

Conjuntos computables y Teorema de Rice

Lógica y Computabilidad*

2do cuatrimestre 2020

ÍNDICE

1. Definiciones	2
1.1. Conjuntos computables	2
1.2. El Teorema de Rice	2
2. Ejercicios de calentamiento	3
2.1. Conjuntos Computables	3
2.2. Teorema de Rice	3
3. Parte 1 - Conjuntos computables	5
4. Parte 2 - Funciones no computables	7

*Basado en apuntes de Julián Dabbah, Franco Frizzo, y M.Emilia Descotte

DEFINICIONES

CONJUNTOS COMPUTABLES

- La **función característica** de un conjunto A es la función

$$f_A(x) = A(x) = \begin{cases} 1 & \text{si } x \in A \\ 0 & \text{en otro caso} \end{cases}$$

- Un conjunto es **primitivo recursivo** si su función característica es primitiva recursiva.
- Un conjunto A es **computable** si su función característica es computable.
- Un conjunto A es **computable** sii es c.e. y co-c.e. '.
- Un conjunto A es **c.e.** si existe $g : \mathbb{N} \rightarrow \mathbb{N}$ tal que

$$A = \{x : g(x) \downarrow\} = \text{dom } g$$

- Un conjunto A es **co-c.e.** si $\bar{A} = A^c$ es c.e.
- Si dos conjuntos A y B son computables (o primitivos recursivos), también son computables (o primitivos recursivos) $A \cup B$, $A \cap B$, y \bar{A} .

EL TEOREMA DE RICE

- Si A es un conjunto de índices no trivial (i.e., $A \neq \emptyset$ y $A \neq \mathbb{N}$), entonces A no es computable.
- Un **conjunto de índices** de programas es un conjunto

$$C = \{x : \Phi_x^{(1)} \in \mathcal{C}\}$$

para alguna clase \mathcal{C} de funciones $\mathbb{N} \rightarrow \mathbb{N}$ parciales computables.

- Un conjunto C es un conjunto de índices si y solo si, para todo par de programas P y Q tales que $\Psi_P^{(1)} = \Psi_Q^{(1)}$ y $\#(P) \in C$, se cumple que $\#(Q) \in C$. (Ejercicio 9, práctica 3)

EJERCICIOS DE CALENTAMIENTO

CONJUNTOS COMPUTABLES

1. ¿A qué conjunto corresponde la siguiente función característica?

$$B(x) = \begin{cases} 1 & \text{si } \Phi_x^{(1)}(x) \downarrow \\ 0 & \text{en otro caso} \end{cases}$$

RESPUESTA: Corresponde al conjunto K .

2. ¿Es K un conjunto computable?

RESPUESTA: No, porque su función característica no lo es.

3. ¿Es $K \cap \{1, 2, 3, 4\}$ un conjunto computable?

RESPUESTA: Sí, porque es finito.

4. ¿Es verdad que un conjunto A es computable si y solo si su **complemento** \bar{A} es computable?

RESPUESTA: Sí. A computable implica \bar{A} computable y \bar{A} computable implica $\bar{\bar{A}} = A$ computable.

5. ¿Es $C = \{x : \Phi_x^{(1)}(x) \uparrow\}$ un conjunto computable? ¿Es primitivo recursivo?

RESPUESTA: No es computable, porque de serlo, su complemento (K) también sería computable. No es primitivo recursivo, porque de serlo, sería también computable.

6. Si $A \cup B$ (o $A \cap B$) es computable, ¿entonces A y B son computables?

RESPUESTA: ¡No!. \mathbb{N} es computable y $\mathbb{N} = A \cup \bar{A}$ para cualquier A , con lo cual, todos los conjuntos serían computables. De la misma manera, \emptyset es computable y $\emptyset = A \cap \bar{A}$.

7. Si A es un conjunto **finito**, ¿entonces es computable?

RESPUESTA: Sí. Más aún, es primitivo recursivo.

Sea $A = \{a_1, a_2, \dots, a_n\}$, la función característica de A es $A(x) = \begin{cases} 1 & \text{si } x = a_1 \\ 1 & \text{si } x = a_2 \\ \dots & \\ 1 & \text{si } x = a_n \\ 0 & \text{en otro caso} \end{cases}$

TEOREMA DE RICE

1. ¿Es $\{x : \Phi_x^{(1)} \text{ es una función constante}\}$ computable?

RESPUESTA: No, es un conjunto de índices **no trivial**: el de la clase de las funciones constantes.

2. ¿Es $\{x : \Phi_x^{(1)} \text{ es una función total}\}$ computable?

RESPUESTA: No, es un conjunto de índices **no trivial**: el de la clase de las funciones totales.

3. ¿Es $\{x : \Phi_x^{(1)} \text{ es una función parcial computable}\}$ computable?

RESPUESTA: Sí, es el conjunto \mathbb{N} , y su característica es la función constante 1.

4. ¿Es $\{x : \Phi_x^{(1)} \text{ es una función estrictamente creciente}\}$ computable?

RESPUESTA: No, es un conjunto de índices **no trivial**: el de la clase de las funciones estrictamente crecientes.

5. ¿Es $\{x : \Phi_x^{(1)} \text{ es una función primitiva recursiva}\}$ computable?

RESPUESTA: No, es un conjunto de índices **no trivial**: el de la clase de las funciones primitivas recursivas.

Moraleja: No es menor que el conjunto de índices que estamos considerando no sea trivial, y por lo tanto, es importante que lo demostremos.

En los ejemplos anteriores, basta con mencionar que la clase de funciones que estamos considerando no es ni la clase vacía ni la que contiene a todas las funciones parciales computables.

6. ¿Es computable el conjunto $A = \{x : \Phi_x^{(1)}(y) \text{ es par siempre que } y \text{ es par}\}$?

RESPUESTA: No, es un conjunto de índices no trivial: son los índices de las funciones que en los números pares devuelven un resultado par, y, además, no es trivial. Por ejemplo, la función identidad pertenece a A , con lo cual no es \emptyset , y la función constante 17 no pertenece a A , con lo cual no es \mathbb{N} .

7. ¿Es computable el conjunto $B = \{x : \text{el programa número } x \text{ termina si recibe como entrada un número par}\}$

RESPUESTA: No, es un conjunto de índices no trivial: son los índices de las funciones parciales computables cuyo dominio son exactamente los números pares. (No olvidar justificar que no es trivial, queda de tarea.)

8. ¿Es computable el conjunto $C = \{x : \text{el programa número } x \text{ tiene a lo sumo 5 instrucciones}\}$?

RESPUESTA: Sí, de hecho, es primitivo recursivo con $C(x) = \begin{cases} 1 & \text{si } |x + 1| \leq 5 \\ 0 & \text{si no} \end{cases}$.

¿Pero no es un conjunto de índices?!

No, no lo es, porque dada una función, hay programas que la computan con una cantidad de instrucciones tan grande como se quiera, y, por lo tanto, dada cualquier clase de funciones, no todos los números de los programas que computen alguna de sus funciones estarán en el conjunto.

9. ¿Es computable el conjunto $D = \{x : \Phi_x^{(1)}(y) = x \text{ para todo } y\}$?

RESPUESTA: No, ya sabemos que no lo es por los primeros ejercicios de la Práctica 3.

Entonces, ¿ D es un conjunto de índices? **No**, porque la propiedad expresada depende del programa que estemos considerando para computar una función dada.

Notar que, para cualquier función parcial computable, existen infinitos programas que la computan. Si tomamos $e_1 \neq e_2$ tales que $\Phi_{e_1}^{(1)} = \Phi_{e_2}^{(1)}$ y $e_1 \in D$, entonces claramente $e_2 \notin D$.

Moraleja: No todos los conjuntos que empiezan con $\{p : p \text{ es el número de un programa que } \dots\}$ o $\{x : \Phi_x \dots\}$ son conjuntos de índices. Igualmente, esto no implica que sean computables, ya que pueden ser no computables por otros motivos (e.g., K).

PARTE 1 – CONJUNTOS COMPUTABLES

Decidir si los siguientes conjuntos son o no computables.

- (a) $A = \{x : \Phi_x^{(1)}(5) = f(10)\}$, donde $f : \mathbb{N} \rightarrow \mathbb{N}$ es una función computable (total) fija. ¿Y si f fuera total, pero no computable?
- (b) $B = \{x : \Phi_x^{(1)}(y) = 0 \text{ para infinitos } y \in \mathbb{N}\}$
- (c) $C = \{x : \text{el programa con número } x \text{ tiene la instrucción } Y \leftarrow Y + 1 \text{ en algún lugar de su código}\}$

RESOLUCIÓN

- (a) En primer lugar, notemos que, como f es total, $f(10)$ es un número fijo. Esto nos permite escribir a A como un conjunto de índices, el correspondiente a la clase de funciones

$$\mathcal{C}_A = \{g : g(5) = f(10)\}$$

¿Alcanza esto para aplicar el teorema de Rice y afirmar que A no es computable? No, falta asegurarnos de que es un conjunto de índices *no trivial*.

- Por un lado, vemos que $A \neq \emptyset$, ya que podemos construir un programa que compute una función que esté en \mathcal{C}_A . Por ejemplo, si P_A es el programa:

$$Y \leftarrow f(10)$$

entonces claramente $\Psi_{P_A}^{(1)}(5) = f(10)$, y por lo tanto $\#(P_A) \in A$.

- Por otra parte, $A \neq \mathbb{N}$, dado que existen programas que computan funciones que no están en \mathcal{C}_A . Por ejemplo, si Q_A es el programa:

$$Y \leftarrow f(10) + 1$$

entonces $\Psi_{Q_A}^{(1)}(5) = f(10) + 1 \neq f(10)$, y por lo tanto $\#(Q_A) \notin A$.

Ahora sí, aplicando el teorema de Rice, podemos afirmar que A no es computable.

¿Y si f fuera total, pero no computable? En este caso, el razonamiento anterior sigue siendo perfectamente válido. Dado que $f(10)$ es un valor fijo, la clase \mathcal{C}_A está bien definida, incluso aunque no podamos computar cuál es dicho valor. También siguen siendo válidos los programas P_A y Q_A . Podemos afirmar entonces que A sigue siendo un conjunto de índices no trivial y, por lo tanto, no computable.

- (b) El conjunto B también es un conjunto de índices, el correspondiente a la clase de funciones

$$\mathcal{C}_B = \{g : g(x) = 0 \text{ para infinitos } x \in \mathbb{N}\}$$

Veamos que es no trivial:

- $B \neq \emptyset$, ya que existen programas que computan funciones que devuelven 0 en infinitos valores. Por ejemplo, si tomamos un programa P_B que compute la constante 0, entonces $\#(P_B) \in B$.
- $B \neq \mathbb{N}$, ya que podemos construir programas que computan funciones que devuelven 0 en a lo sumo finitos valores. Por ejemplo, si Q_B es un programa que computa la constante 1, entonces $\#(Q_B) \notin B$.

Luego, por el teorema de Rice, B no es computable.

- (c) C no solo es un conjunto computable, sino que es primitivo recursivo, ya que podemos escribir su función característica como una composición de funciones que sabemos que son primitivas recursivas. Determinar esta escritura queda como ejercicio, pero es muy similar a los ejercicios de codificación de programas que vimos en la práctica 2.

¿Por qué falló el teorema de Rice? El problema es que C no es un conjunto de índices. Podemos ver que la propiedad que caracteriza a C se refiere a programas y no a funciones. A modo de contraejemplo, estos dos programas:

$$P_C : \quad \begin{array}{l} Y \leftarrow Y + 1 \\ Y \leftarrow Y \div 1 \end{array}$$
$$Q_C : \quad [\text{el programa vacío}]$$

computan la misma función, la constante 0. Sin embargo, $\#(P_C) \in C$, mientras que $\#(Q_C) \notin C$.

PARTE 2 – FUNCIONES NO COMPUTABLES

Usando reducciones y el teorema de Rice, demostrar que las siguientes funciones no son computables:

- (a) $g_1(x, y) = \begin{cases} 1 & \text{si } \Phi_x^{(1)}(y) \text{ está definido y es par} \\ 0 & \text{en otro caso} \end{cases}$
- (b) $g_2(x, y) = \begin{cases} 1 & \text{si } \Phi_x^{(1)} \text{ y } \Phi_y^{(1)} \text{ computan la misma función} \\ 0 & \text{en otro caso} \end{cases}$
- (c) $g_3(x, y) = \begin{cases} 1 & \text{si } \Phi_x^{(1)}(y) = 25 \text{ y } \Phi_y^{(1)}(x) = x \\ 0 & \text{en otro caso} \end{cases}$

RESOLUCIÓN

- (a) Para poder aplicar el teorema de Rice, nos gustaría reducir a g_1 la función característica de algún conjunto de índices no trivial. Vemos que, fijando cualquiera de las dos variables, lo que queda es la función característica de un conjunto de naturales. Sin embargo, dado que x es la variable que aparece en el número de programa, la mejor candidata a fijar para obtener un conjunto de índices parece ser y .

Argumentamos, entonces, que si g_1 fuera computable, también lo sería, por ejemplo,

$$f_1(x) = g_1(x, 42) = \begin{cases} 1 & \text{si } \Phi_x^{(1)}(42) \text{ está definido y es par} \\ 0 & \text{en otro caso} \end{cases}$$

Ahora bien, f_1 es la función característica del conjunto

$$A_1 = \{x : \Phi_x^{(1)}(42) \text{ está definido y es par}\}$$

que es un conjunto de índices, el correspondiente a la clase de funciones

$$\mathcal{C}_{A_1} = \{g : g(42) \text{ está definido y es par}\}$$

Además, A_1 es no trivial: el número de cualquier programa que compute la constante 0 está en A_1 , por lo que $A_1 \neq \emptyset$, pero el número de cualquier programa que se indefina siempre no está en A_1 , por lo que $A_1 \neq \mathbb{N}$.

Por el teorema de Rice, f_1 no puede ser computable, por lo que g_1 tampoco lo es.

- (b) Vamos a probar que no es computable la igualdad de funciones, es decir, que dados dos programas, no es computable decidir si computan la misma función. Más aún, vamos a ver que decidir si un programa se comporta como una función dada no es computable. Para probar esto, fijamos y en el número de algún programa, digamos e , y sea P el programa tal que $\#P = e$. Luego, podemos definir:

$$g_e(x) = g_2(x, e) = \begin{cases} 1 & \text{si } \Phi_x^{(1)} \text{ computa la función } \Psi_P^{(1)} \\ 0 & \text{en otro caso} \end{cases}$$

Como siempre, tenemos que si g_2 fuera computable, g_e sería computable, y vamos a probar que esto no es posible. g_e es la función característica del conjunto de índices $A_2 = \{x : \Phi_x^{(1)} \text{ computa la función } \Psi_P^{(1)}\} = \{x : \Phi_x^{(1)} \in \mathcal{P}\}$, donde $\mathcal{P} = \{\Psi_P^{(1)}\}$ es la clase de funciones que contiene únicamente a $\Psi_P^{(1)}$ (y por lo tanto, su conjunto de índices no es trivial). Luego, por el Teorema de Rice, este conjunto no es computable, o lo que es lo mismo, g_e no es computable. Observemos que este argumento vale para cualquier e , y en particular, si lo fijáramos en el número de algún programa que computa la identidad, tendríamos que $g_{e_{id}}(x)$ es la función que decide si un programa con número x computa la función identidad, que ya vimos anteriormente que no es computable, aplicando el Teorema de la Recursión.

- (c) Igual que en los casos anteriores, nos gustaría poder reducir a g_3 la característica de algún conjunto de índices, fijando alguno de sus parámetros. El problema es que ahora ambos parámetros son interpretados como números de programa, por lo que ya no es tan sencillo decidir cuál de ellos fijar, ni resulta claro si podemos hacerlo para un valor arbitrario.

Sería bueno, para el parámetro que vayamos a fijar, hacer desaparecer de la condición de g_3 la parte en que se lo interpreta como número de programa. Dado que la condición es una conjunción, para lograrlo tenemos que hacer que esa parte sea verdadera para cualquier valor que tome el otro parámetro.

Elegimos fijar el parámetro y en algún valor e apropiado,¹ que determinaremos enseguida. Así, la condición de g_3 queda

$$\Phi_x^{(1)}(e) = 25 \text{ y } \Phi_e^{(1)}(x) = x$$

Queremos que la segunda parte de la conjunción sea verdadera para cualquier valor de x . Esto sucede si tomamos $e = \#(P)$, donde P es un programa que computa la función identidad. Para este e y para cualquier x ,

$$\Phi_x^{(1)}(e) = 25 \text{ y } \Phi_e^{(1)}(x) = x \quad \Leftrightarrow \quad \Phi_x^{(1)}(e) = 25$$

Podemos argumentar, entonces, que si g_3 fuera computable, también lo sería

$$f_3(x) = g_3(x, e) = \begin{cases} 1 & \text{si } \Phi_x^{(1)}(e) = 25 \\ 0 & \text{en otro caso} \end{cases}$$

f_3 es la función característica del conjunto

$$A_3 = \{x : \Phi_x^{(1)}(e) = 25\}$$

que es un conjunto de índices, el correspondiente a la clase de funciones

$$\mathcal{C}_{A_3} = \{g : g(e) = 25\}$$

Además, A_3 es no trivial: el número de cualquier programa que compute la constante 25 está en A_3 , por lo que $A_3 \neq \emptyset$, mientras que número de cualquier programa que compute la constante 26 no está en A_3 , por lo que $A_3 \neq \mathbb{N}$.

Aplicando el teorema de Rice, concluimos que f_3 no es computable, por lo que tampoco puede ser computable g_3 .

¹La demostración también puede completarse fijando el parámetro x para algún valor apropiado. Queda como ejercicio verificarlo.