Verificación de programas I: Precondición más débil

Román Gorojovsky

Algoritmos y Estructuras de Datos

10 de abril de 2024

Plan del día

Plan del día

- Introducción
- Repasos
- Calcular WPs
- Contraejemplos de WPs erróneas

Precondición más débil

Personajes principales

- Precondición P (en lógica)
- Código **S** (en SmallLang)
- Postcondición Q (en lógica)

Personaje secundario pero fundamental

- Aridad o firma de la función (parámetros de entrada y salida)

Precondición más débil

Problema ejemplo

proc esPar (in n: $\mathbb{Z}$) : Bool
requiere $\{$*a definir*$\}$
asegura $\{res = True \leftrightarrow n \bmod 2 = 0\}$

Precondición más débil

Problema ejemplo

`proc esPar` (in n: $\mathbb{Z}$) : Bool
    `requiere` $\{$*a definir*$\}$
    `asegura` $\{res = True \leftrightarrow n \bmod 2 = 0\}$

¿Es válida esta implementación?

```
res := True
```

Precondición más débil

Problema ejemplo

proc esPar (in n: $\mathbb{Z}$) : Bool
requiere $\{a\ definir\}$
asegura $\{res = True \leftrightarrow n \bmod 2 = 0\}$

¿Es válida esta implementación?

```
res := True
```

¡Depende de la precondición!

Precondición más débil

Problema ejemplo

proc esPar (in n: $\mathbb{Z}$) : Bool
  requiere {*a definir*}
  asegura $\{res = True \leftrightarrow n \bmod 2 = 0\}$

¿Es válida esta implementación?

```
res := True
```

¡Depende de la precondición!

$$P \equiv \{n \bmod 2 = 0\}$$

Precondición más débil

Problema ejemplo

`proc esPar` (in n: $\mathbb{Z}$) : Bool
    `requiere` $\{$*a definir*$\}$
    `asegura` $\{res = True \leftrightarrow n \bmod 2 = 0\}$

Precondición más débil

Problema ejemplo

proc esPar (in n: $\mathbb{Z}$) : Bool
  requiere $\{\textit{a definir}\}$
  asegura $\{res = True \leftrightarrow n \bmod 2 = 0\}$

¿Es correcta esta precondición?

$$P \equiv \{n \geq 0\}$$

Precondición más débil

Problema ejemplo

proc esPar (in n: $\mathbb{Z}$) : Bool
    requiere $\{$*a definir*$\}$
    asegura $\{res = True \leftrightarrow n \bmod 2 = 0\}$

¿Es correcta esta precondición?

$$P \equiv \{n \geq 0\}$$

Es demasiado restrictiva

Precondición más débil

Precondición más débil – Idea informal

Es la P que permite que el programa **S** funcione correctamente, pero restringiendo lo menos posible.

Precondición más débil

Precondición más débil – Idea informal

Es la P que permite que el programa **S** funcione correctamente, pero restringiendo lo menos posible.

Principio de diseño

Ser cuidadoso con los resultados que se emiten y generoso con los parámetros que se reciben.

Ejercicios para la primera parte

1. wp(**a := a+1**, $a \geq 0$)
2. wp(**a := a+1; b:= a/2**, $b \geq 0$) *(Ejercicio 2.a de la práctica)*
3. wp(**A[i] := -1**, $\forall (j : \mathbb{Z})(0 \leq j < |A| \rightarrow_L A[j] \geq 0)$)
4. wp(**A[i] := A[i-1]**, $\forall (j : \mathbb{Z})(0 \leq j < |A| \rightarrow_L A[j] \geq 0)$)
5. wp(S, Q) con
   - $S \equiv$

     ```
     if( a < 0 )
       b := a
     else
       b := −a
     endif
     ```

   - $Q \equiv (b = -|a|)$

   (Ejercicio 4.a de la práctica)

Donde $a, b \in \mathbb{R}$, $i \in \mathbb{Z}$, $A : seq < \mathbb{Z} >$

Predicados útiles

Definiciones (copiadas de la teórica)

- Dada una expresión E, llamamos $\text{def}(E)$ a las condiciones necesarias para que E esté definida.
- Dado un predicado Q, el predicado Q^x_E se obtiene reemplazando en Q todas las apariciones libres de la variable x por E.

Axiomas (primera parte)

Definiciones (copiadas de la teórica)

- **Axioma 1:** $wp(\mathbf{x := E}, Q) \equiv \mathrm{def}(E) \wedge_L Q^x_E$
- **Axioma 2:** $wp(\mathbf{skip}, Q) \equiv Q$
- **Axioma 3:** $wp(\mathbf{S1;\ S2}, Q) \equiv wp(\mathbf{S1}, wp(\mathbf{S2}, Q))$

setAt()

Axioma 1 y secuencias

No podemos usar el Axioma 1 para el programa `b[i] := E`, sólo matchea con `x := E` cuando `x` es una variable

Definiciones (copiadas de la teórica)

- $b[i] := E \equiv b := setAt(b, i, E)$
- $\mathrm{def}(setAt(b, i, E)) = (\mathrm{def}(E) \ \wedge \mathrm{def}(b) \ \wedge \ \mathrm{def}(i)) \wedge_L \left(0 \leq i < |b|\right)$
- Dados $0 \leq i, j < |b|$:

$$setAt(b, i, E)[j] \ = \begin{cases} E & \text{si } i = j \\ b[j] & \text{si } i \neq j \end{cases}$$

Axiomas (segunda parte)

Definiciones (copiadas de la teórica)

- **Axioma 1:** $wp(\mathbf{x := E}, Q) \equiv \text{def}(E) \wedge_L Q^x_E$
- **Axioma 2:** $wp(\mathbf{skip}, Q) \equiv Q$
- **Axioma 3:** $wp(\mathbf{S1; S2}, Q) \equiv wp(\mathbf{S1}, wp(\mathbf{S2}, Q))$
- **Axioma 4:** Si $\mathbf{S} =$ **if B then S1 else S2 endif**, entonces

$$wp(\mathbf{S}, Q) \equiv def(B) \wedge_L \Big((B \wedge wp(\mathbf{S1}, Q)) \vee (\neg B \wedge wp(\mathbf{S2}, Q))\Big)$$

WPs con errores

Dado este código y postcondicion

```
if (i mod 2 = 0)
    s[i] = 2*s[i];
else
    s[0] = 3;
endif
```

$$Q \equiv \{(\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j]\ mod\ 2 = 0)\}$$

mostrar que las siguientes WPs son incorrectas, dando un contraejemplo de ser posible

1. $P \equiv \{0 \leq i < |s| \wedge (i\ mod\ 2 = 0)\}$
2. $P \equiv \{0 \leq i < |s| \wedge ((i\ mod\ 2 = 0) \wedge (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j]\ mod\ 2 = 0) \vee (i\ mod\ 2 \neq 0) \wedge (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j]\ mod\ 2 = 0))\}$
3. $P \equiv \{0 \leq i < |s| \wedge (i\ mod\ 2 = 0) \wedge (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j]\ mod\ 2 = 0)\}$