# Computer Science & Engineering Department
## I.I.T Kharagpur

Compilers

*3rd year CSE: 5th Semester (Class Test 1)*

Time limit: 1 hour (Time: 14.15-15.15) Date: 1st Sep., 2022

Max Marks: 15

**Answer all the questions**

**1.** Implement a DFA (Recognizer) for valid Identifiers and constants (int_const and char_const) as given below:

*letter : A | B | . . . | Z | a | b | . . . | z*

*digit : 0 | 1 | 2 | . . . | 9*

*nonzero_digit : 1 | 2 | . . . | 9*

*quote : "*

*Identifiers : letter (letter | digit)**

*int_const : nonzero_digit (digit)**

*char_const : quote (letter)+ quote*

Note: Only one recognizer is required to be implemented for all three Identifiers and constants (int_const and char_const). Knee closure (*), positive closure (+), and or function (|) is as per their usual meaning.

**(5)**

**2.** (a) Consider the grammar G1

G1: <{if, then, a, else, b}, {S, S',E}, {S}, P1>

P1:

**S→ if E then S S' |a**
**S' → else S | ϵ**
**E → b**

In the light of the following string, justify if the grammar G is ambiguous or not.

**if b then if b then a else a**

(b) Explain how can you differentiate between an identifier and a keyword, with the help of a single state transition machine?

**(2.5+2.5)**

3. Parse the string **b++d** by executing the non-deterministic recursive descent parser with the grammar G2 as specified below. Apply the productions strictly following the (increasing) sequence, indicated by the rule#. Clearly show the functions invoked by the parser and all the backtracking steps, if any. Finally, justify, if the parser accepts or rejects this string.

G2: <{+,b,d}, {A,B,C}, {A}, P2>
P2:

| Rule# | Production |
|---|---|
| 1. | A → B * C |
| 2. | A → B + C |
| 3. | B → bbC |
| 4. | B → b |
| 5. | C → +dC |
| 6. | C → +d |

(5)

**********END**********