

198:515 Programming Languages and Compilers I

Problem Set 2

Problem 1

1		goal	::=	expr
2		expr	::=	term + expr
3				term
4		term	::=	factor * term
5				factor
6		factor	::=	num

1. Compute the canonical collection of sets of $LR(1)$ items
2. Construct the $LR(1)$ parse table
3. Is the grammar $LR(1)$ or not? Justify your answer.
4. If the grammar is $LR(1)$, show the behavior of the $LR(1)$ parser on input $\text{num} * \text{num} + \text{num}$, i.e., show stack content, current input, and selected action for each move of the machine.

Problem 2

Show that the following grammar is $LR(1)$ but not $SLR(1)$:

$$S ::= Aa \mid bAc \mid dc \mid bda$$
$$A ::= d$$

Problem 3

Modify the example parser in `~uli/cs515/examples/parser` on the ilab cluster to handle the binary modulo operator “mod”. Note that “mod” has a higher precedence than any other operator except exponentiation (“**”).

To get started, please create your own local version of the parser by copying all the files in the parser directory. . Invoke `make clean`, followed by `make` to generate the parser. For this homework, you will only have to modify the files `scan.l` (scanner) and `parse.y` (parser). To run the parser on a test file, type `parser < mytest`. You should write your own test programs.

Reference: *lex & yacc*, Levine, Mason, and Brown. Two copies are on reserve in the MATH (Hill) library.