# Grammatiche, Sintassi, Semantica

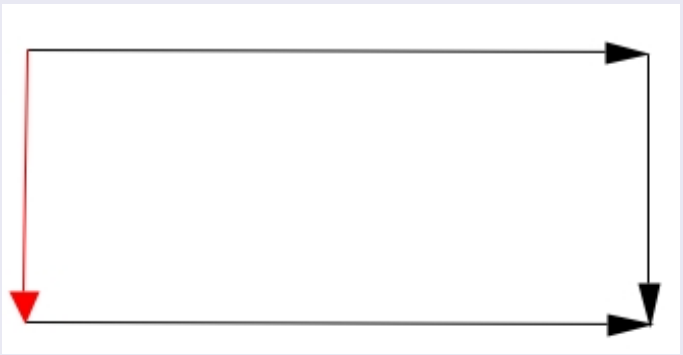
Sommario: 17 marzo, 2020

- Strumenti
- Grammatiche, Parse Tree
- Ambiguità
- Linguaggi Sintattici
- Semantica SOS
- Sintassi e Semantica: Conoscenza Indispensabile

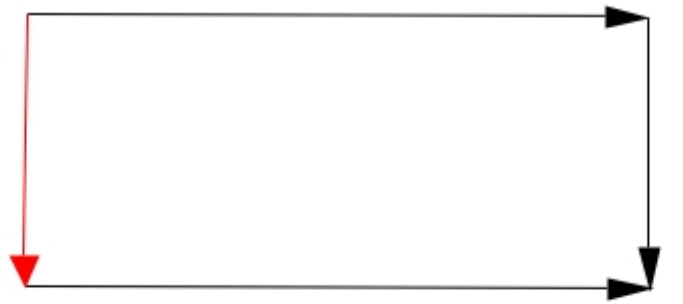
# Esercizio L3.10

**Esercizio (10)**

*Nella costruzione di un compilatore $\mathcal{C}_{\mathcal{L},\mathcal{L}_O}$ utilizziamo le rappresentazioni $-_{\mathcal{L}}$ e $-_{\mathcal{L}_0}$ dei programmi su $\mathcal{D}$, i mappings $\mathcal{U}_{\mathcal{L}}$, $\mathcal{U}_{\mathcal{L}_O}$, in modo da creare un diagramma, come quello sotto, che commuta definendo $\mathcal{C}_{\mathcal{L},\mathcal{L}_O}$. Sapreste mettere i mapping nelle giuste freccie?*



**Soluzione**



# Esercizio L3.15

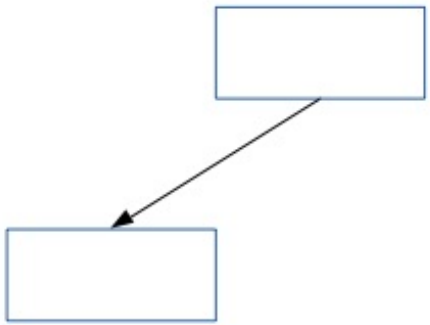
**Esercizio (15)**

*Nella costruzione di un compilatore $\mathcal{C}^{\mathcal{L}_0}_{\mathcal{L},\mathcal{L}_0}$ attraverso cross-compiling ricorriamo a 2 Macchine Astratte: La Macchina Intermedia $\mathcal{M}_{\mathtt{I}}$, con Linguaggio $\mathcal{L}_{\mathtt{I}}$ ed esecutore $\mathcal{E}_{\mathtt{I}}$, e la Macchina Target $\mathcal{M}_0$, con Linguaggio $\mathcal{L}_0$ ed esecutore $\mathcal{E}_0$. Indicate come sempre, con $\mathcal{U}_{\mathcal{L}}$, $\mathcal{U}_{\mathcal{L}_{\mathtt{I}}}$, $\mathcal{U}_{\mathcal{L}_0}$, le universali coinvolte:*

*(a) Si mostri il diagramma con cui è stato ottenuto il compilatore $\mathcal{C}^{\mathcal{L}_0}_{\mathcal{L},\mathcal{L}_0}$, mostrando: Le Macchine coinvolte, i 2 Compilatori componenti, prodotti, e le fasi della loro generazione.*

*(b) Si mostri come i 2 compilatori componenti debbano essere utilizzati per generare $\mathcal{C}^{\mathcal{L}_0}_{\mathcal{L},\mathcal{L}_0}$, fornendo anche una prova della correttezza del procedimento utilizzato*

**Soluzione**



# Esercizi L4.10-1

- Esercizio L4.10.1
  (a) Si completi la definizione di una grammatica le cui produzioni sono sotto.
  (b) Si mostri che la grammatica ottenuta è ambigua.
  (c) Si mostri una grammatica equivalente e non ambigua.

$$\mathtt{E} \rightarrow \mathtt{T} \mid \mathtt{T} + \mathtt{E} \mid \mathtt{T} - \mathtt{E}$$
$$\mathtt{T} \rightarrow \mathtt{A} \mid \mathtt{A} * \mathtt{E}$$

**Soluzione**

(a) $\mathtt{G} = \ldots$

(b) Consideriamo la stringa: `s=A*A+A`. Per essa possiamo mostrare i due diversi parse tree sotto aventi come frontiera `s`

(c) Dobbiamo introdurre ed esprimere la precedenza tra operatori

# Esercizi L4.10-1bis

- Esercizio L4.10.1bis
  Si considerino le produzioni di una grammatica per comandi Cmd di un LP

  ```
  Cmd → if (BExp) Cmd else Cmd
      | if (BExp) Cmd
      | NonConditionalCmd;
  NonConditionalCmd → ...
  BExp → ...
  ```

  dove omettiamo la definizione del nonterminale `BExp` (per espressioni booleane) e `NonConditionalCmd` (per i vari comandi non condizionali)
  (a) Si dimostri che la grammatica è ambigua.
  (b) Si fornisca una grammatica non ambigua per lo stesso linguaggio

  **Soluzione**
  (a)

  (b)

# Esercizi L4.10-1ter

- Esercizio L4.10.1ter
  Si considerino le produzioni di una grammatica per comandi Cmd di un LP

  `Cmd` $\longrightarrow$ **if** (`BExp`) `Cmd` **else** `Cmd`
  | `OtherCmd`
  `OtherCmd` $\longrightarrow$ **if** (`BExp`) `OtherCmd`
  | `NonConditionalCmd`;
  | {`Cmd`}
  `NonConditionalCmd` $\longrightarrow$ ...
  `BExp` $\longrightarrow$ ...

  dove le definizioni di `BExp` e `NonConditionalCmd` sono le stesse assunte in esercizio 2.10.1bis.
  Si consideri il comando sotto (dove false e true sono le 2 costanti booleane).

  **if** (`false`) **if** (`true`) $C_1$; **else** $C_2$;

  (a) L'esecuzione di tale comando (nell'usuale semantica dei condizionali) conduce all'esecuzione di quale dei seguenti comandi?:
  1) $C_1$; 2) $C_2$; 3) nessuno dei precedenti

  (b) Si fornisca una descrizione generale della struttura data ai condizionali, spiegando come ciò garantisca la non ambiguità della composizione di condizionali.

  **Soluzione**
  (a) (Hint: Si calcoli il Parse Tree di tale costrutto... E se ne esamini la struttura in sotto-termini).

  (b)

# Esercizi L4.10-1quater

- Esercizio L4.10.1quater
  Si modifichi la grammatica dell'esercizio 2.10-1ter, affinchè il comando:
  **if** (`false`) **if** (`true`) $\texttt{C}_1$; **else** $\texttt{C}_2$;
  abbia struttura tale che la sua esecuzione non conduca all'esecuzione del coman- do $\texttt{C}_1$ nè del comando $\texttt{C}_2$.

**Soluzione**
(Hint. Invertire la precedenza degli operatori/costrutti `if__else_` ed `if__`.)

# Esercizi. L4.10-2

- Esercizio L4.10.2
  (a) Si dia una grammatica $\mathtt{G}$ non ambigua che generi tutte e sole le sequenze di parentesi angolate bilanciate
  (b) Si dimostri che $\langle\langle\langle\rangle\langle\langle\rangle\rangle\rangle\rangle \in \mathcal{L}(\mathtt{G})$

  **Soluzione**
  (a) Le produzioni di $\mathtt{G}$

  $\mathtt{S} \rightarrow$

  $\mathtt{B} \rightarrow$

  (b)

# Esercizi. L4.10-3

- Esercizio L4.10-3
  Si chiama lineare una grammatica le cui produzioni hanno la forma $\mathtt{A} \to \mathtt{t}\ \mathtt{B}$ oppure $\mathtt{A} \to \mathtt{t}$, dove $\mathtt{A}$, $\mathtt{B}$ siano non terminali e $\mathtt{t}$ sia un terminale o $\epsilon$. Un linguaggio che può essere espresso con una grammatica lineare si chiama *regolare* e può essere riconosciuto da un automa a stati finiti.
  (a) Si mostri che il linguaggio $\mathtt{L} = \{\mathtt{a}^{\mathtt{n}}\mathtt{b}^{\mathtt{m}} \mid \mathtt{n}, \mathtt{m} \geq 1\}$ è regolare.
  (b) Si dimostri che $\mathtt{aaab} \in \mathtt{L}$

  **Soluzione**
  (a) Le produzioni di una grammatica lineare per $\mathtt{L}$

  $\mathtt{S} \to$
  $\mathtt{B} \to$

  (b)

# Esercizi. L4.10-3bis

- Esercizio L4.10-3bis
  (a) Si dia una grammatica non ambigua per che il linguaggio:
  $$\mathtt{L1} = \{\mathtt{a}^{\mathtt{n}}\mathtt{b}^{\mathtt{m}} \mid \mathtt{n} \leq \mathtt{m}\}$$
  (b) Si dica perchè la grammatica per esercizio 2.10-5 non è corretta per L1

  **Soluzione**
  (a) Le produzioni di una grammatica per L1
  $$\mathtt{S} \rightarrow$$
  $$\mathtt{B} \rightarrow$$
  od anche:
  $$\mathtt{S} \rightarrow$$
  $$\mathtt{A} \rightarrow$$
  $$\mathtt{B} \rightarrow$$

  (b)

# Esercizi. L4.10-4

$$\langle X, \sigma \rangle \to \langle \sigma(X), \sigma \rangle$$

$$\langle (n+m), \sigma \rangle \to \langle p, \sigma \rangle \quad \text{where } p = n + m$$

$$\langle (n-m), \sigma \rangle \to \langle p, \sigma \rangle \quad \text{where } p = n - m \text{ e } n \geq m$$

$$\frac{\langle a_1, \sigma \rangle \to \langle a', \sigma \rangle}{\langle (a_1 + a_2), \sigma \rangle \to \langle (a' + a_2), \sigma \rangle} \qquad \frac{\langle a_2, \sigma \rangle \to \langle a'', \sigma \rangle}{\langle (a_1 + a_2), \sigma \rangle \to \langle (a_1 + a''), \sigma \rangle}$$

$$\frac{\langle a_1, \sigma \rangle \to \langle a', \sigma \rangle}{\langle (a_1 - a_2), \sigma \rangle \to \langle (a' - a_2), \sigma \rangle} \qquad \frac{\langle a_2, \sigma \rangle \to \langle a'', \sigma \rangle}{\langle (a_1 - a_2), \sigma \rangle \to \langle (a_1 - a''), \sigma \rangle}$$

- Esercizio L4.10-4
  Si modifichino le regole delle espressioni aritmetiche in modo da prescrivere una valutazione degli argomenti da sinistra a destra per la somma e una da destra a sinistra per la sottrazione.

  **Soluzione**

# Esercizi. L4.10-5

$$\langle \textbf{skip}, \sigma \rangle \to \sigma \quad (c1)$$

$$\langle X := n, \sigma \rangle \to \sigma[X \leftarrow n] \quad (c2) \qquad \frac{\langle a, \sigma \rangle \to \langle a', \sigma \rangle}{\langle X := a, \sigma \rangle \to \langle X := a', \sigma \rangle} \quad (c3)$$

$$\frac{\langle c_1, \sigma \rangle \to \sigma'}{\langle c_1; c_2, \sigma \rangle \to \langle c_2, \sigma' \rangle} \quad (c4) \qquad \frac{\langle c_1, \sigma \rangle \to \langle c_1', \sigma' \rangle}{\langle c_1; c_2, \sigma \rangle \to \langle c_1'; c_2, \sigma' \rangle} \quad (c5)$$

$$\langle \textbf{if tt then } c_1 \textbf{ else } c_2, \sigma \rangle \to \langle c_1, \sigma \rangle \quad (c6)$$
$$\langle \textbf{if ff then } c_1 \textbf{ else } c_2, \sigma \rangle \to \langle c_2, \sigma \rangle \quad (c7)$$

$$\frac{\langle b, \sigma \rangle \to \langle b', \sigma \rangle}{\langle \textbf{if } b \textbf{ then } c_1 \textbf{ else } c_2, \sigma \rangle \to \langle \textbf{if } b' \textbf{ then } c_1 \textbf{ else } c_2, \sigma \rangle} \quad (c8)$$

$$\langle \textbf{while } b \textbf{ do } c, \sigma \rangle \to \langle \textbf{if } b \textbf{ then } c; \textbf{while } b \textbf{ do } c \textbf{ else skip}, \sigma \rangle \quad (c9)$$

- Esercizio L4.10-5
  Siano `c` e `d` i seguenti comandi:
  `c:` `X:=1;`
  `d:` **while**`(X==1)` **do skip**
  Si dia la sequenza di transizioni ottenute a partire dallo stato $\sigma = (\texttt{X}, 0)$

  **Soluzione**

# Sintassi e Semantica: Conoscenza Indispensabile

- Questi esercizi mostrano come sia indispensabile la conoscenza di Sintassi e Semantica per poter usare un LP.
- Non sempre però le due definizioni sono facili da consultare.
- Non sempre sono fornite in modo formale o completo:
  - Sintassi fornita con una grammatica ambigua, corredata di vincoli aggiuntivi.
  - Semantica fornita mediante una sommaria, informale, incompleta, descrizione del comportamento atteso del costrutto.
  - Semantica fornita con una descrizione di esempi di uso del costrutto.
- In questi casi è difficile usare in modo corretto il Linguaggio di Programmazione ed ancora,
- Impossibile garantire la correttezza dei programmi scritti, ed è dunque
- Ragionevole attendersi continui, nuovi, malfunzionamenti dal vecchio programma, o applicazione, o sistema, in uso.
- Il linguaggio C: Vedi gli esempi sull'allocazione dinamica nel Listing allegato, CastEAlloca.zip.