eman ta zabal zazu

Universidad del País Vasco — Euskal Herriko Unibertsitatea

Lenguajes, Computación y Sistemas Inteligentes

Grado en Ingeniería Informática de Gestión y Sistemas de Información

Escuela de Ingeniería de Bilbao (UPV/EHU)

2^o curso

Curso académico 2023-2024

Tema 5: λ-cálculo y computabilidad

JOSÉ GAINTZARAIN IBARMIA

Departamento de Lenguajes y Sistemas Informáticos

Última actualización: 01 - 10 - 2023

Índice general

Índice de figuras

Índice de tablas

Tema 5

λ-cálculo y computabilidad

5.1.

Introducción

En este tema se utilizan el λ-cálculo y el concepto de λ-término universal para probar que algunos problemas de decisión concretos son semidecidibles (es decir, son semicomputables) pero no son decidibles (es decir, no son computables) y que otros problemas de decisión concretos no son ni siquiera semidecidibles (es decir, no son ni siquiera semicomputables).

En el Tema 3, probamos matemáticamente que existen problemas de decisión que no son decidibles (es decir, no son computables). Esa prueba estaba basada en el hecho de que los problemas de decisión pueden ser formulados como lenguajes formados por aquellos datos que han de recibir la respuesta ”Sí” y el número de programas escribibles es inferior al número de lenguajes (o de funciones características).

De la misma forma, en el Tema 3, probamos matemáticamente que existen problemas de decisión que no son ni siquiera semidecidibles (es decir, no son semicomputables). Esa prueba estaba basada también en el hecho de que los problemas de decisión pueden ser formulados como lenguajes formados por aquellos datos que han de recibir la respuesta ”Sí” y el número de programas escribibles es inferior al número de lenguajes (o de funciones semicaracterísticas).

De todas formas, las herramientas de las que disponíamos en el Tema 3 no nos permitían estudiar casos concretos para los cuales no se pudieran escribir programas porque no teníamos un lenguaje de programación que fuese lo más general posible y que permitiese representar cualquier algoritmo.

El λ-cálculo es un lenguaje de programación que permite expresar cualquier algoritmo sin preocuparnos de los tipos de datos y sin posibilidad de que surja ningún tipo de error durante la ejecución de los programas. Por tanto, ahora disponemos de la herramienta más general posible para estudiar algoritmos o programas concretos.

En este Tema 5, al tratar solo con problemas de decisión, se considera que el resultado de aplicar un λ-término a un dato (otro λ-término) será TRUE, FALSE o no habrá respuesta por producirse un cálculo infinito. Cuando un programa no responde porque está inmerso en un cálculo infinito, diremos que el programa está ciclando.

5.2.

Clasificación de lenguajes

En este tema clasificamos los lenguajes definibles sobre un alfabeto $\mathbb{A}$ en tres categorias:

1. **Lenguajes decidibles**. Un lenguaje L definido sobre un alfabeto $\mathbb{A}$ es decidible si su función característica χ_L es computable mediante un λ-término (o algoritmo).

   La función característica χ_L correspondiente a un lenguaje L es la siguiente:

$$\chi_L : \mathbb{A}^* \to \mathbb{2}$$

$$\chi_L(w) = \begin{cases} 1 & \text{si } w \in L \\ 0 & \text{si } w \notin L \end{cases}$$

2. **Lenguajes reconocibles**. Un lenguaje L definido sobre un alfabeto $\mathbb{A}$ es reconocible si su función semicaracterística Σ_L es computable mediante un λ-término (o algoritmo).

   La función semicaracterística Σ_L correspondiente a un lenguaje L es la siguiente:

$$\Sigma_L : \mathbb{A}^* \to \{1, \bot\}$$

$$\Sigma_L(w) = \begin{cases} 1 & \text{si } w \in L \\ \bot & \text{si } w \notin L \end{cases}$$

3. **Lenguajes no reconocibles**. Un lenguaje L definido sobre un alfabeto $\mathbb{A}$ no es reconocible si su función semicaracterística Σ_L no es computable mediante un λ-término (o algoritmo).

En la figura 5.2.1 de la página 14, se muestra la clasificación de todos los lenguajes de $\mathbb{2}^{\mathbb{A}^*}$. Por una parte están los lenguajes reconocibles y por otra parte los lenguajes no reconocibles. Hay muchos más lenguajes no reconocibles que reconocibles. Dentro de los lenguajes reconocibles, algunos son decidibles.

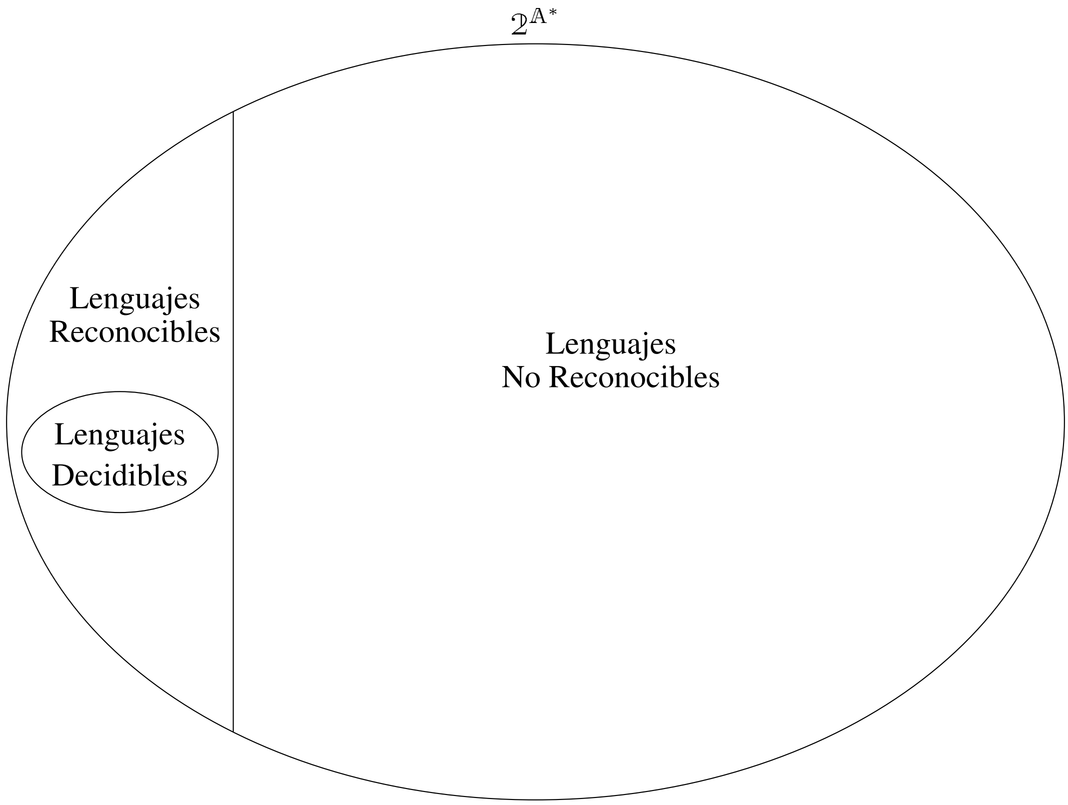

Figura 5.2.1. Clasificación de los lenguajes de $2^{\mathbb{A}^*}$ en decidibles, reconocibles y no reconocibles.

5.3.

Representación gráfica de λ-términos

Recordemos que un λ-término representa un algoritmo para realizar un determinado cálculo.

En general, un λ-término —al igual que cualquier algoritmo— recibe uno o más datos de entrada y efectúa el correspondiente cálculo. Cuando se trabaja con problemas de decisión, el objetivo es terminar con un "Sí" o con un "No" (TRUE O FALSE). Aunque no siempre es posible terminar el proceso de averiguaciones: a veces el cálculo puede ser infinito y nunca se llegará a determinar si la respuesta es "Sí" o "No", es decir, a veces ciclará sin fin.

Para simplificar las demostraciones matemáticas que se van a dar en este tema, vamos a representar los λ-términos mediante esquemas gráficos.

En la figura 5.3.1 de la página 17 se muestra un esquema gráfico, donde se tiene un λ-término E. El dato de entrada es otro λ-término d y hay tres posibles situaciones: (1) respuesta con "Sí" (TRUE); (2) respuesta con "No" (FALSE); (3) sin respuesta (ciclar sin fin). Ese esquema representa la aplicación del λ-término E al λ-término d. Por tanto, ese esquema representa la aplicación (Ed).

El λ-término E será una abstracción de la forma $(\lambda\gamma.S)$ donde el λ-término S representa el cálculo que lleva a cabo la abstracción E. Y el esquema en su totalidad representa la aplicación $((\lambda\gamma.S)d)$.

En la figura 5.3.2 de la página 17 se muestra un esquema gráfico donde se tiene un λ-término universal B. El dato de entrada para B es un par ordenado $\langle Q, d\rangle$ donde Q es un λ-término y d es otro λ-término que servirá de dato de entrada para Q. El λ-término universal B simula primero la ejecución de Q con dato de entrada d y dependiendo de lo que ocurra con esa simulación (Sí, No, Ciclar), B realizará, probablemente, un cálculo adicional para determinar la respuesta final de B (Sí, No, Ciclar).

El λ-término B será una abstracción de la forma $(\lambda\gamma_1.(\lambda\gamma_2.W))$ donde el λ-término W representa el cálculo que lleva a cabo la abstracción B. Y el esquema en su totalidad representa la aplicación $(((\lambda\gamma_1.(\lambda\gamma_2.W))Q)d)$.

Los λ-términos universales son importantes para probar algunos resultados sobre computabilidad e incomputabilidad. En concreto, vamos a utilizar λ-términos universales para probar que algunos lenguajes son reconocibles pero no son decidibles y que algunos lenguajes no son ni siquiera reconocibles.

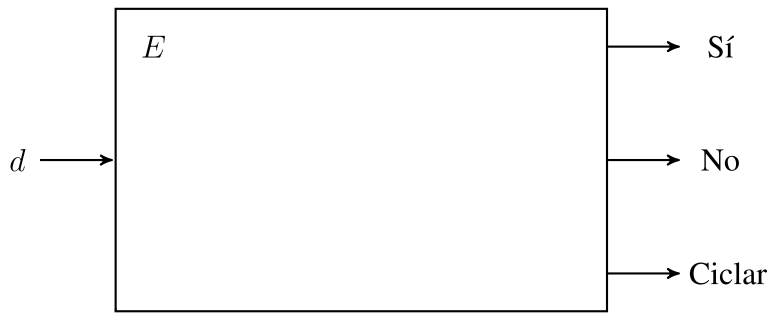

Figura 5.3.1. λ-término E que recibe, como dato de entrada, el λ-término d. A continuación, realiza el cálculo y devuelve el resultado correspondiente (Sí, No, Ciclar). Este gráfico representa la aplicación (Ed) que aplica E a d.

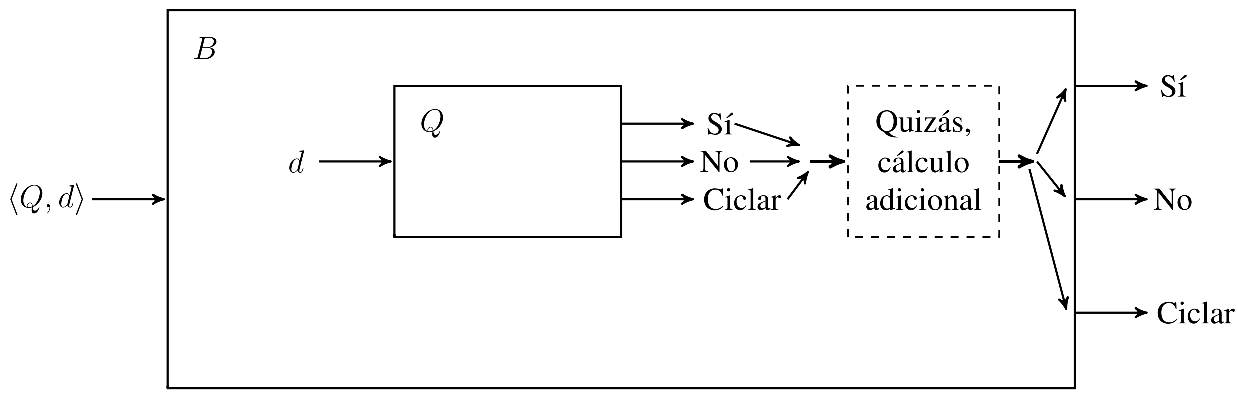

Figura 5.3.2. λ-término universal B que recibe, como dato de entrada, el par ordenado $\langle Q, d\rangle$. A continuación, B simula la ejecución del λ-término Q con el dato d y, en función de lo que haga Q (Sí, No, Ciclar), la máquina B hará lo que corresponda (Sí, No, Ciclar) tras realizar, posiblemente, algún cálculo adicional. Este gráfico representa la aplicación $((BQ)d)$ que aplica B a Q y a d.

5.4.

Lenguajes reconocibles pero no decidibles

Para probar que un lenguaje L definido sobre un alfabeto $\mathbb{A}$ no es decidible hay que probar que no existe ningún algoritmo (ningún λ-término) que sea capaz de decidir para todas las palabras de $\mathbb{A}^*$, si cada una de ellas pertenece o no pertenece al lenguaje L. Dicho de otra forma, hay que probar que no existe ningún algoritmo (ningún λ-término) que sea capaz de calcular la función característica $\boldsymbol{\chi}_L$. Habitualmente, esto se prueba utilizando la técnica de la contradicción. Vamos a estudiar dos ejemplos. En ambos casos, se dará la prueba formal.

5.4.1 Lenguaje $L_{\text{sí}}$: reconocible pero no decidible

Sea el siguiente lenguaje:

$$L_{\text{sí}} = \{\langle Q, d\rangle \mid \langle Q, d\rangle \in \mathbb{A}^* \wedge \text{ el } \lambda\text{-término } Q \text{ responde “Sí” para el dato } d\}$$

$L_{\text{sí}}$ es el lenguaje formado por palabras de la forma $\langle Q, d\rangle$ que representan pares (λ-término, dato de entrada) tal que Q responde “Sí” para d.

En cuanto al alfabeto $\mathbb{A}$, tenemos que $\mathbb{A} = \{\lambda, ., (,), a, b, c, \ldots, y, z, 0, 1, 2, \ldots, 8, 9\}$. Cualquier λ-término puede ser formulado utilizando los símbolos de ese alfabeto.

$\langle Q, d\rangle$ es una palabra mediante la cual se representan el λ-término Q y el dato de entrada d para Q.

5.4.1.1 $L_{\text{sí}}$ es reconocible

Para probar que el lenguaje $L_{\text{sí}}$ es reconocible, hay que probar que existe un λ-término que calcula la función semicaracterística $\boldsymbol{\Sigma}_{L_{\text{sí}}}$.

$$\boldsymbol{\Sigma}_{L_{\text{sí}}} : \mathbb{A}^* \to \{1, \bot\}$$

$$\boldsymbol{\Sigma}_{L_{\text{sí}}}(\langle Q, d\rangle) = \begin{cases} 1 & \text{si } \langle Q, d\rangle \in L_{\text{sí}} \\ \bot & \text{si } \langle Q, d\rangle \notin L_{\text{sí}} \end{cases}$$

O puesto de otra forma:

$$\Sigma_{L_{\text{sí}}} : \mathbb{A}^* \to \{1, \perp\}$$

$$\Sigma_{L_{\text{sí}}}(\langle Q, d\rangle) = \begin{cases} 1 & \text{si se cumple } (Qd) = \text{“Sí”} \\ \perp & \text{si se cumple } (Qd) = \text{“No” o } (Qd) = \perp \end{cases}$$

Es decir, hay que probar que existe un λ-término que, dada una palabra de la forma $\langle Q, d\rangle$ responde que “Sí” si el λ-término Q responde que “Sí” para el λ-término d y cicla ($\perp$) si el λ-término Q no responde que “Sí” para el λ-término d.

En la figura 5.4.1 de la página 22, se muestra un esquema en el que se describe gráficamente un λ-término universal $\mathcal{U}$ que calcula la función semicaracterística $\Sigma_{L_{\text{sí}}}$.

El λ-término universal $\mathcal{U}$ recibe, como dato de entrada, una palabra de la forma $\langle Q, d\rangle$, donde Q es un λ-término y d es un λ-término que representa un dato de entrada para Q. El λ-término universal $\mathcal{U}$ es capaz de reconocer el lenguaje $L_{\text{sí}}$ por medio del siguiente algoritmo:

- El λ-término universal $\mathcal{U}$ aplica Q a d, es decir, simula la ejecución de Q sobre d.
- Si Q responde que “Sí” para d, entonces $\mathcal{U}$ responde que “Sí” para $\langle Q, d\rangle$.
- Si Q responde que “No” para d, entonces $\mathcal{U}$ cicla intencionadamente y no responde para $\langle Q, d\rangle$.
- Si Q cicla con d, es decir, si Q no responde para d, entonces $\mathcal{U}$ se queda a la espera de la respuesta de Q. Pero como la respuesta de Q no llega, $\mathcal{U}$ tampoco responde, es decir, cicla de manera infinita para la palabra $\langle Q, d\rangle$.

Por tanto, $L_{\text{sí}}$ es reconocible porque $\mathcal{U}$ responde con “Sí” a todas las palabras de la forma $\langle Q, d\rangle$ pertenecientes a $L_{\text{sí}}$ y cicla (no responde) a las palabras que no pertenecen a $L_{\text{sí}}$.

Es importante darse cuenta de que si una palabra de la forma $\langle Q, d\rangle$ no pertenece al lenguaje $L_{\text{sí}}$, es decir, si el λ-término Q responde “No” o cicla con el dato d, entonces el λ-término $\mathcal{U}$ cicla con la palabra $\langle Q, d\rangle$. Por tanto, para las palabras que no son de $L_{\text{sí}}$ el λ-término $\mathcal{U}$ cicla.

5.4.1.2 $L_{\text{sí}}$ no es decidible

Para probar que $L_{\text{sí}}$ es decidible, hay que probar que existe un λ-término que calcula la función característica $\chi_{L_{\text{sí}}}$.

$$\chi_{L_{\text{sí}}} : \mathbb{A}^* \to \mathbb{2}$$

$$\chi_{L_{\text{sí}}}(\langle Q, d\rangle) = \begin{cases} 1 & \text{si } \langle Q, d\rangle \in L_{\text{sí}} \\ 0 & \text{si } \langle Q, d\rangle \notin L_{\text{sí}} \end{cases}$$

O puesto de otra forma:

$$\chi_{L_{\text{sí}}} : \mathbb{A}^* \to \mathbb{2}$$

$$\chi_{L_{\text{sí}}}(\langle Q, d\rangle) = \begin{cases} 1 & \text{si se cumple } (Qd) = \text{“Sí”} \\ 0 & \text{si se cumple } (Qd) = \text{“No” o } (Qd) = \bot \end{cases}$$

Es decir, hay que diseñar un λ-término que responde “Sí” a las palabras (o pares) de la forma $\langle Q, d\rangle$ que pertenecen al lenguaje $L_{\text{sí}}$, y que responde “No” a las palabras que no pertenecen a $L_{\text{sí}}$. Por tanto, el λ-término no ha de ciclar nunca. Esto significa que hay que diseñar un λ-término que tiene el siguiente comportamiento:

- Responde “Sí” a las palabras (o pares) de la forma $\langle Q, d\rangle$ que cumplen que el λ-término Q responde “Sí” para el dato d.
- Responde “No” a las palabras (o pares) de la forma $\langle Q, d\rangle$ que cumplen que el λ-término Q responde “No” o cicla para el dato d.

El λ-término $\mathcal{U}$ del apartado anterior no sirve para probar que $L_{\text{sí}}$ es decidible porque $\mathcal{U}$ cicla para las palabras (o pares) de la forma $\langle Q, d\rangle$ que cumplen que el λ-término Q responde “No” o cicla para el dato d.

Ahora vamos a probar, utilizando la técnica de la contradicción, que $L_{\text{sí}}$ no es decidible.

Para ello, vamos a suponer que es decidible, es decir, vamos a suponer que existe un λ-término universal V que calcula la función característica $\chi_{L_{\text{sí}}}$. Dada cualquier palabra de la forma $\langle Q, d\rangle$, el λ-término universal V será capaz de responder que “Sí” para $\langle Q, d\rangle$ si Q responde que “Sí” para d y será capaz de responder que “No” para $\langle Q, d\rangle$ en caso contrario (es decir, V responde que “No” para $\langle Q, d\rangle$ si Q responde “No” o Q cicla para d). Por tanto, V nunca cicla.

En la figura 5.4.2 de la página 22 se muestra un esquema en el que se puede apreciar gráficamente la idea expuesta sobre el funcionamiento de V. Al λ-término V se le dará una palabra de la forma $\langle Q, d\rangle$, es decir, un λ-término Q y un dato de entrada d para el λ-término Q. Se ejecutará la función Q con el dato d y V recogerá el resultado:

- Si el λ-término Q responde “Sí” para el dato d, entonces el λ-término V devolverá “Sí” para la palabra $\langle Q, d\rangle$.
- Si el λ-término Q responde “No” para el dato d o cicla (es decir, no responde) con el dato d, entonces el λ-término V devolverá “No” para la palabra $\langle Q, d\rangle$.
- V nunca cicla.

Ahora generamos la contradicción. La idea es que si existe V, entonces podemos construir o diseñar un λ-término universal Z que es contradictorio.

La descripción del λ-término universal Z es la siguiente:

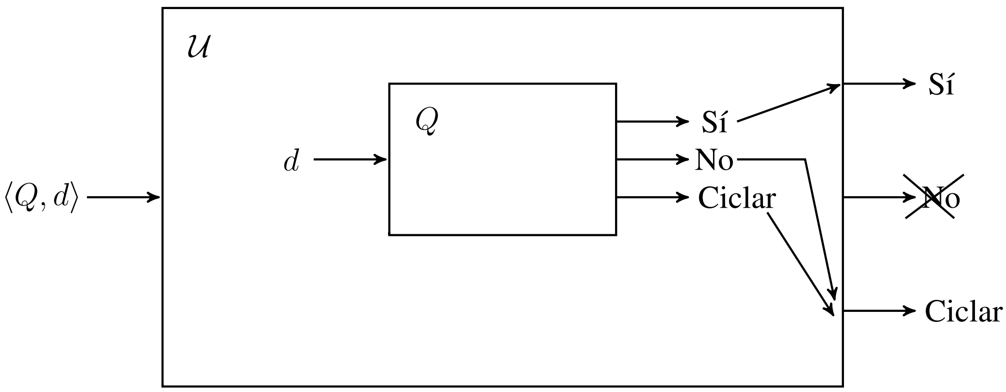

Figura 5.4.1. λ-término universal $\mathcal{U}$ que reconoce el lenguaje $L_{\text{sí}}$, es decir, calcula $\Sigma_{L_{\text{sí}}}$. Apartado 5.4.1.1 de la página 19.

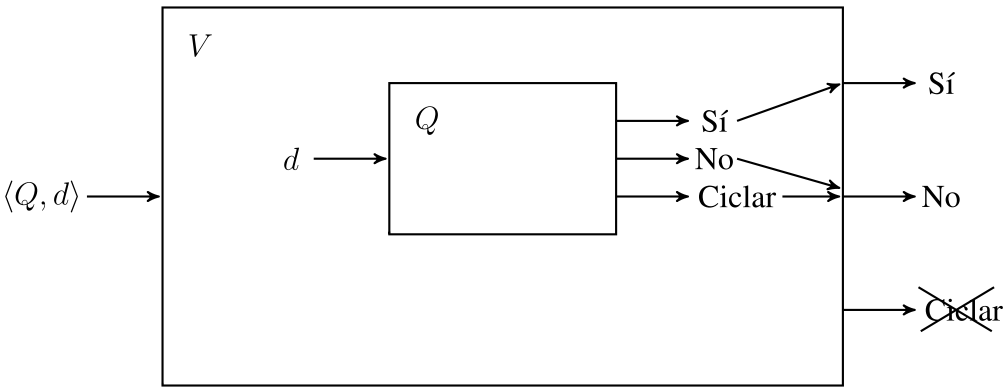

Figura 5.4.2. λ-término universal V que, de existir, decidiría el lenguaje $L_{\text{sí}}$, es decir, calcularía $\chi_{L_{\text{sí}}}$. Apartado 5.4.1.2 de la página 20.

> El λ-término universal Z recibe como entrada un λ-término $\langle Q\rangle$ y responde "No" si V responde "Sí" para el par $\langle Q, Q\rangle$ y responde "Sí" si V responde "No" para el par $\langle Q, Q\rangle$. Por tanto, Z responde para $\langle Q\rangle$ lo contrario de lo que responde V para $\langle Q, Q\rangle$.

En la figura 5.4.3 de la página 25 se muestra un esquema en el que se puede apreciar gráficamente la idea expuesta sobre el funcionamiento de Z. Al λ-término Z se le dará una palabra de la forma $\langle Q\rangle$, es decir, un λ-término Q. El λ-término Z llama a la máquina V con el dato de entrada $\langle Q, Q\rangle$. Ejecutar V con dato de entrada $\langle Q, Q\rangle$ significa que V ejecutará Q con dato de entrada $\langle Q\rangle$. Si el λ-término Q responde "Sí" para el dato Q, entonces el λ-término V devolverá "Sí" para la palabra $\langle Q, Q\rangle$. Si el λ-término Q responde "No" para el dato Q o cicla (es decir, no responde) con el dato Q, entonces el λ-término V devolverá "No" para la palabra $\langle Q, Q\rangle$. Una vez que V haya respondido, el λ-término Z responderá lo contrario de lo que ha respondido V: Si V ha respondido "Sí" para la palabra $\langle Q, Q\rangle$, entonces Z responderá "No" para la palabra $\langle Q\rangle$; Si V ha respondido "No" para la palabra $\langle Q, Q\rangle$, entonces Z responderá "Sí" para la palabra $\langle Q\rangle$.

El λ-término Z nunca cicla. Es decir, siempre responde.

Nótese que Z necesita solo un argumento (la representación de un λ-término Q) mientras que V necesita dos argumentos (la representación de un λ-término Q y un dato de entrada d para Q). Puesto que Z solo tiene un dato de entrada $\langle Q\rangle$ y necesita pasar dos datos a V, lo que hace es pasarle a V el dato duplicado: $\langle Q, \underbrace{Q}_{d}\rangle$. El segundo Q hace el papel de d.

¿Dónde surge la contradicción? La contradicción surge cuando al λ-término Z le pasamos como dato de entrada su propia descripción, es decir, la palabra $\langle Z\rangle$:

- Cuando al λ-término (o función) Z se le pasa la palabra $\langle Z\rangle$, el λ-término Z llama al λ-término (o función) V pasándole el par $\langle Z, Z\rangle$ para que V ejecute la función Z con dato de entrada $\langle Z\rangle$ (esta segunda Z sustituye al parámetro d).

- Si el λ-término Z responde "Sí" para la palabra $\langle Z\rangle$, entonces el λ-término V devolverá "Sí" para la palabra o par $\langle Z, Z\rangle$ y el λ-término Z responderá lo contrario, es decir, el λ-término Z responderá "No" para la palabra $\langle Z\rangle$. CONTRADICCIÓN: Si el λ-término Z responde "Sí" para la palabra $\langle Z\rangle$, entonces el λ-término Z responde "No" para la palabra $\langle Z\rangle$.

- Si el λ-término Z responde "No" o cicla para la palabra $\langle Z\rangle$, entonces la función o el λ-término V devolverá "No" para la palabra o par $\langle Z, Z\rangle$ y el λ-término Z responderá lo contrario, es decir, el λ-término Z responderá "Sí" para la palabra $\langle Z\rangle$. CONTRADICCIÓN: Si el λ-término Z responde "No" o cicla para la palabra $\langle Z\rangle$, entonces el λ-término Z responde "Sí" para la palabra $\langle Z\rangle$.

En definitiva, tenemos que el λ-término Z devuelve distintos resultados para un mismo dato $\langle Z\rangle$. Pero una función o λ-término siempre ha de devolver el mismo resultado para un dato de entrada concreto.

Como el suponer que $L_{\text{sí}}$ es decidible nos ha llevado a una contradicción, concluimos que $L_{\text{sí}}$ no es decidible.

En la figura 5.4.4 de la página 25, se muestra un esquema en el que se puede apreciar gráficamente la idea expuesta sobre la contradicción que Z genera cuando recibe el dato de entrada $\langle Z\rangle$. Recuérdese que Z necesita solo un argumento mientras que V necesita dos argumentos (un λ-término y un dato de entrada para dicho λ-término). Puesto que Z solo tiene un dato de entrada $\langle Z\rangle$ y necesita pasar dos datos a V, lo que hace es pasarle a V el dato duplicado: $\langle Z, \underbrace{Z}_{d}\rangle$. El segundo Z hace el papel de d en el caso general.

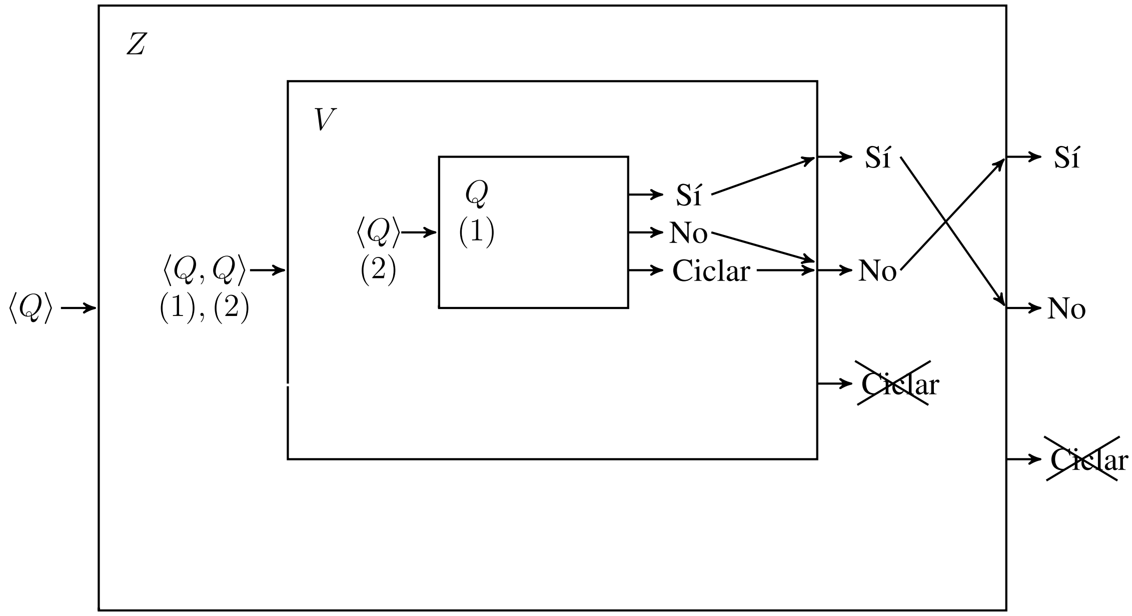

Figura 5.4.3. λ-término universal Z que genera contradicción y sirve para probar que $L_{\text{sí}}$ no es decidible. Apartado 5.4.1.2 de la página 20.

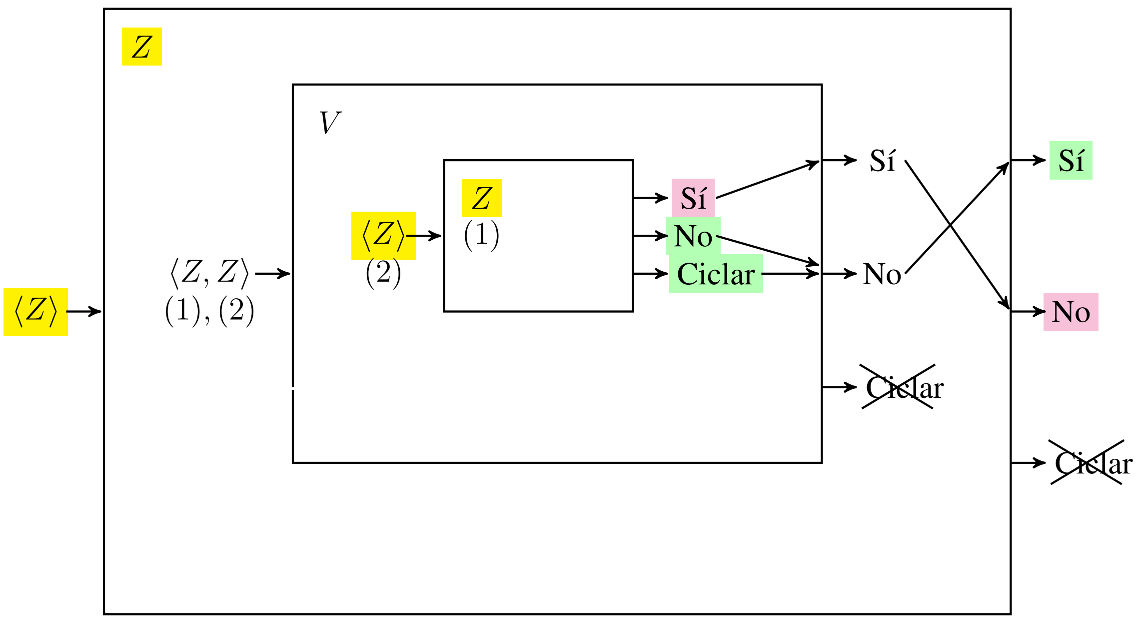

Figura 5.4.4. Contradicción que genera el λ-término universal Z cuando recibe el dato de entrada $\langle Z \rangle$. Apartado 5.4.1.2 de la página 20.

5.4.2 Lenguaje $L_{\text{terminación}}$: reconocible pero no decidible

Sea el siguiente lenguaje:

$$L_{\text{terminación}} = \{\langle Q, d\rangle \mid \langle Q, d\rangle \in \mathbb{A}^* \wedge \text{ el } \lambda\text{-término } Q \text{ responde “Sí” o “No” para el dato } d\}$$

$L_{\text{terminación}}$ es el lenguaje formado por palabras de la forma $\langle Q, d\rangle$ que representan pares (λ-término, dato de entrada) tal que el λ-término Q termina el cálculo correspondiente a d y responde “Sí” o “No”.

En cuanto al alfabeto $\mathbb{A}$, tenemos que $\mathbb{A} = \{\lambda, ., (,), a, b, c, \ldots, y, z, 0, 1, 2, \ldots, 8, 9\}$. Cualquier λ-término puede ser formulado utilizando los símbolos de ese alfabeto.

El problema de averiguar si un λ-término Q termina o no termina el cálculo correspondiente a un dato de entrada d, es conocido como el **problema de parada** (*halt problem* en inglés).

5.4.2.1 $L_{\text{terminación}}$ es reconocible

Para probar que el lenguaje $L_{\text{terminación}}$ es reconocible, hay que probar que existe un λ-término que calcula la función semicaracterística $\Sigma_{L_{\text{terminación}}}$.

$$\Sigma_{L_{\text{terminación}}} : \mathbb{A}^* \to \{1, \bot\}$$

$$\Sigma_{L_{\text{terminación}}}(\langle Q, d\rangle) = \begin{cases} 1 & \text{si se cumple } \langle Q, d\rangle \in L_{\text{terminación}} \\ \bot & \text{si se cumple } \langle Q, d\rangle \notin L_{\text{terminación}} \end{cases}$$

O puesto de otra forma:

$$\Sigma_{L_{\text{terminación}}} : \mathbb{A}^* \to \{1, \bot\}$$

$$\Sigma_{L_{\text{terminación}}}(\langle Q, d\rangle) = \begin{cases} 1 & \text{si se cumple } (Qd) = \text{“Sí” o } (Qd) = \text{“No”} \\ \bot & \text{si se cumple } (Qd) = \bot \end{cases}$$

Es decir, hay que probar que existe un λ-término que para una palabra de la forma $\langle Q, d\rangle$ responde que “Sí” si el λ-término Q responde —con un “Sí” o con un “No”— al dato d.

En la figura 5.4.5 de la página 29, se muestra un esquema en el que se describe gráficamente un λ-término universal R que calcula la función semicaracterística $\Sigma_{L_{\text{terminación}}}$.

El λ-término universal R recibe, como dato de entrada, una palabra de la forma $\langle Q, d\rangle$, donde Q es un λ-término y d es otro λ-término que representa un dato de entrada para Q. El λ-término R es capaz de reconocer el lenguaje $L_{\text{terminación}}$ por medio del siguiente algoritmo:

- El λ-término R aplica Q a d, es decir, simula la ejecución de Q sobre d.

- Si Q responde que "Sí" para d, entonces R responde que "Sí" para $\langle Q, d\rangle$.
- Si Q responde que "No" para d, entonces R responde que "Sí" para $\langle Q, d\rangle$.
- Si Q cicla con d, es decir, si Q no responde para d, entonces R se queda a la espera de la respuesta de Q. Pero como la respuesta de Q no llega, R tampoco responde, es decir, cicla de manera infinita para la palabra $\langle Q, d\rangle$.

Por tanto, $L_{\text{terminación}}$ es reconocible porque R responde con "Sí" a todas las palabras de la forma $\langle Q, d\rangle$ pertenecientes a $L_{\text{terminación}}$ y cicla (no responde) a las palabras que no pertenecen a $L_{\text{terminación}}$.

Es importante darse cuenta de que si una palabra de la forma $\langle Q, d\rangle$ no pertenece al lenguaje $L_{\text{terminación}}$, es decir, si el λ-término Q cicla con el dato d, entonces el λ-término R cicla con la palabra $\langle Q, d\rangle$. Por tanto, para las palabras que no son de $L_{\text{terminación}}$ el λ-término R no responde que "No" sino que cicla.

5.4.2.2 $L_{\text{terminación}}$ no es decidible

Para probar que $L_{\text{terminación}}$ es decidible, hay que probar que existe un λ-término que calcula la función característica $\chi_{L_{\text{terminación}}}$.

$$\chi_{L_{\text{terminación}}} : \mathbb{A}^* \to \mathbb{2}$$

$$\chi_{L_{\text{terminación}}}(\langle Q, d\rangle) = \begin{cases} 1 & \text{si se cumple } \langle Q, d\rangle \in L_{\text{terminación}} \\ 0 & \text{si se cumple } \langle Q, d\rangle \notin L_{\text{terminación}} \end{cases}$$

O puesto de otra forma:

$$\chi_{L_{\text{terminación}}} : \mathbb{A}^* \to \mathbb{2}$$

$$\chi_{L_{\text{terminación}}}(\langle Q, d\rangle) = \begin{cases} 1 & \text{si se cumple } (Qd) = \text{“Sí” o } (Qd) = \text{“No”} \\ 0 & \text{si se cumple } (Qd) = \bot \end{cases}$$

Es decir, hay que diseñar un λ-término que responde "Sí" a las palabras (o pares) de la forma $\langle Q, d\rangle$ que pertenecen al lenguaje $L_{\text{terminación}}$, y que responde "No" a las palabras que no pertenecen a $L_{\text{terminación}}$. Por tanto, el λ-término no ha de ciclar nunca. Esto significa que hay que diseñar un λ-término que tiene el siguiente comportamiento:

- Responde "Sí" a las palabras (o pares) de la forma $\langle Q, d\rangle$ que cumplen que el λ-término Q responde "Sí" o "No" para el dato d.
- Responde "No" a las palabras (o pares) de la forma $\langle Q, d\rangle$ que cumplen que el λ-término Q cicla para el dato d.

El λ-término R del apartado anterior no sirve para probar que $L_{\text{terminación}}$ es decidible porque R cicla para las palabras que no son de ese lenguaje, es decir, R cicla para las palabras de la forma $\langle Q, d\rangle$ que cumplen que el λ-término Q cicla para la palabra d.

Ahora vamos a probar, utilizando la técnica de la contradicción, que $L_{\text{terminación}}$ no es decidible.

Para ello, vamos a suponer que es decidible, es decir, vamos a suponer que existe un λ-término universal H que calcula la función característica $\chi_{L_{\text{terminación}}}$. Dada cualquier palabra de la forma $\langle Q, d\rangle$, el λ-término universal H será capaz de responder que "Sí" para $\langle Q, d\rangle$ si Q responde que "Sí" o que "No" para d y será capaz de responder que "No" para $\langle Q, d\rangle$ si Q cicla para d. Por tanto, H nunca cicla.

En la figura 5.4.6 de la página 29 se muestra un esquema en el que se puede apreciar gráficamente la idea expuesta sobre el funcionamiento de H. Al λ-término H se le dará una palabra de la forma $\langle Q, d\rangle$, es decir, la representación de un λ-término Q y un dato de entrada d para el λ-término Q. Se ejecutará la función Q con el dato d y H recogerá el resultado:

- Si el λ-término Q responde "Sí" o responde "No" para el dato d, entonces el λ-término H devolverá "Sí" para la palabra $\langle Q, d\rangle$.
- Si el λ-término Q cicla (es decir, no responde) con la palabra d, entonces la máquina H devolverá "No" para la palabra $\langle Q, d\rangle$.
- H nunca cicla.

Ahora generamos la contradicción. La idea es que si existe H, entonces podemos construir o diseñar un λ-término universal G que es contradictorio en el sentido de que G será capaz de decidir si una palabra $\langle Q, d\rangle$ es del lenguage $L_{\text{sí}}$ del ejemplo anterior. Dicho de otra forma, G será capaz de calcular la función característica $\chi_{L_{\text{sí}}}$. Pero ya hemos probado antes que $L_{\text{sí}}$ no es decidible, es decir, ya hemos probado antes que $\chi_{L_{\text{sí}}}$ no es calculable.

La descripción del λ-término universal G es la siguiente:

> El λ-término (o función) G recibe como entrada la descripción $\langle Q, d\rangle$ de un λ-término Q y un dato de entrada d para Q. La máquina G llama a la máquina H pasándole $\langle Q, d\rangle$. El λ-término H aplicará el λ-término Q al dato d y, a continuación, H recogerá el resultado:
>
> - Si el λ-término Q responde "Sí" o "No" para el dato d, entonces el λ-término H devolverá "Sí" para la palabra $\langle Q, d\rangle$ y, en ese caso, el λ-término G sabe que Q responde "Sí" o "No" para d y, a continuación, G vuelve a ejecutar Q con dato d. Si Q responde "Sí" para d, entonces G responde "Sí" para $\langle Q, d\rangle$. Si Q responde "No" para d, entonces G responde "No" para $\langle Q, d\rangle$.

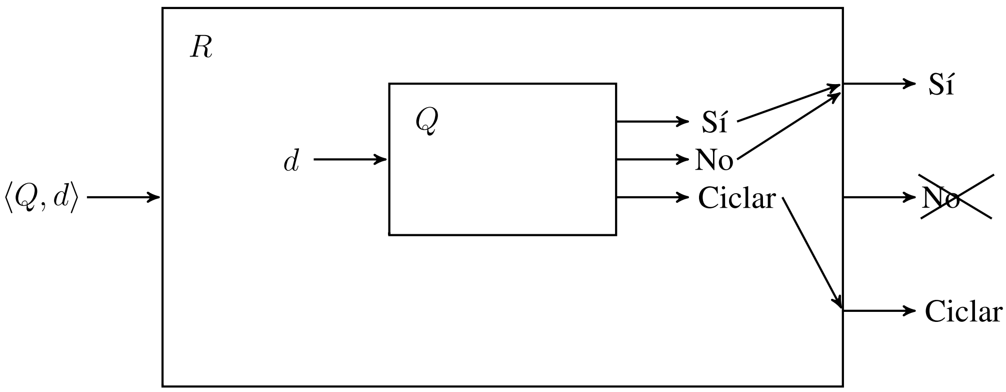

Figura 5.4.5. λ-término universal R que reconoce el lenguaje $L_{\text{terminación}}$, es decir, calcula $\Sigma_{L_{\text{terminación}}}$. Apartado 5.4.2.1 de la página 26.

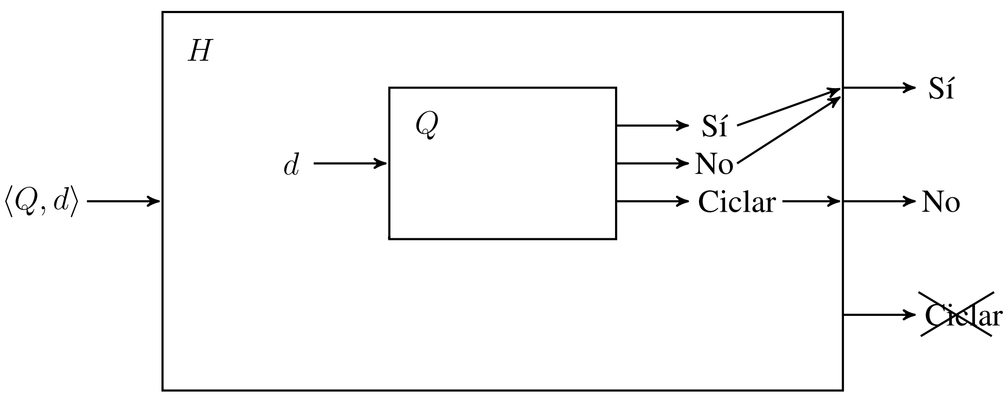

Figura 5.4.6. λ-término universal H que, de existir, decidiría el lenguaje $L_{\text{terminación}}$, es decir, calcularía $\chi_{L_{\text{terminación}}}$. Apartado 5.4.2.2 de la página 27.

- Si el λ-término Q cicla (es decir, no responde) con el dato d, entonces el λ-término H devolverá “No” para la palabra $\langle Q, d\rangle$. En ese caso, G responderá directamente “No” para la palabra $\langle Q, d\rangle$, sin volver a ejecutar Q.

La máquina G nunca cicla. Es decir, siempre responde.

En la figura 5.4.7 de la página 31 se muestra un esquema en el que se puede apreciar gráficamente la idea expuesta sobre el funcionamiento de G.

¿Dónde surge la contradicción? El λ-término G sirve para decidir el lenguaje $L_{\text{sí}}$. Dicho de otra forma, G sirve para calcular la función característica $\chi_{L_{\text{sí}}}$. Pero ya sabemos que $L_{\text{sí}}$ no es decidible. Dicho de otra forma, ya sabemos que $\chi_{L_{\text{sí}}}$ no es calculable. CONTRADICCIÓN: Si existe el λ-término H, entonces el lenguaje $L_{\text{sí}}$ es decidible, es decir, $\chi_{L_{\text{sí}}}$ es calculable, pero anteriormente hemos probado que $L_{\text{sí}}$ no es decidible, es decir, $\chi_{L_{\text{sí}}}$ no es calculable (Apartado 5.4.1.2, página 20). Como la existencia de H genera una contradicción, deducimos que H no puede existir y, por tanto, $L_{\text{terminación}}$ no es decidible.

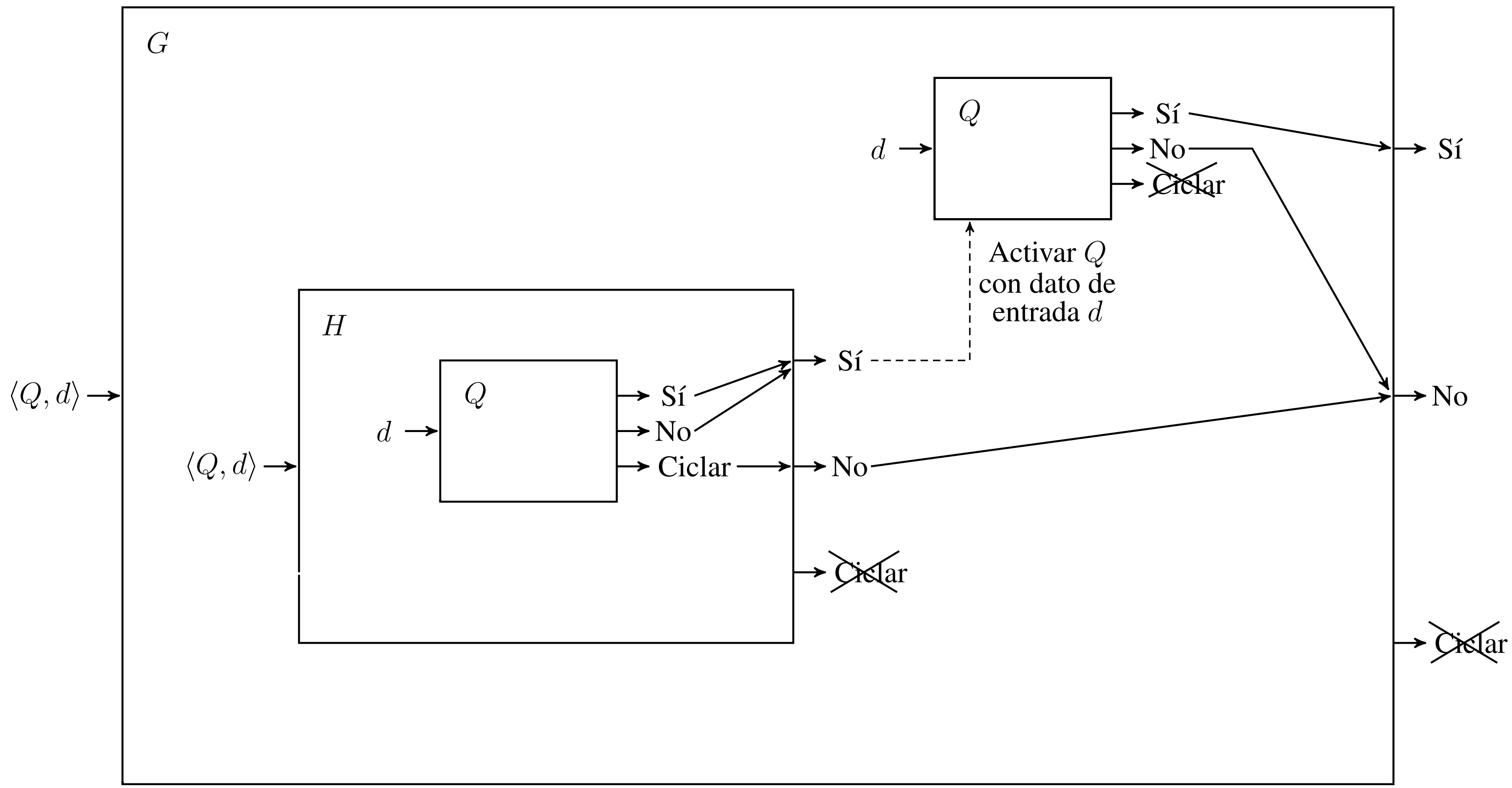

Figura 5.4.7. λ-término universal G que genera contradicción y sirve para probar que $L_{\text{terminación}}$ no es decidible. De existir G, decidiría el lenguaje $L_{\text{sí}}$, es decir, calcularía $\chi_{L_{\text{sí}}}$. Apartado 5.4.2.2 de la página 27.

5.5.

Lenguajes no reconocibles

Sabemos que hay lenguajes no reconocibles porque ya hemos probado que hay lenguajes cuyas funciones semicaracterísticas no son computables (Tema 3).

En este apartado, se dará un ejemplo concreto de lenguaje no reconocible y se probará formalmente que ese lenguaje concreto no es reconocible.

5.5.1 El lenguaje $\overline{L_{\text{sí}}}$ no es reconocible

Consideramos el lenguaje $\overline{L_{\text{sí}}}$, es decir, el lenguaje complementario de $L_{\text{sí}}$:

$$L_{\text{sí}} = \{\langle Q, d\rangle \mid \langle Q, d\rangle \in \mathbb{A}^* \wedge \text{ el } \lambda\text{-término } Q \text{ responde “Sí” para el dato } d\}$$

$$\overline{L_{\text{sí}}} = \{\langle Q, d\rangle \mid \langle Q, d\rangle \in \mathbb{A}^* \wedge \text{ el } \lambda\text{-término } Q \text{ responde “No” o cicla (es decir, no responde) para el dato } d\}$$

donde $\mathbb{A} = \{\lambda, ., (,), a, b, c, \ldots, y, z, 0, 1, 2, \ldots, 8, 9\}$.

Para probar que el lenguaje $\overline{L_{\text{sí}}}$ es reconocible, hay que probar que existe un λ-término que calcula la función semicaracterística $\Sigma_{\overline{L_{\text{sí}}}}$.

$$\Sigma_{\overline{L_{\text{sí}}}} : \mathbb{A}^* \to \{1, \perp\}$$

$$\Sigma_{\overline{L_{\text{sí}}}}(\langle Q, d\rangle) = \begin{cases} 1 & \text{si se cumple } \langle Q, d\rangle \in \overline{L_{\text{sí}}} \\ \perp & \text{si se cumple } \langle Q, d\rangle \notin \overline{L_{\text{sí}}} \end{cases}$$

O puesto de otra forma:

$$\Sigma_{\overline{L_{\text{sí}}}} : \mathbb{A}^* \to \{1, \perp\}$$

$$\Sigma_{\overline{L_{\text{sí}}}}(\langle Q, d\rangle) = \begin{cases} 1 & \text{si se cumple } (Qd) = \text{“No” o } (Qd) = \perp \\ \perp & \text{si se cumple } (Qd) = \text{“Sí”} \end{cases}$$

Es decir, hay que probar que existe un λ-término que responda "Sí" a las palabras de la forma $\langle Q, d\rangle$ que pertenecen a $\overline{L_{\text{sí}}}$, y que cicle para las palabras que no pertenecen a $\overline{L_{\text{sí}}}$. Es decir, hay que diseñar un λ-término que responde "Sí" a las palabras de la forma $\langle Q, d\rangle$ que cumplen la condición de que el λ-término Q responde "No" o cicla para el dato d y que cicla para las palabras de la forma $\langle Q, d\rangle$ que cumplen la condición de que el λ-término Q responde "Sí" para el dato d.

Vamos a probar, utilizando la técnica de la contradicción, que $\overline{L_{\text{sí}}}$ no es reconocible, es decir, vamos a probar que la función semicaracterística $\boldsymbol{\Sigma}_{\overline{L_{\text{sí}}}}$ no es calculable.

Para ello, vamos a suponer que $\overline{L_{\text{sí}}}$ es reconocible, es decir, vamos a suponer que existe un λ-término universal J que calcula la función semicaracterística $\boldsymbol{\Sigma}_{\overline{L_{\text{sí}}}}$. Por tanto, suponemos que existe un λ-término J que para cualquier palabra de la forma $\langle Q, d\rangle$ responde que "Sí" en caso de que $\langle Q, d\rangle$ pertenezca a $\overline{L_{\text{sí}}}$. Para las palabras que no pertenecen a $\overline{L_{\text{sí}}}$ el λ-término J cicla.

Al λ-término J se le dará una palabra de la forma $\langle Q, d\rangle$, es decir, la representación de un λ-término Q y un dato de entrada d para el λ-término Q. Se ejecutará el λ-término Q con el dato d y J recogerá el resultado:

- Si el λ-término Q responde "Sí" para el dato d, entonces el λ-término J ciclará (no responderá) para el dato $\langle Q, d\rangle$.
- Si el λ-término Q responde "No" para el dato d o cicla (es decir, no responde), entonces el λ-término J devolverá "Sí" para la palabra $\langle Q, d\rangle$.

En la figura 5.5.1 de la página 36 se muestra un esquema en el que se puede apreciar gráficamente la idea expuesta sobre el funcionamiento de J.

Por otra parte, sabemos que el lenguaje $L_{\text{sí}}$ es reconocible. De hecho, en el apartado 5.4.1.1 (página 19), hemos diseñado un λ-término $\mathcal{U}$ que responde que "Sí" exactamente a aquellas palabras de la forma $\langle Q, d\rangle$ que pertenecen a $L_{\text{sí}}$. Dicho de otra forma, el λ-término $\mathcal{U}$ reconoce el lenguaje $L_{\text{sí}}$. En la figura 5.4.1 de la página 22, se muestra un esquema en el que se puede apreciar gráficamente la idea expuesta sobre el funcionamiento de $\mathcal{U}$.

Ahora generamos la contradicción. La idea es que si existe J, entonces podemos construir o diseñar un λ-término universal K que es contradictorio en el sentido de que K será capaz de decidir si una palabra $\langle Q, d\rangle$ es del lenguage $L_{\text{sí}}$. Dicho de otra forma, K será capaz de calcular la función característica $\boldsymbol{\chi}_{L_{\text{sí}}}$. Pero ya hemos probado antes que $L_{\text{sí}}$ no es decidible, es decir, ya hemos probado antes que $\boldsymbol{\chi}_{L_{\text{sí}}}$ no es calculable.

La descripción del λ-término universal K es la siguiente:

> El λ-término (o función) K recibe como entrada la descripción $\langle Q, d\rangle$ de un λ-término Q y un dato de entrada d para Q. El λ-término K llama en paralelo (es decir, a la vez) a los λ-términos $\mathcal{U}$ y J pasándoles la palabra $\langle Q, d\rangle$. Los λ-términos

$\mathcal{U}$ y J llamarán (por separado) al λ-término Q con el dato d y, a continuación, esperarán al resultado. $\mathcal{U}$ o J (una de las dos) ha de responder que "Sí" porque o d pertenece a $L_{\text{sí}}$ (y $\mathcal{U}$ responde "Sí") o d no pertenece a $L_{\text{sí}}$ (y J responde "Sí"):

- Si el λ-término Q responde "Sí" para el dato d, entonces el λ-término $\mathcal{U}$ devolverá "Sí" para la palabra $\langle Q, d\rangle$ y, en ese caso, el λ-término K también devolverá "Sí" para la palabra $\langle Q, d\rangle$.
- Si el λ-término Q responde "No" o cicla (es decir, no responde) con el dato d, entonces el λ-término J devolverá "Sí" para la palabra $\langle Q, d\rangle$ y, en ese caso, el λ-término K devolverá "No" para la palabra $\langle Q, d\rangle$.

El λ-término K nunca cicla. Es decir, siempre responde.

En la figura 5.5.2 de la página 37, se muestra un esquema en el que se puede apreciar gráficamente la idea expuesta sobre el funcionamiento de K.

¿Dónde surge la contradicción? El λ-término K sirve para decidir el lenguaje $L_{\text{sí}}$. Dicho de otra forma, K sirve para calcular la función característica $\chi_{L_{\text{sí}}}$. Pero ya sabemos que $L_{\text{sí}}$ no es decidible. Dicho de otra forma, ya sabemos que $\chi_{L_{\text{sí}}}$ no es calculable. CONTRADICCIÓN: Si existe el λ-término J, entonces el lenguaje $L_{\text{sí}}$ es decidible, es decir, $\chi_{L_{\text{sí}}}$ es calculable, pero anteriormente hemos probado que $L_{\text{sí}}$ no es decidible, es decir, $\chi_{L_{\text{sí}}}$ no es calculable (Apartado 5.4.1.2, página 20). Como la existencia de J genera una contradicción, deducimos que J no puede existir y, por tanto, $\overline{L_{\text{sí}}}$ no es reconocible.

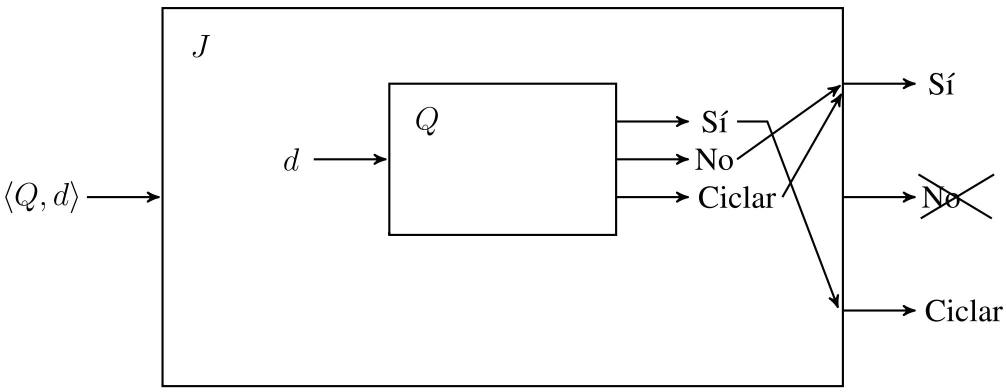

Figura 5.5.1. λ-término universal J que, de existir, reconocería el lenguaje $\overline{L_{\text{sí}}}$ (complementario de $L_{\text{sí}}$), es decir, calcularía $\Sigma_{\overline{L_{\text{sí}}}}$. Apartado 5.5.1 de la página 33.

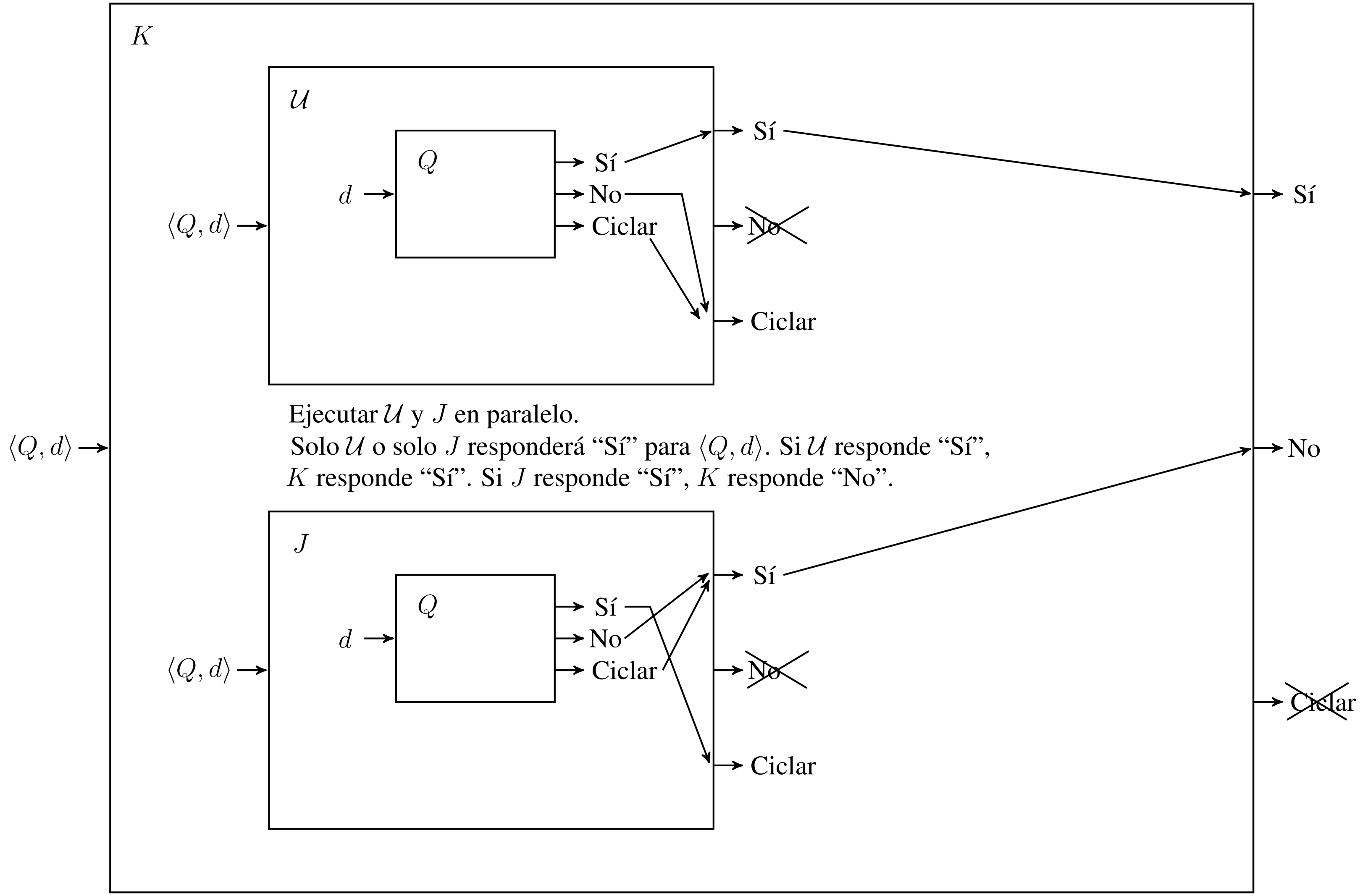

Figura 5.5.2. λ-término universal K que genera contradicción y sirve para probar que $\overline{L_{\text{sí}}}$ no es reconocible. De existir K, decidiría el lenguaje $L_{\text{sí}}$, es decir, calcularía $\chi_{L_{\text{sí}}}$. Apartado 5.5.1 de la página 33.