclude $P$ is true for some $x$ if we have proved predicate $P$ for an element of the domain

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

**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

# Inference Rule for "exists introduction"

As for "for all elimination", we rename the bound variable just before performing the substitution.

For example if we know that $y$ is the smallest number:

$$\frac{\forall z. y \leqslant z}{\exists x. \forall y. x \leqslant y}\ [\exists I]$$

# Inference Rule for “exists elimination”

$$\dfrac{\exists x.P \qquad \begin{matrix}\overline{P[x\backslash y]}^{\,1} \\ \vdots \\ Q\end{matrix}}{Q}\;1\;[\exists E]$$

From the fact that $P$ is true for some $x$ we know that it holds about some element of the domain, but we do not know which

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

This rule is similar to OR-elimination!

# Inference Rule for “exists elimination”

## What could go wrong without this condition?

Assume for the sake of this example that $x \leqslant y$ is defined as $\neg y < x$

Without the condition we could prove:

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{\exists x.\forall y.x \leqslant y}\;{}^{2} \qquad \dfrac{\dfrac{}{0 < z}\;{}^{1} \qquad \dfrac{\dfrac{}{\forall y.z \leqslant y}\;{}^{3}}{z \leqslant 0}\;[\forall E]}{\bot}\;[\neg E]}{\bot}\;\cancel{3\ [\exists E]}}{\neg\exists x.\forall y.x \leqslant y}\;2\ [\neg I]}{0 < z \rightarrow \neg\exists x.\forall y.x \leqslant y}\;1\ [\rightarrow I]
$$

WARNING ⚠ Note that this is **not a correct use** of the $[\exists E]$ rule because $z$ is free in $0 < z$, which is not-yet-discharged when the $[\exists E]$ rule is applied

# Inference Rule for “exists elimination”

Similarly, without the condition we could prove:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{0 < z}\,1 \qquad \dfrac{\dfrac{}{\exists x.\forall y.x \leqslant y}\,2 \qquad \dfrac{\dfrac{}{\forall y.z \leqslant y}\,3}{z \leqslant 0}\,[\forall E]}{z \leqslant 0}\,\cancel{3\ [\exists E]}}{\bot}\,[\neg E]}{\neg\exists x.\forall y.x \leqslant y}\,2\ [\neg I]}{0 < z \rightarrow \neg\exists x.\forall y.x \leqslant y}\,1\ [\rightarrow I]$$

WARNING ⚠ Note that this is **not a correct use** of the $[\exists E]$ rule because $z$ is free in $z \leqslant 0$, the conclusion of the instance of the $[\exists E]$ rule

# Inference Rule for “exists elimination”

We use contexts to make checking this condition more tractable

For example:

$$
\dfrac{\dfrac{}{\exists x.\forall y.x \leqslant y}\,2 \qquad \dfrac{\dfrac{}{0 < z}\,1 \qquad z \leqslant 0}{\bot}\,[\neg E]}{\dfrac{\bot}{\dfrac{\neg\exists x.\forall y.x \leqslant y}{0 < z \rightarrow \neg\exists x.\forall y.x \leqslant y}\,1\ [\rightarrow I]}\,2\ [\neg I]}\,3\ [\exists E]
$$

**Context**:

- 1: $0 < z$
- 2: $\exists x.\forall y.x \leqslant y$
- 3: $\forall y.w \leqslant y$

We cannot pick $z$ anymore as it occurs free in the context

We must pick a fresh variable $w$ not free in the context ($1$ and $2$), the conclusion $\bot$, or $\exists x.\forall y.x \leqslant y$

We cannot conclude our proof anymore

# Inference Rule for "exists elimination"

What could happen if we could pick a variable free in the $\exists$ formula?

Let us assume for the sake of this example that we can use the following rule

$$\dfrac{t < t}{\bot}\ [IRREFL]$$

If we could pick a variable free in the $\exists$ formula, we could derive:

$$\dfrac{\dfrac{\dfrac{}{\exists x.\exists y.x < y}\ 1 \qquad \dfrac{\dfrac{}{\exists y.x < y}\ 2 \qquad \dfrac{\dfrac{}{x < x}\ 3}{\bot}\ [IRREFL]}{\bot}\ \cancel{3\ [\exists E]}}{\bot}\ 2\ [\exists E]}{\neg\exists x.\exists y.x < y}\ 1\ [\neg I]$$

WARNING ⚠ Note that this is **not a correct use** of the $[\exists E]$ rule because $x$ is free in $\exists y.x < y$

# Another Natural Deduction proof with contexts

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

Here is a proof:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\forall x.p(x)}\,1}{p(z)}\,[\forall E] \quad \dfrac{\dfrac{}{\forall y.q(y)}\,2}{q(z)}\,[\forall E]}{p(z) \land q(z)}\,[\land I]}{\dfrac{\forall z.p(z) \land q(z)}{(\forall y.q(y)) \rightarrow \forall z.p(z) \land q(z)}\,2\ [\rightarrow I]}\,[\forall I]}{(\forall x.p(x)) \rightarrow (\forall y.q(y)) \rightarrow \forall z.p(z) \land q(z)}\,1\ [\rightarrow I]
$$

Context:

- 1: $\forall x.p(x)$
- 2: $\forall y.q(y)$

$z$ does not occur free in the context or in the conclusion

# Formal verification

Predicate Logic is more expressive and more convenient than Propositional Logic

- to do **Mathematics**
- to do **program verification**, i.e., to formally/mathematically verify that a program satisfies some formal/mathematical specification

**Simple example**: let the domain be $\mathbb{N}$ and the signature be:

- predicates: $\geqslant$ of arity $2$
- functions: $\mathtt{max}$ of arity $2$; and $0, 1, 2, \ldots$ of arity $0$

Let us define the following function:

$$\mathtt{max3}(t_1, t_2, t_3) \text{ stands for } \mathtt{max}(t_1, \mathtt{max}(t_2, t_3))$$

A specification for $\mathtt{max}$ might be:

$$\forall x.\forall y.\mathtt{max}(x, y) \geqslant x \wedge \mathtt{max}(x, y) \geqslant y$$

# Formal verification

While a specification for `max3` might be:

$$\forall x.\forall y.\forall z.\texttt{max3}(x,y,z) \geqslant x \wedge \texttt{max3}(x,y,z) \geqslant y \wedge \texttt{max3}(x,y,z) \geqslant z$$

Prove that `max3` satisfies this specification using Natural Deduction

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\forall x.\forall y.\texttt{max}(x,y) \geqslant x}{\forall y.\texttt{max}(u,y) \geqslant u}[\forall E]}{\texttt{max3}(u,v,w) \geqslant u}[\forall E] \qquad \ldots}{\texttt{max3}(u,v,w) \geqslant u \wedge \texttt{max3}(u,v,w) \geqslant v \wedge \texttt{max3}(u,v,w) \geqslant w}[\wedge I]}{\forall z.\texttt{max3}(u,v,z) \geqslant u \wedge \texttt{max3}(u,v,z) \geqslant v \wedge \texttt{max3}(u,v,z) \geqslant z}[\forall I]}{\forall y.\forall z.\texttt{max3}(u,y,z) \geqslant u \wedge \texttt{max3}(u,y,z) \geqslant y \wedge \texttt{max3}(u,y,z) \geqslant z}[\forall I]}{\forall x.\forall y.\forall z.\texttt{max3}(x,y,z) \geqslant x \wedge \texttt{max3}(x,y,z) \geqslant y \wedge \texttt{max3}(x,y,z) \geqslant z}[\forall I]$$

We skipped some parts of the proof. For the missing part, we also need to assume that $\geqslant$ is transitive.

# Conclusion

**What did we cover today?**

- Natural Deduction proofs for Predicate Logic
- side conditions

**Further reading**:

- Chapter 8 of
  `http://leanprover.github.io/logic_and_proof/`