



Práctica 7: Verificación de programas - Lógica de Hoare

1) El operador `for` tiene diferentes sintaxis y formas de uso dependiendo del lenguaje de programación. Para este ejercicio consideraremos una similar a la del lenguaje C, donde es frecuente escribir código como:

```
for(i= 0; i < N; i++) {  
    a[i] = b[i];  
}
```

Observemos que `i= 0`, `i++` y `a[i] = b[i]` son comandos, mientras que `i < N` es una expresión. Proponga una traducción, expresada como un comando de nuestro lenguaje de programación minimal para el comando:

```
for(C1; B; C2) {  
    C3;  
}
```

2) Similar al ejercicio anterior, otro comando 'perdido' respecto a la sintaxis que podemos encontrar en otros lenguajes de programación es el `repeat` o `do-while`:

```
do {  
    C  
} while B;
```

Proponga una traducción a nuestro lenguaje de comandos.

3) Sea l tal que $l(x) = 4$, $l(y) = -1$ y $l(z) = 7$. Determine si las siguientes proposiciones son verdaderas o falsas:

1. $l \models x * x < x * y * z$
2. $l \models \forall w, w > 0 \rightarrow w * x \leq w * y$
3. $l \models x + 2 * y = u \rightarrow u \geq z$
4. $l \models \forall w, w + 1 \geq z + y \rightarrow w > 6$

4) Demuestre:

1. $\vdash_{\text{par}} (x > 0) \ y = x+1 \ (y > 1)$
2. $\vdash_{\text{par}} (\top) \ y = x; \ y = x + x + y \ (y = 3x)$
3. $\vdash_{\text{par}} (x > 1) \ a = 1; \ y = x; \ y = y - a \ (y > 0 \wedge x > 1)$

5) Considere el siguiente programa P :

```
if (x > y) {  
    z = y  
}  
else {  
    z = x  
}
```

Demuestre que $\vdash_{\text{par}} (\top) \ P \ (z = \min(x, y))$. Como parte de este ejercicio, seguramente deba expresar en el lenguaje de la lógica de predicados qué significa $z = \min(x, y)$.

6) Para cada tripleta, encuentre un programa P que la haga válida y demuéstrela.

1. $(\top) \ P \ ((x = 3 \rightarrow z = 2 * x) \wedge (\neg(x = 3) \rightarrow z = 6))$
2. $(\top) \ P \ ((x > 0 \rightarrow z = x \wedge y = 5) \wedge (x \leq 0 \rightarrow z = x + 2 \wedge y = x))$

7) Demuestre que $\vdash_{\text{par}} \langle x \geq 0 \rangle \text{ Copy } \langle x = y \rangle$, donde **Copy** es el siguiente programa:

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

8) Demuestre que $\vdash_{\text{par}} \langle y = y_0 \wedge y \geq 0 \rangle \text{ Mult } \langle z = x.y_0 \rangle$, donde **Mult** es el siguiente programa:

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

9) Demuestre que $\vdash_{\text{par}} \langle \neg(y = 0) \rangle \text{ Div } \langle (x = d.y + r) \wedge (r < y) \rangle$, donde **Div** es el siguiente programa que calcula el cociente y el resto de dividir x entre y :

```
r = x;
d = 0;
while (r >= y) {
  r = r - y;
  d = d + 1
}
```