Nota ai lettori

Questi appunti sono basati sulle lezioni dell A.A. 2023/2024, integrate con passi tratti dal libro "Linguaggi Formali e Compilazione", e formattati seguendo la suddivisione in paragrafi di quest'ultimo.

1 Teoria formale del linguaggio

1.1 Alfabeto e linguaggio

Un **alfabeto** è un insieme finito di elementi chiamati **simboli terminali** o **caratteri**. $\Sigma = \{a_1, a_2, \ldots, a_k\}$ è un alfabeto composto da k elementi (la sua cardinalità è k). Una **stringa** (o **parola**) è una sequenza, ovvero un insieme ordinato eventualmente con ripetizioni, di caratteri.

Un **linguaggio** è un insieme di stringhe di un alfabeto specifico. Dato un linguaggio, una stringa che gli appartiene è detta **frase**.

La **cardinalità** di un linguaggio è definita dal numero di frasi che contiene. Se la cardinalità è finita, il linguaggio si dice **finito**.

Un linguaggi finito è una collezione di parole, solitamente chiamate **vocabolario**. Il linguaggio che non contiene frasi è chiamato **insieme vuoto** o **linguaggio** $\emptyset$.

La **lunghezza** $|x|$ di una stringa x è il numero di caratteri che contiene.

1.1.1 Operazioni sulle stringhe

Date le stringhe

$$x = a_1 a_2 \ldots a_h \qquad\qquad y = b_1 b_2 \ldots b_k$$

la **concatenazione**, indicata con $\cdot$, è definita come:

$$x \cdot y = a_1 a_2 \ldots a_h b_1 b_2 \ldots b_k$$

La concatenazione non è commutativa, ma è associativa.

1.1.2 Stringa vuota

La **stringa vuota** (o **nulla**), denotata con ϵ, soddisfa l'identità:

$$x \cdot \varepsilon = \varepsilon \cdot x = x$$

La stringa vuota non deve essere confusa con l'insieme vuoto; infatti, l'insieme vuoto è un linguaggio che non contiene stringhe, mentre il set $\{\varepsilon\}$ ne contiene una, la stringa vuota.

1.1.3 Sottostringa

Sia la stringa $x = uyv$ il prodotto della concatenazione delle stringhe u, y e v: le stringhe u, y e v sono **sottostringhe** di x. In questo caso, la stringa u è un **prefisso** di x e la stringa v è un **suffisso** di x. Una sottostringa non vuota è detta **propria** se non coincide con x.

1.1.4 Inversione di stringa

L'**inverso** di una stringa $x = a_1 a_2 \ldots a_h$ è la stringa $x^R = a_h a_{h-1} \ldots a_1$.

1.1.5 Ripetizione

La potenza m-esima x^m di una stringa x è la concatenazione di x con se stessa per $m-1$ volte. Esempi:

$$x = ab \qquad x^0 = \varepsilon \qquad x^2 = (ab)^2 = abab$$

1.2 Operazioni sul linguaggio

L'inverso L^R di un linguaggio L è l'insieme delle stringhe che sono l'inverso di una frase di L.

2 Automi

2.1 Automi a pila

Gli **automi a pila** sono automi a stati finiti che utilizzano una **pila** (stack) come memoria aggiuntiva.

Un automa a pila è definito dalla 7-upla

$$\langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle$$

- Q: insieme degli stati
- Σ: alfabeto che descrive il linguaggio
- Γ: alfabeto della pila
- δ: funzione di transizione
- q_0: stato iniziale
- Z_0: fondo della pila
- F: stato (o stati) finale

L'input è una tripla, denotata come:

$$(q, a, A) \to (x, XX)$$

con:

- q: stato corrente
- a: il simbolo della stringa da leggere
- A: il contenuto dello stack

Il simbolo di fine stringa è $\swarrow$.

Negli automi a pila ci sono due tipi di accettazione: l'**accettazione per stato finale**, quando è stato consumato tutto l'input e si giunge ad uno stato finale, e l'**accettazione per pila vuota**, quando è stato consumato tutto l'input e la pila è vuota (anche senza Z_0).

Essendo l'automa non deterministico, bisogna fare tutte le computazioni possibili (quindi esplorare tutte le possibilità).

Processiamo una stringa nella forma ca^nb^n con $n \geq 1$, ad esempio $caabb\swarrow$.

$$\Gamma = \{Z_0, X\}$$

Z_0 è sempre presente, X è il simbolo che gestisce il bilanciamento. Consumando c, si passa dallo stato q_0 allo stato q_1.

$$(q_0, c, Z_0) \to (q_1, Z_0)$$

Ora non sarà più possibile incontrare c. Consumiamo a:

$$(q_1, a, Z_0) \to (q_1, Z_0X)$$

X serve a contare le a. La funzione (q_1, Z_0X) rimane in q_1; la testa della pila contiene X, quindi la funzione (q_1, a, Z_0) non può scattare. Bisogna definire una nuova:

$$(q_1, a, X) \to (q_1, XX)$$

Se la parola è corretta, prima o poi si incontrerà una b. Passiamo allo stato q_2 per non incontrare più a.

$$(q_1, b, X) \to (q_2, \varepsilon)$$

Non può esserci Z_0, altrimenti sarebbe come se non avessimo mai incontrato nessuna a. Il passaggio a q_2 è obbligato.

$$(q_2, b, X) \to (q_2, \varepsilon)$$

$$(q_2, \swarrow, X) \to (q_3, Z_0)$$

Incontrerò il fine stringa quando aavrò rimosso tutti gli X dalla pila. Lo stato finale conterrà solo q_3.

Input	Pila	Stato	Commenti
$caabb \swarrow$	Z_0	q_0	devo consumare c e Z_0
$aabb \swarrow$	Z_0	q_1	devo consumare a
$abb \swarrow$	Z_0X	q_1	devo trovare tripla (q_1, a, X)
$bb \swarrow$	Z_0XX	q_1	ogni volta che incontro una a, metto X sulla pila
$b \swarrow$	Z_0X	q_2	consumo la testa della pila, non scrivo nulla
$\swarrow$	Z_0	q_2	
	Z_0	q_3	la parola appartiene al linguaggio

Una grammatica context free genera da un non terminale una sequenza di terminali e non terminali, combinati in qualunque modo; è una quadrupla nella forma

$$G = \langle V, \Sigma, P, S \rangle$$