# TEORÍA DE LA COMPUTACIÓN Y VERIFICACIÓN DE PROGRAMAS

## TP 5 - Verificación de programas (clases 10 y 11)

**Ejercicio 1. La raíz cuadrada entera de un número natural $n$ es el mayor de los números enteros $m$ que cumplen $m^2 \leq n$. Se pide:**

**a) Especificar un programa que dado un número natural n como input, obtenga como output su raíz cuadrada entera m, y mantenga al final el valor del input.**

$\Phi = (n = N \wedge N \varepsilon \aleph, m^2 \leq n \wedge m \varepsilon \aleph \wedge n = N \wedge m > \sqrt{n} - 1)$

**b) ¿Podría agregarse a la especificación que el valor del input no se altere a lo largo de todo el programa? Justificar.**

No, debido a que la especificación consta de una precondición y una postcondición solamente, por lo tanto, lo que el programa haga es transparente, sólo se sabe que se cumplen las aserciones p y q.

**Ejercicio 2. Asumiendo |= {p} S {q}, indicar en cada caso si vale lo afirmado. Justificar las respuestas:**

**a) Si S termina en un estado que satisface q, entonces su estado inicial satisface p.**

El estado inicial puede ser cualquiera, no necesariamente p. Por ejemplo, siendo q x=5 y S:: x:=5, S siempre terminará en un estado que satisface q sin importar el estado inicial.

**b) Si S termina en un estado que no satisface q, entonces su estado inicial no satisface p.**

Prueba por absurdo.

Dado $\neg$(S termina en un estado que no satisface q $\rightarrow$ su estado inicial no satisface p)

$\Leftrightarrow$ S termina en un estado que no satisface q $\wedge$ su estado inicial satisface p

$\Leftrightarrow$ $| = \{p\} S \{\neg q\}$

lo cual es un absurdo debido a que asumimos en un principio |= {p} S {q}

Por lo tanto queda demostrado que si S termina en un estado que no satisface q, entonces su estado inicial no satisface p.

**c) Si S no termina, entonces su estado inicial no satisface p.**

Por la definición de parcialmente correcto,

Si S no termina entonces $val(\pi(S, \sigma)) = \perp$ , dado esto, la definición de parcialmente correcto $(\sigma| = p \wedge val(\pi(S, \sigma)) \neq \perp) \rightarrow val(\pi(S, \sigma))| = q$ , se cumple ya que es falso $val(\pi(S, \sigma)) \neq \perp$ lo cual vuelve a la implicancia verdad.

**d) ¿Las respuestas en (a), (b) y (c) son las mismas considerando la fórmula |= ⟨p⟩ S ⟨q⟩ en lugar de la fórmula |= {p} S {q}?**

Las respuestas (a) y (b) son las mismas, en cambio, la respuesta (c) cambia debido a la definición de correctitud total $\sigma| = p \rightarrow (val(\pi(S,\sigma)) \neq \perp \wedge val(\pi(S,\sigma))| = q)$ . Segun esta definicion y dado |= ⟨p⟩ S ⟨q⟩, S con estado inicial p debe terminar siempre y en estado final q.

**Ejercicio 3. Indicar en cada caso si vale lo afirmado. Justificar las respuestas:**
**a) Si p es x = 0, q es x = 1, y S es while z = 0 do z := 0 od, entonces |= {p} S {q}.**

|={x=0}while z = 0 do z := 0 od{x=1}
Se cumple, ya que loopea y por lo tanto $val(\pi(S,\sigma)) = \perp$ cumpliendo la definición de correctitud parcial.

**b) Si se cumple |= {p1∧p2} S {q1∧q2}, entonces |= {p1} S {q1} o bien |= {p2} S {q2}.**

prueba por absurdo
Dado que ¬(si se cumple |= {p1∧p2} S {q1∧q2}, entonces |= {p1} S {q1} o bien |= {p2} S {q2})
⇒ |= {p1∧p2} S {q1∧q2} ∧ ¬(|= {p1} S {q1} ∨ |= {p2} S {q2})
⇒ |= {p1∧p2} S {q1∧q2} ∧ |≠ {p1} S {q1} ∧ |≠ {p2} S {q2})

Contraejemplo:
Dado
p1 :: x=5, p2 :: y=6, q1 :: x=6, q2 :: y=5, $S_{swap}$ :: z=x;x=y;y=z
: {p1∧p2} S {q1∧q2} se cumple |= {p1∧p2} S {q1∧q2}
Si tuviéramos $\sigma_1$ con $\sigma_1(x) = 5$ y $\sigma_1(y) = 7$ , $\sigma_1| = p1$
entonces $val(\pi(S_{swap},\sigma_1)) = \sigma_2$ con $\sigma_2(x) = 7$ y $\sigma_2(y) = 5$ , $\sigma_2| \neq q1$
por lo que $| = \ \{p1\}\ S\{\neg q1\}$

Dado $\sigma_1| = p1 \wedge p2$ y $\sigma_2| = q1 \wedge q2$ , $\{\sigma_1\}\ S\ \{\sigma_2\}$ ( |= {p1∧p2} S {q1∧q2})
Si $\sigma_1| = p1$

Se cumple
$( (\sigma| = (p1 \wedge p2) \wedge val(\pi(S,\sigma)) \neq \perp) \rightarrow val(\pi(S,\sigma))| = (q1 \wedge q2) )$
Dado
$\sigma| = p1 \wedge val(\pi(S,\sigma)) \neq \perp$

**Ejercicio 4. Sea el siguiente lenguaje de expresiones enteras: e :: 0 | 1 | x | (e1 + e2) | (e1 . e2). Y sea var(e) el conjunto de las variables de e. Se pide definir inductivamente var(e).**
**Ayuda: Por ejemplo, var(x) = {x}.**

1. var(0) = {0}
2. var(1) = {1}

3. var(x) = {x}
4. var(e1+e2) = var(e1)+var(e2)
5. var(e1.e2) = var(e1).var(e2)

**Ejercicio 5. Probar que se cumple, para todo estado σ y para todo par de aserciones p, q, que val(π(S1, σ)) = val(π(S2, σ)) si y sólo si |= {p} S1 {q} ↔ |= {p} S2 {q}. Comentario: Para facilitar la notación, se puede utilizar M(S)(σ) en lugar de val(π(S, σ)).**

Probaré:
( M(S1)(σ) = M(S2)(σ) ) ↔ ( |= {p} S1 {q} ↔ |= {p} S2 {q} )

1)
( M(S1)(σ) = M(S2)(σ) ) → ( |= {p} S1 {q} ↔ |= {p} S2 {q} )

M(S1)(σ) = M(S2)(σ) entonces

- si σ$|\neq p$ se cumple |= {p} S1 {q} ↔ |= {p} S2 {q}
- si M(S1)(σ) = ⊥ y M(S2)(σ) = ⊥ entonces se cumple |= {p} S1 {q} ↔ |= {p} S2 {q}.
- si σ$|=p$, M(S1)(σ) ≠ ⊥ y M(S2)(σ) ≠ ⊥ entonces M(S1)(σ) = σ' y M(S2)(σ) = σ' con σ'|=q entonces se cumple |= {p} S1 {q} ↔ |= {p} S2 {q}.

2)
( |= {p} S1 {q} ↔ |= {p} S2 {q} ) → ( M(S1)(σ) = M(S2)(σ) )

( |= {p} S1 {q} ↔ |= {p} S2 {q} ) entonces
$((\sigma| = p \wedge M(S1)(\sigma) \neq \bot) \rightarrow M(S1)(\sigma)| = q) \leftrightarrow ((\sigma| = p \wedge M(S2)(\sigma) \neq \bot) \rightarrow M(S2)(\sigma))| = q)$
Si $\sigma| \neq p$
Si $M(S1)(\sigma) = \bot$ entonces $M(S2)(\sigma) = \bot$

**Ejercicio 6. Supóngase que se agrega al lenguaje PLW la instrucción repeat S until B, con la semántica habitual. Definir la semántica operacional de dicha instrucción, y extender el método H con una regla para la misma.**

Definición inductiva de la semántica:
Dado
Si $\sigma(B) = verdadero$, $(while\ B\ do\ S\ od, \sigma) \rightarrow (S; while\ B\ do\ S\ od, \sigma)$
$\sigma(B) = falso$, $(while\ B\ do\ S\ od, \sigma) \rightarrow (S, \sigma)$
$(repeat\ S\ until\ B, \sigma) \rightarrow (S; while\ B\ do\ S\ od, \sigma)$

Defino la regla REPEAT para el método H:
Usando la definición inductiva de la semantica $(repeat\ S\ until\ B, \sigma) \rightarrow (S; while\ B\ do\ S\ od, \sigma)$

$$\frac{\dfrac{\{p\}S\{q\},\{q \wedge B\}\ S\ \{q\}}{\{p\}S\{q\},\{q\}while\ B\ do\ S\ od\{q \wedge \neg B\}}}{\{p\}S;\ while\ B\ do\ S\ od\{q \wedge \neg B\}}$$

**Ejercicio 7. Probar utilizando H las fórmulas de correctitud siguientes:**

**a) {x = X} Sabs :: if x > 0 then y := x else y := – x {y = |X|}, siendo |X| el valor absoluto de X.**

Dado Sabs :: if x > 0 then y := x else y := – x se probará:
{x = X} Sabs {y = |X|}

Considerando las asignaciones del programa los 2 primeros pasos son:

1. $\{x = X \wedge x > 0\} y := x \{y \geq 0\}$ (ASI)
2. $\{x = X \wedge x \leq 0\} y := -x \{y \geq 0\}$ (ASI)
3. $(x = X \wedge x > 0) \rightarrow (x = X \wedge x > 0)$ (MAT)
4. $(x = X \wedge \neg(x > 0)) \rightarrow (x = X \wedge x \leq 0)$ (MAT)
5. $\{x = X \wedge x > 0\} y := x \{y \geq 0\}$ (1,3,CONS)
6. $\{x = X \wedge \neg(x > 0)\} y := x \{y \geq 0\}$ (2,4,CONS)
7. $\{x = X\} S_{abs} \{y \geq 0\}$ (5,6,COND)

**b) {x≥0 ^ y≥0} Sprod :: prod := 0; k := y; while k>0 do prod := prod + x; k := k – 1 od {prod=x.y}. Ayuda: Sprod calcula en la variable prod el producto entre x e y.**

Dado Sprod :: prod := 0; k := y; while k>0 do prod := prod + x; k := k – 1 od se probará:
{x≥0 ^ y≥0} Sprod {prod=x.y}

Se estructurará la prueba de la siguiente manera:

a. $\{x \geq 0 \wedge y \geq 0\} prod := 0; k := y \{k \geq 0 \wedge prod \leq x.y\}$
b. $\{k \geq 0 \wedge prod \leq x.y\} while\ k > 0\ do\ prod := prod + x;\ k := k - 1\ od\ \{prod = x.y \wedge k = 0\}$
c. $\{x \geq 0 \wedge y \geq 0\} S_{prod} \{prod = x.y\}$

Prueba de a:

1. $\{0 = prod \wedge prod \leq x.y\} k := y \{k \geq 0 \wedge 0 = prod \wedge prod \leq x.y\}$ (ASI)
2. $\{0 = 0 \wedge x.y \geq 0\} prod := 0 \{0 = prod \wedge x.y \geq prod\}$ (ASI)
3. $\{x \geq 0 \wedge y \geq 0\} prod := 0 ; k := y \{k \geq 0 \wedge prod \leq x.y\}$ (1,2,SEC,CONS)

Prueba de b:

4. $\{prod \leq x.y \wedge k - 1 \geq 0\} k := k - 1 \{k \geq 0 \wedge prod \leq x.y\}$ (ASI)
5. $\{x > 0 \wedge prod + x \leq x.y \wedge k - 1 \geq 0\} prod := prod + x \{prod \leq x.y \wedge k - 1 \geq 0\}$ (ASI)
6. $\{k > 0 \wedge prod < x.y\} prod := prod + x; k := k - 1 \{k \geq 0 \wedge prod \leq x.y\}$ (4,5,SEC,CONS)
7. $\{k \geq 0 \wedge prod \leq x.y\} while\ k > 0\ do\ prod := prod + x;\ k := k - 1\ od\ \{prod = x.y\}$
   (6,REP,CONS)

Prueba de c:

8. $\{x \geq 0 \wedge y \geq 0\}\ S_{prod} \{prod = x.y\}$ (3,7,SEC)