Mathematics for Computer Scientists 2 (G52MC2)

L02 : Coq basics, propositional logic

Thorsten Altenkirch

School of Computer Science University of Nottingham

October 1, 2009

Propositional logic

- **Proposition**: A statement which can be true or false.
- Coq: P : Prop means P is a proposition.
- Propositional variables: stand for any proposition;
 e.g. in Coq:

Variables P Q R : Prop.

- **Tautology**: A proposition containing propositional variables which is always true.
- Propositional constants True, False.
- Basic propositional connectives:

| Name | Math | Coq | English |
|-------------|-----------------------|---------|--------------------|
| Implication | P ightarrow Q | P -> Q | If P then Q |
| Conjunction | $m{P}\wedgem{Q}$ | P /∖ Q | P and Q |
| Disjunction | $P \lor Q$ | P \/ Q | P or Q |
| Equivalence | $P \leftrightarrow Q$ | P <-> Q | P if and only if Q |
| | | | P iff Q |
| Negation | $\neg P$ | ~ P | not P |

Syntactic conventions

 $\bullet \ \rightarrow \text{ is right-associative, i.e.}$

$$P
ightarrow Q
ightarrow R = P
ightarrow (Q
ightarrow R)$$

• \lor , \land bind stronger than \rightarrow (and \leftrightarrow), i.e.

$$P \lor Q \to R = (P \lor Q) \to R$$

• \land binds stronger than \lor :

$$P \lor Q \land R = P \lor (Q \land R)$$

• \neg binds stronger that \land :

$$\neg P \land Q = (\neg P) \land Q$$

- Start a proof with Lemma or Theorem. Give it a name! E.g.
 Lemma andCom : P /\ Q -> Q /\ P
- Coq displays the *proof state*. The user issues *tactics* until Coq says Proof completed.
- Finish with Qed. Leaves the proof state and saves the proof under the given name.
- Read p : P as p is a proof of P, e.g.

andCom : P /\ Q -> Q /\ P

Coq proof state (example)

2 subgoals H : P /\ Q H1 : P H2 : Q =================== Q subgoal 2 is: P

- Two subgoals: currently proving Q, when we are finished we prove P.
- Assumptions above ===...===.
- Assumptions have names,

e.g. H, H1, H2

- Current goal (e.g. Q) is below the line.
- If current Goal = one of the assumptions, use exact, e.g. exact H2.

General pattern

premise (what we need to show) conclusion (what we want to show) name of the rule

- Read proof rules from bottom to top!
- We write Γ ⊢ P for From the set of assumptions Γ (Gamma), we can prove P.
- The symbol ⊢ (turnstyle) replaces Coq's ===...===
- Example:

$$\frac{H: P \in \Gamma}{\Gamma \vdash P} \text{ exact } H$$

• Read $H : P \in \Gamma$ as H : P occurs in Γ .

Rules for implication

$$\frac{\Gamma, H: P \vdash Q}{\Gamma \vdash P \rightarrow Q} \text{ intro } H \qquad \frac{H: P \rightarrow Q \in \Gamma}{\Gamma \vdash P} \text{ apply } H$$

- intro: to prove $P \rightarrow Q$, assume P and prove Q.
- apply: If we know P → Q then to prove Q it is enough to prove P.
- The actual behaviour of apply is more subtle!
- See examples in I01.v

Rules for conjunction

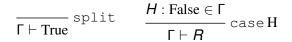
$$\frac{\Gamma \vdash P \quad \Gamma \vdash Q}{\Gamma \vdash P \land Q} \text{ split } \frac{H : P \land Q \in \Gamma}{\Gamma \vdash P \to Q \to R} \text{ elim } H$$

- split: to prove $P \land Q$ prove P and then Q.
- elim: If we know $P \land Q$ then to prove R it is enough to prove $P \rightarrow Q \rightarrow R$.
- See examples in I02.v

Rules for disjunction

$$\frac{\Gamma \vdash P}{\Gamma \vdash P \lor Q} \text{left} \frac{\Gamma \vdash Q}{\Gamma \vdash P \lor Q} \text{right} \frac{H : P \lor Q \in \Gamma}{\Gamma \vdash P \to R \quad \Gamma \vdash Q \to R} \text{case } H$$

- left: to prove *P* ∨ *Q* prove *P*.
- right: to prove $P \lor Q$ prove Q.
- case: If we know $P \lor Q$ then to prove R it is enough to prove $P \to R$ and $Q \to R$.
- See examples in I02.v



- split: to prove True you need to prove nothing.
- case: if you know False you can prove anything.

| connective | Introduction | Elimination |
|----------------|--------------|------------------|
| P ightarrow Q | intro(s) | apply Hyp |
| $P \wedge Q$ | split | elim <i>Hyp</i> |
| True | split | |
| $P \lor Q$ | left,right | case Hyp |
| False | | case Hyp |

