

# Ejercicio 11 Relación 5

Alejandro Manzanares Lemus

Supongamos que una entrada es una palabra de paréntesis. Demostrar que determinar si están emparejados y anidados correctamente está en **L**. Lo están  $((()))$  y  $((()))$ , pero no lo están  $)() ($  ni  $((())) ($ .

Este problema se puede resolver con una MT determinista con dos cintas:

- 1º cinta: Entrada. Solo leer en esta cinta.
- 2º cinta: Al principio vacía, la usaremos de contador, que empezará a 0.

Cada vez que encontremos '(' en la cinta sumamos 1 a la segunda. Cada vez que encontremos ')' restamos 1 a la cinta, pero antes de restar, comprobamos si es 0. Si es 0 y vamos a restar 1, no están emparejados y por tanto, rechazamos. Si llegamos al final de la 1º cinta y la 2º cinta contiene un número distinto a 0 rechazamos, si es igual a 0 aceptamos.

## 1 Máquina de Turing

```
Mientras el simbolo de la primera cinta != #
  Si el simbolo de la primera cinta = '('
    Sumamos 1 en la segunda cinta
  Si no
    Si la segunda cinta = 0
      Rechazamos
    Si no
      Restamos 1 en la segunda cinta
  Si la segunda cinta = 0
    Aceptamos
  Si no
    Rechazamos
```

Como solo escribimos en la 2º cinta, solo tendremos en cuenta esto. Nos damos cuenta que a lo sumo vamos a contar cuantos símbolos tiene la entrada (el número más grande que escribimos es el número de '(', luego para la entrada '((((' estamos contando su longitud) y sabemos que para escribir un número  $x$  necesitamos  $\log(x)$  espacio, en el caso de usar binario  $\log_2(x)$ , luego el problema está en **L**.