Modelos de Computación y Complejidad
Grado en Ingeniería Informática. Tecnologías Informáticas

David Orellana Martín,
Mario de J. Pérez Jiménez
Departamento de Ciencias de la Computación e Inteligencia Artificial

Universidad de Sevilla

Soluciones de ejercicios y problemas

RECURSIVIDAD ENUMERABLE E INDECIDIBILIDAD

Curso 2021-2022

Ejercicio 1: Probar que es GOTO–computable la función parcial $f : \mathbb{N} - \to \mathbb{N}$ definida por:

$$f(x) = \begin{cases} 1 & \textbf{si } \varphi_x(x)\downarrow \\ \uparrow & \textbf{en otro caso} \end{cases}$$

Solución:

Para cada $(x_1, x_2) \in \mathbb{N}^2$ se verifica:

$$(x_1, x_2) \in G(f) \iff f(x_1) = x_2 \iff \varphi_{x_1}(x_1)\downarrow \wedge x_2 = 1 \iff \mathtt{HALT}(x_1, x_1) \wedge x_2 = 1$$

Así pues, el **predicado de pertenencia** al conjunto $G(f)$ está descrito como la **conjunción** de dos predicados:

- el predicado HALT (x, x) (que es es **p.d.**)
- el predicado de igualdad "$x = y$" (que es GOTO-computable).

Por tanto, el predicado de pertenencia citado es **p.d.**(ya que es la conjunción de dos predicados **p.d.**).

En consecuencia, el conjunto $G(f)$ es **r.e.** y, por tanto, la función parcial **f** es GOTO–computable.

Nota interesante: Obsérvese que f es, en realidad, la función característica parcial del conjunto $\mathcal{K}$ del problema de la parada. Además, la función f **sí** es GOTO-computable y, en cambio, su **dominio** **no** es un conjunto GOTO-computable (pues $\text{dom}(f) = \mathcal{K}$).

Ejercicio 2: **Probar que <u>no</u> es GOTO–computable la función <u>total</u> $f : \mathbb{N} \to \mathbb{N}$ definida por:**

$$f(x) = \begin{cases} \varphi_x(x) + 1 & \textbf{si } \varphi_x(x)\downarrow \\ 0 & \textbf{en otro caso} \end{cases}$$

Solución:

Supongamos que la función total f fuese GOTO-computable. Entonces existiría un número natural n tal que $\varphi_n^{(1)} = f$.

Vamos a ver que esto es imposible. Para ello, distingamos dos casos:

Caso 1: Supongamos que $\varphi_n^{(1)}(n)\downarrow$.

En este caso, por una parte, de la definición de f se deduce que $f(n) = \varphi_n^{(1)}(n) + 1$.

Por otra, al ser $\varphi_n^{(1)} = f$ resultaría que $\varphi_n^{(1)}(n) = f(n)$.

De ambas relaciones, se tendría que $\varphi_n^{(1)}(n) = f(n) = \varphi_n^{(1)}(n) + 1$: **contradicción**.

Caso 2: Supongamos que $\varphi_n^{(1)}(n)\uparrow$.

En este caso, por una parte, de la definición de f se deduce que $f(n) = 0$.

Por otra, del hecho de que $\varphi_n^{(1)} = f$ resultaría que $\varphi_n^{(1)}(n) = f(n)$.

De ambas relaciones, se tendría que $\varphi_n^{(1)}(n) = f(n) = 0$; es decir, $\varphi_n^{(1)}(n)\uparrow$.

Esto <u>contradice</u> lo que se ha supuesto de este caso.

Nota interesante: Obsérvese que esta función f **no** es GOTO-computable y, en cambio, su **<u>dominio</u>** **sí** es un conjunto GOTO-computable (pues $\mathrm{dom}(f) = \mathbb{N}$).

Ejercicio 3: Probar que existen funciones tales que:

(a) **No** son GOTO-computables y, en cambio, su **dominio** y su **rango** **sí** son conjuntos GOTO–computables.

(b) **Sí** son GOTO-computables y, en cambio, su **dominio** y su **rango** **no** son conjuntos GOTO–computables.

Solución:

(a) Cualquier predicado que **no** sea GOTO-computable satisface las condiciones exigidas (pues su dominio es $\mathbb{N}$ y su rango es un conjunto finito: $\{0, 1\}$).

Y recuérdese que, por ejemplo, el predicado HALT es **p.d.** pero **no** es GOTO-computable.

(b) Consideremos la función f de aridad 1 sobre $\mathbb{N}$ definida como sigue:

$$f(x) = \begin{cases} x & \text{si } \varphi_x(x)\downarrow \\ \uparrow & \text{si } \varphi_x(x)\uparrow \end{cases}$$

Para cada $(x, y) \in \mathbb{N}^2$ se verifica:

$$(x, y) \in G(f) \Longleftrightarrow \varphi_x(x)\downarrow \wedge\, y = x \Longleftrightarrow x \in \mathcal{K} \;\wedge\; y = x$$

Así pues, el **predicado de pertenencia** al conjunto $G(f)$ es **p.d.** y, por tanto, la función f **sí** es GOTO-computable.

En cambio, dom(f)=rang(f) $= \mathcal{K}$ que, obviamente, **no** son conjuntos GOTO-computables.

Ejercicio 4: Demostrar, por reducción al absurdo, que es vacío el conjunto

$$A = \{n \in \mathbb{N} : \varphi_n^{(1)} \textbf{ es total y } \forall x \in \mathbb{N}\ (x \in \mathcal{K} \Longleftrightarrow \varphi_n^{(1)}(x) = 0)\}$$

Solución: Supongamos que $A \neq \emptyset$. Entonces existiría un número natural n tal que $n \in A$; es decir, dicho número verificaría:

(1) La función $\varphi_n^{(1)}$ es total.

(2) $\forall x \in \mathbb{N}\ (x \in \mathcal{K} \Longleftrightarrow \varphi_n^{(1)}(x) = 0)$.

Luego, para cada número natural x se tendría lo siguiente:

$$\boxed{x \in \mathcal{K}} \overset{(2)}{\Longleftrightarrow} \varphi_n^{(1)}(x) = 0 \overset{def.}{\Longleftrightarrow} \boxed{(x, 0) \in G(\varphi_n^{(1)})}$$

Ahora bien, el conjunto $G(\varphi_x^{(1)})$ es GOTO-computable ya que la función $\varphi_n^{(1)}$ es <u>total</u> y GOTO-computable (véase la transparencia 32 del tema 2).

Por tanto, la relación $x \in \mathcal{K}$ de pertenencia al conjunto $\mathcal{K}$ sería un predicado GOTO–computable; es decir, el conjunto $\mathcal{K}$ sería GOTO-computable.

Lo que es una **contradicción**: se ha probado que $\mathcal{K}$ es **r.e.** pero **no** GOTO-computable.

Ejercicio 5: Probar que el conjunto

$A = \{\langle x, y\rangle \in \mathbb{N}$: el programa de código x para sobre y en, **exactamente**, un número impar de pasos$\}$

es **recursivamente enumerable**.

Solución:

Para cada número natural $a \in \mathbb{N}$ se verifica lo siguiente:

$a \in A \iff \exists x \exists y\ [x = l(a) \wedge y = r(a) \wedge \exists z\ (\mathtt{STEP}^{(1)}(y, x, z) \wedge \forall t < z\ (\neg \mathtt{STEP}^{(1)}(y, x, t)) \wedge z \textbf{ es impar})]$

Este predicado es la cuantificación existencial de un **predicado p.d.** que, a su vez, es la conjunción de otros tres:

- ⋆ El primero ($x = l(a)$) y el segundo ($y = r(a)$) son `GOTO`-computables.
- ⋆ El tercero ($\exists z\ (\mathtt{STEP}^{(1)}(y, x, z) \wedge \forall t < z\ (\neg \mathtt{STEP}^{(1)}(y, x, t)) \wedge z \text{ es impar})$) es un **predicado p.d.**(pues es una cuantificación existencial de la conjunción de tres predicados `GOTO`-computables).

Por tanto, el predicado $a \in A$ es **p.d.** ya que es la cuantificación existencial de un predicado **p.d.**

En consecuencia, el conjunto A es **r.e.**

Ejercicio 6: Sean A y B suconjuntos de $\mathbb{N}^k$, con $k \geq 1$, tales que A es **r.e.** y B es GOTO-computable. Demostrar que el conjunto $A - B = \{\vec{x} \in \mathbb{N}^k \mid \vec{x} \in A \wedge \vec{x} \notin B\}$ es **r.e.** ¿Puede asegurarse que el conjunto $B - A$ sea **r.e.**?

Solución:

Para probar que el conjunto $A - B$ es **r.e.** hemos de demostrar que la relación de pertenencia a dicho conjunto es un **predicado p.d.**

Veámoslo: para cada número tupla $\vec{x} \in \mathbb{N}^k$ se verifica:

$$\vec{x} \in A - B \Longleftrightarrow \vec{x} \in A \wedge \vec{x} \notin B$$

Ahora bien, la relación de pertenencia $\vec{x} \in A - B$ es la conjunción de dos predicados:

- El primero ($\vec{x} \in A$) es **p.d.** ya que A es un conjunto **r.e.**
- El segundo ($\vec{x} \notin B$) es GOTO-computable ya que B es GOTO-computable (la <u>negación</u> de un predicado GOTO-computable también es GOTO-computable).

En consecuencia, el predicado $\vec{x} \in A - B$ será **p.d.** Por tanto, $A - B$ será **r.e.**

A continuación, veamos que **<u>no</u> se puede asegurar** que el conjunto $B - A$ sea **r.e.**

En efecto: la relación de pertenencia a $B - A$ ($\vec{x} \in B - A \Longleftrightarrow \vec{x} \in B \wedge \vec{x} \notin A$) es la conjunción de dos predicados:

- El primero ($\vec{x} \in B$) es GOTO-computable ya que B es GOTO-computable.
- El segundo ($\vec{x} \notin A$) es la <u>negación</u> de un **predicado p.d.** ya que A es un conjunto **r.e.** y, por tanto, **<u>no se puede asegurar</u>** que sea **p.d.**

Así pues, **<u>no</u> se puede asegurar** que el conjunto $B - A$ sea **r.e.**

Ejercicio 7:

Justificar la veracidad o falsedad de las siguientes afirmaciones.

1. Existen conjuntos de números naturales que ni son `GOTO`-computables ni son recursivamente enumerables.
2. La negación de un predicado `GOTO`-computable es, también, un predicado `GOTO`-computable.
3. La negación de un predicado *parcialmente decidible* es, también, un predicado *parcialmente decidible*.
4. La cuantificación existencial de un predicado `GOTO`-computable es, también, un predicado `GOTO`-computable.
5. La cuantificación existencial de un predicado *parcialmente decidible* es, también, un predicado *parcialmente decidible*.
6. La cuantificación universal de un predicado `GOTO`-computable es, también, un predicado `GOTO`-computable.
7. La cuantificación universal de un predicado *parcialmente decidible* es, también, un predicado *parcialmente decidible*.
8. El dominio y la gráfica de una función `GOTO`-computable (**parcial** o **total**) son conjuntos recursivamente enumerables.
9. El conjunto $A = \{x : \text{ el programa de código } x \text{ para sobre } 2x\}$ es recursivamente enumerable.
10. El teorema de Rice permite demostrar que **ciertos** subconjuntos de $\mathbb{N}$ **no** son `GOTO`–computables.
11. El conjunto de índices de **una** función `GOTO`–computable de aridad 1, es un conjunto `GOTO`–computable.
12. El conjunto de índices de **la** función identidad en $\mathbb{N}$ es un conjunto `GOTO`–computable.

1. Existen conjuntos de números naturales que ni son `GOTO`-computables ni son recursivamente enumerables.

Respuesta: **Cierto**.

Justificación de la respuesta: Se ha probado en clase que el conjunto $\mathcal{K}$ del problema de la parada, no es `GOTO`-computable pero, en cambio, sí es **r.e.**

Veamos que su conjunto complementario, $\overline{\mathcal{K}}$, es un conjunto que ni es `GOTO`-computable ni es **r.e.** En efecto:

- ⋆ **El conjunto $\overline{\mathcal{K}}$ no es `GOTO`-computable** : caso contrario, si $\overline{\mathcal{K}}$ fuese `GOTO`-computable entonces también lo sería, $\overline{\overline{\mathcal{K}}}$; es decir, el conjunto $\mathcal{K}$ sería `GOTO`-computable. Lo que es una contradicción.
- ⋆ **El conjunto $\overline{\mathcal{K}}$ no es r.e.**: caso contrario, teniendo presente que $\mathcal{K}$ es un conjunto **r.e.**, del teorema del complemento (*un conjunto es* `GOTO`*-computable* **sii** *él y su complementario son, ambos,* **r.e.**) se deduciría que el conjunto $\mathcal{K}$ sería `GOTO`-computable. Lo que es una contradicción.

2. **La negación de un predicado GOTO-computable es, también, un predicado GOTO-computable.**

Respuesta: **Cierto**.

Justificación de la respuesta: Un predicado $\theta(\vec{x})$ de aridad $k \geq 1$ es una función total booleana, de aridad k sobre $\mathbb{N}$.

La negación $\neg\theta(\vec{x})$ de dicho predicado puede ser descrita así: $1 - \theta(\vec{x})$.

Es decir, $\neg\theta(\vec{x})$ es la diferencia entre la función constante igual a 1 (de aridad k) y el propio predicado.

Por tanto, si $\theta(\vec{x})$ es GOTO-computable entonces $1 - \theta(\vec{x})$ también lo será, por tratarse de una composición de funciones GOTO-computables.

Conclusión: El predicado $\neg\theta(\vec{x})$ también sera GOTO-computable.

3. La negación de un predicado parcialmente decidible es, también, un predicado parcialmente decidible.

Respuesta: **Falso**.

Justificación de la respuesta: Recordemos que el conjunto $\mathcal{K}$ del problema de la parada es **r.e.**, pero, en cambio, **no** es `GOTO`-computable.

- El conjunto $\mathcal{K}$ del problema de la parada es **r.e.**Por tanto, su función característica, $C_{\mathcal{K}}$, será un **predicado parcialmente decidible**.
- El conjunto complementario, $\overline{\mathcal{K}}$, **no** es **r.e.** Por tanto, su función característica, $C_{\overline{\mathcal{K}}}$ **no** será un predicado parcialmente decidible.

Conclusión: $C_{\mathcal{K}}$ es un predicado parcialmente decidible cuya negación **no** lo es.

4. La <u>cuantificación existencial</u> de un predicado GOTO-computable es, también, un predicado GOTO-computable.

Respuesta: Falso.

Justificación de la respuesta: El predicado HALT $(x, y) \equiv \varphi_x^{(1)}(y) \downarrow$ es **indecidible**; es decir, **no** es GOTO-computable.

Ahora bien, dicho predicado puede ser descrito como sigue:

$$\mathtt{HALT}\ (x, y) \equiv \varphi_x^{(1)}(y) \downarrow \equiv \exists t\, (\mathtt{STEP}^{(1)}(y, x, t))$$

Por tanto, el predicado $\mathtt{STEP}^{(1)}(\mathtt{y}, \mathtt{x}, \mathtt{t})$ es GOTO-computable, mientras su cuantificación existencial $\exists \mathtt{t}\, (\mathtt{STEP}^{(1)}(\mathtt{y}, \mathtt{x}, \mathtt{t}))$ **no** es GOTO-computable.

5. La <u>cuantificación existencial</u> de un predicado parcialmente decidible es, también, un predicado parcialmente decidible.

Respuesta: Cierto.

Justificación de la respuesta: Este resultado ha sido descrito en la parte de teoría de la asignatura (ver transparencia 21 del tema 4).

6. La <u>cuantificación universal</u> de un predicado GOTO-computable es, también, un predicado GOTO-computable.

Respuesta: **Falso**.

Justificación de la respuesta: El predicado HALT $(x, y) \equiv \varphi_x^{(1)}(y) \downarrow$ es **indecidible**; es decir, **no** es GOTO-computable.

Por tanto, su negación $\neg$ HALT (x, y) tampoco será GOTO-computable.

Ahora bien, se tiene: $\neg$ HALT $(x, y) \equiv \varphi_x^{(1)}(y) \uparrow \equiv \forall t\,(\neg \mathrm{STEP}^{(1)}(y, x, t))$

En consecuencia, $\neg\mathrm{STEP}^{(1)}(\mathrm{y}, \mathrm{x}, \mathrm{t})$ es un un predicado GOTO-computable tal que su cuantificación universal $\forall \mathrm{t}\,(\neg\mathrm{STEP}^{(1)}(\mathrm{y}, \mathrm{x}, \mathrm{t}))$ **<u>no</u>** es GOTO-computable.

7. La <u>cuantificación universal</u> de un predicado parcialmente decidible es, también, un predicado parcialmente decidible.

Respuesta: **Falso**.

Justificación de la espuesta: El predicado HALT $(x, y) \equiv \varphi_x^{(1)}(y) \downarrow$ **no** es GOTO-computable pero, en cambio **<u>sí</u>** es **p.d.**

Por tanto, su negación $\neg$ HALT (x, y) tampoco será **p.d.**

Ahora bien, se tiene: $\neg$ HALT $(x, y) \equiv \varphi_x^{(1)}(y) \uparrow \equiv \forall t\,(\neg \mathrm{STEP}^{(1)}(y, x, t))$

En consecuencia, $\neg\mathrm{STEP}^{(1)}(\mathrm{y}, \mathrm{x}, \mathrm{t})$ es un un predicado **p.d.** tal que su cuantificación universal $\forall \mathrm{t}\,(\neg\mathrm{STEP}^{(1)}(\mathrm{y}, \mathrm{x}, \mathrm{t}))$ **<u>no</u>** es **p.d.**

8. El dominio y la gráfica de una función GOTO-computable (parcial o total) son conjuntos recursivamente enumerables.

Respuesta: **Cierto**.

Justificación de la respuesta: De acuerdo con la proposición 1 del tema 4 (transparencia 12 de dicho tema), un conjunto es **r.e.** **sii** es el **dominio** de una función GOTO-computable.

Por su parte, se ha probado que:

- ⋆ La **gráfica** de una función **total** GOTO-computable es un conjunto **GOTO-computable** (ver transparencia 32 del tema 2).
- ⋆ La **gráfica** de una función **parcial** GOTO-computable es un conjunto **r.e.** (ver transparencia 18 del tema 4).

En consecuencia, la gráfica de una función (**parcial** o **total**) GOTO-computable es un conjunto **r.e.**

9. El conjunto $A = \{x : \textbf{ el programa de código } x \textbf{ para sobre } 2x\}$ es recursivamente enumerable.

Respuesta: Cierto.

Justificación de la espuesta: Veamos que la relación de pertenencia al conjunto A es un predicado **p.d.**.

En efecto: para cada número natural x se verifica

$$x \in A \Longleftrightarrow \exists t \ (\mathtt{STEP}^{(1)}(2x, x, t))$$

Luego, la relación de pertenencia $x \in A$ al conjunto A es la cuantificación existencial de un predicado GOTO-computable ($\mathtt{STEP}^{(1)}(2x, x, t)$).

En particular, la relación $x \in A$ es la cuantificación existencial de un predicado **p.d.**; es decir, $x \in A$ es un predicado **p.d.**.

En consecuencia, el conjunto A será **r.e.**

10. El teorema de Rice permite demostrar que ciertos subconjuntos de $\mathbb{N}$ no son `GOTO`–computables.

Respuesta: Cierto.

Justificación de la espuesta: El teorema de Rice asegura que los conjuntos de índices correspondientes a ciertas clases de funciones GOTO-computables, **no** son GOTO-computables (ver transparencia 29 del tema 4).

Específicamente, afirma que si $\mathcal{F}$ es un conjunto de funciones GOTO-computables 1-aria tal que:

(a) existe alguna función que pertenece a $\mathcal{F}$; y

(b) existe alguna función GOTO-computable 1-aria que **no** pertenece a $\mathcal{F}$.

entonces el conjunto $A = \{n \in \mathbb{N} \mid \varphi_n^{(1)} \in \mathcal{F}\}$ **no** es GOTO-computable.

11. El conjunto de índices de una función GOTO–computable de aridad 1, es un conjunto GOTO–computable.

Respuesta: **Falso**.

Justificación de la espuesta:

Sea g cualquier función GOTO–computable de aridad 1.

Sea A_g el conjunto de índices de dicha función: $A_g = \{x \in \mathbb{N} \mid \varphi_x^{(1)} = g\}$.

Consideremos la clase de funciones $\mathcal{F} = \{g\}$.

Entonces, el conjunto de índices de las funciones de $\mathcal{F}$ es, precisamente, A_g.

Se verifica:

- Existe alguna función que pertenece a $\mathcal{F}$: obviamente, $g \in \mathcal{F}$.
- Existe alguna función GOTO-computable 1-aria que **no** pertenece a $\mathcal{F}$ (en efecto: el conjunto de funciones GOTO-computables 1-aria es **infinito**).

Del teorema de Rice se deduce que el conjunto $\{x \in \mathbb{N} \mid \varphi_x^{(1)} \in \mathcal{F}\}$ **no es GOTO-computable**.

Ahora bien, $\{x \in \mathbb{N} \mid \varphi_x^{(1)} \in \mathcal{F}\} = \{x \in \mathbb{N} \mid \varphi_x^{(1)} = g\} = A_g$

En consecuencia, el conjunto de índices de **la** función GOTO-computable g **no es GOTO-computable**.

Nota interesante: En realidad se ha probado que el conjunto de índices de **cualquier función GOTO-computable** **nunca es GOTO-computable**.

12. El conjunto de índices de la función identidad en $\mathbb{N}$ es un conjunto GOTO–computable.

Respuesta: Falso.

Justificación de la espuesta:

En el apartado anterior, se ha probado que el conjunto de índices de **cualquier** función GOTO-computable **no** es GOTO-computable: esto justifica la respuesta.

No obstante, vamos a realizar una prueba explícita aplicando el teorema de Rice.

Sea A el conjunto de índices de **la** función $Id_{\mathbb{N}}$: $A = \{x \in \mathbb{N} \mid \varphi_x^{(1)} = Id_{\mathbb{N}}\}$.

Consideremos la clase de funciones $\mathcal{F}$ cuyo único elemento es $Id_{\mathbb{N}}$: $\mathcal{F} = \{Id_{\mathbb{N}}\}$.

Se verifica:

- $\mathcal{F} \neq \emptyset$ ya que $Id_{\mathbb{N}} \in \mathcal{F}$.
- $\mathcal{F} \neq \mathtt{GCOMP}^{(1)}$ ya que, por ejemplo, la función idénticamente nula de aridad 1 pertenece a $\mathtt{GCOMP}^{(1)}$ y, en cambio, **no** pertenece a $\mathcal{F}$.

Del teorema de Rice se deduce que el conjunto $\{x \in \mathbb{N} \mid \varphi_x^{(1)} \in \mathcal{F}\}$ **no es GOTO-computable**.

Es decir, el conjunto $A = \{x \in \mathbb{N} \mid \varphi_x^{(1)} = Id_{\mathbb{N}}\}$ **no es GOTO-computable**.

Ejercicio 8: Probar que **no son GOTO–computables** los conjuntos:

1. $A = \{x \in \mathbb{N} \mid \mathrm{dom}(\varphi_x^{(1)})$ es un conjunto finito$\}$.
2. $B = \{x \in \mathbb{N} \mid \varphi_x^{(1)}$ es un predicado$\}$.
3. $C = \{x \in \mathbb{N} \mid \varphi_x^{(1)}$ es una aplicación sobreyectiva$\}$.
4. $D = \{x \in \mathbb{N} \mid \varphi_x^{(1)}$ es total y $\forall y \ (\varphi_x^{(1)}(y) \leq \varphi_x^{(1)}(y+1))\}$.
5. $E = \{x \in \mathbb{N} \mid \forall y \in \mathbb{N} \ (\varphi_x^{(1)}(y)$ es una potencia de 2$)\}$
6. $F = \{x \in \mathbb{N} \mid$ el programa GOTO de código x calcula la función identidad en $\mathbb{N}$ o bien la aplicación vacía $\}$.
7. $G = \{x \in \mathbb{N} \mid \mathrm{dom}(\varphi_x^{(1)})$ no es GOTO-computable$\}$.
8. $H = \{x \in \mathbb{N} \mid \mathrm{rang}(\varphi_x^{(1)})$ no es GOTO-computable$\}$.

Indicación: Aplíquese el teorema de Rice para probar que un conjunto **no es GOTO–computable**.

1. **El conjunto $A = \{x \in \mathbb{N} \mid \mathrm{dom}(\varphi_x^{(1)})$ es un conjunto finito$\}$ <u>no</u> es `GOTO`-computable .**

Solución:

Consideremos el siguiente conjunto de funciones `GOTO`-computables 1-aria:

$$\mathcal{F} = \{\ f \in \mathtt{GCOMP}^{(1)} \mid \mathrm{dom}(f) \text{ es un conjunto finito}\}$$

Se verifica lo siguiente:

- $I_{\mathcal{F}} = \{x \in \mathbb{N} \mid \varphi_x^{(1)} \in \mathcal{F}\} = \{x \in \mathbb{N} \mid \mathrm{dom}(\varphi_x^{(1)}) \text{ es finito}\} = A$.
- $\mathcal{F} \neq \emptyset$. En efecto: la función vacía pertenece a $\mathcal{F}$ pues su dominio es finito (concretamente, es el conjunto vacío).
- $\mathcal{F} \neq \mathtt{GCOMP}^{(1)}$. En efecto: la función identidad en $\mathbb{N}$ es 1-aria y `GOTO`-computable pero, **no** pertenece a $\mathcal{F}$ pues su dominio ($= \mathbb{N}$) es infinito.

Luego, del teorema de Rice, se deduce que el conjunto de índices $I_{\mathcal{F}}$ **no** es `GOTO`-computable; es decir, el conjunto A **no** es `GOTO`-computable.

2. **El conjunto $B = \{x \in \mathbb{N} \mid \varphi_x^{(1)}$ es un predicado$\}$ no es GOTO-computable.**

Solución:

Consideremos el siguiente conjunto de funciones GOTO-computables 1-aria:

$$\mathcal{F} = \{ f \in \mathtt{GCOMP}^{(1)} \mid f \text{ es un predicado}\}$$

Se verifica lo siguiente:

- $I_{\mathcal{F}} = \{x \in \mathbb{N} \mid \varphi_x^{(1)} \in \mathcal{F}\} = \{x \in \mathbb{N} \mid \varphi_x^{(1)} \text{ es un predicado}\} = B$.
- $\mathcal{F} \neq \emptyset$. En efecto: la función idénticamente nula de aridad 1, $\varphi_0^{(1)}$, es un predicado GOTO-computable. Por tanto, $\varphi_0^{(1)}$ pertenece a $\mathcal{F}$.
- $\mathcal{F} \neq \mathtt{GCOMP}^{(1)}$. En efecto: la función identidad en $\mathbb{N}$ es 1-aria y GOTO-computable pero, **no** pertenece a $\mathcal{F}$ pues **no es un predicado**.

Luego, del teorema de Rice, se deduce que el conjunto de índices $I_{\mathcal{F}}$ **no** es GOTO-computable; es decir, el conjunto B **no** es GOTO-computable.

3. El conjunto $C = \{x \in \mathbb{N} \mid \varphi_x^{(1)}$ es una aplicación sobreyectiva$\}$ no es `GOTO`-computable.

Solución:

Consideremos el siguiente conjunto de funciones `GOTO`-computables 1-aria:

$$\mathcal{F} = \{\ f \in \mathtt{GCOMP}^{(1)} \mid f \text{ es una aplicación sobreyectiva}\}$$

Se verifica lo siguiente:

- $I_{\mathcal{F}} = \{x \in \mathbb{N} \mid \varphi_x^{(1)} \in \mathcal{F}\} = \{x \in \mathbb{N} \mid \varphi_x^{(1)} \text{ es sobreyectiva}\} = C$.
- $\mathcal{F} \neq \emptyset$. En efecto: la función identidad en $\mathbb{N}$ es `GOTO`-computable y biyectiva. Por tanto, $Id_{\mathbb{N}}$ pertenece a $\mathcal{F}$.
- $\mathcal{F} \neq \mathtt{GCOMP}^{(1)}$. En efecto: la función idénticamente nula de aridad 1 es `GOTO`-computable pero, **no** pertenece a $\mathcal{F}$ y que **no es sobreyectiva**.

Luego, del teorema de Rice, se deduce que el conjunto de índices $I_{\mathcal{F}}$ **no** es `GOTO`-computable; es decir, el conjunto C **no** es `GOTO`-computable.

4. El conjunto $D = \{x \in \mathbb{N} \mid \varphi_x^{(1)}$ es total y $\forall y\ (\varphi_x^{(1)}(y) \leq \varphi_x^{(1)}(y+1))\}$ no es `GOTO`-computable.

Solución:

Consideremos el siguiente conjunto de funciones `GOTO`-computables 1-aria:

$$\mathcal{F} = \{\ f \in \mathtt{GCOMP}^{(1)} \mid f \text{ es total} \wedge \forall y\ (f(y) \leq f(y+1))\}$$

Se verifica lo siguiente:

- $I_{\mathcal{F}} = \{x \in \mathbb{N} \mid \varphi_x^{(1)} \in \mathcal{F}\} = \{x \in \mathbb{N} \mid \varphi_x^{(1)} \text{ es total} \wedge \forall y\ (\varphi_x^{(1)}(y) \leq \varphi_x^{(1)}(y+1))\} = D$.
- $\mathcal{F} \neq \emptyset$. En efecto: la función identidad en $\mathbb{N}$, $Id_{\mathbb{N}}$, es total, `GOTO`-computable y verifica $\forall y\ (Id_{\mathbb{N}}(y) \leq Id_{\mathbb{N}}(y+1))$. Por tanto, $Id_{\mathbb{N}}$ pertenece a $\mathcal{F}$.
- $\mathcal{F} \neq \mathtt{GCOMP}^{(1)}$. En efecto: la función vacía es `GOTO`-computable y puede considerarse de aridad 1 pero, **no** pertenece a $\mathcal{F}$ ya que **no es total**.

Luego, del teorema de Rice, se deduce que el conjunto de índices $I_{\mathcal{F}}$ **no** es `GOTO`-computable; es decir, el conjunto D **no** es `GOTO`-computable.

5. El conjunto $E = \{x \in \mathbb{N} \mid \forall y \in \mathbb{N}\ (\varphi_x^{(1)}(y)$ es una potencia de $2)\}$ no es GOTO-computable.

Solución:

Consideremos el siguiente conjunto de funciones GOTO-computables 1-aria:

$$\mathcal{F} = \{\ f \in \mathtt{GCOMP}^{(1)} \mid \forall y\ \exists n\ (f(y) = 2^n)\}$$

Se verifica lo siguiente:

- $I_{\mathcal{F}} = \{x \in \mathbb{N} \mid \varphi_x^{(1)} \in \mathcal{F}\} = \{x \in \mathbb{N} \mid \forall y\ \exists n\ (\varphi_x^{(1)}(y) = 2^n)\} = E$.
- $\mathcal{F} \neq \emptyset$. En efecto: la función $C_2^{(1)}$ constante igual a 2, de aridad 1, es GOTO-computable y verifica $\forall y\ (C_2^{(1)}(y) = 2)$. Por tanto, $C_2^{(1)}$ pertenece a $\mathcal{F}$.
- $\mathcal{F} \neq \mathtt{GCOMP}^{(1)}$. En efecto: la función idénticamente nula de aridad 1 es GOTO-computable pero, **no** pertenece a $\mathcal{F}$ ya que $\forall n\ (2^n \neq 0)$.

Luego, del teorema de Rice, se deduce que el conjunto de índices $I_{\mathcal{F}}$ **no** es GOTO-computable; es decir, el conjunto E **no** es GOTO-computable.

6. El conjunto
$F = \{x \in \mathbb{N} \mid$ **el programa de código** x **calcula la función identidad en** $\mathbb{N}$ **o bien la aplicación vacía**$\}$
<u>no</u> es `GOTO`-computable.

Solución:

Consideremos el siguiente conjunto de funciones `GOTO`-computables 1-aria:

$$\mathcal{F} = \{\ f \in \mathtt{GCOMP}^{(1)} \mid (f = Id_{\mathbb{N}}) \vee (f \text{ es la aplicación vacía})\}$$

Se verifica lo siguiente:

- $I_{\mathcal{F}} = \{x \in \mathbb{N} \mid \varphi_x^{(1)} \in \mathcal{F}\} = \{x \in \mathbb{N} \mid (\varphi_x^{(1)} = Id_{\mathbb{N}}) \vee (\varphi_x^{(1)} \text{ es la aplicación vacía})\} = F$.
- $\mathcal{F} \neq \emptyset$. En efecto: la función identidad en $\mathbb{N}$, $Id_{\mathbb{N}}$, pertenece a $\mathcal{F}$.
- $\mathcal{F} \neq \mathtt{GCOMP}^{(1)}$. En efecto: la función idénticamente nula de aridad 1 es `GOTO`-computable pero, **no** pertenece a $\mathcal{F}$.

Luego, del teorema de Rice, se deduce que el conjunto de índices $I_{\mathcal{F}}$ **no** es `GOTO`-computable; es decir, el conjunto F **no** es `GOTO`-computable.

7. El conjunto $G = \{x \in \mathbb{N} \mid$ dom$(\varphi_x^{(1)})$ no es GOTO-computable$\}$ no es GOTO-computable.

Solución:

Consideremos el siguiente conjunto de funciones GOTO-computables 1-aria:

$$\mathcal{F} = \{\ f \in \mathtt{GCOMP}^{(1)} \mid \mathrm{dom}(f) \text{ no es GOTO-computable}\}$$

Se verifica lo siguiente:

- $I_{\mathcal{F}} = \{x \in \mathbb{N} \mid \varphi_x^{(1)} \in \mathcal{F}\} = \{x \in \mathbb{N} \mid (\mathrm{dom}(\varphi_x^{(1)})$ no es GOTO-computable$\} = G$.
- $\mathcal{F} \neq \emptyset$. En efecto: la función f definida por $f(x) = x$ **sii** $x \in \mathcal{K}$ (véase el apartado (b) del Ejercicio 3) es GOTO-computable, de aridad 1, y su dominio (el conjunto $\mathcal{K}$) no es GOTO-computable. Por tanto, f pertenece a $\mathcal{F}$.
- $\mathcal{F} \neq \mathtt{GCOMP}^{(1)}$. En efecto: la función idénticamente nula de aridad 1 es GOTO-computable pero, **no** pertenece a $\mathcal{F}$ ya que su dominio (el conjunto $\mathbb{N}$) es GOTO- computable.

Luego, del teorema de Rice, se deduce que el conjunto de índices $I_{\mathcal{F}}$ **no** es GOTO-computable; es decir, el conjunto G **no** es GOTO-computable.

8. El conjunto $H = \{x \in \mathbb{N} \mid \text{rang}(\varphi_x^{(1)})$ <u>no</u> es GOTO-computable$\}$ <u>no</u> es GOTO-computable.

Solución:

Consideremos el siguiente conjunto de funciones GOTO-computables 1-aria:

$$\mathcal{F} = \{\ f \in \texttt{GCOMP}^{(1)} \mid \text{rang}(f) \ \underline{\textbf{no}}\ \textbf{es GOTO-computable}\}$$

Se verifica lo siguiente:

- $I_{\mathcal{F}} = \{x \in \mathbb{N} \mid \varphi_x^{(1)} \in \mathcal{F}\} = \{x \in \mathbb{N} \mid \text{rang}(\varphi_x^{(1)})$ <u>no</u> es GOTO-computable$\} = H$.
- $\mathcal{F} \neq \emptyset$. En efecto: la función f definida por $f(x) = x$ **sii** $x \in \mathcal{K}$ (véase el apartado (b) del Ejercicio 3) es GOTO-computable, de aridad 1, y su rango (el conjunto $\mathcal{K}$) <u>no</u> es GOTO-computable. Por tanto, f pertenece a $\mathcal{F}$.
- $\mathcal{F} \neq \texttt{GCOMP}^{(1)}$. En efecto: la función idénticamente nula de aridad 1 es GOTO-computable pero, **no** pertenece a $\mathcal{F}$ ya que su rango (el conjunto $\{0\}$) es GOTO-computable.

Luego, del teorema de Rice, se deduce que el conjunto de índices $I_{\mathcal{F}}$ **no** es GOTO-computable; es decir, el conjunto H **no** es GOTO-computable.

Ejercicio 9: Se considera el siguiente problema de decisión:

Dado un número natural $n \in \mathbb{N}$, determinar si $n \in rang(\varphi_n^{(1)})$

Probar que dicho problema es **indecidible**.

Solución: Probar que dicho problema es **indecidible** equivale a demostrar que el conjunto asociado al mismo, $\mathrm{A} = \{\mathrm{n} \in \mathbb{N} \mid \ \mathrm{n} \in \mathrm{rang}(\varphi_{\mathrm{n}}^{(1)})\}$, **no** es GOTO–computable.

Probémoslo usando el **método por reducción al absurdo**.

Si el conjunto A fuese GOTO-computable, la función parcial $\mathrm{f} : \mathbb{N} - \to \mathbb{N}$ definida por

$$\mathrm{f}(x) = \begin{cases} x \text{ si } x \notin A \\ \uparrow \text{ si } x \in A \end{cases}$$

sería GOTO–computable, pues su gráfica, $G(f)$ es un conjunto **r.e.**

$$(x, y) \in G(f) \Longleftrightarrow (y = x) \wedge (x \notin A)$$

Por tanto, existiría $n \in \mathbb{N}$ tal que $f = \varphi_n^{(1)}$. Se pueden dar dos casos:

- $\boxed{\text{Caso 1: } n \in A}$
  - Por una parte, $n \in A \overset{def.A}{\Rightarrow} n \in \mathrm{rang}(\varphi_n^{(1)}) \overset{\varphi_n^{(1)}=f}{\Rightarrow} n \in \mathrm{rang}(f)$.
  - Por otra, $n \in A \overset{def.f}{\Rightarrow} f(n) \uparrow \Rightarrow n \notin \mathrm{rang}(f)$: **contradicción**.
- $\boxed{\text{Caso 2: } n \notin A}$
  - Por una parte, $n \notin A \overset{def.A}{\Rightarrow} n \notin \mathrm{rang}(\varphi_n^{(1)}) \overset{\varphi_n^{(1)}=f}{\Rightarrow} n \notin \mathrm{rang}(f)$.
  - Por otra, $n \notin A \overset{def.f}{\Rightarrow} f(n) = n \Rightarrow n \in \mathrm{rang}(f)$: **contradicción**.

Ejercicio 10: Se considera el siguiente problema de decisión:

> **Dado un número natural $n \in \mathbb{N}$, determinar si el dominio de la función $\varphi_n^{(1)}$ es un conjunto GOTO-computable**

Probar que dicho problema es **indecidible**.

Solución: Probar que dicho problema es **indecidible** equivale a demostrar que el conjunto $A = \{n \in \mathbb{N} \mid \mathrm{dom}(\varphi_n^{(1)}) \text{ es GOTO-computable}\}$ **no** es GOTO–computable.

Vamos a probarlo usando el **teorema de Rice**.

Consideremos el siguiente conjunto de funciones GOTO-computables 1-aria:

$$\mathcal{F} = \{f \in \mathtt{GCOMP}^{(1)} \mid \mathrm{dom}(f) \text{ es un conjunto GOTO-computable}\}$$

Se verifica lo siguiente:

(1) $I_{\mathcal{F}} = \{n \in \mathbb{N} \mid \varphi_n^{(1)} \in \mathcal{F}\} = \{\ n \in \mathbb{N} \mid \mathrm{dom}(\varphi_n^{(1)}) \text{ es GOTO-computable}\} = A$

(2) $\mathcal{F} \neq \emptyset$. En efecto: la función identidad, $I_{\mathbb{N}}$, en $\mathbb{N}$, es GOTO-computable de aridad 1 y $\mathrm{dom}(I_{\mathbb{N}}) = \mathbb{N}$ es un conjunto GOTO-computable. Por tanto, $I_{\mathbb{N}} \in \mathcal{F}$.

(3) $\mathcal{F} \neq \mathtt{GCOMP}^{(1)}$. En efecto: la función 1-aria $f : \mathbb{N} - \to \mathbb{N}$ definida por $f(x) = 1$ **sii** $x \in \mathcal{K}$ es GOTO-computable (ver ejercicio 1) y $\mathrm{dom}(f) = \mathcal{K}$. Por tanto, $f \notin \mathcal{F}$ ya que $\mathrm{dom}(f) = \mathcal{K}$ **no** es un conjunto GOTO-computable.

De (2) y (3), aplicando el teorema de Rice, se deduce que el conjunto $I_{\mathcal{F}}$ **no** es GOTO-computable. En consecuencia, de (1) se concluye que A **no** es GOTO-computable.

Ejercicio 11: Se considera el siguiente problema de decisión:

Dado un número natural $n \in \mathbb{N}$, determinar si la gráfica de la función $\varphi_n^{(1)}$ no es un conjunto $\mathtt{GOTO}$-computable

Probar que dicho problema es **indecidible**.

Solución: Probar que dicho problema es **indecidible** equivale a demostrar que el conjunto $A = \{n \in \mathbb{N} \mid \text{la gráfica}\, G(\varphi_n^{(1)})\ \underline{\text{no}} \text{ es } \mathtt{GOTO}\text{-computable}\}$ **no** es GOTO–computable.

Vamos a probarlo usando el **teorema de Rice**.

Consideremos el siguiente conjunto de funciones GOTO-computables 1-aria:

$$\mathcal{F} = \{f \in \mathtt{GCOMP}^{(1)} \mid G(f)\ \underline{\text{no}} \text{ es un conjunto } \mathtt{GOTO}\text{-computable}\}$$

Se verifica lo siguiente:

(1) $I_{\mathcal{F}} = \{n \in \mathbb{N} \mid \varphi_n^{(1)} \in \mathcal{F}\} = \{\, n \in \mathbb{N} \mid G(\varphi_n^{(1)})\ \underline{\text{no}} \text{ es } \mathtt{GOTO}\text{-computable}\} = A$

(2) $\mathcal{F} \neq \emptyset$. En efecto: la función f definida por $f(x) = x \Longleftrightarrow x \in \mathcal{K}$ es 1-aria, GOTO-computable. Además, su gráfica $G(f) = \{(x, x) \mid x \in \mathcal{K}\}$ **no** es GOTO-computable (pues si lo fuese, entonces también lo sería $\mathcal{K}$). Por tanto, $f \in \mathcal{F}$.

(3) $\mathcal{F} \neq \mathtt{GCOMP}^{(1)}$. En efecto: la función identidad, $I_{\mathbb{N}}$, en $\mathbb{N}$, es GOTO-computable de aridad 1. Además, su gráfica $G(I_{\mathbb{N}}) = \{(x, x) \mid x \in \mathbb{N}\}$ es un conjunto GOTO-computable (por serlo $\mathbb{N}$). Luego, $I_{\mathbb{N}} \notin \mathcal{F}$.

De (2) y (3), aplicando el teorema de Rice, se deduce que el conjunto $I_{\mathcal{F}}$ **no** es GOTO-computable. En consecuencia, de (1) se concluye que A **no** es GOTO-computable.

Ejercicio 12: Se considera el siguiente problema de decisión:

> **Dado un número natural $n \in \mathbb{N}$, determinar si existe un programa `GOTO` sin instrucciones condicionales que calcula la función $\varphi_n^{(1)}$**

Probar que dicho problema es **indecidible**.

Solución: Probar que dicho problema es **indecidible** equivale a demostrar que el conjunto $\mathrm{A} = \{\mathrm{n} \in \mathbb{N} \mid$ Existe un programa `GOTO` sin condicionales que calcula $\varphi_{\mathrm{n}}^{(1)}\}$ **no** es `GOTO`–computable.

Vamos a probarlo usando el **teorema de Rice**.

Consideremos el siguiente conjunto de funciones `GOTO`-computables 1-aria:

$$\mathcal{F} = \{f \in \mathtt{GCOMP}^{(1)} \mid \text{ Existe un programa GOTO sin condicionales que calcula } f\}$$

Se verifica lo siguiente:

(1) $I_{\mathcal{F}} = \{n \in \mathbb{N} \mid \varphi_n^{(1)} \in \mathcal{F}\} = \{\ n \in \mathbb{N} \mid$ Existe un programa `GOTO` sin condicionales que calcula $\varphi_n^{(1)}\} = A$.

(2) $\mathcal{F} \neq \emptyset$. En efecto: la función $C_0^{(1)}$ idénticamente nula de aridad 1 en $\mathbb{N}$, es calculada por el programa vacío. Por tanto, $C_0^{(1)} \in \mathcal{F}$.

(3) $\mathcal{F} \neq \mathtt{GCOMP}^{(1)}$. En efecto: la función identidad, $I_{\mathbb{N}}$ pertenece a $\mathtt{GCOMP}^{(1)}$. Además, según se ha visto en clase, $I_{\mathbb{N}}$ **no puede ser calculada** por un programa `GOTO` que carezca de intrucciones condicionales. Por tanto, $I_{\mathbb{N}} \notin \mathcal{F}$.

De (2) y (3), aplicando el teorema de Rice, se deduce que el conjunto $I_{\mathcal{F}}$ **no** es `GOTO`-computable. En consecuencia, de (1) se concluye que A **no** es `GOTO`-computable.

Ejercicio 13: Se considera el siguiente problema de decisión:

> **Dado un número natural n, determinar si la computación del programa `GOTO` de código n, con dato de entrada n, no es de parada**

Probar que dicho problema **no es semidecidible** y, por tanto, **es indecidible**.

Solución:

Probar que dicho problema **no es es semidecidible** equivale a demostrar que el conjunto $A = \{n \in \mathbb{N} \mid \varphi_n^{(1)}(n) \uparrow\}$, asociado a dicho problema, **no es r.e.**

Para ello, comencemos observando que el complementario, $\overline{A}$, de dicho conjunto es, precisamente, el conjunto $\mathcal{K}$ del problema de la parada.

Luego, $\overline{A}$ será un conjunto **r.e.** pero **no** `GOTO`-computable.

Puesto que $\overline{A}$ es **r.e.** pero **no** `GOTO`-computable, del teorema del complemento se deduce que el conjunto $\overline{\overline{A}}$ **no será r.e.**

En consecuencia, el conjunto A **no será r.e.** (pues $\overline{\overline{A}} = A$).

Ejercicio 14: Consideremos los siguientes problemas de decisión:

X $\equiv$ **Dado un número natural n, determinar si la función $\varphi_n^{(1)}$ no es inyectiva**

Y $\equiv$ **Dado un número natural n, determinar si la función $\varphi_n^{(1)}$ sí es inyectiva**

Demostrar que el problema X es **indecidible** pero que, en cambio, **sí es semidecidible**.

Demostrar que el problema Y es **indecidible** y que, además, **no es semidecidible**.

Solución:

$$\boxed{\textbf{El problema X es indecidible}}$$

Consideremos el conjunto $A_X = \{n \in \mathbb{N} \mid \varphi_n^{(1)} \textbf{ no es inyectiva}\}$ asociado a X.

Veamos que el conjunto A_X **no** es GOTO-**computable**, utilizando el teorema de Rice.

Consideremos el siguiente conjunto $\mathcal{F}$ de funciones 1-aria y GOTO-computable:

$$\mathcal{F} = \{f \in \mathtt{GCOMP}^{(1)} \mid f \textbf{ no es inyectiva}\}$$

Se verifica lo siguiente:

(1) $I_{\mathcal{F}} = \{n \in \mathbb{N} \mid \varphi_n^{(1)} \in \mathcal{F}\} = \{n \in \mathbb{N} \mid \varphi_n^{(1)} \text{ no es inyectiva } \} = A_X$.

(2) $\mathcal{F} \neq \emptyset$. En efecto: la función $C_0^{(1)}$ idénticamente nula de aridad 1 en $\mathbb{N}$ es GOTO-computable y **no es inyectiva**. Por tanto, $C_0^{(1)} \in \mathcal{F}$.

(3) $\mathcal{F} \neq \mathtt{GCOMP}^{(1)}$. En efecto: la función identidad, $I_{\mathbb{N}}$, en $\mathbb{N}$, es GOTO-computable, de aridad 1 e inyectiva. Por tanto, $I_{\mathbb{N}} \notin \mathcal{F}$.

De (2) y (3), aplicando el teorema de Rice, se deduce que $I_{\mathcal{F}}$ **no** es GOTO-computable. En consecuencia, de (1) se concluye que el conjunto A_X **no** es GOTO-computable.

El problema X es semidecidible

Bastará probar que el conjunto A_X es **r.e.**

Para ello, vamos a caracterizar la correspondiente relación de pertenencia a A_X.

Se verifica lo siguiente

$$n \in A_X \iff \varphi_n^{(1)} \textbf{ no} \text{ es inyectiva} \iff [\exists z \exists t \exists u \ (z \neq t \wedge \varphi_n^{(1)}(z) = u \ \wedge \ \varphi_n^{(1)}(t) = u)]$$

Es decir,

$$n \in A_X \iff [\exists z \exists t \exists u \ (z \neq t \wedge (z, u) \in G(\varphi_n^{(1)}) \wedge (t, u) \in G(\varphi_n^{(1)}))]$$

Ahora bien, el predicado que aparece a la derecha es **p.d.** ya que es la cuantificación existencial de un predicado que, a su vez, es la conjunción de otros tres que son **p.d.**

- Un predicado "ser distinto": $z \neq u$.
- Dos predicados de pertenencia a un conjunto **r.e.**: $G(\varphi_n^{(1)})$.

El problema Y es indecidible

Consideremos el conjunto $A_Y = \{n \in \mathbb{N} \mid \varphi_n^{(1)}$ **es inyectiva**$\}$ asociado al problema Y.

Veamos que el conjunto A_Y **no** es GOTO-**computable**, utilizando el teorema de Rice.

Consideremos el siguiente conjunto $\mathcal{F}$ de funciones 1-aria y GOTO-computable:

$$\mathcal{F} = \{f \in \mathtt{GCOMP}^{(1)} \mid f \textbf{ es inyectiva}\}$$

Se verifica lo siguiente:

(1) $I_{\mathcal{F}} = \{n \in \mathbb{N} \mid \varphi_n^{(1)} \in \mathcal{F}\} = \{n \in \mathbb{N} \mid \varphi_n^{(1)} \text{ es inyectiva }\} = A_Y$.

(2) $\mathcal{F} \neq \emptyset$. En efecto: la función identidad, $I_{\mathbb{N}}$, en $\mathbb{N}$, es GOTO-computable, de aridad 1 e inyectiva. Por tanto, $I_{\mathbb{N}} \in \mathcal{F}$.

(3) $\mathcal{F} \neq \mathtt{GCOMP}^{(1)}$. En efecto: la función $C_0^{(1)}$ idénticamente nula de aridad 1 en $\mathbb{N}$ es GOTO-computable y, además, <u>no</u> es inyectiva. Por tanto, $C_0^{(1)} \notin \mathcal{F}$.

De (2) y (3), aplicando el teorema de Rice, se deduce que $I_{\mathcal{F}}$ **no** es GOTO-computable. En consecuencia, de (1) se concluye que el conjunto A_Y **no** es GOTO-computable.

El problema Y no es semidecidible

Bastará probar que el conjunto A_Y **no es r.e.**

Probémoslo usando el **método por reducción al absurdo**.

Supongamos que el conjunto A_Y fuese **r.e.**

Entonces:

- $\overline{A_X} = \{n \in \mathbb{N} \mid \varphi_n^{(1)}$ **es inyectiva**$\} = A_Y$ sería **r.e.**
- A_X es **r.e.**

Por tanto, A_X y $\overline{A_X}$ serían conjuntos **r.e.**

Luego, del **teorema del complemento** deduciríamos que A_X sería GOTO-computable.

Esto contradice lo que se ha probado en el primer apartado (el conjunto A_X **no es GOTO-computable**).

Ejercicio 15: Se considera el siguiente problema de decisión:

> **Dado un número natural n, determinar si el rango de la función $\varphi_n^{(1)}$ es un conjunto de números primos**

Probar que dicho problema es **indecidible** y que, además, **no** es **semidecidible**.

Solución:

Consideremos el conjunto $A = \{n \in \mathbb{N} \mid \text{rang}(\varphi_n^{(1)}) \subseteq \texttt{PRIMO})\}$ asociado a dicho problema ($\texttt{PRIMO}$ es el conjunto de los números naturales que son primos).

Veamos que el conjunto A **no es $\texttt{GOTO}$-computable** pero que, en cambio, **sí** es **r.e.**

El conjunto A no es $\texttt{GOTO}$-computable

Vamos a utilizar el teorema de Rice y, para ello, consideremos el siguiente conjunto $\mathcal{F}$:

$$\mathcal{F} = \{f \in \texttt{GCOMP}^{(1)} \mid \text{rang}(f) \subseteq \texttt{PRIMO})\}.$$

Se verifica lo siguiente:

(1) $I_\mathcal{F} = \{n \in \mathbb{N} \mid \varphi_n^{(1)} \in \mathcal{F}\} = \{n \in \mathbb{N} \mid \text{rang}(\varphi_n^{(1)}) \subseteq \texttt{PRIMO})\} = A$.

(2) $\mathcal{F} \neq \emptyset$. En efecto: la función $C_2^{(1)}$ constante igual a 2 de aridad 1 en $\mathbb{N}$, es $\texttt{GOTO}$-computable y tal que $\text{rang}(C_2^{(1)}) = \{2\} \subseteq \texttt{PRIMO}$. Por tanto $C_2^{(1)} \in \mathcal{F}$.

(3) $\mathcal{F} \neq \texttt{GCOMP}^{(1)}$. En efecto: la función $C_0^{(1)}$ idénticamente nula de aridad 1 en $\mathbb{N}$, es $\texttt{GOTO}$-computable y tal que $\text{rang}(C_0^{(1)}) = \{0\} \nsubseteq \texttt{PRIMO}$. Por tanto $C_0^{(1)} \notin \mathcal{F}$.

De (2) y (3), aplicando el teorema de Rice, se deduce que $I_\mathcal{F}$ **no** es $\texttt{GOTO}$-computable. En consecuencia, de (1) se concluy e

El conjunto A **no es r.e.**

Teniendo presente que A **no** es GOTO-computable, para probar que A **no es r.e.**, será suficiente demostrar que $\overline{A}$ es un conjunto **r.e.**(por el **teorema del complemento**).

Veamos, pues, que $\overline{A}$ es un conjunto **r.e.**

Para ello, describimos la relación de pertenencia a $\overline{A}$:

- $n \in \overline{A} \iff \exists x \ (x \in \text{rang}(\varphi_n^{(1)}) \ \wedge \ x$ **no** es un número primo$)$
  $\iff \exists x \ \exists y \ \exists z \ \exists t \ (x = \varphi_n^{(1)}(y) \ \wedge \ z \neq 1 \ \wedge \ t \neq 1 \ \wedge \ x = z \cdot t)$

La relación $n \in \overline{A}$ es un **predicado p.d.** ya que se trata de la cuantificación existencial de la conjunción de cuatro predicados, todos ellos **p.d.**

- El primero ($x = \varphi_n^{(1)}(y)$) es un predicado de igualdad asociado a $\texttt{HALT}(n, y)$ y, por tanto, es **p.d.**
- El segundo y el tercero ($z \neq 1$, $t \neq 1$) son predicados "$\neq$" y, por tanto, son GOTO-computables.
- El cuarto ($x = z \cdot t$) es un predicado de igualdad, en donde interviene la función producto. Por tanto, será un predicado GOTO-computable.

Ejercicio 16: Se considera el siguiente problema de decisión:

> **Dado un número natural n, determinar si para cualquier número natural $x \in \mathbb{N}$ se tiene que $\varphi_n^{(1)}(x)$ es un número par**

Probar que dicho problema es **indecidible** y que, además, **no** es **semidecidible**.

Solución:

Consideremos el conjunto $\mathrm{A} = \{\mathrm{n} \in \mathbb{N} \mid \forall \mathrm{x}\ (\varphi_\mathrm{n}^{(1)}(\mathrm{x}) \text{ es un número par})\}$ asociado a dicho problema.

Veamos que el conjunto A **no es GOTO-computable** pero que, en cambio, **sí** es **r.e.**

El conjunto A no es GOTO-computable

Vamos a utilizar el teorema de Rice y, para ello, consideremos el siguiente conjunto $\mathcal{F}$:

$$\mathcal{F} = \{f \in \mathtt{GCOMP}^{(1)} \mid \forall x\ (f(x) \text{ es un número par})\}.$$

Se verifica lo siguiente:

(1) $I_\mathcal{F} = \{n \in \mathbb{N} \mid \varphi_n^{(1)} \in \mathcal{F}\} = \{n \in \mathbb{N} \mid \forall x\ (\varphi_n^{(1)}(x) \text{ es un número par})\} = \mathrm{A}$.

(2) $\mathcal{F} \neq \emptyset$. En efecto: la función $C_2^{(1)}$ constante igual a 2 de aridad 1, es GOTO-computable y tal que $\forall x\,(C_2^{(1)}(x) = 2)$. Por tanto $C_2^{(1)} \in \mathcal{F}$.

(3) $\mathcal{F} \neq \mathtt{GCOMP}^{(1)}$. En efecto: la función identidad, $I_\mathbb{N}$, en $\mathbb{N}$, es GOTO-computable, de aridad 1 e $I_\mathbb{N}(1) = 1$, que **no** es un número par. Por tanto, $I_\mathbb{N} \notin \mathcal{F}$.

De (2) y (3), aplicando el teorema de Rice, se deduce que $I_\mathcal{F}$ **no** es GOTO-computable. En consecuencia, de (1) se concluye que el conjunto A **no** es GOTO-computable.

El conjunto A **no** es r.e.

Para probar que A **no** es r.e., será suficiente demostrar que $\overline{A}$ es un conjunto **r.e.**

- En efecto: si $\overline{A}$ es un conjunto **r.e.** entonces teniendo presente que A **no** es GOTO-computable, se deduce que A **no** es **r.e.** (**teorema del complemento**)

Veamos, pues, que $\overline{A}$ es un conjunto **r.e.**

Para ello, describimos la relación de pertenencia a $\overline{A}$:

- ⋆ $n \in \overline{A} \iff \exists x \; (\varphi_n^{(1)}(x)$ **no** es un número par$) \iff \exists x \; \exists y \; (\varphi_n^{(1)}(x) = 2 \cdot y + 1)$

La relación $n \in \overline{A}$ es un **predicado p.d.** ya que se trata de la cuantificación existencial de un predicado de igualdad ($\varphi_n^{(1)}(x) = 2 \cdot y + 1$) en el que intervienen:

- ⋆ Las funciones $\varphi_n^{(1)}$, la "suma" en $\mathbb{N}$ y el "producto" en $\mathbb{N}$, que son GOTO-computables.