1 Parte Prima — 10 Points

- Dimostrare che il seguente linguaggio su alfabeto $\{\#, 1\}$ non è regolare:

$$\{1^i\#1^j\#1^{i+j} : i, j \geq 1\}.$$

- Dimostrare che se un linguaggio è regolare allora esiste una espressione regolare che lo descrive.

Bisogna verificare che le condizioni del pumping lemma non sono soddisfatte.

$\omega = 1^{p-k}\#1^k\#1^p$ la stringa e' lunga $2p$, p e' il pumping. $\omega = xyz$

- $|xy| \leq p \Rightarrow$ la cardinalita' di y deve essere minore o uguale a p

CASO 1: $y = 1^{p-k}\#1^k \Rightarrow xy^0z = \#1^p \notin L$

CASO 2: $y = 1^{p-k} \Rightarrow xy^0z = \#1^k\#1^p \notin L$

CASO 3: $y = 1^k \Rightarrow xy^0z = 1^{p-k}\#\#1^p \notin L$

CASO 4: $y = 1^p \Rightarrow xy^0z = 1^{p-k}\#1^k\#\# \notin L$

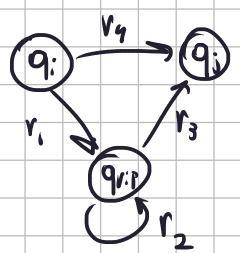
Sia L regolare, e $L(D) = L$, si definsce un GNFA che ha sugli archi delle espressioni regolari, ha 1 solo stato accettante e se non esiste u per cui $\delta(q_i, u) = q_j$ in D, nel GNFA $\delta(q_i, \emptyset) = q_j$.

- ogni stato avra' un arco verso ogni altro stato (compreso se stesso)

Convert (G):

- Se ci sono 2 nodi, ritorna la regex sull'arco
- Altrimenti seleziona $q_{rip} \in Q \setminus \{q_{acc}, q_{start}\}$ e definisci G':
  - G' non ha q_{rip}, ha funzione δ'
  - $\forall q_i, q_j \quad \delta'(q_i, q_j) = (r_1 r_2^* r_3) | r_4$ se

$\delta(q_i, q_j) = r_4 \quad \delta(q_i, q_{rip}) = r_1 \quad \delta(q_{rip}, q_{rip}) = r_2 \quad \delta(q_{rip}, q_j) = r_3$



- Return G'

Convert(G) e' equivalente a G perche' l'aggiornamento delle etichette considera per ogni q_i, q_j, le stringhe per passare da uno all'altro passando per q_{rip}. ∎

~ Sia $DECIDABLE_{TM}$ il linguaggio che consiste di tutte le stringhe $\langle M \rangle$ tali che M è una macchina di Turing ed $L(M)$ è decidibile. Mostrare che $DECIDABLE_{TM}$ è indecidibile.

- Dimostrare che per ogni macchina di Turing non-deterministica ne esiste una deterministica equivalente.

Mostro che $A_{TM} \leq_m DEC_{TM}$ ossia $\langle M,w \rangle \in A_{TM} \iff R(\langle M,w \rangle) = M' \in DEC_{TM}$. Definisco R:

- Su input $\langle M,w \rangle$
- Definisce M' che, su input $\langle M'',x \rangle$
  - Se $M(w)$ accetta, M' rifiuta sempre $\Rightarrow L(M') = \emptyset$ ed e' decidibile
  - Altrimenti accetta se M'' accetta $x \Rightarrow L(M') = A_{TM}$
- L'output e' M'

Una TM M puo' emulare una NTM N esplorando i rami di computazione in ampiezza, usando piu' nastri

- un nastro per mantenere l'input intatto
- un nastro per eseguire il calcolo
- un nastro per memorizzare l'indice del ramo corrente

In particolare, puo' esplorare un ramo, eseguire un passo di computazione, per poi passare a quello successivo.

- Se almeno 1 ramo e' accettante, M si ferma e accetta
- Se almeno 1 ramo rifiuta, M si ferma e rifiuta
- Se ogni ramo va in loop, M va in loop

Chiaramente $L(M) = L(N)$

3 Parte Terza 10 Points

- ~~Dimostrare che la classe *NL* è chiusa rispetto alle operazioni di unione, intersezione, e star di Kleene.~~
- Definire la classe di complessità *coNP* ed il problema *UNSAT*. Dimostrare che *UNSAT* è *coNP*-completo.

$coNP = \{ L \text{ t.c. } \overline{L} \in NP \}$. $\underset{NP}{\overset{A}{\cap}} \leq^p_m SAT \Rightarrow \underset{coNP}{\overset{\overline{A}}{\cap}} \leq^p_m \overline{SAT} \Rightarrow$ ogni $\overline{A} \in coNP$ si riduce a $\overline{SAT} = UNSAT$.

$SAT \in NP \Rightarrow UNSAT \in coNP$

$NL = Ntime(\log(n))$

Se $A \in NL$ e $B \in NL$, M_A decide A in spazio $\log(n)$ e M_B decide B in spazio $\log(n)$.

Definisco M tale che, esegue sia M_A che M_B su 2 nastri, e

- Accetta se entrambe accettano (Inters.)
- Accetta se una accetta (Unione)

Chiaramente M usa spazio $2 \cdot \log(n) = O(\log(n))$, quindi $A \cup B \in NL$ e $A \cap B \in NL$.

Per la star ...