**Abinet Kenore**

**Homework 04**

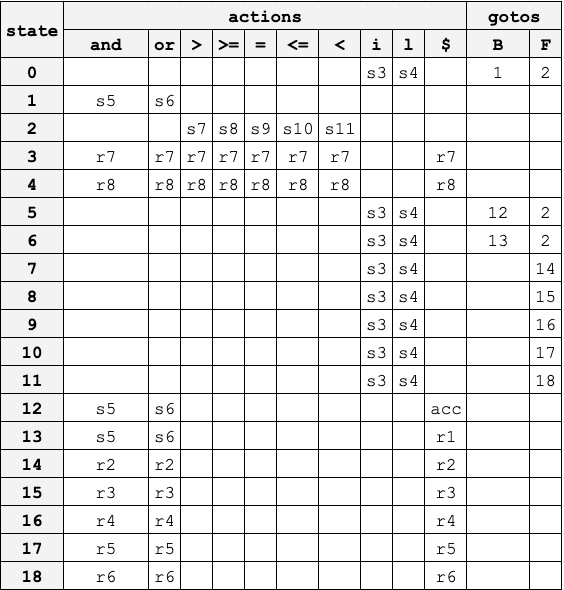
Consider the grammar below specified using EBNF notation.

|  |
| --- |
| <bool\_exp> → <bool\_exp> ( and | or ) <bool\_exp>  <bool\_exp> → <factor> ( > | >= | = | <= | < ) <factor>  <factor> → <identifier> | <literal> |

This grammar can be converted to the format accepted by the [SLR table generator tool](http://jsmachines.sourceforge.net/machines/slr.html) discussed in class, as seen below. Make sure you understand that the two grammars are essentially the same. Substitutions made: <bool\_exp> = B, <factor> = F, <identifier> = i, <literal> = l.

|  |
| --- |
| 0. B -> B and B  1. B -> B or B  2. B -> F > F  3. B -> F >= F  4. B -> F = F  5. B -> F <= F  6. B -> F < F  7. F -> i  8. F -> l |

The next table shows the SLR table built from the grammar using the online tool.



Using the grammar’s productions and the SLR table on the previous page, complete the table below describing the steps taken by an SLR parser, including the contents of the stack and the input, while evaluating a boolean expression. First steps were given to you.

|  |  |  |
| --- | --- | --- |
| **Stack** | **Input** | **Action** |
| 0 | i > l and l = l $ | s3 |
| 0 i 3 | > l and l = l $ | R7 |
| OF3 | > l and l = l $ | S7 |
| 0F2>7 | >l and l = l $ | S4 |
| 0F2>7 l4 | l and l = l $ | R8 |
| 0F2>7 Fl4 | and l = l $ | R2 |
| 0B1 | and l = l $ | S5 |
| 0B1 and 5 | and l =l$ | S4 |
| 0B1 and 5 l4 | l=l$ | R8 |
| 0B1 and 5F2 | =l$ | S9 |
| 0B1 and 5F2 =9 | l$ | S4 |
| 0B1 and 5F2 =9l4 | $ | R8 |
| 0B1 and 5F2 =9F160 | $ | R4 |
| 0B1 and 5 B12 | $ | acc |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |