1 Automi 10 Points

- Sia C un linguaggio *context-free* ed R un linguaggio regolare. Mostrare che il linguaggio $C \cap R$ è *context-free*. Usare questo risultato per mostrare che il linguaggio L delle stringhe $w \in \{a,b,c\}^*$ tali che w contiene lo stesso numero di a, b e c non è *context-free*.
- Dimostrare che un linguaggio è acontestuale se e solo se esiste un PDA che lo riconosce.

Teo: PDA $\equiv$ CFG

char input ↓ stack ↓

Sia $G=(V,\Gamma,R,S)$ una CFG, definisco un PDA $P=(Q,\Gamma,(V\cup\Gamma),\delta,q_0,\{q_{acc}\})$

- $Q=\{q_0,q_{acc},q_{loop}\}$
- $\delta(q_0,\epsilon,\epsilon)=(q_{loop},\$S)$
- $\delta(q_{loop},\epsilon,A)=\{(q_{loop},W) \mid \forall A\to W\in R\}$
- $\delta(q_{loop},a,a)=(q_{loop},\epsilon)$ se $a\in\Gamma$
- $\delta(q_{loop},\epsilon,\$)=(q_{acc},\epsilon)$

$\Rightarrow$ P deriva ogni $w\in L(G)$ nello stack e se l'input e' nello stack lo accetta.

una linguaggio di una grammatica può essere accettato da un PDA

Sia $P=(Q,\Sigma,\Gamma,\delta,q_0,F)$ un PDA, assumo (WLOG) che sia in forma canonica

- La δ non sostituisce ma i termini nello stack, o rimuove o aggiunge
- Esiste un unico stato finale q_{acc}
- Quando la comput. termina (accettando) lo stack e' vuoto.

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

- $A_{pq}\in V$ se $p,q\in Q$
- $A_{pp}\to\varepsilon\in R \quad \forall p\in Q$
- $A_{pq}\to A_{pr}A_{rq} \quad \forall p,r,q\in Q$
- $A_{pq}\to aA_{rs}b$ con $p,q,r,s\in Q$ e $a,b\in\Sigma$ se $\delta(p,a,\epsilon)=(r,u)\wedge\delta(s,b,u)=(q,\epsilon)$

CON PILA VUOTA

Lemma: x porta l'automa da p a q <=SSE=> A_{pq} deriva x. Per ipotesi, A_{pq} deriva x

- se lo deriva in 1 passo, allora e' $A_{pp}\to\varepsilon$, e ε porta da p a p CPV (NOTAZIONE) ← CON PILA VUOTA
- si assume sia vero per n-1 passi
- A_{pq} deriva x in n passi, la regola e' del tipo:
  - $A_{pq}\to A_{pr}A_{rq}$, ma A_{pr} porta da p a r CPV e A_{rq} porta da r a q CPV quindi $A_{pr}A_{rq}$ porta da p a q CPV
  - $A_{pq}\to aA_{rs}b \Rightarrow A_{rs}$ porta da r a s CPV e $\delta(p,a,\epsilon)=(r,u)\wedge\delta(s,b,u)=(q,\epsilon)$ quindi $aA_{rs}b$ porta da p a q CPV

2 Calcolabilità 10 Points

- Considerare il linguaggio L delle stringhe $u \in \{a,b\}^*$ tale che u ha un numero di a che è almeno il doppio del numero di b. Descrivere una macchina di Turing che decide L in tempo $O(n^2)$. Descrivere una macchina di Turing multi-nastro che decide L in tempo $O(n)$.
- Dimostrare che il linguaggio $EQ_{TM} = \{\langle M_1, M_2\rangle : L(M_1) = L(M_2)\}$, dove M_1 ed M_2 sono macchine di Turing, è indecidibile.

$L = \{ w \mid \#_a(w) \geq 2\#_b(w) \}$

M ha la procedura seguente
- legge il nastro
- crea 2 contatori alla fine del nastro
- ogni volta che legge a o b, va ai contatori ed aggiorna
- al termine verifica i contatori

$\Rightarrow$ per ogni cella letta (sono n) scorre tutto il nastro (n iter.) per aggiornare i contatori. M ha costo $O(n^2)$.

M' (multinastro) ha la seguente procedura
- Sul nastro 1 ha l'input
- Sul nastro 2 ha il contatore di a
- Sul nastro 3 ha il contatore di b
- legge ogni cella (n iter.)
- Se legge a o b aggiorna il contatore corrispondente sul suo nastro $O(1)$

$\Rightarrow$ M' opera in $O(n)$

Mostro che $E_{TM} \leq_m EQ_{TM}$, definisco R che:
- Su input $\langle M \rangle$
- Definisce M' tale che:
  - Su ogni input x, rifiuta // $L(M') = \emptyset$
- l'output e' $\langle M, M' \rangle$

$\langle M \rangle \in E_{TM} \Rightarrow L(M) = \emptyset \Rightarrow \langle M, M' \rangle \in EQ_{TM}$

$\langle M \rangle \notin E_{TM} \Rightarrow L(M) \neq \emptyset \Rightarrow \langle M, M' \rangle \notin EQ_{TM}$

3 Complessità 10 Points

- Si supponga l'esistenza di una funzione iniettiva f che mappa interi ad n-bit in interi ad n-bit, tale che: (i) $f(x)$ è calcolabile in tempo polinomiale; (ii) $f^{-1}(x)$ non è calcolabile in tempo polinomiale. Mostrare che il linguaggio delle stringhe $x, y \in \{0,1\}^n$ tali che $f^{-1}(x) < y$ è contenuto in $(NP \cap coNP) \setminus P$.
- Definire il concetto di riduzione. Dimostrare che 3-$COL \leq_m^p SAT$, dove 3-COL è il linguaggio dei grafi 3-colorabili.

$L = \{\langle x, y\rangle \mid f^{-1}(x) < y\}$

- La NTM in questione considera non deterministicamente ogni possibile intero K composto da n bit ($O(n)$)
- Calcola $f(K)$
- if ($f(K) = x$) : Accetta se e solo se $K < y$
- if ($f(K) \neq x$) : Va volontariamente in loop

Essendo f iniettiva, K lo trova sempre, la TM decide L.

$\bar{L} = \{\langle x, y\rangle \mid f^{-1}(x) \geq y\}$

- La NTM in questione considera non deterministicamente ogni possibile intero K composto da n bit ($O(n)$)
- Calcola $f(K)$
- if ($f(K) = x$) : Accetta se e solo se $K \geq y$
- if ($f(K) \neq x$) : Va volontariamente in loop

Essendo f iniettiva, K lo trova sempre, la TM decide $\bar{L}$. $\Rightarrow L \in NP \cap coNP$

L non e' in P perche'
- Bisogna necessariamente calcolare $f^{-1}(x)$
- Non si puo' fare in tempo polinomiale

Def : $A \leq_m^p B$ se $\exists R$ t.c. $w \in A \Leftrightarrow R(w) \in B$ e R ha costo polinomiale.

3-COL si riduce a SAT con $R(\langle G, s, t\rangle)$:

- Considera V vertici di G
- Definisce ϕ e $\forall x_i \in V(G)$ si hanno x_i, x_i' variabili in ϕ
- (x_i, x_i') codificano un colore di x
- $\phi_{3col} = \bigwedge_{1 \leq i \leq n} \overline{x_i \wedge x_i'}$ non possono codificare il colore $(1,1)$
- Se $(i,j) \in E(G)$, esiste $\phi_{i,j} = \overline{(x_i \leftrightarrow x_j) \wedge (x_i' \leftrightarrow x_j')}$
- $\phi = \phi_{3col} \wedge \left(\bigwedge_{(i,j) \in E} \phi_{i,j}\right)$
- l'output di R e' $\phi \Rightarrow \phi \in SAT \Leftrightarrow G$ e' 3-colorabile.