NICTA Advanced Course

Theorem Proving

Principles, Techniques, Applications

HOL

Quasi Orders

$$\lesssim :: \alpha \Rightarrow \alpha \Rightarrow bool$$

is a quasi order iff it satisfies

$x \lesssim x$ (reflexivity) and

$x \lesssim y \wedge y \lesssim z \Longrightarrow x \lesssim z$ (transitivity)

QUASI ORDERS

$$\lesssim :: \alpha \Rightarrow \alpha \Rightarrow bool$$

is a quasi order iff it satisfies

$x \lesssim x$ (reflexivity) and

$x \lesssim y \wedge y \lesssim z \Longrightarrow x \lesssim z$ (transitivity)

(a partial order is also antisymmetric: $x \leq y \wedge y \leq x \Longrightarrow x = y$)

CONTENT

- ➜ Intro & motivation, getting started with Isabelle
- ➜ **Foundations & Principles**
  - Lambda Calculus
  - **Higher Order Logic, natural deduction**
  - Term rewriting
- ➜ Proof & Specification Techniques
  - Datatypes, recursion, induction
  - Inductively defined sets, rule induction
  - Calculational reasoning, mathematics style proofs
  - Hoare logic, proofs about programs

LAST TIME ON HOL

- natural deduction rules for $\wedge$, $\vee$ and $\longrightarrow$

LAST TIME ON HOL

- natural deduction rules for $\wedge$, $\vee$ and $\longrightarrow$
- proof by assumption

LAST TIME ON HOL

- natural deduction rules for $\wedge$, $\vee$ and $\longrightarrow$
- proof by assumption
- proof by intro rule

LAST TIME ON HOL

- natural deduction rules for $\wedge$, $\vee$ and $\longrightarrow$
- proof by assumption
- proof by intro rule
- proof by elim rule

More Proof Rules

IFF, NEGATION, TRUE AND FALSE

$$\frac{}{A = B}\ \text{iffi} \qquad \frac{A = B}{C}\ \text{iffE}$$

$$\frac{A = B}{}\ \text{iffD1} \qquad \frac{A = B}{}\ \text{iffD2}$$

$$\frac{}{\neg A}\ \text{notI} \qquad \frac{\neg A}{P}\ \text{notE}$$

IFF, NEGATION, TRUE AND FALSE

$$\frac{A \Longrightarrow B \quad B \Longrightarrow A}{A = B}\ \text{iffi} \qquad \frac{A = B}{C}\ \text{iffE}$$

$$\frac{A = B}{}\ \text{iffD1} \qquad \frac{A = B}{}\ \text{iffD2}$$

$$\frac{}{\neg A}\ \text{notI} \qquad \frac{\neg A}{P}\ \text{notE}$$

IFF, NEGATION, TRUE AND FALSE

$$\frac{A \Longrightarrow B \quad B \Longrightarrow A}{A = B}\ \text{iffi} \qquad \frac{A = B \quad [\![A \longrightarrow B; B \longrightarrow A]\!] \Longrightarrow C}{C}\ \text{iffE}$$

$$\frac{A = B}{}\ \text{iffD1} \qquad \frac{A = B}{}\ \text{iffD2}$$

$$\frac{}{\neg A}\ \text{notI} \qquad \frac{\neg A}{P}\ \text{notE}$$

IFF, NEGATION, TRUE AND FALSE

$$\frac{A \Longrightarrow B \quad B \Longrightarrow A}{A = B}\ \text{iffi} \qquad \frac{A = B \quad [\![A \longrightarrow B; B \longrightarrow A]\!] \Longrightarrow C}{C}\ \text{iffE}$$

$$\frac{A = B}{A \Longrightarrow B}\ \text{iffD1} \qquad \frac{A = B}{B \Longrightarrow A}\ \text{iffD2}$$

$$\frac{}{\neg A}\ \text{notI} \qquad \frac{\neg A}{P}\ \text{notE}$$

IFF, NEGATION, TRUE AND FALSE

$$\frac{A \Longrightarrow B \quad B \Longrightarrow A}{A = B}\ \text{iffi} \qquad \frac{A = B \quad [\![A \longrightarrow B; B \longrightarrow A]\!] \Longrightarrow C}{C}\ \text{iffE}$$

$$\frac{A = B}{A \Longrightarrow B}\ \text{iffD1} \qquad \frac{A = B}{B \Longrightarrow A}\ \text{iffD2}$$

$$\frac{A \Longrightarrow False}{\neg A}\ \text{notI} \qquad \frac{\neg A}{P}\ \text{notE}$$

IFF, NEGATION, TRUE AND FALSE

$$\frac{A \Longrightarrow B \quad B \Longrightarrow A}{A = B}\ \text{iffi} \qquad \frac{A = B \quad [\![A \longrightarrow B; B \longrightarrow A]\!] \Longrightarrow C}{C}\ \text{iffE}$$

$$\frac{A = B}{A \Longrightarrow B}\ \text{iffD1} \qquad \frac{A = B}{B \Longrightarrow A}\ \text{iffD2}$$

$$\frac{A \Longrightarrow False}{\neg A}\ \text{notI} \qquad \frac{\neg A \quad A}{P}\ \text{notE}$$

IFF, NEGATION, TRUE AND FALSE

$$\frac{A \Longrightarrow B \quad B \Longrightarrow A}{A = B}\ \text{iffi} \qquad \frac{A = B \quad [\![A \longrightarrow B; B \longrightarrow A]\!] \Longrightarrow C}{C}\ \text{iffE}$$

$$\frac{A = B}{A \Longrightarrow B}\ \text{iffD1} \qquad \frac{A = B}{B \Longrightarrow A}\ \text{iffD2}$$

$$\frac{A \Longrightarrow False}{\neg A}\ \text{notI} \qquad \frac{\neg A \quad A}{P}\ \text{notE}$$

$$\frac{}{True}\ \text{TrueI} \qquad \frac{False}{P}\ \text{FalseE}$$

EQUALITY

$$\frac{}{t = t}\ \text{refl} \qquad \frac{s = t}{t = s}\ \text{sym} \qquad \frac{r = s \quad s = t}{r = t}\ \text{trans}$$

EQUALITY

$$\frac{}{t = t}\ \text{refl} \qquad \frac{s = t}{t = s}\ \text{sym} \qquad \frac{r = s \quad s = t}{r = t}\ \text{trans}$$

$$\frac{s = t \quad P\ s}{P\ t}\ \text{subst}$$

EQUALITY

$$\frac{}{t = t}\ \text{refl} \qquad \frac{s = t}{t = s}\ \text{sym} \qquad \frac{r = s \quad s = t}{r = t}\ \text{trans}$$

$$\frac{s = t \quad P\ s}{P\ t}\ \text{subst}$$

Rarely needed explicitly — used implicitly by term rewriting

DEMO

$$\frac{}{P = True \vee P = False}$$ True-False

CLASSICAL

$$\frac{}{P = True \vee P = False}$$ True-False

$$\frac{}{P \vee \neg P}$$ excluded-middle

$$\frac{\neg A \Longrightarrow False}{A}$$ ccontr $$\frac{\neg A \Longrightarrow A}{A}$$ classical

CLASSICAL

$$\frac{}{P = True \vee P = False}$$ True-False

$$\frac{}{P \vee \neg P}$$ excluded-middle

$$\frac{\neg A \Longrightarrow False}{A}$$ ccontr $$\frac{\neg A \Longrightarrow A}{A}$$ classical

➜ **excluded-middle**, **ccontr** and **classical**
not derivable from the other rules.

CLASSICAL

$$\frac{}{P = True \vee P = False}$$ True-False

$$\frac{}{P \vee \neg P}$$ excluded-middle

$$\frac{\neg A \Longrightarrow False}{A}$$ ccontr $$\frac{\neg A \Longrightarrow A}{A}$$ classical

➜ **excluded-middle**, **ccontr** and **classical**
not derivable from the other rules.

➜ if we include True-False, they are derivable

They make the logic “classical”, “non-constructive”

CASES

$$\frac{}{P \vee \neg P}$$ excluded-middle

is a case distinction on type $bool$

CASES

$\frac{}{P \vee \neg P}$ excluded-middle

is a case distinction on type $bool$

Isabelle can do case distinctions on arbitrary terms:

apply (case_tac $term$)

SAFE AND NOT SO SAFE

Safe rules preserve provability

SAFE AND NOT SO SAFE

Safe rules preserve provability

conjI, impI, notI, iffi, refl, ccontr, classical, conjE, disjE

$$\frac{A \quad B}{A \wedge B}$$ conjI

SAFE AND NOT SO SAFE

Safe rules preserve provability

conjI, impI, notI, iffi, refl, ccontr, classical, conjE, disjE

$$\frac{A \quad B}{A \wedge B}$$ conjI

Unsafe rules can turn a provable goal into an unprovable one

SAFE AND NOT SO SAFE

Safe rules preserve provability

conjI, impI, notI, iffi, refl, ccontr, classical, conjE, disjE

$$\frac{A \quad B}{A \wedge B}\ \text{conjI}$$

Unsafe rules can turn a provable goal into an unprovable one

disjI1, disjI2, impE, iffD1, iffD2, notE

$$\frac{A}{A \vee B}\ \text{disjI1}$$

SAFE AND NOT SO SAFE

Safe rules preserve provability

conjI, impI, notI, iffi, refl, ccontr, classical, conjE, disjE

$$\frac{A \quad B}{A \wedge B}\ \text{conjI}$$

Unsafe rules can turn a provable goal into an unprovable one

disjI1, disjI2, impE, iffD1, iffD2, notE

$$\frac{A}{A \vee B}\ \text{disjI1}$$

Apply safe rules before unsafe ones

DEMO

QUANTIFIERS

SCOPE

- Scope of parameters: whole subgoal
- Scope of $\forall, \exists, \ldots$: ends with ; or $\Longrightarrow$

Example:

SCOPE

- Scope of parameters: whole subgoal
- Scope of $\forall, \exists, \ldots$: ends with ; or $\Longrightarrow$

Example:

$$\bigwedge x\ y.\ [\![\ \forall y.\ P\ y \longrightarrow Q\ z\ y;\ \ Q\ x\ y\ ]\!] \Longrightarrow\ \exists x.\ Q\ x\ y$$

means

Scope

- Scope of parameters: whole subgoal
- Scope of $\forall, \exists, \ldots$: ends with ; or $\Longrightarrow$

Example:

$$\bigwedge x\ y.\ [\![\ \forall y.\ P\ y \longrightarrow Q\ z\ y;\ \ Q\ x\ y\ ]\!] \Longrightarrow\ \exists x.\ Q\ x\ y$$

means

$$\bigwedge x\ y.\ [\![\ (\forall y_1.\ P\ y_1 \longrightarrow Q\ z\ y_1);\ \ Q\ x\ y\ ]\!] \Longrightarrow\ (\exists x_1.\ Q\ x_1\ y)$$

Natural deduction for quantifiers

$$\frac{}{\forall x.\ P\ x}\ \text{allI} \qquad \frac{\forall x.\ P\ x}{R}\ \text{allE}$$

$$\frac{}{\exists x.\ P\ x}\ \text{exI} \qquad \frac{\exists x.\ P\ x}{R}\ \text{exE}$$

NATURAL DEDUCTION FOR QUANTIFIERS

$$\frac{\bigwedge x.\ P\ x}{\forall x.\ P\ x}\ \text{allI} \qquad \frac{\forall x.\ P\ x}{R}\ \text{allE}$$

$$\frac{}{\exists x.\ P\ x}\ \text{exI} \qquad \frac{\exists x.\ P\ x}{R}\ \text{exE}$$

NATURAL DEDUCTION FOR QUANTIFIERS

$$\frac{\bigwedge x.\ P\ x}{\forall x.\ P\ x}\ \text{allI} \qquad \frac{\forall x.\ P\ x \quad P\ ?x \Longrightarrow R}{R}\ \text{allE}$$

$$\frac{}{\exists x.\ P\ x}\ \text{exI} \qquad \frac{\exists x.\ P\ x}{R}\ \text{exE}$$

NATURAL DEDUCTION FOR QUANTIFIERS

$$\frac{\bigwedge x.\ P\ x}{\forall x.\ P\ x}\ \text{allI} \qquad \frac{\forall x.\ P\ x \quad P\ ?x \Longrightarrow R}{R}\ \text{allE}$$

$$\frac{P\ ?x}{\exists x.\ P\ x}\ \text{exI} \qquad \frac{\exists x.\ P\ x}{R}\ \text{exE}$$

NATURAL DEDUCTION FOR QUANTIFIERS

$$\frac{\bigwedge x.\ P\ x}{\forall x.\ P\ x}\ \text{allI} \qquad \frac{\forall x.\ P\ x \quad P\ ?x \Longrightarrow R}{R}\ \text{allE}$$

$$\frac{P\ ?x}{\exists x.\ P\ x}\ \text{exI} \qquad \frac{\exists x.\ P\ x \quad \bigwedge x.\ P\ x \Longrightarrow R}{R}\ \text{exE}$$

NATURAL DEDUCTION FOR QUANTIFIERS

$$\frac{\bigwedge x.\ P\ x}{\forall x.\ P\ x}\ \text{allI} \qquad \frac{\forall x.\ P\ x \quad P\ ?x \Longrightarrow R}{R}\ \text{allE}$$

$$\frac{P\ ?x}{\exists x.\ P\ x}\ \text{exI} \qquad \frac{\exists x.\ P\ x \quad \bigwedge x.\ P\ x \Longrightarrow R}{R}\ \text{exE}$$

- **allI** and **exE** introduce new parameters $(\bigwedge x)$.
- **allE** and **exI** introduce new unknowns $(?x)$.

Instantiating Rules

apply (rule_tac x = "$term$" in $rule$)

Like **rule**, but $?x$ in $rule$ is instantiated by $term$ before application.

Similar: **erule_tac**

! x is in $rule$, not in goal !

Two Successful Proofs

1. $\forall x.\, \exists y.\, x = y$

Two Successful Proofs

1. $\forall x.\ \exists y.\ x = y$

apply (rule allI)

1. $\bigwedge x.\ \exists y.\ x = y$

Two Successful Proofs

1. $\forall x.\ \exists y.\ x = y$

apply (rule allI)

1. $\bigwedge x.\ \exists y.\ x = y$

best practice

apply (rule_tac x = "x" in exI)

1. $\bigwedge x.\ x = x$

Two Successful Proofs

1. $\forall x.\ \exists y.\ x = y$

apply (rule allI)

1. $\bigwedge x.\ \exists y.\ x = y$

best practice

apply (rule_tac x = "x" in exI)

1. $\bigwedge x.\ x = x$

apply (rule refl)

Two Successful Proofs

1. $\forall x.\ \exists y.\ x = y$

apply (rule allI)

1. $\bigwedge x.\ \exists y.\ x = y$

best practice

apply (rule_tac x = "x" in exI)

1. $\bigwedge x.\ x = x$

apply (rule refl)

exploration

apply (rule exI)

1. $\bigwedge x.\ x = ?y\ x$

Two Successful Proofs

1. $\forall x.\ \exists y.\ x = y$

apply (rule allI)

1. $\bigwedge x.\ \exists y.\ x = y$

best practice

apply (rule_tac x = "x" in exI)

1. $\bigwedge x.\ x = x$

apply (rule refl)

exploration

apply (rule exI)

1. $\bigwedge x.\ x = ?y\ x$

apply (rule refl)

$?y \mapsto \lambda u.u$

Two Successful Proofs

1. $\forall x.\ \exists y.\ x = y$

apply (rule allI)

1. $\bigwedge x.\ \exists y.\ x = y$

best practice

apply (rule_tac x = "x" in exI)

1. $\bigwedge x.\ x = x$

apply (rule refl)

simpler & clearer

exploration

apply (rule exI)

1. $\bigwedge x.\ x = ?y\ x$

apply (rule refl)

$?y \mapsto \lambda u.u$

shorter & trickier

Two Unsuccessful Proofs

1. $\exists y.\, \forall x.\, x = y$

Two Unsuccessful Proofs

1. $\exists y.\, \forall x.\, x = y$

apply (rule_tac x = ??? in exI)

Two Unsuccessful Proofs

1. $\exists y.\ \forall x.\ x = y$

apply (rule_tac x = ??? in exI)

apply (rule exI)

1. $\forall x.\ x = ?y$

Two Unsuccessful Proofs

1. $\exists y.\ \forall x.\ x = y$

apply (rule_tac x = ??? in exI)

apply (rule exI)

1. $\forall x.\ x = ?y$

apply (rule allI)

1. $\bigwedge x.\ x = ?y$

Two Unsuccessful Proofs

1. $\exists y.\ \forall x.\ x = y$

apply (rule_tac x = ??? in exI)

apply (rule exI)

1. $\forall x.\ x = ?y$

apply (rule allI)

1. $\bigwedge x.\ x = ?y$

apply (rule refl)

$?y \mapsto x$ yields $\bigwedge x'.x' = x$

Two Unsuccessful Proofs

1. $\exists y.\ \forall x.\ x = y$

apply (rule_tac x = ??? in exI)

apply (rule exI)

1. $\forall x.\ x = ?y$

apply (rule allI)

1. $\bigwedge x.\ x = ?y$

apply (rule refl)

$?y \mapsto x$ yields $\bigwedge x'.x' = x$

Principle:

$?f\ x_1 \ldots x_n$ **can only be replaced by term** t
if $params(t) \subseteq x_1, \ldots, x_n$

SAFE AND UNSAFE RULES

Safe allI, exE

Unsafe allE, exI

SAFE AND UNSAFE RULES

Safe allI, exE

Unsafe allE, exI

Create parameters first, unknowns later

DEMO: QUANTIFIER PROOFS

Parameter names are chosen by Isabelle

1. $\forall\, x.\, \exists y.\, x = y$

Parameter names are chosen by Isabelle

1. $\forall\, x.\, \exists y.\, x = y$

apply (rule allI)

1. $\bigwedge x.\, \exists y.\, x = y$

Parameter names are chosen by Isabelle

1. $\forall\, x.\, \exists y.\, x = y$

apply (rule allI)

1. $\bigwedge x.\, \exists y.\, x = y$

apply (rule_tac x = "x" in exI)

Brittle!

Renaming parameters

1. $\forall x.\ \exists y.\ x = y$

apply (rule allI)

1. $\bigwedge x.\ \exists y.\ x = y$

Renaming parameters

1. $\forall x.\ \exists y.\ x = y$

apply (rule allI)

1. $\bigwedge x.\ \exists y.\ x = y$

apply (rename_tac N)

1. $\bigwedge N.\ \exists y.\ N = y$

RENAMING PARAMETERS

1. $\forall x.\ \exists y.\ x = y$

apply (rule allI)

1. $\bigwedge x.\ \exists y.\ x = y$

apply (rename_tac N)

1. $\bigwedge N.\ \exists y.\ N = y$

apply (rule_tac x = "N" in exI)

In general:

(rename_tac $x_1 \ldots x_n$) renames the rightmost (inner) n parameters to $x_1 \ldots x_n$

Forward Proof: frule and drule

apply (frule $< rule >$)

Rule: $[\![A_1; \ldots; A_m]\!] \Longrightarrow A$

Subgoal: 1. $[\![B_1; \ldots; B_n]\!] \Longrightarrow C$

FORWARD PROOF: FRULE AND DRULE

apply (frule $< rule >$)

Rule:	$[\![A_1; \ldots; A_m]\!] \Longrightarrow A$
Subgoal:	1. $[\![B_1; \ldots; B_n]\!] \Longrightarrow C$
Substitution:	$\sigma(B_i) \equiv \sigma(A_1)$

FORWARD PROOF: FRULE AND DRULE

apply (frule $< rule >$)

Rule:	$[\![A_1; \ldots; A_m]\!] \Longrightarrow A$
Subgoal:	1. $[\![B_1; \ldots; B_n]\!] \Longrightarrow C$
Substitution:	$\sigma(B_i) \equiv \sigma(A_1)$
New subgoals:	1. $\sigma([\![B_1; \ldots; B_n]\!] \Longrightarrow A_2)$
	$\vdots$
	m-1. $\sigma([\![B_1; \ldots; B_n]\!] \Longrightarrow A_m)$
	m. $\sigma([\![B_1; \ldots; B_n; A]\!] \Longrightarrow C)$

FORWARD PROOF: FRULE AND DRULE

apply (frule $< rule >$)

Rule:	$[\![A_1; \ldots; A_m]\!] \Longrightarrow A$
Subgoal:	1. $[\![B_1; \ldots; B_n]\!] \Longrightarrow C$
Substitution:	$\sigma(B_i) \equiv \sigma(A_1)$
New subgoals:	1. $\sigma([\![B_1; \ldots; B_n]\!] \Longrightarrow A_2)$
	$\vdots$
	m-1. $\sigma([\![B_1; \ldots; B_n]\!] \Longrightarrow A_m)$
	m. $\sigma([\![B_1; \ldots; B_n; A]\!] \Longrightarrow C)$

Like **frule** but also deletes B_i: **apply** (drule $< rule >$)

Examples for Forward Rules

$$\frac{P \wedge Q}{P}\ \text{conjunct1} \qquad \frac{P \wedge Q}{Q}\ \text{conjunct2}$$

$$\frac{P \longrightarrow Q \quad P}{Q}\ \text{mp}$$

$$\frac{\forall x.\ P\ x}{P\ ?x}\ \text{spec}$$

Forward Proof: OF

$$r\ [\textbf{OF}\ r_1 \ldots r_n]$$

Prove assumption 1 of theorem r with theorem r_1, and assumption 2 with theorem r_2, and ...

Forward Proof: OF

$$r\ [\textbf{OF}\ r_1 \ldots r_n]$$

Prove assumption 1 of theorem r with theorem r_1, and assumption 2 with theorem r_2, and ...

Rule r	$[\![A_1; \ldots; A_m]\!] \Longrightarrow A$
Rule r_1	$[\![B_1; \ldots; B_n]\!] \Longrightarrow B$

Forward Proof: OF

$$r\ [\textbf{OF}\ r_1 \ldots r_n]$$

Prove assumption 1 of theorem r with theorem r_1, and assumption 2 with theorem r_2, and ...

Rule r	$[\![A_1; \ldots; A_m]\!] \Longrightarrow A$
Rule r_1	$[\![B_1; \ldots; B_n]\!] \Longrightarrow B$
Substitution	$\sigma(B) \equiv \sigma(A_1)$

Forward Proof: OF

$$r\ [\textbf{OF}\ r_1 \ldots r_n]$$

Prove assumption 1 of theorem r with theorem r_1, and assumption 2 with theorem r_2, and ...

Rule r	$[\![A_1; \ldots; A_m]\!] \Longrightarrow A$
Rule r_1	$[\![B_1; \ldots; B_n]\!] \Longrightarrow B$
Substitution	$\sigma(B) \equiv \sigma(A_1)$
r [OF r_1]	$\sigma([\![B_1; \ldots; B_n; A_2; \ldots; A_m]\!] \Longrightarrow A)$

FORWARD PROOFS: THEN

r_1 [THEN r_2] means r_2 [OF r_1]

DEMO: FORWARD PROOFS

Hilbert's Epsilon Operator

(David Hilbert, 1862-1943)

$\varepsilon\ x.\ Px$ is a value that satisfies P (if such a value exists)

HILBERT'S EPSILON OPERATOR

(David Hilbert, 1862-1943)

$\varepsilon\ x.\ Px$ is a value that satisfies P (if such a value exists)

ε also known as **description operator**.

In Isabelle the ε-operator is written SOME $x.\ P\ x$

HILBERT'S EPSILON OPERATOR

(David Hilbert, 1862-1943)

$\varepsilon\ x.\ Px$ is a value that satisfies P (if such a value exists)

ε also known as **description operator**.
In Isabelle the ε-operator is written SOME $x.\ P\ x$

$$\frac{P\ ?x}{P\ (\text{SOME}\ x.\ P\ x)}\ \text{someI}$$

More Epsilon

ε implies Axiom of Choice:

$$\forall x.\ \exists y.\ Q\ x\ y \Longrightarrow \exists f.\ \forall x.\ Q\ x\ (f\ x)$$

Existential and universial quantification can be defined with ε.

More Epsilon

ε implies Axiom of Choice:

$$\forall x.\ \exists y.\ Q\ x\ y \Longrightarrow \exists f.\ \forall x.\ Q\ x\ (f\ x)$$

Existential and universial quantification can be defined with ε.

Isabelle also know the definite description operator **THE** (also ι):

$$\frac{}{(\mathsf{THE}\ x.\ x = a) = a}\ \text{the\_eq\_trivial}$$

SOME AUTOMATION

More Proof Methods:

apply (intro <intro-rules>)	repeatedly applies intro rules
apply (elim <elim-rules>)	repeatedly applies elim rules

SOME AUTOMATION

More Proof Methods:

apply (intro <intro-rules>)	repeatedly applies intro rules
apply (elim <elim-rules>)	repeatedly applies elim rules
apply clarify	applies all safe rules that do not split the goal

SOME AUTOMATION

More Proof Methods:

apply (intro <intro-rules>)	repeatedly applies intro rules
apply (elim <elim-rules>)	repeatedly applies elim rules
apply clarify	applies all safe rules that do not split the goal
apply safe	applies all safe rules

SOME AUTOMATION

More Proof Methods:

apply (intro <intro-rules>)	repeatedly applies intro rules
apply (elim <elim-rules>)	repeatedly applies elim rules
apply clarify	applies all safe rules that do not split the goal
apply safe	applies all safe rules
apply blast	an automatic tableaux prover (works well on predicate logic)

SOME AUTOMATION

More Proof Methods:

apply (intro <intro-rules>)	repeatedly applies intro rules
apply (elim <elim-rules>)	repeatedly applies elim rules
apply clarify	applies all safe rules that do not split the goal
apply safe	applies all safe rules
apply blast	an automatic tableaux prover (works well on predicate logic)
apply fast	another automatic search tactic

Epsilon and Automation Demo

WE HAVE LEARNED SO FAR...

➜ Proof rules for negation and contradiction

WE HAVE LEARNED SO FAR...

- ➜ Proof rules for negation and contradiction
- ➜ Proof rules for predicate calculus

WE HAVE LEARNED SO FAR...

- ➜ Proof rules for negation and contradiction
- ➜ Proof rules for predicate calculus
- ➜ Safe and unsafe rules

WE HAVE LEARNED SO FAR...

- ➜ Proof rules for negation and contradiction
- ➜ Proof rules for predicate calculus
- ➜ Safe and unsafe rules
- ➜ Forward Proof

WE HAVE LEARNED SO FAR...

- ➜ Proof rules for negation and contradiction
- ➜ Proof rules for predicate calculus
- ➜ Safe and unsafe rules
- ➜ Forward Proof
- ➜ The Epsilon Operator

WE HAVE LEARNED SO FAR...

- ➜ Proof rules for negation and contradiction
- ➜ Proof rules for predicate calculus
- ➜ Safe and unsafe rules
- ➜ Forward Proof
- ➜ The Epsilon Operator
- ➜ Some automation

Exercises

➜ Download the exercise file and prove all theorems in there.

➜ Prove or disprove:

If every poor person has a rich mother, then there is a rich person with a rich grandmother.