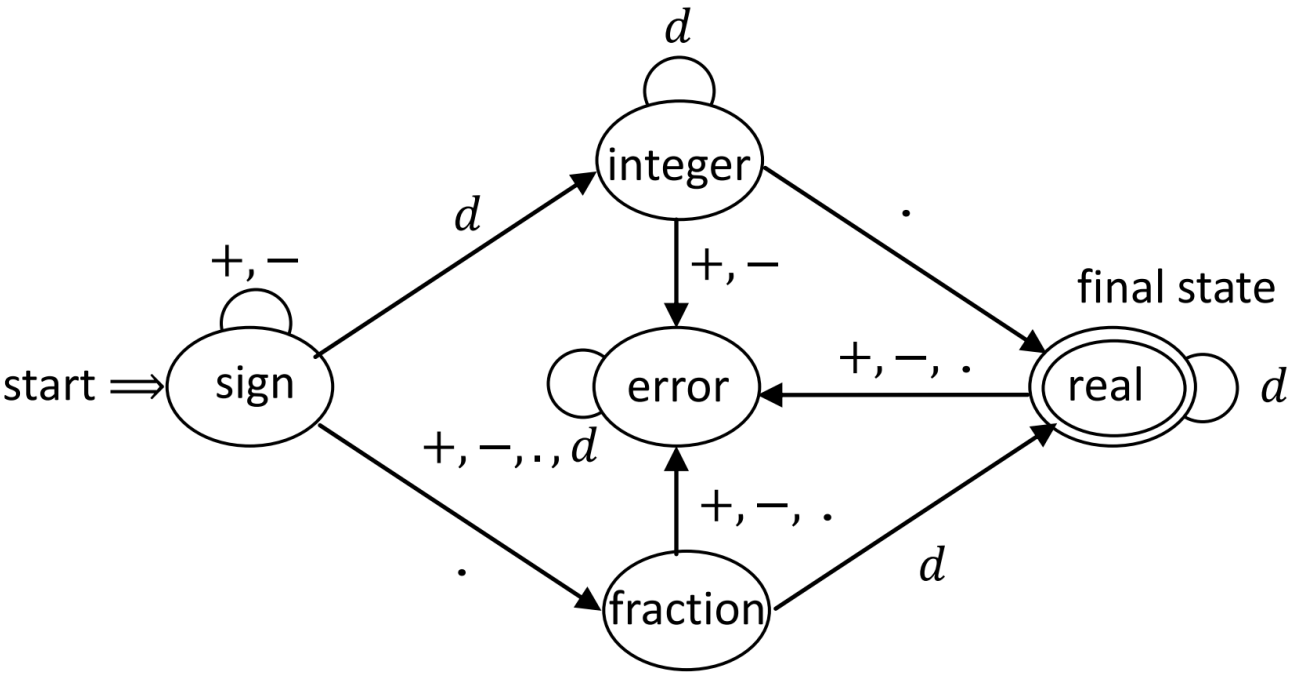
Homework #5

Due date: 21:30, November 3rd, Tuesday, 2015

Real constant recogniser

Let Σ = {+, −, . , 0, 1, 2, 3, 4, 5, 6, 7, 8, 9} be the alphabet of the language of real constants generated by the following finite automaton, where $d = 0, 1, 2, \ldots, 9$.



The real constants accepted (or recognised) by this finite automaton are of the form

$$s_1 s_1 \ldots s_m d_1 d_2 \ldots d_n . e_1 e_2 \ldots e_p$$

where s_i is a plus or minus sign, d_j and e_k are decimal digits, and m , n , p satisfy

(1) $m \geq 0$ and (2) $n \geq 0$, $p \geq 0$, but not both zero

For examples, the following real constants are accepted

```
12.34 12. .34 -12.3 +2.34 +-+12.34    *
```

but the following aren't

```
1234 1.2.3 +1..2 .+23 . +-+-.         *
```

Comment

The real constants recognised by this finite automaton are essentially those of C/C++.

In particular, the real constants in the red-starred line are also legal in C/C++, and those in the blue-starred line are also illegal in C/C++.

The only difference is that **++** and **--** are consecutive here, but they must be separated by spaces in C/C++. For examples, the finite automaton accepts

```
++2.3      --2.3
```

which must be written in C/C++ as

```
+ +2.3     - -2.3
```

Your job is to implement the preceding finite automaton in two ways:

1. Represent states as functions
2. Represent states as constant integers, say

```
const int sign = 0,
          integer = 1,
          fraction = 2,
          real = 3,
          error = 4;
```

   Determine the next state to transit by computation

Requirements

1. You shall write two functions, say

```
void recogniser1(void); // for method 1
void recogniser2(void); // for method 2
```

2. Use the following code to test your functions

```
switch (rand()%2) {
case 0: recogniser1(); break;
case 1: recogniser2(); break;
}
```

   It is up to you to decide if you want to set a new seed for the pseudorandom number generator.
3. See the sample run for the required output format.
   The sample run uses the default seed. The method used to recognise each test datum may be different if a different seed is employed.

Submission

Be sure to upload your source code to E3 by the due date and name your file as "**HW5_xxxxxxx.c**", where **xxxxxxx** is your student ID.

Sample run

```
123.45
Accepted by Method 2

123.
Accepted by Method 1

.45
Accepted by Method 2

+23.456
Accepted by Method 2

-0.
Accepted by Method 2

+.0
Accepted by Method 2
```

```
++--23.45
Accepted by Method 1

1234
Rejected by Method 1

1..2
Rejected by Method 2

1.2.3
Rejected by Method 2

+12.+34
Rejected by Method 1

+123.45+
Rejected by Method 2

.
Rejected by Method 1

+-+-.
Rejected by Method 2

^Z
```