Oblig 2

IN2080

Deadline: April 10, 2024

Hand-in and deadline

Hand in a single PDF file containing your answers to all 3 Problems Devilry. The deadline is **April 10, at 23:59**.

We recommend LaTeX, but all major text editors allows exporting to PDF. You can get help with LaTeX at the group sessions. You can also download the LaTeX source (`.tex`) for this assignment at the assignments page.

Problem 1: Context-free languages

Consider the languages

$$L_1 = \{\mathsf{a}\mathsf{b}^n\mathsf{c}^n \mid n \geq 0\}$$

and

$$L_2 = \{\mathsf{a}^i\mathsf{b}^j\mathsf{c}^k \mid i, j, k \geq 0 \text{ and } i + j = k\}.$$

Problem 1a

Construct a CFG which generates L_1.

Problem 1b

Give the state diagram for a PDA which recognizes L_1.

Problem 1c

Show that L_2 is context-free.

Problem 2: Pumping lemmas

Problem 2a

Consider the language L_2 from Problem 1. Prove, using the pumping lemma for regular languages, that L_2 is not regular.

Problem 2b

Let $L_3 = \{\mathtt{a}^i\mathtt{b}^j\mathtt{c}^k \mid i, j, k \geq 0 \text{ and } i \times j = k\}$.

Prove, using the pumping lemma for context-free languages, that L_3 is not context-free.

Problem 3: Turing machines

Problem 3a

Draw a state diagram for a Turing machine deciding the language from Exercise 3.8a: $\{w \in \{0,1\}^* \mid w \text{ contains an equal number of } \mathtt{0}\text{s and } \mathtt{1}\text{s}\}$

Problem 3b

Give an implementation-level description of a Turing machine deciding the language from exercise 3.8b: $\{w \in \{0,1\}^* \mid w \text{ contains twice as many } \mathtt{0}\text{s as } \mathtt{1}\text{s}\}$.

An example of an "implementation-level description" is Sipser's Example 3.7, which decides the language $\{0^{2^n} \mid n \geq 0\}$:

> On input string w:
>
> 1. Sweep left to right across the tape, crossing off every other 0
> 2. If in stage 1 the tape contained a single 0, accept
> 3. If in stage 1 the tape contained an odd number of 0s, reject
> 4. Return the head to the start of the tape
> 5. Go to stage 1

If you would like to play around and test your Turing machines, a Turing machine simulator is available at `https://github.com/torenord/universaltm`. It runs Turing Machines from a text file in the format specified in the `README`.