Exercise 13.1.18

Construct phrase-structure grammars to generate each of these sets.
We create a grammar $G = (V, T, S, P)$.
Vocabulary: $V = \{0,1,S,A\}$
Terminals: $T = \{0,1\}$
Starting symbol: S
Productions: P will be different for each task.

a) $\{01^{2n} \mid n \geq 0\}$

$$P = \begin{cases} S \to 0A \\ A \to 11A \\ A \to \lambda \end{cases}$$

b) $\{0^n 1^{2n} \mid n \geq 0\}$

$$P = \begin{cases} S \to \lambda \\ S \to 0S11 \end{cases}$$

c) $\{0^n 1^m 0^n \mid m \geq 0 \; and \; n \geq 0\}$

$$P = \begin{cases} S \to A \\ S \to 0S0 \\ A \to 1A \\ A \to \lambda \end{cases}$$

Exercise 13.4.6

Express each of these sets using a regular expression.

a) The set containing all strings with zero, one, or two bits
$(\lambda \cup 0 \cup 1)(\lambda \cup 0 \cup 1)(\lambda \cup 0 \cup 1)$

b) The set of strings of two 0s, followed by zero or more 1s, and ending with a 0
001^*0

c) The set of strings with every 1 followed by two 0s
$(0 \cup 100)^*$

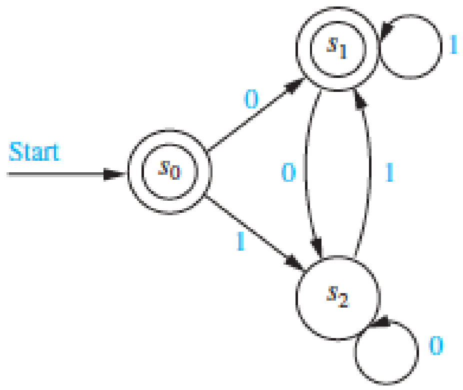
d) The set of strings ending in 00 and not containing 11
$(1 \cup \lambda)(0 \cup 01)^*00$

e) The set of strings containing an even number of 1s
$(0 \cup 11)^*$

Exercise 13.4.16

Construct a regular grammar $G = (V, T, S, P)$ that generates the language recognized by the given finite-state machine.

Peergrade #final



We have the grammar $G = (V, T, S, P)$ where:
Vocabulary: $V = \{0,1, S, A, B\}$
Terminals: $T = \{0,1\}$
Starting symbol: S
Productions:

$$P = \begin{cases} S \rightarrow \lambda \\ S \rightarrow 0A \\ S \rightarrow 1B \\ A \rightarrow \lambda \\ A \rightarrow 1A \\ A \rightarrow 0B \\ B \rightarrow 0B \\ B \rightarrow 1A \end{cases}$$

To explain the many productions;
From our starting symbol S we can do one of three things: Either terminate using the empty string or use a 0 and go to the final state (we call it A) or to the non-final state using a 1 (we call it B)
Whenever we are in the final state (A), we can use a 1 and stay in the final state or we can use a 0 and go to the state B. We can also terminate by using the empty string.
Whenever we are in the non-final state B, we can either use a 1 to go the final state A or we can use a 0 and stay in the non-final state B.