28/09/2022

INGEGNERIA DEL SOFTWARE

Committente

Problema → Definito → Soluzione → Codifica → Testing

ALGO/REQ. FUNZIONALE o.m

Problema → Vincoli

Codifica: implemento Linguaggio di programmazione

Riuso → Testato

Riuso ↓ Concetti di open source e open science

LICENZE

Paradigma imperativo	P. a oggetti	P. logico	P. funzionale
Linguaggio di Programmazione ad alto livello			
Assembler			
Linguaggio macchina			
		CODICE BW	

Hardware

Compilato: compila tutto contemporaneamente

$P_{LAL} \rightarrow \boxed{C} \rightarrow P_{LM}$

Compilatore ad alto livello — Linguaggio macchina

Interpretato: compila riga per riga

$P_{LAL} \rightarrow \boxed{I} \rightarrow$ Eseguiti

Alfabeto e Grammatica

Simbolo lambda = Λ

Λ indica un alfabeto. Esempio: $\Lambda = \{0,1,2,\dots,9\}$

Λ^* contiene la stringa vuota indicata con ε (epsilon) e tutte le possibili stringhe.

$L \subseteq \Lambda$ dove L sta per linguaggio, un sottoinsieme di Λ^*

$LIN = \Lambda^+$ il + indica tutte le possibili stringhe incluse in Λ^* tranne ε.

Esempio

$\Lambda = \{0, 1, 2, \dots, 9, +, -, \times, \div\}$

$\Lambda^* = \{\varepsilon, 012, 0516, 1782, \dots 0+\times 9\div, \dots \underbrace{1+9\times 3}_{\underline{28}}, \dots\}$

$L_1 \subseteq \Lambda^*$

$L_2 \subseteq \Lambda^*$

$L_3 = L_1 \cup L_2 = \{s \mid s \in L_1 \vee s \in L_2\}$

Il trattino basso (_) indica la semantica.

Grammatica libera da contesto

La grammatica applicativa permette di utilizzare simboli per generare stringhe corrette.

Con la grammatica si possono generare espressioni aritmetiche in modo ricorsivo

Esempio

① I quattro operatori: $+, -, \times, /$

② Le parentesi (,)

③ I numeri

} Simboli Terminali

Base: un numero è un'espressione
Induzione: se E è un'espressione, lo sono anche:
(E), E+E, E-E, E×E, E/E

R_1 <Espressione> → <numero>

R_2 <Espressione> → (<Espressione>)

R_3 <Espressione> → <Espressione> + <Espressione>

R_4 <Espressione> → <Espressione> - <Espressione>

R_5 <Espressione> → <Espressione> × <Espressione>

R_6 <Espressione> → <Espressione> / <Espressione>

(3+5)+7

<Espressione> $\xrightarrow[R_3]{}$ <Espressione> + <Espressione>

<Espressione> + <Espressione> $\xrightarrow[R_2]{}$ (<Espressione>) + <Espressione>

(<Espressione>) + <Espressione> $\xrightarrow[R_3]{}$ (<Espressione> + <Espressione>) + <Espressione>

Definizioni

Simboli terminali → simboli dell'alfabeto Δ

Simboli non terminali o categorie sintattiche → <Espressione>, <cifra>, <numero> definiti da noi

Produzioni o regole generative → insieme delle regole

La produzione è **diviso in**

Testa → categoria sintattica

→ → metasimbolo

Corpo → ∅ o più cat. sint. e/o simboli term.

Esempio

Base: <cifra> → 0 | 1 | 2 | 3 | ... | 9 |

Induzione: <numero> → <cifra> | <numero><cifra>

Convenzioni e notazioni

1. Le categorie sintattiche vengono indicate con lettere maiuscole e tra <
2. Con | (OR) si uniscono più regole
3. Si possono rimuovere le <> se non è ambiguo

Definizione di Grammatica

Una grammatica G è definita come una quadrupla:

$$< \Delta, V, S, P >$$

dove

Δ: è l'alfabeto

V: è l'insieme finito delle categorie sintattiche, ovvero variabili che rappresentano sottolinguaggi

$S \in V$: S è una categoria sintattica principale o iniziale

P: è un insieme finito di produzioni (regole).

Ciascuna produzione ha la struttura $A \rightarrow \alpha$ dove $A \in V$ e $\alpha \in (\Delta \cup V)^+$

05/10/2022

Definire una grammatica

$\Lambda_{espressione} = \{ *, -, +, /, 0, 1, \dots, 9 \}$

$V_{espressione} = \{ \langle Espressione \rangle, \langle Numero \rangle, \langle Cifra \rangle \}$

$S_{espressione} = \{ \langle Espressione \rangle \}$

Esercizi

① $L(G_1) = \{ a^n \mid n \geq 1 \} = \{ a, aa, aaa, \dots \}$

$S \rightarrow a \mid aS$

$G_1 \langle \Lambda, V, S, P \rangle$

$\Lambda = \{a\}$ $V = \{S\}$ $S = \{S\}$ $P = \{ S \rightarrow a \mid aS \}$

② $L(G_2) = \{ a^n \mid n \geq 0 \} = \{ \varepsilon, a, aa, aaa, \dots \}$

$S \rightarrow \varepsilon \mid \cancel{a} \mid aS$

elimino l'ambiguità

G_2 : $\Lambda = \{\varepsilon, a\}$ $V = \{S\}$

$S = \{S\}$ $P = \{ S \rightarrow \varepsilon \mid aS \}$

③ $L(G_3) = \{ a^n b^m \mid n \geq 1, m \geq 0 \}$

$S \rightarrow a \mid ab \mid aS \mid bS$

④ $L(G_4) = \{ a^n b^n \mid n \geq 1 \}$

$S \rightarrow ab \mid aSb$

06/10/2022

Esercizio 3

$$L(G_3) = \{a^n b^m \mid n \geq 1, m \geq 0\}$$

$S \rightarrow a \mid aS \mid aC$

$C \rightarrow \varepsilon \mid bC$

Esercizio 5

$$L(G_5) = \{a^n b^m \mid n \geq 0, m \geq 0\}$$

$S \rightarrow aS \mid C$

$C \rightarrow \varepsilon \mid bc \mid \beta bc$

Esercizio 6

$$L(G_6) = \{a^n b^m c^n \mid n \geq 0, m \geq 0\}$$

$S \rightarrow aSc \mid B$

$B \rightarrow \varepsilon \mid bB$

12/10/2022

Ripasso

La grammatica è uno strumento generativo (utilizza la ricorsione)
Ha un passo base e una induzione

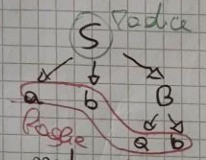
$G = \langle \Delta, V, S, P \rangle$

Δ → definizione linguaggio

S → A

V:
A → … | B
B → … | C
C → …

Albero di derivazione



$S \to abB$
$B \to ab$

Sotto ogni nodo un simbolo terminale o non terminale

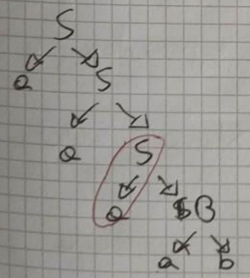
Una grammatica è ambigua se si possono costruire due alberi di derivazione diversi che generano la stessa stringa

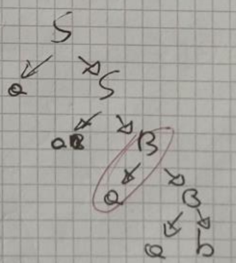
Esempio

$B \to \varepsilon \mid ab \mid aB$
$S \to aB \mid aS$

$L = \left\{ a^n b^m \mid \begin{matrix} m \in \{0,1\} \\ n \geq 1 \end{matrix} \right\}$

Genero aaaab





Parentesi Bilanciate

Stringa in cui si apro parentesi, la chiudo ( ).

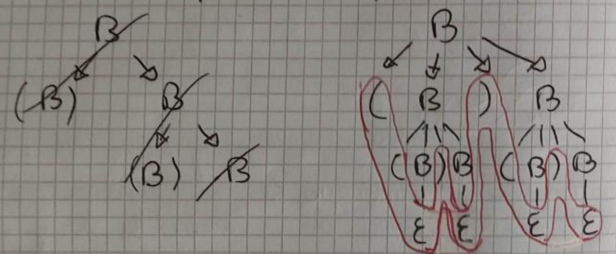
Base: $S \to \varepsilon$ $S \to (S)S$

x y (x) (x)y
(y)

<Bilanciata> → ε | (<Bilanciata>) <Bilanciata>

Tuttavia, ~~è ambigua~~ la generazione è non deterministica ma non è ambigua ~~<Bilanciata>~~

stringa da rappresentare (())()

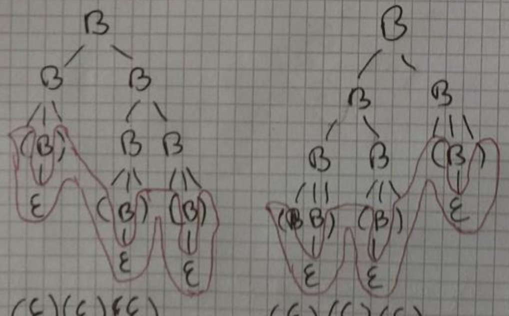


~~l'ambiguità non è determinata dal non determinismo (regola)~~.

Grammatica ambigua per parentesi bilanciate

Esempio

$B \to (B) \mid BB \mid \varepsilon$ stringa da rappresentare ()()()



(ε)(ε)(ε) (ε)(ε)(ε)

Stesso risultato ma alberi diversi
↓
ambigua

~~WHILE~~: Definizione del linguaggio $L(\langle S \rangle)$ associato alla categoria sintattica $\langle S \rangle$

Passo Base: ~~$S \to \varepsilon$~~

per ogni categoria sintattica $\langle S \rangle$ della grammatica, il linguaggio $L(\langle S \rangle)$ sia vuoto

Passo induttivo:

Supponiamo che la grammatica contenga la produzione

$\langle S \rangle \to X_1 X_2 \dots X_m$

in cui ogni simbolo X_i per $i = 1, \dots, m$

è una categoria sintattica oppure un simbolo terminale

Per ciascun $i = 1, \dots, m$ si seleziona una stringa s_i per X_i nel modo seguente:

1. se X_i è un simbolo terminale, allora la stringa s_i è X_i stesso,
2. Se X_i è una categoria sintattica, allora s_i è una qualunque stringa appartenente a $L(x_i)$.
   Se la stessa categoria sintattica x_i appare più volte nel corpo della produzione, si può scegliere da $L(x_i)$ una stringa diversa per ogni occorrenza di X_i

Allora, la concatenazione $s_1, s_2, \dots s_m$ delle stringhe appartiene al linguaggio $L(\langle S \rangle)$. Se $m = 0$ allora abbiamo la stringa vuota $\varepsilon \in L(\langle S \rangle)$

Esempio: Ciclo WHILE

Comando

<Comando> $\to$ while <Condizione> do <Comando>

<Comando> $\to$ if <Condizione> then <Comando>

<Comando> $\to$ if <Condizione> then <Comando> else <Comando>

<Comando> $\to$ begin <ListaComm> end

<Comando> $\to$ <Comando Semplice>

Lista Comandi (c.s. principale)

<Lista Comandi> $\to$ <ListaComandi> ; <Comando>

<ListaComandi> $\to$ <Comando>

13/10/2022

Sistema di transizioni

Un sistema di transizioni S è una tupla

$$<\Gamma, \underset{\text{gamma}}{T}, \rightarrow>$$

dove

- Γ è un insieme i cui elementi sono detti configurazioni
- $T \subseteq \Gamma$ è un sottoinsieme di Γ i cui elementi sono detti configurazioni terminali
- $\rightarrow$ è un insieme di coppie (γ, γ') di configurazioni viene detta relazione di transizione.

Denoteremo con $\gamma \rightarrow \gamma'$ l'appartenenza della coppia $<\gamma, \gamma'>$ alla relazione freccia e chiameremo transizioni tali elementi

Definizione

Dato un sistema di transizioni $S = <\Gamma, T, \rightarrow>$, una derivazione in S è una sequenza (possibilmente infinita) di configurazioni $\gamma_0, \gamma_1, \gamma_2, \dots \gamma_{i-1}, \gamma_i, \dots$

Tale che per ogni $k \geq 1$ abbiamo che $\gamma_{k-1} \rightarrow \gamma_k$ sono in relazione di transizione.

Nel seguito, una derivazione in S viene indicata

$$\gamma_0 \rightarrow \gamma_1 \rightarrow \gamma_2 \rightarrow \dots \rightarrow \gamma_{i-1} \rightarrow \gamma_i \rightarrow \dots$$

Indicheremo con $\gamma \xrightarrow{*} \gamma'$ una derivazione $\gamma_0 \rightarrow \gamma_1 \rightarrow \dots \rightarrow \gamma_i$ in cui $\gamma_0 = \gamma$ e $\gamma' = \gamma_i$

Dato un sistema di transizioni $S = <\Gamma, T, \rightarrow>$ diremo che una configurazione $\gamma \in \Gamma \setminus T$ è bloccata se e solo se non esiste alcuna configurazione γ' t.c. $\gamma \rightarrow \gamma'$

Esempio

Sia $G = <\Delta, V, S, P>$ una grammatica libera da contesto

Definiamo un sis. trans. S_G associato alla grammatica nel seguente modo:

- le configurazioni S_G sono stringhe in $(\Delta \cup V)^*$
- le configurazioni terminali di G $T = \Delta^*$
- dette α e β due configurazioni esiste una transizione $\alpha \rightarrow \beta$ se e

$G = \langle \Delta, V, S, P \rangle$ Esempio

$L = \{a^n \mid n \geq 1\}$ $\Delta = \{a\}$

$S \to a \mid aS$

aaa

$\underset{\gamma}{S} \to aS \to aaS \to \underset{\gamma''}{aaa}$

$\gamma \xrightarrow{*} \gamma'$

solo se:

- α contiene almeno un simbolo non terminale α del tipo $\delta A \gamma$ con $\delta, \gamma \in (\Delta \cup V)^*$ e $A \in V$
- β è del tipo $\delta \eta \gamma$
- $A ::= \eta$ è una produzione di P

È evidente, per costruzione del nostro sistema di transizioni S_G che $\alpha \in L(G)$ se e solo se

simbolo iniziale della grammatica $S \xrightarrow[G]{*} \alpha$

Regole Condizionali

$$\frac{\pi_1\ \pi_2 \ldots \pi_n}{\gamma \to \gamma'}$$

if voto ≥ 18
then promosso = 1
else promosso = -1

η = eta

$$\frac{\text{voto} < 18}{\gamma \to \gamma''}$$

$$\frac{\alpha = \delta A \gamma,\ \ \delta, \gamma \in (\Delta \cup V)^*\ \ \beta = \delta \eta \gamma}{\alpha \xrightarrow[G]{} \beta} \quad (A ::= \eta)$$ ~~η = eta~~

19/10/2022

Esercizio: Definire Grammatica e Sistema di Transizioni

$\Lambda = \{a, b, c, 0, 1, 2\}$

$L = \{ \underbrace{0^m \cdot 1^n 2^m}_{S_1} \underbrace{a^{k-2} bc^k}_{S_2} \mid m \geq 1 \; n \geq 0 \; k \geq 2 \}$

$S ::= \cancel{02bcc} \; S_1 \cdot S_2$

$S_1 ::= \cancel{02} \mid 0 S_1 2 \mid 0 S_3 2$

$S_3 ::= 1 S_3 \mid \varepsilon$

$S_2 ::= a S_2 c \mid bcc$

$$R_1 = \frac{\alpha = \delta S \gamma \quad \delta, \gamma \in \{\varepsilon\} \quad \beta = \delta S_1 S_2 \gamma}{\alpha \to \beta}$$

$$R_2 = \frac{\alpha = \delta S_1 \gamma \quad \delta, \gamma \in \{\varepsilon, 0, 2\} \cancel{\cup \{S_1\}} \quad \beta = \delta 0 S_1 2 \gamma}{\alpha \to \beta}$$

Regole condizionali

$$\frac{\pi_1 \quad \pi_m}{\gamma \to \gamma'}$$

Se tutti i $\pi_1 \dots \pi_m$ sono veri, la macchina esegue la transizione $\gamma \to \gamma'$

$$\frac{\alpha = \delta A \gamma \quad \delta, \gamma \in (\Delta \cup V)^* \quad \beta = \delta \eta \gamma}{\alpha \xrightarrow[G]{} \beta} \qquad (A ::= \eta)$$

Esempio

$$L = \{ a^n b^n \mid n \geq 1 \}$$

$$S ::= ab \mid aSb$$

$$\frac{\alpha = \delta S \gamma \quad \delta, \gamma \in (\{a, b\} \cup \{S\})^* \quad \beta = \delta ab \gamma}{\alpha \to \beta}$$

$$\overset{\delta}{\boxed{\varepsilon}} S \overset{\gamma}{\boxed{\varepsilon}} \to \overset{\delta}{\boxed{a}} S \overset{\gamma}{\boxed{b}} \to \underbrace{\overset{\delta}{\boxed{aa}} S \overset{\gamma}{\boxed{bb}}}_{\alpha} \to \underbrace{\overset{\delta}{\boxed{aa}} \overset{\eta}{\boxed{ab}} \overset{\gamma}{\boxed{bb}}}_{\beta}$$

α e β sono variabili, mentre a, b e S sono costanti.

Basi Teoriche

π $\begin{cases} \text{Uguaglianze} \\ \text{Relazioni} \\ \gamma \to \gamma' \text{ ricorsiva} \end{cases}$

$Vars(\gamma)$: indico tutte le variabili presenti in γ

Definizione COPERTURA DELLE VARIABILI

Sia R una regola condizionale

$$\frac{\pi_1 \dots \pi_m}{\gamma \to \gamma'}$$

~~[illegible]~~ e sia $x \in Vars(R)$. Diciamo allora che x è coperta in R sse vale una delle seguenti condizioni:

1. x occorre in γ; (troveremo x nella prima configurazione transizione
2. $\exists$ in R una premessa π_i del tipo $y := t$ tale che y è aperta in R e $x \in Vars(t)$;
3. $\exists$ in R una premessa π_i del tipo $x := t$ tale che tutte le variabili in $Vars(t)$ sono aperte
4. $\exists$ in R una premessa π_i del tipo $\partial \xrightarrow{i} \partial'$ tale che tutte le variabili in ∂ sono aperte in R e $x \in Vars(\partial')$