

Ejercicios

II.5. EL TIPO DE DATOS DE LOS PROGRAMAS-WHILE

PRERREQUISITOS:

- Conocer la implementaciones de los programas-while y las operaciones para manipularlos
- Entender y diferenciar los conceptos de código (de un programa) e índice (de una función while-computable)

PROBLEMAS:

A. PARA FAMILIARIZARSE CON LA IMPLEMENTACIÓN DE LOS PROGRAMAS-WHILE

1. Teniendo en cuenta la implementación de los programas-while sobre el alfabeto $\Sigma = \{a,b\}$ ¿cuál es el programa P_{aabaab} ? ¿Y P_{72} ? ¿y P_{38755} ? ¿Y qué palabra representa al siguiente programa?

```
X0 := ε;  
if cara?(X1) then X2 := ε; X0 := cdr(X0); end if;  
while nonem?(X1) loop X0 := consb(X0);end loop;
```

¿Puede existir más de una palabra que, interpretada como programa, coincida con el anterior? ¿Por qué?

2. Suponiendo que los programas-while se han definido sobre el alfabeto $\{0,1\}$, busca índices para:
 - a) la función identidad (al menos dos)
 - b) la función constante abc (al menos dos)
 - c) la función vacía
 - d) la función $f(x,y) = \text{primero}(x) \bullet \text{cdr}(y)$
3. Encuentra un programa while P tal que $\varphi_P = C_A$ con $A = \{x: |x|_a > 0\}$.
 - a) Busca un índice para φ_P , siendo el alfabeto $\Sigma = \{a,b\}$.
 - b) Considera ahora el alfabeto $\Sigma = \{a\}$ ¿serviría el mismo programa? En caso afirmativo ¿su codificación seguiría siendo la misma? ¿Por qué?

Si no sirve el mismo programa busca otro que sí sirva e indica cuál sería su código.

B. PARA ACOSTUMBRARSE A MANIPULAR LOS OBJETOS DEL TIPO DE DATOS PROGRAMA-WHILE

4. Demuestra que las siguientes funciones y predicados son while-computables y while-decidibles respectivamente:
- a) $\text{vars_distintas?}: W \times N \rightarrow B$ (nos dice si en el texto del programa aparecen al menos tantas variables *distintas* como indica su segundo argumento)
 - b) $\text{num_asignaciones}: W \rightarrow N$ (cuenta el número de asignaciones que aparecen en el programa)
 - c) $\text{nivel_anidamiento}: W \rightarrow N$ (calcula el máximo nivel de anidamiento de instrucciones, considerando que este se incrementa cada vez que entramos en un if o un while y que se reduce al salir)
 - d) $\text{posible_ciclo?}: W \rightarrow B$ (nos avisa si la variable de control de alguno de los bucles del programa no es modificada por ninguna asignación dentro del while)
 - e) $\text{num_condiciones}: W \rightarrow N$ (cuenta el número de operaciones de consulta del texto del programa)
 - f) $\text{num_vars_inmutables}: W \rightarrow N$ (cuenta el número de variables que aparecen en el programa pero no aparecen en la parte izquierda de ninguna asignación)
 - g) $\text{quitaypon?}: W \rightarrow B$ (determina si un programa-while contiene dos asignaciones seguidas superfluas de la forma $XI := \text{cons}_a(XI); XI := \text{cdr}(XI);$)
5. Sea $\psi: \Sigma^* \rightarrow \Sigma^*$ una función while-computable, y e un índice asociado a ella. ¿Cuáles de las siguientes afirmaciones son ciertas, cuáles no y por qué?
- a) e es el único índice de ψ
 - b) e es el menor índice de ψ
 - c) e pertenece al dominio de ψ
 - d) e pertenece al rango de ψ
 - e) $W_e = \{ x \in \Sigma^*: \psi(x) \cong \varphi_e(x) \}$
 - f) $W_e = \{ x \in \Sigma^*: \psi(x) = \varphi_e(x) \}$
 - g) $W_e \subseteq R_e$
 - h) $R_e \subseteq W_e$
6. Sabemos que dado un programa x existen infinitos programas diferentes que computan la misma función φ_x (a los que llamamos índices de dicha función). Escribe tres macroprogramas P , Q y R que hagan precisamente eso: cuando reciban un programa x deben devolver otro programa equivalente a x . Se entiende que los métodos que usen P , Q y R deben ser sustancialmente distintos.

C. PARA TRABAJAR CON CONJUNTOS DE ÍNDICES

7. Encuentra uno o varios programas-while que pertenezcan y otro u otros que no pertenezcan a cada uno de los siguientes conjuntos (puedes escribir macroprogramas cuya expansión sería el programa while buscado):

- a) $\{ i: \forall j (\varphi_i(j) > j) \}$ b) $\{ i: \forall j (\varphi_i(j) \leq \varphi_i(j+1)) \}$
c) $\{ i: \forall j (\varphi_i(j) = \varphi_i(2*j)) \}$ d) $\{ i: \forall j \varphi_i(j) = \varphi_i(2*j) \wedge R_i \text{ es infinito} \}$
e) $\{ i: W_i \subseteq R_i \}$ f) $\{ i: W_i = R_i \}$
g) $\{ i: R_i \subset W_i \}$ h) $\{ i: W_i \text{ infinito} \wedge W_i \cap R_i = \emptyset \}$
i) $\{ i: W_i \cup R_i = \Sigma^* \wedge W_i \cap R_i = \emptyset \}$ j) $\{ i: W_i \text{ finito} \wedge R_i \text{ finito} \}$
k) $\{ i: i \in W_i \}$ l) $\{ i: i \in R_i \}$
m) $\{ i: \forall j (j \in W_i \leftrightarrow 2*j \in W_i) \}$ n) $\{ i: \forall j (j \in W_i \leftrightarrow j^R \in R_i) \}$

8. Dados los conjuntos VAC, FIN, RFIN, TOT, CONS, INJ, SUP, PERM, K y sus respectivos complementarios:

- a) Indica cuáles de ellos son finitos.
b) Indica cuáles de ellos contienen los índices de la función identidad.
c) Establece todas las relaciones de contenido que puedas entre ellos.

9. Suponiendo que estemos trabajando sobre el alfabeto $\{ a, b \}$, y definimos el conjunto $A = L((ba)^*)$. Demuestra que es while-decidible calculando la función característica correspondiente. Después encuentra un programa-while que pertenezca y otro que no pertenezca a cada uno de los siguientes conjuntos. Puedes escribir macroprogramas cuya expansión sería el programa while buscado (y usar para ello la función while-computable construida previamente):

- a) $\{ i: A \subset W_i \}$ b) $\{ i: R_i \subseteq A \}$
c) $\{ i: W_i = A \}$ d) $\{ i: R_i = A \wedge W_i = \bar{A} \}$
e) $\{ i: \bar{A} \subset W_i \}$ f) $\{ i: R_i \subseteq \bar{A} \}$
g) $\{ i: W_i = \bar{A} \}$ h) $\{ i: W_i = A \wedge R_i = \bar{A} \}$