

ಇಂಡಿಯನ್ ಇನ್ಸ್ಟಿಟ್ಯೂಟ್ ಆಫ್ ಇನ್ಫರ್ಮೇಷನ್ ಟೆಕ್ನಾಲಜಿ,
ಡಿಸ್ಟೆನ್ ಅಂಡ್ ಮಾನ್ಯುಫ್ಯಾಕ್ಚರಿಂಗ್, ಕರ್ನಾಟಕ



भारतीय सूचना प्रौद्योगिकी, अभिकल्पना
एवं विनिर्माण संस्थान, कर्नाटक

Department of Computer Science and Engineering,

Dec 2024- April 2025

Compiler Design Practice (S6-B.Tech) -Assignment 1

Policies for Submission and Evaluation

You must submit your assignment in the moodle (Eduserver) course page, on or before the submission deadline. Also, ensure that your programs in the assignment must compile and execute without. During evaluation your uploaded programs will be checked. Failure to execute programs in the assignment without compilation errors may lead to zero marks for that program.

Your submission will also be tested for plagiarism, by automated tools. In case your code fails to pass the test, you will be straightaway awarded zero marks for this assignment and considered by the examiner for awarding “F” grade in the course. Detection of ANY malpractice regarding the lab course will also lead to awarding an “F” grade.

Last date for submitting : 27/12/2024 12:00 Hrs

Naming Conventions for Submission

Submit a single ZIP (.zip) file (do not submit in any other archived formats like .rar or .tar.gz).

The name of this file must be ASSG<NUMBER>_<ROLLNO>_<FIRSTNAME>.zip. (eg:

ASSG1_118cs0006_LAXMAN.zip). DO NOT add any other files (like temporary files,

inputfiles, etc.) except your source code, into the zip archive. The source codes must be named

as ASSG<NUMBER>_<ROLLNO>_<FIRSTNAME>_<PROGRAM-NO>.<extension>. (For

example: ASSG1_118cs0006_LAXMAN_1.c). If there are multiple parts for a particular

question, then name the source files for each part separately as in

ASSG1_118cs0006_LAXMAN_1b.c.

If you do not conform to the above naming conventions, your submission might not be recognized by some automated tools, and hence will lead to a score of 0 for the submission. So,

make sure that you follow the naming conventions.

Problem Set:

1. Write a program to design Lexical Analyzer in C/C++ Language (to recognize any five keywords, identifiers, numbers, operators and punctuations).

Sample Input:

```
Enter number of lines: 3
int int nti iiit;
float hi 23.5
1a 2b e3;
```

Sample output:

```
Keywords: 3 (int int float)
Operators: 0
Constants: 1 (23.5)
Punctuations: 2 (; ;)
Identifiers: 3 (nti iiit hi)
Tokens: 9
```

Problem 2: Design a deterministic finite automata (DFA) for accepting the language

$$L = \{a^n b^m \mid n \bmod 2 = 0, m \geq 1\}$$

Regular expression for above language L is,

$$L = (aa)^* . b^+$$

Examples:

Input: aabbbb

Output: ACCEPTED

// n = 2 (even) m=3 (>=1)

Input: aaabbbb

Output: NOT ACCEPTED

// n = 3 (odd), m = 3

Input: aaaa

Output: NOT ACCEPTED

// n = 4, m = 0(must be >=1)

Solution:

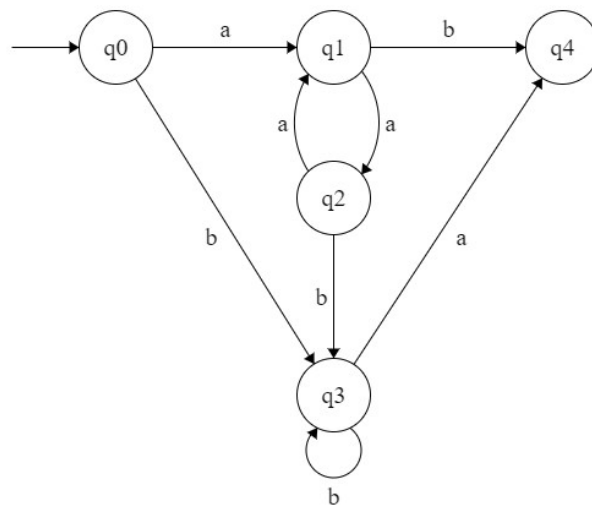
- Construct FA for $(aa)^*$ means having even number of a's.
- Construct FA for b^+ means having any number of b's greater than one.
- Concatenate the two FA and make single DFA.

Any other combination result is the rejection of the input string.

Description:

Given DFA has following states. State 3 leads to the acceptance of the string, whereas states 0, 1, 2 and 4 leads to the rejection of the string.

DFA State Transition Diagram:



Required DFA

Problem 3

Design deterministic finite automata (DFA) with $\Sigma = \{0, 1\}$ that accepts the languages ending with “01” over the characters $\{0, 1\}$.

Output

The output is as follows –

Run 1:
 enter the string to be checked: 10101001
 String is accepted.
 Run 2:
 enter the string to be checked: 10000010
 String is not accepted.

Solution

The strings that are generated for a given language are as follows –

$L = \{01, 001, 101, 110001, 1001, \dots\}$

The minimum length of the string is 2, the number of states that the DFA consists of for the given language is: $2+1 = 3$ states.

