CSEN 1003: Compilers

Tutorial 7 - Simple LR Parsing

Today's Plan

1. LR Parsing

2. SLR Parsing

3. Recap

LR Parsing

- LR(k) Parsers are deterministic shift-reduce bottom up parsers.

LR Parsing

- LR(k) Parsers are deterministic shift-reduce bottom up parsers.
    - Left to right input scanning.
    - Reverse of a Right-most derivation.
    - k symbols of lookahed.

LR Parsing

- LR(k) Parsers are deterministic shift-reduce bottom up parsers.
  - Left to right input scanning.
  - Reverse of a Right-most derivation.
  - k symbols of lookahed.
- LR grammars are grammars for which deterministic LR parsers can be constructed.

Conflicts in Shift-Reduce Parsers

Example

Consider the following grammar:

$$
\begin{array}{lcl}
E & \rightarrow & E + T \mid E - T \mid T \\
T & \rightarrow & T * F \mid T \, / \, F \mid F \\
F & \rightarrow & \textbf{num} \mid \textbf{id}
\end{array}
$$

and the string: **id + num * id**.

	Stack	Input	Action
1	\$	**id - num * id \$**	Shift
2	\$**id**	**- num * id \$**	Reduce $F \rightarrow$ **id**
..	..	..	..
..	..	..	..
13	\$$E$-$T$	\$	Reduce $E \rightarrow E$-T

LR(0) Items and LR(0) DFA

- An LR(0) item is a production rule with a dot somewhere on the RHS.
    - $A \rightarrow \alpha.\beta$ indicates the state of the parser attempting to parse the input using the rule $A \rightarrow \alpha\beta$ where α is already parsed, and the parser is expecting to parse β next (example: $E \rightarrow E.+T$).
    - Whenever, $A \rightarrow \alpha\beta.$, it might be suitable to reduce $\alpha\beta$ to A (example: $E \rightarrow E + T.$).

LR(0) Items and LR(0) DFA

- An LR(0) item is a production rule with a dot somewhere on the RHS.
  - $A \rightarrow \alpha.\beta$ indicates the state of the parser attempting to parse the input using the rule $A \rightarrow \alpha\beta$ where α is already parsed, and the parser is expecting to parse β next (example: $E \rightarrow E.+T$).
  - Whenever, $A \rightarrow \alpha\beta.$, it might be suitable to reduce $\alpha\beta$ to A (example: $E \rightarrow E + T.$).
- The LR(0) DFA keeps track of the state of the parser regarding what we saw so far and what we need to do next.

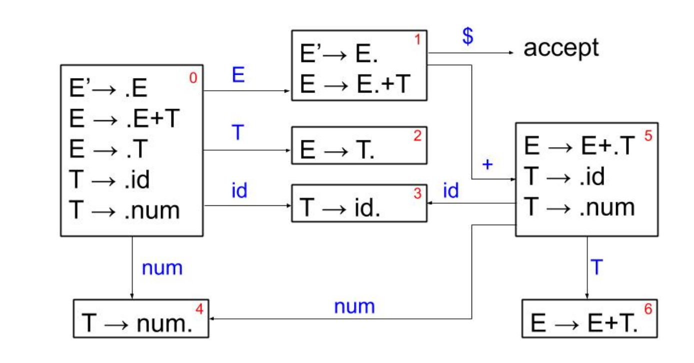
Today's Plan

1. LR Parsing
2. SLR Parsing
3. Recap

Step 1: LR(0) DFA Construction

Example

$$\begin{array}{lcl} E & \to & (a)\ E + T \mid (b)\, T \\ T & \to & (c)\ \textbf{num} \mid (d)\ \textbf{id} \end{array}$$



Step 2: The SLR Parsing Table

- To build a deterministic table, we construct a parsing table.

Step 2: The SLR Parsing Table

- To build a deterministic table, we construct a parsing table.
- The rows are the LR(0) DFA states, the columns are all the terminals and $ and the variables.

Step 2: The SLR Parsing Table

- To build a deterministic table, we construct a parsing table.
- The rows are the LR(0) DFA states, the columns are all the terminals and \$ and the variables.
- The terminals and \$ are called the ACTION table.
- The variables are called the GOTO table.

Step 2: The SLR Parsing Table

- To build a deterministic table, we construct a parsing table.
- The rows are the LR(0) DFA states, the columns are all the terminals and \$ and the variables.
- The terminals and \$ are called the ACTION table.
- The variables are called the GOTO table.
- To fill the table:
  1. $\forall A \in V, \quad GOTO(q, A) = \delta(q, A)$.

Step 2: The SLR Parsing Table

- To build a deterministic table, we construct a parsing table.
- The rows are the LR(0) DFA states, the columns are all the terminals and \$ and the variables.
- The terminals and \$ are called the ACTION table.
- The variables are called the GOTO table.
- To fill the table:
  1. $\forall A \in V, \quad GOTO(q, A) = \delta(q, A)$.
  2. If $A \rightarrow \alpha.a\beta \in q$, $ACTION(q, a) = shift\delta(q, a)$.

Step 2: The SLR Parsing Table

- To build a deterministic table, we construct a parsing table.
- The rows are the LR(0) DFA states, the columns are all the terminals and $ and the variables.
- The terminals and $ are called the ACTION table.
- The variables are called the GOTO table.
- To fill the table:
  1. $\forall A \in V, \quad GOTO(q, A) = \delta(q, A)$.
  2. If $A \to \alpha.a\beta \in q$, $ACTION(q, a) = shift\delta(q, a)$.
  3. If $A \neq S'$ and $A \to \alpha. \in q$, $ACTION(q, a) = reduce\ A \to \alpha$ where $a \in Follow(A)$.

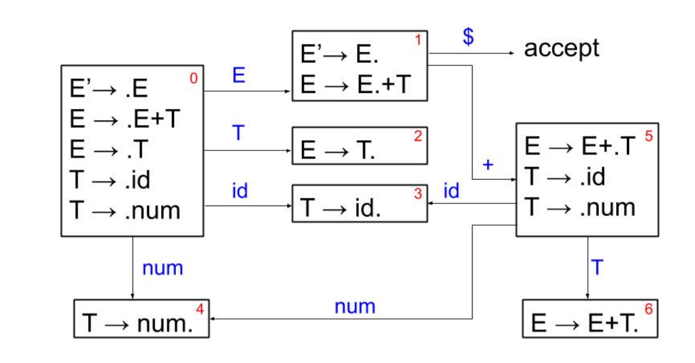
Step 2: The SLR Parsing Table

- To build a deterministic table, we construct a parsing table.
- The rows are the LR(0) DFA states, the columns are all the terminals and \$ and the variables.
- The terminals and \$ are called the ACTION table.
- The variables are called the GOTO table.
- To fill the table:
  1. $\forall A \in V, \quad GOTO(q, A) = \delta(q, A)$.
  2. If $A \to \alpha.a\beta \in q$, $ACTION(q, a) = shift\delta(q, a)$.
  3. If $A \neq S'$ and $A \to \alpha. \in q$, $ACTION(q, a) = reduce\ A \to \alpha$ where $a \in Follow(A)$.
  4. If $S' \to S. \in q$, $ACTION(q, \$) = accept$.
- If conflicts arise while filling the table, then the grammar is not SLR.

LR(0) Automaton

Example

$$\begin{array}{lcl} E & \rightarrow & (a)\ E + T \mid (b)\,T \\ T & \rightarrow & (c)\ \textbf{num} \mid (d)\ \textbf{id} \end{array}$$



SLR Parsing Table Example

Example

$$
\begin{array}{lcl}
E & \rightarrow & (a)\ E + T \mid (b)\,T \\
T & \rightarrow & (c)\ \textbf{num} \mid (d)\ \textbf{id}
\end{array}
$$

Follow(E) $= \{\$,+\}$ Follow(T) $= \{\$,+\}$

State	Action				GOTO	
	+	id	num	$	E	T
0		s3	s4		1	2
1	s5			acc		
2	rb			rb		
3	rc			rc		
4	rd			rd		
5		s3	s4			6
6	ra			ra		

Step 3: The LR Parsing Algorithm

1. Push the start state of the LR(0) automaton to the stack.
2. Loop (S is the top the stack, a is the next input symbol):
   - a. If $ACTION[S, a] = \mathit{shift}\ i$, push i to the stack.
   - b. If $ACTION[S, a] = \mathit{reduce}\ A \to \alpha$, pop $|\alpha| = r$ states off the stack and push $GOTO(q_{n-r}, A)$.

SLR Parsing Example

Example

	Stack	Symbols	Input	Action
1	0		**id + num $**	

SLR Parsing Example

Example

	Stack	Symbols	Input	Action
1	0		**id** + **num** $	Shift 3

SLR Parsing Example

Example

	Stack	Symbols	Input	Action
1	0		**id** + **num** $	Shift 3
2	03	**id**	+ **num** $	

SLR Parsing Example

Example

	Stack	Symbols	Input	Action
1	0		**id + num $**	Shift 3
2	03	**id**	**+ num $**	Reduce $T \to id$

SLR Parsing Example

Example

	Stack	Symbols	Input	Action
1	0		**id** + **num** $	Shift 3
2	03	**id**	+ **num** $	Reduce $T \rightarrow id$
3	02	T	+ **num** $	

SLR Parsing Example

Example

	Stack	Symbols	Input	Action
1	0		**id** + **num** $	Shift 3
2	03	**id**	+ **num** $	Reduce $T \rightarrow id$
3	02	T	+ **num** $	Reduce $E \rightarrow T$
4	01	E	+ **num** $	

SLR Parsing Example

Example

	Stack	Symbols	Input	Action
1	0		**id** + **num** $	Shift 3
2	03	**id**	+ **num** $	Reduce $T \rightarrow id$
3	02	T	+ **num** $	Reduce $E \rightarrow T$
4	01	E	+ **num** $	Shift 5

SLR Parsing Example

Example

	Stack	Symbols	Input	Action
1	0		**id** + **num** $	Shift 3
2	03	**id**	+ **num** $	Reduce $T \rightarrow id$
3	02	T	+ **num** $	Reduce $E \rightarrow T$
4	01	E	+ **num** $	Shift 5
5	015	E+	**num**$	

SLR Parsing Example

Example

	Stack	Symbols	Input	Action
1	0		**id** + **num** $	Shift 3
2	03	**id**	+ **num** $	Reduce $T \rightarrow id$
3	02	T	+ **num** $	Reduce $E \rightarrow T$
4	01	E	+ **num** $	Shift 5
5	015	E+	**num**$	Shift 4
6	0154	E+**num**	$	

SLR Parsing Example

Example

	Stack	Symbols	Input	Action
1	0		**id + num $**	Shift 3
2	03	**id**	**+ num $**	Reduce $T \rightarrow id$
3	02	T	**+ num $**	Reduce $E \rightarrow T$
4	01	E	**+ num $**	Shift 5
5	015	E+	**num$**	Shift 4
6	0154	E+**num**	**$**	Reduce $T \rightarrow num$

SLR Parsing Example

Example

	Stack	Symbols	Input	Action
1	0		**id** + **num** $	Shift 3
2	03	**id**	+ **num** $	Reduce $T \rightarrow id$
3	02	T	+ **num** $	Reduce $E \rightarrow T$
4	01	E	+ **num** $	Shift 5
5	015	E+	**num**$	Shift 4
6	0154	E+**num**	$	Reduce $T \rightarrow num$
7	0156	E+T	$	

SLR Parsing Example

Example

	Stack	Symbols	Input	Action
1	0		**id + num $**	Shift 3
2	03	**id**	**+ num $**	Reduce $T \rightarrow id$
3	02	T	**+ num $**	Reduce $E \rightarrow T$
4	01	E	**+ num $**	Shift 5
5	015	E+	**num$**	Shift 4
6	0154	E+**num**	**$**	Reduce $T \rightarrow num$
7	0156	E+T	**$**	Reduce $E \rightarrow E + T$
8	01	E	**$**	

SLR Parsing Example

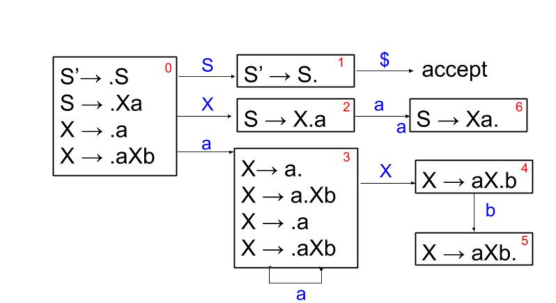
Example

	Stack	Symbols	Input	Action
1	0		**id** + **num** $	Shift 3
2	03	**id**	+ **num** $	Reduce $T \to id$
3	02	T	+ **num** $	Reduce $E \to T$
4	01	E	+ **num** $	Shift 5
5	015	E+	**num**$	Shift 4
6	0154	E+**num**	$	Reduce $T \to num$
7	0156	E+T	$	Reduce $E \to E + T$
8	01	E	$	accept

Conflicts Example - Exercise 7-3

Example

$$
\begin{array}{lcl}
S & \rightarrow & X\mathtt{a} \\
X & \rightarrow & \mathtt{a} \mid \mathtt{a}X\mathtt{b}
\end{array}
$$



State	a	b	$	S	X
3	s3,r$X \rightarrow a$	r$X \rightarrow a$			4

Today’s Plan

1. LR Parsing
2. SLR Parsing
3. Recap

Covered Topics

1. LR Parsing.
2. The LR(0) Automaton and SLR Parsing.

Next Session: LR(1) and LALR Parsing!