Programming Assignment 1

Automata Theory Monsoon 2021, IIIT Hyderabad

November 11, 2021

General Instructions: Read the input and output formats very carefully since the evaluation will be **automated**. The goal is not to stress test your code but to see if you have understood the underlying concepts used in the implementation. There will be a evaluation / viva after the submission, and a part of the grade would depend on it.
Language constraints: The submissions for $Q1$ and $Q2$ are allowed to be either in Javascript or Haskell, using *only* the packages listed in this document. If you want to use any other renderer, you may refer to one of the TAs for approval. In case of $Q3$ and $Q4$, submissions must be C/C++ only and usage of any other language shall lead to no evaluation.

---

1. [15 points] **Simulating L-Systems**
   Due Date: 31 October 2021

   Given the following production rules generate graphics for them using either the Javascript or the Haskell modelling package. Upload an image after the minimum number of iterations mentioned for each sub-part. You may also upload a gif to show the evolution. You may use the following packages:

   - Javascript
   - Haskell

   1. "I'm a mirrorball" (minimum 9 iterations)

      Axiom: G
      Angle: 30
      Rule 1: $G = X - G - X$
      Rule 2: $X = G + Y + G$
      Rule 3: $Y = [+F]F[-F]$

   2. "Is that a tree?" (minimum 4 iterations)

      Axiom: X
      Angle: $(r \bmod 30) - 15 + 10 \times (-1)^{15-(r \bmod 30)}$
      Rule 1: $X = F[-X]X[+X][+X]F - [-X] + X[-X]$
      Rule 2: $F = FF$

      Here, r stands for the last three digits of your roll number. For example, if 2019111234 is your roll number, for you $r = 234$.

3. "Anything that can happen will happen" (minimum 4 iterations)

   Axiom: X
   Angle: 12.5
   Rule 1: $X = F - [[-X] + X] + F[+FX] - X,\ F + [[+X] - X] - F[-FX] + X[X]$
   Rule 2: $F = F[F]F,\ F[+]F,\ F[FF]F$

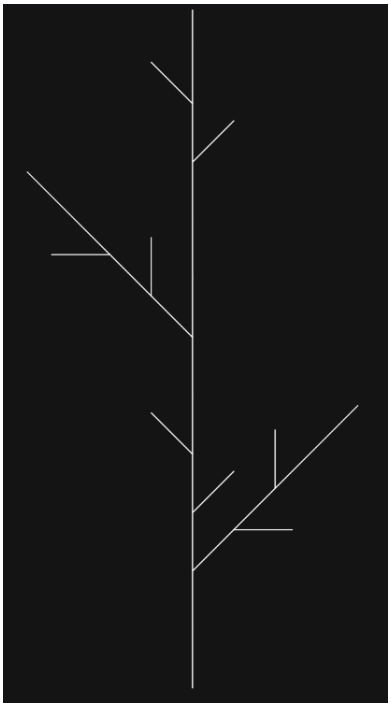
   Each comma separated rule is equally probable. Provide 5 images displaying different L-system configurations obtained from this set of axioms and rules.

4. "Content without context is noise" (minimum 9 iterations)

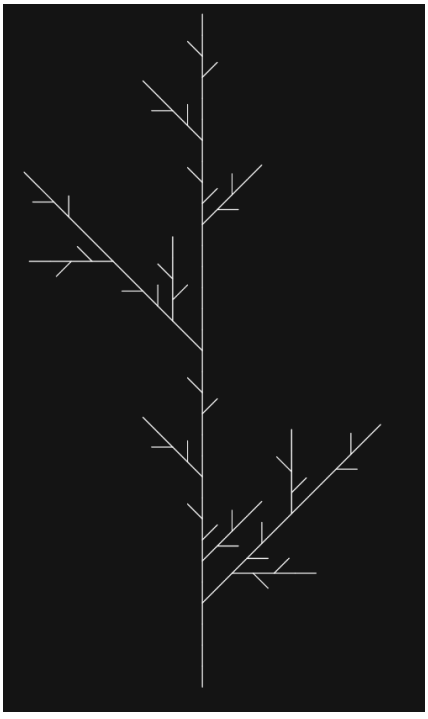
   Axiom: $F + F$
   Angle: 90
   Rule 1: $F = F - F + +F - F$
   Rule 2: $F \langle F \rangle - F = F - +F + F$

2. [10 points] **L-Systems: Find the axioms**
   Due Date: 31 October 2021

   Given the following 2 sets of graphics or final system configurations, come up with the underlying production rules to model that system.

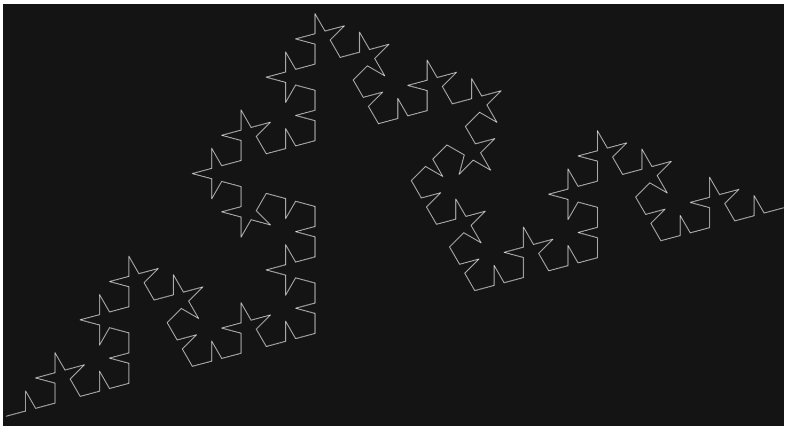


(a) after 2 iterations

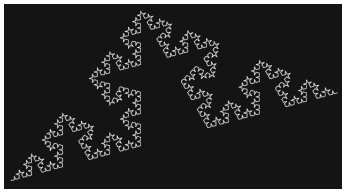


(b) after 3 iterations

Figure 1: Stick Plant



(a) after 4 iterations



(b) after 5 iterations

Figure 2: Santa K(l)osh

For each of the above sub-parts submit a JSON file in the following format. File name for sub-part i: '*i.json*':

```
{
   "axiom": ···,
   "productions": {···},
   "finals": {···}
   ···
}
```

3. [40 points] **Minimization of DFA**
Due date: 26 November 2021
You are given a DFA D which has n states, k transition rules and, a accept states. Your goal is to maintain the functionality of D and reduce n and k.
Constraints on D:
To avoid confusion, assume any property of DFA as the property of D. To cover the grey areas, here are explicit constraints:

- 0 is the start state of the DFA.
- There can be multiple accept states.
- The alphabet can be assumed to be restricted to the symbols used in transition rules.
- The states are represented by 0 to $n-1$
- The alphabet contains lower-case letters only ($a-z$).

Input format: First line will contain 3 numbers n, k and, a respectively.
Next line will consist of a numbers which represent the set of accept states. Next k lines will consist of the transition rules in the format: $s_1\ x\ s_2$. s_1 is the initial state, x is the symbol and s_2 is the final state.

$n\ k\ a$
$a_0\ a_1\ \ldots\ a_{n-1}$
$s_{01}\ x_0\ s_{02}$
$\vdots$
$s_{n-1\ 1}\ x_{n-1}\ s_{n-1\ 2}$

Output format: First line will contain 3 numbers n', k' and, a' respectively.
Next line will consist of a' numbers which represent the set of accept states. Next k' lines will consist of the transition rules in the format: $s_1\ x\ s_2$. s_1 is the initial state, x is the symbol and s_2 is the final state.
Just as assumed in the input format, the output DFA D' must have states represented by numbers between 0 and $n' - 1$.

$n'\ k'\ a'$
$a'_0\ a'_1\ \ldots\ a'_{n-1}$
$s'_{01}\ x'_0\ s'_{02}$
$\vdots$
$s'_{n-1\ 1}\ x'_{n-1}\ s'_{n-1\ 2}$

4. [30 points] **REGEX to NFA**
Due date: 26 November 2021

Write a program to convert a given regular expression into its equivalent NFA.

Input Format
The input will be a file containing the regular expression in a single line as follows:

any_valid_regex

The regex alphabet will only consist of lower-case letters ($a - z$). The operators that will be used are:

1. $+$ for union
2. $*$ for closure/star operation
3. Continuous letters will signify concatenation
4. () for grouping expressions

For example,

$(a + b)^* + ba + c^*$
where the letters $a,\ b$ *and* c comprise the alphabet.

Output Format
The output should be saved to a separate file. The NFA format must be in the same format as the

DFA format mentioned above. The same assumptions/constraints apply as well. For ϵ transitions, use the E symbol to represent ϵ.

Run Instructions
The code will be run as:

```
./a.out < input_file > < output_file >
```

where the input is to be read from the given $< input\_file >$ and the output is to be stored in $< output\_file >$.

Submission details

The submission must be a zip file containing the code and a report explaining the approach for the solutions.

```
rollnumber_prog1.zip
→ q1
    → ./src
    → ./images
→ q2
    → 1.json
    → 2.json
→ q3.cpp
→ q4.cpp
→ report.pdf
```

In $q1$ 'images' should provide the system configurations you have generated after n iterations and 'src' should contain all the code pertaining to the rendering of the systems. Describe in your report where code related to each sub-part of $q1$ is present in 'src'.
In $q2$, for each sub-part of the question, we should have one JSON file. Briefly mention your ideas and approaches in the report.