

# | Entrega 2 MAC Daniel Chico Valderas

## | Índice

- [Ej\\_24](#)
  - [Apartado A](#)
    - [Solución](#)
  - [Apartado B](#)
    - [Solución](#)
- [Ej\\_25](#)

## | Ej 24

24. Indica cuáles de los siguientes problemas son decidibles, semidecidibles o no semidecidibles:

- (a) Dada una MT determinar si todos sus estados son accesibles (se puede llegar a ellos para un cálculo para alguna entrada).
- (b) Dada una MT determinar si su número de estados es menor o igual a 5.

## | Apartado A

Para poder determinar si el problema es decible, vamos a intentar diseñar un algoritmo y estimar su eficiencia

## | Solución

No identifico ninguna propiedad trivial por lo que no se puede asegurar que es un problema no decible. Idearemos un algoritmo y estudiaremos sus propiedades.

Todas las máquinas de Turing por conveniencia tienen una cantidad de estados finita.

### 🔗 Caso base → Lenguajes finitos

Si el tamaño de la palabra que acepta la máquina de Turing es finita, es posible enumerar y desarrollar todos los procesos de cálculo para cada entrada distinta. Esto da un número finito de posibles cálculos que luego se puede ver por que estados pasa cada proceso de cálculo y viendo cual de todos los estados no se visita. Se acepta si se visitan todos, y se rechaza en caso contrario

### 🔗 Caso General → Lenguajes infinitos

Si la cantidad de palabras puede ser infinita, se puede dar el caso de que se visiten todos los estados de la máquina de Turing en un momento dado (Aceptación de la MT), aunque queden palabras por explorar, en este caso la máquina acepta el problema, pero esta puede ciclar infinitamente ya que en un momento dado, no se

puede asegurar que no se vaya a visitar un estado no visitado en el futuro. (la máquina no es capaz de rechazar una entrada)

Viendo que el algoritmo general acepta cualquier máquina de Turing pasado un número de comprobaciones finita, el algoritmo es decidible.

## | Apartado B

Para poder determinar si el problema es decible, vamos a intentar diseñar un algoritmo y estimar su eficiencia

## | Solución

No identifico ninguna propiedad trivial por lo que no se puede asegurar que es un problema no decidible. Idearemos un algoritmo y estudiaremos sus propiedades.

### 🔗 Caso base → Lenguajes finitos

Si el tamaño de la palabra que acepta la máquina de Turing es finita, es posible enumerar y desarrollar todos los procesos de cálculo para cada entrada distinta. Esto da un número finito de posibles cálculos que luego se puede ver por que estados pasa cada proceso de cálculo y se cuentan cuantos estados distintos aparecen, si son mas de 5 rechaza, si son 5 o menos acepta

### 🔗 Caso General → Lenguajes infinitos

Si la cantidad de palabras puede ser infinita, se puede dar el caso de que se visite un estado número 6 en la máquina de Turing en un momento dado (rechazo de la MT), para esa máquina de turing se acaba el problema, pero esta puede ciclar infinitamente ya que mientras no se superen los 5 estados distintos visitados, no se puede asegurar que no se vaya a visitar un nuevo estado no visitado en el futuro. (la máquina no es capaz aceptar una entrada)

Viendo que el algoritmo general cicla en determinados casos, hace que sea un problema semidecidible

## | Ej 25

25. Demuestra que el Problema de la Correspondencia de Post con un alfabeto  $A$  que tiene un sólo elemento es decidible.

El **Problema de Correspondencia de Post** es un problema de decisión indecidible propuesto por **Emil Post**. Dada una instancia con dos listas finitas de palabras, **X** e **Y**, sobre un alfabeto dado, el objetivo es determinar si existe una secuencia de índices

tal que las palabras indicadas por estos índices en **X** e **Y** sean iguales. Si el alfabeto de las palabras sobre las que trabaja el problema, es un alfabeto de un solo elemento, por ejemplo  $A = \{0\}$ , todas las posibles palabras formadas por los distintos lenguajes sobre el alfabeto  $A$  tendrán la forma  $L^i = \{0^i \mid i \in \mathbb{N}\}$ .

En estas condiciones, para comprobar si dos palabras son iguales, si sus caracteres son siempre el mismo en ambos casos, este problema se reduce a contar que el número de caracteres para ambas palabras es el mismo. En caso de que el lenguaje sea infinito, se recorren ambas palabras a la vez, dígito por dígito y en el momento que una de las palabras acabe y la otra no rechaza, si las dos acaban con el mismo número de caracteres, acepta la palabra.