Lezione 12 – modelli generativi

Lezione del 28/03/2025

Ri-facciamo il punto

- Siamo partiti cercando di capire come risolvere automaticamente i problemi
- E abbiamo studiato la soluzione proposta da Alan Turing che, partendo dalla sua analisi del processo di soluzione è arrivato a definire un modello di calcolo: la Macchina di Turing
  - che è un linguaggio per descrivere algoritmi
  - e ogni macchina di Turing è un algoritmo
- Poi, abbiamo introdotto i concetti di linguaggi decidibili e accettabili, e di funzioni calcolabili
  - che corrispondono, informalmente, ai problemi che sappiamo risolvere con la Macchina di Turing
- e abbiamo dimostrato che esistono linguaggi non decidibili o non accettabili e funzioni non calcolabili
  - la possibilità che esistano problemi irrisolvibili – con la Macchina di Turing

Modelli di calcolo e linguaggi

- Abbiamo già detto che sono stati definiti numerosi modelli di calcolo
  - ragionevoli
- che sono ***Turing-equivalenti***
- E fra questi modelli abbiamo visto, a titolo di esempio un linguaggio di programmazione: il PascalMinimo
  - che abbiamo dimostrato essere Turing-equivalente
- Sia la Macchina di Turing che il PascalMinimo si occupano dei linguaggi nella medesima maniera
  - dato un linguaggio (ossia un insieme di parole)
  - progettiamo una macchina di Turing o un programma in PascalMinimo
  - che sappia riconoscere le parole che appartengono a quel linguaggio
- ossia, sono modelli di calcolo *riconoscitori*

Modelli di calcolo e linguaggi

- Tuttavia, è possibile approcciarsi ai linguaggi in modo diverso
  - in un certo senso, duale
- possiamo progettare ***uno strumento capace di generare tutte (e sole) le parole di un linguaggio***
- un siffatto modello di calcolo è un *modello generativo* e prende il nome di *Grammatica*
- e in questa e nelle prossime lezioni studieremo le grammatiche
  - come sono definite
  - in che senso generano linguaggi
  - le proprietà che le caratterizzano
- e le relazioni fra grammatiche e macchine di Turing

Grammatiche

- Una grammatica è una quadrupla G = ⟨ V_T, V_N, P, S ⟩ dove
  - V_T è l'insieme dei simboli terminali (che corrisponde all'alfabeto Σ)
  - V_N è l'insieme dei simboli non terminali
  - S ∈ V_N è un particolare simbolo non terminale detto assioma
  - P è l'insieme delle produzioni, ossia, un insieme di coppie di parole $(\alpha, \beta) \in (V_T \cup V_N)^* \times (V_T \cup V_N)^*$ tali che α contiene almeno un simbolo in V_N
    - una produzione (α, β) viene indicata nella forma $\alpha \rightarrow \beta$
  - almeno una produzione è nella forma S → β
- Obiettivo di una grammatica è generare parole di V_T*
  - a partire dall'assioma
  - e applicando, una dopo l'altra, le produzioni in P
- Prima di capire come si fa a generare una parola mediante una grammatica, abbiamo bisogno di un po' di notazioni

Grammatiche

- Siano α, β due parole: con $\alpha\beta$ indichiamo la parola che è la concatenazione di α e β
  - ESEMPIO: se α =001 e β =abba, allora $\alpha\beta$ = 001abba
- Sia G = $\langle V_T, V_N$, P, S $\rangle$ una grammatica
- Siano x,y ∈ $(V_T \cup V_N)^*$: scriviamo x $\Rightarrow_G$ y e diciamo che y ***deriva direttamente*** (in G) da x se
  - x = $\alpha_1 \boldsymbol{\alpha} \alpha_2$, y =$\alpha_1 \boldsymbol{\beta} \alpha_2$con $\alpha_1, \alpha, \alpha_2, \beta \in (V_T \cup V_N)^*$, e
  - $\boldsymbol{\alpha} \rightarrow \boldsymbol{\beta}$ è una produzione in P
- Siano x,y ∈ $(V_T \cup V_N)^*$: scriviamo x $\Rightarrow_G^*$ y e diciamo che y ***deriva*** (in G) da x se esiste una sequenza di parole $x_1, x_2, \dots, x_n \in (V_T \cup V_N)$ tali che

$$x \Rightarrow_G x_1 \Rightarrow_G x_2 \dots \Rightarrow_G x_n \Rightarrow_G y$$

- Una parola y ∈ $V_T{}^*$ ***è generata da G*** se S $\Rightarrow_G^*$ y
- Una parola generata da G prende il nome di ***forma di frase***

Grammatiche

- Sia G = $\langle V_T, V_N$, P, S $\rangle$ una grammatica
- Il ***linguaggio generato da G***, che indichiamo con **L(G)**, è l'insieme delle parole generate da G, ossia

$$\mathbf{L(G) = \{ y \in V_T{}^* : S \Rightarrow_G^* y \}}$$

- Un altro paio di notazioni:
  - Se due o più produzioni hanno la stessa parte sinistra esse si possono raggruppare mediante il simbolo | (da leggere come ''oppure'')
    - ad esempio, le due produzioni S → a e S → b possono essere scritte nella forma S → a | b
  - ε rappresenta la parola vuota (ossia, la parola che non contiene alcun carattere)
  - una produzione del tipo $\alpha \rightarrow \varepsilon$ viene chiamata **ε-produzione**
  - Λ indica il linguaggio vuoto

ESEMPIO 1

- Sia G = ⟨ V_T, V_N, P, S ⟩ la grammatica tale che
  - V_T = { a,b }, V_N = { S, A, B}
  - P = { S → aSB, S → aB, B → b }
- In questo caso, il linguaggio generato da G è il linguaggio delle paroleche iniziano con un certo numero non nullo di caratteri a e terminano con lo stesso numero di caratteri b
- Vediamolo derivando qualche forma di frase
  - S → aB – che contiene una a e una b
  - S → aSB $\Rightarrow_G$ aSb $\Rightarrow_G$ aabb
  - S → aSB $\Rightarrow_G$ aaSBB $\Rightarrow_G$ aaSbB $\Rightarrow_G$aaaSBbB $\Rightarrow_G$ aaaabBbB $\Rightarrow_G$ aaaabBbb $\Rightarrow_G$ aaaabbbb
- Analogamente, per derivare la forma di frase aaabbb possiamo:
  - S → aSB → aaSBB → aaabBB → aaabbB → aaabbb
  - ma anche S → aSB → aSb → aaSBb → aaSbb → aaabbb
  - ossia, una forma di frase può essere derivata in molti modi diversi

ESEMPIO 1

- Sia $G = \langle V_T, V_N, P, S \rangle$ la grammatica tale che
  - $V_T = \{ a,b \}$, $V_N = \{ S, A, B \}$
  - $P = \{ S \rightarrow aSB, S \rightarrow aB, B \rightarrow b \}$
- Per dimostrare che il linguaggio generato da G è il linguaggio delle parole contenenti lo stesso numero di caratteri a e b, ossia, $L(G) = \{ a^n b^n : n \in \mathbb{N} \}$ occorre dimostrare che
  - 1) comunque si scelga $n \in \mathbb{N}$, la parola $a^n b^n$ può essere derivata da S
  - 2) che ogni parola derivata da S è nella forma $a^n b^n$ per qualche $n \in \mathbb{N}$
- 1) Comunque si scelga $n \in \mathbb{N}$, la parola $a^n b^n$ può essere derivata da S
  - comunque si scelga $n \in \mathbb{N}$,

$$S \rightarrow aSB \Rightarrow_G aaSBB \Rightarrow_G aaaSBBB \Rightarrow_G \ldots \text{(n-1 volte)} \Rightarrow_G a^{n-1}SB^{n-1} \Rightarrow_G a^nB^n$$
$$\Rightarrow_G a^n bB^{n-1} \Rightarrow_G a^n bbB^{n-2} \Rightarrow_G \ldots \Rightarrow_G a^n b^{n-1}B \Rightarrow_G a^n b^n$$

  - dunque, $\{ a^n b^n : n \in \mathbb{N} \} \subseteq L(G)$

ESEMPIO 1

- Sia $G = \langle V_T, V_N, P, S \rangle$ la grammatica tale che
  - $V_T = \{ a,b \}$, $V_N = \{ S, B\}$
  - $P = \{ S \rightarrow aSB, S \rightarrow aB, B \rightarrow b \}$
- 2) Ogni parola derivata da S è nella forma $a^n b^n$ per qualche $n \in \mathbb{N}$, infatti (informalmente):
  - 2.1) osserviamo che le parole nella parte destra delle due produzioni $S \rightarrow aSB$ e $S \rightarrow aB$ contengono lo stesso numero di caratteri a e di caratteri in {b,B}
  - 2.2) se inizialmente utilizziamo la produzione $S \rightarrow aB$, poiché aB contiene il non terminale B, dobbiamo poi utilizzare la produzione $B \rightarrow b$: dunque $S \Rightarrow_G ab$;
  - 2.3) se inizialmente utilizziamo la produzione $S \rightarrow aSB$ otteniamo una parola (contenente anche non terminali) dobbiamo poi applicare una produzione che parte da S oppure la produzione $B \rightarrow b$: in entrambi i casi, otteniamo una parola che contiene lo stesso numero di caratteri a e di caratteri in {b,B}
  - 2.4) infine, alla parola ottenuta al punto 2.3) applichiamo la produzione $B \rightarrow b$ per sostituire, una alla volta, le B con b e otteniamo una parola che contiene lo stesso numero di caratteri a e di caratteri in b
  - pertanto, $L(G) \subseteq \{ a^n b^n : n \in \mathbb{N} \}$

ESEMPIO 2

- La grammatica G = ⟨ V_T, V_N, P, S ⟩ tale che
  - V_T = { a,b }, V_N = { S, A }
  - P = { S → Ab, A → Sa }
- non genera alcuna parola (di lunghezza finita) in V_T
- pertanto, L(G) è il linguaggio vuoto:

$$L(G) = \Lambda$$

ESEMPIO 3

- Ci proponiamo ora di progettare una grammatica G = ⟨ V_T, V_N, P, S ⟩ tale che L(G) sia il linguaggio delle parole palindrome nell'alfabeto { a,b }
- La grammatica che serve allo scopo è la seguente:
  - V_T = { a,b }, V_N = { S }
  - P = { S → aSa, S → bSb, S → a, S → b, S → ε }
- Infatti, informalmente:
  - se inizialmente utilizziamo una delle produzioni S → a, S → b, S → ε otteniamo le parole a oppure b oppure la parola vuota che sono palindrome,
  - se inizialmente utilizziamo una delle produzioni S → aSa o S → bSb otteniamo una parola palindroma nell'alfabeto $V_T \cup V_N$, e quindi, induttivamente, segue che le parole generate da questa grammatica sono palindrome
- provare per credere

ESERCIZI

- 1) Modificare la grammatica dell'ESEMPIO 3 in modo tale che generi solo parole palindrome di lunghezza dispari
- 2) Modificare la grammatica dell'ESEMPIO 3 in modo tale che generi solo parole palindrome di lunghezza pari
- 4) Progettare una grammatica che generi tutte le parole binarie non vuote
- 5) Progettare una grammatica che generi l'insieme di parole

  $$\{ a^n b^m c^h : n, m, h \in \mathbb{N} \}$$
- 6) Progettare una grammatica che generi l'insieme di parole

  $$\{ a^n b^n c^n : n \in \mathbb{N}^+ \}$$
  - più complesso dei precedenti

Soluzione dell'esercizio 5)

- 5) Progettare una grammatica che generi il linguaggio
  $L = \{ a^n b^m c^h : n, m, h \in \mathbb{N} \}$
  - Una grammatica $G = \langle V_T, V_N, P, S \rangle$ che genera L è definita dagli insiemi $V_T = \{a, b, c\}$ e $V_N = \{S, A, B, C\}$ e dalle produzioni seguenti
    - $S \rightarrow ABC$,
    - $A \rightarrow aA \mid a \mid \varepsilon$,
    - $B \rightarrow bB \mid b \mid \varepsilon$,
    - $C \rightarrow cC \mid c \mid \varepsilon$
  - L è generato anche dalla grammatica definita dagli insiemi $V_T = \{a, b, c\}$ e $V_N = \{S, B, C\}$ e dalle produzioni seguenti
    - $S \rightarrow aS \mid B$,
    - $B \rightarrow bB \mid C$
    - $C \rightarrow cC \mid \varepsilon$

Soluzione dell'esercizio 6)

- 6)) Progettare una grammatica che generi il linguaggio
  $L = \{ a^n b^n c^n : n \in \mathbb{N}^+ \}$
  - Una grammatica $G = \langle V_T, V_N, P, S \rangle$ che genera L è definita dagli insiemi $V_T = \{a, b, c\}$ e $V_N = \{S, A, B\}$ e dalle produzioni seguenti
    - $S \rightarrow aBSc \mid abc$,
    - $Ba \rightarrow aB$,
    - $Bb \rightarrow bb$
- Verifichiamo per induzione che le parole di L sono generate da G:
  - base dell'induzione (n=1): poiché G contiene la produzione $S \rightarrow abc$ allora G genera la parola abc
  - ipotesi induttiva: supponiamo che $\mathbf{S \Longrightarrow_G^* a^i b^i c^i}$ per ogni $i \leq n$
  - dimostriamo che G genera la parola $a^{n+1}b^{n+1}c^{n+1}$:

$$S \rightarrow aB\mathbf{S}c \Longrightarrow_G^* aB\,\mathbf{a^n b^n c^n}\,c \Longrightarrow_G aaBa^{n-1}b^nc^{n+1} \Longrightarrow_G aaaBa^{n-2}b^nc^{n+1} \Longrightarrow_G \ldots \Longrightarrow_G a^{n+1}Bb^nc^{n+1} \Longrightarrow_G a^{n+1}b^{n+1}c^{n+1s}$$

- dove la derivazione rossa segue dall'ipotesi induttiva