Ejercicios

I.4. IMPLEMENTACIONES

PRERREQUISITOS:

- ❑ Entender el concepto de implementación
- ❑ Conocer las implementaciones de boléanos, naturales, pilas y vectores

PROBLEMAS:

A. FAMILIARIZARSE CON LA RELACIÓN DE ORDEN ENTRE PALABRAS ESTABLECIDA POR LA FUNCIÓN SIG Y CON LAS FUNCIONES DE CODIFICACIÓN

1. Indica cuáles serían las palabras $\mathbf{w_{47}}$, $\mathbf{w_{234}}$ y $\mathbf{w_{567}}$ según el alfabeto sea:

a) {**0**,**1**} **b)** {**a**,**b**,**c**,**d**,**e**,**f**}

2. Calcula el orden que ocupan las palabras **baca** y **cbafd** sobre el alfabeto {**a**,**b**,**c**,**d**,**e**,**f**}.

3. Sea el alfabeto {**a**,**b**}. Calcula las palabras que codifican las siguientes tuplas:

a) (**abaa**, ε) **b)** (**ba**, **aaa**)

c) (**bb**, **bb**, **aa**) **d)** (**bab**, ε , **a**)

4. Sea el alfabeto {**0**,**2**,**4**,**6**,**8**} ¿Qué palabras codifica la cadena **02680** suponiendo que:

a) es un par? **b)** es un trío? **c)** es una séxtupla?

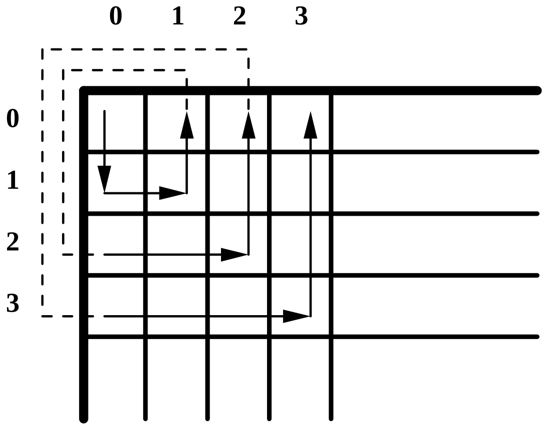
B. ENTENDER LOS MECANISMOS DE CODIFICACIÓN MEDIANTE FUNCIONES ALTERNATIVAS

5. Para variar, queremos definir una función de codificación alternativa, y se nos ocurre la siguiente:

$$\mathbf{enc^2}(x, y) = cod^2(sig(x), ant(y))$$

Sin embargo, la función ***enc*** no es una codificación adecuada de los pares de palabras. Discute por qué razones.

6. Supongamos que, en el momento de asignar códigos a los pares de palabras, en lugar de recorrer los pares de $\Sigma^* \times \Sigma^*$ en diagonales, lo hacemos utilizando el orden que se indica en la figura:



a) Define recursivamente la función que describiría ese recorrido

b) Razona por qué es una función de codificación adecuada.

c) Demuestra que es una función computable.

C. PARA ENTENDER LA RELACIÓN ENTRE DATOS IMPLEMENTADOS Y PALABRAS QUE LOS REPRESENTAN MEDIANTE LAS FUNCIONES DE INTERPRETACIÓN

7. Utilizando las funciones de interpretación vistas en clase y el alfabeto $\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$

a) ¿Cuál es el número natural, el valor booleano, la pila y el vector interpretado por las palabras **baaba**, **aacaac**, **aabca**, $\mathbf{w_{73}}$, $\mathbf{w_{36}}$, $\mathbf{w_{47}}$?

b) ¿Cuál es la palabra que sirve para representar la pila $<\mathbf{w_3}, \mathbf{w_6}, \mathbf{w_2}]$? ¿y la pila $<\mathbf{w_1}, \mathbf{w_3}, \mathbf{w_0}, \mathbf{w_1}]$? ¿y la pila <**bb**, **ac**, **b**]?

c) ¿Cuál es la palabra que interpreta el vector $(\mathbf{w_5}, \mathbf{w_3}, \mathbf{w_2})$? ¿y el vector $(\mathbf{w_7}, \mathbf{w_2}, \mathbf{w_1}, \mathbf{w_0})$? ¿y el vector (**ba**, **ab**, **cc**)?

8. Di si son ciertas o falsas las siguientes afirmaciones, razonando tu respuesta:

a) Para cualquier implementación de los booleanos la expansión del programa X0:= true es   X0 := '$\mathbf{a_1}$'.

b) Dado un tipo de datos **T** y una función de interpretación $\Im_{\mathbf{T}}$ definida para implementarlo, esta debe cumplir para todo **x** de **T** que: $\Im_{\mathbf{T}}^{-1}(\{\mathbf{x}\}) \neq \varnothing$

c) Supongamos que hemos definido dos funciones de interpretación estrictas sobre un tipo de datos **T** cualquiera. La composición de dichas funciones no puede ser otra función de interpretación.

9. Sea (**P**, P_vacía?, empilar, desempilar, cima) el tipo de datos pila.

a) Demuestra que la siguiente es una función de interpretación válida.

$$\Im^*_{\mathbf{P}}(\mathbf{w}) = \begin{cases} <] & \mathbf{w} = \varepsilon \\ \text{empilar}(\mathbf{x}, \Im^*_{\mathbf{P}}(\mathbf{y})) & \text{ant}(\mathbf{w}) = \text{cod}^2(\mathbf{x}, \mathbf{y}) \end{cases}$$

b) ¿Cuál es la pila que representan, según ella, w_{32} y w_{125}?

c) Implementa las funciones del tipo pila de acuerdo con $\Im^*_P$.

D. PARA APRENDER A IMPLEMENTAR OTROS TIPOS DE DATOS SENCILLOS

10. Implementa el tipo de datos de los **enteros,** considerando las operaciones constructoras necesarias para su implementación (es_cero?, succ, pred). Posteriormente demuestra la computabilidad de la operación resta $- : \mathbf{Z} \times \mathbf{Z} \rightarrow \mathbf{Z}$

11. Implementa el tipo de datos **cola**, **C,** con las operaciones que se indican a continuación.

C_vacía?: $\mathbf{C} \rightarrow \mathbf{B}$

frente: $\mathbf{C} -\!\!\rightarrow \Sigma^*$

encolar: $\Sigma^* \times \mathbf{C} \rightarrow \mathbf{C}$

desencolar: $\mathbf{C} -\!\!\rightarrow \mathbf{C}$

E. PARA ACOSTUMBRARSE A MANIPULAR LOS TIPOS DE DATOS BOOLEANO Y NATURAL

12. Demostrar que las siguientes funciones y predicados definidos sobre los números naturales y sobrelos booleanos son while-computables y while-decidibles, respectivamente:

a) **impar?:** $\mathbf{N} \rightarrow \mathbf{B}$

b) **. mod . :** $\mathbf{N} \times \mathbf{N} \rightarrow \mathbf{N}$ devuelve el resto de la división entera.

c) **. ** . :** $\mathbf{N} \times \mathbf{N} \rightarrow \mathbf{N}$ la función potencia: $x{**}y = x^y$.

d) **. --> . :** $\mathbf{B} \times \mathbf{B} \rightarrow \mathbf{B}$ el conectivo lógico implicación

e) **. xor . :** $\mathbf{B} \times \mathbf{B} \rightarrow \mathbf{B}$ el conectivo disyunción exclusiva

F. PARA ACOSTUMBRARSE A MANIPULAR LOS OBJETOS DE LOS TIPOS DE DATOS PILA Y VECTOR

13. Sea $\Sigma = \{\mathbf{a}, \mathbf{b}\}$ y considera la implementación de las pilas vista en teoría.

a) Sea **P** el programa que resulta de expandir

X0 := empilar(‘**ab**’, desempilar(X1));

Determina <u>la palabra</u> (no la pila) $\varphi_P(\mathbf{baab})$

b) Sin utilizar las funciones propias del tipo pila, implementa la función concat_pilas: $\mathbf{P} \times \mathbf{P} \rightarrow \mathbf{P}$, donde concat_pilas($\mathbf{p_1}$, $\mathbf{p_2}$) es la pila que resulta de juntar en su orden original los elementos tanto de $\mathbf{p_1}$ como de $\mathbf{p_2}$, de forma que los de $\mathbf{p_1}$ queden por encima de los de $\mathbf{p_2}$.

14. Sea $\Sigma = \{\mathbf{a}, \mathbf{b}\}$ y considera la implementación de los vectores vista en teoría.

a) Sea **P** el programa que resulta de expandir

X0 := modifica(X1, 2, acceso(X2, 2));

Determina la palabra (no el vector) φ_P(**abaababab**, **bbaaabaa**)

b) Sin utilizar las funciones propias del tipo vector, implementa la función intercambiar: $\mathbf{V} \times \mathbf{N} \times \mathbf{N} \rightarrow \mathbf{V}$, donde intercambiar(**v**, **i**, **j**) es el vector que resulta de intercambiar en v, v(i) con v(j).

15. Demuestra que las siguientes funciones y predicados son while-computables y while-decidibles, respectivamente:

a) suma_vector: $\mathbf{V} \rightarrow \mathbf{N}$ (calcula la suma de las compónentes de un vector).

b) prof_pila: $\mathbf{P} \rightarrow \mathbf{N}$ (calcula la profundidad de una pila).

c) igual_fondo_cima?: $\mathbf{P} \rightarrow \mathbf{P}$ (comprueba si coinciden fondo y cima de la pila).

G. PARA COMPRENDER OTRAS APROXIMACIONES DE LA TEORÍA DE LA COMPUTABILIDAD

16. Imagina que al inicio de la asignatura, al definir el lenguaje de los programas-while, hubiéramos utilizado como tipo de datos básico los naturales $\mathbf{N}$ en lugar de las cadenas de caracteres. La definición de los programas-while habría sido:

Programas básicos:

XI:=0;

XI:= succ(XJ);

Programas inductivos

- P Q
- **if** nonzero?(XI) **then** P **end if**;
- **while** nonzero?(XI) **loop** P **end loop**;

De manera similar habríamos incorporado macros para incluir las operaciones aritméticas habituales (+, -, $*$, /, mod, $**$) y los predicados que consideráramos necesarios (=, <, >, ≥, ≤) en nuestras macroexpresiones y macrocondiciones.

Después habríamos implementado otros tipos de datos, como booleanos, pilas, etc... Supón que queremos implementar como tipo de datos nuevo las palabras sobre el alfabeto {a,b}. Para ello partiríamos de la definición de una función de interpretación, que en este caso sería bastante obvia:

$$\mathfrak{I}_{\{\mathbf{a},\mathbf{b}\}^*}(0) = \varepsilon$$

$$\mathfrak{I}_{\{\mathbf{a},\mathbf{b}\}^*}(\mathbf{n}+1) = \mathbf{sig}(\mathfrak{I}_{\{\mathbf{a},\mathbf{b}\}^*}(\mathbf{n})) = \mathbf{w_i}$$

Donde **sig** y $\mathbf{w_i}$ tienen el significado usual.

a) Demuestra que las funciones $\text{snoc}_{\mathbf{a}}(\mathbf{x}) = \mathbf{x \bullet a}$ y $\text{snoc}_{\mathbf{b}}(\mathbf{x}) = \mathbf{x \bullet b}$ son computables.

b) Generaliza la solución para cualquier alfabeto $\{\mathbf{a_1}, \mathbf{a_2}, \ldots, \mathbf{a_n}\}$.

c) La computabilidad de las funciones $\text{cons}_{\mathbf{a}}$ y $\text{cons}_{\mathbf{b}}$ sería más complicada y necesitaría macros adicionales a las que se han mencionado ¿cuáles y para qué?