Algoritmi e Principi dell'Informatica

Prima Prova in Itinere

17 Aprile 2019

Il tempo a disposizione è di **1h30'**.

Esercizio 1 (punti 5)

Sia h l'omomorfismo (cioè una funzione tale che h(x.y) = h(x).h(y), per x, y stringhe) definito da: h(1) = 1, h(0) = ε.

Si scriva un automa o grammatica a potenza minima per il linguaggio:

$L_1 = \{w.h(w) \mid w \in \{0, 1\}^+\}$.

Esercizio 2 (punti 6)

Si consideri il problema di stabilire se, date due generiche macchine di Turing M1 e M2, esista una stringa accettata da entrambe.

a. Tale problema è decidibile?
b. E' semidecidibile?

Si consideri il problema di stabilire se il linguaggio accettato da una MT è vuoto.

c. Tale problema è decidibile?
d. E' semidecidibile?

Motivare opportunamente le proprie risposte.

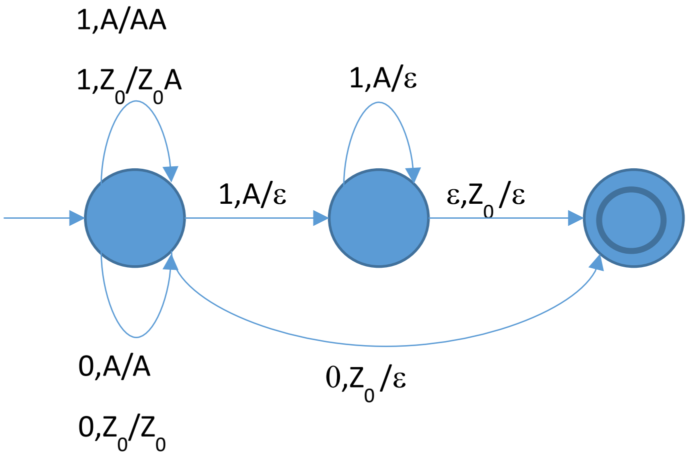
Esercizio 3 (punti 6)

a. Si consideri il linguaggio L_1 dell'esercizio 1. È possibile esprimere L_1 con una formula MFO (logica monadica del prim'ordine)? In caso positivo, si scriva la formula, in caso negativo si motivi la risposta.
b. Come cambia la riposta al punto a) se considero il linguaggio $L_2 = h(L_1) = \{h(x) \mid x \in L_1\}$?

Tracce di Soluzioni

Esercizio 1

Il linguaggio richiede un automa a pila non-deterministico, perché deve identificare la metà degli 1 presenti e, di lì in poi, non ammettere caratteri 0.



Esercizio 2

a) Il problema non è decidibile. Supponiamo per assurdo che esista un algoritmo A in grado di risolvere il problema, ossia che, dati due indici i e j, restituisca "vero" se le TM con indici i e j accettano una stringa in comune e "falso" altrimenti. Allora l'algoritmo A sarebbe anche in grado di risolvere il problema quando uno dei due indici è fissato, ad esempio quando j=100. Tuttavia, per il teorema di Rice, l'insieme degli indici delle TM che accettano una stringa in comune con la TM con indice 100 non è ricorsivo e pertanto non è decidibile stabilire se una generica TM accetti una stringa in comune con la TM con indice 100. Quindi non può esistere un algoritmo A in grado di risolvere il problema di partenza.

b) Il problema è però semidecidibile: è possibile infatti stabilire un principio enumerativo di tutte le coppie <m,n>, dove m indica il numero di passi in cui le TM vengono messe in esecuzione e n è la lunghezza della stringa da passare in ingresso alle TM; per ciascuna coppia <m,n> si provano tutte le (finite) stringhe di lunghezza n. In questo modo, è possibile testare tutte le stringhe in ingresso con un qualunque numero di passi su entrambe le macchine, e se esiste una stringa accettata da entrambe, prima o poi lo si scoprirà.

c) Il problema non è decidibile: il linguaggio vuoto è un particolare insieme di linguaggi, non vuoto (!) e non certo contenente tutti i linguaggi riconosciuti da MT; quindi si può applicare anche in questo caso il teorema di Rice.

d) Il problema non è neanche semidecidibile poiché è invece semidecidibile il problema complemento, ossia il problema di stabilire se un linguaggio *non* è vuoto: basta enumerare con tecnica diagonale y mosse della macchina sulla generica stringa x: se esiste una stringa accettata, prima o poi essa viene riconosciuta. Se fossero semidecidibili sia il problema diretto, sia il problema complemento, allora il problema dovrebbe essere anche decidibile, ma abbiamo mostrato al punto precedente che non lo è.

Esercizio 3

a. La logica MFO è meno espressiva dei linguaggi regolari; essendo necessario un automa a pila n.d. per L_1, non può esistere una formula MFO per L_1.
b. Non cambia: il linguaggio L_2, delle stringhe unarie pari, è esattamente il linguaggio usato per mostrare che MFO non può esprimere tutti i regolari.

Algoritmi e Principi dell'Informatica

Midterm test, English version

April 17th, 2019

The available time to complete the test is **1h30’**.

Exercise 1 (5 points)

Let h be the homomorphism (i.e. a function such that h(x.y) = h(x).h(y), with x, y strings) defined by: h(1) = 1, h(0) = ε.

Define an automaton or a grammar of minimum expressive power for the following language:

$L_1 = \{w.h(w) \mid w \in \{0, 1\}^+\}$.

Exercise 2 (6 points)

Consider the problem of establishing whether, given two generic Turing machines M1 and M2, there exists a string accepted by both machines.

a. Is this problem decidable?
b. Is it semidecidable?

Consider the problem of establishing whether the language accepted by a Turing machine M is empty.

c. Is this problem decidable?
d. Is it semidecidable?

Provide a clear explanation of your answers.

Exercise 3 (6 points)

a. Consider the language L_1 defined in Exercise 1. Is it possible to express such a language with a MFO (Monadic First-Order) formula? If the answer is yes, write the formula; if the answer is no, explain why.
b. Consider the language $L_2 = h(L_1) = \{h(x) \mid x \in L_1\}$ for the problem of point a). Does the answer change for this language? Explain why.