Verificação Formal de Software

Nelma Moreira

Verificação Formal de Software Aula 17

Cálculo de Correção parcial $\mathcal{H}$

$[skip_p\ ]$

$$\{\phi\}\,\texttt{skip}\,\{\phi\}$$

$[ass_p\ ]$

$$\{\phi[E/x]\}\,x := E\,\{\phi\}$$

$[comp_p\ ]$

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

$[if_p\ ]$

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

$[while_p\ ]$

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

$[cons_p\ ]$

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

Mecanização da construção de derivações na lógica de Hoare

De um modo geral, dado um triplo de Hoare ({P}C{Q}) 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 a falta de ambiguidade na escolha das regras:

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

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

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

$$\frac{}{\{\phi\}\,x := E\,\{\psi\}} \text{ se } \models \phi \to \psi[E/x]$$

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

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

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

Sistema $\mathcal{H}_g$

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

Lema 17.1. *Se* $\Gamma \vdash_{\mathcal{H}_g} \{\phi\}C\{\psi\}$ *e* $\models \phi' \to \phi$, $\models \psi \to \psi'$, *então* $\Gamma \vdash_{\mathcal{H}_g} \{\phi'\}C\{\psi'\}$.

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

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

Exercício 17.1. *Completa a demonstração anterior.*

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

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

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

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

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

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

Exercício 17.2. *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*.

Desvantagens de $\mathcal{H}_g$:

- Perdemos alguma capacidade de re-utilizar resultados de correcção (veremos mais adiante como resolver esta questão).

A estratégia de pré-condição mais fraca

Queremos construir uma derivação para um triplo de Hoare $\{\phi\}C\{\psi\}$, 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 $\{\phi\}C_1\{\theta\}$, 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 $\phi \to \theta$, tomamos a pré-condição ϕ como θ.

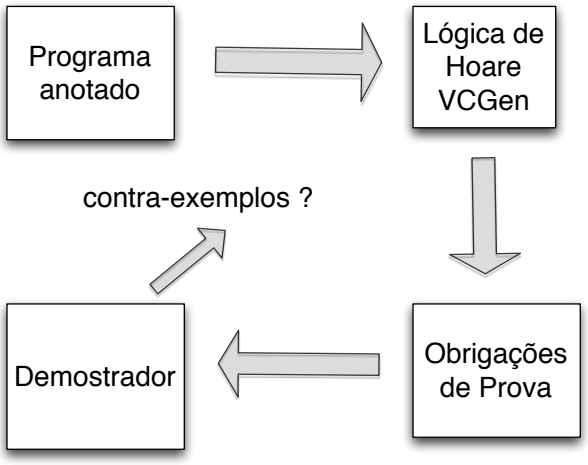
Uma Arquitectura para Verificação de Programas

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

1. Aplicamos os princípios apresentados anteriormente para construir uma derivação com conclusão $\{\phi\}C\{\psi\}$, 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 $\mathcal{T} \vdash_{\mathcal{H}_g} \{\phi\}C\{\psi\}$.

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 ϕ, podemos calcular $wp(C, \phi)$ tal que $\{wp(C, \phi)\}C\{\phi\}$ é válida e se $\{\psi\}C\{\phi\}$ é válida para algum ψ então $\psi \rightarrow wp(C, \phi)$.

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

Algoritmo VCGen

Primeiro calcula as VC não considerando as pré-condições

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

A pré-condição é tomada em consideração:

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

Propriedades de wp e VCG

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

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

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

- Para $C \equiv$ `skip`, temos $\Gamma \vdash_{\mathcal{H}_g} \{\phi\}\texttt{skip}\{\psi\}$ se $\models \phi \to \psi$. 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 $\Gamma \models \phi \to \psi = wp(\texttt{skip},\psi)$.
- $C \equiv$ `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 $\Gamma \models \phi \to \eta = wp(\texttt{while}\,B\,\texttt{do}\,\{\eta\}C\,\psi)$.

Exercício 17.3. *Completa a demonstração anterior.*

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

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

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

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

- Para $C \equiv C_1; C_2$, temos $VCG(\{\phi\}C_1; C_2\{\psi\}) = \{\phi \to wp(C_1; C_2, \psi)\} \cup VC(C_1; C_2, \psi) = \{\phi \to wp(C_1, wp(C_2, \psi))\} \cup VC(C_1, wp(C_2, \psi)) \cup VC(C_2\psi)$. Seja $\eta = wp(C_2, \psi)$. Como $\Gamma \models \phi \to wp(C_1, \eta) \cup VC(C_1, \eta) = VCG(\{\phi\}C_1\{\eta\}$, por I.H. $\Gamma \vdash_{\mathcal{H}_g} \{\phi\}C_1\{\eta\}$. Also $\Gamma \models \eta \to \eta \cup VC(C_2\psi) = VCG(\{\eta\}C_2\{\psi\}$, por I.H. $\Gamma \vdash_{\mathcal{H}_g} \{\eta\}C_2\{\psi\}$, logo $\Gamma \vdash_{\mathcal{H}_g} \{\phi\}C_1; C_2\{\psi\}$

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

- $\Gamma \vdash_{\mathcal{H}_g} \{\phi\}\texttt{skip}\{\psi\}$, se $\Gamma \models \phi \to \psi = VCG(\{\phi\}\texttt{skip}\{\psi\})$.
- $\Gamma \vdash_{\mathcal{H}_g} \{\phi\}\texttt{if}\, B\, \texttt{then}\, C_1\, \texttt{else}\, C_2, \{\psi\}$ se $\Gamma \vdash_{\mathcal{H}_g} \{\phi \wedge B\}C_1\{\psi\}$ e $\Gamma \vdash_{\mathcal{H}_g} \{\phi \wedge \neg B\}C_2\{\psi\}$. Por H.I. $\Gamma \models VCG(\{\phi \wedge B\}C_1\{\psi\}) = \{(\phi \wedge B) \to wp(C_1, \psi)\} \cup VC(C_1, \psi)$ e $\Gamma \models VCG(\{\phi \wedge \neg B\}C_2\{\psi\}) = \{(\phi \wedge \neg B) \to wp(C_2, \psi)\} \cup VC(C_2, \psi)$. 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 $\Gamma \models \{\phi wp(\texttt{if}\, B\, \texttt{then}\, C_1\, \texttt{else}\, C_2, \psi)\}$. Logo $\Gamma \models \{\phi \to wp(\texttt{if}\, B\, \texttt{then}\, C_1\, \texttt{else}\, C_2, \psi)\} \cup VC(C_1, \psi) \cup VC(C_2, \psi) = VCG(\{\phi\}\texttt{if}\, B\, \texttt{then}\, C_1\, \texttt{else}\, C_2\{\psi\})$.

Exercício 17.4. *Completa a demonstração anterior.*

Exemplo

Seja `fact` o seguinte programa:

```
f:=1; i:=1;
while i<= n do {f = (i-1)! and i <= n+1} {
      f:=f*i;
      i:=i+1;
}
```

Vamos calcular

$$\text{VCG}(\{n >= 0\}\texttt{fact}\{f = n!\})$$

com $\theta = f = (i-1)! \wedge i \leq n+1$ e $C_w = f := f * i; i := i + 1$

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

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

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. $f = (i-1)! \wedge i \leq n+1 \wedge i \leq n \rightarrow f = n!$