

## Hoja I.2

### SOLUCIÓN DE ALGUNOS APARTADOS DEL EJERCICIO 6

**6d**  $k(x) \cong$  sustituir en  $x$  cada aparición de **cba** por **c**

ESTRATEGIA: Utilizaremos una variable auxiliar  $X3$  para invertir y analizar  $x$  eliminando **ba** cuando va seguido de **c**. Después invertimos el resultado.

#### PROGRAMA

```

X2:= consa(X1); X2:= cdr(X2);
X3:= ε; X0:= ε;
while nonem?(X2) loop
  if cara?(X2) then X2:= cdr(X2); X3:= consa(X3); end if;
  if carb?(X2) then X2:= cdr(X2); X3:= consb(X3); end if;
  if carc?(X2) then
    X2:= cdr(X2); X3:= consc(X3);
    if carb?(X2) then
      X2:= cdr(X2);X3:= consb(X3);
      if cara?(X2) then
        X2:= cdr(X2);
        X3:= cdr(X3);
        -- porque aparece cba y ya habíamos añadido cb
      end if;
    end if;
  end if;
end loop;
while nonem?(X3) loop
  if cara?(X3) then X3:= cdr(X3); X0:= consa(X0); end if;
  if carb?(X3) then X3:= cdr(X3); X0:= consb(X0); end if;
  if carc?(X3) then X3:= cdr(X3); X0:= consc(X0); end if;
end loop;

```

## Hoja I.2

**6e**  $f'(x) \cong$  eliminar de  $x$  los símbolos aislados

**ESTRATEGIA:** Utilizaremos una variable auxiliar  $X4$  para comprobar si leído un símbolo hay otros iguales a la derecha. Construimos la salida invertida en  $X3$ .

### PROGRAMA

```

X2:= consa(X1); X2:= cdr(X2); X3:= ε; X0:= ε;
while nonem?(X2) loop
  X4:= ε;
  if cara?(X2) then X2:= cdr(X2); if cara?(X2) then
    X2:= cdr(X2); X3:= consa(X3); X3:= consa(X3); --no era aislado
    X4:= consa(X4);
    while nonem?(X4) loop --apuntamos las a's que siguen si las hay
      X4:= ε;
      if cara?(X2) then
        X2:= cdr(X2); X3:= consa(X3); X4:= consa(X4);
      end if;
    end loop;
  end if; end if;
  if carb?(X2) then X2:= cdr(X2); if carb?(X2) then
    X2:= cdr(X2); X3:= consb(X3); X3:= consb(X3);
    X4:= consa(X4);
    while nonem?(X4) loop --
      X4:= ε;
      if carb?(X2) then
        X2:= cdr(X2); X3:= consb(X3); X4:= consa(X4);
      end if;
    end loop;
  end if; end if;
  if carc?(X2) then X2:= cdr(X2); if carc?(X2) then
    X2:= cdr(X2); X3:= consc(X3); X3:= consc(X3);
    X4:= consa(X4);
    while nonem?(X4) loop
      X4:= ε;
      if carc?(X2) then
        X2:= cdr(X2); X3:= consc(X3); X4:= consa(X4);
      end if;
    end loop;
  end if; end if;
end loop;
while nonem?(X3) loop
  if cara?(X3) then X3:= cdr(X3); X0:= consa(X0); end if;
  if carb?(X3) then X3:= cdr(X3); X0:= consb(X0); end if;
  if carc?(X3) then X3:= cdr(X3); X0:= consc(X0); end if;
end loop;

```

## Hoja I.2

$$6f \quad \psi(x,y) \cong \begin{cases} \mathbf{x} & \text{nonem?}(\mathbf{x}) \\ \mathbf{y} & \mathbf{x} = \varepsilon \wedge \text{nonem?}(\mathbf{y}) \\ \perp & \text{c.c.} \end{cases}$$

**ESTRATEGIA:** Comprobaremos si  $\mathbf{x}$  e  $\mathbf{y}$  son vacías y utilizaremos una variable auxiliar  $X3$  para reflejar el resultado con el valor de su primer símbolo. Después tomamos la decisión apropiada según resulte dicho valor.

### PROGRAMA

```

X3:= consa(X1);  X3:= cdr(X3);
X4:= consa(X2);  X4:= cdr(X4);
X5:= ε;    X5:= consa(X5);
while nonem?(X3) loop
    X3:= ε;
    X0:= consa(X1);  X0:= cdr(X0);
    X5:= ε;
end loop;
-- (x≠ε → X0=x ∧ X5=ε) ∧ (x=ε → X5≠ε)
if cara?(X5) then
    while nonem?(X4) loop
        X4:= ε;
        X0:= consa(X2);  X0:= cdr(X0);
        X5:= ε;
    end loop;
-- (x=ε ∧ y≠ε → X0=y ∧ X5=ε) ∧ (x=ε ∧ y=ε → X5≠ε)
    while nonem?(X5) loop X5:= consa(X5); end loop;
end if;

```

## Hoja I.2

$$6g \quad \chi(x,y) \cong \begin{cases} x & \text{prefijo?}(\mathbf{abb}, x) \\ x^R & \text{prefijo?}(\mathbf{abc}, x) \\ \perp & \text{c.c.} \end{cases}$$

**ESTRATEGIA:** Comprobaremos si **abb** ó **abc** son prefijos de X1 y utilizaremos una variable auxiliar X3 para reflejar el resultado con el valor de su primer símbolo. Después tomamos la decisión apropiada según resulte dicho valor.

**PROGRAMA**

```

X2:= consa(X1); X2:= cdr(X2);
X3:= ε; X3:= consa(X3);
if cara?(X2) then
  X2:= cdr(X2);
  if carb?(X2) then
    X2:= cdr(X2);
    if carb?(X2) then X3:= consb(X3); end if;
    if carc?(X2) then X3:= consc(X3); end if;
  end if;
end if;
if cara?(X3) then
-- ni abb ni abc son prefijos de x
  while nonem?(X3) loop X3:= consa(X3); end loop;
end if;
if carb?(X3) then
-- abb es prefijo de x
  X0:= consa(X1); X0:= cdr(X0);
end if;
if carc?(X3) then
-- abc es prefijo de x
  X0:= ε;
  X2:= consa(X1); X2:= cdr(X2);
  while nonem?(X2) loop
    if cara?(X2) then X0:= consa(X0); end if;
    if carb?(X2) then X0:= consb(X0); end if;
    if carc?(X2) then X0:= consa(X0); end if;
    X2:= cdr(X2);
  end loop;
end if;

```

## Hoja I.2

$$6i \quad \psi(x,y) \equiv \begin{cases} \mathbf{a} & |x| \bmod 2 = 0 \wedge |y| \bmod 2 = 0 \\ \mathbf{y} & |x| \bmod 2 \neq 0 \\ \perp & \text{c.c.} \end{cases}$$

**ESTRATEGIA:** Utilizamos las variables X3 y X4 para manipular los datos  $x$  e  $y$ . Eliminamos símbolo a símbolo ambos datos, controlando la paridad resultante mediante dos variables X5 y X6. Cuando X5 comienza por  $\mathbf{a}$  llevaremos contado un número par de símbolos en X3, y cuando empieza por  $\mathbf{b}$ , un número impar. Análogamente con X6 y X4. Además necesitamos una variable X7 para simular alternativas del tipo **if\_then\_else** y conseguir que tras entrar en un **if** no sea posible entrar en el siguiente. Finalmente, tomamos la decisión apropiada según el valor de dichas variables.

### PROGRAMA

```

X3:= consa(X1); X3:= cdr(X3);
X5:= ε; X5:= consa(X5);
while nonem?(X3) loop
  X7:= ε; X7:= consa(X7);
  if cara?(X5) then X5:=consb(X5); X7:= ε; end if;
  if cara?(X7) then
    if carb?(X5) then X5:=consa(X5); end if;
  end if;
  X3:=cdr(X3);
end loop;
-- (cara?(X5) → |X1| mod 2 = 0) ∧ (carb?(X5) → |X1| mod 2 = 1)
X4:= consa(X2); X4:= cdr(X4);
X6:= ε; X6:= consa(X6);
while nonem?(X4) loop
  X7:= ε; X7:= consa(X7);
  if cara?(X6) then X6:=consb(X6); X7:= ε; end if;
  if cara?(X7) then
    if carb?(X6) then X6:=consa(X6); end if;
  end if;
  X4:=cdr(X4);
end loop;
-- (cara?(X6) → |X2| mod 2 = 0) ∧ (carb?(X6) → |X2| mod 2 = 1)
if cara?(X5) then if carb?(X6) then
  while nonem?(X5) loop X5:= consa(X5); end loop;
end if; end if;
X0:= ε; X0:= consa(X0);
if carb?(X5) then X0:= consa(X2); X0:= cdr(X0); end if;

```

## Hoja I.2

$$6i \quad \psi(x,y) \equiv \begin{cases} \mathbf{a} & |x| \bmod 2 = 0 \wedge |y| \bmod 2 = 0 \\ \mathbf{y} & |x| \bmod 2 \neq 0 \\ \perp & \text{c.c.} \end{cases}$$

**SEGUNDA STRATEGIA:** Eliminamos dos símbolos de  $x$  y  $\text{cdr}(x)$  simultáneamente, para controlar la paridad. Si cuando  $\text{cdr}(x)$  es vacío  $x$  no, entonces la longitud de  $x$  es impar. Análogamente para  $y$ . Además necesitamos una variable  $X7$  para simular alternativas del tipo **if\_then\_else** que nos permitan tomar la decisión apropiada según el análisis de  $x$  e  $y$ .

### PROGRAMA

```

X3:= consa(X1); X3:= cdr(X3);
X4:= cdr(X3);
while nonem?(X4) loop
    X3:=cdr(X3); X3:=cdr(X3);
    X4:=cdr(X4); X4:=cdr(X4);
end loop;
-- (X3 = ε → |X1| mod 2 = 0) ∧ (X3 ≠ ε → |X1| mod 2 = 1)
X5:= consa(X2); X5:= cdr(X5);
X6:= cdr(X5);
while nonem?(X6) loop
    X5:=cdr(X5); X5:=cdr(X5);
    X6:=cdr(X6); X6:=cdr(X6);
end loop;
-- (X5 = ε → |X2| mod 2 = 0) ∧ (X5 ≠ ε → |X2| mod 2 = 1)
X7:= ε; X7:= consa(X7);
while nonem?(X3) loop
    X0:= consa(X2); X0:= cdr(X0);
    X7:= ε;
end loop;
if cara?(X7) then
    X0:= consa(X2); X0:= cdr(X0);
    while nonem?(X5) loop
        X5:= consa(X5);
    end loop;
end if;

```

## Hoja I.2

### SOLUCIÓN DE ALGUNOS APARTADOS DEL EJERCICIO 7

$$7b - Q(x) \Leftrightarrow |x|_a \leq |x|_b$$

**ESTRATEGIA:** Meteremos las **a**'s de **x** en **X3** y sus **b**'s en **X4**. Después compararemos las longitudes de **X3** y **X4**, verificando que la primera palabra se cabe antes o al mismo tiempo que la segunda.

Inicializaremos **X0** con el valor **a** (true), y lo cambiaremos a  $\epsilon$  (false) cada vez que contamos una **a** de **X3**, por si acaso no existe la correspondiente **b** de **X4**. En cuanto confirmamos que dicha **b** existe restauramos el valor de **X0** hasta la siguiente iteración.

```

X2:= consa(X1); X2:= cdr(X2);
while nonem?(X2) loop
    if cara?(X2) then X3:= consa(X3); end if;
    if carb?(X2) then X4:= consb(X4); end if;
    X2:= cdr(X2);
end loop;
X0:=  $\epsilon$ ; X0:= consa(X0);
while nonem?(X3) loop
    X3:= cdr(X3);
    X0:=  $\epsilon$ ;
    if carb?(X4) then X4:= cdr(X4); X0:= consa(X0); end if;
end loop;

```

## Hoja I.2

$$7b \quad Q(x) \Leftrightarrow |x|_a \leq |x|_b$$

**SEGUNDA ESTRATEGIA:** Utilizar una variable auxiliar X3 para guardar la diferencia de **a**'s y **b**'s que hay en **x**, de manera que X3 contendrá **a**'s si  $|x|_a > |x|_b$ , contendrá **b**'s si  $|x|_b > |x|_a$  y X3 estará vacía cuando  $|x|_b = |x|_a$ . Además necesitamos una variable X4 para simular alternativas del tipo **if\_then\_else**.

Inicializaremos X0 con el valor **a** (true), y lo cambiaremos a  $\epsilon$  (false) si **x** contiene más **b**'s que **a**'s, es decir si X3 contiene **b**'s.

```

X2:= consa(X1); X2:= cdr(X2);
while nonem?(X2) loop
  X4:=  $\epsilon$ ; X4:= consa(X4);
  if cara?(X2) then
    if carb?(X3) then X3:= cdr(X3); X4:=  $\epsilon$ ; end if;
  --else
    if cara?(X4) then X3:= consa(X3); end if;
  end if;
  if carb?(X2) then
    if cara?(X3) then X3:= cdr(X3); X4:=  $\epsilon$ ; end if;
  --else
    if cara?(X4) then X3:= consb(X3); end if;
  end if;
  X2:= cdr(X2);
end loop;
X0:=  $\epsilon$ ; X0:= consa(X0);

```

```

if carb?(X3) then X0:=  $\epsilon$ ; end if;

```



## Hoja I.2

$$7f \quad Q(x) \Leftrightarrow \exists n, m > 0 \ x = (ab)^n aaa(ab)^m$$

**ESTRATEGIA:** Hemos de comprobar que en la palabra aparecen símbolos concretos por lo que inicializamos X0 a  $\varepsilon$  (false) y hasta que no comprobamos que es así no modificamos X0 a true.

```

X0:=  $\varepsilon$ ;
if cara?(X1) then
  X1:=cdr(X1);
  if carb?(X1) then --hemos comprobado que empieza por ab
    X1:=cdr(X1);
  --eliminamos también las siguientes apariciones de ab
  X3:= consa(X3);
  while nonem?(X3) loop
    X4:= consa(X1); X4:=cdr(X4);
    X3:=  $\varepsilon$ ;
    if cara?(X4) then X4:=cdr(X4);
      if carb?(X4) then
        X1:=cdr(X1); X1:=cdr(X1); X3:= consa(X3);
      end if;end if;
    end loop;
  if cara?(X1) then X1:=cdr(X1);
    if cara?(X1) X1:=cdr(X1);
      then if cara?(X1) then
        X1:=cdr(X1);
        if cara?(X1) then X1:=cdr(X1);
          if carb?(X1) then
            X1:=cdr(X1); X0:= consa(X0);
          --hemos comprobado que continua con aaa.ab
          X3:= consa(X3);
          --eliminamos también las siguientes apariciones de ab
          while nonem?(X3) loop
            X4:= consa(X1); X4:=cdr(X4); X3:=  $\varepsilon$ ;
            if cara?(X4) then X4:=cdr(X4);
              if carb?(X4) then
                X1:=cdr(X1); X1:=cdr(X1); X3:= consa(X3);
              end if;end if;
            end loop;
          end if; end if;
        end if; end if; end if;
      end if; end if;
    while nonem?(X1) loop
      X0:=  $\varepsilon$ ; X1:=  $\varepsilon$ ; --X1 contiene otros símbolos que no guardan la estructura pedida
    end loop;

```

## Hoja I.2

$$\neg \exists x Q(x) \Leftrightarrow \exists y(x = y \bullet y)$$

### □ PRIMERA SOLUCIÓN:

**ESTRATEGIA:** Intentaremos descomponer  $x$  en dos mitades y compararlas. La variable  $X2$  servirá para copiar el dato  $x$  y manipularlo. Utilizaremos un bucle que, en cada iteración, elimina *dos símbolos* de  $X2$ , uno del principio (que guardará en  $X3$ , donde quedará la mitad izquierda del dato  $x$  invertida) y otro del final (que guardará en  $X4$ , donde quedará la mitad derecha).

#### PRIMERA FASE: DESCOMPONER X EN DOS MITADES

Inicializaremos  $X0$  con el valor **a** (true), y sólo lo cambiaremos a  $\varepsilon$  (false) si encontramos que  $Q(x)$  es falso.

Para traspasar el último símbolo de  $X2$  a  $X4$  hay algunas dificultades. Obviamente habrá primero que invertir  $X2$ , pero sin machacarla, porque necesitaremos su valor para la siguiente iteración. Por ello copiaremos  $X2$  en  $X5$  e invertiremos  $X5$  en  $X6$ .

Si  $|x|$  es impar debemos detectarlo, pues en ese caso  $Q(x)$  será falso. Usaremos una variable de control  $X7$  que nos indicará cuándo  $X2$  tiene un único elemento central.

#### SEGUNDA FASE: COMPARAR LAS DOS MITADES

Si  $X0=\varepsilon$ , ya se ha detectado que  $Q(x)$  es falso. En caso contrario, invertiremos  $X3$  en  $X2$  (que habrá quedado desocupada), y compararemos los contenidos de  $X2$  y  $X4$ .

**PROGRAMA**

```

X0:= ε; X0:= consa(X0);
X2:= consa(X1); X2:= cdr(X2);
X3:= ε; X4:= ε;
while nonem?(X2) loop -- X1 = X3R•X2•X4 ∧ |X3| = |X4| (salvo si X0 = ε y |X1| mod 2 = 1)
-- Primero quitamos el símbolo de la izquierda
  if cara?(X2) then X3:=consa(X3); end if;
  if carb?(X2) then X3:=consb(X3); end if;
  X2:= cdr(X2);
-- Después el de la derecha.
  X5:= consa(X2); X5:= cdr(X5); X6:= ε;
  X7:= ε; X7:= consa(X7);
  while nonem?(X5) loop
    X7:=ε;
    if cara?(X5) then X6:=consa(X6); end if;
    if carb?(X5) then X6:=consb(X6); end if;
    X5:=cdr(X5);
  end loop;
  if cara?(X7) then X0:=ε; end if; -- caso |x| es impar, fin del bucle pues X2 se ha terminado
  if cara?(X6) then X4:=consa(X4); end if;
  if carb?(X6) then X4:=consb(X4); end if;
--Reconstruimos X2 para poder repetir el proceso
  while nonem?(X5) loop
    X7:=ε;
    if cara?(X5) then X6:=consa(X6); end if;
    if carb?(X5) then X6:=consb(X6); end if;
    X5:=cdr(X5);
  end loop;
end loop;
-- X1 = X3R•X4 ∧ |X3| = |X4| (salvo si X0 = ε, pues entonces |X1| mod 2 = 1)
if cara?(X0) then
  while nonem?(X3) loop
    if cara?(X3) then X2:= consa(X2); end if;
    if carb?(X3) then X2:= consb(X2); end if;
    X3:= cdr(X3);
  end loop;
-- X1 = X2•X4 ∧ |X3| = |X4|
  while nonem?(X4) loop
    if cara?(X4) then if carb?(X2) then X0:= ε; end if; end if;
    if carb?(X4) then if cara?(X2) then X0:= ε; end if; end if;
    X4:= cdr(X4);
    X2:= cdr(X2);
  end loop;
end if;

```

## □ SEGUNDA SOLUCIÓN:

**ESTRATEGIA:** Intentaremos eliminar de  $x$  su primera mitad y comprobar si lo que queda es prefijo de  $x$ . La variable  $X2$  servirá para copiar el dato  $x$  y manipularlo. Quitaremos de  $X2$  su primera mitad (lo que quede será la segunda mitad). Para detectar cuándo hemos llegado exactamente a la mitad usaremos otra copia  $X3$  del dato  $x$ , y en ella quitaremos los símbolos *de dos en dos*.

**PRIMERA FASE: DESCOMPONER X EN DOS MITADES**

Inicializaremos  $X0$  con el valor  $a$  (true), y sólo lo cambiaremos a  $\varepsilon$  (false) si encontramos que  $Q(x)$  es falso.

Si  $|x|$  es impar debemos detectarlo, pues en ese caso  $Q(x)$  será falso. Usaremos una variable de control  $X4$  que nos indicará cuándo no podremos quitar el segundo símbolo de  $X3$ .

**SEGUNDA FASE: COMPARAR LAS DOS MITADES**

Haremos una copia de  $X1$  en  $X3$  (que ahora estará libre) y compararemos su comienzo con  $X2$ .

**PROGRAMA**

```

X0:= ε; X0:= consa(X0);
X2:= consa(X1); X2:= cdr(X2);
X3:= consa(X1); X3:= cdr(X3);
while nonem?(X3) loop
  X2:= cdr(X2); X3:= cdr(X3);
  X4:= ε; X4:= consa(X4);
  if cara?(X3) then X3:= cdr(X3); X4:= ε; end if;
  if cara?(X4) then
    if carb?(X3) then X3:= cdr(X3); X4:= ε; end if;
  end if;
  if cara?(X4) then X0:=ε; end if;
-- porque en este caso  $|x|$  es impar. Saldremos del bucle porque X3 se ha terminado
end loop;
-- sufijo?(X2, X1)  $\wedge$   $|X1| = 2*|X2|$  (salvo si  $X0 = \varepsilon$ , pues entonces  $|X1| \bmod 2 = 1$ )
if cara?(X0) then
  X3:= consa(X1); X3:= cdr(X3);
  while nonem?(X2) loop
    if cara?(X2) then if carb?(X3) then X0:= ε; end if; end if;
    if carb?(X2) then if cara?(X3) then X0:= ε; end if; end if;
    X2:= cdr(X2); X3:= cdr(X3);
  end loop;
end if;

```

## Hoja I.2

### SOLUCIÓN DEL EJERCICIO 8

#### restar 1 en binario

**ESTRATEGIA:** Utilizamos variables auxiliares para copiar (X2) e invertir la entrada (X3) y otra más (X4) como marca. Para calcular el anterior hay que cambiar el 1 más cercano al final por 0, y los 0's anteriores por 1's.

```

X0:= ε; X3:= ε;
X2:= ε; X2:= cons0(X1); X2:= cdr(X2);
X4:= ε; X4:= cons0(X4);
while nonem?(X2) loop
    if car0?(X2) then X3:= cons0(X3); end if;
    if car1?(X2) then X3:= cons1(X3); X4:= ε; end if;
    X2:=cdr(X2);
end loop;
if car0?(X4) then
    --la palabra solo contenía 0's y el anterior no está definido para este caso
    while nonem?(X4) loop
        X4:= cons0(X4);
    end loop;
end if;
X3:= ε; X4:= cons0(X4);
--cambiamos 0's por 1's y utilizamos X4 para detectar el primer 1
while nonem?(X4) loop
    X4:= ε;
    if car0?(X3) then
        X0:= cons1(X0);
        X3:=cdr(X3);
        X4:= cons0(X4);
    end if;
end loop;
--cambio 1 por 0
X0:= cons0(X0);
X3:=cdr(X3);
while nonem?(X3) loop
    if car0?(X3) then X0:= cons0(X0); end if;
    if car1?(X3) then X0:= cons1(X0); end if;
    X3:=cdr(X3);
end loop;

```

## Hoja I.2

### SOLUCIÓN DEL EJERCICIO 9

Aunque el lenguaje de los programas-while es suficientemente sencillo, en realidad aún podríamos simplificarlo más:

- a) Podemos apañarnos sin las instrucciones de la forma **XI:= $\epsilon$** ;

Esta instrucción asigna a la variable XI la palabra vacía, lo que también podemos conseguir eliminando cada uno de los símbolos que contenga la palabra hasta no dejar ninguno. Basta utilizar un bucle y la primitiva **cdr**:

```
while nonem?(XI) loop
  XI := cdr(XI);
end loop;
```

- b) Podríamos prescindir de la instrucción **if\_then** permitiendo a cambio el uso de los predicados **car<sub>s</sub>?** en los bucles **while\_loop**.

Buscamos un programa equivalente a **if car<sub>s</sub>?(XI) then P end if**; que utilice un bucle en lugar de un **if**.

Basta con utilizar una variable auxiliar para salir del bucle una vez ejecutado P. Tomamos una variable XN con N suficientemente grande para no interferir con el resto del programa y la inicializamos con el mismo valor que XI:

```
AUX:= conss(XI); AUX:=cdr(AUX);
while cars?(AUX)loop
  P
  AUX:=  $\epsilon$ ;
end loop;
```