

ESERCIZIO

Sia \mathcal{G}_1 la seguente grammatica:

$$\begin{aligned} S &\rightarrow AaB \mid b \\ A &\rightarrow BcBaA \mid \epsilon \\ B &\rightarrow \epsilon \end{aligned}$$

Chiamiamo \mathcal{A} l'automa caratteristico per il parsing SLR(1) di \mathcal{G}_1 e J lo stato iniziale di \mathcal{A} . Elencare gli item che appartengono a $J[Aa]$.

17 / 24

ESERCIZIO

Sia \mathcal{G}_1 la seguente grammatica:

$$\begin{aligned} S &\rightarrow AaB \mid b \\ A &\rightarrow BcBaA \mid \epsilon \\ B &\rightarrow \epsilon \end{aligned}$$

Chiamiamo \mathcal{A} l'automa caratteristico per il parsing SLR(1) di \mathcal{G}_1 , H lo stato iniziale di \mathcal{A} , T la tabella di parsing SLR(1) per \mathcal{G}_1 . Se non ci sono conflitti nello stato $H[BcBaBc]$ di T , rispondere "NO CONFLICT". Altrimenti, per ciascuna X tale che $T[H[BcBaBc], X]$ contiene un conflitto, dire, specificando a quale X fa riferimento: (i) di che tipo di conflitto si tratta; (ii) quale/i riduzione/i sono coinvolte.

18 / 24

ESERCIZIO

Sia \mathcal{S} il seguente SDD:

$$\begin{array}{lll} S & \rightarrow & E \quad \{S.v = E.v; \} \\ E & \rightarrow & n \quad \{E.v = n.lexval; \} \\ E & \rightarrow & E_1 \ a \ E_2 \quad \{E.v = E_1.v * E_2.v; \} \\ E & \rightarrow & E_1 \ b \ E_2 \quad \{E.v = E_1.v + E_2.v; \} \end{array}$$

e sia P lo stato iniziale del parser SLR(1) per la grammatica di \mathcal{S} . Il parser ha 4 conflitti shift/reduce. Supponiamo che tutti e 4 i conflitti siano risolti a favore di reduce. Supponiamo inoltre che l'attributo $n.lexval$ del terminale n sia il numero intero rappresentato da n . Se l'input $4b3a3$ non è riconosciuto, rispondere "ERROR". Altrimenti dire quale valore viene valutato per $S.v$ su input $4b3a3$.

ESERCIZIO

Scrivere un SDD S-attribuito, basato su grammatica SLR(1), per la generazione dello *abstract syntax tree* per le espressioni aritmetiche generate dalla grammatica ambigua $E \rightarrow E + E \mid E * E \mid (E) \mid id$. Nel caso si usino funzioni ausiliare spiegare qual è il loro ruolo.

ESERCIZIO

Si assumano le seguenti condizioni: il tipo di a è `array(2, array(3, integer))`; la *base* di a è 0; c, i, j sono interi; la dimensione di un intero è 4. Dire quale codice viene generato, usando la *syntax-directed translation* vista in classe e risolvendo l'ambiguità della grammatica secondo le usuali convenzioni, nell'analisi bottom-up della stringa

$$b = a[i][j] * c$$

21 / 24

ESERCIZIO

Si assumano le seguenti condizioni: il tipo di a è `array(2, array(3, integer))`; la *base* di a è 0; c, i, j sono interi; la dimensione di un intero è 4. Si immagini di risolvere l'ambiguità della grammatica secondo le usuali convenzioni e si consideri l'analisi bottom-up della stringa

$$b = c + a[i][j]$$

Usando la *syntax-directed translation* vista in classe, mostrare la computazione degli attributi e l'eventuale codice generato in corrispondenza della quinta riduzione.

22 / 24

ESERCIZIO

Sia \mathcal{D} la seguente porzione incompleta della *syntax directed translation* vista in classe per la generazione di codice per assegnamenti e array:

$$S \rightarrow id = E \qquad gen(table.get(id) \neq E.addr)$$

$$L \rightarrow L_1[E]$$

Dire quali regole semantiche vanno associate all'ultima produzione per ottenere la traduzione di $L_1[E]$ nell'ipotesi che gli array sono memorizzati in *row-major order*.

23 / 24

ESERCIZIO

Sia \mathcal{S} il seguente SDD:

$$\begin{array}{ll} T \rightarrow BC & \{T.t = C.t; \ C.b = B.t\} \\ B \rightarrow int & \{B.t = integer\} \\ B \rightarrow float & \{B.t = float\} \\ C \rightarrow [num] C_1 & \{C.t = array(num.lexval, C_1.t); \ C_1.b = C.b\} \\ C \rightarrow \epsilon & \{C.t = C.b\} \end{array}$$

Mostrare l'albero di derivazione per $int[2][3][4]$. Fornire un ordine topologico per la valutazione del corrispondente albero annotato e mostrare i passi della valutazione.

24 / 24