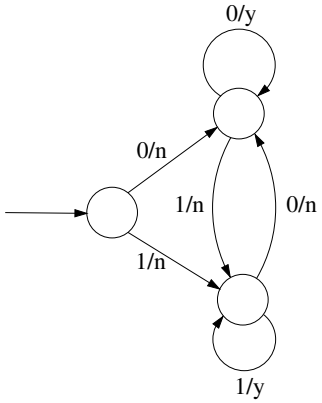
Exercise 1. Mealy machines

A Mealy machine is a 6-tuple $(Q, \Sigma, \Delta, \delta, \lambda, q_0)$ where Q, Σ, δ, and q_0 are defined as for a DFA. The set Δ is called the *output alphabet*, and $\lambda : Q \times \Sigma \to \Delta$ gives the output associated with a transition from state q on input a. We normally write the values of δ and λ next to each other in the transition diagram. For example, the following Mealy machine writes a 'y' symbol for each prefix of its input string that is in the language $(0 + 1)^*(00 + 11)$. For example, it prints "nnyny" on the string "01100".



- For $\Sigma = (0 + 1) \times (0 + 1)$, assume that the input contains the representation of two binary numbers least-significat bit first. Build a Mealy machine that prints the sum of its inputs. For example, for the input string $(0, 1)(1, 0)(1, 1)(0, 0)$, it should print the string 1101. You may assume the last symbol of the input is always $(0, 0)$.

Exercise 2. Regular languages

Whice of the following languages are regular?

- $\{0^m 1^n 0^{m+n} \mid m \geq 1 \wedge n \geq 1\}$
- the set of all strings that do not have three consecutive 0s.
- $\{xwx^R \mid x, w \in (0 + 1)^+\}$ and x^R is x written backward; for example $(011)^R = 110$.
- $\{xx^R w \mid x, w \in (0 + 1)^+\}$

Exercise 3. Regular language

Let L be any subset of 0^*. Prove that L^* is regular.

Exercise 4.

Let L be a language. Define the following languages:

$$\begin{aligned} HALF(L) &= \{x \mid \exists y.|x| = |y| \wedge xy \in L\} \\ SQRT(L) &= \{x \mid \exists y.|y| = |x|^2 \wedge xy \in L\} \\ LOG(L) &= \{x \mid \exists y.|y| = 2^{|x|} \wedge xy \in L\} \end{aligned}$$

Prove that if L is regular, so are the languages $HALF(L)$, $SQRT(L)$, and $LOG(L)$.

Exercise 5.

R is a *congruence relation* if $xRy \Rightarrow wxzRwyz$ for all w and z. Prove that a set is regular iff it is the union of some of the congruence classes of a congruence relation of finite index.

Exercise 6. Laboratory

For the lab exercise, you will need the files `nfa.ml` and `test.txt`, posted on the course web site. You are to:

1. Write a function that produces a DFA from an arbitrary NFA with ϵ transitions.
2. Write the functions to produce a NFA with ϵ transitions from a regular expression.

You can compile the `nfa.ml` file using the command `ocamlc -warn-error A -o nfa nfa.ml`. The program expects a file to parse on the command line. Here is an example run:

```
<jyh:jaoquin 9>cat test.txt
Hello_world             // A u$eless comment
Did you know 1 + 2 < 3.1415926?
Add any other test] str'ings that you _would like!
<jyh:jaoquin 10>./nfa test.txt
Input symbol: "Hello_world"
Input symbol: "\t\t"
Input symbol: "// A u$eless comment"
Input symbol: "\n"
Input symbol: "Did"
Input symbol: " "
Input symbol: "you"
Input symbol: " "
Input symbol: "know"
Input symbol: " "
Input symbol: "1"
Input symbol: " "
Input symbol: "+"
Input symbol: " "
Input symbol: "2"
Input symbol: " "
Input symbol: "<"
Input symbol: " "
Input symbol: "3"
Input symbol: "."
Input symbol: "1415926"
Input symbol: "?"
Input symbol: "\n"
Input symbol: "Add"
Input symbol: " "
Input symbol: "any"
Input symbol: " "
Input symbol: "other"
Input symbol: " "
Input symbol: "test"
Input symbol: "]"
Input symbol: " "
Input symbol: "str'ings"
Input symbol: " "
Input symbol: "that"
Input symbol: " "
Input symbol: "you"
Input symbol: " "
Input symbol: "_would"
Input symbol: " "
```

```
Input symbol: "like"
Input symbol: "!"
```

What to turn in.

Using the command `cs20-submit`, submit the following:

- A `README` text file describing what you did. If you have any problems, mention it in the README file.
- Your completed `nfa.ml` file.
- A transcript of a successful test run of your completed program. Testing is important; more credit will be given to concise test suites that illustrate the correctness properties of your code.