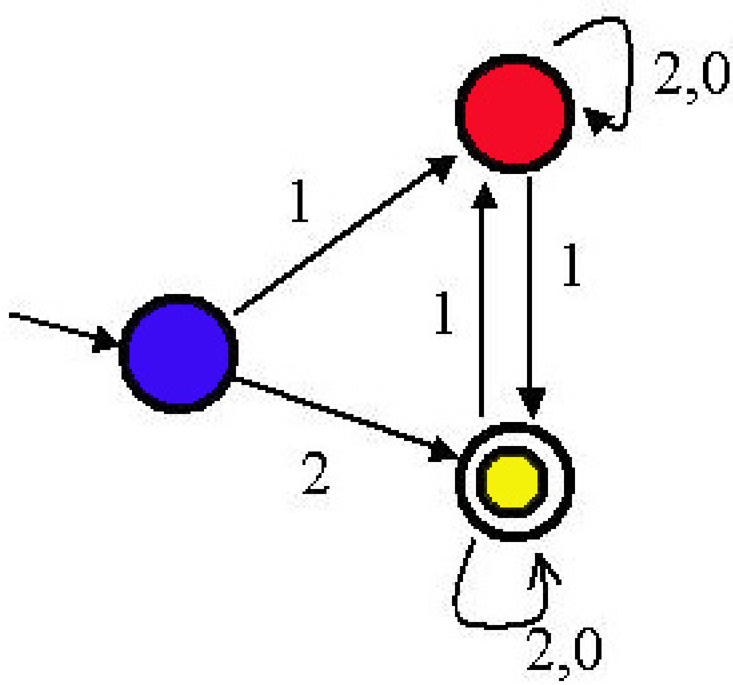
University of Central Florida
School of Computer Science
COT 4210 Spring 2004

Prof. Rene Peralta
Solutions to Homework 2

Consider integers written in base 3 with no leading 0s. Let L be the set of such strings which represent even numbers.

1. Construct a DFA that accepts L.



The red state means "I am odd". The yellow state means "I am even".

2. Construct a left-linear grammar for L.

   Associate with each state U the set of strings whose computation can end in U. Then we have (let B,R,Y denote be blue,red,yellow states, resp.)

$$\begin{aligned} B &\rightarrow \lambda \\ R &\rightarrow (Y+B)1 + R(2+0) \\ Y &\rightarrow B2 + R1 + Y(2+0) \end{aligned}$$

   which simplifies to

$$\begin{aligned} R &\rightarrow (Y+\lambda)1 + R(2+0) \\ Y &\rightarrow 2 + R1 + Y(2+0) \end{aligned}$$

   The starting symbol of the grammar is Y.

3. Write a regular expression for L.

   Using Adler's rule we have

$$R = (Y+\lambda)1(2+0)^*$$

   then

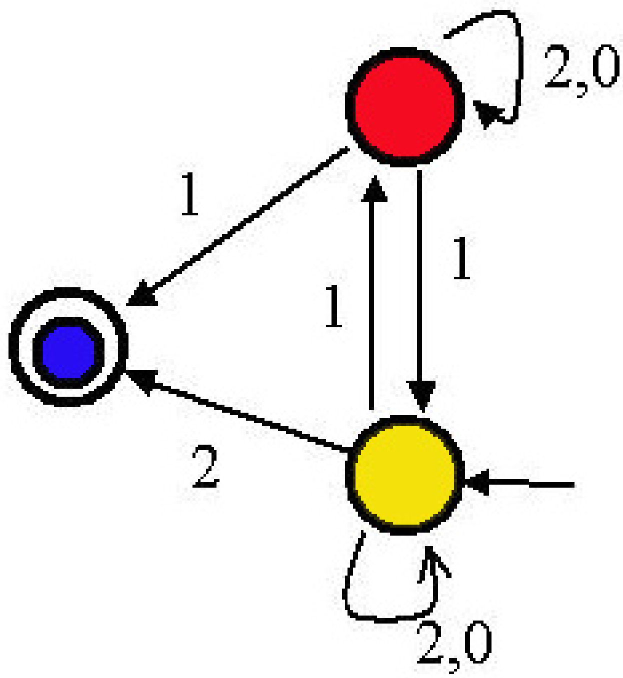
$$Y = 2 + (Y+\lambda)1(2+0)^*1 + Y(2+0)$$

   factoring Y

$$Y = 2 + 1(2+0)^*1 + Y(1(2+0)^*1 + 2 + 0)$$

   and finally

$$Y = (2 + 1(2+0)^*1)(1(2+0)^*1 + 2 + 0)^*$$



4. Write a regular expression for L^r.

   Above is the "reverse" of the automata for L. Just for fun, we can associate with each state U the set of strings accepted if the automaton is <u>started</u> at U. We then have the following system of equations (some obvious steps are skipped)

$$\begin{aligned} R &= (0+2)^*1 + (0+2)^*1Y \\ Y &= (0+2)^*2 + (0+2)^*1R. \end{aligned}$$

   Thus

$$\begin{aligned} Y &= (0+2)^*2 + (0+2)^*1((0+2)^*1 + (0+2)^*1Y) \\ &= (0+2)^*(2 + 1(0+2)^*1) + (0+2)^*1(0+2)^*1Y \\ &= ((0+2)^*1(0+2)^*1)^*(0+2)^*(2+1(0+2)^*1). \end{aligned}$$

This looks suspiciously like a (very ugly) way of saying "an even number of 1" (plus some minor details relating to the "no leading zeroes" constraint). So what's up?

The (radix three) integer represented by a string ω is $n = \sum_i 3^i b_i$, where b_i is the i^{th} symbol in the ω. Since 3 mod 2 = 1, we have

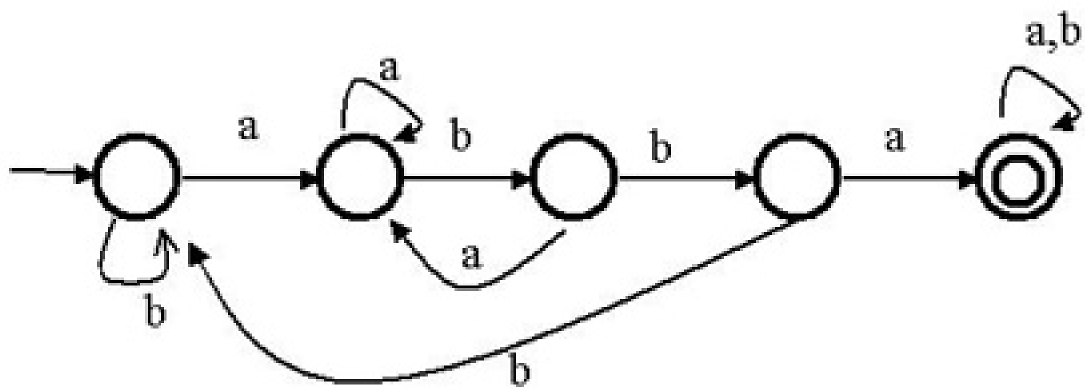
$$n \bmod 2 \equiv \sum_i (b_i \bmod 2)$$

Since 0 and 2 mod 2 are also 0, we have

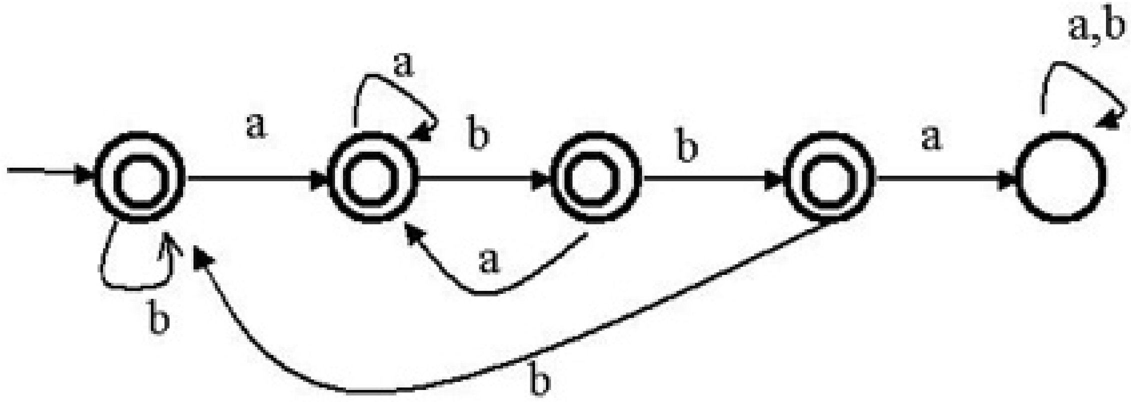
$$n \bmod 2 = (\text{number of 1s in } \omega) \bmod 2.$$

That is, ω represents an even integer if and only if it contains an even number of 1s.

5. Write a grammar for the language (over $\Sigma = \{a, b\}$) consisting of strings not containing the pattern "abba".



The machine above accepts strings that contain the pattern "abba". To "complement" the machine simply switch accepting and non-accepting states:



Writing a grammar is straight-forward:

$$S \to \lambda + aU + bS; \quad U \to \lambda + bV + aU;$$

$$V \to \lambda + aU + bW; \quad W \to \lambda + bS;$$

(why only four non-terminals?).

6. Write a grammar for the language (over $\Sigma = \{a, b\}$) consisting of palindromes with the same number of a's as b's.

   This problem is quite hard. A solution, however, can be easily verified. This phenomenon (finding a solution to a problem is often much more difficult than verifying its correctness) and nobody really knows why. Here is a grammar:

$$\begin{aligned} S &\to aASa + bBSb + \lambda; \\ AB &\to \lambda; \quad BA \to \lambda; \\ Aa &\to aA; \quad aA \to Aa; \end{aligned}$$

   Note this type-0 grammar is not type-1. We know a type-1 grammar exists because the language can be decided in linear space. However, it seems difficult to construct such a grammar.