Problem 1

Let $A \subseteq \{0, 1\}^*$ be the language of a deterministic finite automaton $M$. Consider the language

$$A' = \{ w \in \{0, 1\}^* \mid \exists w' \in A. |w| = |w'| \}.$$

a. Describe the language $A'$ in words.

b. Construct a non-deterministic finite automaton $M'$ that recognizes the language $A'$.

c. Argue briefly that your automaton is correct. (Formal proof not necessary.)

Solution

a. $A'$ is the language of all strings in $\{0, 1\}^*$ whose length is the length of at least one string in $A$.

b. We can take $M'$ to have the same states, start state and accept states as $M$ and the transition function $\delta'(q, c) = \delta(q, 0) \cup \delta(q, 1)$ for all $c$.

c. If some string is accepted by $M'$, we can go through the same sequence of states while now consuming arbitrary characters, thus consuming any sequences of 0s and 1s of the same length and winding up in the same accepting state. Conversely, if a string is accepted by $M'$, then for each transition between successive states $q \rightarrow q'$ in the sequence of states $M'$ will go through, there must have been at least one transition between the same two states in $M$ that gave rise to that transition in $M'$, so there must be some sequence of 0s and 1s we can consume to go through the same sequence of states and wind up in an accepting state of $M$, constituting a string of the same length that is in $A$. 
Problem 2

Use the pumping lemma to prove that the following language over $\Sigma = \{a, b, c, d\}$ is not regular:

$$L = \{a^i b^j c^k d^\ell \mid \ell = i \text{ or } \ell = j \text{ or } \ell = k\}.$$  
(Here, $i, j, k, \ell$ range over non-negative integers.)

Solution

Suppose $L$ was pumpable with pumping length $p$. The string $a^p b^p c^p d^p$ is in the language (with $i = \ell$, $j = k = 0$). Suppose we could decompose it as $xyz$ with $|xy| \leq p$ and $|y| \geq 1$ so that for all $n \in \mathbb{N}$, $xy^n z \in L$. Then we must have $y = a^k$ for some $k \geq 1$ and $x = a^{p-k}$. But then $xy^2 z = a^{p+k} d^k$, which is manifestly not in $L$. So $L$ is not pumpable for any $p$, and hence not regular.

Problem 3

For $k \in \mathbb{N}$, let $n = 2^k + k$ and let $f_k : \{0, 1\}^n \to \{0, 1\}$ be the $n$-bit Boolean function with

$$f_k(x_0, \ldots, x_{2^k-1}, y_0, \ldots, y_{k-1}) = x_{b(y_{k-1}, \ldots, y_0)},$$

where $b(y_{k-1}, \ldots, y_0) = \sum_{i=0}^{k-1} y_i \cdot 2^i$ is the integer with binary representation $y_{k-1} \cdots y_0$. For example, $f(x_0, \ldots, x_{2^k-1}, 0, \ldots, 0) = x_0$ and $f(x_0, \ldots, x_{2^k-1}, 1, \ldots, 1) = x_{2^k-1}$.

Show that for every $k \geq 1$, there exists a Boolean circuit to compute $f_k$ with depth at most $C \cdot k$ and size at most $C' \cdot 2^k$, where $C, C' \geq 1$ are absolute constant (independent from $k$).

Suggested steps:

a. Describe a construction for the circuit to compute the function $f_k$.

b. Prove that the depth of the constructed circuit is at most $C \cdot k$, say for $C = 100$.

c. Prove that the size of the constructed circuit is at most $C' \cdot 2^k$, say for $C' = 100$. 

2
Solution

a. Following the pattern of Homework 1 Problem 1 (and treating the $x_i$ as a “dynamic” truth-table as opposed to the homework problem’s “hardwired” one), we can define circuits $C_k$ computing $f_k$ inductively as follows:

- $C_0(x_0) = x_0$, i.e. the identity circuit containing no gates.
- As the inductive step,

$$C_k(x_0, \ldots, x_{2^k-1}, y_0, \ldots, y_{k-1}) = \text{OR} \ (\text{AND}(y_{k-1}, C_{k-1}(x_{2^k-1}, \ldots, x_{2^k-1}, y_0, \ldots, y_{k-2})), \text{AND}(\text{NOT}(y_{k-1}), C_{k-1}(x_0, \ldots, x_{2^k-1-1}, y_0, \ldots, y_{k-2})))$$

b. We can prove this by structural induction on the definition from part a. Pick $C = 3$. The statement is then manifestly true for the zero case, as it contains 0 gates. In the successor case, the longest path of the new circuit we are constructing either takes us through an OR, AND and NOT gate to an input, or else through an OR and an AND gate through the circuit for $f_{k-1}$, so its depth is $\max\{3, 2 + \text{depth}(C_{k-1})\}$, which is certainly $\leq 3 \cdot k$ if we assume the inductive hypothesis that $\text{depth}(C_{k-1}) \leq 3 \cdot (k - 1)$. □

c. We can likewise prove this by structural induction. Set $C' = 4$. We will prove the stronger statement that $\text{size}(C_k)$ is at most $C' \cdot (2^k - 1)$. In the zero case, the size is again a satisfactory 0. In the step case, we now have a size of exactly $4 + 2 \cdot \text{size}(C_{k-1})$. Assuming the inductive hypothesis that $\text{size}(C_{k-1}) \leq 4 \cdot (2^{k-1} - 1)$, we then get $\text{size}(C_k) \leq 4 + 4 \cdot 2^k - 8 = 4 \cdot (2^k - 1)$ as required. □

Problem 4

Consider the following language over $\{0, 1\}$:

$$B = \{w_0 \cdots w_n \mid b(w_n, \ldots, w_0) \text{ is a multiple of 5}\},$$

where $b(w_n, \ldots, w_0) = \sum_{i=0}^{n} w_i \cdot 2^i$ is the integer with binary representation $w_n \cdots w_0$.

Caution: The order of the bits is reversed. For example, 10011 $\in B$, but 11001 $\notin B$.

a. Construct a deterministic finite automaton for $B$. Describe it in prose or by drawing a diagram. (Hint: Consider the remainder of successive powers of 2 modulo 5.)

b. Argue that your automaton is correct. (Formal proof not necessary.)
Solution 1

a. Note that successive powers of 2 have repeating remainders 1, 2, 4, 3, 1, 2, 4, 3, ... Therefore, there is only a finite number of weights that successive digits could have when contributing to the remainder of the number observed so far modulo 5, and moreover we can determine the weight of the next digit based on the weight of the previous digit. We can therefore construct a DFA as follows:

\[
Q = \{1, \ldots, 4\} \times \{0, \ldots, 4\}
\]

\[
\delta((w, r), c) = (2 \cdot w \mod 5, r + c \cdot w \mod 5);
\]

\[
q_0 = (1, 0);
\]

\[
F = \{(w, 0) \mid w \in \{1, 2, 4, 3\}\}.
\]

b. Reading a string in \(B\) in the canonical order is essentially equivalent to reading a normal binary string in from right to left to determine whether it is divisible by 5. We can therefore argue, e.g. by induction, that the automaton described will be in an accepting state after having read a terminal segment of a binary number if and only if the terminal segment is a binary representation of a number that is a multiple of 5. (That this is the case essentially follows from the definition of positional notation as a sum of digits multiplied by their positional weight and the fact that taking remainders distributes over addition.)

Solution 2

a. By a construction analogous to HW2P5(a) (show an illustration), we can accept \(B^R\) (defined as in HW3P2). Therefore, by reversing all transitions (and vacuously swapping the start and accept state), we obtain an automaton that accepts \((B^R)^R = B\).

b. Correctness of the unreversed automaton follows from the binary string \(wc\) encoding \(2 \ast (\text{encoding of } w) + c\) and taking remainders distributing over addition. Reversing all transitions and swapping start/accept states results in an automaton accepting the reverse language by HW3P2.