# CS 419 Compiler Projects Form

1. **Instructions to be Followed (for the Hard Copy):**
   1. Fill page **NO 4** with the required Fields:
      1. Project Idea: The **Idea** will be assigned to you .
      2. Team Members NO#: Number of team members **7**
      3. Table:
         * ID: Your FCIH ID.
         * Name: Your **Full Name** as registered on College’s Database.
         * Level & Department: Your **(Current)** level and department.
         * Section(Day-from-to): Your Section Day and time slot.
         * Role: Your role in project (**Team leader** OR **Member**).
         * Fill Page **NO 5 & 6** with your (**Regular Expressions**, **Finite automata**, **Parse trees** and **abstract syntax tree**) respectively.
2. **Minus Policies:**
   1. **Project Policy:** affects all projects members including team leader.
   2. **Member Policy:** affects a member of project’s members.
3. **General Notes:**
   1. Total grade of Project is 15
   2. **Deadline** to register yourself and your team on online form **Tuesday 05/04/2022 at 11:59 PM** after that **-2 Project Policy** will be applied**.**
   3. Once you Registered, **NO modifications** will be done.
   4. Allowed only on registration for team in form, duplication will got **-2 Project Policy.**
   5. Each group will be assigned **an Idea, ID and time slot** for **discussion**.
   6. Each team member and team leader in a team **must work in project’s coding phase** (including implementation of **finite automata and parse trees**).
4. **Discussion Notes:**
   1. **Copied Code** will be got ZERO Without Discussion.
   2. By references to Section 3 (General notes) Point V , **-5 Member policy** will be applied to each team member (including team leader) who does not participate in project coding phase **as well as the team leader who does not report this case**.
   3. Each team member must have **a complete knowledge** about the whole project
   4. **Evaluation** will be **Individual Evaluation** not project Evaluation.
   5. **-2 Project Policy** will be applied in case of being late for assigned discussion time slot
   6. **NO discussion will be repeated under any circumstances.**
   7. At Discussion day, in case of offline discussions, each team must have **Hard Copy form** including (**Finite automata and parse trees of team’s project**).
   8. Discussion Day will be **sent later** .

## Notes about Implementation:

* 1. **.Net or PHP** are only allowed.
  2. The project must be a Web (use latest technologies).
  3. Your code must be uploaded to github before discussion.
  4. **- 5 Project policy** will be applied in case of using **Built-in Method** within implementation of the scanner or parser, you must create your **own methods to match** for ex (your regular Expressions).
  5. Each Project must contain a full functional editor (comment, uncomment, put red line under wrong words, auto complete, navigation to function or class, line NO).
  6. Each Project must contain **two buttons** , one button called “**Scan**” to run scanner and other called “**Parse**” to run parser –parser must take output of scanner to do it’s task.
  7. Each project must contain a button named “**Browse**” that allows us to choose a **file from a disk** that allows us to parse or scan this file **Without Showing what is inside the file** and shows the output.
  8. – 3 Project policy will be applied if the content of the file that is mentioned in point V is opened or viewed.

## Notes about Discussion Testing:

* + There will be two types of Testing :

1. **White Box Testing:** This will be from **Editor**.
2. **Black Box Testing:** This will be from “**Browse**” Button

**Thanks,**

# CS 419 Compiler Project Form

**Project Idea:**

Project compiler 1

**Team Members NO#:** 28

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Name** | **Level& Department** | **Section(Day- from-to)** | **Role (Lead/Member)** | **Grade** |
| 201900246 | جورج علاء كرم فاخر | 3 - cs | الاربعاء -12 | member |  |
| 201900800 | مريم اسامة يوسف عبد الملاك | 3 -cs |  | member |  |
| 201900293 | رانيا خالد جابر عبد التواب محمد | 3 -cs | الاربعاء -12 | lead |  |
| 201900306 | رنا عادل فرج احمد الراشدي | 3 -cs | الاربعاء -12 | Member |  |
| 201900355 | سمية شريف محمد محمد | 3 -cs | الأربعاء -12 | Memder |  |
| 201900300 | رضوى عبدالله محمد احمد | 3 -cs | الاربعاء -12 | Member |  |
| 201900351 | سلمى ناصر محمد احمد | 3 -cs | الاربعاء -12 | Member |  |

## Regular Expression, Finite automata and Conversion from RegX to NFA, NFA to DFA

Regular Expression

1. Datatybe:-

(Iow|SIow|Chlo|Chain|Iowf|SIowf)

1. Function:-

(datatype|Worthless) identifer(){}

1. Identifier:-

String(Sting|consitant)\*

1. String:-

([A-Za-z])+

1. relationalOp

(identifer | consitant)(== |<| >| <=| >=| =!) (identifer | consitant)

1. logicOp :-

(identifer | consitant)(&&| || | ~) (identifer | consitant)

1. ArithmeticOp:-

(identifer | conistant)(+|-|\*|/) (identifer | conistant)

1. assignmentOp:-

identifer =( identifer | conistant)

1. Condition :-

If(identifer| conistant) relationalOp (identifer| conistant)({})?

Else({})?

1. Integer :-

Iow(identifier(=identifer|conistant)?|

function);

1. SInteger :-

SIow (identifer (= (+ | - )? conistant)?|function);

1. Character :-

Chlo (identifer (=[A-Za-z])?|function);

1. String :-

Chain (identifer (=[A-Za-z]+)?|function);

1. Float :-

Iowf (identifer (=identifer | digits(. digits)?)?|function);

1. SFloat :-

SIowf (identifer (=[+]?digits(.digits)?)?|function);

1. Void :-

Worthless function

1. Loop :-

Loopwhen/Iteratewhen (identifer =( identifer |digit); ((identifer |digit)relationalOp(identifer |digit)); identifer (ArithmeticOp)=digit)

1. Return:-

Turnback (identifer |digit|function)?

1. Break :-

Stop;

1. Struct :-

Loli identifier {(datatybe(identifier|function);)\*};

1. accessOp:-

identifer -> identifer

1. Braces:-

[Conistant\*]|

{ (datetype|conistant|string|identifier)\*}

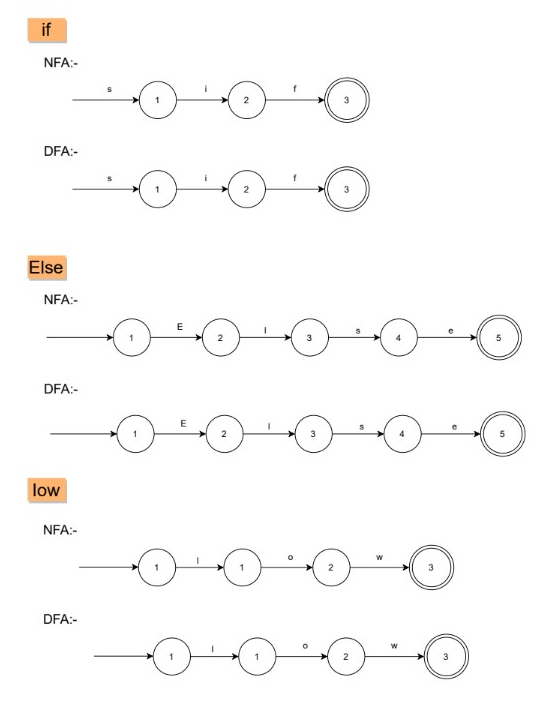
1. QutationMark:-

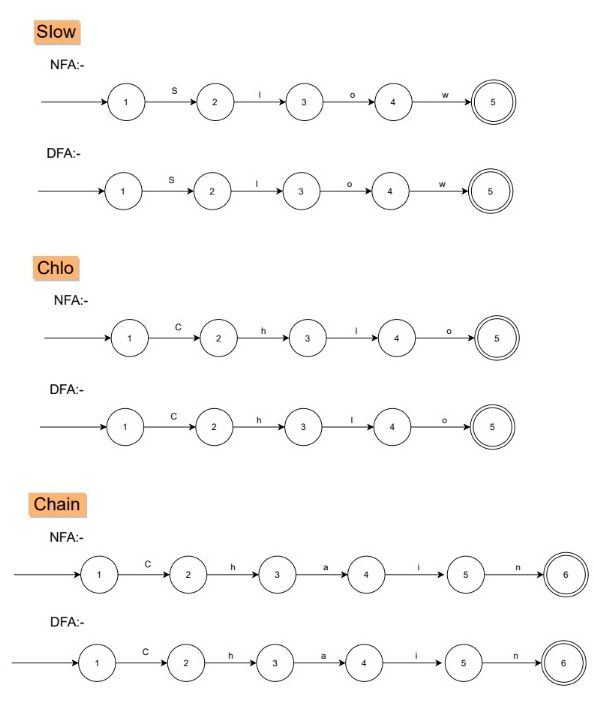
('(Sting|Consitant)'|"(Sting|Consitant)")\*

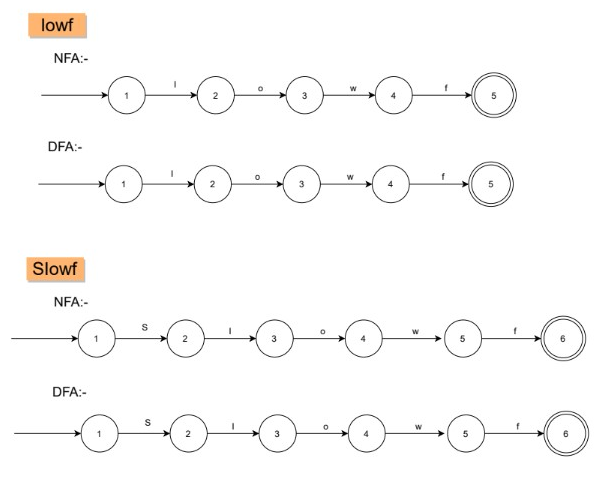
1. Inclusion:-

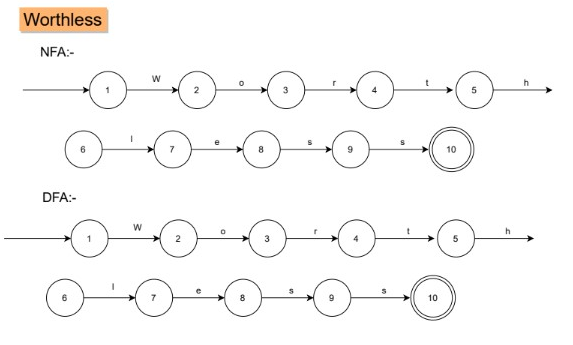
(Include)(QutationMark);

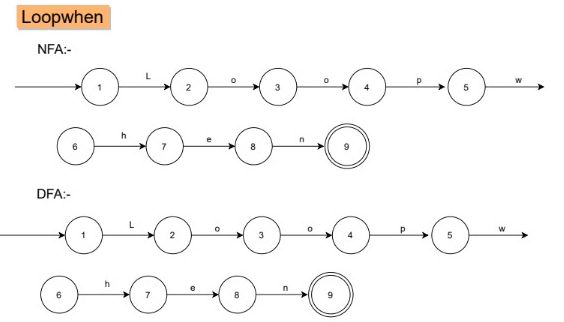
Finite automata and Conversion from RegX to NFA, NFA to DFA

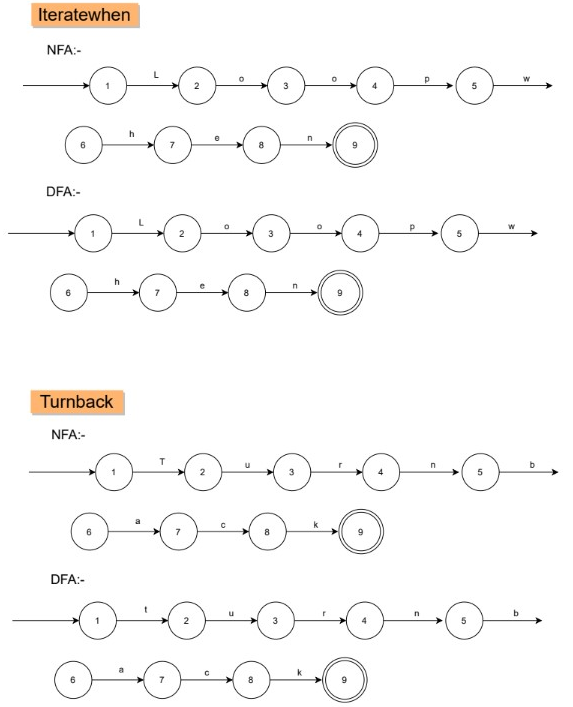
****

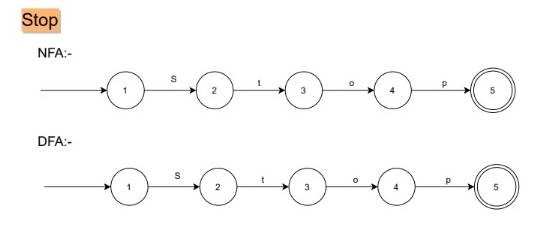
****

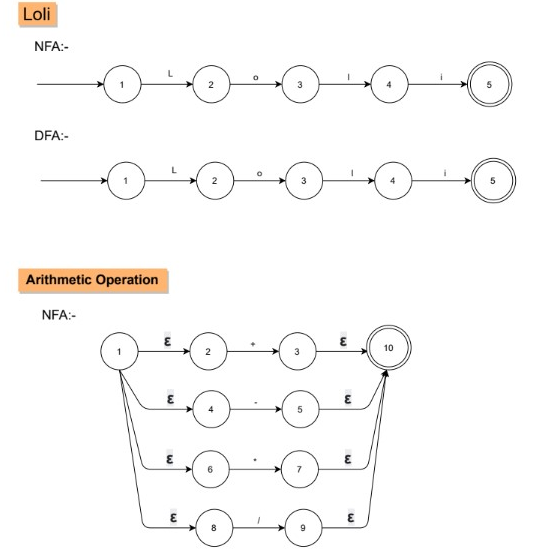
****

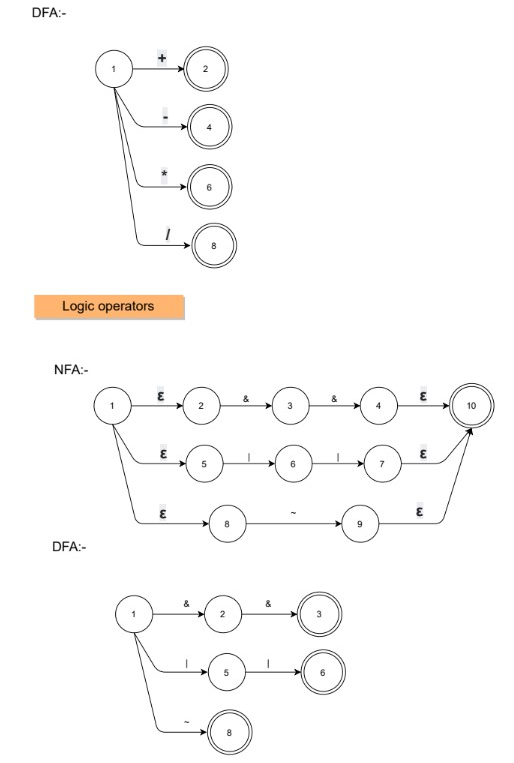
****

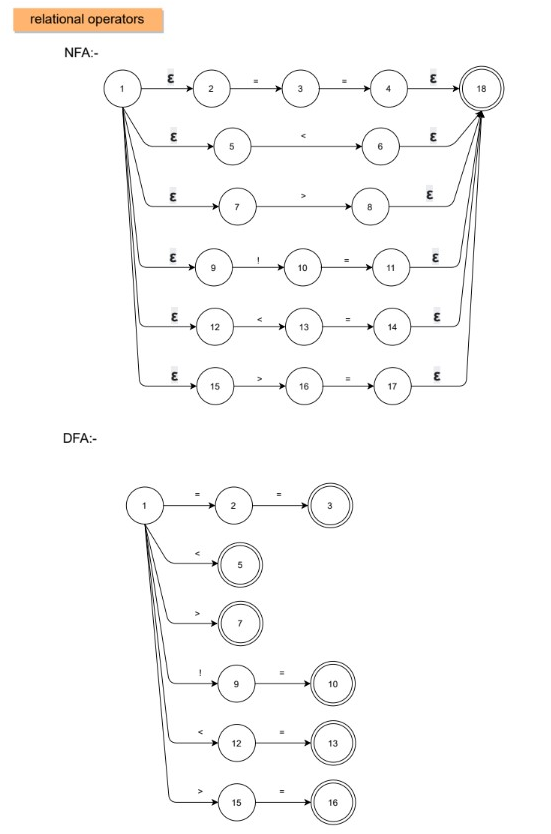
****

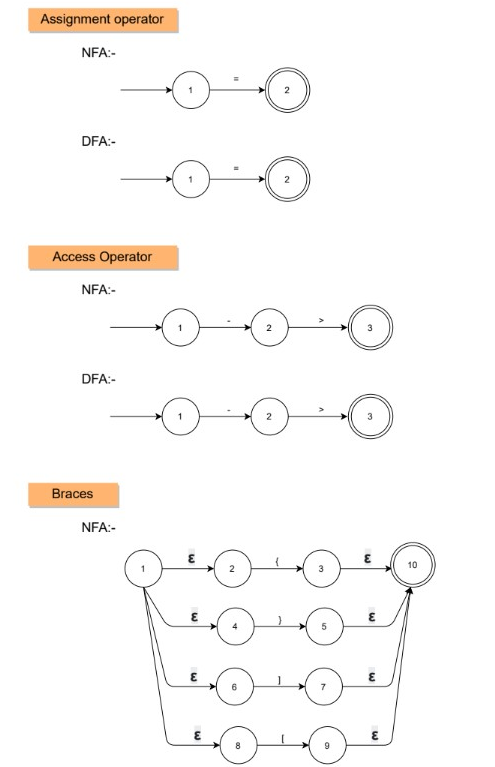
****

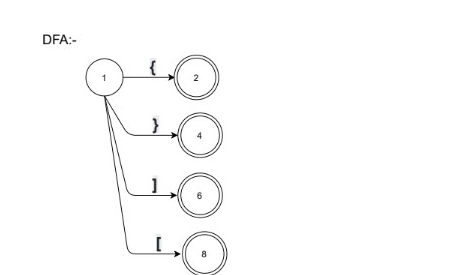
****

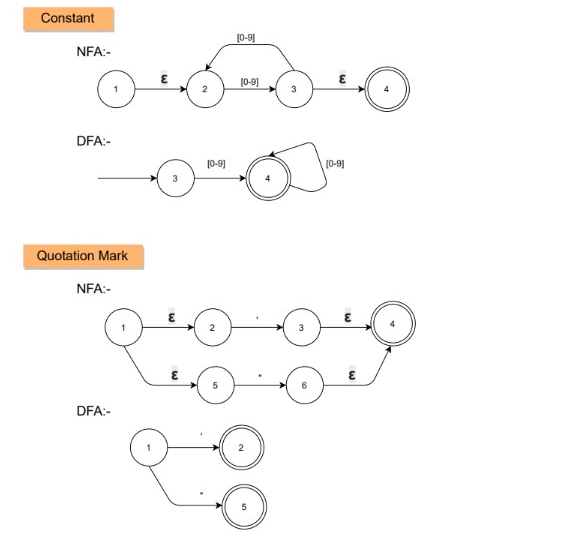
****

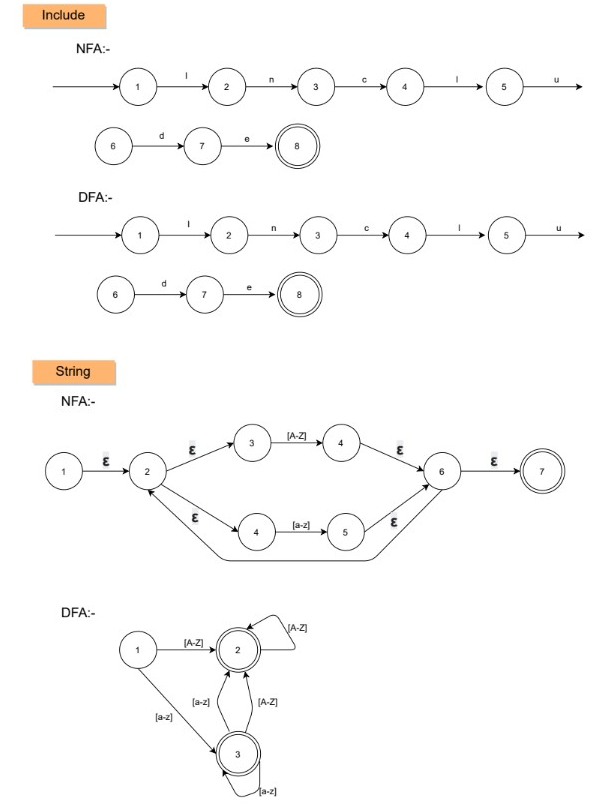
****

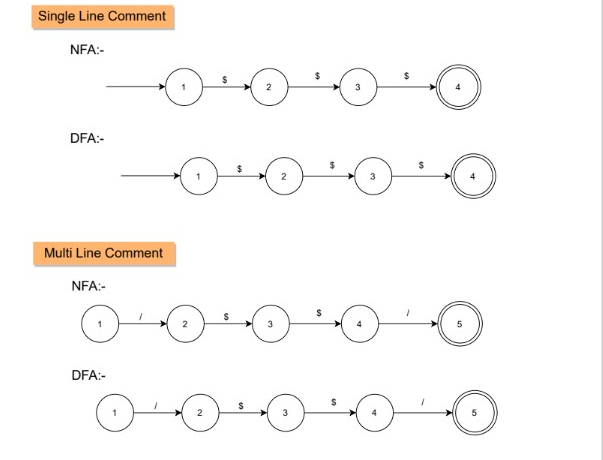
****

****

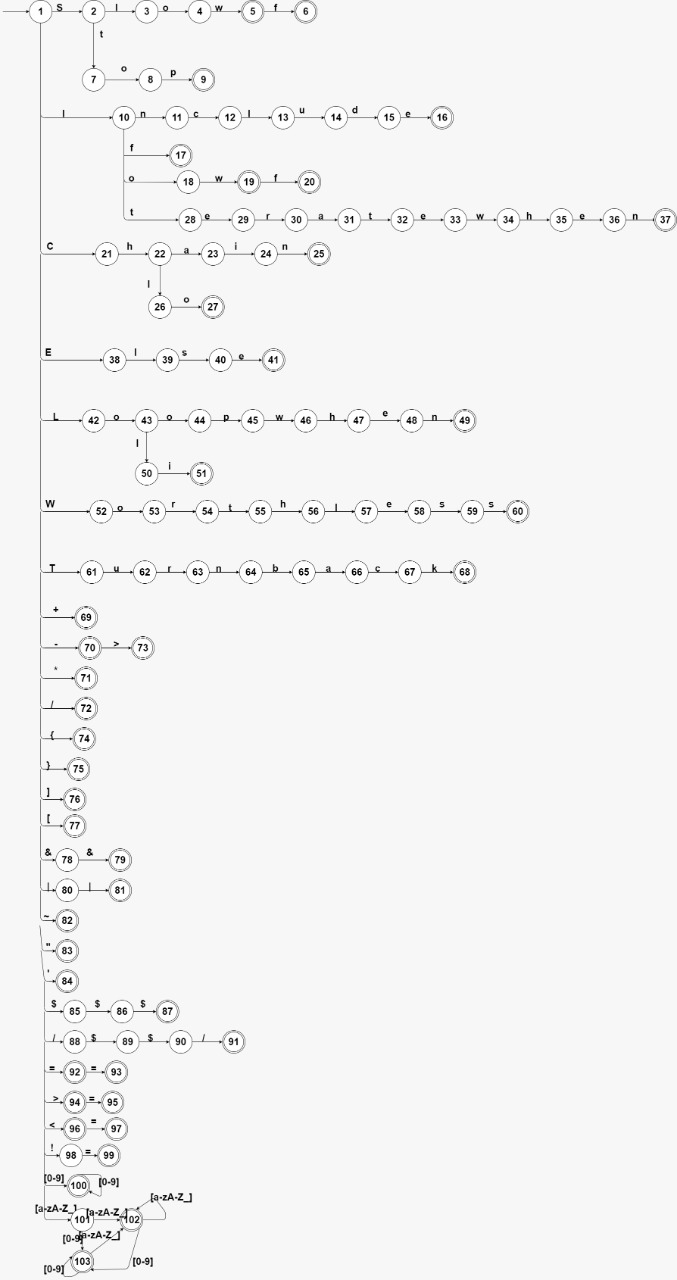
****

****

****

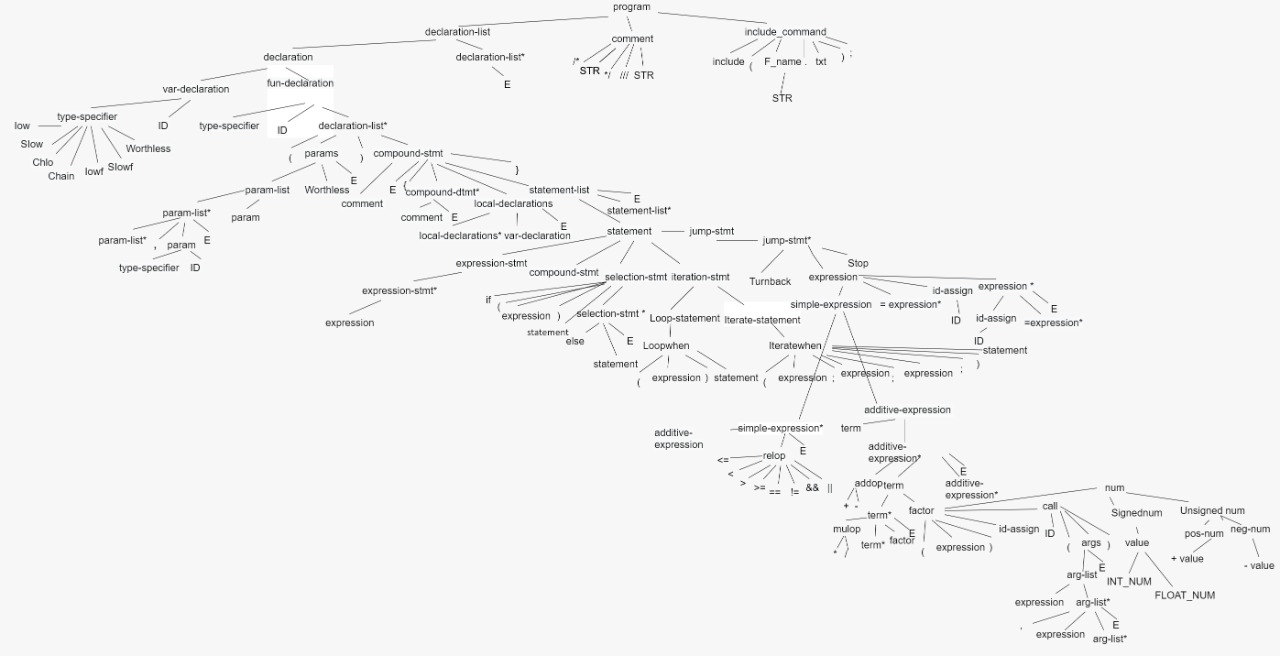
****

Total DFA

****

**Parse tree and Abstract syntax tree**

**Parse tree**

****