Calcolo delle funzioni e macchine di Turing deter.

Analizziamo le MT deterministiche come trasduttori, cioè come dispositivi capaci di realizzare il calcolo di funzioni parziali, definite in un qualunque dominio

Trasduttori per il calcolo di funzioni su stringhe

Dato un trasduttore $M = \langle \Gamma, \bar{b}, Q, q_0, F, \delta \rangle$ ed una funzione $f: \Sigma^* \to \Sigma^*$ $(\Sigma \subseteq \Gamma)$, M calcola la funzione F $\Leftrightarrow$ $\forall x \in \Gamma^*$:

1) Se $x \in \Sigma^*$ e $f(x) = y$ allora $q_0 x \vdash_M^* x \bar{b} q y$ con $q \in F$
2) Se $x \notin \Sigma^*$ oppure se $x \in \Sigma^*$ e $f(x)$ non è definita allora, assumendo la configurazione iniziale $q_0 x$, o non esistono computazioni massimali (non terminano) opp esistono computazioni massimali che non terminano in uno stato finale

Esempio:

La MTD (esempio lezione precedente) calcola la funzione identità vale la computazione $q_0 x \vdash_M^* x b q_F x$ $\forall x \in \{0,1\}^*$

In generale possiamo definire funzioni da un arbitrario dominio D ed un arbitrario codominio C, e ci possiamo ricondurre ai trasduttori operanti su stringhe e in questo modo identificare una codifica di tali domini sottoforma di stringhe di caratteri.

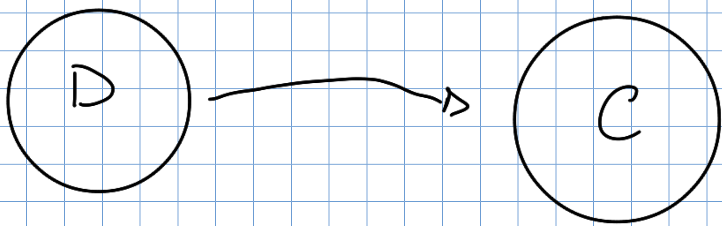
Se consideriamo una funzione intera, sull'alfabeto $\{0, 1, \dots 9\} = \Sigma$ possiamo sempre considerare una funzione intere di lunghezza $|\Sigma|$

Per calcolare il valore $m = f(n)$, partiamo da una configurazione iniziale $q_0 x$, dove x è la codificazione dell'intero n nell' Σ

L'output "m" sul nastro sarà la configurazione finale $x \bar{b} q y$ con $q \in F$ ed y la codificazione di m nell'alfabeto Σ

$D = D_1 \times D_2$

f a più argomenti



Nel caso in cui f è definita in un dominio con più argomenti del tipo $D_1 \times D_2$ basta porre $D = D_1 \times D_2$ in modo da considerare i caratteri come concatenazione delle codifiche di diversi argomenti separati da un carattere speciale

Esempio:

$q_0 \overset{D_1}{10} \# \overset{D_2}{111} \vdash 10 \# 1111 5 q 1001$

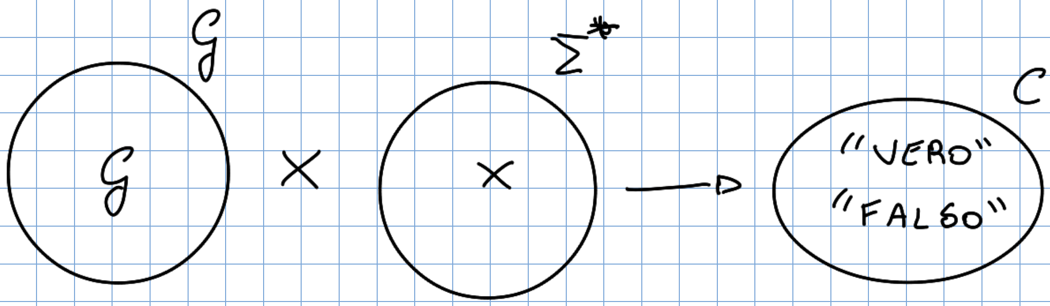
Definizione

Un linguaggio è detto decidibile secondo turing (t-decidibile) se esiste una macchina di turing che lo riconosce

L'esempio 5.4 definisce anche i linguaggi costituiti da stringhe palindrome $\boxed{\text{t-decidibili}}$

Gli ASF sono evidenti MT e il linguaggio di tipo 3 è un linguaggio t-decidibile

Possiamo estendere il concetto di T-decidibilità per la valutazione di predicati, ossia di funzioni con codominio {FALSO, VERO} come per esempio data una qualunque grammatica G di tipo 3 e una qualunque stringa x definita sull'alfabeto Σ di G il predicato appartiene(G, x) che risulta "vero" se $x \in L(G)$ falso altrimenti



$$(G, x) \longmapsto \begin{cases} \text{"vero"} & x \in \mathcal{L}(G) \\ \text{"FALSO"} & x \notin \mathcal{L}(G) \end{cases}$$

Definizione: un linguaggio è detto semi decidibile secondo turing se esiste una MT che lo accetta

Ogni linguaggio T-decidibile $\Rightarrow$ T-semidecidibile
$\not\Leftarrow$

$\hookrightarrow$ questa implicazione vale perché ogni linguaggio riconosciuto da una MT risulta anche accettato

Esempio:

Prendiamo 2 stringhe x e y definiamo la funzione lunghezza $>(x, y)$

"vero" se $|x| > |y|$ e "falso" altrimenti

T-DECIDIBILE

$D = \Sigma^* \times \Sigma^*$

$C = \Sigma^* = \{\text{Vero}, \text{Falso}\}$

Definizione

Una funzione è detta calcolabile secondo Turing (T-calcolabile) se esiste una macchina di Turing che la calcola

- T-calcolabile (esempio 5.4)
- MT con la funzione $f: \Sigma^* \to \Sigma^*$ tale che $f(x) = x$ è T-calcolabile

POSTULATO NOTO COME TESI DI CHURCH-TURING

- Indimostrabile, è unanimamente ritenuto valido. Esso afferma che ogni procedimento algoritmico espresso nell'ambito di un qualunque modello di calcolo è realizzabile mediante una macchina di turing

In base a questa tesi, possiamo parlare in generale di insieme decidibile e di funzione calcolabile senza fare esplicito riferimento alla macchina di turing

La tesi di Church-Turing è indimostrabile perché una dimostrazione di tale postulato richiederebbe che venisse dimostrata l'equivalenza rispetto alla macchina di turing di tutti i possibili metodi di calcolo

---

MACCHINE DI TURING MULTINASTRO

Nasce dall'esigenza di migliorare il fatto che la MT in alcuni contesti è poco agevole per risolvere problemi più strutturati.

e se la confrontiamo con un automa a pila ci rendiamo conto che l'automa a pila ha una separazione tra nastro e input e memoria di lavoro

Indichiamo con MTM, la macchina di turing multinastro a 2 o più nastri

Nastri che contemporaneamente sono accessibili in lettura e scrittura con e' sufficiente di più testine (una per nastro)

Ad esempio, nel caso del riconoscimento delle stringhe palindrome risulta utile disporre due testine, sui due nastri, una sull'estremo sinistro della stringa e una su quello destro, in modo da leggere in maniera efficiente la parola da dx a sx e viceversa

Macchina di turing universale

MT: Astratta, meccanismo elementare, concetti di riconoscimento e accettazione dei linguaggi.

Calcolo di una funzione associata alla MT

Dopo vari passaggi si può dimostrare che i linguaggi di tipo 0 coincidono con i linguaggi accettati dalle macchine di Turing

Definizione di trasduttore (più piccola)

$M = \langle \Gamma, \bar{b}, Q, q_0, F, \delta \rangle$ $\quad f: \Sigma^* \to \Sigma^*$ $\quad (\Sigma \subset \Gamma)$

M calcola $f \iff \forall x \in \Gamma^*$ $<$ se $x \in \Sigma^*$ $\quad f(x) = y \Rightarrow q_0 x \vdash_M^* x \bar{b} q y$, $q \in F$; se $x \notin \Sigma^*$

Estendiamo questa definizione con questa:

Prendiamo una funzione multiargomenti $m : (\Sigma^*)^n \to \Sigma^*$

Diciamo che la macchina M calcola la funzione m se essa realizza la computazione

$q_0 x_1 \bar{b} \dots \bar{b} x_n \vdash_M^* x_1 \bar{b} \dots \bar{b} x_n \bar{b} q y$ con $q \in F \iff m(x_1, \dots x_n) = y$

Definizione

Una macchina di turing universale $U = < \Gamma, \bar{b}, Q', \delta', q_0', F' >$ dove

$\bar{b}$ = blank

Q' = insieme stati

δ' = funzione di transizione

q_0 = stato iniziale di U

F' = insieme degli stati finali

Si dice macchina di turing Universale se essa calcola una funzione $u : (\Gamma^*)^{n+1} \longmapsto \Gamma^*$ con le seguente proprietà: Data una qualunque macchina di turing $M = < \Gamma, \bar{b}, Q, \delta, q_0, F >$ che calcola la funzione $m : (\Gamma^*)^n \longmapsto \Gamma^*$ esiste una string $c_M \in \Gamma^*$, che chiamiamo codifica di M tale che

$$u(c_M, x_1, \dots x_n) = m(x_1 \dots x_n)$$

Compiti delle vacanze

1) MT bene

2) Riprendere tutto (prove in itinere prese)