#1: Finite Automata

Ihab M. E.

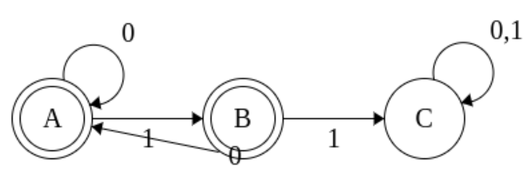
June 28, 2017

1 Formal Definitions:

- **Alphabet $\sum$:** finite set of symbols. Example: Binary Alphabet {0,1}
- A **string** is a set of elements over an alphabet. Example : 0101, from the binary alphabet.
- $\sum^*$is the set of all possible strings over an alphabet $\sum$.
- **Length** of a string: the number of symbols used to represent a symbol, usually denoted by |S|.
- ϵ is the empty string, a string of length zero.
- **Q** is the set of possible states in an automaton.
- δ is the transition function from a state to another state using an input symbol.
- q_0 is the initial state of an automaton.
- F is the set of final/accepting states. $F \subseteq Q$.

2 Transition Function

Take for example the following automaton:



A simple transition function would be: $\delta(A, 1) = B$, which means from A given an input of 1, that will get us to the state B.

Let us first explain what that automaton is supposed to do; it's a very simple automaton whose job is to ensure that the strings of this binary language don't have any consecutive 1's, and as you can see we have two accepting/ final states, one of which is actually the initial state, and we have a dead state.

What is the purpose of that dead state, in your opinion? It stops us from accepting any strings after seeing two consecutive ones, because as we've seen in our simple transition function, given an input of 1 makes us go from the initial state A to B, now we already have 1 in our string, in fact it's our whole string, but if we add another one we have committed a crime -don't worry about it- and we can never produce a string in that case unless we start all over again.

So, we're at B, given another simple transition function with an input symbol 0 gets us to A, $\delta(B, 0) = A$.

And since we're only dealing with one symbol inputs we can implement the following transition table:

	state	1	0
$* \rightarrow$	A	B	A
$*$	B	C	A
	C	C	C

Note: the * means this is an accepting/final state, and the pointer$\rightarrow$ means this is the initial state.

And of course our dead state doesn't go anywhere except itself.

2.1 Extended Transition Function

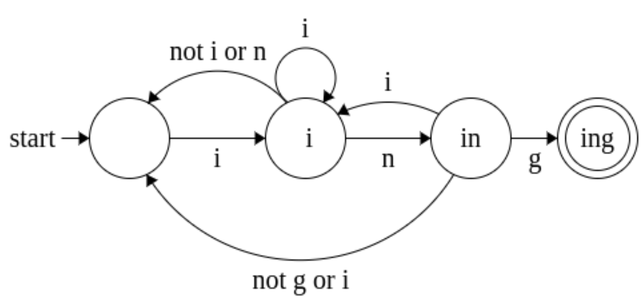
An "extended" transition function is nothing but a regular transition function that takes a string input instead of a one symbol input. For example: $\delta(A, 101) = B$. Given a string input of 101 gets us form A to B. We can actually break this extended transition function into equivalent others;

$\delta(A, 101) = B = \delta(\delta(A, 1), 01)) = \delta(B, 01) = \delta(\delta(B, 0), 1) = \delta(A, 1)$

Notice that as long as our extended function results in an accepting/final state that means our string is accepted by our language.

3 An Example Automaton

Check the following automaton:



This is a simple automaton that tells you whether the word you're processing has "ing" or not.

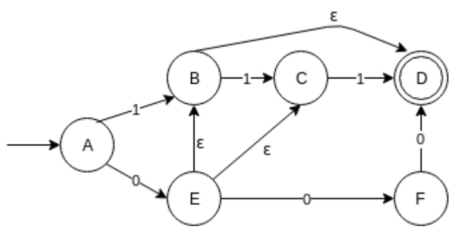


Figure 1: An Example Automaton

4 Epsilon-transition and Closure

Another very important concept of Automata theory is the epsilon transition; in which, obviously, you move from a state to another -or itself- by the empty string as your input for the transition function. Check the following automaton:

The initial state is A, and it only has one accepting/final state, which is D. It's interesting to note that this is a nondeterministic automaton, which is why it has so many epsilons.

Let us get the closure sets CL(S) for each of these states;

$CL(A) = \{A\}$

$CL(B) = \{B, D\}$

$CL(E) = \{E, B, D, C\}$

$CL(F) = \{F\}$

$CL(D) = \{D\}$

5 One Last Note on Transition Functions for NFAs

Since NFAs are unceratin, they have to guess, and they can be at more than one state at the same time. Applying a transition function to that in figure 1 might result in this

$\delta(E, \epsilon) = \{E, B, D, C\}$

Why is D in the set? Since ϵ represents the empty string so it has no length (it has a length of zero, to be exact), so you can't say $\delta(E, \epsilon\epsilon)$ because that would be illegal.

We'll talk more about that when we try converting an NFA to a DFA.