1. Please list and explain the four actions supported by shift-reduce parsers. (2 pt)

Actions of Shift-Reduce Parsers

- **Shift**: shift the next input token onto the top of the stack
- **Reduce**: the string to be reduced **must be at the top of the stack**
  - Locate the left end of the string within the stack and decide what non-terminal to replace that string
- **Accept**: announce successful completion of parsing
- **Error**: discover a syntax error and call an error recovery routine

2. Please describe the possible conflict types during LR(K) parsing. (2 pt)

What Are Those Conflicts?

- Two possible the conflict types during LR(k) parsing

1. **shift/reduce conflicts** exist in a state
   - when table construction cannot use the next *k* tokens to decide whether to **shift the next input token** or **call for a reduction**
     - The bookmark symbol must occur **before a terminal symbol** *t* in one of the state's items, so that a shift of t could be appropriate
     - The bookmark symbol must also occur **at the end of some other item**, so that a reduction in this state is also possible
2. **reduce/reduce conflicts** exist
   - when table construction cannot use the next *k* tokens to distinguish between **multiple reductions that could be applied in the inadequate state**
   - Of course, a state with such a conflict must have **at least two reducible items**
     - Recall that State 2, 4, and 6 in Fig. 6.11 have single item for reduction