

TDT4205 Problem Set 1

Harald Husum

February 2016

1 Regular Languages

1.1

Looking at the DFA in Figure 1 of the Problem set, we intuitively see that it accepts the following regular expression: $0^*10^*1(0|1)^*$. The $(0|1)^*$ -part in it self accepts all binary strings, so the limitation is not an upper bound on the complexity of the strings the automata will accept. Rather the limitation is on how minimal our strings can be. The first part of the expression (0^*10^*1) , can be read as “the string needs to contain at least two 1-es”. Thus strings like ϵ , “0” and “10”, will not be accepted. We can generalize this to $0^*(1^?0)^*$.

1.2

To define a number, we utilize the definition given in Example 3.7 in Compilers:

$$digit \rightarrow [0 - 9]$$

$$digits \rightarrow digit +$$

$$number \rightarrow digits (. digits)? (E [+ -]? digits)?$$

Now; expanding this to a complex number, let us call it “cn”:

$$cn \rightarrow [+ -]? number ([+ -] number? i)?$$

1.3

The fourth rule as given in Compilers states that “(r) is a regular expression denoting $L(r)$ ”. This introduces the possibility of arbitrarily large amounts of balanced parentheses in regular expressions. Since we need some form of backreferencing to handle this balancing, the grammar cannot be regular.

2 A Simple Language for Drawing Lines

2.1

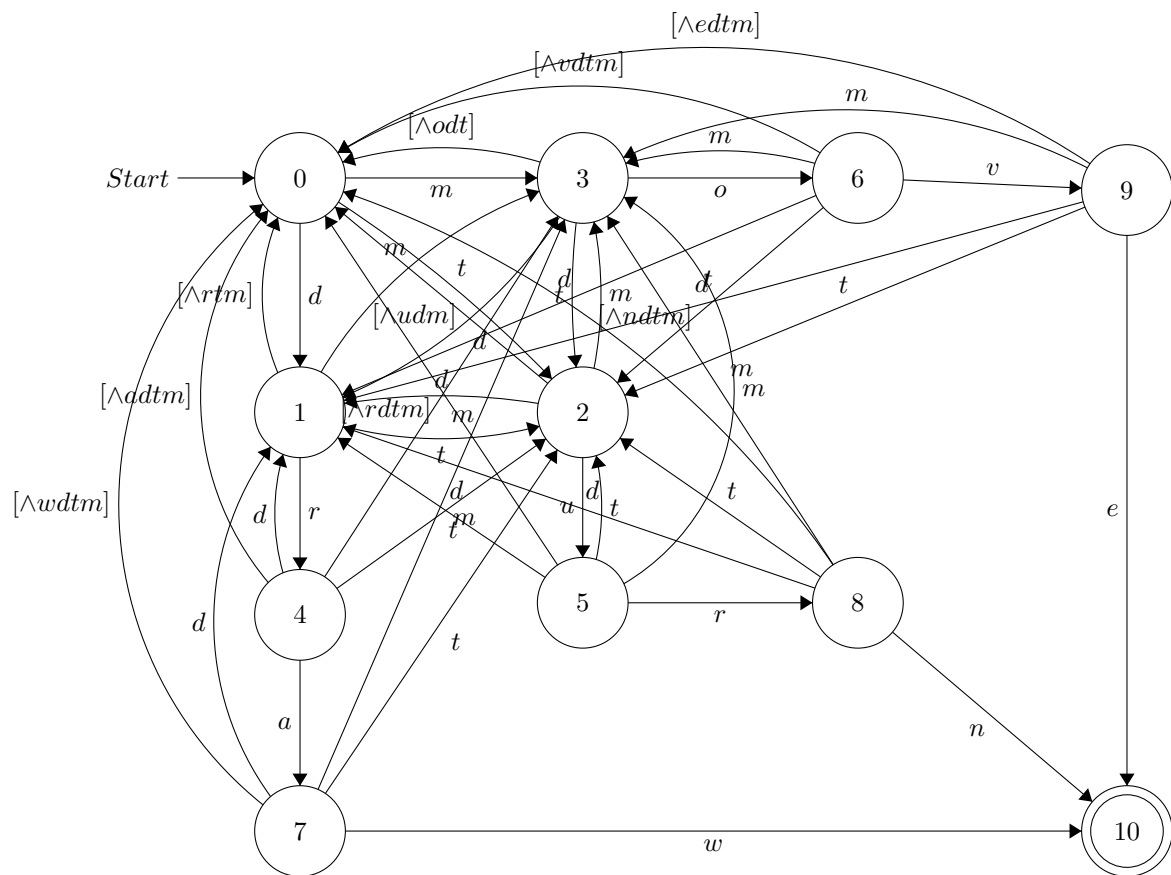


Figure 1: DFA for specified language