ESEMPI ed ESERCIZI

*) Mostrare che

$$A_{CFG} = \{ \langle G, w \rangle : G \in CFG, \; w \in L(G) \}$$

è DECIDIBILE.

Problema: Ci possono essere infinite derivazioni. Idea. G avente una proprietà:

FATTO Se G è in CNF allora $\forall\, w \in L(G)$ t.c. $|w| = n$ sono richiesti $2n-1$ passi per ogni sua derivazione.

Si dimostra per induzione:

- Se $n = 1$, allora c'è la regola $S \to w$ e infatti $2 \cdot 1 - 1 = 1$.
- Se $n \geq 2$, essendo vero per lunghezze $< n$ e sia $w \in L(G)$ lunga n.

Sicuramente: $S \Rightarrow \underbrace{B}_{w_1}\underbrace{C}_{w_2} \overset{*}{\Rightarrow} w$

t.c. $|w_1| = i < n$ e $|w_2| = n - i < n$

Per ipotesi induttiva,

$$\#\text{ passi per } w = 1 + 2i - 1 + 2(n-i) - 1$$

$$= 2i + 2n - 2i - 1$$

$$= 2n - 1$$ ✓

Adesso: una TM prova tutte le derivazioni di lunghezza $2n-1$. Quindi A_{CFG} è decidibile.

*) Anche questo linguaggio è decidibile:

$$E_{CFG} = \{ \langle G \rangle : G \in CFG,\ L(G) = \emptyset \}.$$

Non posso provare a generare tutte le stringhe perché possono essere infinite.

Useremo invece algoritmo di marcatura:

- Su input $\langle G \rangle$, marca tutti i terminali.
- Ripeti fino a che nessuna nuova

variabile è marcata: marca le variabili di A t.c. G ammette la regola $A \to U_1 \dots U_k$ e $U_1 \dots U_k$ marcati.

- Accetta sse S non è marcata.

*) $L = \{ \langle M \rangle : M \in TM$ t.c. M accetta tutte e sole le stringhe di lunghezza dispari in $\{0,1\}^* \}$

non è DECIDIBILE.

Per riduzione: $A_{TM} \leq_m L$. Ovvero definisco $f : \Sigma^* \to \Sigma^*$ t.c. per ogni input $\langle M, w \rangle \in A_{TM}$ sse $f(\langle M, w \rangle) \in L$.

f CALCOLABILE restituisce $\langle M' \rangle$:

- Su input x, se $|x|$ è pari RIFIUTA.
- Altrimenti esegui M su w e accetta x sse M accetta w.

$(\Rightarrow)$ Se $\langle M, w \rangle \in A_{TM}$, $M(w) = ACC$.

Ora $L(M') = \{ x \in \{0,1\}^* : x$ ha lunghezza dispari $\}$

ovvero $\langle M' \rangle \in L$,

$(\Longleftarrow)$ Se $\langle M, w \rangle \notin A_{TM}$. Allora $M(w) \in \{ REJ, \infty \}$. Ma allora $L(M') = \emptyset$ e $\langle M' \rangle \notin L$.

*) $L = \{ \langle M \rangle : M \in TM;$
$L(M) = \{ 0^n 1^n 0^n : n \geq 0 \} \}$

Anche qui: mostro $A_{TM} \leq_m L$. Costruisco $f : \Sigma^* \to \Sigma^*$ calcolabile t.c.

$\langle M, w \rangle \in A_{TM}$ sse $f(\langle M, w \rangle) \in L$.

Codice per $M' = f(\langle M, w \rangle)$:

- Su input x simula M su w.
- Se $M(w) = ACC$ e $x = 0^n 1^n 0^n$

per $n \geq 0$ allora $M'(x) = ACC$; se $x \neq 0^n 1^n 0^n$, $M'(x) = REJ$.

- Se $M(w) = REJ$, $M'(x) = REJ$.

($\Longrightarrow$) Se $\langle M, w \rangle \in A_{TM}$, allora $M(w) = ACC$. In questo caso $L(M') = \{ 0^n 1^n 0^n : n \geq 0 \}$. Quindi $\langle M' \rangle \in L$.

($\Longleftarrow$) Se $\langle M, w \rangle \notin A_{TM}$, $M(w) \in \{REJ,$

$\infty \}$. In questo caso $L(M') = \emptyset$.

$\langle M' \rangle \notin L$.

(Proposte alternative:

- Su input x, se $x \neq 0^n 1^n 0^n$ allora $M'(x) = REJ$.
- Altrimenti, simula $M(w)$ e accetta $M'(x) = ACC$ sse $M(w) = ACC$

$\langle M, w \rangle \in A_{TM}$, $M(w) = ACC$.

$L(M') = \{0^n 1^n 0^n : n \geq 0\}$.

- Se $M(w) \in \{RES, \infty\}$, allora $L(M') = \emptyset$. Quindi va bene.)

*) Dimostrare: Se A è Turing-RIC. e $A \leq_m \bar{A}$ allora A è DECIDIBILE.

Idea: Se A è TURING-RIC e co-TURING-RIC, allora A è decidibile.

Ma inoltre $A \leq_m \bar{A}$ implica $\bar{A} \leq_m A$ con la stessa mapping reduction f.

Siccome A è TURING-RIC., allora $\bar{A}$ pure lo è.

COMPLESSITÀ

Nell'ultima parte del corso ci interessiamo sull'efficienza degli algoritmi (TM) nella risoluzione di problemi (decisione di linguaggi).

Efficienza in termini di risorse:

- Tempo e spazio (memoria). ⟸
- Randomness
- # processori per calcolo parallelo.

Esempio : linguaggio

$$PATH = \{ \langle G, s, t \rangle : \exists\ s \rightsquigarrow t \}$$

G grafo , s e t nodi.

Quanto Tempo ? Quanto spazio ?

La risposta a queste domande fa parte della TEOLIA DELLA COMPLESSITÀ.

Tantissimi problemi aperti :

- $P = NP$? Ovvero, Trovare una soluzione ne è tanto veloce quanto verificarne una?

- $P = PSPACE$? Se posso risolvere un problema in poco spazio, posso risolverlo in poco tempo?
- $P = BPP$? Ogni algoritmo efficiente può essere derandomizzato?

Iniziamo con il definire la complessità di tempo. Caso deterministico.

DEF Sia M una TM decisore. La complessità di tempo di M è $T : \mathbb{N} \to \mathbb{N}$ t.c.:

$$T(n) = \max_{\substack{x \in \Sigma^* \\ |x| = n}} \{ \text{\# passi richiesti da } M(x) \}.$$

Siamo interessati a come questa funzione si comporta in termini della lunghezza dell'input.