Handling of Loops

by Christoph Gladisch

Induction rule:

$$\frac{\overbrace{\Gamma \Rightarrow H(0)}^{\text{Base case}} \qquad \overbrace{\Gamma, i \geqslant 0, H(i) \Rightarrow H(i+1)}^{\text{Step case}} \qquad \overbrace{\Gamma, \forall i. H(i) \Rightarrow \varphi}^{\text{Use case}}}{\Gamma \Rightarrow \varphi}$$

In order to prove φ find a more general formula $H(i)$ such that it can be instantiated and $\forall i.H(i) \Rightarrow \varphi$ can be proven. φ is often an instance of $H(i)$ so that for some term t it even holds that $H(t) \equiv \varphi$.

Loop induction. Heuristics for finding an induction hypothesis:

For a formula φ:

$\varphi = \quad \{U\}\langle\texttt{while(c)\{body\}}\rangle\text{Post}$

the induction hypothesis $H(i)$ has often the form:

$H(i) = \quad \{U\}\underbrace{\{V(i)\}(\text{HPre}(i)}_{\text{Prefix}} \rightarrow \underbrace{\langle\texttt{while(c)\{body\}}\rangle\text{Post}}_{\text{Postfix}})$

Note the similarity to the original formula. The postfix $\langle\texttt{while(c)\{body\}}\rangle$Post remains unchanged. Only the update $\{V(i)\}$ and the formula HPre(i) have to be choosen appropriately. The prefix $\{V(i)\}$(HPre(i) determins the state before the execution of the loop. $\{V(i)\}$ assigns program variables to the right state and HPre(i) puts additional constrains on the state to filter out for instance invalid ranges etc.

HPre(i) has to be choosen such that:

(1) HPre(0) $\rightarrow (c \equiv \text{false})$ the loop terminates and

(2) HPre(0) $\rightarrow$ Post

Because of (2) HPre(0) and Post are very similar. Sometimes it is that HPre(0) $\equiv$ Post.

(3) HPre(n) describes the state before the execution of the loop.

Figure 5. illustrates the states of the prefix before the loop is entered and for the case when the loop iterates 0 times.

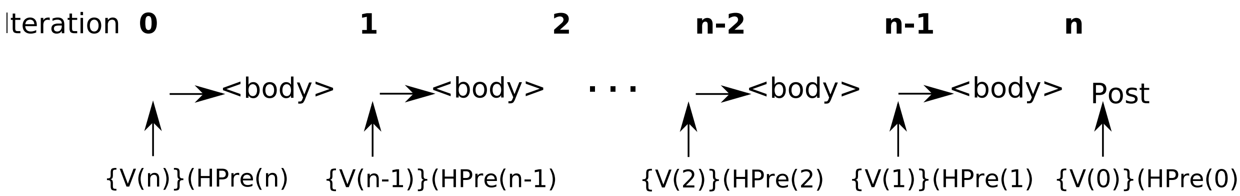

Figure 5. Visualisation of the semantics of the Prefix $\{V(i)\}$(HPre(i) of the induction hypothesis.

Figure 6. Illustrates what happens at the induction step.

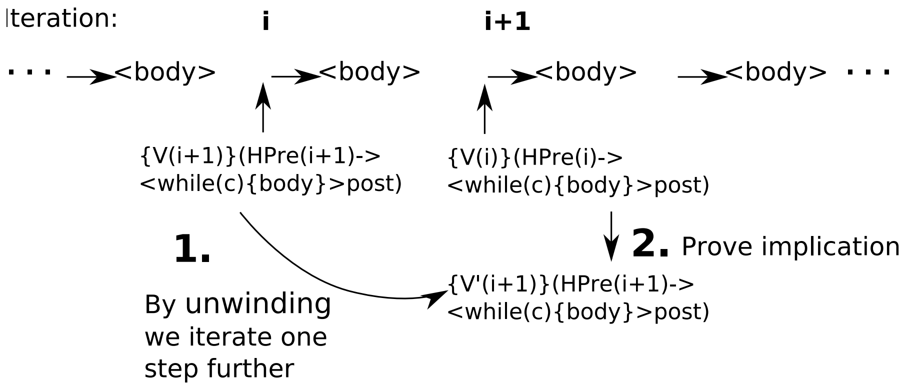

Figure 6. Induction step

A frequent pattern in the step case:

$$\dfrac{\dfrac{\dfrac{C \Rightarrow A}{C \Rightarrow A, D} \qquad \dfrac{B \Rightarrow D}{B, C \Rightarrow D}}{A \to B, C \Rightarrow D}}{A \to B \Rightarrow C \to D}$$

With induction hypothesis pattern:

$$\dfrac{\dfrac{\dfrac{\underbrace{\mathrm{HPre}(i+1)}_{C} \Rightarrow \underbrace{\mathrm{HPre}(i)}_{A}}{\underbrace{\mathrm{HPre}(i+1)}_{C} \Rightarrow \underbrace{\mathrm{HPre}(i)}_{A}} \qquad \dfrac{\underbrace{\{U(i)\}\langle\texttt{while(c)\{body\}}\rangle\mathrm{Post}}_{B}, \underbrace{\mathrm{HPre}(i+1)}_{C} \Rightarrow \underbrace{\{U(i+1)\}\langle\texttt{while(c)\{body\}}\rangle\mathrm{Post}}_{D}}{\underbrace{\{U(i)\}\langle\texttt{while(c)\{body\}}\rangle\mathrm{Post}}_{B}, \underbrace{\mathrm{HPre}(i+1)}_{C} \Rightarrow \underbrace{\{U(i+1)\}\langle\texttt{while(c)\{body\}}\rangle\mathrm{Post}}_{D}}}{\underbrace{\mathrm{HPre}(i)}_{A} \to \underbrace{\{U(i)\}\langle\texttt{while(c)\{body\}}\rangle\mathrm{Post}}_{B}, \underbrace{\mathrm{HPre}(i+1)}_{C} \Rightarrow \underbrace{\{U(i+1)\}\langle\texttt{while(c)\{body\}}\rangle\mathrm{Post}}_{D}}}{\underbrace{\mathrm{HPre}(i)}_{A} \to \underbrace{\{U(i)\}\langle\texttt{while(c)\{body\}}\rangle\mathrm{Post}}_{B} \Rightarrow \underbrace{\mathrm{HPre}(i+1)}_{C} \to \underbrace{\{U(i+1)\}\langle\texttt{while(c)\{body\}}\rangle\mathrm{Post}}_{D}}$$

Frequent problem after unwinding the loop of the right formula in the induction step:
$\{i := x \| n := y\}\langle\texttt{while(c)\{..m=z..\}}\rangle P \Rightarrow \{i := x \| \boldsymbol{m := z} \| n := y\}\langle\texttt{while(c)\{..m=z..\}}\rangle P$
Solution: Generalise the induction hypothesis in the form:
$\forall \boldsymbol{M}.\{i := x \| n := y\}\boldsymbol{\{m := M\}}(\mathrm{HPre} \to \langle\texttt{while(c)\{..m=z..\}}\rangle P)$

Classic Invariant rule:

$$\frac{\Gamma \Rightarrow \{U\}\mathrm{inv} \qquad \mathrm{inv}, c \Rightarrow [\texttt{body}]\mathrm{inv} \qquad \mathrm{inv}, \neg c \Rightarrow \mathrm{Post}}{\Gamma \Rightarrow \{U\}[\texttt{while(c)\{body\}}]\mathrm{Post}}$$

In KeY without modifies clause:

$$\frac{\Gamma \Rightarrow \{U\}\mathrm{inv} \qquad \Rightarrow \mathrm{inv} \to ([\texttt{b=c}]b \to [\texttt{body}]\mathrm{inv}) \qquad \Rightarrow \mathrm{inv} \to \neg c \to \mathrm{Post}}{\Gamma \Rightarrow \{U\}[\texttt{while(c)\{body\}}]\mathrm{Post}}$$

In KeY with modifies clause:

$$\frac{\Gamma \Rightarrow \{U\}\mathrm{inv} \qquad \Gamma \Rightarrow \{U\}\{M\}(\mathrm{inv} \to ([\texttt{b=c}]b \to [\texttt{body}]\mathrm{inv})) \qquad \Gamma \Rightarrow \{U\}\{M\}(\mathrm{inv} \to \neg c \to \mathrm{Post})}{\Gamma \Rightarrow \{U\}[\texttt{while(c)\{body\}}]\mathrm{Post}}$$

For every program variable in the modifies clause an update {M} is created that replaces the modifies program variable by a fresh program variable or in other words by a new skolem function.