

# 1 Parte Prima

10 Points

- Si consideri l'espressione regolare  $1\Sigma^* = 1(0\cup 1)^*$ . Convertire tale espressione regolare in una grammatica acontestuale. Dimostrare la correttezza.
- Enunciare e dimostrare il *pumping lemma* per linguaggi regolari. Fornire un esempio di utilizzo.

$$G = (V, \Sigma, R, S)$$

$$G: S \rightarrow 1A$$

$$A \rightarrow 0A \mid 1A \mid \epsilon$$

$$A \Rightarrow^* w \Rightarrow w \in \Sigma^*$$

DIMOSTRARE PER INDUZIONE LA CORRETTEZZA

Caso Base

$n=0$

$$w = \epsilon \Rightarrow w \in \Sigma^* \quad A \rightarrow \epsilon$$

IPOTESI INDUCTIVA

$n=m$

$$|w| = m$$

$$A \Rightarrow^* w \text{ con } |w|$$

PASSO INDUCTIVO

$n = m+1$

- caso  $w[1] = 0$  per HA A che ridotta a A o 1A in  $|w-1|=m$

- caso  $w[1] = 1$  poi HA A che ridotta a A o 1A in  $|w-1|=m$

$$\Rightarrow w \in \Sigma^* \Rightarrow A \Rightarrow w^*$$

## 2 Parte Seconda

10 Points

- Uno stato di una macchina di Turing è detto inutile se la macchina non entra mai in quello stato per ogni possibile input. Considerare il problema di determinare se una macchina di Turing ha uno stato inutile. Dimostrare che il linguaggio associato a questo problema non è decidibile.

$$WUTIL_1 = \{ \langle M \rangle \mid M \in TM \text{ e } \exists \text{ stato inutile } \in M \}$$

Vogliamo dimostrare la non decidibilità

$$A_{TM} \leq_m WUTIL_1$$

$$f: \Sigma^* \rightarrow \Sigma^*$$

$$\forall \langle M, w \rangle \in A_{TM} \Leftrightarrow f(\langle M, w \rangle) \in WUTIL_1$$

Poiché  $f$  deve essere calcolabile es. da una TM

Prendiamo in input  $\langle M, w \rangle$

Costruiamo una TM  $M'$  con input  $x$  che opera:

con 3 stati ( $q_{start}$ ,  $q_{acc}$ ,  $q_{rej}$ )

- Se  $x = w$  accetta

- Se  $x \neq w$  simula  $M(w)$

Se accetta  $M'$  accetta automaticamente accetta

output  $\langle M' \rangle$

Tuttavia i 3 stati sono utili

$$\Rightarrow \langle M, w \rangle \in A_{TM} \Leftrightarrow w \in L(M) \Leftrightarrow \exists \text{ stato utile} \\ \Leftrightarrow f(\langle M, w \rangle) \in WUTIL_1 \\ = M'$$

□

2° DM (ALTERNATIVA)

$$f: \Sigma^k \rightarrow \Sigma^k$$

$$\forall \langle M, w \rangle \in A_{TM} \Leftrightarrow f(\langle M, w \rangle) \in \text{NOTI}$$

Perché  $f$  deve essere calcolabile essendo una TM

Prendi un input  $\langle M, w \rangle$

crea un TM  $M'$  con 3 stati ( $q_{start}$ ,  $q_{acc}$ ,  $q_{rej}$ )  
con input  $x$

- Se  $x = w$   $M'$  accetta se  $M(w)$  accetta

- Se  $x \neq w$   $M'$  accetta

altrimenti  $\langle M' \rangle$

$\Rightarrow$

$$\langle M, w \rangle \in A_{TM} \Leftrightarrow w \in L(M) \Rightarrow M(w) \text{ accetta}$$

$$\Rightarrow M' \text{ accetta} \Rightarrow q_{acc} \text{ notie} \Rightarrow \langle M' \rangle \in \text{NOTI}$$

$$\Leftarrow) \langle M, w \rangle \notin A_{TM} \Leftrightarrow w \notin L(M) \Rightarrow M(w) \text{ no accetta}$$

$$\begin{array}{l} \text{Se } x = w \text{ usa } q_{rej} \Rightarrow \langle M' \rangle \notin \text{NOTI} \\ \text{Se } x \neq w \text{ accetta} \end{array}$$

### 3 Parte Terza

10 Points

- Una macchina di Turing fortemente non-deterministica ha tre possibili stati finali: *accept*, *reject*, e *notsure*. Tale macchina decide un linguaggio  $L$  come segue: Se  $x \in L$ , tutti i possibili rami di computazione con input  $x$  portano ad uno stato *accept* oppure *notsure*, con almeno un ramo che termina in *accept*. D'altra parte, se  $x \notin L$ , tutti i possibili rami di computazione con input  $x$  portano ad uno stato *reject* oppure *notsure*, con almeno un ramo che termina in *reject*. Dimostrare che  $L$  è decidibile in tempo polinomiale da una macchina di Turing fortemente non-deterministica se e solo se  $L \in NP \cap coNP$ .