Fondamenti di Informatica — Esercizi

Università di Verona
Imbriani Paolo - VR500437
Prof.ssa Isabella Mastroeni

February 3, 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:

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

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

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

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 $S \to 00S0$ oppure $S \to S0$. Quindi so che $\exists x' \in \{0,1\}^*$ tale che:

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

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

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:

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

2 Insiemi produttivi

Esercizio 1

Dimostrare che l'insieme

$$A = \{x \mid W_x = REC \implies \varphi_x(x^2)\downarrow\}$$

oppure

$$A = \{x \mid W_x \neq REC \vee \varphi_x(x^2)\downarrow\}$$

è produttivo.

Il complemento di A è:

$$\overline{A} = \{x \mid W_x = REC \wedge \varphi_x(x^2)\uparrow\}$$

Notiamo come i due insiemi siano entrambi produttivi essendo che in A abbiamo

$$W_x \neq REC$$

non è possibile decidere se un insieme non è ricorsivo. La stessa cosa per $\overline{A}$ essendo che non è possibile decidere se

$$\varphi_x(x^2)\uparrow$$

Per dimostrare che un insieme è produttivo dobbiamo dimostrare che $\overline{K} \preceq A$ o $K \preceq \overline{A}$. Definiamo la funzione parziale ricorsiva:

$$\psi(x, y) = \begin{cases} 1 & x \in K \wedge y \neq x^2 \\ \uparrow & \text{altrimenti} \end{cases}$$

Costruiamo l'algoritmo per ψ:

```
input(x,y)
if y != x^2:
    while true
costruisci phi_x
while true
    esegui next step di phi_x(x)
    if termina: return 1
```

Per il teorema di s-m-n esiste una funzione totale e calcolabile f tale che:

$$\varphi_{g(x)}(y) = \psi(x, y)$$

Sappiamo che se $x \in \overline{K}$ allora $\psi(x,y)\uparrow$ per ogni y.

$$\begin{aligned}
x \in K &\implies \psi(x,y)\downarrow \iff y \neq x^2 \\
&\implies \varphi_{g(x)}(y)\downarrow \iff y \neq g(x)^2 \\
&\implies W_{g(x)} = \mathbb{N}/\mathbb{N}^2 = \text{REC} \implies g(x) \notin A
\end{aligned}$$

$$\begin{aligned}
x \notin K &\implies \forall y \,.\, \psi(x,y)\uparrow \\
&\implies \forall y \,.\, \varphi_{g(x)}(y)\uparrow \\
&\implies \varphi_{g(x)}(g(x)^2)\uparrow \\
&\implies W_{g(x)} = \varnothing \implies g(x) \in A
\end{aligned}$$

Ora dimostriamo che $\overline{A}$ è produttivo. Per dimostrare che $\overline{A}$ è produttivo dobbiamo dimostrare che

$$\overline{K} \preceq \overline{A} \text{ o } K \preceq A$$

Quindi se $x \in K$ allora $g(x) \in A$. Definiamo la funzione parziale ricorsiva:

$$\psi(x,y) = \begin{cases} 1 & \varphi_x(x) \text{ NON termina in meno di } y \text{ passi} \land y \in 2\mathbb{N}+1 \\ \uparrow & y = 1 \\ \uparrow & \text{altrimenti} \end{cases}$$

Costruiamo l'algoritmo per ψ:

```
input(x,y)
if y == 1:
    while true
if y % 2 == 0:
    while true
costruisci phi_x
for 0 to y-1
    esegui next step di phi_x(x)
    if termina
        while true
return 1
```

Per il teorema di s-m-n esiste una funzione totale e calcolabile g tale che:

$$\varphi_{g(x)}(y) = \psi(x,y)$$

Sappiamo che

$$
\begin{aligned}
x \notin K &\implies \psi(x,y)\downarrow \iff y \in 2\mathbb{N}+1 \setminus \{1\} \\
&\implies \forall y \in 2\mathbb{N}+1\setminus\{1\} \,.\, \varphi_{g(x)}(y)\downarrow \\
&\implies W_x = 2\mathbb{N}+1\setminus\{1\} \in REC \land g(x)^2 \notin W_x \iff g(x)=1 \\
&\implies g(x) \notin A
\end{aligned}
$$

$$
\begin{aligned}
x \in K &\implies W_x = [0, n-1] \cap 2\mathbb{N}+1 \setminus \{1\} \\
&\implies \varphi_{g(x)}(y)\downarrow \text{ in n passi} \\
&\implies W_{g(x)} = [0, n-1] \cap 2\mathbb{N}+1 \setminus \{1\} \notin REC \\
&\implies g(x) \in A
\end{aligned}
$$

3 Successioni

Esercizio 1

Definire una successione di insiemi ricorsivi X_n tale che

$$\bigcup_n X_n = k \leftarrow \text{ creativo}$$

- Vanno definiti X_n al valore di n
- X_n sono ricorsivi
- Dimostrare che $\bigcup_n X_n = k$

$$k = \{x \mid \varphi_x(x)\downarrow\}$$

Sappiamo che $y \in \bigcup_n X_n$ corrisponde a $\exists n. y \in X_n$ e che $\downarrow$ è un $\exists n$ tale che per $\varphi_x(x)$ termina in n passi.

$$X_n = \{x \mid \varphi_x(x) \text{ termina in } n \text{ passi}\}$$

Una volta aver mostrato questo insieme le cose da fare sono due:

- X_n è ricorsivo
- $\bigcup_n X_n = k$

Dimostriamo che X_n è ricorsivo. Dobbiiamo costruire l'agloritmo per la funzione caratteristica di X_n.

$$f_n(x) = \begin{cases} 1 & \text{se } x \in X_n \\ 0 & \text{se } x \notin X_n \end{cases}$$

L'algoritmo è il seguente:

```
input(x)
costruisci phi_x
for y to n-1
    esegui un passo di phi_x(x)
    if termina return 0
esegui un passo di phi_x(x)
if termina return 1
else return 0
```

Questo è l'algoritmo per f_n che è totale quindi X_n è ricorsivo. Ora dimostriamo che $\bigcup_n X_n = k$.

$$\begin{aligned}\bigcup_n X_n &= \bigcup_n \{x \mid \varphi_x x \text{ termina in } n \text{ passi}\} \\ &= \{x \mid \exists n \,.\, \varphi_x(x) \text{ termina in } n \text{ passi}\} \\ &= \{x \mid \varphi_x(x) \downarrow\} = k\end{aligned}$$

Esercizio 2

Definire una successione di insiemi ricorsivi X_n tale che

$$\bigcap_n X_n = \overline{k} \leftarrow \text{ produttivo}$$

- Definite X_n al variare di n
- X_n sono ricorsivi
- Dimostrare che $\bigcap_n X_n = \overline{k}$

$$\bigcap_n = \forall n$$

$$\overline{k} = \{x \mid \varphi_x(x) \uparrow\} = \{x \mid \forall n \,.\, \varphi_x(x) \text{ non termina in n passi}\}$$

- Definiamo:

$$X_n = \{x \mid \varphi_x(x) \text{ non termina in n passi}\}$$

Dimostriamo che $f_n(x) = \begin{cases} 1 & \text{se } x \in X_n \\ 0 & \text{se } x \notin X_n \end{cases}$ è ricorsiva. Costruiamo l'algoritmo per f_n:

```
input(x)
costruisci phi_x
for y to n-1
    esegui un passo di phi_x(x)
    if termina return 1
esegui un passo di phi_x(x)
if termina return 0
else return 1
```

- $$\begin{aligned}\bigcap_n X_n &= \bigcap_n \{x \mid \varphi_x(x) \text{ non termina in n passi}\} \\ &= \{x \mid \forall n \,.\, \varphi_x(x) \text{ non termina in n passi}\} \\ &= \{x \mid \varphi_x(x) \uparrow\} = \overline{k}\end{aligned}$$

Esercizio 3

Definire una successione di insiemi ricorsivi X_n tale che

$$A = \{x \mid \bigcap_n \overline{X_n} \preceq W_x\}$$

- Definire X_n al variare di n
- X_n sono ricorsivi
- Dimostrare che $A = \{x \mid \bigcap_n \overline{X_n} \preceq W_x\}$
- Dimostrare che A è produttivo

Capiamo cosa significa $\bigcap_n \overline{X_n} \preceq W_x$. W_x è sempre re per definizione. Mentre $\bigcap_n \overline{X_n}$ non è produttivo.

$$k \preceq W_x$$

Vuol dire che W_x è re ma anche creativo.

- Definiamo:

$$\overline{X_n} = \{x \mid \varphi_x(x) \text{ termina in n passi}\} \quad X_n = \{x \mid \varphi_x(x) \text{ non termina in n passi}\}$$

- $\bigcup_n \overline{X_n} = k$ (esercizio 1)
- $A = \{x \mid k \preceq W_x\}$
- A produttivo se $\overline{k} \preceq A$.

$$\begin{aligned}x \in k &\implies g(x) \notin A \text{ ovvero } W_{g(x)} \text{ non creativo} \rightsquigarrow \text{Ricorsivo} \\ x \notin k &\implies g(x) \in A \text{ ovvero } W_{g(x)} \text{ creativo} \rightsquigarrow \text{K (creativo)}\end{aligned}$$

$$\psi(x, y) = \begin{cases} 1 & x \in k \vee y \in k \\ \uparrow & \text{se non termina} \end{cases}$$

Con ψ parziale ricorsiva.

```
input(x,y)
costruisci phi_x, phi_y
while true
    esegui next step di phi_x(x)
    if termina: return 1
    esegui next step di phi_y(y)
    if termina: return 1
```

Per il teorema di s-m-n esiste una funzione totale e calcolabile g tale che:

$$\varphi_{g(x)}(y) = \psi(x,y)$$

$$\begin{aligned} x \in k &\implies \forall y \,.\, \psi(x,y) \downarrow \\ &\implies \forall y \,.\, \varphi_{g(x)}(y) \downarrow \\ &\implies W_{g(x)} = \mathbb{N} \\ &\implies g(x) \notin A \end{aligned}$$

$$\begin{aligned} x \notin k &\implies y \in k \iff \psi(x,y) \uparrow \\ &\implies y \in k \iff \varphi_{g(x)}(y) \uparrow \\ &\implies W_{g(x)} = k \\ &\implies g(x) \in A \end{aligned}$$

Esercizio 3

Definire una successione di insiemi ricorsivi X_n tale che

$$C = \{x \mid \bigcap_n X_n \preceq W_x\} \text{ sia ricorsivo}$$

Essendo che per definizione W_x è sempre RE allora $\overline{k} \preceq W_x$ è sempre falso. Basta trovare una successione di insiemi ricorsivi t.c he $\bigcap_n X_n = \overline{k}$ (esercizio 2). Per dimostrare che C è ricorsivo basta notare che possiamo ricondurci al problema di appartenenza a $\overline{k}$ che è ricorsivo.

$$C = \{x \mid \bigcap_n X_n \preceq W_x\} = \{x \mid \overline{k} \preceq W_x\} = \emptyset$$

Esercizio 4

Successione di funzioni parziali ricorsive ψ_n tali che $dom(\psi_n)$ sono re non completi (ricorsivi) e che

$$\bigcup_n dom(\psi_n) = \{x \mid 2x \in W_x\}$$

Sappiamo che $\bigcup_n$ corrisponde a $\exists n$.

$$\{x \mid 2x \in W_x\} = \{x \mid \varphi_x(2x) \downarrow\}$$

- Definire $X_n = dom(\psi_n)$
- Dimostrare che X_n sono ricorsivi
- Mostrare che $\bigcup_n dom(\psi_n) = \{x \mid \varphi_x(2x) \downarrow\}$

Definiamo ψ_n t.c. $X_n = dom(\psi_n)$:

$$\psi_n(x) = \begin{cases} 1 & x \in X_n \land \varphi_x(2x) \text{ termina in n passi} \\ \uparrow & \text{altrimenti} \end{cases}$$

Dimostriamo che X_n è ricorsivo. (COMPLETARE)

$$\begin{aligned} \bigcup_n dom(\psi_n) &\implies \{x \mid \exists n. x \in dom(\psi_n)\} \\ &\implies \{x \mid \exists n. \varphi_x(2x) \text{ termina in passi}\} \\ &\implies \{x \mid \varphi_x(2x) \downarrow\} \end{aligned}$$

Esercizio 4

Successione di funzioni parziali ricorsive ψ_n tali che $dom(\psi_n)$ sono re non completi (ricorsivi) e che

$$\bigcap_n dom(\psi_n) = \{x \mid \varphi_x(2^x + 2) \uparrow\}$$

Sappiamo che $\bigcap_n$ corrisponde a $\forall n$. Mentre la divergenza può essere vista come $\forall n$ non termina.

$$X_n = dom(\psi_n) = \{x \mid \varphi_x(2^x + 2) \text{ non termina in n passi}\}$$

$$\psi_n(x) = \begin{cases} 1 & \varphi_x(2^x + 2) \text{ non termina in n passi} \\ \uparrow & \text{altrimenti} \end{cases}$$

Dimostra che X_n è ricorsivo. (COMPLETARE)

$$\begin{aligned} \bigcap_n dom(\psi_n) &\implies \{x \mid \forall n \,.\, x \in X_n\} \\ &\implies \{x \mid \forall n. \varphi_x(2^x + 2) \text{ non termina in n passi}\} \\ &\implies \{x \mid \varphi_x(2^x + 2) \uparrow\} \end{aligned}$$

Esercizio 4

Successione di funzioni parziali ricorsive ψ_n tali che $Range(\psi_n)$ sono creativi e che

$$\bigcap_n Range(\psi_n) = \{x^2 \mid W_x = \mathbb{N}\}$$

Sappiamo che $\bigcap_n$ corrisponde a $\forall n$ e che $W_x = \mathbb{N}$ vuol dire che $\forall n \,.\, n \in W_x$(ovvero $\varphi_x(n)\downarrow$).

- $$X_n = Range(\psi_n) = \{x^2 \mid \varphi_x(n)\downarrow\} = \{x^2 \mid n \in W_x\}$$

$$\psi_n(x) = \begin{cases} x^2 & \varphi_x(n)\downarrow \\ \uparrow & \text{altrimenti} \end{cases}$$

Dimostrare che X_n è creativo. (dimostrare quindi che siano RE e che $k \preceq X_n$)

$$\begin{aligned} \bigcap_n X_n = \bigcap_n Range(\psi_n) &= \{x^2 \mid \forall n \,.\, \varphi_x(n)\downarrow\} \\ &= \{x^2 \mid W_x = \mathbb{N}\} \end{aligned}$$

Esercizio 8

Successione di X_n produttivi tale che $\bigcap_n X_n \preceq k$ (k completo $\Rightarrow \forall X$ *re* e $x \preceq k$).
Una possibile soluzione potrebbe essere

$$\bigcap_n X_n = \varnothing$$

può essere una soluzione in quanto $\varnothing \preceq k$ perché $\varnothing$ è ricorsivo e quindi re. k completo.

$$X_n = \{x \mid W_x = [0,n]\} \forall m \neq n \,.\, W_x \neq [0,m]$$

$$\bigcap X_n = \varnothing \qquad \begin{aligned} y \in X_n &\implies W_y = [0,n] \neq [0,n+1] \\ &\implies y \notin X_{n+1} \end{aligned}$$

visto genericità di y e n allora $\bigcap X_n = \varnothing$. $\varnothing \preceq k$ perché k completo (Sol. alternativa: $X_n = \{x \mid W_x = n\mathbb{N}\}$ o $\{x \mid W_x = \{n\}^{\mathbb{N}}\}$).

4 Esame 03-02-2025

Esercizio 1

Studiare gli insiemi $D_n = dom(\psi_n)$ (e i loro complementari) al variare di $n > 2$ dove:

$$\psi_n(x) = \begin{cases} \lceil \sqrt{n+x} \rceil & 4x+1 \in 2\mathbb{N}+1 \land 4x+1 \in W_x \text{ non termina in meno di n passi} \\ \uparrow & \text{altrimenti} \end{cases}$$

Per studiare gli insiemi D_n e i loro complementari al variare di $n > 2$, iniziamo analizzando la definizione di $\psi_n(x)$. L'insieme D_n è definito come il dominio della funzione parziale ricorsiva ψ_n. Quindi, D_n contiene tutti gli elementi x per cui $\psi_n(x)$ è definita, ovvero:

$$D_n = \{x \mid 4x+1 \in 2\mathbb{N}+1 \land 4x+1 \in W_x \text{ non termina in meno di n passi}\}$$

L'insieme è intuitivamente ricorsivo poiché possiamo fornire un algoritmo che verifica le condizioni sopra elencate per ogni x e quindi riesce a decidere se x appartiene a D_n o meno.

$$f_{D_n}(x) = \begin{cases} 1 & \text{se } x \in D_n \\ 0 & \text{se } x \notin D_n \end{cases}$$

```
input(x, n)
costruisci phi_x
z = 4*x + 1
if z % 2 == 0:
    return 0
for i in range(n):
    esegui un passo di phi_x(z)
    if termina:
        return 0
return 1
```

Abbiamo costruito un algoritmo che verifica se x appartiene a D_n. Quindi, D_n è ricorsivo per ogni $n > 2$. Il complementare di D_n, denotato come $\overline{D_n}$, è l'insieme degli elementi x per cui $\psi_n(x)$ non è definita:

$$\overline{D_n} = \{x \mid 4x+1 \notin 2\mathbb{N}+1 \lor 4x+1 \notin W_x \text{ termina in meno di n passi}\}$$

Anche $\overline{D_n}$ è ricorsivo, poiché possiamo costruire un algoritmo simile a quello precedente che verifica le condizioni per l'appartenenza a $\overline{D_n}$.

$$f_{\overline{D_n}}(x) = \begin{cases} 1 & \text{se } x \in \overline{D_n} \\ 0 & \text{se } x \notin \overline{D_n} \end{cases}$$

```
input(x, n)
costruisci phi_x
z = 4*x + 1
if z % 2 == 1:
    return 1
for i in range(n):
    esegui un passo di phi_x(z)
    if termina:
        return 1
return 0
```

Abbiamo costruito un algoritmo che verifica se x appartiene a $\overline{D_n}$. Quindi, $\overline{D_n}$ è ricorsivo per ogni $n > 2$.

Esercizio 2

Studiare l'insieme:

$$R = \bigcap_{n>0} D_n$$

e il suo complementare.

L'insieme R è definito come l'intersezione di tutti gli insiemi D_n per $n > 0$:

$$R = \bigcap_{n>0} D_n = \{x \mid \forall n > 0, x \in D_n\}$$

Per capire cosa significa appartenere a R, dobbiamo analizzare le condizioni per l'appartenenza a ciascun D_n. Un elemento x appartiene a R se soddisfa le condizioni di appartenenza a tutti gli insiemi D_n. In particolare, ciò significa che per ogni $n > 0$, $4x + 1$ deve essere un numero dispari e deve appartenere a W_x senza terminare in meno di n passi. Quindi, possiamo riscrivere R come:

$$R = \{x \mid \forall n > 0,\ 4x + 1 \in 2\mathbb{N} + 1 \wedge \varphi_x(4x + 1) \downarrow \text{ non deciso in } n \text{ passi}\}$$

Il complementare di R, denotato come $\overline{R}$, è l'insieme degli elementi x che non appartengono a R:

$$\overline{R} = \{x \mid \exists n > 0, x \notin D_n\}$$

Ciò significa che esiste almeno un $n > 0$ tale che x non soddisfa le condizioni di appartenenza a D_n. In particolare, ciò significa che per almeno un $n > 0$, $4x + 1$ deve essere un numero pari o non deve appartenere a W_x o deve terminare in meno di n passi.

$$\overline{R} = \{x \mid 4x+1 \notin 2\mathbb{N}+1 \lor 4x+1 \notin W_x \text{ deciso in un numero finito di passi}\}$$

Posso scriverlo come

$$\overline{R} = \{x \mid \exists n > 0, 4x+1 \notin 2\mathbb{N}+1 \lor \varphi_x(4x+1)\uparrow \text{ deciso in } n \text{ passi}\}$$

Notiamo come R non sia ricorsivo in quanto non è possibile decidere se un programma non termina in un numero finito di passi. R è quindi produttivo. Per dimostrare che R è produttivo dobbiamo dimostrare che

$$\overline{K} \preceq R \text{ o } K \preceq \overline{R}$$

Definiamo la funzione parziale ricorsiva:

$$\psi(x,y) = \begin{cases} 1 & x \in K \\ \uparrow & \text{altrimenti} \end{cases}$$

Costruiamo l'algoritmo per ψ:

```
input(x,y)
costruisci phi_x
while true
    esegui next step di phi_x(x)
    if termina: return 1
```

Per il teorema di s-m-n esiste una funzione totale e calcolabile g tale che:

$$\varphi_{g(x)}(y) = \psi(x,y)$$

Se non termina su tutti gli input, sicuramente non termina su $4g(x)+1$ e quindi non verrà deciso in n passi.

$$\begin{aligned} x \notin K &\implies \forall y \,.\, \psi(x,y)\uparrow \\ &\implies \forall y \,.\, \varphi_{g(x)}(y)\uparrow \\ &\implies W_{g(x)} = \varnothing \\ &\implies \varphi_{g(x)}(4g(x)+1)\uparrow \text{ non verrà mai deciso in n passi} \\ &\implies g(x) \in R \end{aligned}$$

Se termina su tutti gli input, sicuramente termina su $4g(x)+1$ e quindi esiste n tale che termina in n passi.

$$\begin{aligned} x \in K &\implies \forall y \;.\; \psi(x,y)\downarrow \\ &\implies \forall y \;.\; \varphi_{g(x)}(y)\downarrow \\ &\implies W_{g(x)} = \mathbb{N} \\ &\implies \varphi_{g(x)}(4g(x)+1)\downarrow \\ &\implies g(x) \notin R \end{aligned}$$

Dobbiamo fare la stessa cosa per $\overline{R}$. Per provare che sia produttivo dobbiamo dimostrare che

$$K \preceq R$$

In questo caso Definiamo la funzione parziale ricorsiva:

$$\psi(x,y) = \begin{cases} 1 & x \in K \vee y = 4x+1 \\ \uparrow & \text{altrimenti} \end{cases}$$

Costruiamo l'algoritmo per ψ:

Per il teorema di s-m-n esiste una funzione totale e calcolabile g tale che:

$$\varphi_{g(x)}(y) = \psi(x,y)$$

Vorremmo che quando $x \in K$ allora $g(x) \in R$. Questa cosa è possibile solo se

$$\begin{aligned} x \in K &\implies \forall y \;.\; \psi(x,y)\downarrow \\ &\implies \forall y \;.\; \varphi_{g(x)}(y)\downarrow \\ &\implies W_{g(x)} = \mathbb{N} \\ &\implies \varphi_{g(x)}(4g(x)+1)\downarrow \\ &\implies g(x) \in \overline{R} \end{aligned}$$

Esercizio 3

Studiare il seguente insieme (ed il suo complementare)

$$T = \{x \mid (\exists n \in \mathbb{N} \,.\, y = 3^{3n} \Rightarrow \varphi_x(y)\downarrow) \wedge (y \in W_x \Rightarrow \exists n \in \mathbb{N} \,.\, y = 3^n)\}$$

Scriviamo meglio l'insieme T. Sappiamo che $\exists n \in \mathbb{N} \,.\, y = 3^{3n}$ corrisponde a $y \in \{27\}^{\mathbb{N}}$ e $\exists n \in \mathbb{N} \,.\, y = 3^n$ corrisponde a $y \in \{3\}^{\mathbb{N}}$. Sappiamo inoltre che

$$y \in W_x \Leftrightarrow \varphi_x(y)\downarrow$$

Quindi possiamo riscrivere T come:

$$T = \{x \mid \{27\}^{\mathbb{N}} \subseteq W_x \subseteq \{3\}^{\mathbb{N}}\}$$

Questo insieme richiede che per ogni $y \in W_x$ deve essere una potenza di 3 e che tutte le potenze di 27 devono essere in W_x. Questo insieme è intuitivamente produttivo in quanto non è possibile decidere se un programma termina su tutte le potenze di 27 e non è possibile decidere se un programma termina solo su potenze di 3. Per dimostrare che T è produttivo dobbiamo dimostrare che

$$\overline{K} \preceq T \text{ o } K \preceq \overline{T}$$

Definiamo la funzione parziale ricorsiva:

$$\psi(x, y) = \begin{cases} 1 & x \in K \vee y \in \{3\}^{\mathbb{N}} \\ \uparrow & \text{altrimenti} \end{cases}$$

Costruiamo l'algoritmo che calcola ψ:

```
input(x,y)
costruisci phi_x  (procedura effettiva algoritmica calcolabile)
if y in 3^N: (procedura effettiva algoritmica calcolabile)
    return 1
while true
    esegui next step di phi_x(x)
    if termina: return 1
```

Per il teorema di smn esiste una funzione totale ricorsiva g tale che:

$$\varphi_{g(x)}(y) = \psi(x, y)$$

Valgono le seguenti implicazioni:

$$
\begin{aligned}
x \in K &\implies \forall y \,.\, \psi(x,y)\downarrow & x \notin K &\implies \psi(x,y)\downarrow \text{ sse } y \in \{3\}^{\mathbb{N}} \\
&\implies \forall y \,.\, \varphi_{g(x)}(y)\downarrow & &\implies \varphi_{g(x)}(y)\downarrow \text{ sse } y \in \{3\}^{\mathbb{N}} \land y \in \{27\}^{\mathbb{N}} \\
&\implies W_{g(x)} = \mathbb{N} \neq \{3\}^{\mathbb{N}} & &\implies W_{g(x)} = \{3\}^{\mathbb{N}} \supseteq \{27\}^{\mathbb{N}} \\
&\implies g(x) \notin T & &\implies g(x) \in T
\end{aligned}
$$

Dimostriamo ora che $\overline{T}$ è produttivo dimostrando che

$$\overline{K} \preceq \overline{T} \text{ o } K \preceq T$$

Sappiamo che $\overline{T}$ può essere scritto come:

$$\overline{T} = \{x \mid W_x \not\supseteq \{27\}^{\mathbb{N}} \lor W_x \not\subseteq \{3\}^{\mathbb{N}}\}$$

Definiamo la funzione parziale ricorsiva:

$$\psi(x,y) = \begin{cases} 1 & x \in K \land y \in \{3\}^{\mathbb{N}} \\ \uparrow & \text{altrimenti} \end{cases}$$

Costruiamo l'algoritmo che calcola ψ:

```
input(x,y)
costruisci phi_x  (procedura effettiva algoritmica calcolabile)
if y in 3^N: (procedura effettiva algoritmica calcolabile)
    while true
        esegui next step di phi_x(x)
        if termina: return 1
```

Per il teorema di smn esiste una funzione totale ricorsiva g tale che:

$$\varphi_{g(x)}(y) = \psi(x,y)$$

Valgono le seguenti implicazioni:

$$
\begin{aligned}
x \in K &\implies \psi(x,y)\downarrow \text{ sse } y \in \{3\}^{\mathbb{N}} & x \notin K &\implies \forall y \,.\, \psi(x,y)\uparrow \\
&\implies \varphi_{g(x)}(y)\downarrow \text{ sse } y \in \{3\}^{\mathbb{N}} & &\implies \forall y \,.\, \varphi_{g(x)}(y)\uparrow \\
&\implies W_{g(x)} = \{3\}^{\mathbb{N}} \supseteq \{27\}^{\mathbb{N}} & &\implies W_{g(x)} = \varnothing \not\supseteq \{27\}^{\mathbb{N}} \\
&\implies g(x) \in T & &\implies g(x) \notin T
\end{aligned}
$$