Verificación de programas

Dante Zanarini

LCC

21 de noviembre de 2024

Principales ingredientes para la verificación formal de programas

- Un lenguaje de programación;
- Una semántica para dicho lenguaje;
- Un lenguaje para especificar propiedades sobre programas;
- Un mecanismo para demostrar que un programa satisface una especificación;

Preguntas

- ¿Para qué verificar software?
- ¿Vale la pena verificar software?
- ¿Qué me aporta?
- ¿Sirve más allá de lo académico?
- Si yo verifiqué un programa, ¿este nunca va a fallar?
- ...

Nuestro lenguaje de programación

- Como cualquier lenguaje de programación imperativo, distinguimos expresiones de comandos.
- Expresiones enteras:

```
E    ::=    n   | x | (-E)
          | (E + E) | (E * E) | (E - E)
```

donde $\mathtt{n} \in \mathbb{Z}$, $\mathtt{x} \in \mathit{Var}$

- Expresiones booleanas

```
B    ::=    true   | false | (!B)
          | (B & B) | (B || B) | (E < E)
```

- Convenciones sintácticas, abreviaturas, ...

Nuestro lenguaje de programción (2)

- Nuestro lenguaje de Comandos es:

```
C ::= x = E
    | C ; C
    | if B {C} else {C}
    | while B {C}
```

Ejemplo, factorial

Listing 1: Ejemplo de código

```
y = 1;
z = 0;
while (z != x) {
  z = z + 1;
  y = y * z
  }
```

- Observemos que nuestro programa calcula el factorial de `x`, ¿seguro?

Especificando, tripletas de Hoare

- Es muy común que nuestras espeficicaciones informales tengan la forma

  "Dado un número positivo x, devuelve un número y tal que $y.y < x$"
- Si se acuerdan, en Programación 1 se reforzaba mucho esta forma de 'especificar' programas informalmente.
- Nuestra idea es formalizar estas especificaciones mediante **Tripletas de Hoare**:

$$(\!|\phi|\!)\ P\ (\!|\psi|\!)$$

- Para nuestro ejemplo, buscamos un programa P tal que:

$$(\!|x > 0|\!)\ P\ (\!|y.y < x|\!)$$

  sea una tripleta válida.

Tripletas de Hoare $(\!|\phi|\!)$ P $(\!|\psi|\!)$, preguntas

- ¿Qué significa cada parte?
- ¿A qué lenguaje pertenece cada una?
- ¿Qué significado quisiera que tenga que una terna ϕ, P, ψ sea una tripleta válida?
- ¿Qué pasa si ejecuto el programa en un estado donde no se cumple ϕ?
- ¿Para cada par ϕ, ψ hay un P tal que $(\!|\phi|\!)$ P $(\!|\psi|\!)$ es válida?
- ¿Para cada par ϕ, ψ hay un único P tal que $(\!|\phi|\!)$ P $(\!|\psi|\!)$ es válida?
- ¿Qué pasa si el programa no termina?
- ¿Quién fue C. A. R. Hoare?

Tripletas de Hoare $(\!|\phi|\!)$ P $(\!|\psi|\!)$, definiciones

- La fórmula ϕ la llamamos *precondición*
- La fórmula ψ la llamamos *postcondición*
- Ambas fórmulas están definidas sobre una signatura $\mathcal{F} = \{-, \dot{-}, +, *\}$, $\mathcal{P} = \{=, <\}$
- Un *estado* es una función $l : Var \to \mathbb{Z}$
- Decimos que $l \models \phi$ si $[\![\phi]\!]_{\mathcal{M},l} = T$, donde $\mathcal{M}$ es el modelo estándar de los números enteros para $(\mathcal{F}, \mathcal{P})$
- Ejemplo: si l es tal que $l(x) = 2$, $l(y) = -5$, $l(z) = 3$, determinar si:

$$
\begin{aligned}
&l \models 3 * x + y < z \\
&l \models x + y = z \\
&l \models 2 * x - z < y \\
&l \models \exists u(2 * u + x = z) \\
&l \models \forall u(u > 0 \to x * u < z * u)
\end{aligned}
$$

Dos sabores para corrección: parcial y total

Definición (Corrección parcial $\models_{\mathrm{par}}$)

Decimos que una tripleta $(\!|\phi|\!)\ P\ (\!|\psi|\!)$ se cumple bajo corrección parcial ($\models_{\mathrm{par}} (\!|\phi|\!)\ P\ (\!|\psi|\!)$) si, para cualquier estado l que satisface ϕ, si el programa P termina, lo hace en un estado que satisface ψ.

- Si no hay estado inicial que cumpla la precondición, se cumple trivialmente $\models_{\mathrm{par}} (\!|\phi|\!)\ P\ (\!|\psi|\!)$
- Si el programa no termina en ningún estado, se cumple $\models_{\mathrm{par}} (\!|\phi|\!)\ P\ (\!|\psi|\!)$, cualesquiera sean las pre y postcondiciones

Corrección parcial

- Volvamos a la tripleta $(\!|x > 0|\!)$ P $(\!|y.y < x|\!)$, podemos demostrar que si P es

  ```
  y = 0;
  while (y * y < x) {
    y = y + 1
  }
  y = y - 1
  ```

  se cumple $\models_{\mathrm{par}} (\!|x > 0|\!)\ P\ (\!|y.y < x|\!)$
- Pero también si P es
  `y = 0`
- Y también, si P es
  `while (true) {y = 1}`

Dos sabores para corrección: parcial y total

Definición (Corrección total $\models_{\mathrm{tot}}$)

Decimos que una tripleta $(|\phi|)\ P\ (|\psi|)$ se cumple bajo corrección total ($\models_{\mathrm{tot}} (|\phi|)\ P\ (|\psi|)$) si, para cualquier estado l que satisface ϕ, el programa P termina y lo hace en un estado que satisface ψ.

- Si ningún estado inicial cumple la precondición se cumple trivialmente $\models_{\mathrm{tot}} (|\phi|)\ P\ (|\psi|)$
- Si el programa no termina para algún estado que cumple la precondición, **no** se cumple $\models_{\mathrm{tot}} (|\phi|)\ P\ (|\psi|)$
- Si un estado no cumple la precondición y en ese estado el programa no termina, esto no significa que la tripleta no se cumpla. La obligación de terminar es sólo para los estados que cumplen la pre.

Un cálculo para $\models_{\text{par}}$

$$\frac{(\!|\phi|\!)\ \texttt{C1}\ (\!|\eta|\!) \qquad (\!|\eta|\!)\ \texttt{C2}\ (\!|\psi|\!)}{(\!|\phi|\!)\ \texttt{C1;C2}\ (\!|\psi|\!)}\ \text{Comp}$$

$$\frac{}{(\!|\psi[E/x]|\!)\ \texttt{x=E}\ (\!|\psi|\!)}\ \text{Asig}$$

$$\frac{(\!|\phi \wedge B|\!)\ \texttt{C1}\ (\!|\psi|\!) \quad (\!|\phi \wedge \neg B|\!)\ \texttt{C2}\ (\!|\psi|\!)}{(\!|\phi|\!)\ \texttt{if B \{C1\} else \{C2\}}\ \ (\!|\psi|\!)}\ \text{If}$$

$$\frac{(\!|\psi \wedge B|\!)\ \texttt{C}\ (\!|\psi|\!)}{(\!|\psi|\!)\ \texttt{while B \{C\}}\ (\!|\psi \wedge \neg B|\!)}\ \text{WhileP}$$

$$\frac{\vdash_{AR} \phi' \to \phi \qquad (\!|\phi|\!)\ \texttt{C}\ (\!|\psi|\!) \qquad \vdash_{AR} \psi \to \psi'}{(\!|\phi'|\!)\ \texttt{C}\ (\!|\psi'|\!)}\ \text{Debilitamiento}$$

Regla de asignación

Si P es `x = 2`, las siguientes son instancias de la regla:

- $(\!|2 = 2|\!)\ P\ (\!|x = 2|\!)$
- $(\!|2 = 4|\!)\ P\ (\!|x = 4|\!)$
- $(\!|2 = y|\!)\ P\ (\!|x = y|\!)$
- $(\!|2 > 0|\!)\ P\ (\!|x > 0|\!)$

Regla de asignación

Si P es `x = 2`, las siguientes son instancias de la regla:

- $(\!|2 = 2|\!)\ P\ (\!|x = 2|\!)$
- $(\!|2 = 4|\!)\ P\ (\!|x = 4|\!)$
- $(\!|2 = y|\!)\ P\ (\!|x = y|\!)$
- $(\!|2 > 0|\!)\ P\ (\!|x > 0|\!)$

Si P es `x = x+1`, las siguientes son instancias de la regla:

- $(\!|x + 1 = 2|\!)\ P\ (\!|x = 2|\!)$
- $(\!|x + 1 = y|\!)\ P\ (\!|x = y|\!)$
- $(\!|x + 1 + 5 = y|\!)\ P\ (\!|x + 5 = y|\!)$
- $(\!|x + 1 > 0 \wedge y > 0|\!)\ P\ (\!|x > 0 \wedge y > 0|\!)$