Cálculo de Correção parcial $\mathcal{H}$

$[skip_p\ ]$

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

$[ass_p\ ]$

$$\{\varphi[E/x]\}\,x \leftarrow E\,\{\varphi\}$$

$[comp_p\ ]$

$$\frac{\{\varphi\}\,C_1\,\{\eta\} \qquad \{\eta\}\,C_2\,\{\psi\}}{\{\varphi\}\,C_1; C_2\,\{\psi\}}$$

$[if_p\ ]$

$$\frac{\{\varphi\ \wedge\ B\}\,C_1\,\{\psi\} \qquad \{\varphi\ \wedge\ \neg B\}\,C_2\,\{\psi\}}{\{\varphi\}\,\texttt{if}\,B\,\texttt{then}\,C_1\,\texttt{else}\,C_2\,\{\psi\}}$$

$[if'_p\ ]$

$$\frac{\{\varphi_1\}\,C_1\,\{\psi\} \qquad \{\varphi_2\}\,C_2\,\{\psi\}}{\{(B \to \varphi_1)\ \wedge\ (\neg B \to \varphi_2)\}\,\texttt{if}\,B\,\texttt{then}\,C_1\,\texttt{else}\,C_2\,\{\psi\}}$$

$[while_p\ ]$

$$\frac{\{\psi\ \wedge\ B\}\,C\,\{\psi\}}{\{\psi\}\,\texttt{while}\,B\,\texttt{do}\,C\,\{\psi\ \wedge\ \neg B\}}$$

$[cons_p\ ]$

$$\frac{\vdash \varphi' \to \varphi \qquad \{\varphi\}\,C\,\{\psi\} \qquad \vdash \psi \to \psi'}{\{\varphi'\}\,C\,\{\psi'\}}$$

Exemplos

Exercício 4.1. *Mostrar que*

$$\begin{aligned} &\vdash_p \{\mathsf{true}\} \\ &r \leftarrow x; q \leftarrow 0; \\ &\texttt{while}\, y \leq r\, \texttt{do} \\ &\quad r \leftarrow r - y; \\ &\quad q \leftarrow q + 1 \\ &\{r < y\ \wedge\ x = r + (y \times q)\} \end{aligned}$$

$\diamond$

A expressão $x = r + (y \times q)$ é um invariante de ciclo.

Cálculo para a correcção total

Na linguagem imperativa apresentada, o único comando que pode levar à não terminação é o comando `while`.

O *cálculo* $\vdash_{tot}$ irá ser igual ao $\vdash_p$ excepto na regra $\texttt{while}_{tot}$.

Para demonstrar que um programa termina temos que lhe associar uma expressão estritamente decrescente, denominada variante.

No caso do `while`, podemos associar uma expressão inteira não negativa e mostrar que em cada iteração o valor dessa expressão diminui, mantendo-se não negativa: temos a certeza que `while` termina pois essa expressão só pode tomar um número finito de valores até chegar a zero!!!

No caso do factorial:

$y \leftarrow 1; z \leftarrow 0; \texttt{while}\ z \neq x\ do\ (z \leftarrow z+1; y \leftarrow y \times z)$

podemos tomar o variante $x - z$.

Cálculo para a correcção total

Lógica de Hoare (correcção total)

As regras ass_{tot}, $comp_{tot}$, if_{tot} e $cons_{tot}$ coincidem com as do *cálculo* para a correcção parcial.

$[while_{tot}\ ]$

$$\frac{\{\eta \wedge B \wedge E \geq 0 \wedge E = e_0\}\, C\, \{\eta \wedge E \geq 0 \wedge E < e_0\}}{\{\eta \wedge E \geq 0\}\, \texttt{while}\, B\, \texttt{do}\, C\, \{\eta \wedge \neg B\}}$$

onde e_0 é uma variável lógica cujo valor é o da expressão E antes da execução do comando C.

Pré condição mais fraca - $while_{tot}$

$\{\varphi\}$
$\{\eta \wedge E \geq 0\}$
$\texttt{while}\, B\, \texttt{do}$
$\{\eta \wedge B \wedge E \geq 0 \wedge E = e_0\}$
C
$\{\eta \wedge E \geq 0 \wedge E < e_0\}$
$\{\eta \wedge \neg B\}$ $while_{tot}$
$\{\psi\}$ $cons_{tot}$

Exemplo

$\vdash_{tot} \{x \geq 0\} y \leftarrow 1; z \leftarrow 0; \texttt{while}\ z \neq x\ do\ (z \leftarrow z+1; y \leftarrow y \times z) \{y = x!\}$

$\{x \geq 0\}$

$\{1 = 0! \wedge x - 0 \geq 0\}$

$\texttt{y} \leftarrow \texttt{1}$

$\{y = 0! \wedge x - 0 \geq 0\}$

$\texttt{z} \leftarrow \texttt{0}$

$\{y = z! \wedge x - z \geq 0\}$ ass_{tot}

$\texttt{while}\ z \neq x\ do$

$\{$

$\{y = z! \wedge z \neq x \wedge x - z \geq 0 \wedge x - z = e_0\}$ $cons_{tot}$

$\{y \times (z+1) = (z+1)! \wedge x - (z+1) \geq 0 \wedge x - (z+1) < e_0\}$ ass_{tot}

$\texttt{z} \leftarrow \texttt{z} + \texttt{1}$

$\{y \times z = z! \wedge x - z \geq 0 \wedge x - z < e_0\}$ ass_{tot}

$\texttt{y} \leftarrow \texttt{y} \times \texttt{z}$

$\{y = z! \wedge x - z \geq 0 \wedge x - z < e_0\}$

$\}$

$\{y = z! \wedge x = z\}$

$\{y = x!\}$

Como determinar um variante ?

Os variantes são mais difíceis de encontrar que os invariantes...porque não é possível saber genericamente se um programa termina

$\vdash_{tot} \{x > 0\}$

$\texttt{c} = \texttt{x}$

$\texttt{while}(\texttt{c}! = \texttt{1})\texttt{do}$

$\texttt{if}(\texttt{c}\%\texttt{2} == \texttt{0})\texttt{c} = \texttt{c}/\texttt{2}$

$\texttt{else}\ \texttt{c} = \texttt{3} * \texttt{c} + \texttt{1}$

$\{\top\}$

Será que este triplo é válido? Neste caso este triplo só estabelecia a terminação do programa...

Mas não se sabe se termina ou não!

Exercício 4.2. *Mostrar que*

$$\begin{aligned} &\vdash_{tot} \{\neg y = 0\} \\ &r \leftarrow x; q \leftarrow 0; \\ &\texttt{while}\, y \leq r\, \texttt{do} \\ &\quad r \leftarrow r - y; \\ &\quad q \leftarrow q + 1 \\ &\{r < y \ \wedge \ x = r + (y \times q)\} \end{aligned}$$

$\diamond$

Exercício 4.3. *Mostra que*

$$\begin{aligned} &\vdash_{tot} \{y > 0\} \\ &\texttt{while}\, y \leq r\, \texttt{do} \\ &\quad r \leftarrow r - y; \\ &\quad q \leftarrow q + 1 \\ &\quad \{true\} \end{aligned}$$

$\diamond$

Mecanização da construção de derivações na lógica de Hoare

De um modo geral, dado um triplo de Hoare ($\{\varphi\}C\{\psi\}$) aplicamos as regras a partir da conclusão, assumindo que as condições auxiliares se verificam.

- Se todas as condições auxiliares se verificarem então construímos uma demonstração;
- Se alguma das condições auxiliares não se verifica, a árvore construída não constitui uma dedução válida, mas será possível construir uma outra árvore que o seja?

Existe uma estratégia para construir as árvores de forma a poder concluir (caso algumas das condições auxiliares não se verifique) que não existe uma derivação para o triplo dado.

Mecanização da lógica de Hoare

A maior parte das regras do cálculo de Hoare têm a *propriedade de sub-fórmula:*

> *todas as asserções que ocorrem nas premissas de uma regra também ocorrem na sua conclusão.*

As excepções são:

- A regra *comp*, que requer uma condição intermédia;
- A regra *cons*, onde a pré-condição e a pós-condição têm que ser "adivinhadas".

Outra propriedade desejável é que não seja ambígua a escolha das regras:

- A regra *cons*, pode ser aplicada para qualquer triplo de Hoare.

Versão da lógica de Hoare sem *cons*: sistema $\mathcal{H}_g$

$$\frac{}{\{\varphi\}\,\mathsf{skip}\,\{\psi\}} \text{ se } \models \varphi \to \psi$$

$$\frac{}{\{\varphi\}\,x \leftarrow E\,\{\psi\}} \text{ se } \models \varphi \to \psi[E/x]$$

$$\frac{\{\varphi\}\,C_1\,\{\eta\} \qquad \{\eta\}\,C_2\,\{\psi\}}{\{\varphi\}\,C_1; C_2\,\{\psi\}}$$

$$\frac{\{\varphi \ \wedge \ B\}\,C_1\,\{\psi\} \qquad \{\varphi \ \wedge \ \neg B\}\,C_2\,\{\psi\}}{\{\varphi\}\,\texttt{if}\,B\,\texttt{then}\,C_1\,\texttt{else}\,C_2\,\{\psi\}}$$

$$\frac{\{\eta \ \wedge \ B\}\,C\,\{\eta\}}{\{\varphi\}\,\texttt{while}\,B\,\texttt{do}\,\{\eta\}C\,\{\psi\}} \text{ se } \models \varphi \to \eta \text{ e } \models \eta \ \wedge \ \neg B \to \psi$$

Sistema $\mathcal{H}_g$

É fácil de demonstrar que a regra *cons* é derivável em $\mathcal{H}_g$.

Lema 4.1. *Se* $\Gamma \vdash_{\mathcal{H}_g} \{\varphi\}C\{\psi\}$ *e* $\models \varphi' \to \varphi$, $\models \psi \to \psi'$, *então* $\Gamma \vdash_{\mathcal{H}_g} \{\varphi'\}C\{\psi'\}$.

Demonstração: Por indução sobre a derivação $\Gamma \vdash_{\mathcal{H}_g} \{\psi\}C\{\varphi\}$. Vamos ver os casos para o `skip` e para a sequência.

- Para $C \equiv \texttt{skip}$, temos $\Gamma \vdash_{\mathcal{H}_g} \{\varphi\}\texttt{skip}\{\psi\}$, se $\models \varphi \to \psi$. Temos $\models \varphi' \to \varphi$, $\models \varphi \to \psi$ e $\models \psi \to \psi'$, logo $\models \varphi' \to \psi'$, o que significa que temos $\Gamma \vdash_{\mathcal{H}_g} \{\varphi'\}\texttt{skip}\{\psi'\}$.
- Para $C \equiv C_1; C_2$, temos $\Gamma \vdash_{\mathcal{H}_g} \{\varphi\}C_1; C_2\{\psi\}$, se $\Gamma \vdash_{\mathcal{H}_g} \{\varphi\}C_1\{\eta\}$ e $\Gamma \vdash_{\mathcal{H}_g} \{\eta\}C_2\{\psi\}$. Mas então por H.I. temos $\Gamma \vdash_{\mathcal{H}_g} \{\varphi'\}C_1\{\eta\}$ (uma vez que $\models \varphi' \to \varphi$ e $\models \eta \to \eta$) e $\Gamma \vdash_{\mathcal{H}_g} \{\eta\}C_2\{\psi'\}$ (uma vez que $\models \eta \to \eta$ e $\models \psi \to \psi'$), logo $\Gamma \vdash_{\mathcal{H}_g} \{\varphi'\}C_1; C_2\{\psi'\}$.

Exercício 4.4. *Completa a demonstração anterior.*

Equivalência $\mathcal{H}$ e $\mathcal{H}_g$

$\Gamma \vdash_{\mathcal{H}} \{\varphi\}C\{\psi\}$ se e só se $\Gamma \vdash_{\mathcal{H}_g} \{\varphi\}C\{\psi\}$

($\Rightarrow$) Por indução sobre a derivação $\Gamma \vdash_{\mathcal{H}} \{\psi\}C\{\varphi\}$, usando o lema anterior. Vamos ver os casos para atribuição e para a regra da consequência.

- Temos $\Gamma \vdash_{\mathcal{H}} \{\varphi[E/x]\}x \leftarrow E\{\varphi\}$ e $\models \varphi[E/x] \rightarrow \varphi[E/x]$, logo $\Gamma \vdash_{\mathcal{H}_g} \{\varphi[E/x]\}x \leftarrow E\{\varphi\}$
- Pela regra da consequência temos $\Gamma \vdash_{\mathcal{H}} \{\varphi\}C\{\psi\}$, se $\Gamma \vdash_{\mathcal{H}} \{\varphi'\}C\{\psi'\}$ e $\models \varphi \rightarrow \varphi'$, $\models \psi' \rightarrow \psi$.
  Por H.I. temos $\Gamma \vdash_{\mathcal{H}_g} \{\varphi'\}C\{\psi'\}$, logo pelo lema anterior temos $\Gamma \vdash_{\mathcal{H}_g} \{\varphi\}C\{\psi\}$.

($\Leftarrow$) Por indução sobre a derivação $\Gamma \vdash_{\mathcal{H}_g} \{\psi\}C\{\varphi\}$. Vamos ver os casos para a atribuição e para o condicional.

- Temos $\Gamma \vdash_{\mathcal{H}_g} \{\psi\}x \leftarrow E\{\varphi\}$ se $\models \psi \rightarrow \varphi[E/x]$. Como $\Gamma \vdash_{\mathcal{H}} \{\varphi[E/x]\}x \leftarrow E\{\varphi\}$ e $\models \psi \rightarrow \varphi[E/x]$ e $\models \psi \rightarrow \psi$, então pela regra da consequência, temos $\Gamma \vdash_{\mathcal{H}} \{\psi\}x \leftarrow E\{\varphi\}$.
- Temos $\Gamma \vdash_{\mathcal{H}_g} \{\psi\}\texttt{if}\, B\, \texttt{then}\, C_1\, \texttt{else}\, C_2\, \{\varphi\}$, se $\Gamma \vdash_{\mathcal{H}_g} \{\psi \wedge B\}C_1\{\varphi\}$ e $\Gamma \vdash_{\mathcal{H}_g} \{\psi \wedge \neg B\}C_2\{\varphi\}$.
  Por H.I. $\Gamma \vdash_{\mathcal{H}} \{\psi \wedge B\}C_1\{\varphi\}$ e $\Gamma \vdash_{\mathcal{H}} \{\psi \wedge \neg B\}C_2\{\varphi\}$, logo $\Gamma \vdash_{\mathcal{H}} \{\psi\}\texttt{if}\, B\, \texttt{then}\, C_1\, \texttt{else}\, C_2\, \{\varphi\}$

Exercício 4.5. *Completa a demonstração anterior.*

Pós e Contras

Vantagens de $\mathcal{H}_g$:

- Eliminamos a ambiguidade provocada pela regra *cons*.
- Eliminamos uma das regras sem a propriedade de sub-fórmula.

No entanto, ainda é necessário "adivinhar" pré-condições intermédias para *comp*.

A estratégia de pré-condição mais fraca

Queremos construir uma derivação para um triplo de Hoare $\{\varphi\}C\{\psi\}$, onde φ pode ou não ser conhecido (nesse caso escrevemos $\{?\}C\{\psi\}$).

1. Se φ for conhecido, então aplicamos a única regra possível de $\mathcal{H}_g$. Se C for $C_1; C_2$, então construímos uma sub-derivação da forma $\{?\}C_2\{\psi\}$. Eventualmente quando concluírmos esta derivação podemos prosseguir com $\{\varphi\}C_1\{\theta\}$, com θ obtido da sub-derivação anterior.

2. Se φ é desconhecido, a construção procede da mesma forma, excepto que no caso das regras skip, atribuição e ciclos, com uma condição auxiliar $\varphi \to \theta$, tomamos a pré-condição φ como θ.

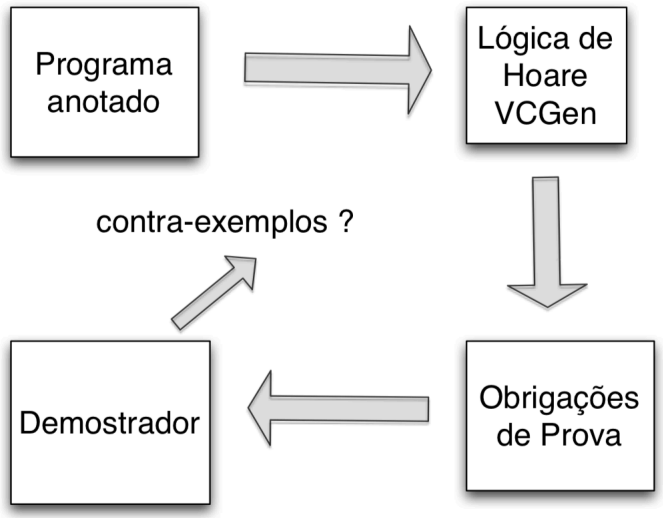
Uma Arquitectura para Verificação de Programas

Dado um triplo de Hoare $\{\varphi\}C\{\psi\}$ e uma teoria $\mathcal{T}$:

1. Aplicamos os princípios apresentados anteriormente para construir uma derivação com conclusão $\{\varphi\}C\{\psi\}$, assumindo que todas as condições auxiliares geradas no processo se verificam.
2. Cada fórmula de primeira ordem gerada como condição auxiliar (chamada neste contexto de *condição de verificação* (VC)) tem que ser verificada numa ferramenta de prova.
3. Se todas as condições de verificação são classificadas como $\mathcal{T}$-válidas, então $\mathcal{T} \vdash_{\mathcal{H}_g} \{\varphi\}C\{\psi\}$.

Nota: como não existe ambiguidade na construção das árvores, podemos eliminar essa parte do processo e simplesmente gerar as VC usando um *Gerador de condições de verificação*(VCGen).

Duas fases para a verificação



Um algoritmo VCGen: cálculo das pré-condições mais fracas (wp)

Dado um programa C e uma pós-condição ψ, podemos calcular $wp(C, \psi)$ tal que $\{wp(C, \psi)\}C\{\psi\}$ é válida e se $\{\varphi\}C\{\psi\}$ é válida para algum φ então $\varphi \to wp(C, \psi)$.

$$\begin{array}{rcl}
wp(\textbf{skip}, \psi) & = & \psi \\
wp(x \leftarrow E, \psi) & = & \psi[E/x] \\
wp(C_1; C_2, \psi) & = & wp(C_1, wp(C_2, \psi)) \\
wp(\texttt{if}\, B\, \textbf{then}\, C_1\, \textbf{else}\, C_2, \psi) & = & (B \to wp(C_1, \psi)) \\
 & & \wedge(\neg B \to wp(C_2, \psi)) \\
wp(\textbf{while}\, B\, \textbf{do}\, \{\eta\}C, \psi) & = & \eta
\end{array}$$

Algoritmo VCGen

Primeiro calcula as VC não considerando as pré-condições

$$\begin{array}{rcl}
VC(\texttt{skip}, \psi) & = & \emptyset \\
VC(x \leftarrow E, \psi) & = & \emptyset \\
VC(C_1; C_2, \psi) & = & VC(C_1, wp(C_2, \psi)) \cup VC(C_2, \psi) \\
VC(\texttt{if}\, B\, \texttt{then}\, C_1\, \texttt{else}\, C_2, \psi) & = & VC(C_1, \psi) \cup VC(C_2, \psi) \\
VC(\texttt{while}\, B\, \texttt{do}\, \{\eta\}C, \psi) & = & \{(\eta \ \wedge \ B) \to wp(C, \eta)\} \cup \\
 & & \{(\eta \ \wedge \ \neg B) \to \psi\} \cup VC(C, \eta)
\end{array}$$

A pré-condição é tomada em consideração:

$$VCG(\{\varphi\}C\{\psi\}) = \{\varphi \to wp(C, \psi)\} \cup VC(C, \psi)$$

Exemplo

Seja fact o seguinte programa:

$f \leftarrow 1; i \leftarrow 1;$
while $i \leq n$ **do**
  $\{f = (i-1)! \wedge i \leq n+1\}$ ▷ Invariante
  $f \leftarrow f * i;$
  $i \leftarrow i + 1;$

Vamos calcular

$$\mathrm{VCG}(\{n \geq 0\}\mathsf{fact}\{f = n!\})$$

com

$$\begin{array}{rcl} \theta & = & f = (i-1)! \wedge i \leq n+1 \\ C_w & = & f \leftarrow f * i; i \leftarrow i+1 \end{array}$$

$$\begin{array}{rcl}
 & & VC(\mathtt{fact}, f = n!) \\
 & = & VC(f \leftarrow 1; i \leftarrow 1, wp(\mathbf{while}\ i \leq n\ \mathbf{do}\{\theta\}C_w, f = n!)) \\
 & & \cup VC(\mathbf{while}\ i \leq n\ \mathtt{do}\{\theta\}C_w, f = n!) \\
 & = & VC(f \leftarrow 1; i \leftarrow 1, \theta) \cup \{\theta \wedge i \leq n \rightarrow wp(C_w, \theta)\} \\
 & & \cup \{\theta \wedge i > n \rightarrow f = n!\} \cup VC(C_w, \theta) \\
 & = & VC(f \leftarrow 1, wp(i \leftarrow 1, \theta)) \cup VC(i \leftarrow 1, \theta) \\
 & & \cup \{f = (i-1)! \wedge i \leq n+1 \wedge i \leq n \rightarrow wp(f \leftarrow f * i; i \leftarrow i+1, \theta)\} \\
 & & \cup \{f = (i-1)! \wedge i \leq n+1 \wedge i > n \rightarrow f = n!\} \\
 & & \cup VC(f = f * i, wp(i \leftarrow i+1, \theta)) \cup VC(i \leftarrow i+1, \theta) \\
 & = & \emptyset \cup \emptyset \cup \{f = (i-1)! \wedge i \leq n+1 \wedge i \leq n \\
 & & \qquad \rightarrow wp(f \leftarrow f * i, f = (i+1-1)! \wedge i+1 \leq n+1)\} \\
 & & \cup \{f = (i-1)! \wedge i \leq n+1 \wedge i \leq n \rightarrow f = n!\} \cup \emptyset \cup \emptyset \\
 & = & \{f = (i-1)! \wedge i \leq n+1 \wedge i \leq n \rightarrow f * i = (i+1-1)! \\
 & & \wedge\, i+1 \leq n+1, f = (i-1)! \wedge i \leq n+1 \wedge i \leq n \rightarrow f = n!\}
\end{array}$$

$$\begin{array}{rcl}
 & & VCG(\{n \geq 0\}\mathsf{fact}\{f = n!\}) \\
 & = & \{n \geq 0 \rightarrow wp(\mathsf{fact}, f = n!)\} \cup VC(\mathsf{fact}, f = n!) \\
 & = & \{n \geq 0 \rightarrow wp(f \leftarrow 1; i \leftarrow 1; wp(\mathbf{while}\ i \leq n\ \mathbf{do}\{\theta\}C_w, f = n!), \\
 & & f = (i-1)! \wedge i \leq n+1 \wedge i \leq n \rightarrow f * i = (i+1-1)! \\
 & & \wedge i+1 \leq n+1, f = (i-1)! \wedge i \leq n+1 \wedge i \leq n \rightarrow f = n!\} \\
 & = & \{n \geq 0 \rightarrow wp(f \leftarrow 1; i \leftarrow 1; \theta), \\
 & & f = (i-1)! \wedge i \leq n+1 \wedge i \leq n \rightarrow f * i = (i+1-1)! \\
 & & \wedge i+1 \leq n+1, f = (i-1)! \wedge i \leq n+1 \wedge i \leq n \rightarrow f = n!\}
\end{array}$$

Chegamos às seguintes obrigações de prova:

1. $n \geq 0 \rightarrow 1 = (1-1)! \wedge 1 \leq n+1$
2. $f = (i-1)! \wedge i \leq n+1 \wedge i \leq n \rightarrow f * i = (i+1-1)! \wedge i+1 \leq n+1)$
3. $f = (i-1)! \wedge i \leq n+1 \wedge i \leq n \rightarrow f = n!$

Propriedades de wp e VCG

Dado um comando C e uma asserção ψ se $\Gamma \vdash_{\mathcal{H}_g} \{\varphi\}C\{\psi\}$, para alguma pré-condição φ, então

1. $\Gamma \vdash_{\mathcal{H}_g} \{wp(C,\psi)\}C\{\psi\}$
2. $\Gamma \models \varphi \to wp(C,\psi)$

Demonstração: Por indução sobre C. Vamos ver os casos de `skip` e `while`.

- Para $C \equiv$ `skip`, temos $\Gamma \vdash_{\mathcal{H}_g} \{\varphi\}\texttt{skip}\{\psi\}$ se $\models \varphi \to \psi$. Note-se que $wp(\texttt{skip},\psi) = \psi$.
  1. Trivialmente temos $\Gamma \vdash_{\mathcal{H}_g} \{\psi\}\texttt{skip}\{\psi\}$, uma vez que $\models \psi \to \psi$.
  2. Por hipótese temos $\Gamma \models \varphi \to \psi = wp(\texttt{skip},\psi)$.
- $C \equiv$ `while`, temos $\Gamma \vdash_{\mathcal{H}_g} \{\psi\}\,\texttt{while}\,B\,\texttt{do}\,\{\eta\}C\,\{\psi\}$ se $\Gamma \vdash_{\mathcal{H}_g} \{\eta \wedge B\}C\{\eta\}$ e $\models \psi \to \eta$, $\models \eta \wedge \neg B \to \psi$. Note-se que $wp(\texttt{while}\,B\,\texttt{do}\,\{\eta\}C,\psi) = \eta$
  1. Como $\models \eta \to \eta$, e por hipótese $\models \eta \wedge \neg B \to \psi$ e $\Gamma \vdash_{\mathcal{H}_g} \{\eta \wedge B\}C\{\eta\}$, então $\Gamma \vdash_{\mathcal{H}_g} \{\eta\}\,\texttt{while}\,B\,\texttt{do}\,\{\eta\}C\,\{\psi\}$
  2. Por hipótese temos $\Gamma \models \varphi \to \eta = wp(\texttt{while}\,B\,\texttt{do}\,\{\eta\}C\,\psi)$.

Exercício 4.6. *Completa a demonstração anterior.*

Teorema 4.1 (Adequação de VCGen)**.** *Seja $\{\varphi\}C\{\psi\}$ um triplo de Hoare e Γ um conjunto de asserções.*

$$\Gamma \models VCG(\{\varphi\}C\{\psi\}) \text{ se e só se } \Gamma \vdash_{\mathcal{H}_g} \{\varphi\}C\{\psi\}.$$

($\Rightarrow$) Por indução sobre a derivação C. Vamos ver os casos para atribuição e para a regra da sequência.

- Para $C \equiv x \leftarrow E$, temos $VCG(\{\varphi\}X \leftarrow E\{\psi\}) = \{\varphi \to wp(X \leftarrow E,\psi)\} \cup VC(x \leftarrow E,\psi) = \{\varphi \to \psi[E/x]\}$. Se $\Gamma \models \varphi \to \psi[E/x]$, então pela regra da atribuição $\Gamma \vdash_{\mathcal{H}_g} \{\varphi\}C\{\psi\}$.

Adequação do VCG

- Para $C \equiv C_1; C_2$, temos

$$\begin{aligned} VCG(\{\varphi\}C_1;C_2\{\psi\}) &= \{\varphi \to wp(C_1;C_2,\psi)\} \cup VC(C_1;C_2,\psi) \\ &= \{\varphi \to wp(C_1, wp(C_2,\psi))\} \\ &\quad \cup VC(C_1, wp(C_2,\psi)) \cup VC(C_2,\psi). \end{aligned}$$

Seja $\eta = wp(C_2, \psi)$. Como

$$\Gamma \models \varphi \to wp(C_1, \eta) \cup VC(C_1, \eta) = VCG(\{\varphi\}C_1\{\eta\},$$

por I.H. $\Gamma \vdash_{\mathcal{H}_g} \{\varphi\}C_1\{\eta\}$.
Também $\Gamma \models \eta \to \eta \cup VC(C_2, \psi) = VCG(\{\eta\}C_2\{\psi\})$, por I.H. $\Gamma \vdash_{\mathcal{H}_g} \{\eta\}C_2\{\psi\}$, logo $\Gamma \vdash_{\mathcal{H}_g} \{\varphi\}C_1; C_2\{\psi\}$

($\Leftarrow$) Por indução sobre a derivação $\Gamma \vdash_{\mathcal{H}_g} \{\psi\}C\{\varphi\}$. Vamos ver os casos para o `skip` e para o condicional.

- $\Gamma \vdash_{\mathcal{H}_g} \{\varphi\}\texttt{skip}\{\psi\}$, se $\Gamma \models \varphi \to \psi = VCG(\{\varphi\}\texttt{skip}\{\psi\})$.

Adequação do VCG

- $\Gamma \vdash_{\mathcal{H}_g} \{\varphi\}\texttt{if}\, B\, \texttt{then}\, C_1\, \texttt{else}\, C_2, \{\psi\}$ se $\Gamma \vdash_{\mathcal{H}_g} \{\varphi \wedge B\}C_1\{\psi\}$ e $\Gamma \vdash_{\mathcal{H}_g} \{\varphi \wedge \neg B\}C_2\{\psi\}$. Por H.I.

  $$\Gamma \models VCG(\{\varphi \wedge B\}C_1\{\psi\}) = \{(\varphi \wedge B) \to wp(C_1, \psi)\} \cup VC(C_1, \psi)$$

  e

  $$\Gamma \models VCG(\{\varphi \wedge \neg B\}C_2\{\psi\}) = \{(\varphi \wedge \neg B) \to wp(C_2, \psi)\} \cup VC(C_2, \psi).$$

  Note-se que,

  $$wp(\texttt{if}\, B\, \texttt{then}\, C_1\, \texttt{else}\, C_2, \psi) = B \to wp(C_1, \psi) \wedge \neg B \to wp(C_2, \psi)\},$$

  logo

  $$\Gamma \models \{\varphi \to wp(\texttt{if}\, B\, \texttt{then}\, C_1\, \texttt{else}\, C_2, \psi)\}.$$

  Logo $\Gamma \models \{\varphi \to wp(\texttt{if}\, B\, \texttt{then}\, C_1\, \texttt{else}\, C_2, \psi)\} \cup VC(C_1, \psi) \cup VC(C_2, \psi) = VCG(\{\varphi\}\texttt{if}\, B\, \texttt{then}\, C_1\, \texttt{else}\, C_2\{\psi\})$.

Exercício 4.7. *Completa a demonstração anterior.*