

Indice

1	Convenevoli	2
2	Definizione di un linguaggio	2

1 Convenevoli

Studieremo dei sistemi particolari (macchine/grammatiche) e il loro comportamento (il linguaggio descritto) e le risorse utilizzate (risorse utilizzate, risorse per trasformazioni da una macchina e l'altra, ...).

2 Definizione di un linguaggio

Quando si parla di un linguaggio ci sono due aspetti fondamentali:

- sintassi, formata da
 - i segni o simboli utilizzati
 - le regole che ci permettono di combinarli
- semantica, il processo che associa alle frasi del linguaggio un significato

Noi ci occuperemo principalmente di sintassi.

In ambito linguistico nel 1956 Chomsky introduce le grammatiche formali. Questi sono degli oggetti matematici, il cui scopo originale era descrivere in modo rigoroso la grammatica dell'inglese. Nello stesso periodo si stava sviluppando l'idea di compilatore, specificamente il compilatore Fortran. Si vide che il modello proposto da Chomsky andava benissimo per formalizzare i linguaggi di programmazione.

Definiamo

- l'alfabeto (Σ) è un insieme finito non vuoto di simboli, e.g.

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

- una stringa o parola è una sequenza di simboli sull'alfabeto. La sua lunghezza è definita come il numero di simboli che la compongono

$$|abc| = 3$$

Tra tutte le parole abbiamo una parola speciale, che è la parola vuota costituita da zero simboli. Questa è indicata con ϵ (alcuni la indicano con λ o Λ). La lunghezza di questa è 0.

- indichiamo l'insieme di tutte le possibili stringhe sull'alfabeto Σ con Σ^* . Queste sono enumerabili.

Chiamiamo Σ^n l'insieme di tutte le stringhe di n lettere sull'alfabeto Σ .

- definiamo un'operazione

$$\cdot : \Sigma^* \times \Sigma^* \mapsto \Sigma^*$$

come la concatenazione o prodotto di due stringhe. Ad esempio

$$x = abbac$$

$$y = ca$$

$$xy = abbacca$$

$$yx = caabbac$$

La parola vuota ϵ è l'identità sia destra che sinistra di questa operazione. L'operazione è associativa ma non commutativa. La struttura $\langle \Sigma^*, \cdot, \epsilon \rangle$ è detto monoide libero generato dall'alfabeto Σ .

- data una stringa $x \in \Sigma^*$ diciamo che y è prefisso di x se

$$\exists z \in \Sigma^* | x = yz$$

Una stringa ha un numero di prefissi pari alla sua lunghezza + 1. y è prefisso proprio di x se è prefisso e non è ϵ . Similmente definiamo il suffisso di una parola x , quindi y è suffisso di x se

$$\exists x \in \Sigma^* | x = zy$$

Infine diciamo che y è fattore di x se

$$\exists z, w \in \Sigma^* | x = zyw$$

I prefissi e i suffissi sono particolari tipi di fattori. Ogni parola ha all'incirca $\frac{n^2}{2}$ fattori, infatti è facile notare che $x = aaaa$ ha diversi fattori uguali.

Una sottosequenza di solito è una serie di elementi scelti da una parola, quindi un fattore può essere definito come una sottosequenza contigua. Ad esempio una sottosequenza di $x = abbac$ può essere abc . Sottostringa e sottoparola di solito sono sinonimi di fattore, ma a volte possono essere utilizzate come sinonimo di sottosequenza.

- un linguaggio L è definito come un qualunque sottoinsieme di Σ^* . Abbiamo alcuni linguaggi particolari:
 - il linguaggio vuoto $L =$
 - il linguaggio della parola vuota $L = \epsilon$
 - il linguaggio pieno $L = \Sigma^*$

Ad esempio definiamo il linguaggio delle parole che finiscono con due a come

$$L = \{w \in \Sigma^* \mid \exists y \in \Sigma^* w = yaa\}$$

quindi con un insieme finito di simboli siamo riusciti a rappresentare un insieme infinito. Questo tipo di descrizione è detta dichiarativa.

Definiamo l'alfabeto $\Sigma = 0, 1, \dots, 9$, e definiamo L come tutte le sequenze formate da 3 cifre decimali che appaiono consecutivamente nella rappresentazione del π greco.

In altri casi utilizzeremo delle descrizioni:

- generative: fornisco un insieme di regole che permette di costruire tutte le possibili stringhe del linguaggio, ad esempio le grammatiche
- riconoscitive: fornisco un riconoscitore che riceve una stringa x e risponde alla domanda $x \stackrel{?}{\in} L$