← • HOME • UP • →

Top/Bottom-up parsing/SLR(1) parsers

9.5. SLR(1) Parsing (Dragon Book pages 247…)

Recall that the LR(0) finite-state machine is used to read the stack from bottom to top. The state of the machine when the top of the stack is reached tells what the parser should do.

We need to add actions to states. Here are rules for adding actions.

The parser that we create this way is called a *Simple LR(1)*, or SLR(1), parser.

1. The state that contains $S' \rightarrow S\cdot$ accepts provided the lookahead is **$**.
2. If a state q contains LR(0) item $N \rightarrow \alpha\cdot$, where the dot is at the end, then the parser does a reduce by production $N \rightarrow \alpha$ when the machine ends in state q, provided the lookahead is in FOLLOW(N).
3. If a state has a transition out of it labeled by token t, then the parser does a shift if it ends in that state and the lookahead is t.

Example

Let's do a shift-reduce parse of **n*n** using the LR(0) finite-state machine from the preceding page.

Stack	Input	State	Action
$	**n * n $**	0	Shift
$ n	*** n $**	5	Reduce by $F \rightarrow$ **n**
$ F	*** n $**	3	Reduce by $T \rightarrow F$
$ T	*** n $**	2	Shift
$ T *****	**n $**	7	Shift
$ T *** n**	**$**	5	Reduce by $F \rightarrow$ **n**
$ T ***** F	**$**	10	Reduce by $T \rightarrow T * F$
$ T	**$**	2	Reduce by $E \rightarrow T$
$ E	**$**	1	Accept

← • HOME • UP • →

Top/Bottom-up parsing/SLR(1) parsers