MA/CSSE 474   Day 22 Summary

1. We can eliminate symmetric recursive rules by adding new intermediate nonterminals.

$S^* \to \varepsilon$
$S^* \to S$
$S \to SS_1$
$S \to S_1$
$S_1 \to (S)$
$S_1 \to ()$

2. Another example: Arithmetic expressions:

$E \to E + T$
$E \to T$
$T \to T * F$
$T \to F$
$F \to (E)$

3. A **normal *form*** *F* for a set *C* of data objects is a form, i.e., a set of syntactically valid objects, with the following two properties:
   a) For every element *c* of *C*, except possibly a finite set of special cases, there exists some element *f* of *F* such that *f* is equivalent to *c* with respect to some set of tasks.
   b) *F* is simpler than the original form in which the elements of *C* are written.
      i) By "simpler" we mean that at least some tasks are easier to perform on elements of *F* than they would be on elements of *C*.

4. **Chomsky Normal Form**, in which all rules are of one of the following two forms:
   a) $X \to a$, where $a \in \Sigma$, or
   b) $X \to BC$, where *B* and *C* are elements of $V - \Sigma$.

5. Upper and lower bounds on number of steps in a derivation of a string whose length is n?

6. Converting a grammar to CNF is straightforward but tedious; read about it in the book or slides and practice it for homework.

7. **Greibach Normal Form**, in which all rules are of the form $X \to a\,\beta$, where $a \in \Sigma$ and $\beta \in N^*$.

8. Upper and lower bounds on number of steps in a derivation of a string whose length is n?

9. You are not required to look at the algorithm for converting to GNF (If you are interested see Appendix D)

10. **PDA definition:** $M = (K, \Sigma, \Gamma, \Delta, s, A)$, where
    a) ***K*** is a finite set of **states**
    b) Σ is the finite **input alphabet**
    c) Γ is the finite **stack alphabet** [note that Σ and Γ can contain some of the same symbols]
    d) ***s*** $\in K$ is the **initial (start) state**
    e) ***A*** $\subseteq K$ is the set of **accepting states**, and
    f) Δ is the **transition relation.** It is a finite subset of $(K \times (\Sigma \cup \{\varepsilon\}) \times \Gamma^*) \times (K \times \Gamma^*)$
       i) i.e. (state, single input symbol or ε, string of stack symbols) → (state, string of stack symbols)
       ii) The first "string of stack symbols" will almost always be a single symbol or ε.
       iii) Note that this is nondeterministic; there can be one, many, or zero transitions out of a given configuration.

11. Configurations:
    a) A **configuration** of M is an element of $K \times \Sigma^* \times \Gamma^*$.
        i) (current state, remaining unread input, what's on the stack (left end is top of stack)
    b) The **initial configuration** of M is (s, w, ε), where w is the input string.
12. The stack.
    a) Left end of the string is top of stack
    b) If the stack contains *def* and we push *abc*, the new stack content is *abcdef*.
13. Machine transitions: $(q_1, cw, \gamma_1\gamma) \mid\!\!-_M (q_2, w, \gamma_2\gamma)$ iff $((q_1, c, \gamma_1), (q_2, \gamma_2)) \in \Delta$.
14. Yields, Computations, Acceptance, L(M), Rejection
    a) Let $\mid\!\!-_M{}^*$ be the reflexive, transitive closure of $\mid\!\!-_M$.
    b) *Configuration* C_1 ***yields*** configuration C_2 iff $C_1 \mid\!\!-_M{}^* C_2$
    c) A ***computation*** by M is a finite sequence of configurations $C_0, C_1, \ldots, C_n$ for some $n \geq 0$ such that:
        i) C_0 is an initial configuration,
        ii) C_n is of the form (q, ε, γ), for some state $q \in K_M$ and some string γ in Γ^*, and
        iii) $C_0 \mid\!\!-_M C_1 \mid\!\!-_M C_2 \mid\!\!-_M \ldots \mid\!\!-_M C_n$.
    d) In an ***accepting computation*** *of M*, $C = (s, w, \varepsilon) \mid\!\!-_M{}^* (q, \varepsilon, \varepsilon)$, *and* $q \in A$.
        i) M ***accepts*** a string w iff it has at least one accepting computation that begins with (s, w, ε).
    e) **Messy:** Note that there are many possibilities for non-acceptance:
        i) Read all the input and halt in a non-accepting state,
        ii) Read all the input and halt in an accepting state with non-empty stack,
        iii) Loop forever doing epsilon-transitions and never finish reading the input, or
        iv) Reach a dead end where there are no legal transitions.
    f) L(M), the ***language accepted by*** *M*, is $\{ w \in \Sigma : \text{M accepts w} \}$
    g) A computation C of M is a ***rejecting computation*** iff:
        i) $C = (s, w, \varepsilon) \mid\!\!-_M{}^* (q, w', \alpha)$,
        ii) C is not an accepting computation, and
        iii) M has no moves that it can make from (q, ε, α).
    h) M ***rejects*** a string w iff all of its computations reject.
        i) Note that it is possible that, on input w, M neither accepts nor rejects.
15. We look at PDA's for BAL, A^nB^n, wcw^R. Make sure that you understand how these work.

16. A PDA for $\{a^n b^{2n} : n \geq 0\}$

17. A PDA for PalEven $= \{ww^R : w \in \{a, b\}^*\}$