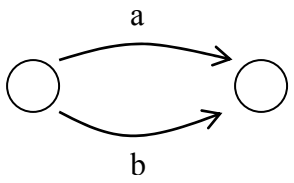
Regular Expressions and Finite Automata

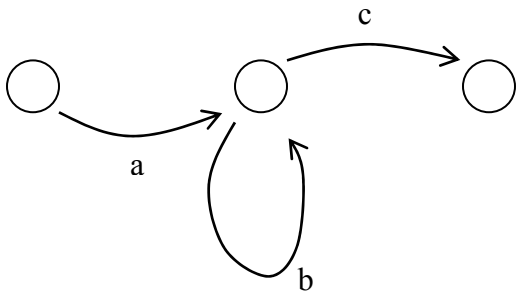
This text shows how to convert a regular expression into a graphical representation of a finite automata and then to an algoritm that analyzes this automata. However, the techniques shown are informal — some regular expressions cannot be transformed by them.

Each of the items below show how to convert a regular expression into the graphical representation of a finite automata.

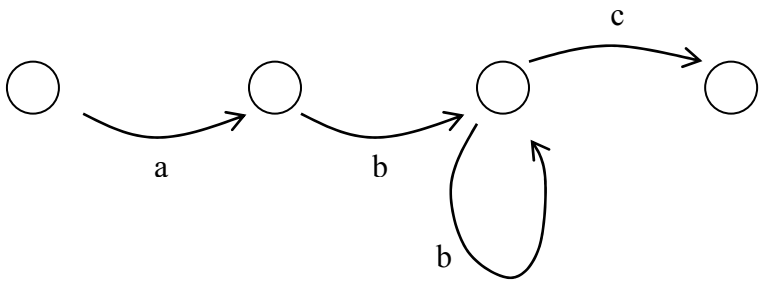
a) or. Example: a | b



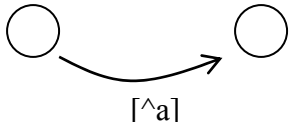
b) zero or more repetitions. Example: ab*c



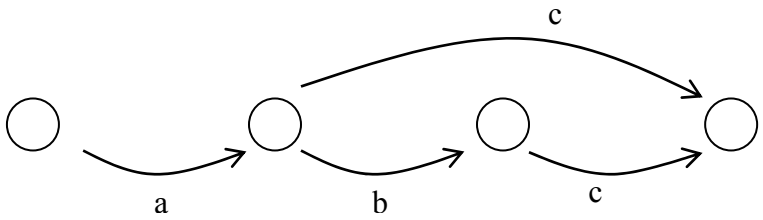
c) One or more repetitions. Example: ab+c



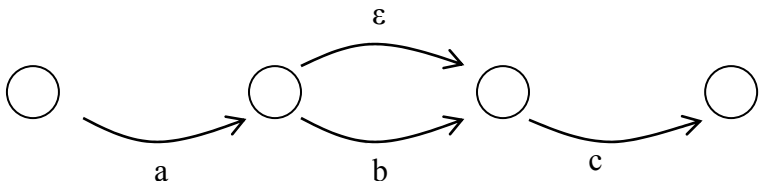
d) Everything but something. Example: [^a]



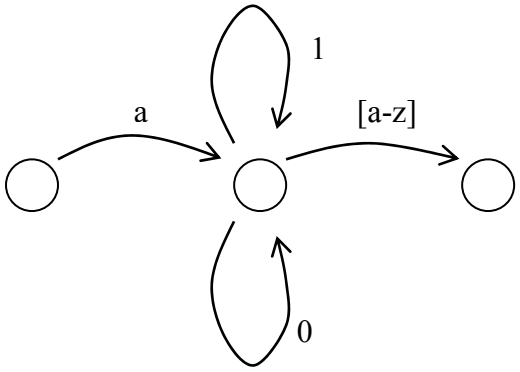
e) Option. Example: ab?c



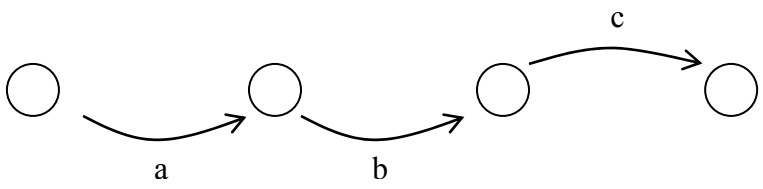
or



f) Repetition of optional things. Example: a(0|1)*[a-z]



g) Sequence of things. Example: abc



Now we show how to convert a regular expression into a Java method that recognizes the expression. The example should explain itself.

An example of a regular expression would be:

```
a(0|1)*#+[a-z](a|b)0*
```

The analyzer to recognize it would be:

```
void analyze(char in[]) {
   int s = 1;
   int k = 0;
   while ( s != 6 ) {
      switch (s) {
         case 1:
            if ( in[k] != 'a' )
               error();
            else {
               k++;
               s = 2;
            }
            break;
         case 2:
            if ( in[k] == '0' || in[k] == '1' )
               k++;
            else  if ( in[k] == '#' ) {
               k++;
               s = 3;
            }
            else
               error();
            break;

         case 3:
            if ( in[k] == '#' )
               k++;
            else if ( in[k] >= 'a' && in[k] <= 'z' ) {
               k++;
```

```
               s = 4;
            }
            else
               error();
            break;
         case 4:
            if ( in[k] == 'a' || in[k] == 'b' ) {
               k++;
               s = 5;
            }
            else
               error();
            break;
         case 5:
            if ( in[k] == '0' )
               k++;
            else
               s = 6;
            break;
      }
   }
}
```

Caveat: some regular expressions cannot be converted into a finite automaton by the techniques given in this manual. For example, the method fails when usign the regular expressions

```
b*b
a+a
(a|b)+ab
b?b
```

Why? In b*b, the automaton would advance through all b's in the input, consuming even the b that would match the last b in "b*b".