

1 Enunciado

El objetivo de esta última práctica será estudiar la corrección de los programas proporcionados en la sección 3, en base a sus enunciados. Los ejercicios planteados son preguntas de exámenes de años anteriores. Se facilitan las reglas de Hoare para la corrección parcial en la sección 2. Esta práctica se corregirá en clase los días 2, 3 y 4 de Junio.

2 Reglas de Hoare para corrección parcial

- Skip

$$\frac{}{\{A\} \text{ skip}; \{A\}}$$

- Asignación

$$\frac{}{\{A[a/X]\} \text{ x} := \text{a}; \{A\}}$$

- Secuencia

$$\frac{\{A\} c_0; \{C\} \quad \{C\} c_1 \{B\}}{\{A\} c_0; c_1 \{B\}}$$

- Condicional

$$\frac{\{A \wedge b\} c_0; \{B\} \quad \{A \wedge \neg b\} c_1 \{B\}}{\{A\} \text{ if } b \text{ then } c_0 \text{ else } c_1 \text{ fi } \{B\}}$$

- Implicación

$$\frac{\models (A \Rightarrow A') \quad \{A'\} c \{B'\} \quad \models (B' \Rightarrow B)}{\{A\} c \{B\}}$$

- While

$$\frac{\{A \wedge b\} c \{A\}}{\{A\} \text{ while } b \text{ do } c \text{ done } \{A \wedge \neg b\}}$$

3 Ejercicios

Ejercicio 1 - Probar formalmente la corrección total de la siguiente especificación. El algoritmo almacena en i la mayor potencia de 2 que es a lo sumo n . La cota para el bucle es $n - i$.

```
{0 < n}
i := 1;
{inv : 0 < i ≤ n ∧ ∃ p : i = 2p}
while (2 · i ≤ n) do
    i := 2 · i;
done
{0 < i ≤ n < 2 · i ∧ ∃ p : i = 2p}
```

Ejercicio 2 - Verifica la corrección de la siguiente aserción de corrección parcial.

```
{0 ≤ b}
z := 1;
x := a;
y := b;
{inv : y ≥ 0 ∧ z · (xy) = ab}
while (y ≠ 0) do
    if impar(y) then
        z := z · x;
        y := y - 1;
    else
        x := x · x;
        y := y/2;
    fi
done
{z = ab}
```

Ejercicio 3 - Demostrar la corrección del siguiente programa anotado.

```
{A ≥ 0}
x := A;
y := B;
s := 0 :
{inv : x · y + s = A · B} ∧ x ≥ 0}
while ( y ≠ 0 ) do
  {inv : x · y + s = A · B ∧ x > 0}
  while (x mod 2 = 0) do
    y := 2 · y;
    x := x/2;
  done
  s := s + y;
  x := x - 1;
done
{s = A · B}
```

Ejercicio 4 - Explica la semántica del siguiente programa, comprueba que $\{ \text{inv} : 0 \leq r \wedge x = q \cdot y + r \}$ es un invariante del bucle y que el programa es correcto.

```
{x ≥ 0}
q := 0;
r := x;
while (r ≥ y) do
  r := r - y;
  q := q + 1;
done
{0 ≤ r < y ∧ x = q · y + r}
```
