COSC 4P42 - Cheat Sheet

Natural deduction rules and Coq implementation

$$\frac{\varphi \quad \psi}{\varphi \wedge \psi}\ \wedge\mathrm{I}$$

`And_Intro.`
Replaces the current goal $A \wedge B$ by the two goals A and B.

$$\frac{\varphi \wedge \psi}{\varphi}\ \wedge\mathrm{E1}$$

`And_Elim_1 in H.`
Applies to an assumption of the form `H :` $A \wedge B$ and generates a new assumption `H0 :` A

$$\frac{\varphi \wedge \psi}{\psi}\ \wedge\mathrm{E2}$$

`And_Elim_2 in H.`
Applies to an assumption of the form `H :` $A \wedge B$ and generates a new assumption `H0 :` B

`And_Elim_all in H.`
Applies to an assumption of the form `H :` $A \wedge B$ and replaces it with the two assumptions `H :` A and `H0 :` B. The tactic is then recusively applied to `H` and `H0`.

$$\frac{\varphi}{\varphi \vee \psi}\ \vee\mathrm{I1}$$

`Or_Intro_1.`
Replaces the current goal $A \vee B$ by the goal A.

$$\frac{\psi}{\varphi \vee \psi}\ \vee\mathrm{I2}$$

`Or_Intro_2.`
Replaces the current goal $A \vee B$ by the goal B.

$$\frac{\varphi \vee \psi \quad \begin{array}{c}[\varphi]\\ \vdots \\ \chi\end{array} \quad \begin{array}{c}[\psi]\\ \vdots \\ \chi\end{array}}{\chi}\ \vee\mathrm{E}$$

`Or_Elim in H.`
Applies to an assumption of the form `H :` $A \vee B$. It generates two proof obligations with assumptions `H :` A resp. `H :` B and the current goal.

$$\dfrac{\begin{array}{c}[\varphi]\\ \vdots \\ \psi\end{array}}{\varphi \to \psi}\ \to\mathrm{I}$$

`Impl_Intro.`
Replaces the current goal $A \to B$ by B and adds the assumption `H : ` A.

$$\dfrac{\varphi \to \psi \quad \varphi}{\psi}\ \to\mathrm{E}$$

`Impl_Elim in H and H0.`
Applies to the two assumptions of the form `H :` $A \to B$ and `H0 :` A and adds the new assumption `H1 :` B.

$$\dfrac{\begin{array}{c}[\varphi]\\ \vdots \\ \bot\end{array}}{\neg\varphi}\ \neg\mathrm{I}$$

`Not_Intro.`
Replaces the current goal $\sim A$ by `False` and adds the assumption `H : ` A.

$$\dfrac{\neg\varphi \quad \varphi}{\bot}\ \neg\mathrm{E}$$

`Not_Elim in H and H0.`
Applies to the two assumptions of the form `H :` $\sim A$ and `H0 :` A and adds the new assumption `H1 : False`.

$$\dfrac{\begin{array}{c}[\neg\varphi]\\ \vdots \\ \bot\end{array}}{\varphi}\ \mathrm{PBC}$$

`PBC.`
Replaces the current goal A by `False` and adds the assumption `H :` $\sim A$.

$$\dfrac{\varphi}{\forall x{:}\varphi}\ \forall\mathrm{I}$$ if x does not occur free in any premises of this subtree

`Forall_Intro.`
Replaces the current goal `forall` x, A by A and adds the variable $x : A$ to the assumptions.

$$\dfrac{\forall x{:}\varphi}{\varphi[t/x]}\ \forall\mathrm{E}$$

`Forall_Elim in H with t.`
Applies to an assumption of the form `H :` `forall` x, A. It generates a new assumption `H0 :` $A[t/x]$.

$$\dfrac{\varphi[t/x]}{\exists x{:}\varphi}\ \exists\mathrm{I}$$

`Exists_Intro with t.`
Replaces the current goal `exists` x, A by $A[t/x]$.

$$\dfrac{\exists x{:}\varphi \quad \begin{matrix}[\varphi] \\ \vdots \\ \chi\end{matrix}}{\chi}\ \exists E$$

if x does not occur free in χ and in any premises of the right subtree accept φ

`Exists_Elim in H.`
Applies to an assumption of the form `H : exists` x, A. It adds the variable $x : A$ and the new assumption `H0 :` A.

Hoare rules and Coq implementation

(Skip)

$\{\varphi\}\texttt{skip}\{\varphi\}$

`Hoare_skip_rule.`
Applies to a goal of the form {{ `A` }} `Skip` {{ `A` }}. It solves the goal.

(Assignment)

$\{\psi[a/x]\}x := a\{\psi\}$

`Hoare_assignment_rule.`
Applies to a goal of the form {{ `A[t/x]` }} `x ::= t` {{ `A` }}. It solves the goal.

(Sequencing)

$$\frac{\{\varphi\}c_0\{\chi\} \quad \{\chi\}c_1\{\psi\}}{\{\varphi\}c_0; c_1\{\psi\}}$$

`Hoare_sequence_rule with B.`
Applies to a goal of the form {{ `A` }} $\texttt{c}_0$`;;`$\texttt{c}_1$ {{ `C` }} and replaces it by the two goals {{ `A` }} $\texttt{c}_0$ {{ `B` }} and {{ `B` }} $\texttt{c}_1$ {{ `C` }}.

(Conditional)

$$\frac{\{\varphi \wedge b\}c_0\{\psi\} \quad \{\varphi \wedge \neg b\}c_1\{\psi\}}{\{\varphi\}\texttt{if } b \texttt{ then } c_0 \texttt{ else } c_1 \texttt{ fi}\{\psi\}}$$

`Hoare_if_rule.`
Applies to a goal of the form {{ `A` }} `If b Then` $\texttt{c}_0$ `Else` $\texttt{c}_1$ `Fi` {{ `B` }} and replaces it by the two goals {{ `A` ∧ `b = true` }} $\texttt{c}_0$ {{ `B` }} and {{ `A` ∧ `b = false` }} $\texttt{c}_1$ {{ `C` }}.

(Loop)

$$\frac{\{\varphi \wedge b\}c\{\varphi\}}{\{\varphi\}\texttt{while } b \texttt{ do } c \texttt{ od}\{\varphi \wedge \neg b\}}$$

`Hoare_while_rule.`
Applies to a goal of the form {{ `I` }} `While b Do c Od` {{ `I` ∧ `b = false` }} and replaces it by {{ `I` ∧ `b = true` }} `c` {{ `I` }}.

(Consequence)

$$\frac{\models \varphi \rightarrow \varphi' \quad \{\varphi'\}c\{\psi'\} \quad \models \psi' \rightarrow \psi}{\{\varphi\}c\{\psi\}}$$

`Hoare_consequence_rule with A' and B'.`
Applies to a goal of the form `{{ A }} c {{ B }}` and replaces it by the three goals `A → A'`, `{{ A' }} c {{ B' }}`, and `B' → B`.

`Hoare_consequence_rule_left with A'.`
Identical to `Hoare_consequence_rule with A' and B.` Just two new goals are generated.

`Hoare_consequence_rule_right with B'.`
Identical to `Hoare_consequence_rule with A and B'.` Just two new goals are generated.

Additional Hoare Tactics

`Hoare_tactic.` Applies the rules (Skip), (Assignment), and (Conditional) starting at the end of the program using the rules (Sequencing) and (Consequence) and the weakest pre-condition approach. Stops when it encounters a loop.

`Hoare_while_tactic with I.` Works like `Hoare_tactic.` but can handle one loop at the top level of the program (i.e. a loop that is not within an if-statement). When it encounters a loop it uses `I` as the invariant.