

CMPU 240 · Spring 2026

Assignment 6

Submissions due: 31 March, 1:30 p.m.

Corrections due: 2 April, 1:30 p.m.

Exercise 1

Consider the language $L_1 = \{w \in \{0, 1\}^* \mid w \text{ is not a palindrome}\}$.

- a. Design a pushdown automaton for L_1 . (Design it directly; don't convert from a grammar.)

Your solution for part (a) should be submitted as a csv file saved from tock, using the provided Colab notebook.

- b. Describe how your PDA design works. What is each state for? How are you using the stack?

Note that this is harder than it was to design a PDA to accept palindromes! I think you need at least five states to do it.

Exercise 2

Recall the context-free grammar for arithmetic expressions from Example 2.4 in the textbook. For convenience, let's rename its variables with single letters as follows:

$$E \rightarrow E+T \mid T$$

$$T \rightarrow T \times F \mid F$$

$$F \rightarrow (E) \mid a$$

- a. Convert this context-free grammar to an equivalent PDA, using the method shown in class and outlined in the proof of Lemma 2.21 in the textbook.

You can use the abbreviation to push multiple symbols onto the stack in a single transition. Your answer can be given by drawing a state-transition diagram or writing a formal tuple.

- b. Using either the \vdash operation presented in class or a table that clearly shows the state, the remaining input, and the stack contents (with the top at the left), illustrate the operation of the PDA you constructed in part (a) in accepting the string $a \times (a+a)$.

Exercise 3

Imagine adding more stacks to a PDA – would this change the power of the model? At first, it doesn't seem like it should because, much like we made DFAs whose states represented multiple states that an NFA could be, we can introduce new stack symbols to represent what's on top of both stacks.

For example, if the stack alphabet was $\{0, 1, \$\}$, you would introduce new symbols like

A for $(0, 0)$, a 0 on each stack,

B for $(1, 1)$, a 1 on each stack,

C for $(0, 1)$, a 0 on the first stack and a 1 on the second stack,

...

Unfortunately, this (clever, handsome) construction does *not* let us make a PDA that simulates a PDA with two stacks. What (informally) is wrong here?