

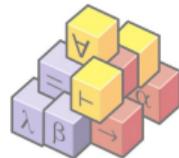
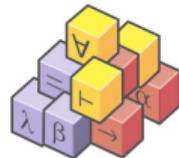
🌻 Automating Natural Deduction 1 🌻

Simon Foster Jim Woodcock
University of York

18th August 2022

Overview

- 1 Low-Level Proof Scripts
- 2 Natural Deduction Rules for Propositional Calculus
- 3 Automation with the Classical Reasoner



Outline

- 1 Low-Level Proof Scripts
- 2 Natural Deduction Rules for Propositional Calculus
- 3 Automation with the Classical Reasoner

Motivation

- The `simp` tactic uses equations ($s = t$) to rewrite and simplify a goal.
- Proof by simplification is the **gold standard** for efficiency and usability.
- But it's not always possible to prove a goal with just the simplifier.
- Isabelle provides tactics for automating **natural deduction**.
- One of the **core reasoning techniques**, alongside the simplifier.
- You need to understand the use of low-level **proof scripts**.

Proof Script

```
lemma "P  $\implies$  (P  $\wedge$  (P  $\vee$  Q))"  
  apply (rule conjI)  
  apply assumption  
  apply (rule disjI1)  
  apply assumption  
  done
```

Motivation

- The **simp** tactic uses equations ($s = t$) to rewrite and simplify a goal.
- Proof by simplification is the **gold standard** for efficiency and usability.
- But it's not always possible to prove a goal with just the simplifier.
- Isabelle provides tactics for automating **natural deduction**.
- One of the **core reasoning techniques**, alongside the simplifier.
- You need to understand the use of low-level **proof scripts**.

Proof Script

```
lemma "P  $\implies$  (P  $\wedge$  (P  $\vee$  Q))"  
  apply (rule conjI)  
  apply assumption  
  apply (rule disjI1)  
  apply assumption  
  done
```

Motivation

- The **simp** tactic uses equations ($s = t$) to rewrite and simplify a goal.
- Proof by simplification is the **gold standard** for efficiency and usability.
- But it's not always possible to prove a goal with just the simplifier.
- Isabelle provides tactics for automating **natural deduction**.
- One of the **core reasoning techniques**, alongside the simplifier.
- You need to understand the use of low-level **proof scripts**.

Proof Script

```
lemma "P  $\implies$  (P  $\wedge$  (P  $\vee$  Q))"  
  apply (rule conjI)  
  apply assumption  
  apply (rule disjI1)  
  apply assumption  
  done
```

Motivation

- The **simp** tactic uses equations ($s = t$) to rewrite and simplify a goal.
- Proof by simplification is the **gold standard** for efficiency and usability.
- But it's not always possible to prove a goal with just the simplifier.
- Isabelle provides tactics for automating **natural deduction**.
- One of the **core reasoning techniques**, alongside the simplifier.
- You need to understand the use of low-level **proof scripts**.

Proof Script

```
lemma "P  $\implies$  (P  $\wedge$  (P  $\vee$  Q))"  
  apply (rule conjI)  
  apply assumption  
  apply (rule disjI1)  
  apply assumption  
  done
```

Motivation

- The **simp** tactic uses equations ($s = t$) to rewrite and simplify a goal.
- Proof by simplification is the **gold standard** for efficiency and usability.
- But it's not always possible to prove a goal with just the simplifier.
- Isabelle provides tactics for automating **natural deduction**.
- One of the **core reasoning techniques**, alongside the simplifier.
- You need to understand the use of low-level **proof scripts**.

Proof Script

```
lemma "P  $\implies$  (P  $\wedge$  (P  $\vee$  Q))"  
  apply (rule conjI)  
  apply assumption  
  apply (rule disjI1)  
  apply assumption  
  done
```

Motivation

- The **simp** tactic uses equations ($s = t$) to rewrite and simplify a goal.
- Proof by simplification is the **gold standard** for efficiency and usability.
- But it's not always possible to prove a goal with just the simplifier.
- Isabelle provides tactics for automating **natural deduction**.
- One of the **core reasoning techniques**, alongside the simplifier.
- You need to understand the use of low-level **proof scripts**.

Proof Script

```
lemma "P  $\implies$  (P  $\wedge$  (P  $\vee$  Q))"  
  apply (rule conjI)  
  apply assumption  
  apply (rule disjI1)  
  apply assumption  
  done
```

Motivation

- The **simp** tactic uses equations ($s = t$) to rewrite and simplify a goal.
- Proof by simplification is the **gold standard** for efficiency and usability.
- But it's not always possible to prove a goal with just the simplifier.
- Isabelle provides tactics for automating **natural deduction**.
- One of the **core reasoning techniques**, alongside the simplifier.
- You need to understand the use of low-level **proof scripts**.

Proof Script

```
lemma "P  $\implies$  (P  $\wedge$  (P  $\vee$  Q))"  
  apply (rule conjI)  
  apply assumption  
  apply (rule disjI1)  
  apply assumption  
  done
```

Motivation

- The **simp** tactic uses equations ($s = t$) to rewrite and simplify a goal.
- Proof by simplification is the **gold standard** for efficiency and usability.
- But it's not always possible to prove a goal with just the simplifier.
- Isabelle provides tactics for automating **natural deduction**.
- One of the **core reasoning techniques**, alongside the simplifier.
- You need to understand the use of low-level **proof scripts**.

Proof Script

```
lemma "P  $\implies$  (P  $\wedge$  (P  $\vee$  Q))"  
  apply (rule conjI)  
  apply assumption  
  apply (rule disjI1)  
  apply assumption  
  done
```

Motivation

- The **simp** tactic uses equations ($s = t$) to rewrite and simplify a goal.
- Proof by simplification is the **gold standard** for efficiency and usability.
- But it's not always possible to prove a goal with just the simplifier.
- Isabelle provides tactics for automating **natural deduction**.
- One of the **core reasoning techniques**, alongside the simplifier.
- You need to understand the use of low-level **proof scripts**.

Proof Script

```
lemma "P  $\implies$  (P  $\wedge$  (P  $\vee$  Q))"  
  apply (rule conjI)  
  apply assumption  
  apply (rule disjI1)  
  apply assumption  
  done
```

Motivation

- The **simp** tactic uses equations ($s = t$) to rewrite and simplify a goal.
- Proof by simplification is the **gold standard** for efficiency and usability.
- But it's not always possible to prove a goal with just the simplifier.
- Isabelle provides tactics for automating **natural deduction**.
- One of the **core reasoning techniques**, alongside the simplifier.
- You need to understand the use of low-level **proof scripts**.

Proof Script

```
lemma "P  $\implies$  (P  $\wedge$  (P  $\vee$  Q))"  
  apply (rule conjI)  
  apply assumption  
  apply (rule disjI1)  
  apply assumption  
  done
```

Proof Scripts and Tactics

- `Low-level proof language` are the “machine code” for Isar.
- `Proof script`: sequence of commands acting on a proof state.
- `Proof state`:
 - Current goal.
 - Current collection of hypotheses and outstanding conjectures.
- Commands invoke `proof tactics` to decompose and refine the goals.

- `Proof script` uses the `apply` keyword to describe a tactic.
- Tactics can fail (have `subject` in the wrong form).
- `Proof` is complete once all `subgoals` are eliminated.
- `Goal` terminated with `done` (`qed`).

Proof Scripts and Tactics

- Low-level proof language are the “machine code” for Isar.
- Proof script: sequence of commands acting on a proof state.
- Proof state: subgoals.
- Subgoals: collection of hypotheses and outstanding conjectures.
- Commands invoke proof tactics to decompose and refine the goals.

• Proof script uses the `apply` keyword to describe a tactic.

• Tactics can fail (give subgoals) in the wrong form.

• Proof is complete once all subgoals are eliminated.

• Some are related with `apply` and `use`.

Proof Scripts and Tactics

- Low-level proof language are the “machine code” for Isar.
- **Proof script**: sequence of commands acting on a **proof state**.
- Proof state: subgoals.
- Subgoals: collection of hypotheses and outstanding conjectures.
- Commands invoke **proof tactics** to decompose and **refine** the goals.

• Proof script uses the `proof` keyword to describe a tactic.

• Tactics can fail (due to goal not in the wrong form).

• Proof is complete once all subgoals are eliminated.

• Some are defined with `by`.

Proof Scripts and Tactics

- Low-level proof language are the “machine code” for Isar.
- **Proof script**: sequence of commands acting on a **proof state**.
- **Proof state**: **subgoals**.
- Subgoals: collection of hypotheses and outstanding conjectures.
- Commands invoke **proof tactics** to decompose and **refine** the goals.

• Proof script uses the `proof` keyword to describe a tactic.

• `proof` is followed by a sequence of tactic commands.

• `proof` is complete once all subgoals are eliminated.

• See `proof` chapter in the `IsarRef` manual.

Proof Scripts and Tactics

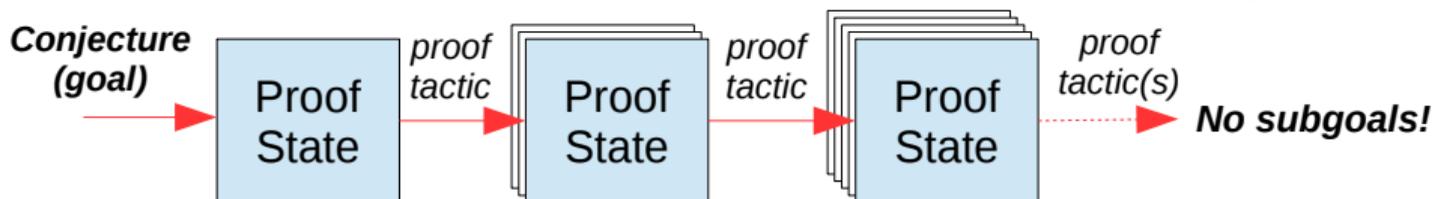
- Low-level proof language are the “machine code” for Isar.
- **Proof script**: sequence of commands acting on a **proof state**.
- **Proof state**: **subgoals**.
- **Subgoals**: collection of hypotheses and outstanding conjectures.
- Commands invoke **proof tactics** to decompose and **refine** the goals.

Proof Scripts and Tactics

- Low-level proof language are the “machine code” for Isar.
- **Proof script**: sequence of commands acting on a **proof state**.
- **Proof state**: **subgoals**.
- **Subgoals**: collection of hypotheses and outstanding conjectures.
- Commands invoke **proof tactics** to decompose and **refine** the goals.

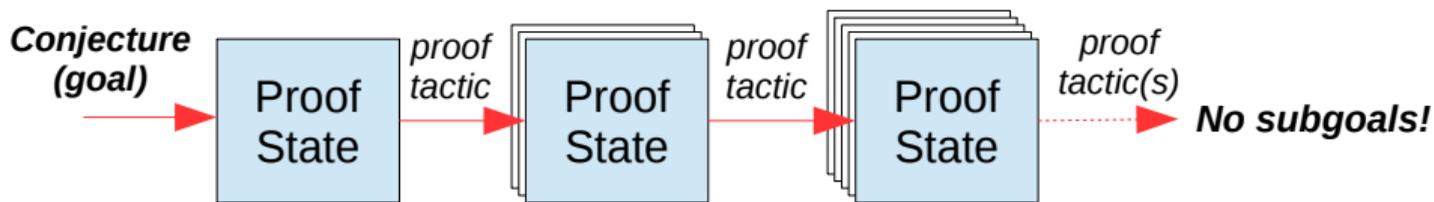
Proof Scripts and Tactics

- Low-level proof language are the “machine code” for Isar.
- **Proof script**: sequence of commands acting on a **proof state**.
- **Proof state**: **subgoals**.
- **Subgoals**: collection of hypotheses and outstanding conjectures.
- Commands invoke **proof tactics** to decompose and **refine** the goals.



Proof Scripts and Tactics

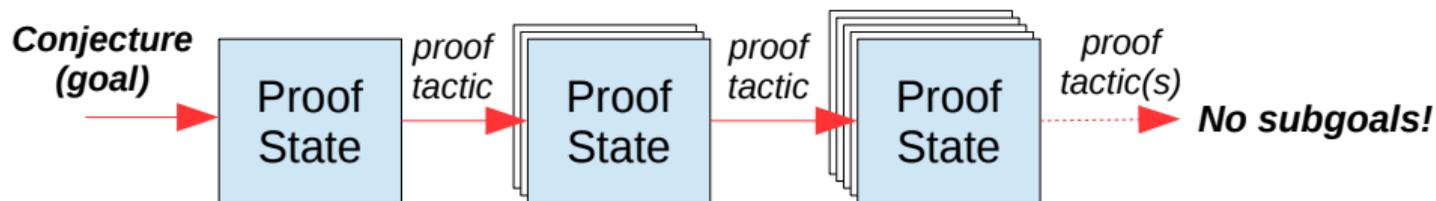
- Low-level proof language are the “machine code” for Isar.
- **Proof script**: sequence of commands acting on a **proof state**.
- **Proof state**: **subgoals**.
- **Subgoals**: collection of hypotheses and outstanding conjectures.
- Commands invoke **proof tactics** to decompose and **refine** the goals.



- Proof script uses the **apply** keyword to execute a tactic.
- Tactics can fail if the subgoal is in the wrong form.
- Proof is complete once all subgoals are eliminated.
- Script terminated with **done** (cf. **qed**).

Proof Scripts and Tactics

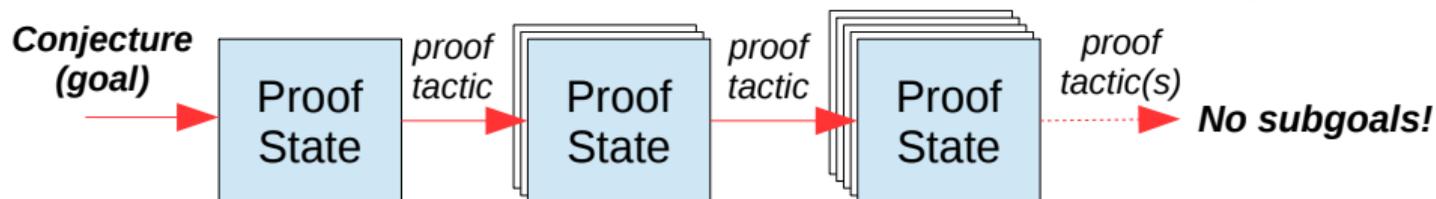
- Low-level proof language are the “machine code” for Isar.
- **Proof script**: sequence of commands acting on a **proof state**.
- **Proof state**: **subgoals**.
- **Subgoals**: collection of hypotheses and outstanding conjectures.
- Commands invoke **proof tactics** to decompose and **refine** the goals.



- Proof script uses the **apply** keyword to execute a tactic.
- Tactics can fail if the subgoal is in the wrong form.
- Proof is complete once all subgoals are eliminated.
- Script terminated with **done** (cf. **qed**).

Proof Scripts and Tactics

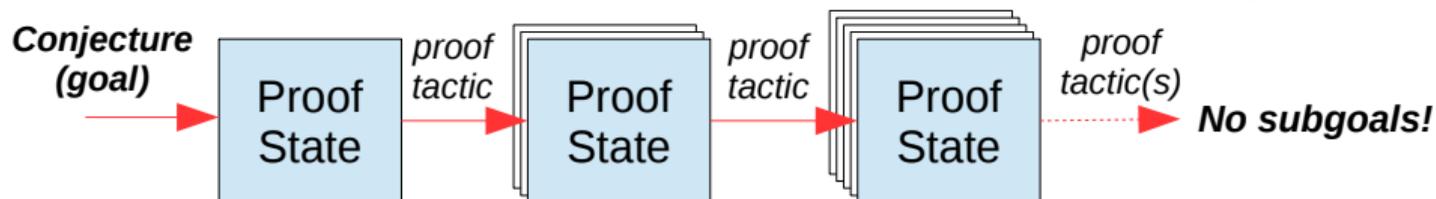
- Low-level proof language are the “machine code” for Isar.
- **Proof script**: sequence of commands acting on a **proof state**.
- **Proof state**: **subgoals**.
- **Subgoals**: collection of hypotheses and outstanding conjectures.
- Commands invoke **proof tactics** to decompose and **refine** the goals.



- Proof script uses the **apply** keyword to execute a tactic.
- Tactics can fail if the subgoal is in the wrong form.
- Proof is complete once all subgoals are eliminated.
- Script terminated with **done** (cf. **qed**).

Proof Scripts and Tactics

- Low-level proof language are the “machine code” for Isar.
- **Proof script**: sequence of commands acting on a **proof state**.
- **Proof state**: **subgoals**.
- **Subgoals**: collection of hypotheses and outstanding conjectures.
- Commands invoke **proof tactics** to decompose and **refine** the goals.



- Proof script uses the **apply** keyword to execute a tactic.
- Tactics can fail if the subgoal is in the wrong form.
- Proof is complete once all subgoals are eliminated.
- Script terminated with **done** (cf. **qed**).

Proof State

- The proof state consists of n subgoals.
- The aim is to discharge (remove by proof) all remaining subgoals.
- Subgoals have the form $[P_1, \dots, P_k] \rightarrow Q$.
- P_1, \dots, P_k is a set of hypotheses – can be used to prove the subgoal.
- Q is the conclusion of the subgoal (the thing to be proved).
- Example: $\forall x. \exists y. x + y = 0$ creates a single initial subgoal:
 $[\forall x. x + 0 = x] \rightarrow \exists y. x + y = 0$
- Is not closed on the first subgoal, but closed on all (simp_1-1).
- We can manipulate a subgoal in several ways:
 - Split into several additional subgoals by introduction or elimination.
 - Merge into one assumption or insert further ones, e.g. from theorems.
 - Discharge a literal subgoal: e.g. True or $P \rightarrow P$ using assumption.

Proof State

- The proof state consists of n subgoals.
- The aim is to **discharge** (remove by proof) all remaining subgoals.
- Subgoals have the form $\llbracket P_1; \dots; P_n \rrbracket \implies Q$.
- $P_1 \dots P_n$ is a set of hypotheses – can be used to prove the subgoal.
- Q is the conclusion of the subgoal; the thing to be proved.
- **assumes** $P_1 \dots P_n$ **shows** Q creates a single initial subgoal:
$$\llbracket P_1; \dots; P_n \rrbracket \implies Q$$
- Tactics often act on the **first subgoal**, but some act on **all** (`simp_all`).
- We can manipulate a subgoal in several ways:
- Split into several additional subgoals by **introduction** or **elimination**.
- Manipulate the assumptions or insert further ones, e.g. from theorems.
- Discharge **trivial** subgoals, e.g. True or $P \implies P$ using **assumption**.

Proof State

- The proof state consists of n subgoals.
- The aim is to **discharge** (remove by proof) all remaining subgoals.
- Subgoals have the form $\llbracket P_1; \dots; P_n \rrbracket \implies Q$.
- $P_1 \dots P_n$ is a set of hypotheses – can be used to prove the subgoal.
- Q is the conclusion of the subgoal; the thing to be proved.
- **assumes** $P_1 \dots P_n$ **shows** Q creates a single initial subgoal:
$$\llbracket P_1; \dots; P_n \rrbracket \implies Q$$
- Tactics often act on the **first subgoal**, but some act on **all** (`simp_all`).
- We can manipulate a subgoal in several ways:
- Split into several additional subgoals by **introduction** or **elimination**.
- Manipulate the assumptions or insert further ones, e.g. from theorems.
- Discharge **trivial** subgoals, e.g. True or $P \implies P$ using **assumption**.

Proof State

- The proof state consists of n subgoals.
- The aim is to **discharge** (remove by proof) all remaining subgoals.
- Subgoals have the form $\llbracket P_1; \dots; P_n \rrbracket \implies Q$.
 - $P_1 \dots P_n$ is a set of hypotheses – can be used to prove the subgoal.
 - Q is the conclusion of the subgoal; the thing to be proved.
 - **assumes** $P_1 \dots P_n$ **shows** Q creates a single initial subgoal:
$$\llbracket P_1; \dots; P_n \rrbracket \implies Q$$
 - Tactics often act on the **first subgoal**, but some act on **all** (`simp_all`).
 - We can manipulate a subgoal in several ways:
 - Split into several additional subgoals by **introduction** or **elimination**.
 - Manipulate the assumptions or insert further ones, e.g. from theorems.
 - Discharge **trivial** subgoals, e.g. True or $P \implies P$ using **assumption**.

Proof State

- The proof state consists of n subgoals.
- The aim is to **discharge** (remove by proof) all remaining subgoals.
- Subgoals have the form $\llbracket P_1; \dots; P_n \rrbracket \implies Q$.
- $P_1 \dots P_n$ is a set of hypotheses – can be used to prove the subgoal.
- Q is the conclusion of the subgoal; the thing to be proved.
- **assumes** $P_1 \dots P_n$ **shows** Q creates a single initial subgoal:
$$\llbracket P_1; \dots; P_n \rrbracket \implies Q$$
- Tactics often act on the **first subgoal**, but some act on **all** (`simp_all`).
- We can manipulate a subgoal in several ways:
- Split into several additional subgoals by **introduction** or **elimination**.
- Manipulate the assumptions or insert further ones, e.g. from theorems.
- Discharge **trivial** subgoals, e.g. True or $P \implies P$ using **assumption**.

Proof State

- The proof state consists of n subgoals.
- The aim is to **discharge** (remove by proof) all remaining subgoals.
- Subgoals have the form $\llbracket P_1; \dots; P_n \rrbracket \implies Q$.
- $P_1 \dots P_n$ is a set of hypotheses – can be used to prove the subgoal.
- Q is the conclusion of the subgoal; the thing to be proved.
- **assumes** $P_1 \dots P_n$ **shows** Q creates a single initial subgoal:
$$\llbracket P_1; \dots; P_n \rrbracket \implies Q$$
- Tactics often act on the **first subgoal**, but some act on **all** (`simp_all`).
- We can manipulate a subgoal in several ways:
- Split into several additional subgoals by **introduction** or **elimination**.
- Manipulate the assumptions or insert further ones, e.g. from theorems.
- Discharge **trivial** subgoals, e.g. True or $P \implies P$ using **assumption**.

Proof State

- The proof state consists of n subgoals.
- The aim is to **discharge** (remove by proof) all remaining subgoals.
- Subgoals have the form $\llbracket P_1; \dots; P_n \rrbracket \implies Q$.
- $P_1 \dots P_n$ is a set of hypotheses – can be used to prove the subgoal.
- Q is the conclusion of the subgoal; the thing to be proved.
- **assumes** $P_1 \dots P_n$ **shows** Q creates a single initial subgoal:

$$\llbracket P_1; \dots; P_n \rrbracket \implies Q$$

- Tactics often act on the **first subgoal**, but some act on **all** (`simp_all`).
- We can manipulate a subgoal in several ways:
- Split into several additional subgoals by **introduction** or **elimination**.
- Manipulate the assumptions or insert further ones, e.g. from theorems.
- Discharge **trivial** subgoals, e.g. True or $P \implies P$ using **assumption**.

Proof State

- The proof state consists of n subgoals.
- The aim is to **discharge** (remove by proof) all remaining subgoals.
- Subgoals have the form $\llbracket P_1; \dots; P_n \rrbracket \implies Q$.
- $P_1 \dots P_n$ is a set of hypotheses – can be used to prove the subgoal.
- Q is the conclusion of the subgoal; the thing to be proved.
- **assumes** $P_1 \dots P_n$ **shows** Q creates a single initial subgoal:

$$\llbracket P_1; \dots; P_n \rrbracket \implies Q$$

- Tactics often act on the **first subgoal**, but some act on **all** (`simp_all`).
- We can manipulate a subgoal in several ways:
- Split into several additional subgoals by **introduction** or **elimination**.
- Manipulate the assumptions or insert further ones, e.g. from theorems.
- Discharge **trivial** subgoals, e.g. True or $P \implies P$ using **assumption**.

Proof State

- The proof state consists of n subgoals.
- The aim is to **discharge** (remove by proof) all remaining subgoals.
- Subgoals have the form $\llbracket P_1; \dots; P_n \rrbracket \implies Q$.
- $P_1 \dots P_n$ is a set of hypotheses – can be used to prove the subgoal.
- Q is the conclusion of the subgoal; the thing to be proved.
- **assumes** $P_1 \dots P_n$ **shows** Q creates a single initial subgoal:
$$\llbracket P_1; \dots; P_n \rrbracket \implies Q$$
- Tactics often act on the **first subgoal**, but some act on **all** (**simp_all**).
- We can manipulate a subgoal in several ways:
- Split into several additional subgoals by **introduction** or **elimination**.
- Manipulate the assumptions or insert further ones, e.g. from theorems.
- Discharge **trivial** subgoals, e.g. True or $P \implies P$ using **assumption**.

Proof State

- The proof state consists of n subgoals.
- The aim is to **discharge** (remove by proof) all remaining subgoals.
- Subgoals have the form $\llbracket P_1; \dots; P_n \rrbracket \implies Q$.
- $P_1 \dots P_n$ is a set of hypotheses – can be used to prove the subgoal.
- Q is the conclusion of the subgoal; the thing to be proved.
- **assumes** $P_1 \dots P_n$ **shows** Q creates a single initial subgoal:
$$\llbracket P_1; \dots; P_n \rrbracket \implies Q$$
- Tactics often act on the **first subgoal**, but some act on **all** (**simp_all**).
- We can manipulate a subgoal in several ways:
 - Split into several additional subgoals by **introduction** or **elimination**.
 - Manipulate the assumptions or insert further ones, e.g. from theorems.
 - Discharge **trivial** subgoals, e.g. True or $P \implies P$ using **assumption**.

Proof State

- The proof state consists of n subgoals.
- The aim is to **discharge** (remove by proof) all remaining subgoals.
- Subgoals have the form $\llbracket P_1; \dots; P_n \rrbracket \implies Q$.
- $P_1 \dots P_n$ is a set of hypotheses – can be used to prove the subgoal.
- Q is the conclusion of the subgoal; the thing to be proved.
- **assumes** $P_1 \dots P_n$ **shows** Q creates a single initial subgoal:
$$\llbracket P_1; \dots; P_n \rrbracket \implies Q$$
- Tactics often act on the **first subgoal**, but some act on **all** (`simp_all`).
- We can manipulate a subgoal in several ways:
- Split into several additional subgoals by **introduction** or **elimination**.
- Manipulate the assumptions or insert further ones, e.g. from theorems.
- Discharge **trivial** subgoals, e.g. True or $P \implies P$ using `assumption`.

Proof State

- The proof state consists of n subgoals.
- The aim is to **discharge** (remove by proof) all remaining subgoals.
- Subgoals have the form $\llbracket P_1; \dots; P_n \rrbracket \implies Q$.
- $P_1 \dots P_n$ is a set of hypotheses – can be used to prove the subgoal.
- Q is the conclusion of the subgoal; the thing to be proved.
- **assumes** $P_1 \dots P_n$ **shows** Q creates a single initial subgoal:
$$\llbracket P_1; \dots; P_n \rrbracket \implies Q$$
- Tactics often act on the **first subgoal**, but some act on **all** (**simp_all**).
- We can manipulate a subgoal in several ways:
- Split into several additional subgoals by **introduction** or **elimination**.
- Manipulate the assumptions or insert further ones, e.g. from theorems.
- Discharge **trivial** subgoals, e.g. True or $P \implies P$ using **assumption**.

Proof State

- The proof state consists of n subgoals.
- The aim is to **discharge** (remove by proof) all remaining subgoals.
- Subgoals have the form $\llbracket P_1; \dots; P_n \rrbracket \implies Q$.
- $P_1 \dots P_n$ is a set of hypotheses – can be used to prove the subgoal.
- Q is the conclusion of the subgoal; the thing to be proved.
- **assumes** $P_1 \dots P_n$ **shows** Q creates a single initial subgoal:
$$\llbracket P_1; \dots; P_n \rrbracket \implies Q$$
- Tactics often act on the **first subgoal**, but some act on **all** (**simp_all**).
- We can manipulate a subgoal in several ways:
- Split into several additional subgoals by **introduction** or **elimination**.
- Manipulate the assumptions or insert further ones, e.g. from theorems.
- Discharge **trivial** subgoals, e.g. **True** or $P \implies P$ using **assumption**.

Outline

- 1 Low-Level Proof Scripts
- 2 Natural Deduction Rules for Propositional Calculus
- 3 Automation with the Classical Reasoner

Natural Deduction

Natural Deduction Rules

$$\frac{\Gamma \vdash P_1 \quad \Gamma \vdash P_2 \cdots \Gamma \vdash P_n}{\Gamma \vdash Q}$$

- Meaning: a proof of the conclusion Q follows from premises $P_1 \cdots P_n$.
- $\Gamma \vdash Q$ is a sequent: valid when Q holds subject to the hypotheses in Γ .

Example (Syllogism)

$$\frac{\text{Socrates is a man} \quad \text{All men are mortal}}{\text{Socrates is a mortal}}$$

Two main kinds of deduction rules:

- Introduction rules (rule towards backwards reasoning)
- Elimination rules (rule away from backwards reasoning)

Natural Deduction

Natural Deduction Rules

$$\frac{\Gamma \vdash P_1 \quad \Gamma \vdash P_2 \cdots \Gamma \vdash P_n}{\Gamma \vdash Q}$$

- **Meaning:** a proof of the **conclusion** Q follows from **premises** $P_1 \cdots P_n$.
- $\Gamma \vdash Q$ is a **sequent**: valid when Q holds subject to the hypotheses in Γ .

Example (Syllogism)

$$\frac{\text{Socrates is a man} \quad \text{All men are mortal}}{\text{Socrates is a mortal}}$$

Two main kinds of deduction rules:

• **Introduction rules** (rule for **deriving** a formula)

• **Elimination rules** (rule for **using** a formula)

Natural Deduction

Natural Deduction Rules

$$\frac{\Gamma \vdash P_1 \quad \Gamma \vdash P_2 \cdots \Gamma \vdash P_n}{\Gamma \vdash Q}$$

- **Meaning:** a proof of the **conclusion** Q follows from **premises** $P_1 \cdots P_n$.
- $\Gamma \vdash Q$ is a **sequent**: valid when Q holds subject to the hypotheses in Γ .

Example (Syllogism)

$$\frac{\text{Socrates is a man} \quad \text{All men are mortal}}{\text{Socrates is a mortal}}$$

Two main kinds of deduction rules:

• **Introduction rules** (rule for \vdash) towards reasoning

• **Elimination rules** (rule for \vdash) towards reasoning

Natural Deduction

Natural Deduction Rules

$$\frac{\Gamma \vdash P_1 \quad \Gamma \vdash P_2 \cdots \Gamma \vdash P_n}{\Gamma \vdash Q}$$

- **Meaning:** a proof of the **conclusion** Q follows from **premises** $P_1 \cdots P_n$.
- $\Gamma \vdash Q$ is a **sequent**: valid when Q holds subject to the hypotheses in Γ .

Example (Syllogism)

$$\frac{\text{Socrates is a man} \quad \text{All men are mortal}}{\text{Socrates is a mortal}}$$

Two main kinds of deduction rules:

Introduction rules (Rule 1) for the connectives

Elimination rules (Rule 2) for the connectives

Natural Deduction

Natural Deduction Rules

$$\frac{\Gamma \vdash P_1 \quad \Gamma \vdash P_2 \cdots \Gamma \vdash P_n}{\Gamma \vdash Q}$$

- **Meaning:** a proof of the **conclusion** Q follows from **premises** $P_1 \cdots P_n$.
- $\Gamma \vdash Q$ is a **sequent**: valid when Q holds subject to the hypotheses in Γ .

Example (Syllogism)

$$\frac{\text{Socrates is a man} \quad \text{All men are mortal}}{\text{Socrates is a mortal}}$$

Two main kinds of deduction rules:

Introduction rules

Elimination rules

Natural Deduction

Natural Deduction Rules

$$\frac{\Gamma \vdash P_1 \quad \Gamma \vdash P_2 \cdots \Gamma \vdash P_n}{\Gamma \vdash Q}$$

- **Meaning:** a proof of the **conclusion** Q follows from **premises** $P_1 \cdots P_n$.
- $\Gamma \vdash Q$ is a **sequent**: valid when Q holds subject to the hypotheses in Γ .

Example (Syllogism)

$$\frac{\text{Socrates is a man} \quad \text{All men are mortal}}{\text{Socrates is a mortal}}$$

Two main kinds of deduction rules:

Implication Introduction Rule

Implication Elimination Rule

Natural Deduction

Natural Deduction Rules

$$\frac{\Gamma \vdash P_1 \quad \Gamma \vdash P_2 \cdots \Gamma \vdash P_n}{\Gamma \vdash Q}$$

- **Meaning:** a proof of the **conclusion** Q follows from **premises** $P_1 \cdots P_n$.
- $\Gamma \vdash Q$ is a **sequent**: valid when Q holds subject to the hypotheses in Γ .

Example (Syllogism)

$$\frac{\text{Socrates is a man} \quad \text{All men are mortal}}{\text{Socrates is a mortal}}$$

Two main kinds of deduction rules:

Natural Deduction

Natural Deduction Rules

$$\frac{\Gamma \vdash P_1 \quad \Gamma \vdash P_2 \cdots \Gamma \vdash P_n}{\Gamma \vdash Q}$$

- **Meaning:** a proof of the **conclusion** Q follows from **premises** $P_1 \cdots P_n$.
- $\Gamma \vdash Q$ is a **sequent**: valid when Q holds subject to the hypotheses in Γ .

Example (Syllogism)

$$\frac{\text{Socrates is a man} \quad \text{All men are mortal}}{\text{Socrates is a mortal}}$$

Two main kinds of deduction rules:

Natural Deduction

Natural Deduction Rules

$$\frac{\Gamma \vdash P_1 \quad \Gamma \vdash P_2 \cdots \Gamma \vdash P_n}{\Gamma \vdash Q}$$

- **Meaning:** a proof of the **conclusion** Q follows from **premises** $P_1 \cdots P_n$.
- $\Gamma \vdash Q$ is a **sequent**: valid when Q holds subject to the hypotheses in Γ .

Example (Syllogism)

$$\frac{\text{Socrates is a man} \quad \text{All men are mortal}}{\text{Socrates is a mortal}}$$

Two main kinds of deduction rules:

- ➊ Introduction rules (**rule tactic**): backwards reasoning.
- ➋ Elimination rules (**erule tactic**): forwards reasoning.

Natural Deduction

Natural Deduction Rules

$$\frac{\Gamma \vdash P_1 \quad \Gamma \vdash P_2 \cdots \Gamma \vdash P_n}{\Gamma \vdash Q}$$

- **Meaning**: a proof of the **conclusion** Q follows from **premises** $P_1 \cdots P_n$.
- $\Gamma \vdash Q$ is a **sequent**: valid when Q holds subject to the hypotheses in Γ .

Example (Syllogism)

$$\frac{\text{Socrates is a man} \quad \text{All men are mortal}}{\text{Socrates is a mortal}}$$

Two main kinds of deduction rules:

- 1 Introduction rules (**rule** tactic): backwards reasoning.
- 2 Elimination rules (**erule** tactic): forwards reasoning.

Introduction Rules

- Question: how do we prove the conclusion P ?
- By finding subgoals $P_1 \cdots P_n$, based on current subgoal's conclusion.
- Proof by introduction is often called **backwards reasoning**.

Introduction Rules

- Invoke introduction rules using the **rule** tactic, e.g. **apply**(rule conjI).
- Matches subgoal's conclusion, **copies** any hypotheses to new subgoals.
- Most rules are **safe** introduction rules, but disjI1 and disjI2 aren't.

Introduction Rules

- **Question:** how do we prove the conclusion P ?
- By finding subgoals $P_1 \cdots P_n$, based on current subgoal's conclusion.
- Proof by introduction is often called **backwards reasoning**.

Introduction Rules

- Invoke introduction rules using the **rule** tactic, e.g. **apply**(rule conjI).
- Matches subgoal's conclusion, **copies** any hypotheses to new subgoals.
- Most rules are **safe** introduction rules, but disjI1 and disjI2 aren't.

Introduction Rules

- **Question:** how do we prove the conclusion P ?
- By finding subgoals $P_1 \cdots P_n$, based on current subgoal's conclusion.
- Proof by introduction is often called **backwards reasoning**.

Introduction Rules

- Invoke introduction rules using the **rule** tactic, e.g. **apply**(rule conjI).
- Matches subgoal's conclusion, **copies** any hypotheses to new subgoals.
- Most rules are **safe** introduction rules, but disjI1 and disjI2 aren't.

Introduction Rules

- **Question:** how do we prove the conclusion P ?
- By finding subgoals $P_1 \cdots P_n$, based on current subgoal's conclusion.
- **Proof by introduction** is often called **backwards reasoning**.

Introduction Rules



- Invoke introduction rules using the **rule** tactic, e.g. **apply**(rule conjI).
- Matches subgoal's conclusion, **copies** any hypotheses to new subgoals.
- Most rules are **safe** introduction rules, but disjI1 and disjI2 aren't.

Introduction Rules

- **Question:** how do we prove the conclusion P ?
- By finding subgoals $P_1 \cdots P_n$, based on current subgoal's conclusion.
- **Proof by introduction** is often called **backwards reasoning**.

Introduction Rules

$$\frac{\Gamma \vdash P \quad \Gamma \vdash Q}{\Gamma \vdash P \wedge Q} \text{ conjI}$$

$$\frac{-}{\Gamma \vdash \text{True}} \text{ TrueI}$$

$$\frac{\Gamma \vdash P}{\Gamma \vdash P \vee Q} \text{ disjI1}$$

$$\frac{\Gamma \vdash Q}{\Gamma \vdash P \vee Q} \text{ disjI2}$$

- Invoke introduction rules using the **rule** tactic, e.g. **apply**(rule conjI).
- Matches subgoal's conclusion, **copies** any hypotheses to new subgoals.
- Most rules are **safe** introduction rules, but disjI1 and disjI2 aren't.

Introduction Rules

- **Question:** how do we prove the conclusion P ?
- By finding subgoals $P_1 \cdots P_n$, based on current subgoal's conclusion.
- **Proof by introduction** is often called **backwards reasoning**.

Introduction Rules

$$\frac{\Gamma \vdash P \quad \Gamma \vdash Q}{\Gamma \vdash P \wedge Q} \text{ conjI}$$

$$\frac{-}{\Gamma \vdash \text{True}} \text{ TrueI}$$

$$\frac{\Gamma \vdash P}{\Gamma \vdash P \vee Q} \text{ disjI1}$$

$$\frac{\Gamma \vdash Q}{\Gamma \vdash P \vee Q} \text{ disjI2}$$

- Invoke introduction rules using the **rule** tactic, e.g. **apply**(rule conjI).
- Matches subgoal's conclusion, **copies** any hypotheses to new subgoals.
- Most rules are **safe** introduction rules, but disjI1 and disjI2 aren't.

Introduction Rules

- **Question:** how do we prove the conclusion P ?
- By finding subgoals $P_1 \cdots P_n$, based on current subgoal's conclusion.
- **Proof by introduction** is often called **backwards reasoning**.

Introduction Rules

$$\frac{\Gamma \vdash P \quad \Gamma \vdash Q}{\Gamma \vdash P \wedge Q} \text{ conjI} \qquad \frac{-}{\Gamma \vdash \text{True}} \text{ TrueI}$$

$$\frac{\Gamma \vdash P}{\Gamma \vdash P \vee Q} \text{ disjI1} \qquad \frac{\Gamma \vdash Q}{\Gamma \vdash P \vee Q} \text{ disjI2}$$

- Invoke introduction rules using the **rule** tactic, e.g. **apply**(rule conjI).
- Matches subgoal's conclusion, **copies** any hypotheses to new subgoals.
- Most rules are **safe** introduction rules, but disjI1 and disjI2 aren't.

Introduction Rules

- **Question:** how do we prove the conclusion P ?
- By finding subgoals $P_1 \cdots P_n$, based on current subgoal's conclusion.
- **Proof by introduction** is often called **backwards reasoning**.

Introduction Rules

$$\frac{\Gamma \vdash P \quad \Gamma \vdash Q}{\Gamma \vdash P \wedge Q} \text{ conjI}$$

$$\frac{-}{\Gamma \vdash \text{True}} \text{ TrueI}$$

$$\frac{\Gamma \vdash P}{\Gamma \vdash P \vee Q} \text{ disjI1}$$

$$\frac{\Gamma \vdash Q}{\Gamma \vdash P \vee Q} \text{ disjI2}$$

- Invoke introduction rules using the **rule** tactic, e.g. **apply**(rule conjI).
- Matches subgoal's conclusion, **copies** any hypotheses to new subgoals.
- Most rules are **safe** introduction rules, but disjI1 and disjI2 aren't.

Introduction Rules

- **Question:** how do we prove the conclusion P ?
- By finding subgoals $P_1 \cdots P_n$, based on current subgoal's conclusion.
- **Proof by introduction** is often called **backwards reasoning**.

Introduction Rules

$$\frac{\Gamma \vdash P \quad \Gamma \vdash Q}{\Gamma \vdash P \wedge Q} \text{ conjI}$$

$$\frac{-}{\Gamma \vdash \text{True}} \text{ TrueI}$$

$$\frac{\Gamma \vdash P}{\Gamma \vdash P \vee Q} \text{ disjI1}$$

$$\frac{\Gamma \vdash Q}{\Gamma \vdash P \vee Q} \text{ disjI2}$$

- Invoke introduction rules using the **rule** tactic, e.g. `apply(rule conjI)`.
- Matches subgoal's conclusion, **copies** any hypotheses to new subgoals.
- Most rules are **safe** introduction rules, but `disjI1` and `disjI2` aren't.

Introduction Rules

- **Question:** how do we prove the conclusion P ?
- By finding subgoals $P_1 \cdots P_n$, based on current subgoal's conclusion.
- **Proof by introduction** is often called **backwards reasoning**.

Introduction Rules

$$\frac{\Gamma \vdash P \quad \Gamma \vdash Q}{\Gamma \vdash P \wedge Q} \text{ conjI}$$

$$\frac{-}{\Gamma \vdash \text{True}} \text{ TrueI}$$

$$\frac{\Gamma \vdash P}{\Gamma \vdash P \vee Q} \text{ disjI1}$$

$$\frac{\Gamma \vdash Q}{\Gamma \vdash P \vee Q} \text{ disjI2}$$

- Invoke introduction rules using the **rule** tactic, e.g. **apply**(rule conjI).
- Matches subgoal's conclusion, **copies** any hypotheses to new subgoals.
- Most rules are **safe** introduction rules, but disjI1 and disjI2 aren't.

Introduction Rules

- **Question:** how do we prove the conclusion P ?
- By finding subgoals $P_1 \cdots P_n$, based on current subgoal's conclusion.
- **Proof by introduction** is often called **backwards reasoning**.

Introduction Rules

$$\frac{\Gamma \vdash P \quad \Gamma \vdash Q}{\Gamma \vdash P \wedge Q} \text{ conjI}$$

$$\frac{-}{\Gamma \vdash \text{True}} \text{ TrueI}$$

$$\frac{\Gamma \vdash P}{\Gamma \vdash P \vee Q} \text{ disjI1}$$

$$\frac{\Gamma \vdash Q}{\Gamma \vdash P \vee Q} \text{ disjI2}$$

- Invoke introduction rules using the **rule** tactic, e.g. **apply**(rule conjI).
- Matches subgoal's conclusion, **copies** any hypotheses to new subgoals.
- Most rules are **safe** introduction rules, but disjI1 and disjI2 aren't.

Introduction Rules

- **Question:** how do we prove the conclusion P ?
- By finding subgoals $P_1 \cdots P_n$, based on current subgoal's conclusion.
- **Proof by introduction** is often called **backwards reasoning**.

Introduction Rules

$$\frac{\Gamma \vdash P \quad \Gamma \vdash Q}{\Gamma \vdash P \wedge Q} \text{ conjI}$$

$$\frac{-}{\Gamma \vdash \text{True}} \text{ TrueI}$$

$$\frac{\Gamma \vdash P}{\Gamma \vdash P \vee Q} \text{ disjI1}$$

$$\frac{\Gamma \vdash Q}{\Gamma \vdash P \vee Q} \text{ disjI2}$$

- Invoke introduction rules using the **rule** tactic, e.g. **apply**(rule conjI).
- Matches subgoal's conclusion, **copies** any hypotheses to new subgoals.
- Most rules are **safe** introduction rules, but disjI1 and disjI2 aren't.

Example: Proof by Introduction

Natural Deduction Proof

```
proof
  assume P
  conjI
  disjI1
```

Proof Script

```
lemma "P  $\implies$  (P  $\wedge$  (P  $\vee$  Q))"
  apply (rule conjI)
  (* Subgoal 1 *)
  apply assumption
  (* Subgoal 2 *)
  apply (rule disjI1)
  apply assumption
done
```

Example: Proof by Introduction

Natural Deduction Proof

$$\frac{\frac{\text{asm } P}{\text{conjI } P} \quad \frac{\text{asm } P}{\text{disjI1 } P}}{P \vdash P \wedge (P \vee Q)}$$

Proof Script

```
lemma "P ==> (P ^ (P v Q))"  
  apply (rule conjI)  
  (* Subgoal 1 *)  
  apply assumption  
  (* Subgoal 2 *)  
  apply (rule disjI1)  
  apply assumption  
done
```

Example: Proof by Introduction

Natural Deduction Proof

$P \vdash P \wedge (P \vee Q)$

Proof Script

```
lemma "P ==> (P ^ (P v Q))"  
  apply (rule conjI)  
  (* Subgoal 1 *)  
  apply assumption  
  (* Subgoal 2 *)  
  apply (rule disjI1)  
  apply assumption  
  done
```

Example: Proof by Introduction

Natural Deduction Proof

$$\frac{\text{asm } P \quad \text{disjI1 } P \vee Q}{P \vdash P \wedge (P \vee Q)} \text{conjI}$$

Proof Script

```
lemma "P ==> (P ^ (P v Q))"  
  apply (rule conjI)  
  (* Subgoal 1 *)  
  apply assumption  
  (* Subgoal 2 *)  
  apply (rule disjI1)  
  apply assumption  
  done
```

Example: Proof by Introduction

Natural Deduction Proof

$$\frac{P \vdash P \quad P \vdash P \vee Q}{P \vdash P \wedge (P \vee Q)} \text{conjI}$$

Proof Script

```
lemma "P ==> (P ^ (P v Q))"  
  apply (rule conjI)  
  (* Subgoal 1 *)  
  apply assumption  
  (* Subgoal 2 *)  
  apply (rule disjI1)  
  apply assumption  
  done
```

Example: Proof by Introduction

Natural Deduction Proof

$$\frac{\frac{}{P \vdash P} \text{asm} \quad \frac{\frac{}{P \vdash P} \text{asm} \quad \frac{}{P \vdash P \vee Q} \text{disjI1}}{P \vdash P \wedge (P \vee Q)} \text{conjI}}{P \vdash P \wedge (P \vee Q)} \text{conjI}$$

Proof Script

```
lemma "P ==> (P ^ (P v Q))"
  apply (rule conjI)
  (* Subgoal 1 *)
  apply assumption
  (* Subgoal 2 *)
  apply (rule disjI1)
  apply assumption
  done
```

Example: Proof by Introduction

Natural Deduction Proof

$$\frac{\frac{}{P \vdash P} \text{asm} \quad \frac{\frac{}{P \vdash P} \text{asm} \quad \frac{}{P \vdash P \vee Q} \text{disjI1}}{P \vdash P \wedge (P \vee Q)} \text{conjI}}{P \vdash P \wedge (P \vee Q)} \text{conjI}}$$

Proof Script

```
lemma "P ==> (P ^ (P v Q))"  
  apply (rule conjI)  
  (* Subgoal 1 *)  
  apply assumption  
  (* Subgoal 2 *)  
  apply (rule disjI1)  
  apply assumption  
  done
```

Example: Proof by Introduction

Natural Deduction Proof

$$\frac{\frac{-}{P \vdash P} \text{asm} \quad \frac{\frac{\text{asm}}{P \vdash P} \text{disjI1}}{P \vdash P \vee Q} \text{conjI}}{P \vdash P \wedge (P \vee Q)} \text{conjI}$$

Proof Script

```
lemma "P ==> (P ^ (P v Q))"  
  apply (rule conjI)  
  (* Subgoal 1 *)  
  apply assumption  
  (* Subgoal 2 *)  
  apply (rule disjI1)  
  apply assumption  
  done
```

Example: Proof by Introduction

Natural Deduction Proof

$$\frac{\frac{-}{P \vdash P} \text{asm} \quad \frac{\frac{\text{asm}}{P \vdash P} \text{disjI1}}{P \vdash P \vee Q} \text{conjI}}{P \vdash P \wedge (P \vee Q)} \text{conjI}$$

Proof Script

```
lemma "P ==> (P ^ (P v Q))"  
  apply (rule conjI)  
  (* Subgoal 1 *)  
  apply assumption  
  (* Subgoal 2 *)  
  apply (rule disjI1)  
  apply assumption  
  done
```

Example: Proof by Introduction

Natural Deduction Proof

$$\frac{\frac{-}{P \vdash P} \text{asm} \quad \frac{\text{asm} \quad P \vdash P}{P \vdash P \vee Q} \text{disjI1}}{P \vdash P \wedge (P \vee Q)} \text{conjI}$$

Proof Script

```
lemma "P ==> (P ^ (P v Q))"  
  apply (rule conjI)  
  (* Subgoal 1 *)  
  apply assumption  
  (* Subgoal 2 *)  
  apply (rule disjI1)  
  apply assumption  
  done
```

Example: Proof by Introduction

Natural Deduction Proof

$$\frac{\frac{-}{P \vdash P} \text{asm} \quad \frac{P \vdash P}{P \vdash P \vee Q} \text{disjI1}}{P \vdash P \wedge (P \vee Q)} \text{conjI}$$

Proof Script

```
lemma "P ==> (P ^ (P v Q))"  
  apply (rule conjI)  
  (* Subgoal 1 *)  
  apply assumption  
  (* Subgoal 2 *)  
  apply (rule disjI1)  
  apply assumption  
  done
```

Example: Proof by Introduction

Natural Deduction Proof

$$\frac{\frac{-}{P \vdash P} \text{asm} \quad \frac{\frac{-}{P \vdash P} \text{asm}}{P \vdash P \vee Q} \text{disjI1}}{P \vdash P \wedge (P \vee Q)} \text{conjI}$$

Proof Script

```
lemma "P ==> (P ^ (P v Q))"  
  apply (rule conjI)  
  (* Subgoal 1 *)  
  apply assumption  
  (* Subgoal 2 *)  
  apply (rule disjI1)  
  apply assumption  
done
```

Example: Proof by Introduction

Natural Deduction Proof

$$\frac{\frac{-}{P \vdash P} \text{asm} \quad \frac{\frac{-}{P \vdash P} \text{asm}}{P \vdash P \vee Q} \text{disjI1}}{P \vdash P \wedge (P \vee Q)} \text{conjI}$$

Proof Script

```
lemma "P ==> (P ^ (P v Q))"  
  apply (rule conjI)  
  (* Subgoal 1 *)  
  apply assumption  
  (* Subgoal 2 *)  
  apply (rule disjI1)  
  apply assumption  
done
```

Example: Proof by Introduction

Natural Deduction Proof

$$\frac{\frac{-}{P \vdash P} \text{asm} \quad \frac{\frac{-}{P \vdash P} \text{asm}}{P \vdash P \vee Q} \text{disjI1}}{P \vdash P \wedge (P \vee Q)} \text{conjI}$$

Proof Script

```
lemma "P ==> (P ^ (P v Q))"  
  apply (rule conjI)  
  (* Subgoal 1 *)  
  apply assumption  
  (* Subgoal 2 *)  
  apply (rule disjI1)  
  apply assumption  
done
```

Example: Proof by Introduction (Again)

Proof State

```
lemma "P  $\implies$  (P  $\wedge$  (P  $\vee$  Q))"  
  apply (rule conjI)  
  (* Subgoal 1 *)  
  apply assumption  
  (* Subgoal 2 *)  
  apply (rule disjI1)  
  apply assumption  
done
```

Example: Proof by Introduction (Again)

Proof State

```
lemma "P  $\implies$  (P  $\wedge$  (P  $\vee$  Q))"  
  apply (rule conjI)  
  (* Subgoal 1 *)  
  apply assumption  
  (* Subgoal 2 *)  
  apply (rule disjI1)  
  apply assumption  
done
```

Example: Proof by Introduction (Again)

Proof State

1 $P \implies (P \wedge (P \vee Q)).$

lemma "P \implies (P \wedge (P \vee Q))"

```
apply (rule conjI)
(* Subgoal 1 *)
apply assumption
(* Subgoal 2 *)
apply (rule disjI1)
apply assumption
done
```

Example: Proof by Introduction (Again)

Proof State

① $P \implies P.$

② $P \implies P \vee Q.$

```
lemma "P  $\implies$  (P  $\wedge$  (P  $\vee$  Q))"
```

```
  apply (rule conjI)
```

```
    (* Subgoal 1 *)
```

```
    apply assumption
```

```
    (* Subgoal 2 *)
```

```
    apply (rule disjI1)
```

```
    apply assumption
```

```
  done
```

Example: Proof by Introduction (Again)

Proof State

① $P \implies P.$

② $P \implies P \vee Q.$

```
lemma "P  $\implies$  (P  $\wedge$  (P  $\vee$  Q))"
```

```
  apply (rule conjI)
```

```
  (* Subgoal 1 *)
```

```
  apply assumption
```

```
  (* Subgoal 2 *)
```

```
  apply (rule disjI1)
```

```
  apply assumption
```

```
done
```

Example: Proof by Introduction (Again)

Proof State

1 $P \implies P \vee Q.$

```
lemma "P  $\implies$  (P  $\wedge$  (P  $\vee$  Q))"
```

```
  apply (rule conjI)
```

```
  (* Subgoal 1 *)
```

```
  apply assumption
```

```
  (* Subgoal 2 *)
```

```
  apply (rule disjI1)
```

```
  apply assumption
```

```
done
```

Example: Proof by Introduction (Again)

Proof State

1 $P \implies P \vee Q.$

```
lemma "P  $\implies$  (P  $\wedge$  (P  $\vee$  Q))"
```

```
  apply (rule conjI)
```

```
  (* Subgoal 1 *)
```

```
  apply assumption
```

```
  (* Subgoal 2 *)
```

```
  apply (rule disjI1)
```

```
  apply assumption
```

```
done
```

Example: Proof by Introduction (Again)

Proof State

1 $P \implies P.$

```
lemma "P  $\implies$  (P  $\wedge$  (P  $\vee$  Q))"
```

```
  apply (rule conjI)
```

```
  (* Subgoal 1 *)
```

```
  apply assumption
```

```
  (* Subgoal 2 *)
```

```
  apply (rule disjI1)
```

```
  apply assumption
```

```
done
```

Example: Proof by Introduction (Again)

Proof State

No subgoals!

```
lemma "P  $\implies$  (P  $\wedge$  (P  $\vee$  Q))"
```

```
  apply (rule conjI)
```

```
  (* Subgoal 1 *)
```

```
  apply assumption
```

```
  (* Subgoal 2 *)
```

```
  apply (rule disjI1)
```

```
  apply assumption
```

```
done
```

Example: Proof by Introduction (Again)

Proof State

```
lemma "P  $\implies$  (P  $\wedge$  (P  $\vee$  Q))"  
  apply (rule conjI)  
  (* Subgoal 1 *)  
  apply assumption  
  (* Subgoal 2 *)  
  apply (rule disjI1)  
  apply assumption  
done
```

More Introduction Rules

Introduction Rules

```
lemma "P  $\longrightarrow$  (P  $\wedge$  (True = True))"  
apply (rule impI) (* P  $\implies$  (P  $\wedge$  (True = True)) *)  
apply (rule conjI) (* P  $\implies$  P and P  $\implies$  (True = True)) *)  
apply (assumption) (* P  $\implies$  (True = True)) *)  
apply (rule refl)  
done
```

More Introduction Rules

Introduction Rules

$$\frac{P, \Gamma \vdash Q}{\Gamma \vdash P \longrightarrow Q} \text{ impI} \quad \frac{\Gamma \vdash P \longrightarrow Q \quad \Gamma \vdash Q \longrightarrow P}{\Gamma \vdash P \longleftrightarrow Q} \text{ iffI} \quad \frac{-}{\Gamma \vdash t = t} \text{ refl}$$

```
lemma "P  $\longrightarrow$  (P  $\wedge$  (True = True))"
```

```
apply (rule impI) (* P  $\implies$  (P  $\wedge$  (True = True)) *)
```

```
apply (rule conjI) (* P  $\implies$  P and P  $\implies$  (True = True) *)
```

```
apply (assumption) (* P  $\implies$  (True = True) *)
```

```
apply (rule refl)
```

```
done
```

More Introduction Rules

Introduction Rules

$$\frac{P, \Gamma \vdash Q}{\Gamma \vdash P \longrightarrow Q} \text{impI} \quad \frac{\Gamma \vdash P \longrightarrow Q \quad \Gamma \vdash Q \longrightarrow P}{\Gamma \vdash P \longleftrightarrow Q} \text{iffI} \quad \frac{-}{\Gamma \vdash t = t} \text{refl}$$

```
lemma "P  $\longrightarrow$  (P  $\wedge$  (True = True))"
```

```
apply (rule impI) (* P  $\implies$  (P  $\wedge$  (True = True)) *)
```

```
apply (rule conjI) (* P  $\implies$  P and P  $\implies$  (True = True) *)
```

```
apply (assumption) (* P  $\implies$  (True = True) *)
```

```
apply (rule refl)
```

```
done
```

More Introduction Rules

Introduction Rules

$$\frac{P, \Gamma \vdash Q}{\Gamma \vdash P \rightarrow Q} \text{impI} \quad \frac{\Gamma \vdash P \rightarrow Q \quad \Gamma \vdash Q \rightarrow P}{\Gamma \vdash P \leftrightarrow Q} \text{iffI} \quad \frac{-}{\Gamma \vdash t = t} \text{refl}$$

```
lemma "P  $\rightarrow$  (P  $\wedge$  (True = True))"  
apply (rule impI) (* P  $\implies$  (P  $\wedge$  (True = True)) *)  
apply (rule conjI) (* P  $\implies$  P and P  $\implies$  (True = True)) *)  
apply (assumption) (* P  $\implies$  (True = True)) *)  
apply (rule refl)  
done
```

More Introduction Rules

Introduction Rules

$$\frac{P, \Gamma \vdash Q}{\Gamma \vdash P \rightarrow Q} \text{impI} \quad \frac{\Gamma \vdash P \rightarrow Q \quad \Gamma \vdash Q \rightarrow P}{\Gamma \vdash P \leftrightarrow Q} \text{iffI} \quad \frac{-}{\Gamma \vdash t = t} \text{refl}$$

```
lemma "P  $\rightarrow$  (P  $\wedge$  (True = True))"
```

```
apply (rule impI) (* P  $\Rightarrow$  (P  $\wedge$  (True = True)) *)
```

```
apply (rule conjI) (* P  $\Rightarrow$  P and P  $\Rightarrow$  (True = True) *)
```

```
apply (assumption) (* P  $\Rightarrow$  (True = True) *)
```

```
apply (rule refl)
```

```
done
```

More Introduction Rules

Introduction Rules

$$\frac{P, \Gamma \vdash Q}{\Gamma \vdash P \rightarrow Q} \text{impI} \quad \frac{\Gamma \vdash P \rightarrow Q \quad \Gamma \vdash Q \rightarrow P}{\Gamma \vdash P \leftrightarrow Q} \text{iffI} \quad \frac{-}{\Gamma \vdash t = t} \text{refl}$$

lemma "P \rightarrow (P \wedge (True = True))"

```
apply (rule impI) (* P  $\Rightarrow$  (P  $\wedge$  (True = True)) *)
apply (rule conjI) (* P  $\Rightarrow$  P and P  $\Rightarrow$  (True = True) *)
apply (assumption) (* P  $\Rightarrow$  (True = True) *)
apply (rule refl)
done
```

More Introduction Rules

Introduction Rules

$$\frac{P, \Gamma \vdash Q}{\Gamma \vdash P \longrightarrow Q} \text{impI} \quad \frac{\Gamma \vdash P \longrightarrow Q \quad \Gamma \vdash Q \longrightarrow P}{\Gamma \vdash P \longleftrightarrow Q} \text{iffI} \quad \frac{-}{\Gamma \vdash t = t} \text{refl}$$

lemma "P \longrightarrow (P \wedge (True = True))"

```
apply (rule impI) (* P  $\implies$  (P  $\wedge$  (True = True)) *)  
apply (rule conjI) (* P  $\implies$  P and P  $\implies$  (True = True)) *)  
apply (assumption) (* P  $\implies$  (True = True)) *)  
apply (rule refl)  
done
```

More Introduction Rules

Introduction Rules

$$\frac{P, \Gamma \vdash Q}{\Gamma \vdash P \rightarrow Q} \text{impI} \quad \frac{\Gamma \vdash P \rightarrow Q \quad \Gamma \vdash Q \rightarrow P}{\Gamma \vdash P \leftrightarrow Q} \text{iffI} \quad \frac{-}{\Gamma \vdash t = t} \text{refl}$$

lemma "P \rightarrow (P \wedge (True = True))"

apply (rule impI) (* P \implies (P \wedge (True = True)) *)

apply (rule conjI) (* P \implies P and P \implies (True = True)) *)

apply (assumption) (* P \implies (True = True)) *)

apply (rule refl)

done

More Introduction Rules

Introduction Rules

$$\frac{P, \Gamma \vdash Q}{\Gamma \vdash P \longrightarrow Q} \text{ impI} \quad \frac{\Gamma \vdash P \longrightarrow Q \quad \Gamma \vdash Q \longrightarrow P}{\Gamma \vdash P \longleftrightarrow Q} \text{ iffI} \quad \frac{-}{\Gamma \vdash t = t} \text{ refl}$$

lemma "P \longrightarrow (P \wedge (True = True))"

apply (rule impI) (* P \implies (P \wedge (True = True)) *)

apply (rule conjI) (* P \implies P and P \implies (True = True) *)

apply (assumption) (* P \implies (True = True) *)

apply (rule refl)

done

More Introduction Rules

Introduction Rules

$$\frac{P, \Gamma \vdash Q}{\Gamma \vdash P \rightarrow Q} \text{impI} \quad \frac{\Gamma \vdash P \rightarrow Q \quad \Gamma \vdash Q \rightarrow P}{\Gamma \vdash P \leftrightarrow Q} \text{iffI} \quad \frac{-}{\Gamma \vdash t = t} \text{refl}$$

lemma "P \rightarrow (P \wedge (True = True))"

apply (rule impI) (* P \implies (P \wedge (True = True)) *)

apply (rule conjI) (* P \implies P and P \implies (True = True) *)

apply (assumption) (* P \implies (True = True) *)

apply (rule refl)

done

More Introduction Rules

Introduction Rules

$$\frac{P, \Gamma \vdash Q}{\Gamma \vdash P \rightarrow Q} \text{impI} \quad \frac{\Gamma \vdash P \rightarrow Q \quad \Gamma \vdash Q \rightarrow P}{\Gamma \vdash P \leftrightarrow Q} \text{iffI} \quad \frac{-}{\Gamma \vdash t = t} \text{refl}$$

lemma "P \rightarrow (P \wedge (True = True))"

apply (rule impI) (* P \implies (P \wedge (True = True)) *)

apply (rule conjI) (* P \implies P and P \implies (True = True) *)

apply (assumption) (* P \implies (True = True) *)

apply (rule refl)

done

More Introduction Rules

Introduction Rules

$$\frac{P, \Gamma \vdash Q}{\Gamma \vdash P \rightarrow Q} \text{impI} \quad \frac{\Gamma \vdash P \rightarrow Q \quad \Gamma \vdash Q \rightarrow P}{\Gamma \vdash P \leftrightarrow Q} \text{iffI} \quad \frac{-}{\Gamma \vdash t = t} \text{refl}$$

lemma "P \rightarrow (P \wedge (True = True))"

apply (rule impI) (* P \implies (P \wedge (True = True)) *)

apply (rule conjI) (* P \implies P and P \implies (True = True) *)

apply (assumption) (* P \implies (True = True) *)

apply (rule refl)

done

More Introduction Rules

Introduction Rules

$$\frac{P, \Gamma \vdash Q}{\Gamma \vdash P \rightarrow Q} \text{impI} \quad \frac{\Gamma \vdash P \rightarrow Q \quad \Gamma \vdash Q \rightarrow P}{\Gamma \vdash P \leftrightarrow Q} \text{iffI} \quad \frac{-}{\Gamma \vdash t = t} \text{refl}$$

lemma "P \rightarrow (P \wedge (True = True))"

apply (rule impI) (* P \implies (P \wedge (True = True)) *)

apply (rule conjI) (* P \implies P and P \implies (True = True) *)

apply (assumption) (* P \implies (True = True) *)

apply (rule refl)

done

More Introduction Rules

Introduction Rules

$$\frac{P, \Gamma \vdash Q}{\Gamma \vdash P \longrightarrow Q} \text{ impI} \quad \frac{\Gamma \vdash P \longrightarrow Q \quad \Gamma \vdash Q \longrightarrow P}{\Gamma \vdash P \longleftrightarrow Q} \text{ iffI} \quad \frac{-}{\Gamma \vdash t = t} \text{ refl}$$

lemma "P \longrightarrow (P \wedge (True = True))"

apply (rule impI) (* P \implies (P \wedge (True = True)) *)

apply (rule conjI) (* P \implies P and P \implies (True = True) *)

apply (assumption) (* P \implies (True = True) *)

apply (rule refl)

done

Elimination Rules

What can we deduce from hypotheses P ? Return a subgoal.

Elimination Rules for Propositional Calculus

Elimination rules use `erule` instead of `apply` (erule/ass/E).

Find first hypothesis matching first premise of elimination rule.

Replace subgoal with remaining premises: $[P \wedge Q] \Rightarrow B$.

Elimination Rules

- What can we deduce from hypothesis P_i ? **Refine** a subgoal.

Elimination Rules for Propositional Calculus

- Elimination rules use **erule** tactical **apply** (`erule conjE`).
- Find **first hypothesis** matching first premise of elimination rule.
- Replace subgoal with remaining premises: $\llbracket P; Q \rrbracket \implies R$.

Elimination Rules

- What can we deduce from hypothesis P_i ? **Refine** a subgoal.

Elimination Rules for Propositional Calculus

$$\frac{P, Q, \Gamma \vdash R}{P \wedge Q, \Gamma \vdash R} \text{ conjE}$$

$$\frac{P, \Gamma \vdash R \quad Q, \Gamma \vdash R}{P \vee Q, \Gamma \vdash R} \text{ disjE}$$

$$\frac{\Gamma \vdash P \quad Q, \Gamma \vdash R}{P \rightarrow Q, \Gamma \vdash R} \text{ impE}$$

$$\frac{P \rightarrow Q, Q \rightarrow P, \Gamma \vdash R}{P \leftrightarrow Q, \Gamma \vdash R} \text{ iffE}$$

$$\frac{P \wedge Q \quad \llbracket P; Q \rrbracket \Rightarrow R}{R} \text{ conjE}$$

$$\frac{P \vee Q \quad P \Rightarrow R \quad Q \Rightarrow R}{R} \text{ disjE}$$

$$\frac{P \rightarrow Q \quad P \quad Q \Rightarrow R}{R} \text{ impE}$$

$$\frac{P \leftrightarrow Q \quad \llbracket P \rightarrow Q; Q \rightarrow P \rrbracket \Rightarrow R}{R} \text{ iffE}$$

- Elimination rules use **erule** tactical **apply** (erule conjE).
- Find **first hypothesis** matching first premise of elimination rule.
- Replace subgoal with remaining premises: $\llbracket P; Q \rrbracket \Rightarrow R$.

Elimination Rules

- What can we deduce from hypothesis P_i ? **Refine** a subgoal.

Elimination Rules for Propositional Calculus

$$\frac{P, Q, \Gamma \vdash R}{P \wedge Q, \Gamma \vdash R} \text{ conjE}$$

$$\frac{P, \Gamma \vdash R \quad Q, \Gamma \vdash R}{P \vee Q, \Gamma \vdash R} \text{ disjE}$$

$$\frac{\Gamma \vdash P \quad Q, \Gamma \vdash R}{P \rightarrow Q, \Gamma \vdash R} \text{ impE}$$

$$\frac{P \rightarrow Q, Q \rightarrow P, \Gamma \vdash R}{P \leftrightarrow Q, \Gamma \vdash R} \text{ iffE}$$

$$\frac{P \wedge Q \quad [P; Q] \Rightarrow R}{R} \text{ conjE}$$

$$\frac{P \vee Q \quad P \Rightarrow R \quad Q \Rightarrow R}{R} \text{ disjE}$$

$$\frac{P \rightarrow Q \quad P \quad Q \Rightarrow R}{R} \text{ impE}$$

$$\frac{P \leftrightarrow Q \quad [P \rightarrow Q; Q \rightarrow P] \Rightarrow R}{R} \text{ iffE}$$

- Elimination rules use **erule** tactical **apply** (erule conjE).
- Find **first hypothesis** matching first premise of elimination rule.
- Replace subgoal with remaining premises: $[P; Q] \Rightarrow R$.

Elimination Rules

- What can we deduce from hypothesis P_i ? **Refine** a subgoal.

Elimination Rules for Propositional Calculus

$$\frac{P, Q, \Gamma \vdash R}{P \wedge Q, \Gamma \vdash R} \text{ conjE}$$

$$\frac{P, \Gamma \vdash R \quad Q, \Gamma \vdash R}{P \vee Q, \Gamma \vdash R} \text{ disjE}$$

$$\frac{\Gamma \vdash P \quad Q, \Gamma \vdash R}{P \rightarrow Q, \Gamma \vdash R} \text{ impE}$$

$$\frac{P \rightarrow Q, Q \rightarrow P, \Gamma \vdash R}{P \leftrightarrow Q, \Gamma \vdash R} \text{ iffE}$$

$$\frac{P \wedge Q \quad [P; Q] \Rightarrow R}{R} \text{ conjE}$$

$$\frac{P \vee Q \quad P \Rightarrow R \quad Q \Rightarrow R}{R} \text{ disjE}$$

$$\frac{P \rightarrow Q \quad P \quad Q \Rightarrow R}{R} \text{ impE}$$

$$\frac{P \leftrightarrow Q \quad [P \rightarrow Q; Q \rightarrow P] \Rightarrow R}{R} \text{ iffE}$$

- Elimination rules use **erule** tactical **apply** (erule conjE).
- Find **first hypothesis** matching first premise of elimination rule.
- Replace subgoal with remaining premises: $[P; Q] \Rightarrow R$.

Elimination Rules

- What can we deduce from hypothesis P_i ? **Refine** a subgoal.

Elimination Rules for Propositional Calculus

$$\frac{P, Q, \Gamma \vdash R}{P \wedge Q, \Gamma \vdash R} \text{ conjE}$$

$$\frac{P, \Gamma \vdash R \quad Q, \Gamma \vdash R}{P \vee Q, \Gamma \vdash R} \text{ disjE}$$

$$\frac{\Gamma \vdash P \quad Q, \Gamma \vdash R}{P \rightarrow Q, \Gamma \vdash R} \text{ impE}$$

$$\frac{P \rightarrow Q, Q \rightarrow P, \Gamma \vdash R}{P \leftrightarrow Q, \Gamma \vdash R} \text{ iffE}$$

$$\frac{P \wedge Q \quad [P; Q] \Rightarrow R}{R} \text{ conjE}$$

$$\frac{P \vee Q \quad P \Rightarrow R \quad Q \Rightarrow R}{R} \text{ disjE}$$

$$\frac{P \rightarrow Q \quad P \quad Q \Rightarrow R}{R} \text{ impE}$$

$$\frac{P \leftrightarrow Q \quad [P \rightarrow Q; Q \rightarrow P] \Rightarrow R}{R} \text{ iffE}$$

- Elimination rules use **erule** tactical **apply** (erule conjE).
- Find **first hypothesis** matching first premise of elimination rule.
- Replace subgoal with remaining premises: $[P; Q] \Rightarrow R$.

Elimination Rules

- What can we deduce from hypothesis P_i ? **Refine** a subgoal.

Elimination Rules for Propositional Calculus

$$\frac{P, Q, \Gamma \vdash R}{P \wedge Q, \Gamma \vdash R} \text{ conjE}$$

$$\frac{P, \Gamma \vdash R \quad Q, \Gamma \vdash R}{P \vee Q, \Gamma \vdash R} \text{ disjE}$$

$$\frac{\Gamma \vdash P \quad Q, \Gamma \vdash R}{P \rightarrow Q, \Gamma \vdash R} \text{ impE}$$

$$\frac{P \rightarrow Q, Q \rightarrow P, \Gamma \vdash R}{P \leftrightarrow Q, \Gamma \vdash R} \text{ iffE}$$

$$\frac{P \wedge Q \quad [P; Q] \Rightarrow R}{R} \text{ conjE}$$

$$\frac{P \vee Q \quad P \Rightarrow R \quad Q \Rightarrow R}{R} \text{ disjE}$$

$$\frac{P \rightarrow Q \quad P \quad Q \Rightarrow R}{R} \text{ impE}$$

$$\frac{P \leftrightarrow Q \quad [P \rightarrow Q; Q \rightarrow P] \Rightarrow R}{R} \text{ iffE}$$

- Elimination rules use **erule** tactical **apply** (erule conjE).
- Find **first hypothesis** matching first premise of elimination rule.
- Replace subgoal with remaining premises: $[P; Q] \Rightarrow R$.

Elimination Rules

- What can we deduce from hypothesis P_i ? **Refine** a subgoal.

Elimination Rules for Propositional Calculus

$$\frac{P, Q, \Gamma \vdash R}{P \wedge Q, \Gamma \vdash R} \text{ conjE}$$

$$\frac{P, \Gamma \vdash R \quad Q, \Gamma \vdash R}{P \vee Q, \Gamma \vdash R} \text{ disjE}$$

$$\frac{\Gamma \vdash P \quad Q, \Gamma \vdash R}{P \rightarrow Q, \Gamma \vdash R} \text{ impE}$$

$$\frac{P \rightarrow Q, Q \rightarrow P, \Gamma \vdash R}{P \leftrightarrow Q, \Gamma \vdash R} \text{ iffE}$$

$$\frac{P \wedge Q \quad \llbracket P; Q \rrbracket \Rightarrow R}{R} \text{ conjE}$$

$$\frac{P \vee Q \quad P \Rightarrow R \quad Q \Rightarrow R}{R} \text{ disjE}$$

$$\frac{P \rightarrow Q \quad P \quad Q \Rightarrow R}{R} \text{ impE}$$

$$\frac{P \leftrightarrow Q \quad \llbracket P \rightarrow Q; Q \rightarrow P \rrbracket \Rightarrow R}{R} \text{ iffE}$$

- Elimination rules use **erule** tactical **apply** (erule conjE).
- Find **first hypothesis** matching first premise of elimination rule.
- Replace subgoal with remaining premises: $\llbracket P; Q \rrbracket \Rightarrow R$.

Elimination Rules

- What can we deduce from hypothesis P_i ? **Refine** a subgoal.

Elimination Rules for Propositional Calculus

$$\frac{P, Q, \Gamma \vdash R}{P \wedge Q, \Gamma \vdash R} \text{ conjE}$$

$$\frac{P, \Gamma \vdash R \quad Q, \Gamma \vdash R}{P \vee Q, \Gamma \vdash R} \text{ disjE}$$

$$\frac{\Gamma \vdash P \quad Q, \Gamma \vdash R}{P \rightarrow Q, \Gamma \vdash R} \text{ impE}$$

$$\frac{P \rightarrow Q, Q \rightarrow P, \Gamma \vdash R}{P \leftrightarrow Q, \Gamma \vdash R} \text{ iffE}$$

$$\frac{P \wedge Q \quad \llbracket P; Q \rrbracket \Rightarrow R}{R} \text{ conjE}$$

$$\frac{P \vee Q \quad P \Rightarrow R \quad Q \Rightarrow R}{R} \text{ disjE}$$

$$\frac{P \rightarrow Q \quad P \quad Q \Rightarrow R}{R} \text{ impE}$$

$$\frac{P \leftrightarrow Q \quad \llbracket P \rightarrow Q; Q \rightarrow P \rrbracket \Rightarrow R}{R} \text{ iffE}$$

- Elimination rules use **erule** tactical **apply** (erule conjE).
- Find **first hypothesis** matching first premise of elimination rule.
- Replace subgoal with remaining premises: $\llbracket P; Q \rrbracket \Rightarrow R$.

Elimination Rules

- What can we deduce from hypothesis P_i ? **Refine** a subgoal.

Elimination Rules for Propositional Calculus

$$\frac{P, Q, \Gamma \vdash R}{P \wedge Q, \Gamma \vdash R} \text{ conjE}$$

$$\frac{P, \Gamma \vdash R \quad Q, \Gamma \vdash R}{P \vee Q, \Gamma \vdash R} \text{ disjE}$$

$$\frac{\Gamma \vdash P \quad Q, \Gamma \vdash R}{P \rightarrow Q, \Gamma \vdash R} \text{ impE}$$

$$\frac{P \rightarrow Q, Q \rightarrow P, \Gamma \vdash R}{P \leftrightarrow Q, \Gamma \vdash R} \text{ iffE}$$

$$\frac{P \wedge Q \quad \llbracket P; Q \rrbracket \Rightarrow R}{R} \text{ conjE}$$

$$\frac{P \vee Q \quad P \Rightarrow R \quad Q \Rightarrow R}{R} \text{ disjE}$$

$$\frac{P \rightarrow Q \quad P \quad Q \Rightarrow R}{R} \text{ impE}$$

$$\frac{P \leftrightarrow Q \quad \llbracket P \rightarrow Q; Q \rightarrow P \rrbracket \Rightarrow R}{R} \text{ iffE}$$

- Elimination rules use **erule** tactical **apply** (erule conjE).
- Find **first hypothesis** matching first premise of elimination rule.
- Replace subgoal with remaining premises: $\llbracket P; Q \rrbracket \Rightarrow R$.

Elimination Rules

- What can we deduce from hypothesis P_i ? **Refine** a subgoal.

Elimination Rules for Propositional Calculus

$$\frac{P, Q, \Gamma \vdash R}{P \wedge Q, \Gamma \vdash R} \text{ conjE}$$

$$\frac{P, \Gamma \vdash R \quad Q, \Gamma \vdash R}{P \vee Q, \Gamma \vdash R} \text{ disjE}$$

$$\frac{\Gamma \vdash P \quad Q, \Gamma \vdash R}{P \rightarrow Q, \Gamma \vdash R} \text{ impE}$$

$$\frac{P \rightarrow Q, Q \rightarrow P, \Gamma \vdash R}{P \leftrightarrow Q, \Gamma \vdash R} \text{ iffE}$$

$$\frac{P \wedge Q \quad \llbracket P; Q \rrbracket \Rightarrow R}{R} \text{ conjE}$$

$$\frac{P \vee Q \quad P \Rightarrow R \quad Q \Rightarrow R}{R} \text{ disjE}$$

$$\frac{P \rightarrow Q \quad P \quad Q \Rightarrow R}{R} \text{ impE}$$

$$\frac{P \leftrightarrow Q \quad \llbracket P \rightarrow Q; Q \rightarrow P \rrbracket \Rightarrow R}{R} \text{ iffE}$$

- Elimination rules use **erule** tactical **apply** (erule conjE).
- Find **first hypothesis** matching first premise of elimination rule.
- Replace subgoal with remaining premises: $\llbracket P; Q \rrbracket \Rightarrow R$.

Elimination Rules

- What can we deduce from hypothesis P_i ? **Refine** a subgoal.

Elimination Rules for Propositional Calculus

$$\frac{P, Q, \Gamma \vdash R}{P \wedge Q, \Gamma \vdash R} \text{ conjE}$$

$$\frac{P, \Gamma \vdash R \quad Q, \Gamma \vdash R}{P \vee Q, \Gamma \vdash R} \text{ disjE}$$

$$\frac{\Gamma \vdash P \quad Q, \Gamma \vdash R}{P \rightarrow Q, \Gamma \vdash R} \text{ impE}$$

$$\frac{P \rightarrow Q, Q \rightarrow P, \Gamma \vdash R}{P \leftrightarrow Q, \Gamma \vdash R} \text{ iffE}$$

$$\frac{P \wedge Q \quad \llbracket P; Q \rrbracket \Rightarrow R}{R} \text{ conjE}$$

$$\frac{P \vee Q \quad P \Rightarrow R \quad Q \Rightarrow R}{R} \text{ disjE}$$

$$\frac{P \rightarrow Q \quad P \quad Q \Rightarrow R}{R} \text{ impE}$$

$$\frac{P \leftrightarrow Q \quad \llbracket P \rightarrow Q; Q \rightarrow P \rrbracket \Rightarrow R}{R} \text{ iffE}$$

- Elimination rules use **erule** tactical **apply** (erule conjE).
- Find **first hypothesis** matching first premise of elimination rule.
- Replace subgoal with remaining premises: $\llbracket P; Q \rrbracket \Rightarrow R$.

Elimination Rules

- What can we deduce from hypothesis P_i ? **Refine** a subgoal.

Elimination Rules for Propositional Calculus

$$\frac{P, Q, \Gamma \vdash R}{P \wedge Q, \Gamma \vdash R} \text{ conjE}$$

$$\frac{P, \Gamma \vdash R \quad Q, \Gamma \vdash R}{P \vee Q, \Gamma \vdash R} \text{ disjE}$$

$$\frac{\Gamma \vdash P \quad Q, \Gamma \vdash R}{P \rightarrow Q, \Gamma \vdash R} \text{ impE}$$

$$\frac{P \rightarrow Q, Q \rightarrow P, \Gamma \vdash R}{P \leftrightarrow Q, \Gamma \vdash R} \text{ iffE}$$

$$\frac{P \wedge Q \quad \llbracket P; Q \rrbracket \Rightarrow R}{R} \text{ conjE}$$

$$\frac{P \vee Q \quad P \Rightarrow R \quad Q \Rightarrow R}{R} \text{ disjE}$$

$$\frac{P \rightarrow Q \quad P \quad Q \Rightarrow R}{R} \text{ impE}$$

$$\frac{P \leftrightarrow Q \quad \llbracket P \rightarrow Q; Q \rightarrow P \rrbracket \Rightarrow R}{R} \text{ iffE}$$

- Elimination rules use **erule** tactical **apply** (erule conjE).
- Find **first hypothesis** matching first premise of elimination rule.
- Replace subgoal with remaining premises: $\llbracket P; Q \rrbracket \Rightarrow R$.

Elimination Rules

- What can we deduce from hypothesis P_i ? **Refine** a subgoal.

Elimination Rules for Propositional Calculus

$$\frac{P, Q, \Gamma \vdash R}{P \wedge Q, \Gamma \vdash R} \text{ conjE}$$

$$\frac{P, \Gamma \vdash R \quad Q, \Gamma \vdash R}{P \vee Q, \Gamma \vdash R} \text{ disjE}$$

$$\frac{\Gamma \vdash P \quad Q, \Gamma \vdash R}{P \rightarrow Q, \Gamma \vdash R} \text{ impE}$$

$$\frac{P \rightarrow Q, Q \rightarrow P, \Gamma \vdash R}{P \leftrightarrow Q, \Gamma \vdash R} \text{ iffE}$$

$$\frac{P \wedge Q \quad \llbracket P; Q \rrbracket \Rightarrow R}{R} \text{ conjE}$$

$$\frac{P \vee Q \quad P \Rightarrow R \quad Q \Rightarrow R}{R} \text{ disjE}$$

$$\frac{P \rightarrow Q \quad P \quad Q \Rightarrow R}{R} \text{ impE}$$

$$\frac{P \leftrightarrow Q \quad \llbracket P \rightarrow Q; Q \rightarrow P \rrbracket \Rightarrow R}{R} \text{ iffE}$$

- Elimination rules use **erule** tactical **apply** (erule conjE).
- Find **first hypothesis** matching first premise of elimination rule.
- Replace subgoal with remaining premises: $\llbracket P; Q \rrbracket \Rightarrow R$.

Example: Proof by Elimination (1)

How do we prove $P \wedge (P \longrightarrow Q) \longrightarrow Q$?

Natural Deduction Proof

```
1. P ∧ (P → Q)
2. P
3. P → Q
4. Q
5. P ∧ (P → Q) → Q
```

Proof Script

```
lemma "P ∧ (P → Q) → Q"
  apply (rule impI)
  apply (erule conjE)
  apply (erule impE)
  (* Subgoal 1 *)
  apply assumption
  (* Subgoal 2 *)
  apply assumption
done
```

Example: Proof by Elimination (1)

How do we prove $P \wedge (P \longrightarrow Q) \longrightarrow Q$?

Natural Deduction Proof

Proof Script

```
lemma "P ∧ (P → Q) → Q"  
  apply (rule impI)  
  apply (erule conjE)  
  apply (erule impE)  
    (* Subgoal 1 *)  
    apply assumption  
  (* Subgoal 2 *)  
  apply assumption  
done
```

Example: Proof by Elimination (1)

How do we prove $P \wedge (P \rightarrow Q) \rightarrow Q$?

Natural Deduction Proof

$\frac{P \Rightarrow P}{\text{asm}} \quad \frac{[P; Q] \Rightarrow Q}{\text{asm}}$
 $\frac{[P \rightarrow Q]; P \Rightarrow Q}{\text{impE}}$
 $\frac{P \wedge (P \rightarrow Q) \Rightarrow Q}{\text{conjE}}$
 $\frac{P \wedge (P \rightarrow Q) \rightarrow Q}{\text{impI}}$

Proof Script

```
lemma "P ∧ (P → Q) → Q"  
  apply (rule impI)  
  apply (erule conjE)  
  apply (erule impE)  
    (* Subgoal 1 *)  
    apply assumption  
  (* Subgoal 2 *)  
  apply assumption  
done
```

Example: Proof by Elimination (1)

How do we prove $P \wedge (P \rightarrow Q) \rightarrow Q$?

Natural Deduction Proof

$$\frac{\frac{\frac{P \Rightarrow P}{\text{asm}} \quad \frac{[P; Q] \Rightarrow Q}{\text{asm}}}{(P \rightarrow Q); P \Rightarrow Q}{\text{conjE}} \quad P \wedge (P \rightarrow Q) \Rightarrow Q}{P \wedge (P \rightarrow Q) \rightarrow Q}{\text{impI}}$$

Proof Script

```
lemma "P ∧ (P → Q) → Q"
  apply (rule impI)
  apply (erule conjE)
  apply (erule impE)
  (* Subgoal 1 *)
  apply assumption
  (* Subgoal 2 *)
  apply assumption
done
```

Example: Proof by Elimination (1)

How do we prove $P \wedge (P \rightarrow Q) \rightarrow Q$?

Natural Deduction Proof

$$\frac{\begin{array}{l} \text{asm } P \Rightarrow P \quad \text{asm } [P; Q] \Rightarrow Q \\ \text{impE} \quad [(P \rightarrow Q); P] \Rightarrow Q \\ \text{conjE} \quad P \wedge (P \rightarrow Q) \Rightarrow Q \end{array}}{P \wedge (P \rightarrow Q) \rightarrow Q} \text{impI}$$

Proof Script

```
lemma "P ∧ (P → Q) → Q"
  apply (rule impI)
  apply (erule conjE)
  apply (erule impE)
  (* Subgoal 1 *)
  apply assumption
  (* Subgoal 2 *)
  apply assumption
done
```

Example: Proof by Elimination (1)

How do we prove $P \wedge (P \rightarrow Q) \rightarrow Q$?

Natural Deduction Proof

$$\frac{P \wedge (P \rightarrow Q) \Rightarrow Q}{P \wedge (P \rightarrow Q) \rightarrow Q} \text{impI}$$

(Faded background text: $\text{asm } P \Rightarrow P$, $\text{asm } [P; Q] \Rightarrow Q$, $\text{impE } [(P \rightarrow Q); P] \Rightarrow Q$, conjE)

Proof Script

```
lemma "P ∧ (P → Q) → Q"  
  apply (rule impI)  
  apply (erule conjE)  
  apply (erule impE)  
    (* Subgoal 1 *)  
    apply assumption  
    (* Subgoal 2 *)  
    apply assumption  
  done
```

Example: Proof by Elimination (1)

How do we prove $P \wedge (P \rightarrow Q) \rightarrow Q$?

Natural Deduction Proof

$$\frac{\frac{\frac{asm}{P \Rightarrow P} \quad \frac{asm}{[P; Q] \Rightarrow Q}}{[(P \rightarrow Q); P] \Rightarrow Q} \text{conjE}}{P \wedge (P \rightarrow Q) \Rightarrow Q} \text{conjE}}{P \wedge (P \rightarrow Q) \rightarrow Q} \text{impI}$$

Proof Script

```
lemma "P ∧ (P → Q) → Q"  
  apply (rule impI)  
  apply (erule conjE)  
  apply (erule impE)  
    (* Subgoal 1 *)  
    apply assumption  
  (* Subgoal 2 *)  
  apply assumption  
done
```

Example: Proof by Elimination (1)

How do we prove $P \wedge (P \rightarrow Q) \rightarrow Q$?

Natural Deduction Proof

$$\frac{\frac{\frac{asm \quad P \Rightarrow P}{impE} \quad \frac{asm \quad [P; Q] \Rightarrow Q}{conjE}}{[(P \rightarrow Q); P] \Rightarrow Q}}{P \wedge (P \rightarrow Q) \Rightarrow Q} \quad conjE}{P \wedge (P \rightarrow Q) \rightarrow Q} \quad impI$$

Proof Script

```
lemma "P ∧ (P → Q) → Q"  
  apply (rule impI)  
  apply (erule conjE)  
    apply (erule impE)  
      (* Subgoal 1 *)  
      apply assumption  
    (* Subgoal 2 *)  
    apply assumption  
  done
```

Example: Proof by Elimination (1)

How do we prove $P \wedge (P \rightarrow Q) \rightarrow Q$?

Natural Deduction Proof

$$\frac{\frac{\frac{\text{asm } P \Rightarrow P \quad \text{asm } [P; Q] \Rightarrow Q}{\text{impE}}}{\frac{[(P \rightarrow Q); P] \Rightarrow Q}{\text{conjE}}}}{\frac{P \wedge (P \rightarrow Q) \Rightarrow Q}{\text{impI}}}$$
$$\frac{P \wedge (P \rightarrow Q) \Rightarrow Q}{P \wedge (P \rightarrow Q) \rightarrow Q}$$

Proof Script

```
lemma "P ∧ (P → Q) → Q"  
  apply (rule impI)  
  apply (erule conjE)  
  apply (erule impE)  
    (* Subgoal 1 *)  
    apply assumption  
    (* Subgoal 2 *)  
    apply assumption  
  done
```

Example: Proof by Elimination (1)

How do we prove $P \wedge (P \rightarrow Q) \rightarrow Q$?

Natural Deduction Proof

$$\frac{\frac{\frac{P \Rightarrow P \quad \llbracket P; Q \rrbracket \Rightarrow Q}{\llbracket (P \rightarrow Q); P \rrbracket \Rightarrow Q} \text{impE}}{P \wedge (P \rightarrow Q) \Rightarrow Q} \text{conjE}}{P \wedge (P \rightarrow Q) \rightarrow Q} \text{impI}$$

Proof Script

```
lemma "P ∧ (P → Q) → Q"  
  apply (rule impI)  
  apply (erule conjE)  
  apply (erule impE)  
    (* Subgoal 1 *)  
    apply assumption  
  (* Subgoal 2 *)  
  apply assumption  
done
```

Example: Proof by Elimination (1)

How do we prove $P \wedge (P \rightarrow Q) \rightarrow Q$?

Natural Deduction Proof

$$\frac{\frac{\frac{}{P \Rightarrow P} \text{asm} \quad \frac{}{[P; Q] \Rightarrow Q} \text{asm}}{[(P \rightarrow Q); P] \Rightarrow Q} \text{conjE}}{P \wedge (P \rightarrow Q) \Rightarrow Q} \text{impI}}{P \wedge (P \rightarrow Q) \rightarrow Q} \text{impE}$$

Proof Script

```
lemma "P ∧ (P → Q) → Q"  
  apply (rule impI)  
  apply (erule conjE)  
  apply (erule impE)  
    (* Subgoal 1 *)  
    apply assumption  
  (* Subgoal 2 *)  
  apply assumption  
done
```

Example: Proof by Elimination (1)

How do we prove $P \wedge (P \rightarrow Q) \rightarrow Q$?

Natural Deduction Proof

$$\frac{\frac{\frac{}{P \Rightarrow P} \text{asm}}{\frac{[(P \rightarrow Q); P] \Rightarrow Q} \text{conjE}} \text{impI}}{\frac{[P; Q] \Rightarrow Q} \text{asm}} \text{impE}}{P \wedge (P \rightarrow Q) \rightarrow Q}$$

Proof Script

```
lemma "P ∧ (P → Q) → Q"  
  apply (rule impI)  
  apply (erule conjE)  
  apply (erule impE)  
    (* Subgoal 1 *)  
    apply assumption  
    (* Subgoal 2 *)  
  apply assumption  
done
```

Example: Proof by Elimination (1)

How do we prove $P \wedge (P \rightarrow Q) \rightarrow Q$?

Natural Deduction Proof

$$\frac{\frac{\frac{}{P \Rightarrow P} \text{asm}}{\frac{[(P \rightarrow Q); P] \Rightarrow Q} \text{conjE}} \text{impI}}{\frac{[P; Q] \Rightarrow Q} \text{asm}} \text{impE}}{P \wedge (P \rightarrow Q) \rightarrow Q}$$

Proof Script

```
lemma "P ∧ (P → Q) → Q"  
  apply (rule impI)  
  apply (erule conjE)  
  apply (erule impE)  
    (* Subgoal 1 *)  
    apply assumption  
    (* Subgoal 2 *)  
    apply assumption  
done
```

Example: Proof by Elimination (1)

How do we prove $P \wedge (P \rightarrow Q) \rightarrow Q$?

Natural Deduction Proof

$$\frac{\frac{\frac{}{P \Rightarrow P} \text{asm}}{\frac{[(P \rightarrow Q); P] \Rightarrow Q} \text{conjE}} \text{impI}}{\frac{[P; Q] \Rightarrow Q} \text{asm}} \text{impE}}{P \wedge (P \rightarrow Q) \rightarrow Q}$$

Proof Script

```
lemma "P ∧ (P → Q) → Q"
  apply (rule impI)
  apply (erule conjE)
  apply (erule impE)
    (* Subgoal 1 *)
  apply assumption
    (* Subgoal 2 *)
  apply assumption
done
```

Example: Proof by Elimination (1)

How do we prove $P \wedge (P \rightarrow Q) \rightarrow Q$?

Natural Deduction Proof

$$\frac{\frac{\frac{}{P \Rightarrow P} \text{asm}}{\frac{[(P \rightarrow Q); P] \Rightarrow Q} \text{conjE}} \text{impI}}{\frac{[P; Q] \Rightarrow Q} \text{asm}} \text{impE}}{P \wedge (P \rightarrow Q) \rightarrow Q}$$

Proof Script

```
lemma "P ∧ (P → Q) → Q"  
  apply (rule impI)  
  apply (erule conjE)  
  apply (erule impE)  
    (* Subgoal 1 *)  
    apply assumption  
    (* Subgoal 2 *)  
  apply assumption  
done
```

Example: Proof by Elimination (2): $(P \wedge P) \longleftrightarrow P$

Proof Script

```
lemma "(P ∧ P) ↔ P"  
  apply (rule iffI)  
    apply (erule conjE)  
    apply assumption  
  apply (rule conjI)  
    apply assumption  
    apply assumption  
  done
```

Example: Proof by Elimination (2): $(P \wedge P) \longleftrightarrow P$

$$\begin{array}{l} \text{asm} \\ \text{conjE} \\ \text{iffI} \end{array} \frac{}{P \wedge P \Rightarrow P} \quad \begin{array}{l} \text{asm} \\ \text{conjI} \end{array} \frac{P \Rightarrow P}{P \wedge P \Rightarrow P} \quad \begin{array}{l} \text{asm} \\ \text{conjI} \end{array} \frac{P \Rightarrow P}{P \wedge P \Rightarrow P}$$
$$(P \wedge P) \longleftrightarrow P$$

Proof Script

```
lemma "(P ∧ P) ↔ P"  
  apply (rule iffI)  
  apply (erule conjE)  
  apply assumption  
  apply (rule conjI)  
  apply assumption  
  apply assumption  
done
```

Example: Proof by Elimination (2): $(P \wedge P) \longleftrightarrow P$

$$\frac{\begin{array}{ccc} \text{asm} & \text{asm} & \text{asm} \\ \text{conjE} & \text{conjI} & \\ \frac{P, P \Rightarrow P}{P \wedge P \Rightarrow P} & \frac{P \Rightarrow P}{P \Rightarrow P \wedge P} & P \Rightarrow P \end{array}}{(P \wedge P) \longleftrightarrow P} \text{iffI}$$

Proof Script

```
lemma "(P ∧ P) ↔ P"
  apply (rule iffI)
  apply (erule conjE)
  apply assumption
  apply (rule conjI)
  apply assumption
  apply assumption
done
```

Example: Proof by Elimination (2): $(P \wedge P) \longleftrightarrow P$

$$\frac{\begin{array}{c} \text{asm} \\ \text{conjE} \\ P, P \Rightarrow P \end{array} \quad \begin{array}{c} \text{asm} \\ \text{conjI} \\ P \Rightarrow P \wedge P \end{array} \quad \begin{array}{c} \text{asm} \\ P \Rightarrow P \end{array}}{(P \wedge P) \longleftrightarrow P} \text{ iffI}$$

Proof Script

```
lemma "(P ∧ P) ↔ P"  
  apply (rule iffI)  
    apply (erule conjE)  
    apply assumption  
  apply (rule conjI)  
    apply assumption  
    apply assumption  
  done
```

Example: Proof by Elimination (2): $(P \wedge P) \longleftrightarrow P$

$$\frac{\frac{\text{asm } [P; P] \Rightarrow P}{P \wedge P \Rightarrow P} \text{ conjE} \quad \frac{\text{asm } P \Rightarrow P \quad \text{asm } P \Rightarrow P}{P \Rightarrow P \wedge P} \text{ conjI}}{(P \wedge P) \longleftrightarrow P} \text{ iffI}$$

Proof Script

```
lemma "(P ∧ P) ↔ P"  
  apply (rule iffI)  
  apply (erule conjE)  
  apply assumption  
  apply (rule conjI)  
  apply assumption  
  apply assumption  
done
```

Example: Proof by Elimination (2): $(P \wedge P) \longleftrightarrow P$

$$\frac{
 \frac{
 \frac{
 \text{asm} \overline{[P; P]} \Rightarrow P
 }{
 P \wedge P \Rightarrow P
 } \text{conjE}
 }{
 P \Rightarrow P \wedge P
 } \text{conjI}
 }{
 (P \wedge P) \longleftrightarrow P
 } \text{iffI}$$

Proof Script

```

lemma "(P ∧ P) ↔ P"
  apply (rule iffI)
  apply (erule conjE)
  apply assumption
  apply (rule conjI)
  apply assumption
  apply assumption
done

```

Example: Proof by Elimination (2): $(P \wedge P) \longleftrightarrow P$

$$\begin{array}{c}
 \frac{\overline{[P; P]} \Rightarrow P \quad \text{asm}}{P \wedge P \Rightarrow P} \quad \text{conjE} \\
 \hline
 (P \wedge P) \longleftrightarrow P \quad \text{iffI}
 \end{array}$$

Proof Script

```

lemma "(P ∧ P) ↔ P"
  apply (rule iffI)
  apply (erule conjE)
  apply assumption
  apply (rule conjI)
  apply assumption
  apply assumption
  done
  
```

Example: Proof by Elimination (2): $(P \wedge P) \longleftrightarrow P$

$$\begin{array}{c}
 \frac{\overline{[P; P]} \Rightarrow P \quad \text{asm}}{P \wedge P \Rightarrow P \quad \text{conjE}} \\
 \hline
 (P \wedge P) \longleftrightarrow P \quad \text{iffI}
 \end{array}$$

Proof Script

```

lemma "(P ∧ P) ↔ P"
  apply (rule iffI)
  apply (erule conjE)
  apply assumption
  apply (rule conjI)
  apply assumption
  apply assumption
  done
  
```

Example: Proof by Elimination (2): $(P \wedge P) \longleftrightarrow P$

$$\begin{array}{c}
 \frac{\overline{[P; P] \Rightarrow P} \text{ asm}}{P \wedge P \Rightarrow P} \text{ conjE} \quad \frac{\frac{P \Rightarrow P \text{ asm} \quad P \Rightarrow P \text{ asm}}{P \Rightarrow P \wedge P} \text{ conjI}}{P \wedge P \iff P} \text{ iffI} \\
 (P \wedge P) \longleftrightarrow P
 \end{array}$$

Proof Script

```

lemma "(P ∧ P) ↔ P"
  apply (rule iffI)
  apply (erule conjE)
  apply assumption
  apply (rule conjI)
  apply assumption
  apply assumption
done
  
```

Example: Proof by Elimination (2): $(P \wedge P) \longleftrightarrow P$

$$\frac{\frac{\frac{}{[P; P] \Rightarrow P} \text{asm}}{P \wedge P \Rightarrow P} \text{conjE} \quad \frac{\frac{P \Rightarrow P \quad P \Rightarrow P}{P \Rightarrow P \wedge P} \text{conjI}}{P \wedge P \Rightarrow P} \text{iffI}}{(P \wedge P) \longleftrightarrow P} \text{iffI}$$

Proof Script

```
lemma "(P ∧ P) ↔ P"  
  apply (rule iffI)  
  apply (erule conjE)  
  apply assumption  
  apply (rule conjI)  
  apply assumption  
  apply assumption  
done
```

Example: Proof by Elimination (2): $(P \wedge P) \longleftrightarrow P$

$$\frac{\frac{\frac{}{[P; P] \Rightarrow P} \text{asm}}{P \wedge P \Rightarrow P} \text{conjE} \quad \frac{\frac{\frac{}{P \Rightarrow P} \text{asm}}{P \Rightarrow P \wedge P} \text{conjI}}{P \Rightarrow P \wedge P} \text{iffI}}{(P \wedge P) \longleftrightarrow P} \text{iffI}$$

Proof Script

```

lemma "(P ∧ P) ↔ P"
  apply (rule iffI)
  apply (erule conjE)
  apply assumption
  apply (rule conjI)
  apply assumption
  apply assumption
done
  
```

Example: Proof by Elimination (2): $(P \wedge P) \longleftrightarrow P$

$$\frac{\frac{\frac{}{[P; P] \Rightarrow P} \text{asm}}{P \wedge P \Rightarrow P} \text{conjE}}{\frac{\frac{}{P \Rightarrow P} \text{asm} \quad \frac{P \Rightarrow P}{P \Rightarrow P \wedge P} \text{conjI}}{P \Rightarrow P \wedge P} \text{iffI}}{(P \wedge P) \longleftrightarrow P} \text{iffI}$$

Proof Script

```

lemma "(P ∧ P) ↔ P"
  apply (rule iffI)
  apply (erule conjE)
  apply assumption
  apply (rule conjI)
  apply assumption
  apply assumption
  done
  
```

Example: Proof by Elimination (2): $(P \wedge P) \longleftrightarrow P$

$$\frac{\frac{\frac{}{[P; P] \Rightarrow P} \text{asm}}{P \wedge P \Rightarrow P} \text{conjE}}{\frac{\frac{}{P \Rightarrow P} \text{asm}}{P \Rightarrow P \wedge P} \text{conjI}}{(P \wedge P) \longleftrightarrow P} \text{iffI}}$$

Proof Script

```

lemma "(P ∧ P) ↔ P"
  apply (rule iffI)
  apply (erule conjE)
  apply assumption
  apply (rule conjI)
  apply assumption
  apply assumption
  done
  
```

Example: Proof by Elimination (2): $(P \wedge P) \longleftrightarrow P$

$$\frac{\frac{\frac{}{[P; P] \Rightarrow P} \text{asm}}{P \wedge P \Rightarrow P} \text{conjE} \quad \frac{\frac{\frac{}{P \Rightarrow P} \text{asm}}{P \Rightarrow P \wedge P} \text{conjI}}{P \wedge P \Rightarrow P} \text{iffI}}{(P \wedge P) \longleftrightarrow P} \text{iffI}$$

Proof Script

```
lemma "(P ∧ P) ↔ P"  
  apply (rule iffI)  
  apply (erule conjE)  
  apply assumption  
  apply (rule conjI)  
  apply assumption  
  apply assumption  
done
```

Example: Proof by Elimination (2): $(P \wedge P) \longleftrightarrow P$

$$\frac{\frac{\frac{}{[P; P] \Rightarrow P} \text{asm}}{P \wedge P \Rightarrow P} \text{conjE} \quad \frac{\frac{\frac{}{P \Rightarrow P} \text{asm}}{P \Rightarrow P \wedge P} \text{conjI}}{P \wedge P \Rightarrow P} \text{iffI}}{(P \wedge P) \longleftrightarrow P} \text{iffI}$$

Proof Script

```
lemma "(P ∧ P) ↔ P"  
  apply (rule iffI)  
  apply (erule conjE)  
  apply assumption  
  apply (rule conjI)  
  apply assumption  
  apply assumption  
done
```

Outline

- 1 Low-Level Proof Scripts
- 2 Natural Deduction Rules for Propositional Calculus
- 3 Automation with the Classical Reasoner

Natural Deduction in Isar

```
lemma "P  $\wedge$  (P  $\longrightarrow$  Q)  $\longrightarrow$  Q"  
proof (rule impI)  
  assume "P  $\wedge$  (P  $\longrightarrow$  Q)"  
  hence "P  $\longrightarrow$  Q" "P"  
    by (erule_tac conjE, simp_all)  
  thus "Q"  
    by (erule_tac impE, simp_all)  
qed
```

Benefit of being readable without seeing the proof state.

A little too verbose for simple predicates.

Natural Deduction in Isar

```
lemma "P  $\wedge$  (P  $\longrightarrow$  Q)  $\longrightarrow$  Q"  
proof (rule impI)  
  assume "P  $\wedge$  (P  $\longrightarrow$  Q)"  
  hence "P  $\longrightarrow$  Q" "P"  
    by (erule_tac conjE, simp_all)  
  thus "Q"  
    by (erule_tac impE, simp_all)  
qed
```

benefit of being readable without seeing the proof state.

Apply to various other simple predicates.

Natural Deduction in Isar

```
lemma "P  $\wedge$  (P  $\longrightarrow$  Q)  $\longrightarrow$  Q"  
proof (rule impI)  
  assume "P  $\wedge$  (P  $\longrightarrow$  Q)"  
  hence "P  $\longrightarrow$  Q" "P"  
    by (erule_tac conjE, simp_all)  
  thus "Q"  
    by (erule_tac impE, simp_all)  
qed
```

benefit of being readable without seeing the proof state.

Apply the same idea for simple proof steps.

Natural Deduction in Isar

```
lemma "P  $\wedge$  (P  $\longrightarrow$  Q)  $\longrightarrow$  Q"  
proof (rule impI)  
  assume "P  $\wedge$  (P  $\longrightarrow$  Q)"  
  hence "P  $\longrightarrow$  Q" "P"  
    by (erule_tac conjE, simp_all)  
  thus "Q"  
    by (erule_tac impE, simp_all)  
qed
```

benefit of being readable without losing the proof state.

Let's now look at the proof above:

Natural Deduction in Isar

```
lemma "P  $\wedge$  (P  $\longrightarrow$  Q)  $\longrightarrow$  Q"  
proof (rule impI)  
  assume "P  $\wedge$  (P  $\longrightarrow$  Q)"  
  hence "P  $\longrightarrow$  Q" "P"  
    by (erule_tac conjE, simp_all)  
  thus "Q"  
    by (erule_tac impE, simp_all)  
qed
```

benefit of being readable without leaving the proof state.

Can be used to write more readable proofs.

Natural Deduction in Isar

```
lemma "P  $\wedge$  (P  $\longrightarrow$  Q)  $\longrightarrow$  Q"  
proof (rule impI)  
  assume "P  $\wedge$  (P  $\longrightarrow$  Q)"  
  hence "P  $\longrightarrow$  Q" "P"  
    by (erule_tac conjE, simp_all)  
  thus "Q"  
    by (erule_tac impE, simp_all)  
qed
```

Natural Deduction in Isar

```
lemma "P  $\wedge$  (P  $\longrightarrow$  Q)  $\longrightarrow$  Q"  
proof (rule impI)  
  assume "P  $\wedge$  (P  $\longrightarrow$  Q)"  
  hence "P  $\longrightarrow$  Q" "P"  
    by (erule_tac conjE, simp_all)  
  thus "Q"  
    by (erule_tac impE, simp_all)  
qed
```

Natural Deduction in Isar

```
lemma "P  $\wedge$  (P  $\longrightarrow$  Q)  $\longrightarrow$  Q"  
proof (rule impI)  
  assume "P  $\wedge$  (P  $\longrightarrow$  Q)"  
  hence "P  $\longrightarrow$  Q" "P"  
    by (erule_tac conjE, simp_all)  
  thus "Q"  
    by (erule_tac impE, simp_all)  
qed
```

Natural Deduction in Isar

```
lemma "P  $\wedge$  (P  $\longrightarrow$  Q)  $\longrightarrow$  Q"  
proof (rule impI)  
  assume "P  $\wedge$  (P  $\longrightarrow$  Q)"  
  hence "P  $\longrightarrow$  Q" "P"  
    by (erule_tac conjE, simp_all)  
  thus "Q"  
    by (erule_tac impE, simp_all)  
qed
```

- Benefit of being readable without seeing the proof state.
- A little too verbose for simple predicates.

Natural Deduction in Isar

```
lemma "P  $\wedge$  (P  $\longrightarrow$  Q)  $\longrightarrow$  Q"  
proof (rule impI)  
  assume "P  $\wedge$  (P  $\longrightarrow$  Q)"  
  hence "P  $\longrightarrow$  Q" "P"  
    by (erule_tac conjE, simp_all)  
  thus "Q"  
    by (erule_tac impE, simp_all)  
qed
```

- Benefit of being readable without seeing the proof state.
- A little too verbose for simple predicates.

Natural Deduction in Isar

```
lemma "P  $\wedge$  (P  $\longrightarrow$  Q)  $\longrightarrow$  Q"  
proof (rule impI)  
  assume "P  $\wedge$  (P  $\longrightarrow$  Q)"  
  hence "P  $\longrightarrow$  Q" "P"  
    by (erule_tac conjE, simp_all)  
  thus "Q"  
    by (erule_tac impE, simp_all)  
qed
```

- Benefit of being readable without seeing the proof state.
- A little too verbose for simple predicates.

Automating Natural Deduction

- Isn't all this low level deduction too much work?
- Fortunately, Isabelle automates natural deduction using the **blast** tactic.
- **Tableaux prover**: represent proof tree structure.

Example

```
lemma "P  $\longrightarrow$  (P  $\wedge$  (P  $\vee$  Q))" by blast
lemma "P  $\wedge$  (P  $\longrightarrow$  Q)  $\longrightarrow$  Q" by blast
```

- Blast searches **proof tree** using introduction and elimination rules.
- `blast intro: thms elim: thms, or [intro], [elim], and [dest]`.
- **Safe** rules marked with "!", e.g. `[intro!]`.
- **Unsafe** rules applied only when no alternative.
- Blast is **one-shot**: all or nothing.

Automating Natural Deduction

- Isn't all this low level deduction too much work?
- Fortunately, Isabelle automates natural deduction using the **blast** tactic.
- **Tableaux prover**: represent proof tree structure.

Example

```
lemma "P  $\longrightarrow$  (P  $\wedge$  (P  $\vee$  Q))" by blast
lemma "P  $\wedge$  (P  $\longrightarrow$  Q)  $\longrightarrow$  Q" by blast
```

- Blast searches **proof tree** using introduction and elimination rules.
- `blast intro: thms elim: thms, or [intro], [elim], and [dest]`.
- **Safe** rules marked with "!", e.g. `[intro!]`.
- **Unsafe** rules applied only when no alternative.
- Blast is **one-shot**: all or nothing.

Automating Natural Deduction

- Isn't all this low level deduction too much work?
- Fortunately, Isabelle automates natural deduction using the **blast** tactic.
- **Tableaux prover**: represent proof tree structure.

Example

```
lemma "P  $\longrightarrow$  (P  $\wedge$  (P  $\vee$  Q))" by blast
lemma "P  $\wedge$  (P  $\longrightarrow$  Q)  $\longrightarrow$  Q" by blast
```

- Blast searches **proof tree** using introduction and elimination rules.
- `blast intro: thms elim: thms, or [intro], [elim], and [dest]`.
- **Safe** rules marked with "!", e.g. `[intro!]`.
- **Unsafe** rules applied only when no alternative.
- Blast is **one-shot**: all or nothing.

Automating Natural Deduction

- Isn't all this low level deduction too much work?
- Fortunately, Isabelle automates natural deduction using the **blast** tactic.
- **Tableaux prover**: represent proof tree structure.

Example

```
lemma "P  $\longrightarrow$  (P  $\wedge$  (P  $\vee$  Q))" by blast
lemma "P  $\wedge$  (P  $\longrightarrow$  Q)  $\longrightarrow$  Q" by blast
```

- Blast searches **proof tree** using introduction and elimination rules.
- `blast intro: thms elim: thms, or [intro], [elim], and [dest]`.
- **Safe** rules marked with "!", e.g. `[intro!]`.
- **Unsafe** rules applied only when no alternative.
- Blast is **one-shot**: all or nothing.

Automating Natural Deduction

- Isn't all this low level deduction too much work?
- Fortunately, Isabelle automates natural deduction using the **blast** tactic.
- **Tableaux prover**: represent proof tree structure.

Example

```
lemma "P  $\longrightarrow$  (P  $\wedge$  (P  $\vee$  Q))" by blast
lemma "P  $\wedge$  (P  $\longrightarrow$  Q)  $\longrightarrow$  Q" by blast
```

- Blast searches **proof tree** using introduction and elimination rules.
- `blast intro: thms elim: thms, or [intro], [elim], and [dest]`.
- **Safe** rules marked with "!", e.g. `[intro!]`.
- **Unsafe** rules applied only when no alternative.
- Blast is **one-shot**: all or nothing.

Automating Natural Deduction

- Isn't all this low level deduction too much work?
- Fortunately, Isabelle automates natural deduction using the **blast** tactic.
- **Tableaux prover**: represent proof tree structure.

Example

```
lemma "P  $\longrightarrow$  (P  $\wedge$  (P  $\vee$  Q))" by blast
lemma "P  $\wedge$  (P  $\longrightarrow$  Q)  $\longrightarrow$  Q" by blast
```

- Blast searches **proof tree** using introduction and elimination rules.
- `blast intro: thms elim: thms, or [intro], [elim], and [dest]`.
- **Safe** rules marked with "!", e.g. `[intro!]`.
- **Unsafe** rules applied only when no alternative.
- Blast is **one-shot**: all or nothing.

Automating Natural Deduction

- Isn't all this low level deduction too much work?
- Fortunately, Isabelle automates natural deduction using the **blast** tactic.
- **Tableaux prover**: represent proof tree structure.

Example

lemma "P \longrightarrow (P \wedge (P \vee Q))" **by** blast

lemma "P \wedge (P \longrightarrow Q) \longrightarrow Q" **by** blast

- Blast searches **proof tree** using introduction and elimination rules.
- `blast intro: thms elim: thms, or [intro], [elim], and [dest]`.
- Safe rules marked with "!", e.g. `[intro!]`.
- Unsafe rules applied only when no alternative.
- Blast is **one-shot**: all or nothing.

Automating Natural Deduction

- Isn't all this low level deduction too much work?
- Fortunately, Isabelle automates natural deduction using the **blast** tactic.
- **Tableaux prover**: represent proof tree structure.

Example

lemma "P \longrightarrow (P \wedge (P \vee Q))" **by** blast

lemma "P \wedge (P \longrightarrow Q) \longrightarrow Q" **by** blast

- Blast searches **proof tree** using introduction and elimination rules.
- `blast intro: thms elim: thms, or [intro], [elim], and [dest]`.
- **Safe** rules marked with "!", e.g. `[intro!]`.
- **Unsafe** rules applied only when no alternative.
- Blast is **one-shot**: all or nothing.

Automating Natural Deduction

- Isn't all this low level deduction too much work?
- Fortunately, Isabelle automates natural deduction using the **blast** tactic.
- **Tableaux prover**: represent proof tree structure.

Example

```
lemma "P  $\longrightarrow$  (P  $\wedge$  (P  $\vee$  Q))" by blast
```

```
lemma "P  $\wedge$  (P  $\longrightarrow$  Q)  $\longrightarrow$  Q" by blast
```

- Blast searches **proof tree** using introduction and elimination rules.
- **blast intro: thms elim: thms**, or **[intro]**, **[elim]**, and **[dest]**.
- Safe rules marked with "!", e.g. **[intro!]**.
- Unsafe rules applied only when no alternative.
- Blast is **one-shot**: all or nothing.

Automating Natural Deduction

- Isn't all this low level deduction too much work?
- Fortunately, Isabelle automates natural deduction using the **blast** tactic.
- **Tableaux prover**: represent proof tree structure.

Example

lemma "P \longrightarrow (P \wedge (P \vee Q))" **by** blast

lemma "P \wedge (P \longrightarrow Q) \longrightarrow Q" **by** blast

- Blast searches **proof tree** using introduction and elimination rules.
- **blast intro: thms elim: thms**, or **[intro]**, **[elim]**, and **[dest]**.
- **Safe** rules marked with "!", e.g. **[intro!]**.
- **Unsafe** rules applied only when no alternative.
- Blast is **one-shot**: all or nothing.

Automating Natural Deduction

- Isn't all this low level deduction too much work?
- Fortunately, Isabelle automates natural deduction using the **blast** tactic.
- **Tableaux prover**: represent proof tree structure.

Example

lemma "P \longrightarrow (P \wedge (P \vee Q))" **by** blast

lemma "P \wedge (P \longrightarrow Q) \longrightarrow Q" **by** blast

- Blast searches **proof tree** using introduction and elimination rules.
- **blast intro: thms elim: thms**, or **[intro]**, **[elim]**, and **[dest]**.
- **Safe** rules marked with "!", e.g. **[intro!]**.
- **Unsafe** rules applied only when no alternative.
- Blast is **one-shot**: all or nothing.

Automating Natural Deduction

- Isn't all this low level deduction too much work?
- Fortunately, Isabelle automates natural deduction using the **blast** tactic.
- **Tableaux prover**: represent proof tree structure.

Example

```
lemma "P  $\longrightarrow$  (P  $\wedge$  (P  $\vee$  Q))" by blast
```

```
lemma "P  $\wedge$  (P  $\longrightarrow$  Q)  $\longrightarrow$  Q" by blast
```

- Blast searches **proof tree** using introduction and elimination rules.
- **blast intro: thms elim: thms**, or **[intro]**, **[elim]**, and **[dest]**.
- **Safe** rules marked with "!", e.g. **[intro!]**.
- **Unsafe** rules applied only when no alternative.
- Blast is **one-shot**: all or nothing.

Combining Deduction and Simplification with Auto

- auto is a powerful tactic combining deduction (eg, `intro`) and simplification (eg, `simp`) together to help you simplify a goal.
- auto loses information as long as the specified tactic rules are truly safe.
- General technique for breaking a complex goal into several smaller goals:

```
(intro ... (auto simp add: ...)) ... (auto simp add: ...)
```
- There are also meta-tactics like `force` and `fastforce` that auto can produce unsound and bizarre results.
- Still need manual deduction to understand why automated proof fails.

Combining Deduction and Simplification with Auto

- `auto` is a powerful tactic combining deduction (e.g., `blast`) and `simp`.
- Not one-shot: applies `safe` rules repeatedly, simplifying if possible.
- Won't lose information as long as the specified safe rules are truly safe.
- General technique for breaking a complex goal into several parts.

```
lemma "(x::nat) = y2 → x ≥ y ∧ x ≥ 0"  
  by (auto simp add: power2_nat_le_imp_le)
```

- There are also one-shot variants: `force` and `fastforce`.
- `auto` can produce unpredictable and bizarre results.
- Still need manual deduction to understand why automated proof fails.

Combining Deduction and Simplification with Auto

- `auto` is a powerful tactic combining deduction (e.g., `blast`) and `simp`.
- **Not one-shot**: applies `safe` rules repeatedly, simplifying if possible.
- Won't lose information as long as the specified safe rules are truly safe.
- General technique for breaking a complex goal into several parts.

```
lemma "(x::nat) = y2 → x ≥ y ∧ x ≥ 0"  
  by (auto simp add: power2_nat_le_imp_le)
```

- There are also one-shot variants: `force` and `fastforce`.
- `auto` can produce unpredictable and bizarre results.
- Still need manual deduction to understand why automated proof fails.

Combining Deduction and Simplification with Auto

- `auto` is a powerful tactic combining deduction (e.g., `blast`) and `simp`.
- **Not one-shot**: applies `safe` rules repeatedly, simplifying if possible.
- Won't lose information as long as the specified safe rules are truly safe.
- General technique for breaking a complex goal into several parts.

```
lemma "(x::nat) = y2 → x ≥ y ∧ x ≥ 0"  
  by (auto simp add: power2_nat_le_imp_le)
```

- There are also one-shot variants: `force` and `fastforce`.
- `auto` can produce unpredictable and bizarre results.
- Still need **manual deduction** to understand why automated proof fails.

Combining Deduction and Simplification with Auto

- `auto` is a powerful tactic combining deduction (e.g., `blast`) and `simp`.
- **Not one-shot**: applies `safe` rules repeatedly, simplifying if possible.
- Won't lose information as long as the specified safe rules are truly safe.
- General technique for breaking a complex goal into several parts.

```
lemma "(x::nat) = y2 → x ≥ y ∧ x ≥ 0"  
  by (auto simp add: power2_nat_le_imp_le)
```

- There are also one-shot variants: `force` and `fastforce`.
- `auto` can produce unpredictable and bizarre results.
- Still need **manual deduction** to understand why automated proof fails.

Combining Deduction and Simplification with Auto

- `auto` is a powerful tactic combining deduction (e.g., `blast`) and `simp`.
- **Not one-shot**: applies `safe` rules repeatedly, simplifying if possible.
- Won't lose information as long as the specified safe rules are truly safe.
- General technique for breaking a complex goal into several parts.

```
lemma "(x::nat) = y2 → x ≥ y ∧ x ≥ 0"  
by (auto simp add: power2_nat_le_imp_le)
```

- There are also one-shot variants: `force` and `fastforce`.
- `auto` can produce unpredictable and bizarre results.
- Still need `manual deduction` to understand why automated proof fails.

Combining Deduction and Simplification with Auto

- `auto` is a powerful tactic combining deduction (e.g., `blast`) and `simp`.
- **Not one-shot**: applies `safe` rules repeatedly, simplifying if possible.
- Won't lose information as long as the specified safe rules are truly safe.
- General technique for breaking a complex goal into several parts.

```
lemma "(x::nat) = y2 → x ≥ y ∧ x ≥ 0"  
  by (auto simp add: power2_nat_le_imp_le)
```

- There are also one-shot variants: `force` and `fastforce`.
- `auto` can produce unpredictable and bizarre results.
- Still need `manual deduction` to understand why automated proof fails.

Combining Deduction and Simplification with Auto

- `auto` is a powerful tactic combining deduction (e.g., `blast`) and `simp`.
- **Not one-shot**: applies `safe` rules repeatedly, simplifying if possible.
- Won't lose information as long as the specified safe rules are truly safe.
- General technique for breaking a complex goal into several parts.

```
lemma "(x::nat) = y2 → x ≥ y ∧ x ≥ 0"  
  by (auto simp add: power2_nat_le_imp_le)
```

- There are also one-shot variants: `force` and `fastforce`.
- `auto` can produce unpredictable and bizarre results.
- Still need **manual deduction** to understand why automated proof fails.

Applications

- We looked at automating natural deduction in the propositional calculus.
- We demonstrated a far more general technique.
- We define bespoke logics in Isabelle and then reason about them.
- In particular, we can program verification (e.g., Hoare logic).
- Natural deduction is an important weapon in the proof arsenal.

Applications

- We looked at automating natural deduction in the propositional calculus.
- We demonstrated a far more general technique.
- We define bespoke logics in Isabelle and then reason about them.
- In particular, we can program verification (e.g., Hoare logic).
- Natural deduction is an important weapon in the proof arsenal.

Applications

- We looked at automating natural deduction in the propositional calculus.
- We demonstrated a far more general technique.
- We define bespoke logics in Isabelle and then reason about them.
- In particular, we can program verification (e.g., Hoare logic).
- Natural deduction is an important weapon in the proof arsenal.

Applications

- We looked at automating natural deduction in the propositional calculus.
- We demonstrated a far more general technique.
- We define bespoke logics in Isabelle and then reason about them.
- In particular, we can **program verification** (e.g., **Hoare logic**).
- Natural deduction is an important weapon in the proof arsenal.

Applications

- We looked at automating natural deduction in the propositional calculus.
- We demonstrated a far more general technique.
- We define bespoke logics in Isabelle and then reason about them.
- In particular, we can **program verification** (e.g., **Hoare logic**).
- Natural deduction is an important weapon in the proof arsenal.

Conclusion

This Lecture

- Apply-style proof scripts.
- Natural deduction in Isabelle/HOL.
- The classical reasoner.

Next Lecture

- Predicate calculus (quantifiers etc.).

Conclusion

This Lecture

- Apply-style proof scripts.
- Natural deduction in Isabelle/HOL.
- The classical reasoner.

Next Lecture

- Predicate calculus (quantifiers etc.).

Conclusion

This Lecture

- Apply-style proof scripts.
- Natural deduction in Isabelle/HOL.
- The classical reasoner.

Next Lecture

- Predicate calculus (quantifiers etc.).

Conclusion

This Lecture

- Apply-style proof scripts.
- Natural deduction in Isabelle/HOL.
- The classical reasoner.

Next Lecture

- Predicate calculus (quantifiers etc.).

Conclusion

This Lecture

- Apply-style proof scripts.
- Natural deduction in Isabelle/HOL.
- The classical reasoner.

Next Lecture

- Predicate calculus (quantifiers etc.).

Conclusion

This Lecture

- Apply-style proof scripts.
- Natural deduction in Isabelle/HOL.
- The classical reasoner.

Next Lecture

- Predicate calculus (quantifiers etc.).

Conclusion

This Lecture

- Apply-style proof scripts.
- Natural deduction in Isabelle/HOL.
- The classical reasoner.

Next Lecture

- Predicate calculus (quantifiers etc.).