# Algoritmi e Principi dell'Informatica

# Appello del 15 settembre 2017

**Esercizio 1 (8 punti)**

a) Definire l'automa a potenza minima che riconosce il linguaggio

$L = \{wb^*w^R : w \in a^+b^*\}$

b) Analizzare i seguenti linguaggi e fornire una spiegazione esauriente riguardo alla classe di automi necessaria per il loro riconoscimento.

$L' = \{wa^+b^*w^R : w \in a^+b^*\}$

$L'' = \{ww^Rw : w \in a^+b^*\}$

**Esercizio 2 (8 punti)**

Dato un input *x* ed una TM deterministica *y*, si definisce lunghezza finita dell'esecuzione di *y* mediante *x* il numero totale di transizioni (realizzate da *y* leggendo l'input *x*) che portano la macchina *y* dalla configurazione iniziale ad una configurazione di arresto (se la computazione non termina, la lunghezza finita non è definita).

È decidibile stabilire se un generico input *x* induca in una generica TM *y* un'esecuzione di lunghezza finita pari?

**SOLUZIONI:**

**Esercizio 1**

a) Le parole del linguaggio $L = \{wb^*w^R : w \in a^+b^*\}$ sono della forma $a^i b^j b^* b^j a^i$ con $j \geq 0$ e $i>0$ ovvero:

$a^i b^{2j+h} a^i$

con $h \geq 0$. Pertanto, le parole del linguaggio sono della forma $a^i b^* a^i$.

E' sufficiente quindi un PDA per riconoscere L.

b) Le parole del linguaggio $L' = \{wa^+b^*w^R : w \in a^+b^*\}$ sono della forma $a^i b^j a^+b^* b^j a^i$ con $j \geq 0$ e $i>0$. Il linguaggio L' è dunque definibile come $L' = L_1' \cup L_2'$ con:

$L_1' = \{a^i a^+b^* a^i : i>0\}$ quando $j=0$

$L_1'' = \{a^i b^j a^+b^* b^j a^i : i>0, j>0\}$

Entrambi i linguaggi $L_1'$ e $L_1''$, considerati singolarmente, sono riconoscibili da un PDA. Tuttavia la loro unione richiede un NPDA. Informalmente, le parole $a^i a^+b^* a^i$ sono prefisso delle parole in $L_1''$. Per questo motivo, l'automa non può distinguere il suffisso $a^i$ delle parole in $L_1'$, il cui riconoscimento richiede lo svuotamento della pila, dal termine $a^+$ presente nelle parole di $L_1''$, che invece non deve modificare il contenuto di pila.

Le parole del linguaggio $L'' = \{ww^Rw : w \in a^+b^*\}$ sono della forma $a^i b^j b^j a^i a^i b^j$, ovvero:

$a^i b^{2j} a^{2i} b^j$.

In questo caso, il linguaggio richiede una MT.

**Esercizio 2**

Riduzione di HALT a P

1. Se $M_y$ non è completa, costruisco $M'_y$ completa in cui ogni run o termina in accettazione o va in loop. Altrimenti, $M'_y = M_y$

2. Costruisco una $M''_y$ in cui ogni stato q di $M'_y$ è raddoppiato in <q,pari> e <q,dispari> e definisco la funzione di transizione di stato di $M''_y$ in questo modo:

q -[…]→q' in $M'_y$ allora

- <q,pari> -[…]→<q',dispari> e
- <q,dispari> -[…]→<q',pari>.

Inoltre, se q è di accettazione in $M'_y$ allora <q,pari> è di accettazione. Da <q,dispari>, mediante opportune transizioni (che coprono tutte le combinazioni di simboli possibili ottenibili sui nastri), $M''_y$ va in uno stato di accettazione (creato apposta per gestire i run accettanti di lunghezza dispari di $M'_y$ che in $M''_y$ hanno dunque lunghezza pari). Lo stato iniziale di $M''_y$ è <$q_0$,pari>.

Quindi, dato input generico x, $M''_y$ o va in loop oppure accetta con un run di lunghezza pari.

Risolvere P su $M''_y$ equivale a stabilire se $M_y$ termina su x.