Homework 4: Context-free languages

Theory of Computing (CSE 30151), Spring 2025

Due: 2025-02-21 11:59pm

Instructions

- Create a PDF file (or files) containing your solutions. You can write your solutions by hand, but please scan them into a PDF.

- Please name your PDF file(s) as follows to ensure that the graders give you credit for all of your work:

  - If you're making a complete submission, name it *netid*-hw4.pdf, where *netid* is replaced with your NetID.

  - If you're submitting some problems now and want to submit other problems later, name it *netid*-hw4-part123.pdf, where 123 is replaced with the problem number(s) you are submitting at this time.

- Submit your PDF file(s) in Canvas.

Problems (10 points each)

1. **Arithmetic expressions.** Consider the grammar G_4 (page 105) for arithmetic expressions, with start symbol E:

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid \texttt{a} \mid \texttt{b} \mid \texttt{c}$$

   (a) [cf. Exercise 2.1] Give derivations for the following strings. You may write them either as a sequence of rewrites ($E \Rightarrow \cdots$) or as a tree.

      i. `a + b + c`
      ii. `a * b + c`
      iii. `a * (b + c)`

(b) Modify G_4 to allow an exponentiation operator $\uparrow$.

- It should have *higher precedence* than multiplication; that is, in the derivation of the string `a * b ↑ c`, there should be a nonterminal that rewrites to `b ↑ c`, and there should not be a nonterminal that rewrites to `a * b`.

- It should be (unlike `*` and `+`) *right-associative*; that is, in the derivation of the string `a ↑ b ↑ c`, there should be a nonterminal that rewrites to `b ↑ c`, and there should not be a nonterminal that rewrites to `a ↑ b`.

2. Write both a PDA *and* a CFG for the language (page 80):

$$C = \{w \in \{0, 1\}^* \mid w \text{ has an equal number of 0s and 1s}\}.$$

Please include a brief explanation of why they work. (If you design a PDA and then convert it to a CFG, your explanation for the CFG can simply be, "I converted my PDA to a CFG," and similarly if you convert a CFG to a PDA.)

3. [Exercise 2.6b] Write both a PDA *and* a CFG for the language

$$L_3 = \overline{\{0^n 1^n \mid n \geq 0\}}.$$

For example, `000111` $\notin L_3$. Please include a brief explanation of why they work. (If you design a PDA and then convert it to a CFG, your explanation for the CFG can simply be, "I converted my PDA to a CFG," and similarly if you convert a CFG to a PDA.)

Hint: First prove that this is equal to $\{0^m 1^n \mid m \neq n\} \cup \overline{0^* 1^*}$.