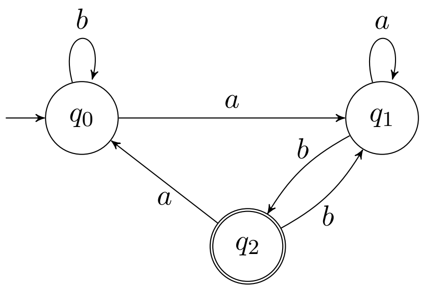
Lecture L&M 4

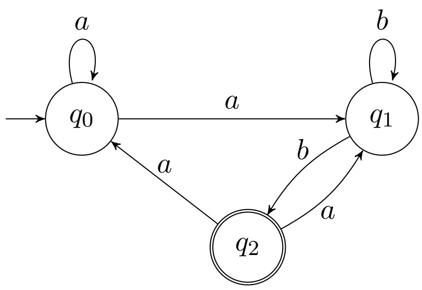
Composition of Automata, Context-Free Grammars

4.1 Projection and cartesian product

Let M_1 be the automaton defined by the following state diagram:



Let M_2 be the automaton defined by the following state diagram:



A. Draw the state diagrams of $M_1 \upharpoonright \{a\}$ and $M_2 \upharpoonright \{a\}$.

B. Draw the state diagram of $M_1 \times M_2$.

4.2 Context-free grammars

Do the following exercises from the book:

C. Let G be the grammar

$$\begin{array}{lcl} S & \rightarrow & abSc \mid A \\ A & \rightarrow & cAd \mid cd. \end{array}$$

(a) Give the derivation of *ababccddcc*.

(b) Build the derivation tree for the derivation in part (a).

(c) Use set notation to define $L(G)$.

D. Let G be the grammar

$$\begin{array}{lcl} S & \rightarrow & SAB \mid \lambda \\ A & \rightarrow & aA \mid a \\ B & \rightarrow & bB \mid \lambda. \end{array}$$

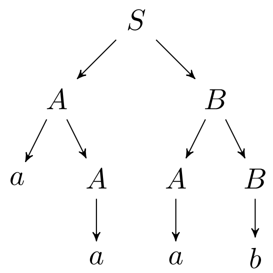
(a) Give a leftmost derivation of *abbaab*.

(b) Give two leftmost derivations of *aa*.

(c) Build the derivation tree for the derivations in part (b).

(d) Give a regular expression for $L(G)$.

E. Let DT be the derivation tree



(a) Give a leftmost derivation that generates the tree DT.

(b) Give a rightmost derivation that generates the tree DT.

F. Construct a grammar over $\{a, b\}$ whose language is $\{a^m b^n \mid 0 \leq n \leq m \leq 3n\}$.

G. For the following grammar, give a regular expression or set-theoretic definition for the language of the grammar. Show that the grammar is ambiguous and construct an equivalent unambiguous grammar.

$$\begin{array}{lcl} S & \rightarrow & aSA \mid \lambda \\ A & \rightarrow & bA \mid \lambda \end{array}$$

4.3 From regular grammars to NFA

For each of the following grammars G, construct an NFA M such that $\mathcal{L}(M) = \mathcal{L}(G)$.

H. G is the regular grammar:

$$\begin{array}{lcl} S & \rightarrow & aA \\ A & \rightarrow & aA \mid bA \mid b. \end{array}$$

I. G is the regular grammar

$$\begin{array}{lcl} S & \rightarrow & aS \mid bA \mid \lambda \\ A & \rightarrow & aA \mid bS. \end{array}$$

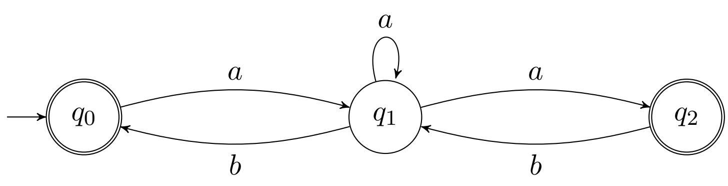
J. *(extra)* G is the regular grammar

$$\begin{array}{lcl} S & \rightarrow & aA \\ A & \rightarrow & aA \mid bB \\ B & \rightarrow & bB \mid \lambda. \end{array}$$

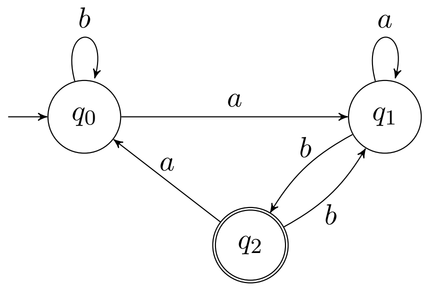
4.4 From NFA to regular grammars

Now, for each of the following NFA M, construct a regular grammar G such that $\mathcal{L}(G) = \mathcal{L}(M)$.

K. M is the NFA:



L. M is the NFA:



M. *(extra)* M is the NFA:

(*Note:* First remove the λ-transitions!)