TestiGara - QuintaEdizione

Esercizi di Gara della V edizione

AVVISI:

- Se non specificato altrimenti negli esercizi, le sequenze iniziali su nastro si intendono *non vuote*, ovvero contenenti almeno un simbolo.
- Per numero decimale si intende un numero positivo o nullo rappresentato con le cifre 0, 1, 2, ..., 9, senza zeri iniziali non significativi; per esempio 0 e 19 sono numeri validi, mentre 0032 deve essere scritto come 32.
- Nel fornire le soluzioni, ricordarsi di pulire il nastro finale da ogni simbolo che non costituisca la risposta!

Esercizio 1 [Sostituzione di caratteri]. Programmare una macchina di Turing che, dato un nastro iniziale contenente una sequenza arbitraria di simboli A e B, *sostituisca* ogni occorrenza di due simboli consecutivi AB con due simboli CD. Esempi:

NASTRO INIZIALE	*NASTRO FINALE*
AABABBBAABAAABAABAA	ACDCDBBACDAACDACDAA
BBBBAAA	BBBBAAA
AABB	ACDB

Esercizio 2 [Parentesi bilanciate]. Una sequenza di parentesi quadre e graffe annidate si dice *bilanciata* secondo la seguente definizione: (i) la sequenza vuota è bilanciata; (ii) se *S* e *T* sono sequenze bilanciate allora anche le due sequenze **{** *S* **}** *T* e **[** *S* **]** *T* sono bilanciate. Programmare una macchina di Turing che, dato un nastro iniziale contenente una sequenza (non vuota) di parentesi quadre e graffe, termini la sua esecuzione lasciando sul nastro la *sola* sequenza SI se la sequenza iniziale è bilanciata e la *sola* sequenza NO altrimenti. Esempi:

NASTRO INIZIALE	*NASTRO FINALE*
[]{[]}	SI
[{]}	NO
[[{}][]{}]	SI

Esercizio 3 [Schedina]. Una colonna di schedina *S* è una sequenza simboli 1, 2 e X. Data la colonna vincente *V*, anch'essa costituita da una sequenza di altrettanti simboli scelti tra 1, 2 e X, si vuole verificare che *S* sia vincente, ovvero ci siano *almeno 12* risultati indovinati tra quelli riportati in *V*. Programmare una macchina di Turing che, dato un nastro iniziale contenente le sequenze *S* e *V* separate dal simbolo *, termini la sua esecuzione lasciando sul nastro la *sola* sequenza SI se *S* è vincente e la *sola* sequenza NO altrimenti. Esempi:

NASTRO INIZIALE	*NASTRO FINALE*
1X1X2X21X1X12*1X1X2X21X1X12	SI
1X1X2X21X1X12*1X1X2X21X1X21	NO
1X1X2X21X1X12*1X1X2X21X1112	SI

Esercizio 4 [Divisione per due]. Programmare una macchina di Turing che, dato un nastro iniziale contenente un numero pari decimale *N*, termini la sua esecuzione lasciando sul nastro il risultato della divisione di *N* per 2. Esempi:

NASTRO INIZIALE	*NASTRO FINALE*
1234	617
130	65
0	0

Esercizio 5 [Raddoppio di sequenza]. Programmare una macchina di Turing che, dato un nastro iniziale contenente una sequenza arbitraria *S* di simboli A, B e C, termini la sua esecuzione lasciando sul nastro la sequenza *SS*, cioè la sequenza originale duplicata. Esempi:

NASTRO INIZIALE	*NASTRO FINALE*
ABACB	ABACBABACB
AB	ABAB
C	CC

Esercizio 6 [Divisibilità per sei]. Programmare una macchina di Turing che, dato un nastro iniziale contenente un numero decimale *N*, termini la sua esecuzione lasciando sul nastro la *sola* sequenza SI se *N* è divisibile per 6 e la *sola* sequenza NO altrimenti. Esempi:

NASTRO INIZIALE	*NASTRO FINALE*
30	SI
16	NO
0	SI

Esercizio 7 [Espressioni booleane]. Si vogliono applicare ripetutamente le seguenti regole di sostituzione, dove la sequenza di due o tre simboli (in neretto) a sinistra di ogni freccia va sostituita con il simbolo corrispondente a destra della freccia:

- Sostituzioni NOT: **!0** -> 1, **!1** -> 0
- Sostituzioni AND: ***00** -> 0, ***01** -> 0, ***10** -> 0, ***11** -> 1
- Sostituzioni OR: **+00** -> 0, **+01** -> 1, **+10** -> 1, **+11** -> 1

Una sequenza *S* di simboli 0, 1, !, * e + si dice *risolvibile* se applicando ripetutamente le sostituzioni suddette, in qualunque ordine, si ottiene alla fine un unico simbolo, chiamato *soluzione*, ovvero il simbolo 0 oppure il simbolo 1. Per esempio, se *S* è la sequenza +*1+01*0!*01, si possono applicare le sostituzioni riportate sotto, ottenendo 1 come soluzione (si noti che, nel caso in cui più sostituzioni siano applicabili, l'ordine di applicazione non è rilevante):

+*1**+01***0!*01 -> +*11*0!*01

+***11***0!*01 -> +1*0!*01

+1*0!***01** -> +1*0!0

+1*0**!0** -> +1*01

+1***01** -> +10

+10 -> 1

Programmare una macchina di Turing che, dato un nastro iniziale contenente una sequenza *S* risolvibile, termini la sua esecuzione lasciando sul nastro la soluzione di *S*. Non importa come le sostituzioni vengano realizzate ed eseguite sulla macchina di Turing; è sufficiente che la soluzione finale calcolata (0 oppure 1) sia corretta. Esempi:

NASTRO INIZIALE	*NASTRO FINALE*
1	1
*1!*1+01	0
!!1	1

Esercizio 8 [Somma]. Programmare una macchina di Turing che, dato un nastro iniziale contenente due numeri decimali *N* e *M*, separati dal simbolo +, termini la sua esecuzione lasciando sul nastro la somma di *N* e *M*. Esempi:

NASTRO INIZIALE	*NASTRO FINALE*
30+85	115
23+0	23
0+0	0

Esercizio 9 [Sequenza prefissa]. Una sequenza di simboli A, B, e C si dice *prefissa* secondo la seguente definizione: (i) la sequenza composta di un solo simbolo (A, B oppure C) è prefissa; (ii) se *S* è una sequenza prefissa allora anche le sequenze *SS*A, *SS*B e *SS*C, costruite duplicando *S* e aggiungendo un simbolo in fondo, sono sequenze prefisse. Per esempio, A, AAA, AAC, AACAACC e AACAACCAACAACCA sono prefisse, mentre AA, ABA, AABA e ABAABAC non lo sono (ABAABAC non è prefissa perché ABA non è prefissa). Programmare una macchina di Turing che, dato un nastro iniziale contenente una sequenza di simboli A, B, e C, termini la sua esecuzione lasciando sul nastro la *sola* sequenza SI se la sequenza è prefissa e la *sola* sequenza NO altrimenti. Esempi:

NASTRO INIZIALE	*NASTRO FINALE*
B	SI
AB	NO
AABAABC	SI

Esercizio 10 [Crivello di Eratostene]. Un intero *q* > 1 si dice *primo* se è divisibile solo per 1 e per se stesso. Per esempio, 2, 3, 5, 7, 11, 13, 17 e 19 sono primi. Dato un numero decimale *M*, si vogliono individuare tutti i numeri primi *q* <= *M* usando il seguente algoritmo, che rappresenta una versione semplificata del "crivello di Eratostene". Si marcano inizialmente come primi tutti i numeri da 2 a *M*. Sia *q* l'ultimo numero primo trovato (inizialmente *q* = 2). Si marcano come "non primi" tutti i numeri maggiori di *q* che sono multipli di *q*. Per esempio, se *q* = 2, si marcano 4, 6, 8, 10, 12, 14, ecc. Quindi, si pone *q* uguale al successivo numero che risulta marcato come primo, e si ripete la marcatura finché non ci sono ulteriori primi da esaminare. Usando il simbolo P per marcare un primo e il simbolo N per un "non primo", i numeri che rimangono marcati con P alla fine del crivello sono i numeri primi. Per esempio, per *M* = 20, il crivello esegue i seguenti passi:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

N P P P P P P P P P P P P P P P P P P P

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 q = 2

N P P N P N P N P N P N P N P N P N P N

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 q = 3

N P P N P N P N N N P N P N N N P N P N

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20   q = 5,7,11,13,17,19

N P P N P N P N N N P N P N N N P N P N

Per *M* >= 2, sia data una sequenza formata da un simbolo N seguito da *M* - 1 simboli P, dove l'*i*-esimo simbolo (P o N) corrisponde al numero 1 <= *i* <= *M*. Programmare una macchina di Turing che, dato un nastro iniziale contenente la suddetta sequenza di *M* simboli NPPPPPPPPPPPPPPP..., esegua il crivello di Eratostene e termini l'esecuzione lasciando sul nastro la sequenza di *M* simboli NPPNPNPNNNPNPNNNPNPN... in cui ciascuna P corrisponde a un numero *primo q* <= *M*. Esempi:

NASTRO INIZIALE	*NASTRO FINALE*
NPPPPPPPPPPP	NPPNPNPNNNPN
NPPPPPPPPPPPPPPP	NPPNPNPNNNPNPNNN
NP	NP