Fondamenti di Informatica — Esercizi

Università di Verona
Imbriani Paolo - VR500437
Professor Isabella Mastroeni

January 16, 2026

Contents

1 Grammatiche CF

Esercizio 1.25

Dimostrare che il seguente linguaggio sia context-free:

$$L = \{0^n 1^m 0^{n+m} \mid n, m \in \mathbb{N}\}$$

Soluzione:

Proof. Dimostriamo la seconda tesi per induzione su n.
Base: Sia $n = 2$. L'unica derivazione di lunghezza 2 è $S \to A \to \varepsilon$ dove $\varepsilon \in L$.
Passo induttivo: Supponiamo che la tesi sia vera per ogni derivazione di lunghezza minore o uguale a n.

$$\forall k \in \mathbb{N} \;.\; k \leq n : S \Rightarrow_k x \implies x \in L \; con \; x = 0^i 1^j 0^{i+j} \; con \; i, j \geq 0$$

Scomponiamo la derivazione a partire dalle prime produzioni. Dimostriamo che vale per S $\Rightarrow_{n+1} x$ con $x \in \{0, 1\}^*$. L'ultima produzione sarà sicuramente $A \to \varepsilon$ ovvero:

$$\text{[COND 1]} \quad S \Rightarrow_{n+1} x \equiv \overbrace{S \Rightarrow_n yAz \to y\varepsilon z}^{n+1} = yz = x$$

Osservando $COND1$ notiamoc he per come è fatta la grammatica y e z possono essere scritte come:

- $y = 0^h 1^k$ e $z = 0^l$ con $h, k, l \geq 0$

$$S \Rightarrow_n yAz \equiv S \Rightarrow_{n-1} \hat{y}A\hat{z} \to \hat{y}10\hat{z} = yz$$

- $y = 0^h$ e $z = 0^l$ con $h, l \geq 0$

Consideriamo il primo caso. Allora possiamo prendere la de

Esercizio 1.18

Dimostrare che il seguente linguaggio sia context-free:

$$L = \{0^{2n} 1 0^{n+m} \mid m, n \geq 0\}$$

Soluzione: Intutivamente sappiamo che il primo gruppo di zeri e il secondo gruppo di zeri sono legati da una relazione di dipendenza. Per ogni 2 zeri nel primo gruppo, c'è un solo 0 nel secondo gruppo. Tuttavia il secondo gruppo di zeri ha un numero variabile di zeri in più (m).

Possiamo quindi costruire una grammatica che generi il linguaggio come segue:

$$S \to 00S0 \mid S0 \mid 1$$

Dimostriamo che tale grammatica genera il linguaggio L. Abbiamo due tesi da dimostrare:

1. $x \in L \wedge x = 0^{2i}10^{i+j}$ con $i, j \geq 0 \implies S \Rightarrow_* x$ per induzione su $|x|$.
2. $S \Rightarrow_n x \implies x \in L \wedge x = 0^{2i}10^{i+j}$ con $i, j \geq 0$ per induzione su n.

Proof. 1. Dimostriamo la prima tesi per induzione su $|x|$.
Base: $|x| = 1 \implies x = 1$ con $i = 0, j = 0$. Allora $S \Rightarrow 1$.
Passo induttivo: Supponiamo che la tesi sia vera per ogni stringa di lunghezza minore o uguale a n.

$$\forall x \in \Sigma^* \,.\, |x| < n,\ x \in L \wedge x = 0^{2i}10^{i+j} \implies S \Rightarrow_* x$$

Dimostriamo che vale per $|x| = n + 1$. Il caso $i = j = 0$ lo abbiamo già visto nella base. Supponiamo $i > 0$. Allora x può essere scritto come $x = 00x'0$ con $x' = 0^{2(i-1)}10^{(i-1)+j}$. Inoltre abbiamo il caso in cui $j > 1$ e $i = 0$ in cui x può essere scritto come $x = y0$ con $y = 10^{j-1}$. In entrambi i casi abbiamo che $|x'| < |x| = n + 1$, quindi possiamo applicare l'ipotesi induttiva e ottenere che esiste una derivazione $S \Rightarrow_* x'$. Vediamo i due casi:

- Sia $x' = 0^{2(i-1)}10^{(i-1)+j}$ con $i > 1$. Per come è fatta la grammatica abbiamo che

$$\begin{aligned} S \Rightarrow_* 0^{2(i-1)}10^{(i-1)+j} \equiv S &\Rightarrow_* 0^{2(i-2)}10^{(i-2)+j} \\ &\to 00(0^{2(i-2)}10^{(i-2)+j})0 \\ &\to 0^{2(i-1)}10^{(i-1)+j} = x' \end{aligned}$$

  Ma allora possiamo scrivere la derivazione:

$$\begin{aligned} S &\Rightarrow_* 0^{2(i-2)}10^{(i-2)+j} \\ &\to 00(0^{2(i-1)}10^{(i-1)+j})0 \\ &\to 0^{2i}10^{i+j} = x \end{aligned}$$

- Sia $x' = 10^{j-1}$ con $j > 1$. Per come è fatta la grammatica abbiamo che

$$\begin{aligned} S \Rightarrow_* 10^{j-1} \equiv S &\Rightarrow_* 10^{j-2} \\ &\to S0^{j-2} \\ &\to 10^{j-1} = x' \end{aligned}$$

Ma allora possiamo scrivere la derivazione:

$$\begin{aligned} S &\Rightarrow_* S0^{j-2} \\ &\rightarrow S00^{j-2} \\ &\rightarrow S0^{j-1} \\ &\rightarrow S00^{j-1} \\ &\rightarrow 10^j = x \qquad \square \end{aligned}$$

Proof. 2. Dimostriamo la seconda tesi per induzione su n.
Base: $n = 1$. L'unica derivazione di lunghezza 1 è $S \rightarrow 1$ e $1 \in L$.
Passo induttivo: Supponiamo che la tesi sia vera per ogni derivazione di lunghezza minore o uguale a n.

$$\forall k \in \mathbb{N} \,.\, k \leq n : S \Rightarrow_k x \implies x \in L \wedge x = 0^{2i}10^{i+j} \; con \; i, j \geq 0$$

Scomponiamo la derivazione a partire dalle prime produzioni. Dimostriamo che vale per S $\Rightarrow_{n+1} x$ con $x \in \{0,1\}^*$. Una tale derivazione può iniziare in due modi, con le produzioni $S \rightarrow 00S0$ oppure $S \rightarrow S0$. Quindi so che $\exists x' \in \{0,1\}^*$ tale che:

$$\text{[COND 1.1]} \quad S \Rightarrow_{n+1} x \equiv \overbrace{S \rightarrow S0 \Rightarrow_n \underbrace{x'}_{0^*1} 0 = x}^{n+1}$$

$$\text{[COND 1.2]} \quad S \Rightarrow_{n+1} x \equiv \overbrace{S \rightarrow 00S0 \Rightarrow_n 00 \underbrace{x'}_{0^*10^*} 0 = x}^{n+1}$$

Consideriamo la condizione 1.1. Per l'ipotesi induttiva sappiamo che $S \Rightarrow_n x'$ con $x' \in L$ e $x' = 0^{2i}10^{i+j}$ con $i, j \geq 0$. Allora possiamo scrivere:

$$x = x'0 = 0^{2i}10^{i+j}0 = 0^{2i}10^{i+(j+1)} \in L \text{ (per def) con } i, j+1 \geq 0$$

Consideriamo la condizione 1.2. Per l'ipotesi induttiva sappiamo che $S \Rightarrow_n x'$ con $x' \in L$ e $x' = 0^{2i}10^{i+j}$ con $i, j \geq 0$. Allora possiamo scrivere:

$$x = 00x'0 = 00(0^{2i}10^{i+j})0 = 0^{2(i+2)}10^{(i+1)+j} \in L \text{ (per def) con } i+1, j \geq 0 \quad \square$$

Esercizio 1.26

Dimostrare che la seguente famiglia di linguaggi sia context-free:

$$L_m = \{b^{mn}c^{4n} \mid n \in \mathbb{N}\},\ \Sigma = \{a, b, c\}$$

Soluzione: Consideriamo il linguaggio:

$$L_0 = \{c^{4n} \mid n \in \mathbb{N}\}$$

L_0 contiene una sequenza di c multipla di 4, incluso la stringa vuota. La grammatica che lo genera è la seguente:

$$S \to ccccS \mid \epsilon$$

Notiamo che questo linguaggio è regolare (e quindi anche CF) e possiamo vedere L_m come concatenazione di due linguaggi:

$$L_{\alpha_m} = \{b^{mn} \mid n \in \mathbb{N}\}$$

e

$$L_0 = L_\beta = \{c^{4n} \mid n \in \mathbb{N}\} \quad \text{che sappiamo essere regolare (e quindi CF)}$$

Ora, L_{α_m} sappiamo che per:

- $m = 0$ è il linguaggio $\{\epsilon\}$ che è regolare (e quindi CF)
- $m = 1$ è il linguaggio $\{b^n \mid n \in \mathbb{N}\}$ che è regolare (e quindi CF)

$L_{\alpha_{m>1}}$ è il linguaggio costituito da tutte le stringhe di b con lunghezza multipla di m. L'espressione regolare che lo descrive è la seguente:

$$(b^m)^*$$

Di conseguenza se

$$L_m = L_{\alpha_m} \cdot L_\beta$$

Allora possiamo concludere che L_m è context-free per ogni $m \geq 0$, essendo la concatenazione di due linguaggi CF. Possiamo inoltre dire che per ogni m esiste una grammatica che genera L_m:

$$S \to b^m S \mid \varepsilon$$

Esercizio 1.25

Dimostrare che il seguente linguaggio sia context-free:

$$L = \{x \in \{0, 1\}^* \mid |x|_0 > |x|_1\}$$

Soluzione: Il linguaggio L contiene tutte le stringhe composte da 0 e 1 in cui il numero di 0 è strettamente maggiore del numero di 1. Alcuni esempi sono $L = \{000, 0001, 0010, 0100, 1000, 0011, 0101, 0110, 1001, 1010, 1100, 00011\}$.

Possiamo costruire una grammatica che genera il linguaggio come segue:

$$S \to A0A \quad A \to A1A0A \mid A0A1A \mid A0A \mid 0 \mid \varepsilon$$

Dimostriamo che tale grammatica genera il linguaggio L. Abbiamo due tesi da dimostrare:

1. $x \in L \wedge |x|_0 > |x|_1 \implies S \Rightarrow_* x$ per induzione su $|x|$.

2. $S \Rightarrow_n x \implies x \in L \wedge |x|_0 > |x|_1$ per induzione su n.

Proof. 1. Dimostriamo la prima tesi per induzione su $|x|$.
Base: $|x| = 1 \implies x = 0$. Allora $S \Rightarrow 0$.
Passo induttivo: Supponiamo che la tesi sia vera per ogni stringa di lunghezza minore o uguale a n.

$$\forall x \in \Sigma^* \,.\, |x| \leq n,\ x \in L \wedge |x|_0 > |x|_1 \implies S \Rightarrow_* x$$

Dimostriamo che vale per $|x| = n + 1$. Ad ogni produzione, devo sempre mantenere il vantaggio di 0 su 1. In ogni produzione in cui aggiungo 1, devo per forza aggiungere uno 0. Se $x \in L$ diversa dalla base (0), allora esiste la stringa in eccesso 0 dentro x quindi $\exists v, w \in \{0, 1\}^*$ tali che $x = v0w$. Per come è fatta la grammatica possiamo riordinare tutte le produzioni in modo da generare prima tutti i simboli della stringa (alternati dal simbolo A) e poi sostituire gli A con le stringhe vuote o con zeri in base all'esigenza. Prendiamo $x' = vw$, allora $|x'|_0 = |x|_0 - 1$ e $|x'|_1 = |x|_1$ quindi $|x'| < |x| = n + 1$ e quindi possiamo applicare l'ipotesi induttiva, ovvero esiste $S \Rightarrow_* x'$. Il passo iniziale ci permette di sapere con certezza che verranno sempre create stringhe con almeno uno 0 in più rispetto agli 1. Per quanto detto, questa derivazione può essere riscritta come: (dove $|v| = h$ e $|w| = k$ e v_i e w_i sono i simboli di v e w rispettivamente):

$$\begin{aligned} &A \Rightarrow_* x' = vw \equiv \\ &A \Rightarrow_* Av_1A \dots Av_hAw_1A \dots Aw_kA \\ &\quad \Rightarrow_{h+k+1} v_1 \dots v_h w_1 \dots w_k = vw = x' \end{aligned}$$

Sappiamo che A può generare stringhe dove il numero di 0 è uguale al numero di 1 che hanno al massimo $|x|_0 = |x|_1$. Ma con il passo iniziale S manteniamo sempre un 0 in più. e quindi possiamo costruire:

$$\begin{aligned} S \to A0A &\Rightarrow_* Av_1A \dots Av_h0w_1A \dots Aw_kA \\ &\Rightarrow_{h+j} v_1 \dots v_h0w_1 \dots w_k = v0w = x \quad \square \end{aligned}$$