El Problema de la Detención de la Máquina de Turing

Departamento de Teoría de la Computación
Facultad de Informática
Universidad Nacional del Comahue

Teoría de la Computación II - 2015

Temario

El Problema de la Detención de las MT

Este es un problema de decisión de particular importancia para la Ciencia de la Computación.

El Problema de la Detención de las MT (The Halting Problem)

Dada una MT T y una cadena α, ¿existe un algoritmo para decidir si T se detendrá con α como entrada?

Turing, a fines de 1930, enunció y demostró que este problema no es soluble :(

El Problema de la Detención de las MT

Teorema

El problema de la detención de la máquina de Turing no es algorítmicamente soluble.

Noción Intuitiva

¿Cómo podemos analizar este problema desde un punto de vista procedural o computacional?

Supongamos que el problema es soluble y que el procedimiento (o algoritmo):

Halt(*P*,*E*,**Res**)

recibe una MT *P*, una entrada *E* y resuelve el problema.

Es decir,

Si P se detiene con la entrada E, entonces Halt retorna Sí.

Si P entra en un ciclo infinito, devuelve No.

Obs: *P* sería la descripción codificada de una máquina de Turing.

Noción Intuitiva

En particular, si la entrada es la misma descripción, sería válida la llamada:

Halt(*P*,*P*,**Res**)

En este caso, *Halt* nos indica si la MT *P* se detiene o no, comenzando con *P* en la cinta.

Consideremos ahora el siguiente algoritmo que cicla para un caso particular...

Noción Intuitiva

```
Procedure Diagonal(X):
    Repetir
        Halt(X,X,Res)
    Hasta Res=No
```

Diagonal(*X*)

- **Se detiene**: sssí *Halt* responde No. Es decir, sssí *X* cicla infinitamente.
- **Cicla infinitamente**: sssí *Halt* responde Sí. Es decir, sssí *X* se detiene.

¿Qué pasa si ejecutamos *Diagonal*(*Diagonal*)?

Noción Intuitiva

¿Qué pasa si ejecutamos *Diagonal*(*Diagonal*)?

$$Diagonal(X)\begin{cases} \text{Se detiene} & \text{sssí } \textit{Halt} \text{ responde No} \\ & \text{Es decir, sssí } X \text{ cicla infinitamente.} \\ \text{Cicla infinitamente} & \text{sssí } \textit{Halt} \text{ responde Sí} \\ & \text{Es decir, sssí } X \text{ se detiene.} \end{cases}$$

El procedimiento *Diagonal* se detiene sssí *Halt* responde NO. Es decir, sssí *Diagonal* cicla infinitamente.

El procedimiento *Diagonal* cicla infinitamente sssí *Halt* responde SÍ. Es decir, sssí *Diagonal* se detiene.

Noción Intuitiva

¿Qué pasa si ejecutamos *Diagonal*(*Diagonal*)?

$$\mathit{Diagonal}(X)\begin{cases} \text{Se detiene} & \text{sssí } \mathit{Halt} \text{ responde No} \\ & \text{Es decir, sssí } X \text{ cicla infinitamente.} \\ \text{Cicla infinitamente} & \text{sssí } \mathit{Halt} \text{ responde Sí} \\ & \text{Es decir, sssí } X \text{ se detiene.} \end{cases}$$

Diagonal se detiene sssí *Diagonal* cicla infinitamente.

Diagonal cicla infinitamente sssí *Diagonal* se detiene.

Esto es un absurdo y proviene de asumir que existe un algoritmo para resolver el problema de la detención.

$\therefore$ el Problema de la Detención no es SOLUBLE.

Formalizando la noción intuitiva..

Sea el lenguaje

$$K_0 = \{ \underbrace{\rho(M)\rho(w)}_{\text{Codif. de } M \text{ y codif. de } w} : \text{la MT } M \text{ se detiene con } w\}$$

Una cadena $\rho(M)\rho(w)$ pertenece a K_0 si y solo si la MT M se detiene con la entrada w.

Formalizando la noción intuitiva..

Sea el lenguaje

$$K_0 = \{ \underbrace{\rho(M)\rho(w)}_{\text{Codif. de } M \text{ y codif. de } w} : \text{la MT } M \text{ se detiene con } w\}$$

¿Será K_0 un lenguaje recursivo?

¿Será K_0 un lenguaje recursivo enumerable?

Formalizando la noción intuitiva..

Sea el lenguaje

$$K_0 = \{ \underbrace{\rho(M)\rho(w)}_{\text{Codif. de } M \text{ y codif. de } w} : \text{la MT } M \text{ se detiene con } w\}$$

K_0 es recursivo enumerable

El lenguaje K_0 es semidecidido por la MUT. La MUT se detiene (solamente) con los miembros de K_0.

MUT($\rho(M)\rho(w)$) se detiene sssí M se detiene con w.

$\therefore$ MUT($\rho(M)\rho(w)$) semidecide a K_0.

¿Será K_0 un lenguaje recursivo?

Formalizando la noción intuitiva..

K_0 es recursivo enumerable, ¿será recursivo?

Si lo fuera, existiría una MT M_0 que decide al lenguaje K_0.

Es decir, M_0 determinaría si una MT M se detiene o no con una cadena w.

De esta manera, M_0 resolvería el problema de la detención...

Formalizando la noción intuitiva..

K_0 es recursivo enumerable, ¿será recursivo?

Si lo fuera, existiría una MT M_0 que decide al lenguaje K_0.

Si K_0 es recursivo, entonces también lo es el lenguaje

$$K_1 = \{\rho(M) : \text{la MT } M \text{ se detiene con } \rho(M)\}$$

Es decir, K_1 está formado por codificaciones de MT's M tal que M se detiene con su propia codificación como entrada.

Formalizando la noción intuitiva..

Si

$$K_1 = \{\rho(M) : \text{la MT } M \text{ se detiene con } \rho(M)\}$$

es recursivo, también lo es su complemento CK_1 (por teorema antes visto):

$$\begin{aligned} CK_1 \;&=\; \{w : w \text{ no es la codif. de una MT o es la codif.} \\ &\quad\;\; \text{de una MT } M \text{ tal que no se detiene con } \rho(M)\} \end{aligned}$$

Casualmente, el lenguaje CK_1 es análogo a nuestro procedimiento Diagonal. ¿Puede CK_1 ser recursivo?

Veremos que no. Y tampoco K_1 ni K_0..

Formalizando la noción intuitiva..

$$CK_1 = \{w : w \text{ no es la codif. de una MT o es la codif. de una MT } M \text{ tal que no se detiene con } \rho(M)\}$$

Probaremos que CK_1 no es recursivo.

¡Ni siquiera es recursivo enumerable!

Recordemos que:

Recursivo $\rightarrow$ Recursivo Enumerable

o equivalentemente

$\neg$Recursivo Enumerable $\rightarrow$ $\neg$Recursivo

Formalizando la noción intuitiva..

$$CK_1 = \{w : w \text{ no es la codif. de una MT o es la codif. de una MT } M \text{ tal que no se detiene con } \rho(M)\}$$

Supongamos que M^* es una MT que semidecide a CK_1. Luego,

- M^* se detiene con w si $w \in CK_1$.
- M^* no se detiene con w si $w \notin CK_1$.

Ahora preguntémonos,

$$¿\rho(M^*) \in CK_1 \text{ ?}$$

Formalizando la noción intuitiva..

$$CK_1 = \{w : w \text{ no es la codif. de una MT o es la codif. de una MT } M \text{ tal que no se detiene con } \rho(M)\}$$

¿$\rho(M^*) \in CK_1$?

Por definición de CK_1:

- $\rho(M^*) \in CK_1$ sssí M^* **no se detiene** con $\rho(M^*)$.

Por definición de Rec. enumerable:

- M^* **se detiene** con $\rho(M^*)$ sssí $\rho(M^*) \in CK_1$.

Absurdo!

Formalizando la noción intuitiva..

$$
\begin{array}{rcl}
CK_1 & = & \{w : w \text{ no es la codif. de una MT o es la codif.} \\
 & & \text{de una MT } M \text{ tal que no se detiene con } \rho(M)\}
\end{array}
$$

Entonces:

- CK_1 no es recursivo enumerable y, por lo tanto, tampoco es recursivo.
- K_1 no es recursivo (porque su complemento no es recursivo!).
- K_0 tampoco es recursivo (caso particular de K_1).
- Luego, M_0 no existe y el problema de la detención es **insoluble**.

¿Qué implicancias tienen los resultados que vimos sobre los lenguajes recursivos y recursivos enumerables?

Teorema

El lenguaje K_0 no es recursivo. Por lo tanto, la clase de los lenguajes recursivos es un subconjunto estricto de la clase de lenguajes recursivos enumerables.

Vimos que K_0 no es recursivo. Pero K_0 sí es recursivo enumerable porque es semidecidido por la M.U.T.

Lenguajes Recursivos y Recursivos Enumerables

¿Qué implicancias tienen los resultados que vimos sobre los lenguajes recursivos y recursivos enumerables?

Teorema

La clase de los lenguajes recursivos enumerables no es cerrada bajo complemento.

Corolario

Existen lenguajes que no son recursivos enumerables.

Vimos que K_1 es recursivo enumerable pero comprobamos que CK_1 (el complemento de K_1) no lo es.

Lenguajes Recursivos y Recursivos Enumerables

Ya vimos que si un lenguaje es recursivo entonces también es recursivo enumerable.

También puede probarse que:

Teorema

Un lenguaje L es recursivo sssi L y su complemento $\overline{L}$ son recursivos enumerables.

Teorema

Un lenguaje L es recursivo sssi L y su complemento $\overline{L}$ son recursivos enumerables.

Demostración ($\Rightarrow$)

Si L es recursivo entonces L es recursivo enumerable (por teorema demostrado anteriormente).

Si L es recursivo entonces $\overline{L}$ es recursivo (teorema probado anteriormente) y, por lo tanto, es recursivo enumerable.

Dem. ($\Leftarrow$): L y $\overline{L}$ son recursivos enumerables entonces L es recursivo

Supongamos que L y $\overline{L}$ son semidecididos por M_1 y M_2, respectivamente. Entonces podemos construir una MTND M (de 2 cintas) que decida a L de la siguiente manera:

- M comienza ubicando la entrada w en ambas cintas y las cabezas lectoras en los extremos derechos de ambas entradas.
- M simula el funcionamiento de M_1 y M_2 en paralelo, usando la 1ra. y 2da. cinta respectivamente
- Dado que M_1 o M_2 paran con w (pero no ambas), M eventualmente alcanza una configuración en la cual la versión simulada de M_1 o de M_2 está por parar.
- Cuando esto sucede, M determina cuál de las MT está por parar y M se detiene con las cintas indicando si $w \in L$ o si $w \notin L$.

Lenguajes Recursivos y Recursivos Enumerables

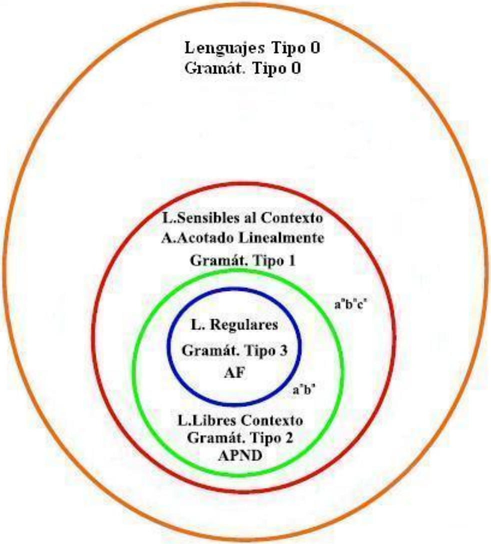
Por lo tanto,

Teorema

Un lenguaje L es recursivo sssi L y su complemento $\overline{L}$ son recursivos enumerables.

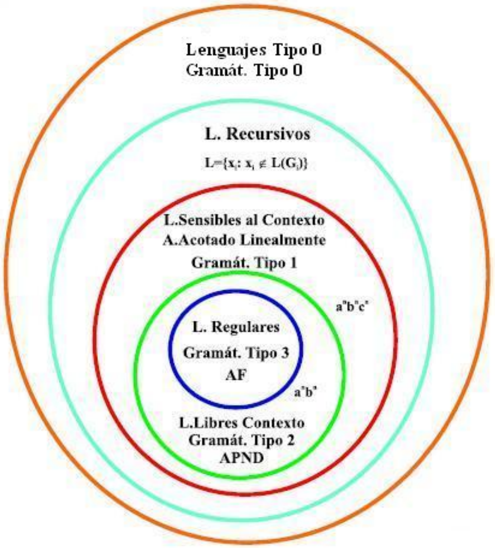
Jerarquía de los Lenguajes Formales

A partir de la Jerarquía de Chomsky:



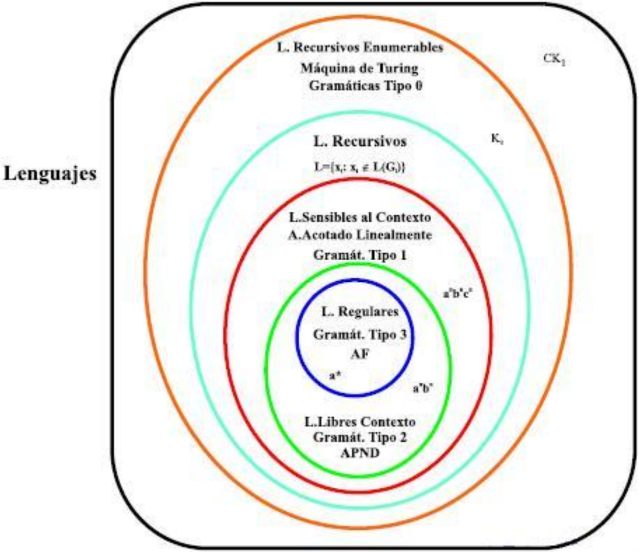
Jerarquía de los Lenguajes Formales

Al final de '*Teoría de la Computación I*' teníamos que:



La Gran Foto Final...

Lenguajes Formales, Máquinas y Jerarquía de Chomsky



¿Preguntas?