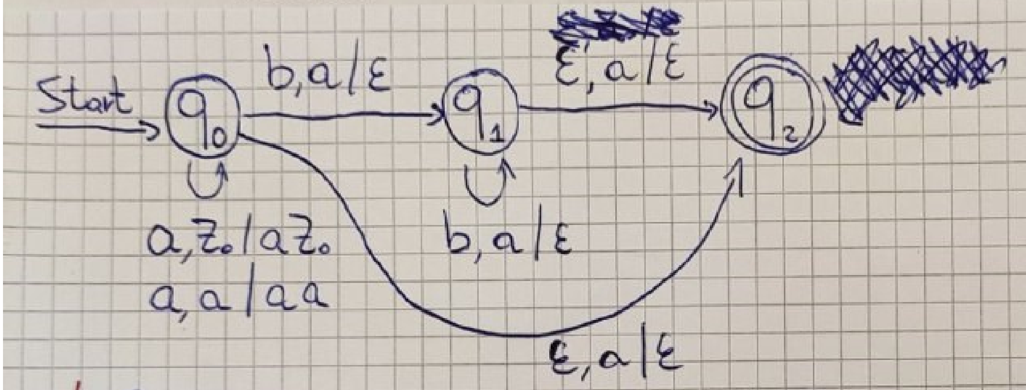
AUTOMI E LINGUAGGI FORMALI
II COMPITINO DEL 7 GIUGNO 2019

Tempo a disposizione: 1 h 30 m

1. Scrivere un DPDA che riconosca il linguaggio, $\{a^n b^m \mid 0 \leq m \leq n, n > 0\}$. Cercate di spiegare a parole perché il vostro DPDA funziona. Come accetta il vostro DPDA? Perché avete scelto quel metodo di accettazione?



✓ a
✓ aabb
✓ aaaabb
✗ aabba
✓ aaa
✗ bb

Es. 2

La differenza tra i due PL sta nel fatto che nel PL per CF devi considerare che il "pezzo" che pompi possa essere preceduto da u. Quindi devi valutare che vwx, a differenza di xy (del PL dei regolari) possa iniziare in qualunque punto della stringa.

z=0^n1^n non è reg perché pompi solo zeri e non 1. Non puoi dire non sia CF perché se prendi vwx = 0^p1^p, e pompi vwx rimani nel linguaggio. In altri termini puoi scegliere z=0^f0^p1^p1^f dove u=0^f vwx=0^p1^p y=1^f

Es. 3

Dimostrare che il linguaggio $\{a^n b^m a^n | m \geq n\}$ non è CF.

Se un linguaggio non è context-free allora non è regolare e, per non essere regolare, bisogna applicare il solito pumping lemma. Si può subito notare come il numero di a sia almeno "2n" con il numero di b che è "m". Tuttavia la proprietà asserisce che m >=n e dunque dovremmo avere, per le solite condizioni:
$y \neq 0$, $xy <= p$ e $xy^iz \in L$
una parola $w=a^{2k}b^m$ tale che sia $w=a^{2k}b^{p-m-2k}$. Sapendo che m > 2k, necessariamente la parola contiene più a che b nella sottostringa "xy" e dunque il linguaggio non è regolare e dunque non potrà essere CF.

Es. 4

Il Teorema di Rice afferma che qualsiasi proprietà non banale dei linguaggi ricorsivamente enumerabili (RE) è indecidibile. Una proprietà è non banale se esiste almeno un insieme ricorsivamente enumerabile che la soddisfa e almeno un altro che non la soddisfa.

Supponiamo che esista una proprietà P dei linguaggi ricorsivamente enumerabili tale che l'insieme degli insiemi ricorsivamente enumerabili che soddisfano P sia ricorsivo. Consideriamo l'insieme S di tutte le macchine di Turing che accettano un linguaggio L tale che L soddisfi P. Sia T l'insieme di tutte le macchine di Turing che accettano un linguaggio diverso da L e che non soddisfano P.

Ora, consideriamo l'insieme S' di tutte le macchine di Turing che appartengono all'insieme S e che accettano il linguaggio vuoto. In altre parole, S' è l'insieme di tutte le macchine di Turing che soddisfano P e che accettano il linguaggio vuoto. Siccome l'insieme vuoto non soddisfa la proprietà P, allora S' è un insieme vuoto.

Supponiamo per assurdo che l'insieme degli insiemi ricorsivamente enumerabili che soddisfano P sia ricorsivo. Allora, possiamo decidere se una macchina di Turing appartiene all'insieme S' o meno. Se l'insieme S' è vuoto, allora qualsiasi macchina di Turing che accetta il linguaggio vuoto non soddisfa la proprietà P e appartiene all'insieme T. Quindi, possiamo decidere se una macchina di Turing appartiene all'insieme T o meno.

Ma questo ci porta ad una contraddizione, perché se fosse possibile decidere se una macchina di Turing appartiene all'insieme S' o all'insieme T, allora saremmo in grado di decidere se una macchina di Turing accetta il linguaggio vuoto o un linguaggio diverso dal vuoto, il che è noto essere indecidibile per il teorema di Rice.

Pertanto, la supposizione iniziale che l'insieme degli insiemi ricorsivamente enumerabili che soddisfano P sia ricorsivo è falsa. Quindi, la proprietà P non è decidibile per il Teorema di Rice.

Es. 5

(1) Essendo SetPartition in altre parole trovare {(S1, S2) | sum S1 = sum S2 }, dati i due insiemi e sommando gli elementi si puo' verificare in tempo polinomiale (O(n)) ch la soluzione è corretta. Certificato (S1, S2). Quindi SetPartition è NP.

(2) Sapendo che SubsetSum è NP-Completo, posso usare SetPartitioning nel seguente modo per risolvere SubsetSum, costruendo dall'input (S, t) un insieme S1 = S U {s+t, 2s-t} dove "s" è la somma degli elementi in S.

Applicando SetParitioning ottengo dunque che

1. se c'è una sottolista propria di S, A la cui somma è t, allora S1 sarà partizionato in S1 U {2s - t} e S \ S1 U {s + t}. Infatti la somma della prima partizione sarebbe t + (2s - t) = 2s e della seconda (s - t) + (s + t) = 2s

2. se S1 può essere partizionato allora c'è una sottolista di S la cui somma è t. Infatti, dato che (s + t) + (2s - t) = 3s e ogni parte è 2s, i due elementi appartengono a parti differenti. Se {2s - t} in una delle due partizioni, A, il resto degli elementi di A sommati danno infatti t.