CS 131 Compilers: Discussion 5: Shift-Reduce Parsing

杨易为 季杨彪 尤存翰

{yangyw,jiyb,youch}@shanghaitech.edu.cn

2021年3月31日

1 Shift-Reduce Parsing the Lambda Calculus.

We’ll look again at the lambda calculus grammar:

```
var  : ID ;
expr : var
     .  ‘(’  ‘l’  var  ‘.’  expr  ‘)’
     |  ‘(’  expr expr  ‘)’  ;
```

1. Is this grammar LL(1)?
2. We’ll now use the following LR(1) parsing table to parse some strings with this grammar.
3. Is this grammar LR(0)?

Answer:

2 Altering the Lambda Calculus.

Suppose we want to add an optional extension that allows raising avarto a power.We define the grammar as

```
expr : var
     | var '-' NUM
     |  ‘(’  ‘lambda’  var  ‘.’  expr  ‘)’
     |  ‘(’  expr expr  ‘)’  ;
var  : ID ;
```

1. Is this grammar LR(0)?
2. Which state in the parsing table would we need to modify to parse this grammar?

   Answer:

3 Stack in Shift-Reduce Parsing.

Suppose it is given that shift-reduce parsing is equivalent to finding the rightmostderivation in reverse. Prove that during shift-reduce parsing, we can only reduce thetopmost items in the stack (i.e. we don’ t need to worry about reducing something inthe middle; hence the usage of a stack is justified)

4 Exercises

Consider the following CFG, which has the set of terminals $T = \{\mathbf{a}, \mathbf{b}\}$

$$S \to X\mathbf{a}$$
$$X \to \mathbf{a} \mid \mathbf{a}X\mathbf{b}$$

1. Construct a DFA for viable prefiexes of this grammar using LR(0) items.
2. Identify a shift-reduce conflict in this grammar under the SLR(1) rules.
3. Assuming that an SLR(1) parser resolves shif-reduce confilcts by choosing to shift, show the operation of such a parser on the input string **aaba**.
4. Suppose that the production $X \leftarrow \epsilon$ is added to this grammar. Identify a reduce-reduce conflict in the resulting grammar under the SLR(1) rules.

Answer: