Advanced Topics in Theoretical Computer Science **Problem Set 2**
Due October 25, 2012

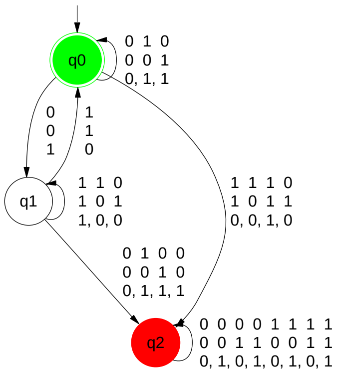
Problem 2.1. (15 points) Let $\Sigma_3 = \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \ldots, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right\}$. The alphabet Σ_3 contains all size 3 columns of 0s and 1s. A string of symbols in Σ_3 gives three rows of 0s and 1s. Consider each row to be a binary number and let

$$G = \{w \in \Sigma_3^* \mid \text{the bottom row of } w \text{ is the sum of the top two rows}\}.$$

For example, $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \in G$, but $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \notin G$. Draw the state diagram of a finite automaton that accepts the language G. Is your automaton deterministic?

Hint: An input word is read bit-triple by bit-triple, and the most significant bit first. You have to take care whether a possible carry-over from the *next* bit has occurred or not.

Solution:



This finite automaton is deterministic. State q_0 is both the initial and the final state of the automaton. State q_0 represents situations in which no carry came from the next bit, and state q_1 represents situations where there was a carry from the next bit (this has to be 'discharged' before accepting). State q_2 represents 'error' situations, that is when the input read so far violates the required property.

Note that a nondeterministic finite automaton is also possible, e.g., we could remove state q_2 and the corresponding transitions.

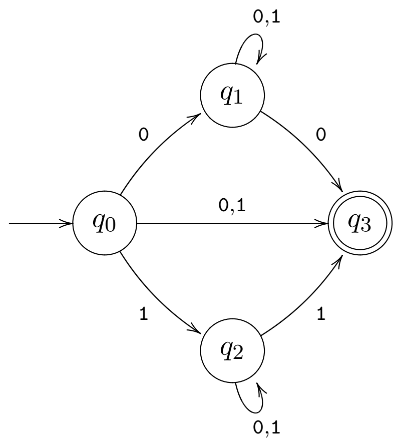
We read the input bit-triple by bit-triple, and the most significant bits first. We have to take care of a possible carry-over. Since we see not only the operand bits of the sum but also the result bit, we can determine whether a carry-over took place or not in the input that we have *not* seen so far. Based on this we can decide if the next triple will contribute to a valid summation string, or if we have to reject. So the automaton has two states for remembering 'carry expected' and 'carry not expected', as well as a state to remember that there was a carry mismatch, i.e., the word has to be rejected no matter what the rest will be.

(15 points: 1 for the answer whether deterministic or not, 6 for the states, 8 for the transitions, including labels)

Problem 2.2. (15 points) Let $\Sigma = \{0, 1\}$. Let $L \subseteq \Sigma^*$ be the language of all words that contain at least one letter, and that begin and end with the same letter. For example, $0, 00, 1011 \in L$ and $\varepsilon, 01, 1100 \notin L$.

a. Draw the transition diagram of a nondeterministic finite automaton N that accepts L and has 4 states. (8 points)

Solution:



b. Turn the automaton N into a deterministic finite automaton M using the subset construction. Label each state of the DFA M with a set of states of the NFA N. (7 points)

Solution: