Sensatez, completitud, verificación de programas concurrentes (clases 12 y 13)

Comentario: Hacer mínimamente los ejercicios 1 al 4.

Ejercicio 1. Sea el lenguaje de expresiones enteras: e :: 0 | 1 | x | (e_1 + e_2) | (e_1 . e_2). Y sea var(e) el conjunto de las variables de *e*. Se pide definir inductivamente var(e). P.ej.: var(0) = ∅.

Ejercicio 2. Probar la sensatez de la regla de invariancia vista en clase:

$$\frac{\{p\}\ S\ \{q\}}{\{r \wedge p\}\ S\ \{r \wedge q\}}$$

cuando las variables libres de r son disjuntas con las variables modificables por S.
Ayuda: Utilizar inducción matemática fuerte sobre la longitud de las pruebas.

Ejercicio 3. Supóngase que se agrega al lenguaje de programación visto en clase, la instrucción repeat S until B, con la semántica habitual (se ejecuta S, se evalúa B, si se cumple B se termina la repetición, y si no se cumple B se vuelve al comienzo). Se pide:

a. Definir la semántica operacional de la instrucción.
b. Proponer una regla de prueba para la misma.

Ejercicio 4. Probar sin recurrir a la completitud de H (es decir que la prueba debe ser sintáctica) que para todo programa S y toda aserción q se cumple:

$$\vdash \{false\}\ S\ \{q\}$$

Ayuda: Utilizar inducción estructural sobre la forma de los programas S, similar a lo visto en clase para probar sintácticamente la fórmula {true} S {true}.

Ejercicio 5. Probar que para todo estado σ y para todo par de aserciones p, q, se cumple:

$$val(\pi(S_1, \sigma)) = val(\pi(S_2, \sigma)) \text{ sii } \{p\}\ S_1\ \{q\} \leftrightarrow \{p\}\ S_2\ \{q\}$$

Ejercicio 6. Se define la postcondición más fuerte de la siguiente manera:

$$post(p, S) = \{\sigma' \mid \exists\sigma : \sigma \models p \wedge val(\pi(S, \sigma)) = \sigma' \neq \bot\}$$

es decir que un estado está en post(p, S) si es el estado final de una computación finita de S que arranca desde un estado inicial que satisface p. Y se define la precondición liberal más débil de la siguiente manera:

$$pre(S, q) = \{\sigma \mid \forall\sigma': val(\pi(S, \sigma)) = \sigma' \neq \bot \longrightarrow \sigma' \models q\}$$

es decir que un estado está en pre(S, q) si es el estado inicial a partir del cual se obtiene, por la ejecución de S, si termina, un estado final que satisface q. Probar:

a) $\{p\}\ S\ \{q\} \leftrightarrow post(p, S) \subseteq \{\sigma \mid \sigma \models q\}$
b) $\{p\}\ S\ \{q\} \leftrightarrow \{\sigma \mid \sigma \models p\} \subseteq pre(S, q)$