CMSC 330
Practice Problems 3

For this exercise, we'll call a backtracking recursive-descent parser a *backtracking* parser, and a non-backtracking recursive-descent parser a *predictive* parser.

---

1. Consider the grammar
   S → ABd | aBc
   A → ϵ
   B → b | c

   (a) Compute First sets for each production

   (b) Can the grammar be parsed by a predictive parser?

   (c) Implement a predictive parser for the grammar

   (d) Use your parser to parse the strings "bd" and "acc". Show the sequence of procedure calls in the parse, and what symbols remain to be parsed at each point.

---

2. Consider the grammar
   S → AS | b
   A → SA | a

   (a) Compute First sets for each production

   (b) Can the grammar be parsed by a predictive parser?

   (c) Can the grammar be parsed by a backtracking parser?

   (d) Is the grammar ambiguous? Prove your answer.

   (e) Are all ambiguous grammars non-parseable by predictive parsers?

   (f) Are all non-ambiguous grammars parseable by predictive parsers?

---

3. Consider the grammar
   S → (L) | a
   L → L , S | S

   (a) Compute First sets for each production

   (b) Can the grammar be parsed by a predictive parser?

   (c) Can the grammar be parsed by a backtracking parser?

   (d) Rewrite grammar using the rule for eliminating left recursion

   (e) Compute First sets for each production

   (f) Can the resulting grammar be parsed by a predictive parser?

   (g) Write a predictive parser for the grammar

   (h) Use your parser to parse the string "(a,a)". Show the sequence of calls in the parse, and what symbols remain to be parsed at each point.

---

4. Consider the grammar
   E → E + T | T
   T → a | (E)

   (a) Compute First sets for each production

   (b) Is the grammar ambiguous?

   (c) Can the grammar be parsed by a predictive parser?

   (d) Can the grammar be parsed by a backtracking parser?

   (e) Rewrite grammar using the rule for eliminating left recursion

   (f) Compute First sets for each production

   (g) Can the resulting grammar be parsed by a predictive parser?

   (h) Write a predictive parser for the grammar

   (i) Use your parser to parse the string "a + a + a".

5. Consider the grammar
   $E \rightarrow T + E \mid T$
   $T \rightarrow a \mid (E)$

   (a) Can the grammar be parsed by a predictive parser?

   (b) Would the grammar accept the same language as the grammar in the previous problem?

   (c) What is the difference between this grammar and the previous grammar rewritten to eliminate left recursion?

6. Rewrite the following grammars so they can be parsed by a predicative parser by eliminating left recursion and applying left factoring where necessary.

   (a) $S \rightarrow S + a \mid b$

   (b) $S \rightarrow S + a \mid S + b \mid c$

   (c) $S \rightarrow S + a \mid S + b \mid \epsilon$

   (d) $S \rightarrow a\ b\ c \mid a\ c$

   (e) $S \rightarrow a\ b\ c \mid a\ b\ b$

   (f) $S \rightarrow a\ b\ c \mid a\ b$

   (g) $S \rightarrow a\ a \mid a\ b \mid a\ c$

   (h) $S \rightarrow a\ a \mid a\ b \mid a$

   (i) $S \rightarrow a\ a \mid a\ b \mid \epsilon$

   (j) $S \rightarrow a\ S\ c \mid a\ S\ b \mid b$

   (k) $S \rightarrow a\ S\ c \mid a\ S\ b \mid a$

   (l) $S \rightarrow a\ S\ c \mid a\ S \mid a$