Operazioni tra linguaggi

Materia	Fondamenti di Informatica: Linguaggi Formali
Data	@March 11, 2024

Potenza di un linguaggio

Sia $L \subseteq \Sigma^*$. La **potenza di un linguaggio** è denotata con L^n con $n \geq 0$ ed è così definita:

- $L^0 = \{\, \epsilon \,\}$;
- $L^n = L^{n-1} \cdot L \;\; \forall n > 0$.

Visto che qualunque linguaggio elevato a 0 è uguale a $\{\, \epsilon \,\}$, anche $\Lambda^0 = \{\, \epsilon \,\}$.

La cardinalità di L^n è finita $\iff$ $|L|$ è finita. Però **non è detto che** $|L^n| = |L|^n$, perché possiamo ottenere una certa stringa in modi diversi in base agli elementi di L. Diremo quindi che se $|L| = k, |L^n| \leq k^n$.

Star di Kleene, chiusura riflessiva di un linguaggio

Sia $L \subseteq \Sigma^*$ un linguaggio. La **chiusura riflessiva di** L si denota con L^* ed è così definita:

$$L^* = \bigcup_{n \geq 0} L^n \; oppure \; \bigcup_{n}^{\infty} L^n$$

L'operatore $*$ viene chiamato **star di Kleene** (letto "clini"). La star di Kleene è l'operazione, il risultato è la chiusura riflessiva.

Il linguaggio che si ottiene è un **linguaggio infinito**, in quanto operiamo l'unione di linguaggi di numero infinito. Tuttavia, se $L = \{\, \epsilon \,\}$, la chiusura riflessiva di L è uguale a $\{\, \epsilon \,\}$; anche quando $L = \Lambda$ avremo $\Lambda^* = \{\, \epsilon \,\}$.

Se partiamo da Σ, l'alfabeto si può considerare come un linguaggio in cui ogni stringa si ottiene con la concatenazione $\epsilon \cdot a \;\; \forall a \in \Sigma$. Abbiamo detto che Σ^* per definizione è l'insieme di tutte le stringhe su Σ. La definizione **coincide con la chiusura riflessiva di** Σ, infatti otteniamo sempre l'insieme di tutte le stringhe possibili, che si possono ottenere con tutte le possibili concatenazione dei simboli di Σ.

Chiusura positiva di un linguaggio (non riflessiva)

Sia $L \subseteq \Sigma^*$. La chiusura di L si denota L^+ ed è così definita:

$$L^+ = \bigcup_{n \geq 1} L^n$$

Ne segue che **non necessariamente** $\epsilon \in L^+$, ed in particolare $\epsilon \in L^+ \iff \epsilon \in L$. Inoltre, $\Lambda^+ = \Lambda$.

Tale chiusura viene anche chiamata **chiusura positiva di** L.

Possiamo inoltre affermare che $L^* = L^+ \cup \{ \epsilon \}$. Inoltre, $L^+ = L^* \setminus \{ \epsilon \} \iff \epsilon \notin L$.

Complemento di un linguaggio

Sia $L \subseteq \Sigma^*$. Il **complemento di** L **è l'insieme** $L^C = \Sigma^* \setminus L$**. L'insieme non è obbligatoriamente infinito**, infatti posso anche considerare $L = \Sigma^* \setminus \{ a, b \}$: il complemento $L^C = \{ a, b \}$.

Rappresentazioni finite di linguaggi

Per rappresentare un linguaggio vi sono due metodi principali:

- il **metodo riconoscitivo**: si prende una stringa in input e viene elaborato il risultato di appartenenza della stringa ad un linguaggio. La macchina che applica tale metodo è il **riconoscitore**.
- il **metodo generativo**: con una stringa iniziale e un **insieme finito di regole**, riusciamo a **generare tutte le stringhe del linguaggio**, quindi prendiamo la stringa iniziale s e generiamo ogni stringa del linguaggio tramite le regole generative.

Riconoscitori

Il **riconoscitore** è uno strumento in grado di **riconoscere l'appartenenza di stringhe ad un determinato linguaggio**. Il modello di riconoscitore prevede un **nastro** su cui viene scritta questa stringa, come la **macchina di Turing**. Il nastro è diviso in **celle**, su cui vengono scritti i simboli o il **simbolo blank**, cioè un simbolo vuoto che generalmente non appartiene a Σ. Il nastro è **potenzialmente infinito**. Abbiamo inoltre una **testina** in grado di leggere una cella per volta e, nel caso della macchina di Turing, di scrivere sempre una cella per volta. Nel nostro modello la testina si può muovere solo di **una cella per**

volta. Il riconoscitore, inoltre, ha uno **stato** indicabile con la lettera q. La macchina può anche avere una **memoria ausiliaria**, che segue il modello della **pila** (stack).

Supponiamo di avere $L = \{ a^{2n} \mid n \geq 1 \}$. Allora un riconoscitore può leggere una stringa costituita da sole a, e controllare che il numero letto è pari o dispari, alternando ad ogni cella letta lo stato q_{disp} e lo stato q_{pari}.

Il riconoscitore ha una **serie di regole dettate dalla funzione di transizione del riconoscitore**. Se al riconoscitore viene data una stringa $x \in \Sigma^*$, la stringa sarà scritta un simbolo alla volta nelle celle del nastro. Il riconoscitore un passo per volta legge la stringa e darà un output: la stringa appartiene al linguaggio di riferimento o non appartiene a tale linguaggio. **Ad ogni momento della computazione, solamente un numero finito di celle è diverso da blank**, quindi solo un numero finito di celle contiene simboli: tuttavia non possiamo limitare il numero di celle, perché **dobbiamo essere in grado di leggere stringhe di qualsiasi lunghezza**, pertanto il nastro sarà di lunghezza infinita.

Questo modello di riconoscitore è **puramente concettuale**, anche perché potenzialmente il nastro è illimitato.

La macchina di Turing nasce come tentativo di formalismo per la logica e per i suoi teoremi, in particolar modo come **simulazione del ragionamento umano**. Inoltre, esiste la congettura di Turing che afferma che **ogni funzione per cui esiste un algoritmo ha una corrispondente macchina di Turing che la calcola**.

La computazione può essere considerata come una **successione di configurazioni** a partire da una configurazione iniziale. Una **configurazione** è una "fotografia" dello stato attuale del riconoscitore. Allora il risultato della computazione coincide con la **configurazione finale del riconoscitore**. Il passaggio dalla configurazione c_1 alla configurazione c_2 si indica con $c_1 \vdash c_2$.

Automi a stati finiti

Si definisce **automa** un qualunque riconoscitore. Un automa si dice **a stato finito se**:

- il nastro è **semi-finito**, quindi non consideriamo un nastro infinito in entrambe le direzioni ma un nastro con un inizio e senza una fine;
- la testina si **muove solo a destra** ed è in grado **solo di operare la lettura.**
- la testina si muove **solo se la cella letta ha un simbolo diverso da 'blank'**: in tal caso l'automa **termina la computazione.**

Essendo un riconoscitore, ha una **funzione di transizione** che permette di cambiare configurazione a partire dallo **stato attuale dell’automa** e dal **contenuto della cella letta**.