CS164 Programming Languages and Compilers **Spring 2024**

Discussion Worksheet 4

Recall that to build the LR(1) parsing DFA:

- Every state is a closed set of LR(1) parsing items, formed with the CLOSURE operation
- If S is the start production, The start state is the closure of $[S' \to \bullet S, \$]$, where S' is some dummy nonterminal if S has multiple production rules.
- If a state "State" contains the item $[X \to a \bullet yb, b]$, we add a transition labeled y to a state that contains the items TRANSITION(State, y)

The algorithms are reproduced here for your reference, but you should learn these.

procedure CLOSURE(Items)
  repeat
    for each $[X \to a \bullet Yb, a]$ in Items **do**
      for each production $Y \to g$ **do**
        for each $b \in$FIRST(ba) **do**
          add $[Y \to \bullet g, b]$ to Items
  until Items is unchanged
  return Items

procedure TRANSITION(State, y)
  Items $\leftarrow \emptyset$
  for $[X \to a \bullet yb, b] \in$State **do**
    add $[X \to ay \bullet b, b]$ to Items
  return CLOSURE(Items)

1 LR(1) Conflicts

We are looking at strings of lowercase and uppercase letters. We want to separate the string into individual sequences of lowercase and uppercase letters. Here is a basic CFG to accomplish this task. `lower` and `upper` are tokens for lowercase and uppercase letters individually.

$$S \to \varepsilon \mid S\ L \mid S\ U$$
$$L \to \varepsilon \mid L\ \texttt{lower}$$
$$U \to \varepsilon \mid U\ \texttt{upper}$$

Exercise 1 Recall that a reduce/reduce conflict occurs when a parsing state contains at least two items:

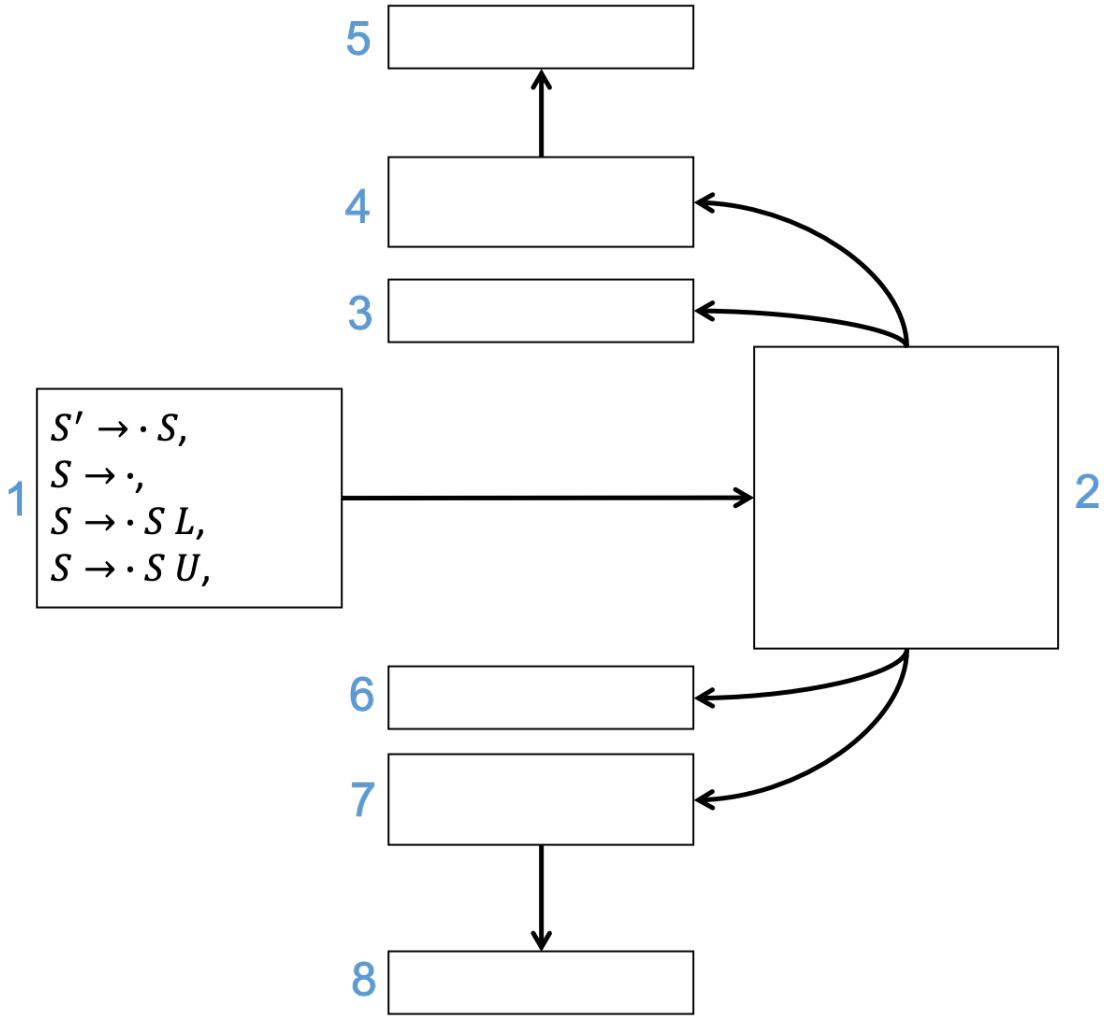
$$[X \to \alpha\bullet, a] \quad [Y \to \beta\bullet, a]$$

1.1 There is a reduce/reduce conflict in this grammar. Identify the smallest string with reduce/reduce conflicts, the rules that can be used to parse it, and the ways the parser can reduce the string.

1.2 Generally, in order to remove reduce/reduce conflicts, we need to remove parsing rules or make them stricter. Change two of the parsing rules in the grammar to remove the reduce/reduce conflict.

Exercise 2 Recall that a shift/reduce conflict occurs when a state contains $[X \to \alpha \bullet a\gamma, b], [Y \to \gamma\bullet, a]$.

2.1 There is still a shift/reduce conflict in the grammar from 1.2. In fact, there are two. Draw the DFA for the grammar and identify all the conflicts. (The right box has 7 items.)



2.2 For each conflict, explain what precedence we should implement and why.

2 LR Parsing

Recall that from the DFA we can create a parse table where:

- a state s with an item $[X \to \alpha\bullet, b]$ reduces with $X \to \alpha$ on b
- a state with transtion $s \to^b s'$ shifts on b

Exercise 3 Create a parse table from the DFA from 2.1, keeping in mind the precedences we chose in 2.2.

	`lower`	`upper`	$	S	L	U
s1						
s2						
s3						
s4						
s5						
s6						
s7						
s8						

Exercise 4 Is our grammar (with precedence modifications from 2.2) LR(0)? Explain.