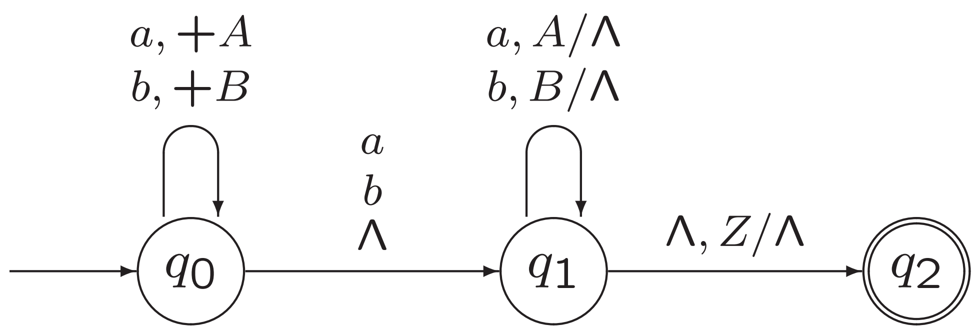
Exercise 5.32.

Lem M be the PDA below, accepting

$$\textit{Pal} = \{y \in \{a,b\}^* \mid y = y^r\} = \{xx^r, xax^r, xbx^r \mid x \in \{a,b\}^*\}$$

(by empty stack). Let $x = ababa$. Find a sequence of moves of M by which x is accepted, and give the corresponding leftmost derivation in the CFG obtained from M as in Theorem 5.29.



Exercise 5.38.

In each case, the grammar with the given productions satisfies the LL(1) property. For each one, draw the deterministic PDA obtained as in Example 5.32.

a. $S \to S_1\$ \qquad S_1 \to AS_1 \mid \Lambda \qquad A \to aA \mid b$

Exercise 5.39.

If G is the CFG with productions

$$S \to T\$ \quad T \to T[T] \mid \Lambda$$

then you can see by considering an input string like [ ] [ ] [ ], which has the leftmost derivation

$$S \Rightarrow T\$ \Rightarrow T[T]\$ \Rightarrow T[T][T]\$ \Rightarrow T[T][T][T]\$ \Rightarrow^* [\,]\,[\,]\,[\,]\$$$

that the combination of next input symbol and top stack symbol does not determine the next move.

Exercise 5.39. (continued)

$$S \to T\$ \quad T \to T[T] \mid \Lambda$$

The problem, referred to as *left recursion* is the production $T \to T[T]$ for variable T, whose right side starts with the same variable T.

In general, if a CFG has the productions $T \to T\alpha \mid \beta$, where β does not begin with T, the left recursion can be eliminated by noticing that the strings derivable from T use these productions are strings of the form $\beta\alpha^n$, where $n \geq 0$.

The productions $T \to \beta U$ and $U \to \alpha U \mid \Lambda$ then generate the same strings with no left recursion. Use this technique to find an LL(1) grammar corresponding to the grammar G.

Exercise.

Generalize the elimination of left recursion to multiple productions, as in $T \to T\alpha_1 \mid \ldots \mid T\alpha_m \;\mid\; \beta_1 \mid \ldots \mid \beta_n$, with $m, n \geq 1$.

Exercise 5.40.

Another situation that obviously prevents a CFG from beging LL(1) is several productions involving the same variable whose right sides begin with the same symbol(s). The problem can often be eliminated by *factoring*: For example, the productions $T \to a\alpha \mid a\beta$ can be replaced by $T \to aU$ and $U \to \alpha \mid \beta$.

Use this technique (possibly more than once) on the CFG G having productions

$$S \to T\$ \qquad T \to [T] \;\mid\; [\,]T \;\mid\; [T]T \;\mid\; \Lambda$$

Is the resulting grammar LL(1)?

Exercise 5.41.

In each case, the grammar with the given productions does not satisfy the LL(1) property. Find an equivalent LL(1) grammar by eliminating left recursion and factoring (when applicable)

a. $S \to S_1\$ \qquad S_1 \to aaS_1b \ | \ ab \ | \ bb$

b. $S \to S_1\$ \qquad S_1 \to S_1A \ | \ \Lambda \qquad A \to Aa \ | \ b$

c. $S \to S_1\$ \qquad S_1 \to S_1T \ | \ ab \qquad T \to aTbb \ | \ ab$

Exercise 5.42.

Let G be the CFG with the following productions:

$$S \to S_1\$ \qquad S_1 \to S_1T \mid ab \qquad T \to aTbb \mid a$$

(which is almost the grammar from Exercise 5.41(c)).

Show that the grammar obtained from G by eliminating left recursion and factoring is not LL(1). Find a string that does not work, and identify the point at which looking ahead one symbol in the input is not enough to decide what move the PDA should make.