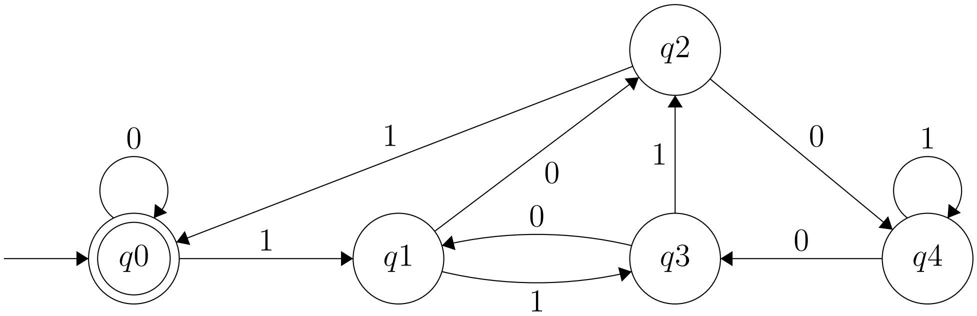
CS 381 - A2

Martin Mueller

Due: Friday, February 21^{st}, 2020

1. (7 points) Show that the set of all strings $x \in \{0,1\}^*$, such that x is either a multiple of 3 or x is a multiple of 5, is regular. Give a digitally-drawn state diagram that is well-organized and the states have intuitive yet concise labels. Give a concise and precise English description of how your DFA works. Finally, give a formal description of your DFA, contained in the file A2-1.dfa, that runs in the Automata-Simulator.



The above DFA determines if a given binary string is divisible by 5. It does so by processing the remainder of the string. Each state q_n represents the remainder n of the given string divided by 5. Any string with a remainder of 0 when divided by 5 is, by definition, divisible by 5.

From this DFA, we can gather that the language $\{x|x$ is a multiple of 5$\}$ is regular. We also know from a DFA derived in class that the language $\{x|x$ is a multiple of 3$\}$ is regular. We can now use these two DFAs to derive the above language. Also discussed and proved in class was that the class of regular languages is closed under the union operation. If we take the union of the language of the DFA derived in class with the language of the DFA above, we can obtain the language $\{x|x$ is either a multiple of 3 or a multiple of 5$\}$. Since this language can be obtained through a union of two regular languages, it stands to reason that this language is also regular.

2. (13 points) Let A and B be languages comprised of binary strings. Define the following "implication" operator $\rightarrow$ on sets as follows: $A \rightarrow B = \{x \mid x \in A \rightarrow x \in B\}$. Show that the class of regular languages is closed under implication. In other words, if A_1 and A_2 are regular languages, then so is $A_1 \rightarrow A_2$. Your construction must be deterministic. Give a full construction, the level of detail of which should be similar to the level of detail we used in class to show that the class of regular languages is closed under union. Also, give a concise and precise English description of your proof "strategy" or "idea" so I can get a good sense of what you're doing.

I will be using definitions of logical operators and DFAs to show it is possible to construct a regular language from $A \rightarrow B$ provided A and B are regular languages. Let us start with first rewriting the implication operator definition:

$$\begin{aligned} A \rightarrow B &= \{x \mid x \in A \rightarrow x \in B\} \\ &= \{x \mid \neg(x \in A) \vee (x \in B)\} \\ &= \{x \mid x \notin A\} \cup \{x \mid x \in B\} \\ &= \neg A \cup B \end{aligned}$$

From this simplification, we can gather two things: One, we know that B is a regular language by the definition of the implication operator, and two, since the union of two regular languages results in a regular language, if $\neg A$ is regular, then the class of regular languages is closed under the implication operator. To prove that $\neg A$ is a regular language, let's construct a DFA that recognizes $\neg A$. We'll start by constructing a DFA M that recognizes A:

$$\begin{aligned} M &= (Q, \Sigma^*, \delta, q_0, F) \\ Q &= \text{the set of all the states of } M \\ \Sigma^* &= \{0, 1\}^* \\ \delta &= \text{the transition function of } M \\ q_0 &= \text{the start state} \\ F &= \text{the set of all accept states of } M \end{aligned}$$

From this, we can construct M', a machine that recognizes the language $\neg A$. If A is the set of all strings M accepts, then $\neg A$ is the set of all strings that M doesn't accept. This is simply all the states of M that aren't in F, or $Q - F$. Using this, we can construct M': $M' = (Q, \Sigma^*, \delta, q_0, Q - F)$. Since we are able to construct a DFA M' that recognizes the language $\neg A$, the language $\neg A$ is regular, and therefore, since B is also regular and the class of regular languages is closed under the union operation, the class of regular languages is also closed under the implication operation.

Submit your PDF (generated from LaTeX) and DFA files in one ZIP file.