Linguaggi Formali e Automi

Informatica Triennale: I anno
Docente: Beatrice Palano
Data: 17 giugno 2015

Matricola: .... Cognome: .... Nome: ....

La durata dell'esame scritto è di **2 ore**. Con la sufficienza si accede all'orale.

Buono	Sufficiente	Insufficiente

---

1. Dati i due linguaggi $A = \{0,1\}^* \cdot \{0\}$ e $B = \{0,1\}^* \cdot \{1\}$, rispondere alle seguenti domande **giustificando la risposta**:

   (a) Che linguaggio è $A \cap B$?

   Soluzione: Considerando che A è l'insieme delle parole binarie che terminano con 0 e B l'insieme delle parole binarie che terminano con 1, $A \cap B = \emptyset$. Infatti, non esiste nessuna parola binaria che abbia come suffisso contemporaneamente il simbolo 0 e 1.

   (b) $A \cup B = \{0,1\}^*$?

   Soluzione: Considerando che $\{0,1\}^*$ è l'insieme di tutte le parole binarie di lunghezza finita compresa la parola vuota ε, la risposta è NO. L'unione dei linguaggi A e B produce l'insieme delle parole binarie che terminano o con 0 o con 1, ma ε non soddisfa nessuna delle due condizioni perchè non contiene alcun simbolo, quindi non è compresa nell'unione $A \cup B$.

   (c) $A \cdot B = B \cdot A$?

   Soluzione: In genere, il prodotto di due linguaggi non è commutativo. Ma questo non basta per giustificare la risposta che è NO. È sufficiente osservare che un linguaggio, ad esempio $A \cdot B$, contiene una parola che l'altro linguaggio, $B \cdot A$, non contiene. Consideriamo la parola 01 essa è chiaramente una parola appartenente ad $A \cdot B$, ma terminando con 1 non può chiaramente appartenere a $B \cdot A$, che contiene solo parole con suffisso 0.

(d) $A^* = A \cup \{\varepsilon\}$?

Soluzione: La risposta è SI. Per dimostrare l'uguaglianza di due linguaggi bisogna far verdere l'inclusione reciproca tra di essi, dobbiamo mostrare cioè che:

$$A^* \subseteq A \cup \{\varepsilon\} \text{ e che } A^* \supseteq A \cup \{\varepsilon\}$$

$A^* \subseteq A \cup \{\varepsilon\}$. Si osservi che le parole in A^* sono parole ottenute dalla concatenazione di parole con suffisso 0 e pertanto contenute in A. L'unica parola che fa eccezione a questa regola è ε, ma essa è espressamente inclusa nel linguaggio $A \cup \{\varepsilon\}$.
$A^* \supseteq A \cup \{\varepsilon\}$. Il linguaggio A e il linguaggio $A^0 = \{\varepsilon\}$ sono chiaramente inclusi in A^* per definizione. Infatti, $A^* = A^0 \cup A^1 \cup A^2 \cup \cdots$.

2. Dare la definizione di linguaggio ricorsivo.

Soluzione: Informalmente, un linguaggio ricorsivo è un linguaggio che ammette un sistema riconoscitivo, cioè un sistema che è in grado di stabilire l'appartenenza o meno delle parole al linguaggio stesso. Tale risposta non può essere completamente accetta perchè non è formalizzata e priva della possibilità di essere utilizzata a livello progettuale nella costruzione di programmi che mostrano proprietà sui linguaggi ricorsivi.

Pertanto, la risposta che deve essere data è la definizione *formale* di linguaggio ricorsivo.
Definizione:
Un linguaggio $L \subseteq \Sigma^*$ si dice ricorsivo quando esiste un algoritmo implementato da un programma w tale che, per ogni $x \in \Sigma^*$, si ha:

$$F_w(x) = \begin{cases} 1 & x \in L \\ 0 & x \notin L \end{cases}$$

dove con F_w si indica la semantica del programma w.

3. Sia la grammatica

$$G = (\ T = \{(,)\},\ V = \{S\},\ S,\ P = \{S \to (S),\, S \to (\,)S,\, S \to (\,)\}\ ).$$

(a) Rispondere alle seguenti domande **giustificando la risposta**:

- La parola "$(\,)(\,)((()))$" appartiene a $L(G)$?

  Soluzione: Si, infatti una parola appartiene al linguaggio generato da una grammatica, in questo caso G, quando è possibile individuare una derivazione per essa in G. Per giustificare la risposta dobbiamo quindi dare l'intera derivazione della parola: $S \Rightarrow (\,)S \Rightarrow (\,)(\,)S \Rightarrow (\,)(\,)(S) \Rightarrow (\,)(\,)((S)) \Rightarrow (\,)(\,)((()))$.
- La parola "(S)" appartiene a $L(G)$?

  Soluzione: No, una parola per appartiene al linguaggio generato da una grammatica, in questo caso G, non deve solo ammettere una derivazione per essa in G, ma deve anche essere composta da soli simboli terminali che in questo caso sono definiti dall'insieme $T = \{(,)\}$. La parola (S) non appartiene a T^*, infatti, ammette come fattore il simbolo $S \in V$. Nessuna parola contenente una variabile o metasimbolo di G appartiene a $L(G)$.

(b) Completare la frase:
La grammatica G **non** è di tipo .... **Soluzione:** Tre.

4. Dare una grammatica di tipo 2 per il linguaggio $\{a^n c^m b^n \mid n, m \geq 0\}$.

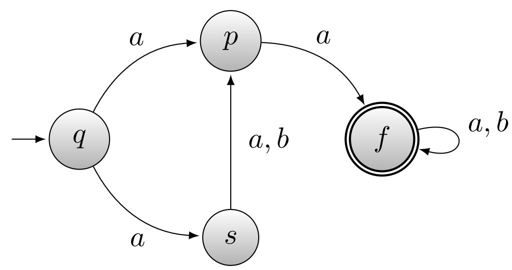
Soluzione: Diamo una possibile grammatica di tipo 2 per il linguaggio $\{a^n c^m b^n \mid n, m \geq 0\}$, ricordando la grammatica per il famoso linguaggio $\{a^n b^n \mid n \geq 0\}$ ed usandola per ottenere quella del linguaggio in questione. Definiamo:

$$G = (\ T = \{a, b, c\},\ V = \{S, C\},\ S,\ P = \{S \to aSb,\, S \to C,\, C \to cC,\, C \to \varepsilon\}\ ).$$

La grammatica G permette di generare parole della forma $a^n S b^n$, per $n \geq 0$, semplicemente iterando la regola $S \to aSb$. Per eliminare la variabile S dobbiamo necessariamente passare dalla variabile C con la regola $S \to C$, ottenendo $a^n C b^n$. A questo punto è possibile inserire un numero arbitrario m, di simboli c con le regole $C \to cC$, $C \to \varepsilon$, ottenendo una parola della forma $a^n c^m b^n$. Si noti che dovendo una derivazione partire da S, l'assioma di G, non è possibile ottenere altre parole che non siano della forma richiesta.

Dobbiamo ora giustificare che la grammatica G che abbiamo appena dato sia di tipo 2. Una grammatica è di tipo 2 se le sue regole di produzione sono della forma $A \to \beta$ con $A \in V$ e $\beta \in (V \cup T)^*$. Ogni regola in P soddisfa questa condizione e pertanto G è di tipo 2.

5. Sia A il seguente automa nondeterministico seguente:



Disegnare un automa deterministico equivalente ad A.

Soluzione: Usando la subset construction mostrata nel teorema di equivalenza tra gli automi a stati finiti deterministici e nondeterministici si ottiene l'automa deterministico seguente: