Algoritmos y Estructura de Datos

Clase práctica 4 - Demostración de corrección de ciclos

Miércoles 6 de Septiembre de 2023

¿Qué vamos a ver hoy?

- Teorema del **invariante**
- Teorema de **terminación**
- Demostración de corrección

Teorema del invariante

Teorema. Si def(B) y existe un predicado I tal que se cumple

1. $P \rightarrow I$
2. $\{I \wedge B\} S \{I\}$
3. $I \wedge \neg B \rightarrow Q$

Y el ciclo termina, entonces tenemos que:

$$\{P\} \textbf{ while B do S endwhile } \{Q\}$$

Teorema de terminación

Teorema. Sea $\mathbb{V}$ el producto cartesiano de los dominions de las variables del programa y sea I un invariante del ciclo **while B do S endwhile**. Si existe una función $f_v : \mathbb{V} \to \mathbb{Z}$ tal que se cumple

1. $(\forall v_0 : \mathbb{Z})(\{I \wedge B \wedge f_v = v_0\}$ **S** $\{f_v < v_0\})$
2. $I \wedge f_v \leq 0 \to \neg B$

Entonces la ejecución del ciclo **while B do S endwhile** siempre termina.

Importante

- Para demostrar $Ant \to Cons$ podemos usar Ant como hipótesis, es decir, tomar sus afirmaciones como verdaderas
- Para pensar un invariante de ciclo, tengo que observar las cosas que se hacen dentro del ciclo, las pre y poscondiciones. Pensar qué cosas necesito para poder demostrar lo que se requiere en los teoremas
- Para pensar en una función variante, observo la guarda y veo cómo se modifican los valores que intervienen

Consigna

Para los siguientes problemas se pide:

1. Escribir la precondición y poscondición del ciclo
2. Proponer un **invariante** y demostrar que el ciclo es parcialmente correcto
3. Proponer una **función variante** que permita demostrar que el ciclo termina

Primer ejercicio

`proc productoria` (in a: $seq\langle\mathbb{Z}\rangle$, out prod: $\mathbb{Z}$)
`requiere` $\{|a| \bmod 2 = 0\}$
`asegura` $\{a = \prod_{i=0}^{|a|-1} a[i]\}$

Implementación

$i := 0;$
$prod := 1;$
while $i < a.size() - 1$ **do**
  $prod := prod * a[i] * a[i+1];$
  $i := i + 2;$
end while

`proc copiarSecuencia` (in s: $seq\langle\mathbb{Z}\rangle$, inout r: $seq\langle\mathbb{Z}\rangle$)
  `requiere` $\{r = R_0 \land |s| = |r|\}$
  `asegura` $\{|s| = |r| \land_L (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] = r[j])\}$

Implementación

$i := 0;$
while $i < s.size()$ **do**
  $r[i] := s[i];$
  $i := i + 1;$
end while

¡Terminamos por hoy!

¡Con esto ya pueden resolver hasta el ejercicio 10 de la guía 3!