PROPOSITIONAL REASONING IN ISABELLE

PROBLEM w/ rule METHOD

e.g. Consider the disjE rule:

disjE : $[[?P \vee ?Q ;\ ?P \Rightarrow ?R ;\ ?Q \Rightarrow ?R]] \Rightarrow ?R$

→ If we have the goal:

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

Then applying disjE rule produces 3 new goals:

$[[(A \wedge B) \vee C ;\ D]] \rightarrow ?P \vee ?Q$ → can be solved by applying assumption

↳ This seems pointlessly roundabout

$[[(A \wedge B) \vee C ;\ D ;\ ?P]] \Rightarrow B \vee C$

↳ We often want to use one of our assumptions in our proof

$[[(A \wedge B) \vee C ;\ D ;\ ?Q]] \Rightarrow B \vee C$

SOLN

erule METHOD

→ used when the conclusion of rule matches the concl. of the current goal AND the first premise of rule matches a premise of the current goal

→ e.g.

disjE : $[[P \vee Q ;\ P \Rightarrow R ;\ Q \Rightarrow R]] \Rightarrow R$ [omit '?']

Apply erule disjE

goal : $[[\boxed{(A \wedge B)} \vee \boxed{C} ;\ D]] \Rightarrow \boxed{B \vee C}$

The subgoals yielded are $[[D ;\ (A \wedge B)]] \Rightarrow B \vee C$

$[[D ;\ C]] \Rightarrow B \vee C$

→ We eliminate an assumption frm the rule and the goal and must derive the rule's other assumptions using our goal's other assumptions

drule METHOD

→ someRule : $[[P_1 ; \cdots ; P_m]] \Rightarrow Q$

goal : $[[A_1 ; \cdots ; A_n]] \Rightarrow C$

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

$$[[A_2' ; \cdots ; A_n']] \Rightarrow P_2'$$
$$\vdots$$
$$[[A_2' ; \cdots ; A_n']] \Rightarrow P_m'$$
$$[[Q' ; A_2' ; \cdots ; A_n']] \Rightarrow C'$$

→ We delete an assumption, replacing it w/ the cond. of the rule

frule METHOD

→ w/ above someRule & goal, we generate the goals:

$$[[A_1' ; A_2' ; \cdots ; A_n']] \Rightarrow P_2'$$
$$\vdots$$
$$[[A_1' ; A_2' ; \cdots ; A_n']] \Rightarrow P_m'$$
$$[[Q' ; A_1' ; A_2' ; \cdots ; A_n']] \Rightarrow C'$$

→ This is like drule except the assumption in our goal is kept.

MORE METHODS

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

→ e.g. apply (erule_tac Q = "$B \wedge D$" in conjE)

conjE: $[[P \wedge Q ; [[P ; Q]] \Rightarrow R]] \Rightarrow R$

to the subgoal $[[A \wedge B ; C \wedge B \wedge D]] \Rightarrow B \wedge D$

generates new goal $[[A \wedge B ; C ; B \wedge D]] \Rightarrow B \wedge D$

SEQUENT CALCULUS / L-SYSTEMS

→ Instead of elimination rules:

ie.
$$\frac{\Gamma \vdash P \vee Q \qquad \Gamma, P \vdash R \qquad \Gamma, Q \vdash R}{\Gamma \vdash R} \ (\text{disjE})$$

We have left introduction rules → often much easier to use in a backwards, goal-directed style

$$\frac{\Gamma, P \vdash R \qquad \Gamma, Q \vdash R}{\Gamma, P \vee Q \vdash R}$$

→ This corresponds to applying rules using erule in Isabelle

THE CUT RULE

$$\boxed{\frac{\Gamma \vdash P \qquad \Gamma, P \vdash Q}{\Gamma \vdash Q}}$$

allows the use of a lemma P in a proof of Q.

We can now reuse P multiple times in the proof of Q.

→ In Isabelle, `cut_tac lemmaName` — adds the concl. of *lemmaName* as a new assumption, and its assumptions as new subgoals

`subgoal_tac P` — adds P as a new assumption and introduces P as a new subgoal

→ excluded_middle : "$P \vee \neg P$"

→ notnotD : "$\neg\neg P \Rightarrow P$"

apply (cut_tac P="P" in excluded_middle)

goal: $[\![\neg\neg P;\ P \vee \neg P]\!] \Rightarrow P$

→ For excluded_middle :

$$\cfrac{\cfrac{[\![\neg(P \vee \neg P);\ \neg P]\!] \vdash \text{False} \qquad [\![\neg(P \vee \neg P);\ P]\!] \vdash \text{False}}{\neg(P \vee \neg P) \vdash \text{False}} \ (\text{cut\_tac P="P" in ccontr})}{\vdash P \vee \neg P} \ (\text{ccontr})$$

$$\frac{\Gamma \vdash P \qquad \Gamma \vdash Q}{\Gamma \vdash P \wedge Q} \text{ (conjI)} \qquad \frac{\Gamma, P, Q \vdash R}{\Gamma, P \wedge Q \vdash R} \text{ (e conjE)}$$

$$\frac{\Gamma \vdash P}{\Gamma \vdash P \vee Q} \text{ (disjI1)} \qquad \frac{\Gamma \vdash Q}{\Gamma \vdash P \vee Q} \text{ (disjI2)} \qquad \frac{\Gamma, P \vdash R \qquad \Gamma, Q \vdash R}{\Gamma, P \vee Q \vdash R} \text{ (e disjE)}$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \longrightarrow B} \text{ (impI)} \qquad \frac{\Gamma \vdash P \qquad \Gamma, Q \vdash R}{\Gamma, P \longrightarrow Q \vdash R} \text{ (e impE)} \qquad \frac{\Gamma \vdash A \longrightarrow B \qquad \Gamma \vdash A}{\Gamma \vdash B} \text{ (mp)}$$

no right-intro rule for $\bot$

$$\frac{}{\Gamma, \bot \vdash P} \text{ (e FalseE)}$$

$$\frac{\Gamma, P \vdash \bot}{\Gamma \vdash \neg P} \text{ (notI)} \qquad \frac{\Gamma \vdash P}{\Gamma, \neg P \vdash R} \text{ (e notE)} \qquad \frac{}{\Gamma \vdash \neg P \vee P} \text{ (excluded\_middle)}$$

LCF - Logic for Computable fns

Isabelle uses two strategies to maintain soundness:

- A small trusted kernel: internally, every proof is broken down into primitive rule applications which are checked by a small piece of hand-verified code. This is the "LCF" model. So new *proof procedures* cannot introduce unsoundness.
- Encourages *definitional* extension of the logic: new concepts are introduced by definition rather than axiomatisation (more on this in Lecture 6). So new definitions cannot introduce unsoundness.

Threats to (practical) soundness still exist, including: Have we proved what we thought we proved? Are the formulas displayed on screen correctly? ...

Axioms are not forbidden but strongly discouraged in case they are inconsistent (e.g. someone adds a fact tht. "x/x = 1" for all nums)