Automata

Course Notes

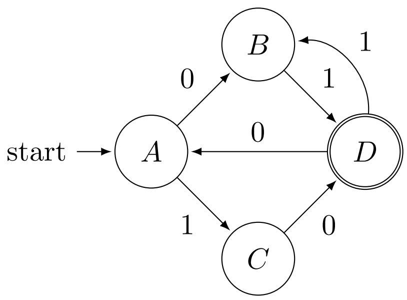
12 September 2015 – 5 November 2015

Contents

Week 1: 12 September 2015 – 18 September 2015

Homework Assignment 1.

Question 2: The finite automaton from in Figure 1 accepts no word of length zero, no word of length one, and only two words of length two (01 and 10). There is a fairly simple recurrence equation for the number $N(k)$ of words of length k that this automaton accepts. Discover this recurrence and demonstrate your understanding by identifying the correct value of $N(k)$ for some particular k.



$\delta(q,x)$	0	1
$\triangleright$ A	B	C
B	-	D
C	D	-
$*$ D	A	B

Figure 0.1: The state diagram and transition table for the DFA in Questions 1 and 2.

Solution. Suppose we are already at node D (that is, we've made an acceptable string). How many strings of length k could have gotten us here? Define Q_n) to be the number of paths of length n that terminate at Q. Initially we know that $A_0) = 1$, $B_0 = C_0 = D_0 = 0$. Now we also know the following recursion relations:

$$A_n = D_{n-1},\ B_n = A_{n-1} + D_{n-1},\ C_n = A_{n-1},\ D_n = B_{n-1} + C_{n-1}$$

After some simplification this becomes the recurrence relation:

$$D_n = D_{n-2} + 2D_{n-3},\ D_0 = D_1 = D_3 = 0,\ D_2 = 2.$$

Now define $D(x) = \sum_{n\geq 0} D_n x^n$ to be the ordinary generating function of D_n. Taking our original relation, we multiply both sides by x^n and sum over $n \geq 3$:

$$\sum_{n\geq 3} D_n x^n = \sum_{n\geq 3} D_{n-2}x^n + 2\sum_{n\geq 3} D_{n-3}x^n \quad \Rightarrow \quad -2x^2 + \sum_{n\geq 0} D_n x^n = x^2 \sum_{j\geq 0} D_j x^j + 2x^3 \sum_{k\geq 0} D_k x^k$$

$$-2x^2 + D(x) = x^2 D(x) + 2x^3 D(x) \quad \Rightarrow \quad D(x) = \frac{2x^2}{1 - x^2 - 2x^3}$$

Therefore D_n, the number of paths from A to D of length n, is just the coefficient of x^n in the Taylor expansion of $D(x)$! By "Wolfram's Theorem," we obtain the following values for D_n. ∎

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
D_n	0	0	2	0	2	4	2	8	10	12	26	32	50	84	114

Figure 0.2: The first 15 values of D_n.

Question 3: Given the above transition table (I also included the state diagram) of a simple DFA with start state A and accepting state B, we want to show that this automaton accepts exactly those strings with an odd number of 1's, or more formally: $\delta(A, w) = B$ if and only if w has an odd number of 1's. (Here, δ is the extended transition function of the automaton.)

$\delta(q, x)$	0	1
$\triangleright$ A	A	B
$*$ B	B	A

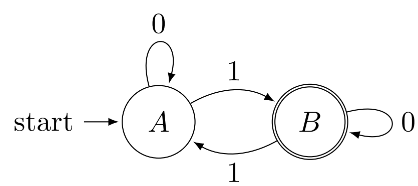


Figure 0.3:
The state diagram and transition table for the DFA in Question 3.

Proof. We proceed by induction on $|w|$, the length of w. For the base case, suppose $|w| = 0$. Then $w = \varepsilon$, since the only word of length zero is the empty word. Now w must have zero 1's as well, and by definition $\delta(A, \varepsilon) = A$. Since zero is an even number, the statement holds for the base case.

Now assume that $\delta(A, x) = B$ if and only if x has an odd number of 1's, for all $|x| \leq |w|$, and consider w. We have two cases: w is of the form $w = x0$ or else it is of the form $w = x1$. Moreover, x in each case may have either an even number or else an odd number of 1's. We need merely to exhaust all cases.

(1) Let $w = x0$ and x has an even number of 1's. Then so does w, since we've appended no additional 1's. Now $\delta(A, w) = \delta(A, x0) = \delta(\delta(A, x), 0) = \delta(A, 0) = A$, by the inductive hypothesis and the transition table. Hence the statement holds for this case.

(2) Let $w = x0$ and x has an odd number of 1's. Then w has an odd number of 1's, and $\delta(A, w) = \delta(A, x0) = \delta(\delta(A, x), 0) = \delta(B, 0) = B$. Again the statement holds.

(3) Let $w = x1$ and x has an even number of 1's. Then w has an odd number of 1's since we just added an additional 1. Now $\delta(A, w) = \delta(A, x1) = \delta(\delta(A, x), 1) = \delta(A, 1) = B$. Again the statement holds.

(4) Let $w = x1$ and x has an odd number of 1's. Then w has an even number of 1's, and

$\delta(A, w) = \delta(A, x1) = \delta(\delta(A, x), 1) = \delta(B, 1) = A$. Since the statement holds for this case as well as the other three, we have completed the proof. ■

Question 4: Consider the non-deterministic finite automaton. Which strings does it accept?

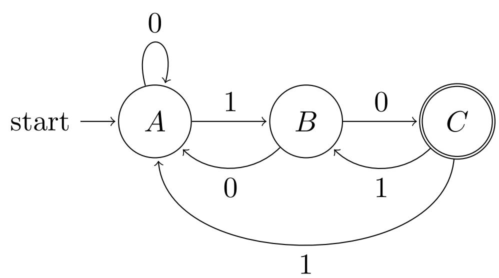
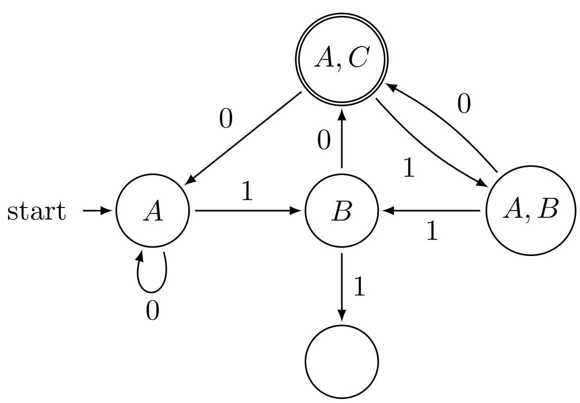


Figure 0.4: The state diagram for the non-deterministic automaton in Question 4.

Solution. We're given (e.g.) the following options: 0010010, 00010111, 01010011, and 1011101. In this case, the solution is trivial: we see that the only edge leading to the accepting state C is labeled 0, and C has no looped edges; hence the NFA cannot accept any strings that end in a 1. It remains to verify that 0010010 is indeed an accepted string: indeed, the path $A \to A \to B \to A \to A \to B \to C$ corresponds with (part of) the output of $\delta_N(A, 0010010)$. ■

Question 5: Convert the NFA to a determinstic automaton. Which state is not reachable from the start state {A}?



$\delta_D(\{q\}, x)$	0	1
▷ {A}	{A}	{B}
{B}	{A,C}	{ }
∗ {A,C}	{A}	{A,B}
{A,B}	{A, C}	{B}

Figure 0.5:
The state diagram and transition table for the DFA constructed in Question 5.

Solution. We convert the NFA to a DFA using the subset construction. Notice that, in particular, the states $\{B, C\}$ and $\{A, B, C\}$ are not included, and therefore are not reachable from $\{A\}$. ■

Challenge Problem 1.

Let L be the language with alphabet $\{0, 1, 2\}$ consisting of strings that do not have any three consecutive 0's, any three consecutive 1's, or any three consecutive 2's. Prove that L is a regular language (hint: design automata or regular expressions for some simpler languages and then use closure properties of regular languages to get L). Harder is to design a DFA A for which the language is L itself, but we encourage you try to design one as a second part of this exercise.

Solution (closure). This solution exploits the fact that regular languages are closed under intersection. We first show that if $\Sigma = \{0, 1, 2\}$, then $L_a = \{x \in \Sigma^* : aaa \notin x\}$ is a regular language for any $a \in \Sigma$. Now, the choice of symbols is arbitrary, so the machine in Table 2 works for any choice of a

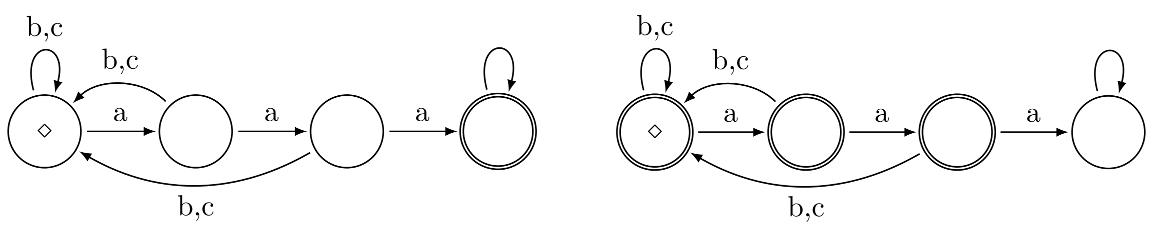


Figure 0.6: The DFA state diagrams for L'_a (left) and L_a (right).

(respectively, b and c) from Σ. Now we have three regular languages, L_0, L_1, L_2, each corresponding to the appropriate choice. Now we can take the intersection to get our new regular langugage, where none of 000, 111, or 222 appears in a word... except that this is exactly our original L! Hence L is a regular language, and the problem is solved. ■

$\delta(q, x)$	0	1	2
$* \triangleright$	x0	x1	x2
$*$ x0	x00	x1	x2
$*$ x1	x0	x11	x2
$*$ x2	x0	x1	x22
$*$ x00	err	x1	x2
$*$ x11	x0	err	x2
$*$ x22	x0	x1	err
err	err	err	err

Solution (explicit). In this solution we construct an explicit DFA that accepts L, thus proving that L is a regular language. We need the DFA to track whether it has seen three consecutive symbols—in which case it goes to an error state—but otherwise it can accept any word. This yields a relatively simple transition table. Unfortunately, the state diagram corresponding to this table is non-planar, so it cannot be easily drawn. ■