COL226: Programming Languages

Mon 14 Feb 2022 **MinorQ4** 5+15 (+5 for PwD) minutes Max marks 10

Instructions:

1. Download the paper and write your name and entry number in the designated space on top and *do not forget to sign the honour statement below.*
2. Answer the question(s). *Answers will be judged for correctness, efficiency and elegance.*
4. If there are minor mistakes in the question, correct them explicitly and answer the question accordingly. If the question is totally wrong, give adequate reasons why it is wrong with detailed counter-examples, if necessary.
4. Scan the paper with your completed answer.
5. Upload it on Gradescope 2102-COL226 page within the given time. *Make sure the first page with your name, entry no and signature is also the first page of your uploaded file*
6. Late submissions (within 2 minutes of submission deadline) on the portal will attract a penalty of 10% of the total marks allotted to the paper for each minute of delay and 20% for each minute of delay thereafter.
7. Email submissions after the closing of the portal will not be evaluated (You get a 0).
8. Uploads without the first page details (including signature) may be awarded 0 marks.

I abide by the Honour code that I have signed on my admission to IIT Delhi. I have neither given any help to anybody nor received any help from anybody nor from any site or other sources in solving the question(s) in this paper.

Signature: **Date:**

[4+3+3=10 marks]

It is well-known that given the arity of each operator in the programming language, an expression with infix, post-fix or mix-fix operators may be transformed into a semantically equivalent bracket-free expression in which all the operators are used in prefix form. That is,

- bracketing symbols are not required,
- associativity and precedence rules are not required to capture the order of operations,

Hence an expression $\sim$ `(10 - 3) / 2` where

- $\sim$ is the unary negation operator on integers, and has the highest precedence,
- `/` is the left-associative binary integer division operator having a higher precedenc than `-` but lower than $\sim$, and
- `-` is the left-associative binary integer subtraction operator having the lowest precedence.

may be represented in bracket-free prefix form as the sequence of tokens `/` $\sim$ `- 10 3 2` which may then be evaluated by a simple recursive procedure using the order and arity of the individual operators in the sequence of tokens.

For integer expressions containing the above three operators,

1. Design an unambiguous grammar G_1 (suitable for top-down parsing) where `-` and `/` are used in infix form and $\sim$ is used in prefix form.
2. Design a bracket-free grammar G_2 for integer expressions in which all operators are used in prefix form.
3. G_2 is a bracket-free context-free grammar. But since there are no brackets, is it purely right-linear, purely left-linear, linear (containing a mix of only right-linear and left-linear productions), or none of the above? Justify your answer.