Formal Languages and Computability 6

Ragnar Björn Ingvarsson, rbi3

16. október 2024

1 Let $\Sigma = \{0, 1\}$ and consider $A = \{01^n01^n \mid n \geq 0\}$. First show if A is regular or not. If A is nonregular languages, give a CFG or a PDA to show that it is context free.

To prove that the language is nonregular we use pumping lemma.

Let $s = 01^p01^p$, and we get three cases for the splitting into xyz:

- $y = 0$: Here, for xy^iz we can say that $i = 2$ and we get the string 001^p0^p which is not in A so here lies a contradiction.
- $y = 01^k$, $k < p$: Here we can pump to any $i > 1$ and get $(01^k)^i1^{p-k}01^p$ which is clearly not in A so we get a contradiction.
- $y = 1^k$, $k < p$: Here we can pump until the number of y's exceeds p by saying $i = p + 1$, and we get a contradiction since such a string is not in A.

We see that for the string $s = 01^p01^p$ all three cases lead to a contradiction, bringing us to the conclusion that A is nonregular.

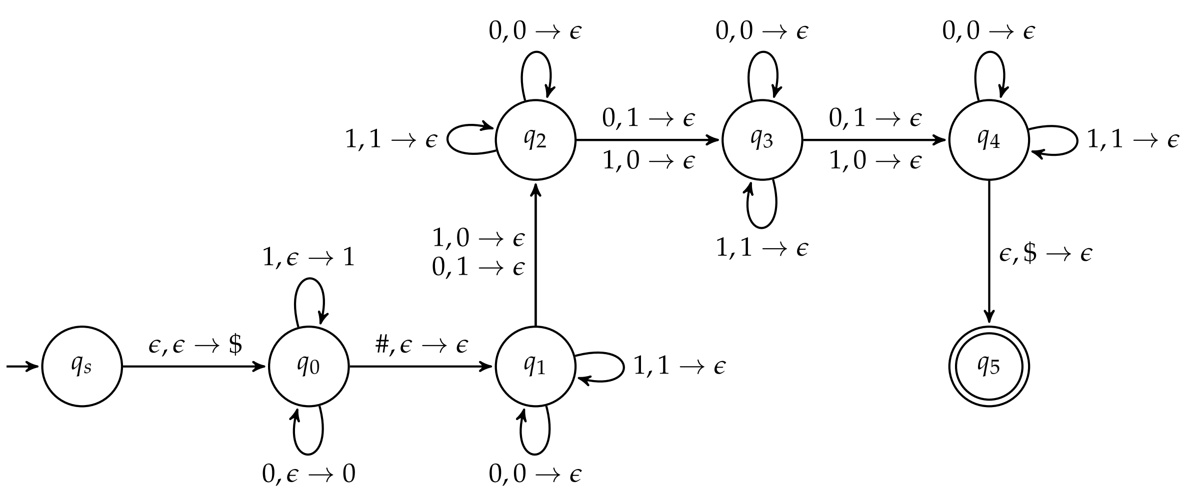
We can then give a CFG to show that it is context free and such a CFG would look like this:

$$S \rightarrow 0A$$

$$A \rightarrow 1A1 \mid 0$$

Where S is the starting state.

2 Let A denote the language that corresponds to checking whether the Hamming distance between two binary strings is three. Show that A is CFL by constructing a PDA that recognizes it.



3 Suppose that a CS student wants to encode events taking place in an ice-hockey game to a string using alphabet $\Gamma = \{a, b, c, 1, 2, 3, x, y, |\}$.

a) Let A denote the language corresponding to all valid strings. Show that A is a regular language.

First we will give a RE for one period.

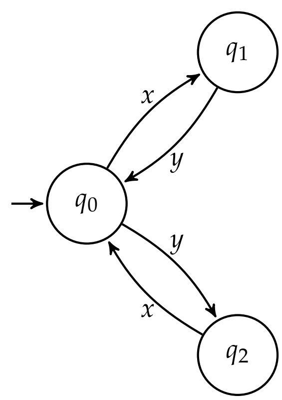
$$((1 \mid 2 \mid 3)(a \mid b \mid c)) \mid ((a \mid b \mid c)(1 \mid 2 \mid 3))(x^* y^* (a \mid b \mid c)^* (1 \mid 2 \mid 3)^*)^*$$

We can then denote this regular expression as p and then just say that the regular expression for the whole game is

$$p \ \text{"}|\text{"} \ p_1 \ \text{"}|\text{"} \ p_2$$

b) Let B denote those interesting games where at no point in time either team leads by more than 1 goal. Is B regular or nonregular?

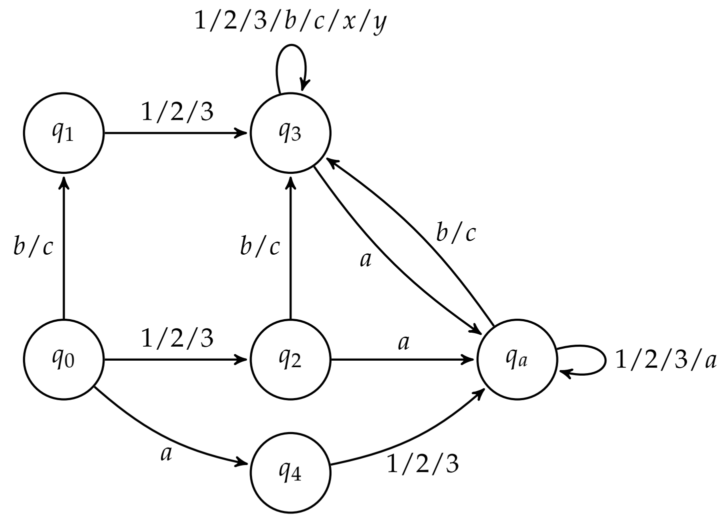
This is regular as we can give a rough outline of a NFA that describes it:



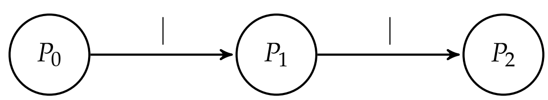
Here, the q_0 state is for the score being 0-0, the q_1 for 1-0 and q_2 for 0-1, where each state is a NFA that describes a normal game and each transition is to and from the same state within those NFA's. With this we can ensure that at no point in time does the lead ever get greater than 1 and we can see that this can be described with a NFA so it is regular.

c) Plus-minus statistic is an often used performance metric: a goal means +1 for the wing, and when the opponent scores the wing on ice receives -1. Let C denote the games where home team's 1st wing has a positive plus-minus statistic. Show that C is context-free.

We show that it is context-free by giving a PDA for it, and to begin with we can give a NFA for a single period with two states for determining if a is on the ice or not. I choose here to make a NFA since it never interacts with the stack as we need the stack to count the plus-minus statistic.

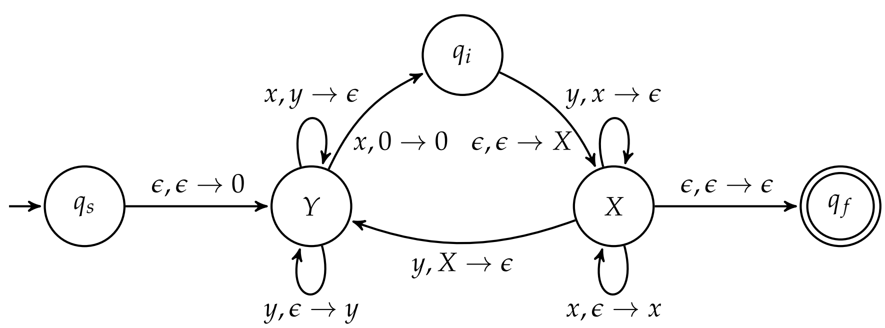


Notice that we dont define any start or end states since this is only going to be part of the whole PDA. We will then name this chunk P for period and we can then link up a whole game with such a NFA:



Where each P state has a transition from both q_3 and q_a to the q_0 state of the next P on the symbol "|".

Now, we can have two instances of this NFA called X and Y. Then we can draw out the final PDA consisting of these two states along with a start state q_s and finish state q_f.



Where aside from the paths leading from and to the start and end state, all paths only travel from q_a to q_a, in the corresponding period. Then there is the path from the start state which leads to the q_0 state in the first period of q_y and the end state where to we can travel from the q_3 or q_a state in the final period of q_x.

The paths relating to the intermittent state q_i are as such; the path to q_i is from q_a and the path from q_i is to q_a of X in the same period as was in Y.

The logic here is that we have two states of the game, negative and positive, Y and X. When we are situated in Y, the negative state, we can be sure that our stack contains no plus scores, that is, x's. And the reverse is also true that when we are in

X, no y's are in the stack. This ensures that we know the exact state of the game at any given time instead of having to count the score at the end.