FSM Project

October 1, 2019

Abstract

In this project you should implement simple automatic code generator and estimate feasibility of making it bigger.
Submission is in two parts:

- Code generator software (7% of course marks)
- Estimating feasibility of the project (3% of course marks)

Submission due on October 14.

State and input names can be any string. It can be ”s0” and ”i0”, as in the example below, it also can be ”open” and ”push” or any other string.

1 Problem description

It is description of Project objectives.
For actual question see next section.
You are provided with specification of the FSM as a text file. Each line of this file describes one state of the FSM in format:

sourceState1,input1:destState1,input2:destState2

For example, file:

s0,i0:s1,i1:s0
s1,i0:s3,i1:s2
s2,i0:s0
s3,X

describes FSM shown at Fig.1.

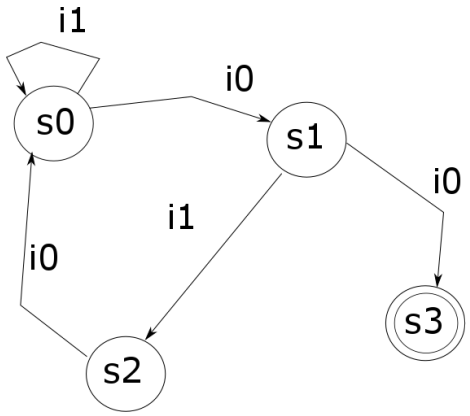

Figure 1: Example FSM

X marks the accepting state. First line describes **start** state.

Your software should read text file with FSM specifications and generate cpp file which implements this FSM. cpp file should compile and run without any modifications, i.e. it should contain all **#include**s, **int main()** and curly brackets in proper places.

As an example, generated code for FSM shown in Fig,1 is printed at Listing below. Code formatting leaves a lot to be desired.

Consider C++ program as **vector** of **strings**.
Some parts of the program are fixed. Lines 1 to 6 and lines 37 to 45 in the listing below are not changing - you can fill them before program starts generating

custom part of the code. In this example it is **Proc()** function.
Your software should produce whole cpp file, but only part which has to be customized is **Proc()** function.

Proposed algorithm (but it is not compulsory to use and you can do better):

1. Fill fixed parts of the program
2. Read one line of FSM specification file
3. Parse the string just read. FSM approach works well here.
4. Insert strings into the code vector of strings

```
1    if (state = xxx) = {
```

   where xxx is name of the state - first token in the string

5. For each of pair **i:s** insert string

```
1       if (input == i) { state=s; return 0;}
```

   Do not forget opening and closing curly brackets.

6. Repeat for next line of the specifications file (i.e. next state) until End Of File
7. Do not forget closing bracket }.

After that it should produce following minimum code:

Listing 1: Minimum generated code

```
1  #include <iostream>
2  using namespace std;
3  string state;
4  string input;
5  int accept = 0;
6  int Proc(string input)
7  {
8   if ( state=="s0")
9   {
10    if ( input=="i0")
11    {
12      state="s1";
13    } else
14    if ( input=="s1")
15    {
16      state="s0";
17    }
18  } else
19  if ( state=="s1")
20  {
21    if ( input=="i0")
22    {
23     state="s3";
24    }
25    if ( input=="i1")
```

```
26      {
27        state="s2"; return 0;
28      }
29    } else
30    if ( state=="s2")
31    {
32      if ( input=="i0")
33      {
34       state="s0"; return 0;
35      }
36    }
37    if (state == "s3") return 1; // accepting state
38    return 0;
39  }
40
41  int main(){
42   state = "s0";
43   while(1){
44     cout<<" state="<<state<<" enter signal"<<endl;
45     cin>>input;
46     Proc(input);
47    cout<<" new state="<<state<<endl; // p
48   }
49  }
```

There

2 Marking of the code

- Your program reads specification file. Your program generates C++ code which looks OK - 50%.
- Generated code compiles and runs - 15%.
- Generated code recognizes that one of the states is **accepting** and prints out that FSM is in this state - 15 %.
- Your program rejects (error message on the screen) malformed (multiple commas **,** and colons **:**) specification string - 20%

3 Report

You are employee of medium-size company. Your manager (not exactly technical person) asks you to estimate if company should invest in development of automatic code generator as he heard that you have been working on it as a hobby. He thinks that such generator can make coding a trivial task producing perfect code each time based on customer requirements.
Project time-frame is two years.

Your task:

1. Make a report of what you achieved so far with your code generator.

Include description of your software (1 page).

2. Estimate if something like that is possible at all. You should find and review some technical reports and scientific papers. 5 papers will be sufficient. You don't have to understand all the details but should estimate feasibility of the project.
2. Write letter to the boss with your recommendations. Letter can start: "Dear Bill (or Steve, or Mark)! Regarding our conversation...". No more then 5 pages. Keep in mind that Mr. Boss wants this code generator badly but it will be your responsibility to implement the project if you recommend to proceed.
Your letter should contain:

- Description of your software, its limitations and strength (3 pages maximum)
- Is software like that available commercially or as an open source?
- What are research efforts in this area?
- Do you recommend to proceed with the project?

Explain the reasoning for the recommendation.

4 Submission

- Code: Please submit your zipped cpp files.
- Letter - pdf file, not longer than 5 pages