Lenguaje While original

- En While original se accede y se modifica el valor de las variables a través del estado $s \in State = (Var \to \mathbb{Z})$. Ello supone que absolutamente toda variable tiene un valor en cualquier momento.

Ejercicio 1 : Modificar la semántica para que podamos capturar que los "programas" inician su ejecución en un estado en el que todas las variables (al menos las que se manejen en él) porten de valor 0.

Ejercicio 2 : Modificar la semántica para que las variables puedan estar indefinidas. Estudiar con detalle qué connotaciones conllevaría dicha extensión. Evidentemente, ciertos programas (o por el momento, simplemente ciertas instrucciones) en ciertos estados darán lugar a error. Indicar cómo podríamos capturar estos errores con nuestra definición semántica y en qué casos podríamos garantizar que no se producirán con total seguridad. Tratar de definir una "semántica abstracta parcial" que nos indique que determinadas instrucciones (detectando "lo más posible") nunca producirán errores aunque se ejecuten sobre el estado en el que todas las variables están (inicialmente) indefinidas.

Ejercicio 3 : Versionar el Ejercicio 1, partiendo ahora del estado en el que todas las variables están inicialmente indefinidas. Demostrar, en el marco que resulte adecuado para ello, que si una variable deja de estar indefinida, jamás podrá volver a estarlo. Como corolario, si a lo largo de su ejecución se inicializan (sin errores previos) todas las variables

de un programa, a partir de que ello sucede nunca podrán producirse errores, ¿ cómo capturarías ese hecho en tu semántica. ¿ Es posible que una ejecución de un programa termine felizmente, sin sin embargo haber inicializado todas las variables que maneja ? ¿ Podría ello ocurrir incluso fuera cual fuera el estado sobre el que lo ejecutaramos ? ¿ Cuándo ? ¿ Qué supondría tal suceso ? ¿ Se te ocurre "algo interesante" que "convendría" hacerles a estos programas (instrucciones) ?

Bloques : declaraciones con efecto local

- Se declaran variables que se superponen al efecto del estado "global".
- Una vez declarada la variable "se deja de ver" la original, pero sólo hasta que "salimos" del bloque, momento en que recuperamos el valor "congelado" de la variable global.
- Los bloques se pueden anidar "redeclarando" algunas variables, dando lugar a una "pila escondida" de valores "congelados" : cada nueva declaración "congela" a la versión vigente de la variable y pasamos a manejar la nueva. Al terminar un bloque, se "destruyen" las variables declaradas en él, "recuperandose", con su valor anterior, la última versión congelada de cada una.

Ejercicio 4 : Versionar el Ejercicio 3 de modo que cada "programa" sólo pueda manejar inicialmente las variables declaradas en su cabecera.

Procedimientos : se pueden declarar también en un bloque

- Como las variables en los bloques, los procedimientos sólo se pueden utilizar dentro del bloque que contiene su declaración, pero ello incluye todos los cuerpos de los procedimientos declarados en el bloque, "antes y después" de cada uno.