Lezione 17 – grammatiche context-free

Lezione del 10/04/2025

G2 ⊆ G1: inclusione propria o impropria?

- Poiché le grammatiche di tipo 2 sono anche grammatiche di tipo 1, allora ogni linguaggio generato da una grammatica di tipo 2 è anche generato da una grammatica di tipo 1
- ossia, la classe G2 dei linguaggi generati da grammatiche di tipo 2 è un sottoinsieme della classe G1 dei linguaggi generati da grammatiche di tipo 1
- Ma cosa possiamo dire del viceversa?
- Ossia, sarà forse che la classe G1 dei linguaggi generati da grammatiche di tipo 1 è un sottoinsieme della classe G2 dei linguaggi generati da grammatiche di tipo 2?
  - così che G2 = G1
- Oppure sarà che le due classi sono distinte, cioè esiste un linguaggio in G1 che non appartiene a G2?
  - così che G2 ⊂ G1
- Per rispondere a queste domande ci viene in aiuto il prossimo lemma

Grammatiche di tipo 2: pumping lemma

- Il lemma seguente descrive una proprietà soddisfatta da tutti i linguaggi context-free
- **Pumping lemma per i linguaggi context-free (o lemma di Bar-Hillel)**: per ogni linguaggio context-free L esiste un intero $p_L > 0$ (dipendente esclusivamente da L) tale che per ogni parola $z \in L$ se $|z| \geq p_L$ allora esistono cinque parole u, v, w, x, y tali che
  1. z = uvwxy (z si può esprimere come concatenazione di u, v, w, x, y)
  2. $|vwx| \leq p_L$
  3. $|vx| \geq 1$ (v e x non possono essere entrambe vuote)
  4. uv^hwx^hy è in L per ogni $h \geq 0$.

- Il Lemma di Bar-Hiller stabilisce una *condizione necessaria* che deve essere soddisfatta da un linguaggio affinché esso sia di tipo 2
  - ossia, *ogni linguaggio context-free* soddisfa 1., 2., 3. e 4.
- ma tale condizione non è una condizione sufficiente

Grammatiche di tipo 2: pumping lemma

- Il Lemma di Bar-Hiller non è una *condizione sufficiente* per stabilire che un linguaggio è context-free
  - ossia, *esistono linguaggi* non context-free le cui parole soddisfano 1., 2., 3. e 4.

- per questa ragione, il lemma di Bar-Hiller si utilizza ''al negativo''
- ossia, per dimostrare che un linguaggio ***non*** è context-free

- E come si fa?
  - Si mostra che le sue parole non soddisfano il lemma perché:

  poiché ogni linguaggio context-free soddisfa il lemma, allora

  un linguaggio che non soddisfa il lemma non è context-free!

- E ora vediamo un esempio

Grammatiche di tipo 2: pumping lemma

- ESEMPIO: dimostriamo che il linguaggio $L_{a=b=c} = \{ a^n b^n c^n : n \geq 1 \}$ non è di tipo 2
  - supponiamo per assurdo che L sia context-free e sia $p_L > 0$ la costante definita nel lemma di Ben-Hiller
  - consideriamo la seguente parola in $L_{a=b=c}$: $z = a^{p_L} b^{p_L} c^{p_L}$
  - poiché $|z| = 3\,p_L > p_L$, per il lemma z può essere scritta nella forma uvwxy con $|vwx| \leq p_L$ e v e x non entrambe vuote
  - inoltre, il lemma ci dice che uv^hwx^hy è in *L* per ogni h≥ *0* – ***in particolare, uwy è in*** $\boldsymbol{L_{a=b=c}}$
  - Poiché $|vwx| \leq p_L$ e la 'a' più a destra e la 'c' più a sinistra di z distano p_L+1 posizioni, allora vwx può contenere al massimo due simboli distinti
    - ossia, vwx è un certo numero k di 'a' seguite da al più p_L-k 'b', oppure un certo numero k di 'b' seguite da al più p_L-k 'c',
  - allora, poiché v e x non sono entrambe vuote, v contiene almeno una 'a' o una 'b', o x contiene almeno una 'b' o una 'c'
  - allora, esiste almeno un carattere fra 'a', 'b' e 'c' che compare meno di p_L volte in uwy (quello o quelli in v e in x), e ne esiste almeno un altro che compare p_L volte (quello che non è in vwx).
  - dunque, poiché tutte le parole di L hanno lo stesso numero di caratteri 'a', 'b', 'c', **uwy non può appartenere a** $\boldsymbol{L_{a=b=c}}$: un assurdo. Quindi L non è context-free.

Grammatiche di tipo 2: chiusura

- Ci occupiamo ora di studiare se la proprietà di essere context-free si trasporta all'unione e all'intersezione di linguaggi context-free
- **Unione di linguaggi context-free**: se L_1 e L_2 sono due linguaggi context-free allora $L = L_1 \cup L_2$ è context-free
  - infatti: siano $G_1 = \langle V_T^1, V_N^1, P_1, S_1 \rangle$ e $G_2 = \langle V_T^2, V_N^2, P_2, S_2 \rangle$ le due grammatiche context-free tali che $L_1 = L(G_1)$ e $L_2 = L(G_2)$
  - allora la grammatica $G = \langle V_T, V_N, P, S \rangle$ tale che $V_T = V_T^1 \cup V_T^2$, $V_N = V_N^1 \cup V_N^2 \cup \{S\}$ e $P = P_1 \cup P_2 \cup \{S \rightarrow S_1, S \rightarrow S_2\}$ è context-free e genera L (ossia, L=L(G))
- **Intersezione di linguaggi context-free**: se L_1 e L_2 sono due linguaggi context-free non è detto che $L = L_1 \cap L_2$ sia context-free
  - infatti: abbiamo dimostrato che i linguaggi è $L_{a=b,c} = \{ a^n b^n c^m : n \geq 1 \land m \geq 1 \}$ e $L_{a,b=c} = \{ a^m b^n c^n : m \geq 1 \land n \geq 1 \}$ sono context-free
  - il linguaggio $L_{a=b=c} = \{ a^n b^n c^n : n \geq 1 \}$ è l'intersezione dei due linguaggi $L_{a=b,c}$ e $L_{a,b=c}$, ossia $L_{a=b,c} = \cap L_{a,b=c}$
  - abbiamo dimostrato che $L_{a=b=c}$ non è context-free

Grammatiche di tipo 2: chiusura

- Le due proprietà su unione e intersezione dei linguaggi context-free sono riassunte nei seguenti due teoremi

  TEOREMA G.7: l'insieme dei linguaggi context-free è chiusa rispetto all'unione

  TEOREMA G.8: l'insieme dei linguaggi context-free non è chiusa rispetto all'intersezione

- Inoltre, si può dimostrare il seguente teorema

  TEOREMA G.9: l'insieme dei linguaggi context-free non è chiusa rispetto al complemento

Grammatiche di tipo 2: automi a pila

- I linguaggi context-free sono un sottoinsieme (proprio) dei linguaggi di tipo 1 e, pertanto, già sappiamo che essi sono decidibili
- Ci occupiamo ora di mostrare che, in effetti, i linguaggi context-free sono decisi da un modello di calcolo *strettamente meno potente* della Macchina di Turing di tipo riconoscitore: l'Automa a pila (in breve, PDA – PushDown Automata)
  - *strettamente meno potente*: ossia, ogni PDA può essere simulato da una macchina di Turing, ma non viceversa
- Informalmente, un PDA consiste di:
  - una unità di controllo che, ad ogni istante, può trovarsi in uno *stato interno* appartenente ad un certo insieme Q che contiene, fra gli altri, lo stato particolare q_0 che fa partire la computazione e un sottoinsieme Q_F di stati che fanno terminare la computazione
  - una coppia di nastri semi-infiniti suddivisi in un infinito numero di celle, ciascuna delle quali, ad ogni istante, può essere vuota o contenere un simbolo appartenente a un alfabeto Σ (sul primo nastro) o a un alfabeto Γ (sul secondo nastro), e sui quali si muovono le testine di lettura/scrittura
  - a ogni istante, dipendentemente dallo stato interno e da ciò che è letto dalle testine, vengono eseguite le azioni specificate da una *funzione di transizione* δ

Grammatiche di tipo 2: automi a pila

- Dunque, sui due nastri vengono scritti caratteri appartenenti a insiemi differenti
  - in particolare, deve essere $\Sigma \cap \Gamma = \emptyset$
- in effetti, i due nastri sono utilizzati in modo differente:
  - il primo nastro è quello sul quale viene scritto l'input (una parola x in Σ^*), è un nastro di sola lettura e la testina ad esso associata può rimanere ferma o muoversi verso destra (*ma mai a sinistra*) – in particolare, una volta raggiunto l'ultimo carattere di x la testina sul primo nastro rimane ferma
  - il secondo nastro, contiene inizialmente un solo carattere speciale di Γ, in genere indicato con Z_0, e viene gestito con una politica LIFO (last in first out) come, appunto, le pile (e per questa ragione, d'ora in poi lo chiameremo ***pila***): la testina ad esso associata è *sempre* posizionata sul carattere più a destra cosicché per leggere il carattere immediatamente a sinistra del carattere più a destra è necessario cancellare quest'ultimo prima di spostare a sinistra la testina
- Abbiamo ora la notazione necessaria per definire formalmente un **PDA** come una settupla $\langle \Sigma , \Gamma , Z_0 , Q , Q_F , q_0 , \delta \rangle$

Grammatiche di tipo 2: automi a pila

- Analogamente alle macchina di Turing, possiamo definire lo stato di un PDA come una tripla $\langle q, x, \gamma \rangle$ che specifica lo stato interno in cui si trova il PDA, il contenuto del primo nastro e il contenuto del secondo nastro
  - non è necessario specificare la posizione delle testine perché la testina sul primo nastro è sempre posizionata sul carattere più a sinistra e la testina sul secondo nastro è sempre posizionata sul carattere più a destra
- La *funzione di transizione* δ ha un significato analogo all'insieme delle quintuple di una macchina di Turing
- Formalmente,

$$\delta: Q \times (\Sigma \cup \{\boldsymbol{\varepsilon}\}) \times \Gamma \longrightarrow \mathcal{P}\ (Q \times \Gamma^*)$$

  è una funzione parziale che associa a uno stato interno in Q, a un carattere in Σ o a nessun carattere in Σ (indicato con ε) e a un carattere in Γ un'azione da compiere scelta in un insieme di coppie (nuovo stato, parola in Γ^*)
  - $\mathcal{P}\ (Q \times \Gamma^*)$ è l'insieme delle parti di $Q \times \Gamma^*$
- Osserviamo che, *poiché l'azione da compiere deve essere scelta in un insieme*, quel che stiamo definendo è un **PDA non deterministico**

Grammatiche di tipo 2: automi a pila

- Specificatamente:
  - siano $a \in \Sigma$, $Z \in \Gamma$ e $\gamma \in \Gamma^*$ e q_1, $q_2 \in Q$
  - sia $(ax, \beta Z, q_1)$ lo stato del PDA, con $x \in \Sigma^*$ e $\beta \in \Gamma^*$
  - allora si sceglie una coppia $(q_2, \gamma) \in$ **$\delta(q_1, a, Z)$** $\cup$ **$\delta(q_1, \varepsilon, Z)$**, per modificare lo stato del PDA come di seguito descritto
  - se è stata scelta $(q_2, \gamma) \in$ **$\delta(q_1, a, Z)$** allora
    - la testina sul primo nastro viene mossa a destra di una posizione dopo aver cancellato il carattere letto,
    - il carattere Z sulla pila viene cancellato e, se $\gamma \neq \varepsilon$ sulla pila viene scritta (un carattere per cella) la parola γ (in ogni caso la testina viene posizionata sul carattere più a destra sul nastro pila)
    - infine, il PDA entra nello stato q_2
    - ossia, il nuovo stato dell'automa è $(x, \beta\gamma, q_2)$

Grammatiche di tipo 2: automi a pila

- Specificatamente:
  - siano $a \in \Sigma$, $Z \in \Gamma$ e $\gamma \in \Gamma^*$ e $q_1, q_2 \in Q$
  - sia $(ax, \beta Z, q_1)$ lo stato del PDA, con $x \in \Sigma^*$ e $\beta \in \Gamma^*$
  - allora si sceglie una coppia $(q_2, \gamma) \in \boldsymbol{\delta(q_1, a, Z)} \cup \boldsymbol{\delta(q_1, \varepsilon, Z)}$, per modificare lo stato del PDA come di seguito descritto
  - se è stata scelta $(q_2, \gamma) \in \boldsymbol{\delta(q_1, \varepsilon, Z)}$ allora
    - la testina sul primo nastro rimane ferma,
    - il carattere Z sulla pila viene cancellato e, se $\gamma \neq \varepsilon$, sulla pila viene scritto (un carattere per cella) la parola γ (in ogni caso la testina viene posizionata sul carattere più a destra sul nastro pila)
    - infine, il PDA entra nello stato q_2
    - ossia, il nuovo stato dell'automa è $(ax, \beta\gamma, q_2)$

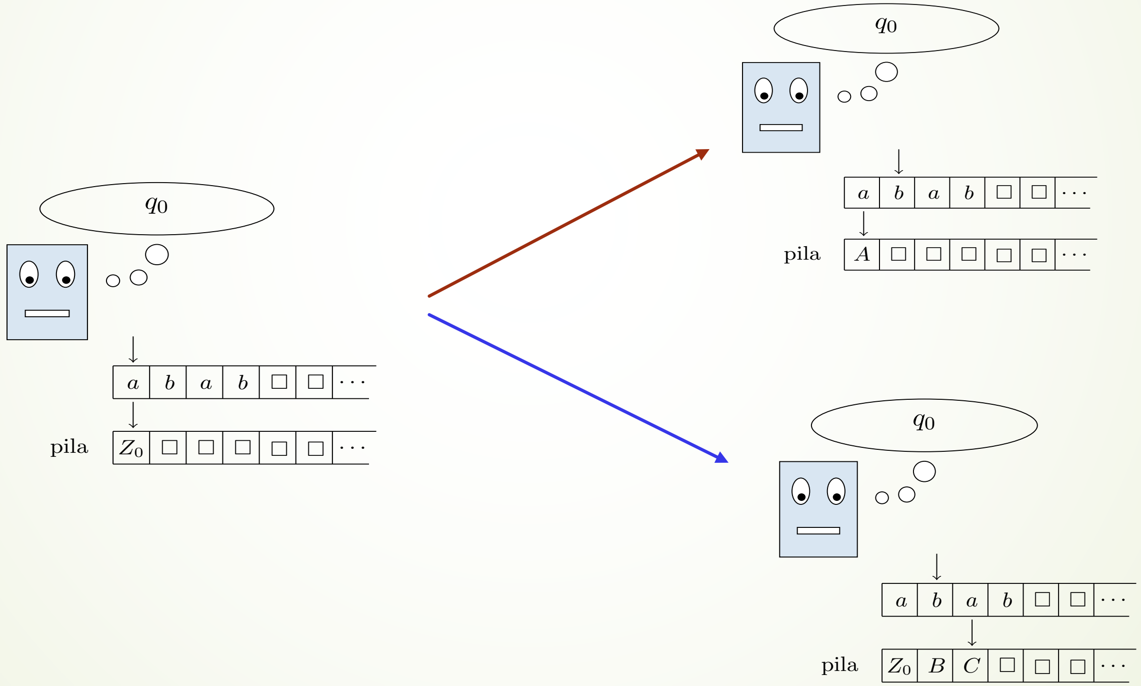
Grammatiche di tipo 2: automi a pila

- Specificatamente:
  - siano $a \in \Sigma$, $Z \in \Gamma$ e $\gamma \in \Gamma^*$ e q_1, $q_2 \in Q$
  - sia $(ax, \beta Z, q_1)$ lo stato del PDA, con $x \in \Sigma^*$ e $\beta \in \Gamma^*$
  - allora si sceglie una coppia $(q_2, \gamma) \in \boldsymbol{\delta(q_1, a, Z)} \cup \boldsymbol{\delta(q_1, \varepsilon, Z)}$, e si modifica lo stato del PDA come abbiamo descritto

- Quindi, la scelta di un elemento (q_2, γ) in $\delta(q_1, a, Z) \cup \delta(q_1, \varepsilon, Z)$ determina la transizione ($\vdash$) del PDA da uno stato $\langle q_1, ax, Z\beta \rangle$ a un nuovo stato:
  - se $(q_2, \gamma) \in \delta(q_1, a, Z)$ il PDA entra nello stato $\langle q_2, x, \beta\gamma \rangle$, ossia esegue la transizione $\langle q_1, ax, \beta Z \rangle \vdash \langle q_2, x, \beta\gamma \rangle$,
  - se $(q_2, \gamma) \in \delta(q_1, \varepsilon, Z)$ il PDA entra nello stato $\langle q_2, x, \beta\gamma \rangle$, ossia esegue la transizione $\langle q_1, ax, \beta Z \rangle \vdash \langle q_2, ax, \beta\gamma \rangle$
- Osserviamo che $\gamma \in \Gamma^*$, ossia può accadere che $\gamma = \varepsilon$
  - in questo caso, il simbolo in cima alla pila viene rimosso e la testina si sposta di una posizione a sinistra

Grammatiche di tipo 2: automi a pila

- **ESEMPIO**: nel caso in cui $\delta(q_0, a, Z_0) = \{(q_0, A), (q_0, Z_0 B)\}$, l'automa può eseguire una transizione dallo stato iniziale $\langle q_0, abab, Z_0 \rangle$ o allo stato **$\langle q_0, bab, A \rangle$** o allo stato **$\langle q_0, bab, Z_0BC \rangle$**



Grammatiche di tipo 2: automi a pila

OSSERVAZIONI:

- la parola vuota ε ha un diverso significato quando usata come argomento della funzione di transizione o come risultato della sua applicazione
  - quando ε compare nel risultato della applicazione di δ, ad esempio quando $(q_2, \varepsilon) \in \delta(q_1, a, Z)$: in questo caso, l'uso di ε fa sì che venga cancellato il simbolo in cima alla pila (senza che esso venga sovrascritto da altri simboli)
  - quando ε compare come argomento della funzione di transizione, ad esempio quando $\delta(q_1, \varepsilon, Z) = \{ \dots \}$: in questo caso, la parola ε sta a significare che qualunque transizione all'interno dell'insieme può essere eseguita ogniqualvolta il carattere in cima alla pila è Z *indipendentemente da cosa viene letto sul nastro* (e proprio perché il carattere sul nastro non viene utilizzato la testina su di esso non si muove a destra)
- Le transizioni del tipo $\delta(q_1, \varepsilon, Z) = \{ \dots \}$ vengono dette **ε-regole**
  - come stiamo per vedere la presenza di ε-regole ha importanti conseguenze sulla terminazione delle computazioni di un automa a pila

Grammatiche di tipo 2: automi a pila

OSSERVAZIONI:

- non esistono transizioni nella forma $\delta(q_1, \square, Z) = \{ \dots \}$ ossia transizioni che vengono eseguite quando sul nastro viene letto il blank
- conseguentemente, se la funzione di transizione dell'automa non contiene ε-regole allora tutte le computazioni di quell'automa hanno lunghezza limitata dalla lunghezza dell'input
  - ossia, una volta che tutti i caratteri dell'input sono stati letti la computazione termina
- ma se la funzione di transizione di un automa contiene ε-regole (che non ''consumano'' caratteri dell'input) non esiste più un limite sulla lunghezza delle sue computazioni
  - che possono anche non terminare!
- Per questo parleremo soltanto di accettazione di un linguaggio da parte di un automa a pila, e non di decisione
  - e per farlo dobbiamo prima capire quando un PDA accetta una parola

Grammatiche di tipo 2: automi a pila

- Sono state definite due diverse modalità di accettazione per i PDA: sia $\mathcal{M} = \langle \Sigma , \Gamma , Z_0 , Q , Q_F , q_0 , \delta \rangle$ un automa a pila e sia $x \in \Sigma^*$
  - modalità di **accettazione per stato finale**: x è accettato da $\mathcal{M}$ se esiste una sequenza di transizioni dallo stato $\langle q_0, x, Z_0 \rangle$ a uno stato $\langle q_F, \varepsilon, \gamma \rangle$ per qualche $\gamma \in \Gamma^*$ e $q_F \in Q_F$; in questo caso, per ogni $\gamma \in \Gamma^*$ e $q_F \in Q_F$, **$\langle q_F, \varepsilon, \gamma \rangle$ è una configurazione di accettazione**
  - modalità di **accettazione per pila vuota**: x è accettato da $\mathcal{M}$ se esiste una sequenza di transizioni dallo stato $\langle q_0, x, Z_0 \rangle$ a uno stato $\langle q, \varepsilon, \varepsilon \rangle$ per qualche $q \in Q$; in questo caso, per ogni $q \in Q$, **$\langle q, \varepsilon, \varepsilon \rangle$ è una configurazione di accettazione**
- In corrispondenza a ciascuna delle due modalità, sono definiti
  - il *linguaggio accettato da $\mathcal{M}$ nella modalità di accettazione per stato finale*, indicato come **L($\mathcal{M}$)**
  - il *linguaggio accettato da $\mathcal{M}$ nella modalità di accettazione per pila vuota*, indicato come **N($\mathcal{M}$)**

Grammatiche di tipo 2: automi a pila

- Le due modalità di accettazione sono, sostanzialmente, equivalenti, come specificato nel seguente teorema

TEOREMA G.10: per ogni linguaggio L, esiste un PDA M che accetta L per pila vuota se e soltanto se esiste un PDA M' che accetta L per stato finale

- Come conseguenza del precedente teorema, possiamo parlare di ***insieme dei linguaggi accettati da automi a pila*** indipendentemente dalla modalità di accettazione

- Infine, il prossimo teorema mostra che tale insieme coincide con l'insieme dei linguaggi context-free

TEOREMA G.11: un linguaggio L è context-free se e soltanto se esiste un PDA M che accetta L