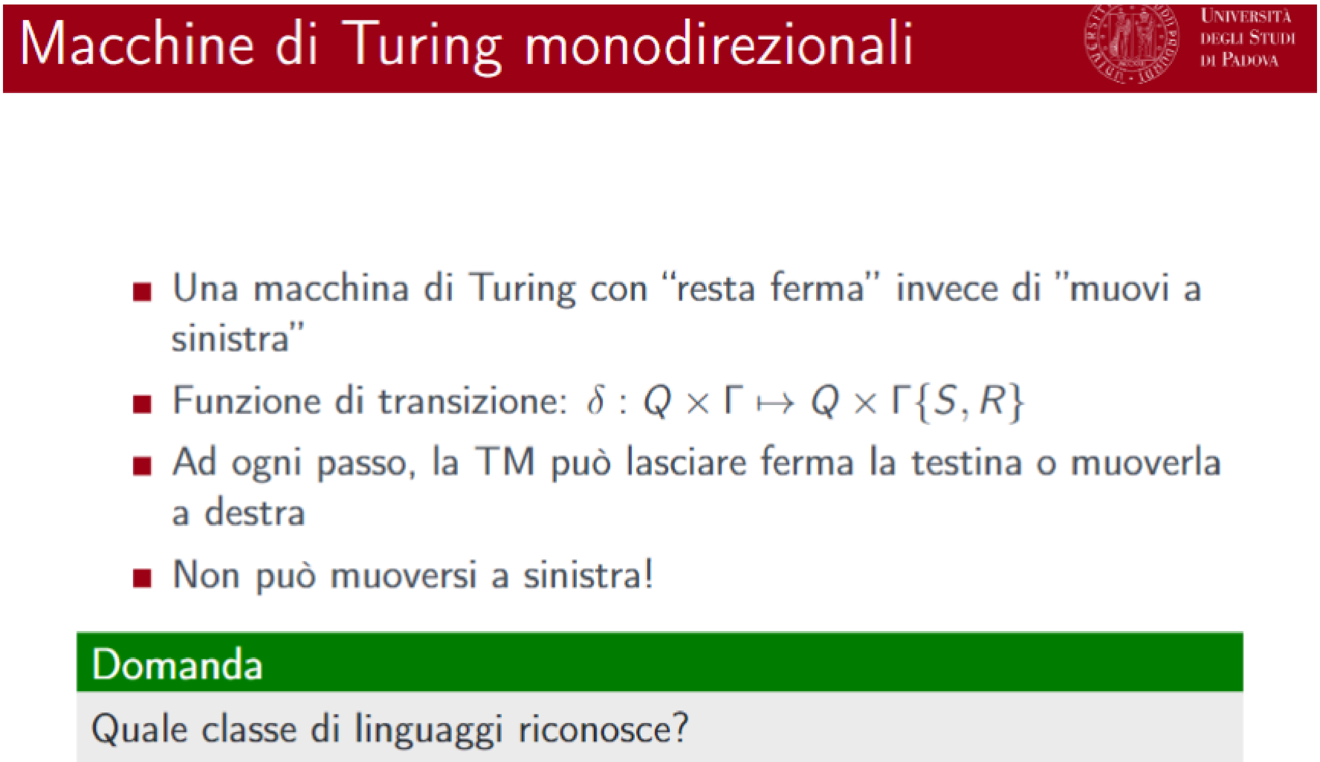

Una **Turing machine monodirezionale** (TM-R/S) è una variante di macchina di Turing che, a ogni passo, può:

- scrivere un simbolo;
- **restare** sulla stessa cella (S) oppure **muoversi a destra** (R);
- **non** può mai muoversi a sinistra.

Formalmente: $\delta : Q \times \mathit{\Gamma} \longrightarrow Q \times \mathit{\Gamma} \times S, R. \delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{S, R\}$.

Teorema

La classe di linguaggi riconosciuti da una TM-R/S coincide con la classe dei **linguaggi regolari**.

Dimostrazione (succinta)

1. Ogni TM-R/S è simulabile da un DFA

Poiché la testina non torna mai su una cella già lasciata, l'unica memoria persistente dell'elaborazione è lo **stato** della macchina e il **simbolo attualmente sotto la testina**. Costruiamo un DFA i cui stati sono coppie $(q, a)(q, a)$ con $q \in Qq \in Q$ e $a \in \mathit{\Gamma} a \in \Gamma$:

- alla lettura di un simbolo xx dall'input il DFA replica la transizione

$\delta(q, a) = (q\prime, b, m)\delta(q, a) = (q', b, m)$;

- se $m = Sm = S$ il DFA rimane sul medesimo simbolo di input (può essere implementato suddividendo la mossa in due step, ma questo non altera la finitezza); $sem = Rm = R$ avanza di un simbolo.

Il numero di stati è finito ($| \ Q \ | \cdot | \ \mathit{\Gamma} \ | |Q| \cdot |\Gamma|$), quindi il linguaggio riconosciuto è regolare.

2. Ogni DFA è trivialmente simulabile da una TM-R/S

Basta ignorare la possibilità di «scrivi» e «resta»: la TM legge l'input da sinistra a destra, aggiornando il proprio stato come farebbe il DFA, e accetta negli stessi casi.

Conclusione

Le due inclusioni mostrano l'equivalenza:
$L(TM\text{-}R/S) = L(DFA) = REG.\mathcal{L}(\text{TM-R/S}) = \mathcal{L}(\text{DFA}) = \mathbf{REG}$.