Teoria dei linguaggi

Indice

1. Introduzione

1.1. Richiamo sugli insiemi

Un **insieme** S è una collezione di elementi di un certo dominio U

Fissato un insieme S, se è finito allora $|S|$ indica la sua **cardinalità**, ovvero il numero di elementi che contiene

Un insieme particolare è l'**insieme vuoto** $\emptyset$, l'unico insieme che ha cardinalità 0

Sugli insiemi possiamo definire alcune operazioni:
- *intersezione* $A \cap B$ contiene gli elementi comuni di A e B
- *unione* $A \cup B$ contiene gli elementi di A e B assieme
- *differenza* $A - B$ o A/B contiene gli elementi di A che non sono in B
- *complementare* A^c o $\overline{A}$ contiene gli elementi di U cje non sono in A
- *sottoinsieme* $A \subset B$ proprio oppure $A \subseteq B$ non proprio

1.2. Richiamo sugli alfabeti e i linguaggi

Un **alfabeto** è un insieme di caratteri e simboli sul quale è possibile definire un **linguaggio**: quest'ultimo infatti è un insieme di parole costruite a partire dall'alfabeto dato

Ogni **parola** è una sequenza di caratteri (detti anche simboli o lettere) dell'alfabeto

Tra le parole di un linguaggio c'è anche la **parola vuota** ε

Un linguaggio può essere **finito** (linguaggio composto dalle parole italiane) oppure **infinito** (linguaggio sull'alfabeto $\{(,)\}$ delle parole ben bilanciate)

Per rappresentare i linguaggi possiamo usare diversi modelli
- **dichiarativo**: date delle regole, si verifica se una parola rispetta o meno queste regole, come nelle *espressioni regolari*
- **generativo**: date delle regole, si parte da *un certo punto* e si generano tutte le parole di quel linguaggio con le regole date
- **riconoscitivo**: si usano dei *modelli di calcolo* che prendono in input una parola e dicono se appartiene o meno al linguaggio

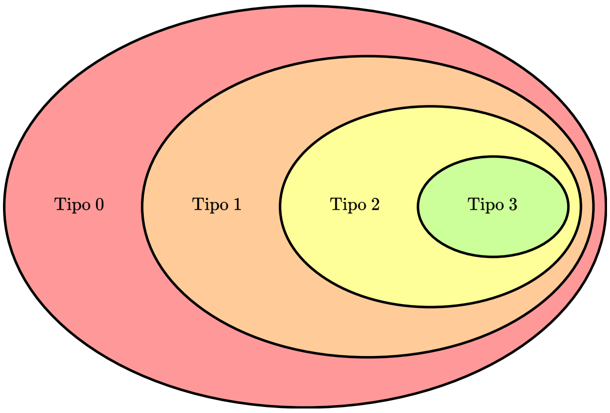
Considerando il linguaggio sull'alfabeto $\{(,)\}$ delle parole ben bilanciate, proviamo a dare due modelli
- *generativo*: a partire da una sorgente S devo applicare delle regole per derivate tutte le parole di questo linguaggio
  Proviamo a definire alcune regole per questo linguaggio
  - la parola vuota ε è ben bilanciata
  - se x è ben bilanciata, allora anche (x) è ben bilanciata
  - se x, y sono ben bilanciate, allora anche xy sono ben bilanciate

  I modelli generativi generano le **grammatiche**
- *riconoscitivo*: abbiamo una *black-box* che prende una parola e ci dice se appartiene o meno al linguaggio (in realtà questa *black-box* potrebbe non terminare mai la sua esecuzione)
  Proviamo a definire alcune regole per riconoscere tutte le parole di questo linguaggio:
  - il numero di (è uguale al numero di)
  - per ogni prefisso, il numero di (deve essere maggiore o uguale al numero di)

1.3. Gerarchia di Chomsky

Negli anni 50 **Noam Chomsky** studia la generazione dei linguaggi formali e crea una **gerarchia di grammatiche formali**

- **tipo 0**: grammatiche che generano tutti i linguaggi, sono senza restrizioni e come modello equivalente hanno le **macchine di Turing**
- **tipo 1**: grammatiche *context-sensitive* (dipendenti dal contesto), come modello equivalente hanno le **Linear Bounded Automata**
- **tipo 2**: grammatiche *context-free* (libere dal contesto), hanno come modello equivalente gli **automi a pila**
- **tipo 3**: grammatiche regolari, hanno come modello equivalente gli **automi a stati finiti**



1.4. Richiamo sulla notazione delle parole

Andremo ad indicare con le lettere maiuscole dell'alfabeto greco un **alfabeto**, come ad esempio Σ o Γ

Sia Σ un alfabeto, indichiamo con Σ^* tutte le parole sull'alfabeto Σ, compresa la parola vuota ε, e con Σ^+ tutte le parole non vuote sull'alfabeto Σ: in poche parole, $\Sigma^+ = \Sigma^* / \{\varepsilon\}$

Vediamo prima alcune operazioni sulle parole
- la *concatenazione* è la composizione di due parole x, y che formano la parola $w = xy$, e in generale $w_1 = xy \neq yz = w_2$
- la *lunghezza* di una parola w si indica con $|w|$
- il *numero di occorrenze* di una lettera $a \in \Sigma$ nella parola w si indica con $|w|_a$

Vediamo ora alcune proprietà delle parole
- **fattore**: y si dice *fattore* di w se esistono $x, z \in \Sigma^*$ tali che $xyz = w$
- **prefisso**: x si dice *prefisso* di w se esiste $y \in \Sigma^*$ tale che $xy = w$
  - **prefisso proprio** se $y \neq \varepsilon$
  - **prefisso non banale** se $x \neq \varepsilon$
- **suffisso**: y si dice *suffisso* di w se esiste $x \in \Sigma^*$ tale che $xy = w$
  - **suffisso proprio** se $x \neq \varepsilon$
  - **suffisso non banale** se $y \neq \varepsilon$
- **sottosequenza**: x si dice *sottosequenza* di w se x è ottenuta eliminando 0 o più caratteri da w, anche non in ordine

Terminiamo definendo l'operazione **reversal**, ovvero se $w = a_1 a_2 ... a_n$, allora $w^R = a_n a_{n-1} ... a_1$

Una parola w si dice **palindroma** se $w = w^R$