Automated Reasoning

Jacques Fleuriot

September 26, 2013

Propositional Reasoning in Isabelle [1]

Jacques Fleuriot

[1] With contributions by Paul Jackson

Recap

Last time we

- Looked at rules for the **natural deduction** calculus
- Introduced **sequent** style notation for derivation, $\Gamma \vdash \phi$

Today we will

- See how propositional proof works in Isabelle
- Time permitting, look at two full set of sequent rules

Sequents in Isabelle

Sequents expressed using **meta-implication** $\Longrightarrow$

$$P_1, P_2, \ldots, P_n \vdash Q$$

expressed as

$$P_1 \Longrightarrow (P_2 \Longrightarrow \ldots \Longrightarrow (P_n \Longrightarrow Q) \ldots)$$

or

$$[\![P_1; P_2; \ldots; P_n]\!] \Longrightarrow Q$$

Meta-implication is also used to express relationship between premises and conclusion of rules.

$$\frac{\begin{array}{c} P \\ \vdots \\ Q \end{array}}{P \longrightarrow Q}\ \mathit{impI} \qquad \text{is written as} \quad (P \Longrightarrow Q) \Longrightarrow (P \longrightarrow Q)$$

Interlude: Isabelle *Getting Started Guide*

This document on the AR slides webpage explains

- How Isabelle interactive theorem prover is started up
- Basics of *Proof General* user interface for Isabelle
- Examples of propositional logic proofs in Isabelle

The `rule` Method

Used when the conclusion of theorem matches the conclusion of the current goal. It applies the theorem backward.

Consider the theorem `disjI1`

$$P \Longrightarrow P \vee Q$$

Applying `rule disjI1` to goal

$$[\![A; B; C]\!] \Longrightarrow (A \wedge B) \vee D$$

yields the subgoal

$$[\![A; B; C]\!] \Longrightarrow A \wedge B$$

Matching and Unification

In applying rule

$$P \Longrightarrow P \vee Q$$

to goal

$$[\![A; B; C]\!] \Longrightarrow (A \wedge B) \vee D$$

The pattern $P \vee Q$ is **matched** with the target $(A \wedge B) \vee D$ to yield the instantiation $P \mapsto A \wedge B,\ Q \mapsto D$ which make the pattern and target the same. The following goal results

$$[\![A; B; C]\!] \Longrightarrow A \wedge B$$

In general, if the goal conclusion contains schematic variables, the rule and goal conclusions are **unified** i.e. both are instantiated so as to make them the same.

More on unification later!

General definition of method `rule`

When we apply the method *`rule SomeRule`* where

$$SomeRule : [\![P_1; \ldots; P_m]\!] \Longrightarrow Q$$

to the goal

$$[\![A_1; \ldots; A_n]\!] \Longrightarrow C$$

where Q and C can be unified, we generate the goals

$$\begin{array}{c} [\![A'_1; \quad \ldots; \quad A'_n]\!] \Longrightarrow P'_1 \\ \vdots \\ [\![A'_1; \quad \ldots; \quad A'_n]\!] \Longrightarrow P'_m \end{array}$$

where $A'_1, A'_2, \ldots, A'_n, P'_1, P'_2, \ldots, P'_m$ are the results of applying the substitution which unifies Q and C to $A_1, A_2, \ldots, A_n, P_1, P_2, \ldots, P_m$. That is, we must now derive each of the rule's assumptions using our goal's assumptions.

The `erule` Method

Used when the conclusion of theorem matches the conclusion of the current goal and the first premise of theorem matches a premise of the current goal.

Consider the theorem `disjE`

$$[\![P \vee Q; P \Longrightarrow R; Q \Longrightarrow R]\!] \Longrightarrow R$$

Applying `erule disjE` to goal

$$[\![(A \wedge B) \vee C; D]\!] \Longrightarrow B \vee C$$

yields the subgoals

$$[\![D; (A \wedge B)]\!] \Longrightarrow B \vee C \qquad [\![D; C]\!] \Longrightarrow B \vee C$$

General definition of method `erule`

When we apply the method *`erule someRule`* where

$$SomeRule : [\![P_1; \ldots; P_m]\!] \Longrightarrow Q$$

to the goal

$$[\![A_1; \ldots; A_n]\!] \Longrightarrow C$$

where P_1 and A_1 are unifiable and Q and C are unifiable, we generate the goals:

$$\begin{array}{llll} [\![A_2'; & \ldots; & A_n']\!] \Longrightarrow P_2' \\ & \vdots & \\ [\![A_2'; & \ldots; & A_n']\!] \Longrightarrow P_m' \end{array}$$

where $A_2', A_3', \ldots, A_n', P_2', P_3' \ldots, P_m'$ are the results of applying the substitution, which unifies P_1 to A_1 and Q to C, to $A_2, A_3, \ldots, A_n, P_2, P_3, \ldots, P_m$. That is, we **eliminate** an assumption from the rule and the goal, and must derive the rule's other assumptions using our goal's other assumptions.

General definition of method `drule`

When we apply the method *`drule someRule`* where

$$SomeRule : [\![P_1; \ldots; P_m]\!] \Longrightarrow Q$$

to the goal

$$[\![A_1; \ldots; A_n]\!] \Longrightarrow C$$

where P_1 and A_1 are unifiable, we generate the goals:

$$\begin{array}{rrr} [\![A_2'; & \ldots; & A_n']\!] \Longrightarrow P_2' \\ & \vdots & \\ [\![A_2'; & \ldots; & A_n']\!] \Longrightarrow P_m' \\ [\![Q'; A_2'; & \ldots; & A_n']\!] \Longrightarrow C' \end{array}$$

where $A_2', A_3', \ldots, A_n', P_2', P_3' \ldots, P_m', Q', C'$ are the results of applying the substitution which unifies P_1 and A_1 to $A_2, A_3, \ldots, A_n, P_2, P_3, \ldots, P_m, Q, C$. That is, we **delete** an assumption from the goal, replacing it with the conclusion of the rule.

General definition of method `frule`

When we apply the method *`frule someRule`* where

$$SomeRule : [\![P_1; \ldots; P_m]\!] \Longrightarrow Q$$

to the goal

$$[\![A_1; \ldots; A_n]\!] \Longrightarrow C$$

where P_1 and A_1 are unifiable, we generate the goals:

$$\begin{array}{rrr} [\![A'_1; & \ldots; & A'_n]\!] \Longrightarrow P'_2 \\ & \vdots & \\ [\![A'_1; & \ldots; & A'_n]\!] \Longrightarrow P'_m \\ [\![Q'; A'_1; & \ldots; & A'_n]\!] \Longrightarrow C' \end{array}$$

where $A'_1, A'_2, \ldots, A'_n, P'_2, \ldots, P'_m, Q', C'$ are the results of applying the substitution which unifies P_1 and A_1 to $A_1, A_2, \ldots, A_n, P_2, \ldots, P_m, Q, C$. This is like *`drule`* except the assumption in our goal is kept.

More Methods

- *rule_tac*, *erule_tac*, *drule_tac* and *frule_tac* are like their counterparts, but you can give substitutions for variables in the rule before they are applied.

Example

$$\textit{erule\_tac } Q=\text{"}B \wedge D\text{"} \textbf{ in } \textit{conjE}$$

applied to the subgoal

$$[\![A \wedge B;\ C \wedge B \wedge D]\!] \Longrightarrow B \wedge D$$

generates the new goal

$$[\![A \wedge B;\ C;\ B \wedge D]\!] \Longrightarrow B \wedge D$$

- Isabelle also provides advanced tactics, like *simp* and *auto* which perform some **automatic deduction**.

Addendum: Sequent-style rules

Two kinds of sequent-style rule systems:

- N-systems: use natural-deduction rules for introduction and elimination of operators in the conclusion
- L-systems: use instead *left* and *right* introduction rules.

Right introduction rules are the same as natural deduction introduction rules

Left introduction rules introduce operators in the set of assumptions.

Example:

$$\frac{\Gamma, P \vdash R \quad \Gamma, Q \vdash R}{\Gamma, P \vee Q \vdash R}\ \mathit{disjLeftI}$$

This is easily derived from the sequent *disjE* rule and the sequent assumption rule

$$\frac{}{\Gamma, P \vdash P}\ \mathit{assum}$$

Isabelle rules are phrased in a natural deduction style, but proofs in practice use these rules in an L-like fashion.

Addendum: Sequent-style Natural Deduction Rules

$\wedge$ $\quad \dfrac{\Gamma \vdash P \quad \Gamma \vdash Q}{\Gamma \vdash P \wedge Q}\ \mathit{conjI} \qquad \dfrac{\Gamma \vdash P \wedge Q \quad \Gamma, P, Q \vdash R}{\Gamma \vdash R}\ \mathit{conjE}$

$\vee$ $\quad \dfrac{\Gamma \vdash P}{\Gamma \vdash P \vee Q}\ \mathit{disjI1} \qquad \dfrac{\Gamma \vdash Q}{\Gamma \vdash P \vee Q}\ \mathit{disjI2} \qquad \dfrac{\Gamma \vdash P \vee Q \quad \Gamma, P \vdash R \quad \Gamma, Q \vdash R}{\Gamma \vdash R}\ \mathit{disjE}$

$\longrightarrow$ $\quad \dfrac{\Gamma, P \vdash Q}{\Gamma \vdash P \longrightarrow Q}\ \mathit{impI} \qquad \dfrac{\Gamma \vdash P \longrightarrow Q \quad \Gamma \vdash P \quad \Gamma, Q \vdash R}{\Gamma \vdash R}\ \mathit{impE}$

$\longleftrightarrow$ $\quad \dfrac{\Gamma, Q \vdash P \quad \Gamma, P \vdash Q}{\Gamma \vdash P \longleftrightarrow Q}\ \mathit{iffI} \qquad \dfrac{\Gamma \vdash Q \longleftrightarrow P \quad \Gamma \vdash Q}{\Gamma \vdash P}\ \mathit{iffD1} \qquad \dfrac{\Gamma \vdash P \longleftrightarrow Q \quad \Gamma \vdash Q}{\Gamma \vdash P}\ \mathit{iffD2}$

$\neg$ $\quad \dfrac{\Gamma, P \vdash \bot}{\Gamma \vdash \neg P}\ \mathit{notI} \qquad \dfrac{\Gamma \vdash \neg P \quad \Gamma \vdash P}{\Gamma \vdash R}\ \mathit{notE} \qquad \dfrac{\Gamma, \neg P \vdash \bot}{\Gamma \vdash P}\ \mathit{ccontr} \qquad \dfrac{}{\Gamma \vdash P \vee \neg P}\ \mathit{excluded\_middle}$

$\dfrac{}{\Gamma, P \vdash P}\ \mathit{assumption}$

Addendum: Sequent-style L-System Rules

$\wedge$ $\quad \dfrac{\Gamma \vdash P \quad \Gamma \vdash Q}{\Gamma \vdash P \wedge Q}\ conjI \qquad \dfrac{\Gamma, P, Q \vdash R}{\Gamma, P \wedge Q \vdash R}\ e\ conjE$

$\vee$ $\quad \dfrac{\Gamma \vdash P}{\Gamma \vdash P \vee Q}\ disjI1 \qquad \dfrac{\Gamma \vdash Q}{\Gamma \vdash P \vee Q}\ disjI2 \qquad \dfrac{\Gamma, P \vdash R \quad \Gamma, Q \vdash R}{\Gamma, P \vee Q \vdash R}\ e\ disjE$

$\longrightarrow$ $\quad \dfrac{\Gamma, P \vdash Q}{\Gamma \vdash P \longrightarrow Q}\ impI \qquad \dfrac{\Gamma \vdash P \quad \Gamma, Q \vdash R}{\Gamma, P \longrightarrow Q \vdash R}\ e\ impE$

$\longleftrightarrow$ $\quad \dfrac{\Gamma, Q \vdash P \quad \Gamma, P \vdash Q}{\Gamma \vdash P \longleftrightarrow Q}\ iffI \qquad \dfrac{\Gamma, P \longrightarrow Q, Q \longrightarrow P \vdash R}{\Gamma, P \longleftrightarrow Q \vdash R}\ e\ iffE$

$\neg$ $\quad \dfrac{\Gamma, P \vdash \bot}{\Gamma \vdash \neg P}\ notI \qquad \dfrac{\Gamma \vdash P}{\Gamma, \neg P \vdash R}\ e\ notE \qquad \dfrac{\Gamma, \neg P \vdash \bot}{\Gamma \vdash P}\ ccontr \qquad \dfrac{}{\Gamma \vdash P \vee \neg P}\ excluded\_middle$

$\quad \dfrac{}{\Gamma, P \vdash P}\ assumption \qquad \dfrac{\Gamma, P \vdash R \quad \Gamma \vdash P}{\Gamma \vdash R}\ subgoal\_tacP$

- Note that *e* **rule** is short for *erule* **rule**. In These rules are also called *left introduction* rules.
- Variations on left intro rules preserve operator in first. premise