STRUMENTI FORMALI PER LA BIOINFORMATICA

Grammatiche e Linguaggi Context-free
Parte 1

Linguaggi e Informatica

In informatica i linguaggi formali sono di vitale importanza.

Linguaggi e Informatica

In informatica i linguaggi formali sono di vitale importanza.

Essi sono utilizzati come rappresentazione formale di problemi di decisione e l'eventuale soluzione algoritmica a tali problemi si traduce nell'esistenza o meno di modelli di computazione per il riconoscimento delle stringhe di un linguaggio.

In informatica i linguaggi formali sono di vitale importanza.

Essi sono utilizzati come rappresentazione formale di problemi di decisione e l'eventuale soluzione algoritmica a tali problemi si traduce nell'esistenza o meno di modelli di computazione per il riconoscimento delle stringhe di un linguaggio.

La teoria sviluppata è uno degli esempi più evidenti di come lo studio di un argomento teorico, e la comprensione delle proprietà formali dei suoi elementi abbia permesso lo sviluppo di tecniche automatiche per risolvere efficientemente compiti che richiedevano tempi e risorse non banali.

L'esempio più significativo è quello della costruzione di quelle parti dei compilatori che acquisiscono i programmi degli utenti, ne verificano la correttezza "lessicale" e "sintattica" e ne trasformano la rappresentazione in modo da agevolare i passi successivi di analisi, ottimizzazione e generazione del codice da eseguire.

L'esempio più significativo è quello della costruzione di quelle parti dei compilatori che acquisiscono i programmi degli utenti, ne verificano la correttezza "lessicale" e "sintattica" e ne trasformano la rappresentazione in modo da agevolare i passi successivi di analisi, ottimizzazione e generazione del codice da eseguire.

Questi *analizzatori lessicali* e *analizzatori sintattici*, come vengono chiamati, sono disponibili fin dalla fine degli anni '70 e in brevissimo tempo realizzano quello che in precedenza ne richiedeva molto di più.

L'esempio più significativo è quello della costruzione di quelle parti dei compilatori che acquisiscono i programmi degli utenti, ne verificano la correttezza "lessicale" e "sintattica" e ne trasformano la rappresentazione in modo da agevolare i passi successivi di analisi, ottimizzazione e generazione del codice da eseguire.

Questi *analizzatori lessicali* e *analizzatori sintattici*, come vengono chiamati, sono disponibili fin dalla fine degli anni '70 e in brevissimo tempo realizzano quello che in precedenza ne richiedeva molto di più.

Una serie di risultati teorici ha reso possibile costruire programmi che "programmano".

Esamineremo i linguaggi da due punti di vista complementari, basati sull'approccio *riconoscitivo* e su quello *generativo*.

Esamineremo i linguaggi da due punti di vista complementari, basati sull'approccio *riconoscitivo* e su quello *generativo*.

In particolare, nell'approccio riconoscitivo studieremo procedure, sotto forma di "automi", che definiscono un linguaggio attraverso "l'accettazione" delle stringhe che lo costituiscono.

Esamineremo i linguaggi da due punti di vista complementari, basati sull'approccio *riconoscitivo* e su quello *generativo*.

In particolare, nell'approccio riconoscitivo studieremo procedure, sotto forma di "automi", che definiscono un linguaggio attraverso "l'accettazione" delle stringhe che lo costituiscono.

Seguendo l'approccio generativo invece, si definiscono meccanismi di calcolo che sono sistemi di riscrittura particolari e sono chiamati "grammatiche", attraverso le quali si possono ottenere, appunto generandole, le stringhe del linguaggio in questione.

Partiremo da grammatiche abbastanza semplici ma che giocano un ruolo importante nella sintassi dei linguaggi di programmazione, le grammatiche context-free.

Grammatiche context-free

Partiremo da grammatiche abbastanza semplici ma che giocano un ruolo importante nella sintassi dei linguaggi di programmazione, le grammatiche context-free.

Le grammatiche context-free possono descrivere alcuni aspetti che hanno una struttura ricorsiva.

Grammatiche context-free

Partiremo da grammatiche abbastanza semplici ma che giocano un ruolo importante nella sintassi dei linguaggi di programmazione, le grammatiche context-free.

Le grammatiche context-free possono descrivere alcuni aspetti che hanno una struttura ricorsiva.

Questo le rende utili in una varietà di applicazioni.

Grammatiche context-free

Partiremo da grammatiche abbastanza semplici ma che giocano un ruolo importante nella sintassi dei linguaggi di programmazione, le grammatiche context-free.

Le grammatiche context-free possono descrivere alcuni aspetti che hanno una struttura ricorsiva.

Questo le rende utili in una varietà di applicazioni.

Inizialmente le grammatiche context-free furono usate nello studio dei linguaggi naturali.

Grammatiche context-free - un esempio

Un esempio di grammatica context-free, che chiameremo G_1:

$$
\begin{aligned}
A &\rightarrow 0A1 \\
A &\rightarrow B \\
B &\rightarrow \#
\end{aligned}
$$

Grammatiche context-free - un esempio

Un esempio di grammatica context-free, che chiameremo G_1:

$$\begin{aligned} A &\rightarrow 0A1 \\ A &\rightarrow B \\ B &\rightarrow \# \end{aligned}$$

Una grammatica consiste di un insieme di *regole di sostituzione*, chiamate **produzioni**.

Grammatiche context-free - un esempio

Un esempio di grammatica context-free, che chiameremo G_1:

$$\begin{aligned} A &\rightarrow 0A1 \\ A &\rightarrow B \\ B &\rightarrow \# \end{aligned}$$

Una grammatica consiste di un insieme di *regole di sostituzione*, chiamate **produzioni**.

Ogni regola è costituita da un simbolo, chiamato **variabile**, e una stringa separati da una freccia.

Grammatiche context-free - un esempio

Un esempio di grammatica context-free, che chiameremo G_1:

$$\begin{aligned} A &\rightarrow 0A1 \\ A &\rightarrow B \\ B &\rightarrow \# \end{aligned}$$

Una grammatica consiste di un insieme di *regole di sostituzione*, chiamate **produzioni**.

Ogni regola è costituita da un simbolo, chiamato **variabile**, e una stringa separati da una freccia.

La stringa consiste di variabili e altri simboli chiamati **terminali**.

Grammatiche context-free - un esempio

Un esempio di grammatica context-free, che chiameremo G_1:

$$\begin{aligned} A &\rightarrow 0A1 \\ A &\rightarrow B \\ B &\rightarrow \# \end{aligned}$$

Una grammatica consiste di un insieme di *regole di sostituzione*, chiamate **produzioni**.

Ogni regola è costituita da un simbolo, chiamato **variabile**, e una stringa separati da una freccia.

La stringa consiste di variabili e altri simboli chiamati **terminali**.

Le variabili sono spesso rappresentate da lettere maiuscole. I terminali sono spesso rappresentati da lettere minuscole, numeri o simboli speciali.

Una delle variabili è chiamata la **variabile iniziale**. In alcuni testi (Sipser) si assume che sia la variabile sul lato sinistro della regola più in alto, a meno che non sia diversamente specificato.

Grammatiche context-free - un esempio

Una delle variabili è chiamata la **variabile iniziale**. In alcuni testi (Sipser) si assume che sia la variabile sul lato sinistro della regola più in alto, a meno che non sia diversamente specificato.

Per esempio, la grammatica G_1 ha tre regole:

$$\begin{aligned} A &\rightarrow 0A1 \\ A &\rightarrow B \\ B &\rightarrow \# \end{aligned}$$

Grammatiche context-free - un esempio

Una delle variabili è chiamata la **variabile iniziale**. In alcuni testi (Sipser) si assume che sia la variabile sul lato sinistro della regola più in alto, a meno che non sia diversamente specificato.

Per esempio, la grammatica G_1 ha tre regole:

$$\begin{aligned} A &\rightarrow 0A1 \\ A &\rightarrow B \\ B &\rightarrow \# \end{aligned}$$

Le variabili di G_1 sono A e B.

Grammatiche context-free - un esempio

Una delle variabili è chiamata la **variabile iniziale**. In alcuni testi (Sipser) si assume che sia la variabile sul lato sinistro della regola più in alto, a meno che non sia diversamente specificato.

Per esempio, la grammatica G_1 ha tre regole:

$$\begin{aligned} A &\rightarrow 0A1 \\ A &\rightarrow B \\ B &\rightarrow \# \end{aligned}$$

Le variabili di G_1 sono A e B.

A è la variabile iniziale.

Grammatiche context-free - un esempio

Una delle variabili è chiamata la **variabile iniziale**. In alcuni testi (Sipser) si assume che sia la variabile sul lato sinistro della regola più in alto, a meno che non sia diversamente specificato.

Per esempio, la grammatica G_1 ha tre regole:

$$\begin{aligned} A &\rightarrow 0A1 \\ A &\rightarrow B \\ B &\rightarrow \# \end{aligned}$$

Le variabili di G_1 sono A e B.

A è la variabile iniziale.

I terminali di G_1 sono 0, 1 e $\#$.

Grammatiche context-free - descrizione informale

Una grammatica può essere usata per descrivere un linguaggio generando ogni stringa del linguaggio nel seguente modo.

Grammatiche context-free - descrizione informale

Una grammatica può essere usata per descrivere un linguaggio generando ogni stringa del linguaggio nel seguente modo.

1. Scrivi la variabile iniziale.

Grammatiche context-free - descrizione informale

Una grammatica può essere usata per descrivere un linguaggio generando ogni stringa del linguaggio nel seguente modo.

1. Scrivi la variabile iniziale.
2. Trova una variabile che è stata scritta e una regola che inizia con quella variabile. Sostituisci la variabile scritta con il lato destro di quella regola.

Grammatiche context-free - descrizione informale

Una grammatica può essere usata per descrivere un linguaggio generando ogni stringa del linguaggio nel seguente modo.

1. Scrivi la variabile iniziale.
2. Trova una variabile che è stata scritta e una regola che inizia con quella variabile. Sostituisci la variabile scritta con il lato destro di quella regola.
3. Ripeti il passo 2 fino a quando non vi sono più variabili.

Grammatiche context-free - descrizione informale

Per esempio, la grammatica G_1

$$\begin{aligned} A &\to 0A1 \\ A &\to B \\ B &\to \# \end{aligned}$$

genera la stringa $000\#111$.

Grammatiche context-free - descrizione informale

Per esempio, la grammatica G_1

$$\begin{aligned} A &\to 0A1 \\ A &\to B \\ B &\to \# \end{aligned}$$

genera la stringa $000\#111$.

Una sequenza di sostituzioni per ottenere una stringa è chiamata una **derivazione**.

Grammatiche context-free - descrizione informale

Per esempio, la grammatica G_1

$$\begin{aligned} A &\to 0A1 \\ A &\to B \\ B &\to \# \end{aligned}$$

genera la stringa $000\#111$.

Una sequenza di sostituzioni per ottenere una stringa è chiamata una **derivazione**.

Una derivazione della stringa $000\#111$ nella grammatica G_1 è

$$A \Rightarrow 0A1 \Rightarrow 00A11 \Rightarrow 000A111 \Rightarrow 000B111 \Rightarrow 000\#111.$$

Grammatiche context-free - descrizione informale

Per esempio, la grammatica G_1

$$\begin{aligned} A &\to 0A1 \\ A &\to B \\ B &\to \# \end{aligned}$$

genera la stringa $000\#111$.

Una sequenza di sostituzioni per ottenere una stringa è chiamata una **derivazione**.

Una derivazione della stringa $000\#111$ nella grammatica G_1 è

$$A \Rightarrow 0A1 \Rightarrow 00A11 \Rightarrow 000A111 \Rightarrow 000B111 \Rightarrow 000\#111.$$

La stessa informazione può essere anche rappresentata mediante un disegno (**albero sintattico** o **parse tree**).

Tutte le stringhe generate in questo modo costituiscono il **linguaggio della grammatica**.

Grammatiche context-free - descrizione informale

Tutte le stringhe generate in questo modo costituiscono il **linguaggio della grammatica**.

Il linguaggio $L(G_1)$ della grammatica G_1 è

$$\{0^n\#1^n \mid n \geq 0\}$$

Un esempio informale: parole palindrome

Una parola è palindroma se letta da sinistra a destra o da destra a sinistra dà luogo alla stessa sequenza.

Un esempio informale: parole palindrome

Una parola è palindroma se letta da sinistra a destra o da destra a sinistra dà luogo alla stessa sequenza.

Quindi una parola $w \in \Sigma^*$ è palindroma se e solo se $w = w^R$.

Un esempio informale: parole palindrome

Una parola è palindroma se letta da sinistra a destra o da destra a sinistra dà luogo alla stessa sequenza.

Quindi una parola $w \in \Sigma^*$ è palindroma se e solo se $w = w^R$.

Esempi: alla, otto, abba, amima.

Un esempio informale: parole palindrome

Definizione ricorsiva delle parole palindrome sull'alfabeto $\{a, b\}$:

Un esempio informale: parole palindrome

Definizione ricorsiva delle parole palindrome sull'alfabeto $\{a, b\}$:

PASSO BASE: a, b e ϵ sono parole palindrome.

Un esempio informale: parole palindrome

Definizione ricorsiva delle parole palindrome sull'alfabeto $\{a, b\}$:

PASSO BASE: a, b e ϵ sono parole palindrome.

PASSO RICORSIVO: Se x è una parola palindroma allora anche axa e bxb sono parole palindrome.

Un esempio informale: parole palindrome

Definizione ricorsiva delle parole palindrome sull'alfabeto $\{a, b\}$:

PASSO BASE: a, b e ϵ sono parole palindrome.

PASSO RICORSIVO: Se x è una parola palindroma allora anche axa e bxb sono parole palindrome.

Esempio: $abba$ è palindroma

Un esempio informale: parole palindrome

Definizione ricorsiva delle parole palindrome sull'alfabeto $\{a, b\}$:

PASSO BASE: a, b e ϵ sono parole palindrome.

PASSO RICORSIVO: Se x è una parola palindroma allora anche axa e bxb sono parole palindrome.

Esempio: $abba$ è palindroma

(PASSO BASE) ϵ è una parola palindroma

Un esempio informale: parole palindrome

Definizione ricorsiva delle parole palindrome sull'alfabeto $\{a, b\}$:

PASSO BASE: a, b e ϵ sono parole palindrome.

PASSO RICORSIVO: Se x è una parola palindroma allora anche axa e bxb sono parole palindrome.

Esempio: $abba$ è palindroma

(PASSO BASE) ϵ è una parola palindroma

(PASSO RICORSIVO) $b\epsilon b = bb$ è una parola palindroma

Un esempio informale: parole palindrome

Definizione ricorsiva delle parole palindrome sull'alfabeto $\{a, b\}$:

PASSO BASE: a, b e ϵ sono parole palindrome.

PASSO RICORSIVO: Se x è una parola palindroma allora anche axa e bxb sono parole palindrome.

Esempio: $abba$ è palindroma

(PASSO BASE) ϵ è una parola palindroma

(PASSO RICORSIVO) $b\epsilon b = bb$ è una parola palindroma

(PASSO RICORSIVO) $abba$ è una parola palindroma.

Un esempio informale: dalla definizione ricorsiva alla grammatica

Le regole che definiscono le parole palindrome sull'alfabeto $\{a, b\}$ nella notazione delle grammatiche libere dal contesto:

Un esempio informale: dalla definizione ricorsiva alla grammatica

Le regole che definiscono le parole palindrome sull'alfabeto $\{a, b\}$ nella notazione delle grammatiche libere dal contesto:

1. $S \to \epsilon$

Un esempio informale: dalla definizione ricorsiva alla grammatica

Le regole che definiscono le parole palindrome sull'alfabeto $\{a, b\}$ nella notazione delle grammatiche libere dal contesto:

1. $S \to \epsilon$
2. $S \to a$

Un esempio informale: dalla definizione ricorsiva alla grammatica

Le regole che definiscono le parole palindrome sull'alfabeto $\{a, b\}$ nella notazione delle grammatiche libere dal contesto:

1. $S \to \epsilon$
2. $S \to a$
3. $S \to b$

Un esempio informale: dalla definizione ricorsiva alla grammatica

Le regole che definiscono le parole palindrome sull'alfabeto $\{a, b\}$ nella notazione delle grammatiche libere dal contesto:

1. $S \to \epsilon$
2. $S \to a$
3. $S \to b$
4. $S \to aSa$

Un esempio informale: dalla definizione ricorsiva alla grammatica

Le regole che definiscono le parole palindrome sull'alfabeto $\{a, b\}$ nella notazione delle grammatiche libere dal contesto:

1. $S \to \epsilon$
2. $S \to a$
3. $S \to b$
4. $S \to aSa$
5. $S \to bSb$

Definizione di una grammatica context-free

Definizione

Una grammatica libera dal contesto (o grammatica context-free, o CFG o grammatica) G, è una quadrupla

$$G = (V, T, P, S)$$

dove:

Definizione di una grammatica context-free

Definizione

Una grammatica libera dal contesto (o grammatica context-free, o CFG o grammatica) G, è una quadrupla

$$G = (V, T, P, S)$$

dove:

- V: **Insieme finito di variabili** *(dette anche non terminali o categorie sintattiche).*

Definizione di una grammatica context-free

Definizione

Una grammatica libera dal contesto (o grammatica context-free, o CFG o grammatica) G, è una quadrupla

$$G = (V, T, P, S)$$

dove:

- V: **Insieme finito di variabili** *(dette anche non terminali o categorie sintattiche).*
- T: **Insieme finito di simboli terminali** *(o alfabeto dei terminali).* $T \cap V = \emptyset$.

Definizione di una grammatica context-free

Definizione

Una grammatica libera dal contesto (o grammatica context-free, o CFG o grammatica) G, è una quadrupla

$$G = (V, T, P, S)$$

dove:

- V: **Insieme finito di variabili** *(dette anche non terminali o categorie sintattiche).*
- T: **Insieme finito di simboli terminali** *(o alfabeto dei terminali).* $T \cap V = \emptyset$.
- P: **Insieme finito delle produzioni (o regole)**.

Definizione di una grammatica context-free

Definizione

Una grammatica libera dal contesto (o grammatica context-free, o CFG o grammatica) G, è una quadrupla

$$G = (V, T, P, S)$$

dove:

- V: **Insieme finito di variabili** *(dette anche non terminali o categorie sintattiche).*
- T: **Insieme finito di simboli terminali** *(o alfabeto dei terminali).* $T \cap V = \emptyset$.
- P: **Insieme finito delle produzioni (o regole)**.
- S: *Una variabile, detta* **simbolo iniziale** *(o start symbol o variabile iniziale).*

Definizione di una grammatica context-free

Definizione

Ogni produzione consiste di tre parti.

Definizione di una grammatica context-free

Definizione

Ogni produzione consiste di tre parti.

1. *Una variabile (definita parzialmente dalla produzione), detta anche testa della produzione*

Definizione di una grammatica context-free

Definizione

Ogni produzione consiste di tre parti.

1. *Una variabile (definita parzialmente dalla produzione), detta anche testa della produzione*
2. *Il simbolo di produzione* $\rightarrow$

Definizione di una grammatica context-free

Definizione

Ogni produzione consiste di tre parti.

1. *Una variabile (definita parzialmente dalla produzione), detta anche testa della produzione*
2. *Il simbolo di produzione* $\rightarrow$
3. *Una stringa di zero o più terminali e variabili, detta anche corpo della produzione.*

- Ogni variabile rappresenta un linguaggio.

- Ogni variabile rappresenta un linguaggio.
- La variabile iniziale rappresenta il linguaggio della grammatica.

- Ogni variabile rappresenta un linguaggio.
- La variabile iniziale rappresenta il linguaggio della grammatica.
- Altre eventuali variabili rappresentano insiemi ausiliari di stringhe che contribuiscono a definire il linguaggio del simbolo iniziale.

- Ogni variabile rappresenta un linguaggio.
- La variabile iniziale rappresenta il linguaggio della grammatica.
- Altre eventuali variabili rappresentano insiemi ausiliari di stringhe che contribuiscono a definire il linguaggio del simbolo iniziale.
- L'insieme dei terminali è l'alfabeto delle parole del linguaggio da definire. Non ha elementi in comune con l'insieme delle variabili.

- Ogni variabile rappresenta un linguaggio.
- La variabile iniziale rappresenta il linguaggio della grammatica.
- Altre eventuali variabili rappresentano insiemi ausiliari di stringhe che contribuiscono a definire il linguaggio del simbolo iniziale.
- L'insieme dei terminali è l'alfabeto delle parole del linguaggio da definire. Non ha elementi in comune con l'insieme delle variabili.
- Le produzioni rappresentano la definizione ricorsiva del linguaggio della grammatica.

Esempio - Parole palindrome

Una grammatica G_{pal} per le parole palindrome sull'alfabeto $\{a, b\}$ è

$$G_{pal} = (\{S\}, \{a, b\}, P, S)$$

Esempio - Parole palindrome

Una grammatica G_{pal} per le parole palindrome sull'alfabeto $\{a, b\}$ è

$$G_{pal} = (\{S\}, \{a, b\}, P, S)$$

dove

$$P = \{S \to \epsilon, S \to a, S \to b, S \to aSa, S \to bSb\}$$

Esempio - Espressioni aritmetiche

Vogliamo definire una grammatica che “rappresenti”, semplificandole, le espressioni aritmetiche in un tipico linguaggio di programmazione;

Esempio - Espressioni aritmetiche

Vogliamo definire una grammatica che “rappresenti”, semplificandole, le espressioni aritmetiche in un tipico linguaggio di programmazione;

- ci limitiamo agli operatori $+$ e $*$, corrispondenti a somma e moltiplicazione;

Esempio - Espressioni aritmetiche

Vogliamo definire una grammatica che “rappresenti”, semplificandole, le espressioni aritmetiche in un tipico linguaggio di programmazione;

- ci limitiamo agli operatori $+$ e $*$, corrispondenti a somma e moltiplicazione;
- assumiamo che gli operandi siano identificatori;

Esempio - Espressioni aritmetiche

Vogliamo definire una grammatica che "rappresenti", semplificandole, le espressioni aritmetiche in un tipico linguaggio di programmazione;

- ci limitiamo agli operatori $+$ e $*$, corrispondenti a somma e moltiplicazione;
- assumiamo che gli operandi siano identificatori;
- limitiamo l'insieme degli identificatori alle stringhe sull'alfabeto $\{a, b, 0, 1\}$ che iniziano per a o per b.

Definizione ricorsiva - Espressioni aritmetiche

PASSO BASE: Un identificatore *I* è un'espressione aritmetica.

Definizione ricorsiva - Espressioni aritmetiche

PASSO BASE: Un identificatore I è un'espressione aritmetica.

PASSO RICORSIVO: se E_1 ed E_2 sono espressioni aritmetiche allora

$E_1 + E_2$ è un'espressione aritmetica,

$E_1 * E_2$ è un'espressione aritmetica,

(E_1) è un'espressione aritmetica.

Esempio - Espressioni aritmetiche

Un identificatore sarà un elemento del linguaggio

$$\{a, b\}\{a, b, 0, 1\}^*$$

Esempio - Espressioni aritmetiche

Un identificatore sarà un elemento del linguaggio

$$\{a, b\}\{a, b, 0, 1\}^*$$

Quindi un identificatore sarà un elemento dell'espressione regolare

$$(a + b)(a + b + 0 + 1)^*$$

Esempio - Espressioni aritmetiche

Un identificatore sarà un elemento del linguaggio

$$\{a, b\}\{a, b, 0, 1\}^*$$

Quindi un identificatore sarà un elemento dell'espressione regolare

$$(a + b)(a + b + 0 + 1)^*$$

Nota. Nelle espressioni regolari, alcuni autori usano $+$ per denotare l'operatore $\cup$.

Esempio - Espressioni aritmetiche

Un identificatore sarà un elemento del linguaggio

$$\{a, b\}\{a, b, 0, 1\}^*$$

Quindi un identificatore sarà un elemento dell'espressione regolare

$$(a + b)(a + b + 0 + 1)^*$$

Nota. Nelle espressioni regolari, alcuni autori usano $+$ per denotare l'operatore $\cup$.

Dobbiamo fare riferimento a una definizione ricorsiva degli elementi di

$$\{a, b\}\{a, b, 0, 1\}^*$$

Definizione ricorsiva - Identificatori su $\{a, b, 0, 1\}$

PASSO BASE: a e b sono identificatori.

Definizione ricorsiva - Identificatori su $\{a, b, 0, 1\}$

PASSO BASE: a e b sono identificatori.

PASSO RICORSIVO: se I è un identificatore allora

Ia è un identificatore,

Ib è un identificatore,

$I0$ è un identificatore,

$I1$ è un identificatore.

Esempio - Espressioni

Una grammatica che “rappresenti”, semplificandole, le espressioni aritmetiche in un tipico linguaggio di programmazione.

Esempio - Espressioni

Una grammatica che “rappresenti”, semplificandole, le espressioni aritmetiche in un tipico linguaggio di programmazione.

Avremo bisogno di due variabili:
- E per le espressioni,
- I per gli identificatori.

Esempio - Espressioni

- Le produzioni:

Esempio - Espressioni

- Le produzioni:
  1. $E \rightarrow I$

Esempio - Espressioni

- Le produzioni:
  1. $E \rightarrow I$
  2. $E \rightarrow E + E$

Esempio - Espressioni

- Le produzioni:
  1. $E \rightarrow I$
  2. $E \rightarrow E + E$
  3. $E \rightarrow E * E$

Esempio - Espressioni

- Le produzioni:
  1. $E \rightarrow I$
  2. $E \rightarrow E + E$
  3. $E \rightarrow E * E$
  4. $E \rightarrow (E)$

Esempio - Espressioni

1. $I \rightarrow a$

Esempio - Espressioni

1. $I \rightarrow a$
2. $I \rightarrow b$

Esempio - Espressioni

1. $l \rightarrow a$
2. $l \rightarrow b$
3. $l \rightarrow la$

Esempio - Espressioni

1. $I \rightarrow a$
2. $I \rightarrow b$
3. $I \rightarrow Ia$
4. $I \rightarrow Ib$

Esempio - Espressioni

1. $I \rightarrow a$
2. $I \rightarrow b$
3. $I \rightarrow Ia$
4. $I \rightarrow Ib$
5. $I \rightarrow I0$

Esempio - Espressioni

1. $I \rightarrow a$
2. $I \rightarrow b$
3. $I \rightarrow Ia$
4. $I \rightarrow Ib$
5. $I \rightarrow I0$
6. $I \rightarrow I1$

Esempio - Espressioni

Una CFG per espressioni aritmetiche:

$$G_E = (\{E, I\}, T, P, E)$$

Esempio - Espressioni

Una CFG per espressioni aritmetiche:

$$G_E = (\{E, I\}, T, P, E)$$

dove

$$T = \{+, *, (,), a, b, 0, 1\}$$

Esempio - Espressioni

Una CFG per espressioni aritmetiche:

$$G_E = (\{E, I\}, T, P, E)$$

dove

$$T = \{+, *, (,), a, b, 0, 1\}$$

e

$P = \{E \to I,\ E \to E + E,\ E \to E * E,\ E \to (E),\ I \to a,$
$I \to b,\ I \to Ia,\ I \to Ib,\ I \to I0,\ I \to I1\}$

Notazione compatta per le produzioni

Invece di

$$A \to \alpha_1, A \to \alpha_2, \ldots, A \to \alpha_n$$

Notazione compatta per le produzioni

Invece di

$$A \to \alpha_1, A \to \alpha_2, \ldots, A \to \alpha_n$$

possiamo scrivere

$$A \to \alpha_1 | \alpha_2 | \ldots | \alpha_n$$

Notazione compatta per le produzioni

Invece di

$$A \to \alpha_1, A \to \alpha_2, \ldots, A \to \alpha_n$$

possiamo scrivere

$$A \to \alpha_1 | \alpha_2 | \ldots | \alpha_n$$

Esempio. In $G_{pal} = (\{S\}, \{a, b\}, P, S)$,

$$P = \{S \to \epsilon \mid a \mid b \mid aSa \mid bSb\}$$

Derivazione

Applichiamo le produzioni di una grammatica per dedurre che alcune parole sono nel linguaggio di una variabile.

Derivazione

Applichiamo le produzioni di una grammatica per dedurre che alcune parole sono nel linguaggio di una variabile.

La deduzione può seguire due strade:

Applichiamo le produzioni di una grammatica per dedurre che alcune parole sono nel linguaggio di una variabile.

La deduzione può seguire due strade:

1. usare l'inferenza ricorsiva (dal corpo alla testa)

Applichiamo le produzioni di una grammatica per dedurre che alcune parole sono nel linguaggio di una variabile.

La deduzione può seguire due strade:

1. usare l'inferenza ricorsiva (dal corpo alla testa)
2. usare la derivazione (dalla testa al corpo)

Applichiamo le produzioni di una grammatica per dedurre che alcune parole sono nel linguaggio di una variabile.

La deduzione può seguire due strade:

1. usare l'inferenza ricorsiva (dal corpo alla testa)
2. usare la derivazione (dalla testa al corpo)

Non tratteremo l'inferenza ricorsiva.

Definizione

Sia $G = (V, T, P, S)$ *una grammatica context-free.*

Definizione

Sia $G = (V, T, P, S)$ *una grammatica context-free.*

Sia $\alpha A \beta \in (V \cup T)^*$, *con* $A \in V$ *e* $\alpha, \beta \in (V \cup T)^*$,

Sia $A \to \gamma \in P$.

Definizione

Sia $G = (V, T, P, S)$ *una grammatica context-free.*

Sia $\alpha A \beta \in (V \cup T)^*$, *con* $A \in V$ *e* $\alpha, \beta \in (V \cup T)^*$,
Sia $A \to \gamma \in P$.

Allora diremo che

$$\alpha A \beta \text{ deriva (direttamente) } \alpha \gamma \beta \text{ in } G$$

e scriveremo

$$\alpha A \beta \underset{G}{\Rightarrow} \alpha \gamma \beta$$

Definizione

Sia $G = (V, T, P, S)$ *una grammatica context-free.*

Sia $\alpha A \beta \in (V \cup T)^*$, *con* $A \in V$ *e* $\alpha, \beta \in (V \cup T)^*$,
Sia $A \to \gamma \in P$.

Allora diremo che

$$\alpha A \beta \text{ deriva (direttamente) } \alpha \gamma \beta \text{ in } G$$

e scriveremo

$$\alpha A \beta \underset{G}{\Rightarrow} \alpha \gamma \beta$$

o, se G *è chiara dal contesto, semplicemente*

$$\alpha A \beta \Rightarrow \alpha \gamma \beta$$

Derivazione diretta - Esempi

Quindi $\alpha A \beta$ deriva direttamente $\alpha \gamma \beta$ se $A \to \gamma \in P$, con α, β stringhe di variabili e terminali.

Derivazione diretta - Esempi

Quindi $\alpha A \beta$ deriva direttamente $\alpha \gamma \beta$ se $A \to \gamma \in P$, con α, β stringhe di variabili e terminali.

Nella grammatica definita dalle regole
$S \to \epsilon \mid a \mid b \mid aSa \mid bSb$

Derivazione diretta - Esempi

Quindi $\alpha A \beta$ deriva direttamente $\alpha \gamma \beta$ se $A \to \gamma \in P$, con α, β stringhe di variabili e terminali.

Nella grammatica definita dalle regole
$S \to \epsilon \mid a \mid b \mid aSa \mid bSb$

- $S \Rightarrow aSa$, con $\alpha = \beta = \epsilon$

Derivazione diretta - Esempi

Quindi $\alpha A \beta$ deriva direttamente $\alpha \gamma \beta$ se $A \to \gamma \in P$, con α, β stringhe di variabili e terminali.

Nella grammatica definita dalle regole
$S \to \epsilon \mid a \mid b \mid aSa \mid bSb$

- $S \Rightarrow aSa$, con $\alpha = \beta = \epsilon$
- $aaSaa \Rightarrow aabSbaa$, con $\alpha = \beta = aa$

Derivazione diretta - Esempi

Quindi $\alpha A \beta$ deriva direttamente $\alpha\gamma\beta$ se $A \to \gamma \in P$, con α, β stringhe di variabili e terminali.

Nella grammatica definita dalle regole
$S \to \epsilon \mid a \mid b \mid aSa \mid bSb$

- $S \Rightarrow aSa$, con $\alpha = \beta = \epsilon$
- $aaSaa \Rightarrow aabSbaa$, con $\alpha = \beta = aa$
- $abSba \Rightarrow abba$, con $\alpha = ab,\ \beta = ba$

Derivazione diretta - Esempi

$\alpha A \beta$ deriva direttamente $\alpha \gamma \beta$ se $A \to \gamma \in P$, con α, β stringhe di variabili e terminali.

Derivazione diretta - Esempi

$\alpha A \beta$ deriva direttamente $\alpha \gamma \beta$ se $A \to \gamma \in P$, con α, β stringhe di variabili e terminali.

Nella grammatica definita dalle regole
$E \to E + E \mid E * E \mid (E) \mid I$,
$I \to a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

Derivazione diretta - Esempi

$\alpha A \beta$ deriva direttamente $\alpha\gamma\beta$ se $A \to \gamma \in P$, con α, β stringhe di variabili e terminali.

Nella grammatica definita dalle regole
$E \to E + E \mid E * E \mid (E) \mid I$,
$I \to a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

- $E \Rightarrow I$, con $\alpha = \beta = \epsilon$

Derivazione diretta - Esempi

$\alpha A \beta$ deriva direttamente $\alpha\gamma\beta$ se $A \to \gamma \in P$, con α, β stringhe di variabili e terminali.

Nella grammatica definita dalle regole
$E \to E + E \mid E * E \mid (E) \mid I$,
$I \to a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

- $E \Rightarrow I$, con $\alpha = \beta = \epsilon$
- $E + E \Rightarrow E + E * E$, con $\alpha = E+,\ \beta = \epsilon$

Derivazione diretta - Esempi

$\alpha A \beta$ deriva direttamente $\alpha\gamma\beta$ se $A \to \gamma \in P$, con α, β stringhe di variabili e terminali.

Nella grammatica definita dalle regole
$E \to E + E \mid E * E \mid (E) \mid I$,
$I \to a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

- $E \Rightarrow I$, con $\alpha = \beta = \epsilon$
- $E + E \Rightarrow E + E * E$, con $\alpha = E+,\ \beta = \epsilon$
- $a * (E + E) \Rightarrow a * (I + E)$, con $\alpha = a * (,\ \beta = +E)$

Derivazione - Esempio

Nella grammatica definita dalle regole
$S \to \epsilon \mid a \mid b \mid aSa \mid bSb$ abbiamo

Derivazione - Esempio

Nella grammatica definita dalle regole
$S \to \epsilon \mid a \mid b \mid aSa \mid bSb$ abbiamo

$S \Rightarrow aSa$, $aSa \Rightarrow abSba$, $abSba \Rightarrow abba$

Derivazione - Esempio

Nella grammatica definita dalle regole
$S \to \epsilon \mid a \mid b \mid aSa \mid bSb$ abbiamo

$S \Rightarrow aSa$, $aSa \Rightarrow abSba$, $abSba \Rightarrow abba$

Definiremo una nuova operazione:

$$S \stackrel{*}{\Rightarrow} abba$$

Estendiamo la relazione $\Rightarrow$ per rappresentare zero, uno o più passi di derivazione.

Estendiamo la relazione $\Rightarrow$ per rappresentare zero, uno o più passi di derivazione.

Definizione

Estendiamo la relazione $\Rightarrow$ per rappresentare zero, uno o più passi di derivazione.

Definizione

- **BASE:** Ogni stringa deriva se stessa

$$\forall \alpha \in (V \cup T)^* \quad \alpha \overset{*}{\underset{G}{\Rightarrow}} \alpha$$

Estendiamo la relazione $\Rightarrow$ per rappresentare zero, uno o più passi di derivazione.

Definizione

- **BASE:** Ogni stringa deriva se stessa

$$\forall \alpha \in (V \cup T)^* \quad \alpha \overset{*}{\underset{G}{\Rightarrow}} \alpha$$

- **INDUZIONE:**

$\forall \alpha, \beta, \gamma \in (V \cup T)^*$, se α deriva β e β deriva (direttamente) γ, allora α deriva γ:

$$\text{se } \alpha \overset{*}{\underset{G}{\Rightarrow}} \beta \text{ e } \beta \underset{G}{\Rightarrow} \gamma \text{ allora}$$

$$\alpha \overset{*}{\underset{G}{\Rightarrow}} \gamma$$

In altri termini α deriva β in G, $\alpha \overset{*}{\underset{G}{\Rightarrow}} \beta$, se esistono parole $\gamma_1, \ldots, \gamma_n$, $n \geq 1$, tali che

In altri termini α deriva β in G, $\alpha \overset{*}{\underset{G}{\Rightarrow}} \beta$, se esistono parole $\gamma_1, \ldots, \gamma_n$, $n \geq 1$, tali che

1. $\alpha = \gamma_1$,

In altri termini α deriva β in G, $\alpha \overset{*}{\underset{G}{\Rightarrow}} \beta$, se esistono parole $\gamma_1, \ldots, \gamma_n$, $n \geq 1$, tali che

1. $\alpha = \gamma_1$,
2. $\beta = \gamma_n$,

In altri termini α deriva β in G, $\alpha \overset{*}{\underset{G}{\Rightarrow}} \beta$, se esistono parole $\gamma_1, \ldots, \gamma_n$, $n \geq 1$, tali che

1. $\alpha = \gamma_1$,
2. $\beta = \gamma_n$,
3. Per $i = 1, 2, \ldots, n-1$, si ha $\gamma_i \underset{G}{\Rightarrow} \gamma_{i+1}$

In altri termini α deriva β in G, $\alpha \overset{*}{\underset{G}{\Rightarrow}} \beta$, se esistono parole $\gamma_1, \ldots, \gamma_n$, $n \geq 1$, tali che

1. $\alpha = \gamma_1$,
2. $\beta = \gamma_n$,
3. Per $i = 1, 2, \ldots, n-1$, si ha $\gamma_i \underset{G}{\Rightarrow} \gamma_{i+1}$

In questo caso diremo anche che α deriva β in $n-1$ passi.

In altri termini α deriva β in G, $\alpha \overset{*}{\underset{G}{\Rightarrow}} \beta$, se esistono parole $\gamma_1, \ldots, \gamma_n$, $n \geq 1$, tali che

1. $\alpha = \gamma_1$,
2. $\beta = \gamma_n$,
3. Per $i = 1, 2, \ldots, n-1$, si ha $\gamma_i \underset{G}{\Rightarrow} \gamma_{i+1}$

In questo caso diremo anche che α deriva β in $n-1$ passi.

Ovviamente G è omessa se è chiara dal contesto.

Derivazione - Esempio

Mostriamo come E deriva $a * (a + b00)$ nella grammatica $G_E = (\{E, I\}, T, P, E)$ per espressioni aritmetiche dove

$$T = \{+, *, (,), a, b, 0, 1\}$$

e

$$\begin{aligned} P \quad = \quad & \{E \to E + E \mid E * E \mid (E) \mid I, \\ & I \to a \mid b \mid Ia \mid Ib \mid I0 \mid I1\} \end{aligned}$$

cioè

$$E \stackrel{*}{\Rightarrow} a * (a + b00)$$

Derivazione - Esempio

$$E \Rightarrow E * E \Rightarrow I * E \Rightarrow a * E \Rightarrow a * (E) \Rightarrow$$

$$a * (E + E) \Rightarrow a * (I + E) \Rightarrow a * (a + E) \Rightarrow a * (a + I) \Rightarrow$$

$$a * (a + I0) \Rightarrow a * (a + I00) \Rightarrow a * (a + b00)$$

Derivazione - Esempio

Nell'esempio precedente abbiamo scelto di sostituire sempre la variabile più a sinistra. Ma questo non è necessario.

Derivazione - Esempio

Nell'esempio precedente abbiamo scelto di sostituire sempre la variabile più a sinistra. Ma questo non è necessario.

Una derivazione corretta:

Nell'esempio precedente abbiamo scelto di sostituire sempre la variabile più a sinistra. Ma questo non è necessario.

Una derivazione corretta:

$$E \Rightarrow E * E \Rightarrow E * (E)$$

Derivazione - Esempio

Nell'esempio precedente abbiamo scelto di sostituire sempre la variabile più a sinistra. Ma questo non è necessario.

Una derivazione corretta:

$$E \Rightarrow E * E \Rightarrow E * (E)$$

Possiamo anche fare scelte che non condurranno alla stessa stringa di terminali:

$$E \Rightarrow E + E$$

Derivazioni a sinistra

In una derivazione a sinistra (derivazione canonica sinistra, derivazione leftmost) si impone che ad ogni passo di derivazione si sostituisca la variabile più a **sinistra** con il corpo di una delle sue produzioni.

Derivazioni a sinistra

In una derivazione a sinistra (derivazione canonica sinistra, derivazione leftmost) si impone che ad ogni passo di derivazione si sostituisca la variabile più a **sinistra** con il corpo di una delle sue produzioni.

Simboli utilizzati:

$\underset{lm}{\Rightarrow}$ per un passo, $\overset{*}{\underset{lm}{\Rightarrow}}$ per zero, uno o più passi

Negli esempi seguenti ci riferiremo alla grammatica $G_E = (\{E, I\}, T, P, E)$ per espressioni aritmetiche dove

$$T = \{+, *, (,), a, b, 0, 1\}$$

e

$$\begin{aligned} P \quad = \quad & \{E \to E + E \mid E * E \mid (E) \mid I, \\ & I \to a \mid b \mid Ia \mid Ib \mid I0 \mid I1\} \end{aligned}$$

Derivazione a sinistra - Esempio

$$E \underset{lm}{\Rightarrow} E * E \underset{lm}{\Rightarrow} I * E \underset{lm}{\Rightarrow} a * E \underset{lm}{\Rightarrow} a * (E) \underset{lm}{\Rightarrow}$$

$$a * (E + E) \underset{lm}{\Rightarrow} a * (I + E) \underset{lm}{\Rightarrow} a * (a + E) \underset{lm}{\Rightarrow} a * (a + I) \underset{lm}{\Rightarrow}$$

$$a * (a + I0) \underset{lm}{\Rightarrow} a * (a + I00) \underset{lm}{\Rightarrow} a * (a + b00)$$

Derivazioni a destra

Analogamente, in una derivazione a destra (derivazione canonica destra, derivazione rightmost) si impone che ad ogni passo di derivazione si sostituisca la variabile più a **destra** con il corpo di una delle sue produzioni.

Derivazioni a destra

Analogamente, in una derivazione a destra (derivazione canonica destra, derivazione rightmost) si impone che ad ogni passo di derivazione si sostituisca la variabile più a **destra** con il corpo di una delle sue produzioni.

Simboli utilizzati:

$\underset{rm}{\Rightarrow}$ per un passo, $\overset{*}{\underset{rm}{\Rightarrow}}$ per zero, uno o più passi

Derivazioni a destra

Analogamente, in una derivazione a destra (derivazione canonica destra, derivazione rightmost) si impone che ad ogni passo di derivazione si sostituisca la variabile più a **destra** con il corpo di una delle sue produzioni.

Simboli utilizzati:

$\underset{rm}{\Rightarrow}$ per un passo, $\overset{*}{\underset{rm}{\Rightarrow}}$ per zero, uno o più passi

Se la grammatica non è chiara dal contesto, il suo nome può apparire sotto i simboli.

Derivazione a destra- Esempio

$$E \underset{rm}{\Rightarrow} E * E \underset{rm}{\Rightarrow} E * (E) \underset{rm}{\Rightarrow} E * (E + E) \underset{rm}{\Rightarrow} E * (E + I) \underset{rm}{\Rightarrow}$$

$$E * (E + I0) \underset{rm}{\Rightarrow} E * (E + I00) \underset{rm}{\Rightarrow} E * (E + b00) \underset{rm}{\Rightarrow}$$

$$E * (I + b00) \underset{rm}{\Rightarrow} E * (a + b00) \underset{rm}{\Rightarrow} I * (a + b00) \underset{rm}{\Rightarrow} a * (a + b00)$$

Derivazione a sinistra - Esempio

Due derivazioni a sinistra diverse della stessa stringa di terminali in G_E.

$$E \underset{lm}{\Rightarrow} E * E \underset{lm}{\Rightarrow} E + E * E \underset{lm}{\Rightarrow} I + E * E \underset{lm}{\Rightarrow} a + E * E \underset{lm}{\Rightarrow}$$
$$a + I * E \underset{lm}{\Rightarrow} a + a * E \underset{lm}{\Rightarrow} a + a * I \underset{lm}{\Rightarrow} a + a * a$$

$$E \underset{lm}{\Rightarrow} E + E \underset{lm}{\Rightarrow} I + E \underset{lm}{\Rightarrow} a + E \underset{lm}{\Rightarrow} a + E * E \underset{lm}{\Rightarrow}$$
$$a + I * E \underset{lm}{\Rightarrow} a + a * E \underset{lm}{\Rightarrow} a + a * I \underset{lm}{\Rightarrow} a + a * a$$

Derivazione a sinistra

Vedremo a breve che la nozione di derivazione a sinistra (e la sua duale di derivazione a destra) è importante per caratterizzare quelle grammatiche che esprimono in modo non ambiguo l'intuizione che abbiamo rispetto a come vanno eseguiti i programmi.

Derivazione a sinistra

Vedremo a breve che la nozione di derivazione a sinistra (e la sua duale di derivazione a destra) è importante per caratterizzare quelle grammatiche che esprimono in modo non ambiguo l'intuizione che abbiamo rispetto a come vanno eseguiti i programmi.

Supponiamo che nella stringa $a + a * a$ gli identificatori rappresentino valori numerici. Noi vorremmo che venisse valutata prima l'espressione $a * a$ e poi l'espressione $a + a * a$.

Derivazione a sinistra

Vedremo a breve che la nozione di derivazione a sinistra (e la sua duale di derivazione a destra) è importante per caratterizzare quelle grammatiche che esprimono in modo non ambiguo l'intuizione che abbiamo rispetto a come vanno eseguiti i programmi.

Supponiamo che nella stringa $a + a * a$ gli identificatori rappresentino valori numerici. Noi vorremmo che venisse valutata prima l'espressione $a * a$ e poi l'espressione $a + a * a$.

La grammatica G_E non sembra essere adatta a questo scopo.

Definizione

Sia $G = (V, T, P, S)$ *una grammatica context free.*

Il **linguaggio di** G *è*

$$L(G) = \{w \in T^* \mid S \overset{*}{\underset{G}{\Rightarrow}} w\}$$

Nota: Le parole di $L(G)$ sono tutte e sole le stringhe di caratteri terminali per le quali esiste una derivazione dal simbolo iniziale.

Il linguaggio di una CFG - Esempio

Sia $G_{pal} = (\{S\}, \{a, b\}, P, S)$, con

$$P = \{S \to \epsilon, S \to a, S \to b, S \to aSa, S \to bSb\}.$$

Il linguaggio della grammatica G_{pal} è

$$L(G_{pal}) = \{w \in \{a, b\}^* \mid w = w^R\}$$

Definizione

Un linguaggio $L \subseteq T^*$ è **context-free** *se esiste una grammatica context free $G = (V, T, P, S)$ tale che*

$$L = L(G)$$

Il linguaggio

$$L = \{w \in \{a, b\}^* \mid w = w^R\}$$

è context-free perché $L = L(G_{pal})$, dove G_{pal} è la grammatica context-free definita dalle produzioni

$$S \to \epsilon \mid a \mid b \mid aSa \mid bSb$$

Nota: Per provare che L è context free, occorre:

1. Definire una grammatica G (con alfabeto dei terminali uguale all'alfabeto di L).

Nota: Per provare che L è context free, occorre:

1. Definire una grammatica G (con alfabeto dei terminali uguale all'alfabeto di L).
2. Provare che se $w \in L$ allora esiste una derivazione in G di w dal simbolo iniziale di G.

Nota: Per provare che L è context free, occorre:

1. Definire una grammatica G (con alfabeto dei terminali uguale all'alfabeto di L).
2. Provare che se $w \in L$ allora esiste una derivazione in G di w dal simbolo iniziale di G.
3. Provare che se $S \overset{*}{\underset{G}{\Rightarrow}} w$, dove w è una stringa di terminali ed S è il simbolo iniziale di G, allora $w \in L$.