


Appunti di Linguaggi Formali e Compilatori

Matteo Gianello

8 gennaio 2015

Quest'opera è stata rilasciata con licenza Creative Commons Attribuzione - Non commerciale - Condividi allo stesso modo 4.0 Internazionale. Per leggere una copia della licenza visita il sito web <http://creativecommons.org/licenses/by-nc-sa/4.0/>. .

Indice

1	Introduzione	3
1.1	Introduzione alla teoria dei linguaggi formali	3
1.1.1	Operazioni con le stringhe	4

1 Introduzione

Prima di entrare nel dettaglio della nostra trattazione definiamo che cos'è un *Linguaggio Formale*. Innanzitutto dobbiamo distinguere tra un *linguaggio natura* ed uno *artificiale*, un linguaggio naturale è quello che utilizziamo tutti i giorni e che si basa sul significato delle parole e soprattutto non possiede una struttura formale. Un linguaggio artificiale, invece, è quello utilizzato per la comunicazione tra macchine, è di tipo non verbale ma soprattutto è formale.

Un linguaggio si definisce **formale** se la sua *sintassi* (ovvero la sua struttura) e la sua *semantica* (ovvero il suo significato) sono definiti da precisi algoritmi. Questo permette di definire delle *procedure* precise per determinare la correttezza di una frase che appartiene ad un linguaggio e il suo significato. In un contesto più ristretto un **linguaggio formale** è un *oggetto matematico* costruito su di un *alfabeto* tramite regole assiomatiche chiamate *grammatiche* o tramite strumenti matematici chiamati *automi*.

1.1 Introduzione alla teoria dei linguaggi formali

Vediamo ora alcune definizioni che ci saranno utili durante il corso:

Alfabeto: insieme *finito* di elementi

$$\Sigma = \{a_1, a_2, a_3, \dots, a_k\}$$

Cardinalità di un alfabeto: è il numero di elementi che lo compongono

$$|\Sigma| = k$$

Stringa: insieme ordinato di elementi dell'alfabeto che possono essere anche ripetuti.

$$\Sigma = \{a, b, c\}$$

$$\text{Stringe} : abc \vee aab \vee ac \vee bbb$$

Linguaggio: insieme finito o infinito di stringhe di un alfabeto

$$L_1 = \{ab, ac, abc\}$$

$$L_2 = \{bc, bbc, \dots\}$$

La struttura insiemistica del linguaggio formale ha due livelli di profondità:

- Insieme non ordinato di elementi non atomici (*stringhe*) che sono
- sequenze ordinate di elementi atomici (*elementi o terminali*).

Cardinalità di un linguaggio: può essere finita o infinita.

$$|L_1| = |\{ab, ac, abc\}| = 3$$

Lunghezza di una stringa: indica il numero dei suoi elementi

$$|abb| = 3 \quad |abbc| = 4$$

Uguaglianza tra stringhe: due stringhe si definiscono uguali se e solo se hanno la stessa lunghezza, ed i suoi elementi coincidono ordinatamente da sinistra a destra.

$$x = a_1 a_2 \dots a_h \quad y = b_1 b_2 \dots b_k$$

$$x = y \iff h = k \wedge a_i = b_i \quad \forall i = 1 \rightarrow h$$

$x = abccbc$ <i>prefissi</i> : $a, ab, abc, abcc, abccb, abccbc$ <i>suffissi</i> : $c, bc, cbc, ccbc, bccbc, abccbc$ <i>sottostringhe</i> : $....., bc, cc, cb, ...$

Figura 1: Esempio di sottostringhe con prefissi e suffissi

1.1.1 Operazioni con le stringhe

La *concatenazione* è l'operazione fondamentale con le stringhe, si tratta di fare il prodotto tra esse. Questa operazione è un'operazione di base e non fa altro che disporre in fila le due stringhe.

$$x = a_1 a_2 \dots a_h \quad y = b_1 b_2 \dots b_k$$

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

Questo tipo di operazione gode della proprietà associativa e va ad influire sulla lunghezza della stringa come vediamo dalla 1 e dalla 2.

$$(xy)z = x(yz) \tag{1}$$

$$|xyz| = |x| + |y| + |z| \tag{2}$$

La concatenazione ci permette di introdurre un altro concetto relativo alle stringhe, ovvero quello della *stringa vuota* o *nulla* indicata con ε questa stringa è l'elemento neutro rispetto all'operazione di concatenazione, infatti, concatenando ε a destra o a sinistra di una stringa la stringa non cambia.

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

Tuttavia bisogna prestare attenzione a non confondere la stringa nulla con l'insieme vuoto Φ . Una volta capita l'operazione di concatenazione si possono individuare in una stringa delle *sotto-stringhe*, ovvero stringhe più piccole concatenate tra loro per formare una stringa più grande. Ad esempio data una stringa x essa può essere vista come un insieme di sotto-stringhe concatenate

$$x = u y v$$

Dove:

- y è detta sotto-stringa.
- u è chiamato prefisso.
- v è chiamato suffisso.

Un esempio di suffissi, prefissi e sotto-stringhe è mostrato in Figura 1.

Si definiscono *sotto-stringhe proprie* tutte quelle stringhe che hanno $u \neq \varepsilon$ e $v \neq \varepsilon$

Elenco delle figure

1	Esempio di sottostringhe con prefissi e suffissi	4
---	--	---