

CS 6353 Compiler Construction, Homework #2

1. Eliminate left recursions and left factors for the following grammar.

$$S \rightarrow A E B$$
$$A \rightarrow Ax \mid Ay \mid Ba \mid a$$
$$E \rightarrow = \mid \neq$$
$$B \rightarrow Ab \mid b$$

2. Consider the following grammar. Note that `id`, `+`, `[`, `]`, and `“,”` are terminals.

$$E \rightarrow E + T \mid T$$
$$T \rightarrow \text{id} \mid \text{id}[] \mid \text{id}[X]$$
$$X \rightarrow E , E \mid E$$

- Eliminate left recursion in the grammar.
- Perform left factoring for the grammar.
- Compute the First set for all symbols in the grammar.
- Compute the Follow set for all non-terminals in the grammar.
- Build an LL(1) parse table for the grammar.
- Parse the string `id + id[id+id, id[]]`. Show the stack, the input, and the action taken.
- Build the parse tree while you are parsing. Show your parse tree.

3. Consider the following grammar.

$$S \rightarrow As$$
$$A \rightarrow BCA$$
$$A \rightarrow BCa$$
$$B \rightarrow b$$
$$C \rightarrow c$$

- Show that the grammar is not LL(1).
- Is the grammar LL(k)? If so, give the k value and show the parsing table.