CSC341 Lab 3B

Havin Lim, Carter English

February 10th 2024

1 Language Complement

We assume that A is recognized by a DFA $L_1 = (Q, \Sigma, \delta, q_0, F)$.
To prove that $\bar{A}$ is a complement of A, we need to create a DFA that recognizes $\bar{A}$ and not A.
Now we make another DFA $L_2 = (Q, \Sigma, \delta, q_0, F_2)$.
The difference here is $F_2 = Q - F$.
Through the second DFA, the accepting states of L_2 are certainly the opposite of L_1 and they will accept any strings that have been rejected by one another.

2 Thinking inductively

1

The naturals have a similar inductive definition to the regular expressions. Here, the base case is 0 and the recursive case is that every other natural is the successor of some natural. The naturals can be thought of as isomorphic to the language $\{c\}^*$ for any character c, as it contains the empty string and every other string in the set is the concatenation of c and another string in the set. The set of regular expressions as a whole is therefore much more expansive than the naturals are.

2

In proof by induction over $\mathbb{N}$, we need the base case 0, and (assuming we only need one base case, as it is often beneficial to include a couple more) the step case is every natural ≥ 1. The inductive hypothesis is that whatever we are trying to prove works for every natural less than k, and we want to prove that it works for k.

In proof by induction over regular expression, we have three base cases; a singleton language, a language containing the empty string, and the empty language. We also have three inductive cases, where we assume S is the union, concatenation, or star of some language, and the inductive hypothesis is that those languages are regular.

3 Syntactic sugar (is sweet)

No; as proven in Lab 3A, for any regular expression R, R^+ and $R^?$ can be written as the concatenation or union respectively of two regular expressions, and is therefore a regular expression. Thus everything that is a regular expression under the new definition is a regular expression under the old definition, so the theorem still holds.

(2) When we add elements to our data type, any code we wrote that performs case analysis on this data type now needs to be modified in order to account for the additional structure. Similarly, if we were to modify our definition of a regular expression, any key theorems we proved using case analysis now need to be modified to consider these additional cases.

4 Ending this lab with a bang

(a): $0^! = 1$, which is a singleton in Σ and is therefore a regular expression
(b): $1^! = 0$, which is a regular expression by the same reasoning
(c, d): We have that ϵ and $\emptyset$ are regular expressions, so $\epsilon^!$ and $\emptyset^!$ are regular expressions
(e): We have $(R_1 \cup R_2)^! = R_1^! \cup R_2^!$. By the inductive hypothesis, $R_1^!$ and $R_2^!$ are regular expressions, so by closure under union $R_1^! \cup R_2^!$ is a regular expression
(f): By the same reasoning, $(R_1 R_2)! = R_1!R_2!$ is a regular expression by the inductive hypothesis and closure under concatenation
(g): By the inductive hypothesis, $R_1^!$ is closed, so by closure under $*$, $(R_1^!)^*$ is a regular expression