1 Automi **10 Points**

- Considerare la grammatica G così definita:

$$S \to WbT$$
$$T \to aWbT | bVaT | \varepsilon$$
$$W \to aWbW | \varepsilon$$
$$V \to bVaV | \varepsilon.$$

Mostrare che *aabbbbaab* appartiene ad $L(G)$. Descrivere un PDA equivalente alla grammatica G.

- Enunciare e dimostrare il *pumping lemma* per linguaggi regolari. Fornire un esempio di utilizzo.

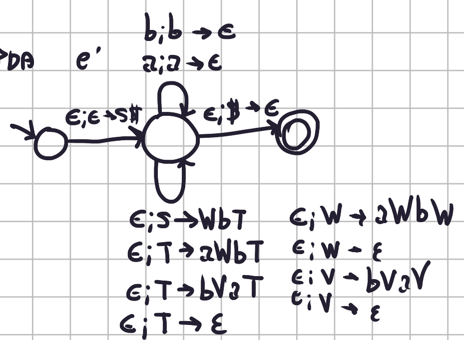
Mostrerò la successione di derivazioni in G

$$S \to WbT \xrightarrow{W \to aWbW} aWbWbT \xrightarrow{W \to aWbW} aaWbWbWbT \xrightarrow{W \to \varepsilon} aabWbWbT \xrightarrow{T \to bVaT} aabWbWbbVaT \xrightarrow{T \to aWbT} aabWbWbbVaaWbT$$

$$\downarrow \begin{matrix} W \to \varepsilon \\ V \to \varepsilon \\ T \to \varepsilon \end{matrix}$$

aabbbbaab ✓

Il PDA e'



Pumping Lemma: Se L e' REG e $w \in L$, $\exists\ p$ t.c. $p \leq |w|$ e $w = xyz$ t.c.

- $|xy| \leq p$
- $|y| \neq 0$
- $xy^iz \in L \quad \forall i$

Dimo: Sia $w \in L$ generata da un DFA $D = (Q, \Sigma, \delta, q_0, F)$, sia $p = |Q|$ e sia $|w| = n \geq p$. $w = w_1 \dots w_n$. Sia $r_1, r_2 \cdots, r_{n+1}$ la successione di stati che D computa su w. $\forall i \quad \delta(r_i, w_i) = r_{i+1}$

Oss: essendo $n+1 > p = |Q|$, nei primi $p+1$ stati, uno si ripete:

$$r_1, r_2 \cdots r_j, \cdots r_l, \cdots r_{p+1} \quad \text{e} \quad r_j = r_l$$

$\Rightarrow$ Suddivido w come segue $w = xyz$

$x = w_1 \cdots w_{j-1}$ essendo $l \leq p+1 \Rightarrow l-1 \leq p$ e $|xy| = l-1$
$y = w_j \cdots w_{l-1}$
$z = w_l \cdots w_n$ essendo $l-1 \geq j \Rightarrow |y| = l-1-j-1 \geq 1$

essendo che

- x porta da r_1 a r_j
- y da r_j a $r_l = r_j$
- z da r_l a r_n

$\Rightarrow y^i$ porta da r_j a $r_l \Rightarrow xy^iz \in L$ ∎

Esempio: $L = \{0^n 1^n \mid n \geq 1\}$

$w = 0P0P1^{2P}$
$|xy| \leq p \Rightarrow y$ contiene solo zeri $\Rightarrow xy^iz$ ha piu' 1 che 0 $\Rightarrow$ non e' in L.

2 Calcolabilità

10 Points

- Sia $NONREGULAR_{TM}$ il linguaggio contenente le stringhe $\langle M\rangle$ che costituiscono codifiche valide di macchine di Turing tali che $L(M)$ non è regolare. Mostrare che $A_{TM} \leq_m NONREGULAR_{TM}$.
- Dimostrare che esistono linguaggi che non sono decidibile e neppure Turing-riconoscibili.

Definisco R t.c.

- Su input $\langle M,w\rangle$
- Definisce M' che
  - Su input x
  - Se $M(w)$ accetta, accetta x se $x = 0^n 1^n 2^n$
  - Se $M(w)$ rifiuta, accetta x
- L'output e' M'

$$\langle M,w\rangle \in A_{TM} \Rightarrow L(M') = \{0^n 1^n 2^n \mid n \in \mathbb{N}\} \Rightarrow \langle M'\rangle \in NONREGULAR_{TM}$$

$$\langle M,w\rangle \notin A_{TM} \Rightarrow \left\{\begin{array}{l} M(w) \text{ rifiuta} \Rightarrow L(M') = \Sigma^* \\ M(w) = \infty \Rightarrow L(M') = \emptyset \end{array}\right\} \Rightarrow \langle M'\rangle \notin NONREGULAR_{TM}$$

Se L e' Decid. e' anche Ric. So che A_{TM} non e' decidibile perche':

H decide A_{TM}, Definisco $D(\langle M\rangle) = 1 - H(\langle M,\langle M\rangle\rangle) \Rightarrow D(\langle D\rangle) = 1 - H(\langle D,\langle D\rangle\rangle) = 1 - D(\langle D\rangle) \Rightarrow 1 = 0$

Quindi dimostro che $\overline{A_{TM}}$ non e' riconoscibile

Sia U t.c. riconosce A_{TM}

Per assurdo U' riconosce $\overline{A_{TM}}$

Definisco H che su input $\langle M,w\rangle$

- esegue $U(\langle M,w\rangle)$ e $U'(\langle M,w\rangle)$
- per ipotesi una delle due accetta
- Se U accetta D accetta, Se U' accetta D rifiuta

$\Rightarrow$ H decide $A_{TM} \Rightarrow$ IMPOSSIBILE. ■

3 **Complessità** 10 Points

- Un cammino in un grafo non orientato è detto semplice se non contiene vertici ripetuti. Sia *LONGEST-PATH* = $\{\langle G, s, t, k\rangle : G = (V, E)$ grafo non orientato, $s, t \in V$, esiste un cammino semplice $s \rightsquigarrow t$ di lunghezza $\geq k\}$. Mostrare che *LONGEST-PATH* $\in$ *NP*.
- Enunciare e dimostrare il teorema di gerarchia di spazio.

Definisco un verificatore V che

- Su input $\langle G, s, t, k\rangle$ $\langle y\rangle$
- Interpreta il certificato $\langle y\rangle$ come la successione di archi di un cammino
- $\langle y\rangle = (s, v_1)(v_1, v_2)\dots(v_k, t)$
- Controlla se gli archi in $\langle y\rangle$ sono presenti in $E(G)$ $O(n)$
- Conta il numero di archi, e verifica che sono $\geq k$ $O(n)$
- Verifica che in $\langle y\rangle$ i nodi NON sono ripetuti $O(n^2)$
- Se i 3 controlli vanno a buon fine V accetta, altrimenti rifiuta

$\Rightarrow$ V e' un verificatore polinomiale.

Teo: Siano $S_1(n)$ e $S_2(n)$ tali che $S_2(n) = \Omega(S_1(n))$, allora

$$\exists L \in SPACE(S_2(n)) \setminus SPACE(S_1(n))$$

Dim: Sia $[\cdot]_{TM}$ una codifica per le TM, definisco D t.c.

- Su input x considera $[x]_{TM} = M$
- esegue $M(x)$
- Se $M(x)$ termina usando $O(S_1(n))$ celle, $D(x) = \overline{M(x)}$
- Se $M(x)$ prova a usare piu' di $O(S_2(n))$ termina

chiaramente $L(D) \in SPACE(S_2(n))$. FATTO : $L(D) \notin SPACE(S_1(n))$

Pongo per assurdo che Q decide $L(D)$ in spazio $S_1(n)$

Sia x_Q t.c. $[x_Q]_{TM} = Q$

- Eseguo $D(x_Q)$
- Esegue $Q(x_Q)$
- termina usando (per ipotesi) $O(S_1(n))$ celle
- Quindi $D(x_Q) \neq Q(x_Q) \Rightarrow$ contraddizione. ■