# Hoja II.9

## SOLUCIÓN DE UN APARTADO DEL EJERCICIO 4

**4b-** $$g(x) \cong \begin{cases} \varphi_x(x)+1 & \varphi_x(x)\downarrow \\ 0 & \text{c.c.} \end{cases}$$

Suponemos que **g** es computable. Gracias a **g** disponemos de cierta información sobre lo que ocurre con cualquier programa de código **x** si se aplica sobre el propio **x** como dato. Basándonos en esta información podemos definir una función diagonal $\delta$ diferente de todas las funciones $\varphi_x$ en el punto del que tenemos información. Para establecer quién será $\delta$(**x**) para cada valor **x** tendremos en cuenta lo siguiente:

- Si **g**(**x**)$\neq$0 entonces sabemos que $\varphi_x$(**x**) converge y además que su valor es g(**x**)-1. Por tanto, podemos hacer que $\delta$ sea diferente de $\varphi_x$ en el punto **x** definiendo $\delta$(**x**) con un valor diferente, o usar el valor indefinido. Intentaremos hacer esto último.

- Si **g**(**x**)=0 la dificultad está en que no sabemos en qué caso de la función **g** nos encontramos. O bien $\varphi_x$(**x**) es igual a 1 y estamos en el primer caso, o bien $\varphi_x$(**x**) diverge. Pero sí sabemos que tiene que ser obligatoriamente una de las dos cosas. Por tanto podemos hacer que $\delta$ sea diferente de $\varphi_x$ en el punto **x** asignando a $\delta$ (**x**) cualquier valor diferente de 1, como por ejemplo 100.

El esquema quedaría ilustrado en la siguiente tabla-ejemplo

| | **g(x)** | **0** | **25** | **28** | **39** | **0** | **0** |
|---|---|---|---|---|---|---|---|
| $\delta$ | **x** | $\varphi_0$ | $\varphi_1$ | $\varphi_2$ | $\varphi_3$ | $\varphi_4$ | $\varphi_5$ |
| **100** | 0 | $\perp \vee 1$ | | | | | |
| $\perp$ | 1 | | 15 | | | | |
| $\perp$ | 2 | | | 18 | | | |
| $\perp$ | 3 | | | | 29 | | |
| **100** | 4 | | | | | $\perp \vee 1$ | |
| **100** | 5 | | | | | | $\perp \vee 1$ |

De esta forma :

- $\delta(0)$ permite diferenciar a $\delta$ de $\varphi_0$, ya que $\delta(0) = 100$ y $\varphi_0(0) = 1$ ó $\varphi_0(0)\uparrow$
- $\delta(1)$ permite diferenciar a $\delta$ de $\varphi_1$, ya que $\delta(1)\uparrow$ y $\varphi_1(1) = 15$
- $\delta(2)$ permite diferenciar a $\delta$ de $\varphi_2$, ya que $\delta(2)\uparrow$ y $\varphi_2(2) = 18$

En general, $\delta(\mathbf{x})$ permite diferenciar a $\delta$ de $\varphi_{\mathbf{x}}$, ya que cuando $\varphi_{\mathbf{x}}(\mathbf{x})\downarrow$ con un valor distinto de 10, $\delta(\mathbf{x})\uparrow$, y cuando $\varphi_{\mathbf{x}}(\mathbf{x})\uparrow$ o $\varphi_{\mathbf{x}}(\mathbf{x})=10$ entonces $\delta(\mathbf{x})$ es 100

Definimos formalmente la función:

$$\delta(\mathbf{x}) \cong \begin{cases} 100 & \mathbf{f}(\mathbf{x}) = 0 \\ \perp & \text{c.c.} \end{cases}$$

No hay dificultad en comprobar que $\delta$ es computable toda vez que hemos supuesto la computabilidad de **f**:

```
X0 := 100;
if  f(X1) = 20  then X0 :=  ⊥ ; end if;
```

Una vez expandidas sus macros este programa se convertirá en un programa while, cuyo código será un índice **e** de la función $\delta$, es decir, que $\delta \cong \varphi_{\mathbf{e}}$.

Veamos que, tal y como hemos razonado la construcción de $\delta$, ese índice es imposible, porque obtenemos una contradicción si aplicamos la función $\delta$ en el punto **e**.

$$\delta(\mathbf{e})=100 \overset{\text{def. }\delta}{\Rightarrow} f(\mathbf{e})=0 \overset{\text{def. f.}}{\Rightarrow} \varphi_{\mathbf{e}}(\mathbf{e})\uparrow \vee \varphi_{\mathbf{e}}(\mathbf{e})=1 \overset{\delta\cong\varphi_e}{\Rightarrow} \delta(\mathbf{e})\uparrow \vee \delta(\mathbf{e})=1 \Rightarrow \neg\delta(\mathbf{e})=100$$

$$\neg\delta(\mathbf{e})=100 \overset{\text{def. }\delta}{\Rightarrow} \delta(\mathbf{e})\uparrow \overset{\text{def. }\delta}{\Rightarrow} f(\mathbf{e})\neq 0 \overset{\text{def. f.}}{\Rightarrow} \varphi_{\mathbf{e}}(\mathbf{e})\downarrow \overset{\delta\cong\varphi_e}{\Rightarrow} \delta(\mathbf{e})\downarrow \overset{\text{def. }\delta}{\Rightarrow} \delta(\mathbf{e}) = 100$$

La contradicción en el punto **e** surge de la única hipótesis que hemos hecho, la de suponer que **f** era computable.

# Hoja II.9

## SOLUCIÓN DEL EJERCICIO 6

$$\mathbf{f}(\mathbf{x}) = \begin{cases} \text{true} & \mathbf{x} \in \text{TOT} \wedge R_{\mathbf{x}} \subseteq \mathbf{P} \\ \text{false} & \text{c.c.} \end{cases}$$

Vamos a aplicar la fase de planificación del método para ver qué tipo de funciones podrían servir como funciones de diagonalización. Debemos suponer que la función **f** es computable, ya que basándonos en ello se construye la función diagonal **d** distinta de todas las funciones computables.

Si **f** es computable, dado un programa cualquiera de código **x**, podemos preguntar por f(**x**), y siempre obtendremos respuesta. Podemos observar que tenemos distinto grado de información según si la respuesta es true o no:

Si f(**x**) = false la información que obtenemos sobre **x** es mínima: $\varphi_{\mathbf{x}}$ puede ser no total y devolver siempre un par (para los casos en que la función esté definida) o puede ser total pero devolver un impar en algunos casos o puede ser no total y además con un rango que contenga algún impar. Por tanto, tendríamos muy difícil conseguir que **d** sea diferente de $\varphi_{\mathbf{x}}$ en un punto concreto. La única forma de diferenciarnos sería construyendo una función globalmente diferente, es decir, una función que sí verifique el predicado de la función **f**: construiríamos una **d** total que y que devuelva siempre un valor par.

Si f(**x**) = true sabemos que $\varphi_{\mathbf{x}}$ converge para todos los datos y además que siempre devuelve un valor par. Ello nos da una información muy valiosa para conseguir que **d** sea diferente de $\varphi_{\mathbf{x}}$, ya que podemos elegir cualquier valor para diferenciarnos localmente. Lo más sencillo será elegir el de la diagonal, el punto **x**. En cuanto a la forma de diferenciarnos tenemos la restricción de que **d** ha de verificar el predicado de la función **f**.

En resumen, cualquier estrategia de construcción de la función diagonal ha de asignar a d(**x**) siempre un valor par.

Si ¬f(**x**) podría ser 2, 10, 2 ∗x o cualquier otro valor par.

Si f(**x**) como $\varphi_{\mathbf{x}}$ es total podemos preguntar por $\varphi_{\mathbf{x}}(\mathbf{x})$, entonces nuestra función **d** ha de devolver un valor par diferente, podría ser $\varphi_{\mathbf{x}}(\mathbf{x})+20$, $\varphi_{\mathbf{x}}(\mathbf{x})+2$, $\varphi_{\mathbf{x}}(\mathbf{x})+4$ o cualquier otro valor par y diferente de $\varphi_{\mathbf{x}}(\mathbf{x})$.

Como podemos ver no todas las funciones del enunciado tienen estas características:

**$d_1$**: INCORRECTA. Es una función total ya que si f(**x**) es cierto $\varphi_{\mathbf{x}}(\mathbf{x})$ converge. Pero no siempre devuelve un valor par porque en caso de devolver x (cuando ¬f(**x**)) éste índice puede ser par o impar. Por tanto esta función no puede diferenciarse de aquellas que no verifican f.

**$\delta_2$**: INCORRECTA. Es una función no total ya que si f(**x**) es cierto **$\delta_2$**(**x**) diverge. Aunque siempre devuelve un valor par (cuando converge) nos permitiría diferenciarnos únicamente de las funciones que verifican f.

**$d_3$**: CORRECTA es una función total ya que si f(**x**) es cierto $\varphi_{\mathbf{x}}(\mathbf{x})$ converge y además siempre devuelve un valor par.

Podemos ver en la siguiente tabla-ejemplo la información que tendríamos y lo que haríamos siguiendo esta definición para valores concretos de **f**.

Como podemos ver en la tabla utilizamos los puntos de la diagonal para la diferenciación localmente de las funciones cuyos índices verifican **f**.

| | | **f(0)** | **¬f(1)** | **f(2)** | **¬f(3)** | **¬f(4)** | **¬f(5)** | |
|---|---|---|---|---|---|---|---|---|
| **x** | $d_3$ | $\varphi_0$ | $\varphi_1$ | $\varphi_2$ | $\varphi_3$ | $\varphi_4$ | $\varphi_5$ | **…** |
| **0** | 4 | 2 | | 30 | | | 4 | |
| **1** | 2 | 2 | | 500 | | | 8 | |
| **2** | 502 | 100 | | 500 | | | 10 | |
| **3** | 2 | 4 | ↑? | 6 | ↑? | ↑? | 3788 | … |
| **4** | 2 | 100 | | 8 | | | 24 | |
| **5** | 30 | 20 | | 16 | | | 28 | |
| | | | | … | | | | |

De esta forma y como se ve en los ejemplos, la función diagonal **$d_3$** es diferente de cada una de las que hay en la tabla en al menos un punto:

**$d_3$** es distinta de la función $\varphi_0$ en el punto 0, ya que **h**(0) = 2 y $\varphi_0(0)=4$

**$d_3$** es distinta de la función $\varphi_1$, ya que **$d_3$** es total y Ran(**$d_3$**) ⊆ P y sabemos que $\varphi_1$ no cumple esas dos propiedades porque **¬f(1)**.

**$d_3$** es distinta de la función $\varphi_2$ en el punto 2, ya que **$d_3$**(2) = 502 y $\varphi_2(2)=500$

**$d_3$** es distinta de la función $\varphi_3$, ya que **$d_3$** es total y Ran(**$d_3$**) $\subseteq$ P y sabemos que $\varphi_3$ no cumple esas dos propiedades porque **$\neg f(3)$**.

Además de acuerdo a la hipótesis de computabilidad de **f** tenemos que **$d_3$** será computable como muestra el programa que figura a continuación

```
if f(X1) then
        X0 := Φ(X1, X1)+2;
else X0 := 2;
end if;
```

**$d_4$**: INCORRECTA. Es una función que no verifica las condiciones del predicado **f** ya que si $\neg$f(**x**) puede ocurrir que $\varphi_x(x)$ diverja o devuelva un valor impar (también podría ser par pero no podemos asegurarlo) por lo que **$d_4$**(**x**) también podría diverger o ser par o impar. Además en el caso de que se verifique f(x) tampoco aseguramos la diferenciación local porque $\varphi_x(x)$ también podría ser el par 10.

**$d_5$**: INCORRECTA. Es una función total que siempre devuelve un valor par pero no es computable. Según la definición cuando $\neg$f(**x**) hemos de devolver $\varphi_x(x)+4$ en los casos en que $\varphi_x(x) \bmod 2 = 0$ y 10 en el otro caso. Es imposible poder hacer un programa que distinga estos dos casos y siempre devuelva un valor, porque si $\varphi_x(x)$ diverge según la definición **$d_5$**(**x**) = 10; pero f(x) no nos da información de lo que pasa en cada punto y no tenemos forma de saberlo salvo que lo comprobemos ejecutándolo y en caso de divergencia entonces no podríamos devolver 10.

# Hoja II.9

## SOLUCIÓN DE UN APARTADO DEL EJERCICIO 9

**9b-** $A = \{(\mathbf{x},\mathbf{y},\mathbf{z}): \varphi_{\mathbf{x}}(\mathbf{y}) = \mathbf{y}^{\mathbf{z}}\}$

El primer paso es ver qué información nos aportaría la función característica del conjunto A sobre cada una de las funciones computables: podemos obtener cierta información de lo que ocurre en cada uno de los puntos de cualquier función $\varphi_{\mathbf{x}}$ . Pero debemos preguntar por un **z** concreto por ejemplo 2 (valdría cualquiera o incluso podría ser variable para cada función **x** ó para cada punto **y**). Así $(\mathbf{x},\mathbf{y},2) \in A$ significaría que $\varphi_{\mathbf{x}}(\mathbf{y}) = \mathbf{y}^2$ y $(\mathbf{x},\mathbf{y},2) \notin A$ significaría que $\varphi_{\mathbf{x}}(\mathbf{y}) \neq \mathbf{y}^2$ o bien que $\varphi_{\mathbf{x}}(\mathbf{y})\uparrow$. Como tenemos información local de lo que ocurre en cada uno de los puntos, para aplicar el método de diagonalización en este caso basta con saber lo que ocurre en el punto **x** de cada función $\varphi_{\mathbf{x}}$.

Por tanto nos basta con una particularización de la función característica del conjunto A sobre la que aplicar directamente el método de diagonalización, la función

$$f(\mathbf{x}) \cong \begin{cases} \text{true} & \varphi_{\mathbf{x}}(\mathbf{x}) = \mathbf{x}^2 \\ \text{false} & \text{c.c.} \end{cases}$$

Si el conjunto A fuera decidible su función característica $C_A(\mathbf{x},\mathbf{y},\mathbf{z})$ sería computable, por lo que $f(\mathbf{x}) = C_A(\mathbf{x},\mathbf{x},\mathbf{2})$ también lo sería. Veamos que f no es computable y tendremos como consecuencia que el conjunto dado no es decidible.

Intuitivamente podemos pensar que la función f no es computable ya que en el caso de que la función de índice **x** diverja en el punto **x** no podremos dar una respuesta, porque para ello debemos comprobar que el programa de $\varphi_{\mathbf{x}}$ no termina. La función **f** nos proporciona información sobre lo que ocurre en determinados puntos de cada función. Según si el resultado de **f**(**x**) es true o false sabemos si el valor que devuelve la función $\varphi_{\mathbf{x}}$ en el punto **x** es $\mathbf{x}^2$ ó no.

Para construir una función $\boldsymbol{\delta}$ que sea distinta de todas las funciones $\varphi_{\mathbf{i}}$, debe diferenciarse de cada una de ellas en el punto del que tenemos información. En la tabla que muestra toda la información que tenemos hemos sombreado dichos puntos, y vemos que en este caso coincide con la *diagonal*. Para que $\boldsymbol{\delta}$ sea diferente de $\varphi_{\mathbf{0}}$ debe hacerlo en el punto **0** y como además sabemos que la función en ese punto vale **0**, haremos que la nueva función diverja (o converja con otro valor diferente). Para que sea diferente de la función $\varphi_1$ debe serlo en el punto **1**, que es de donde tenemos información, y como además sabemos que la función $\varphi_1$ en ese punto no es 1, (bien porque diverge o bien porque toma un valor distinto de 1) podemos hacer que la nueva función **δ** tome el valor 1. Y así sucesivamente, cada función $\varphi_i$ nos dice por medio de **f** cómo definir **δ**(**i**). Tenemos cómo definir **δ** para todas las entradas.

| | | **f(0)** | **¬f(1)** | **f(2)** | **f(3)** | **¬f(4)** | **¬f(5)** |
|---|---|---|---|---|---|---|---|
| | δ | $\varphi_0$ | $\varphi_1$ | $\varphi_2$ | $\varphi_3$ | $\varphi_4$ | $\varphi_5$ |
| **0** | ⊥ | 0 | | | | | |
| **1** | **1** | | **≠1 ó ⊥** | | | | |
| **2** | ⊥ | | | 4 | | | |
| **3** | ⊥ | | | | 9 | | |
| **4** | **16** | | | | | **≠16 ó ⊥** | |
| **5** | **25** | | | | | | **≠25 ó ⊥** |

Definimos formalmente la función **δ**:

$$\delta(\mathbf{x}) \cong \begin{cases} \mathbf{x}^2 & \neg f(\mathbf{x}) \\ \perp & f(\mathbf{x}) \end{cases}$$

Una vez definida la función y suponiendo que f es computable, tenemos que la función **δ** es computable (por ser una función definida por casos), por lo que existe un índice $\mathbf{e} \in \Sigma^*$, tal que $\boldsymbol{\delta} \cong \varphi_e$.

Veamos que, tal y como hemos razonado la construcción de **δ**, ese índice es imposible, porque obtenemos una contradicción si aplicamos la función **δ** en el punto **e**.

$$\boldsymbol{\delta}(\mathbf{e})\downarrow \overset{\text{def.}\,\delta}{\Leftrightarrow} \boldsymbol{\delta}(\mathbf{e})=\mathbf{e}^2 \overset{\text{def.}\,\delta}{\Leftrightarrow} \neg\mathbf{f}(\mathbf{e}) \overset{\text{def. f}}{\Leftrightarrow} \varphi_{\mathbf{e}}(\mathbf{e})\uparrow \text{ ó } \varphi_e(\mathbf{e}) \neq \mathbf{e}^2 \Leftrightarrow \neg(\varphi_{\mathbf{e}}(\mathbf{e})= \mathbf{e}^2) \overset{\delta\cong\varphi_e}{\Leftrightarrow} \neg\boldsymbol{\delta}(\mathbf{e})= \mathbf{e}^2 \overset{\text{def.}\,\delta}{\Leftrightarrow} \boldsymbol{\delta}(\mathbf{e})\uparrow$$

$$\boldsymbol{\delta}\,(\mathbf{e})\uparrow \overset{\text{def.}\,\delta}{\Leftrightarrow} \mathbf{f}(\mathbf{e}) \overset{\text{def. f}}{\Leftrightarrow} \varphi_{\mathbf{e}}(\mathbf{e})= \mathbf{e}^2 \overset{\delta\cong\varphi_e}{\Leftrightarrow} \boldsymbol{\delta}\,(\mathbf{e})= \mathbf{e}^2 \Leftrightarrow \boldsymbol{\delta}\,(\mathbf{e})\downarrow$$

Por tanto **δ** (**e**) no puede ni converger ni divergir (en los dos casos tenemos contradicción). Por tanto la contradicción en el punto **e** surge de la única hipótesis que hemos hecho, de suponer que **f** era computable.