You can answer in any order that you like, but indicate the question numbers in your answers. The questions are roughly sorted by difficulty except for ★ questions, which are harder.

1 Pumping Lemmas

Let $\mathbb{N} := \{0, 1, 2, \ldots\}$. For $(a, b, c) \in \mathbb{N}^3$ define $L_{a,b,c} := \{0^{an}1^{bn}2^{cn} : n \geq 0\}$ over the alphabet $\Sigma := \{0, 1, 2\}$.

1.1. Find all triples $(a, b, c) \in \mathbb{N}^3$ for which $L_{a,b,c}$ is regular.

- 1.1.1) For $L_{a,b,c}$ that are regular, justify your answer by giving DFAs, NFAs or regular expressions.
- 1.1.2) For $L_{a,b,c}$ that are not regular, justify your answer using the pumping lemma.

1.2. Find all triples $(a, b, c) \in \mathbb{N}^3$ for which $L_{a,b,c}$ is context-free.

- 1.2.1) For $L_{a,b,c}$ that are context-free, justify your answer by giving CFGs or PDAs.
- 1.2.2) For $L_{a,b,c}$ that are not context-free, justify your answer via the pumping lemma.

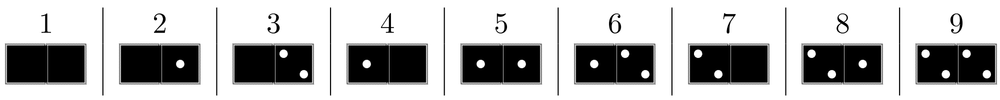
(*Suggestion: Assume $a \leq b \leq c$ and briefly argue that your answers also go through for the other cases.*)

2 A Game of Dominoes

The game of dominoes is played with rectangular tiles, where each tile t has a left number $l(t)$ and a right number $r(t)$. In the original game, each left or right number is an integer between 0 and 6. The value 0 is sometimes called a *joker*.

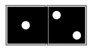
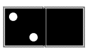
2.1 Binary dominoes with a joker

For the sake of simplicity, we only consider dominoes with left or right numbers 0 (joker), 1, and 2. Thus, there are 9 such dominoes:



A sequence of dominoes $d_1 \cdots d_n$ is called *valid* if for all $1 \leq i < n$ either the right number of d_i is equal to left number of d_{i+1}, or one of them is a joker, i.e., $r(d_i) = l(d_{i+1})$ or $0 \in \{r(d_i), l(d_{i+1})\}$).

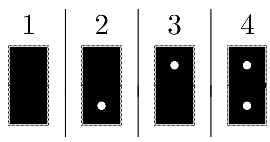
2.1. Show that the language of all valid domino sequences is regular.

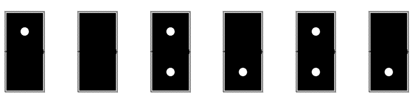
The *value(s)* of a domino is either the sum of its left and right numbers if these are both non-zero, or all values that can be obtained by substituting a 1 or a 2 for the zeros. For instance,  has value 3, and  has values 3 and 4.

2.2. Show that the language of sequences with an even value is regular.

2.2 Binary dominoes

We now consider vertical dominoes with values 0 and 1 only. (The value 0 is no longer called a joker and is treated as a regular integer.) There are now only 4 dominoes (see on the right)





When reading a sequence of dominoes from left to right we can obtain two binary numbers, one for the top row and one for the bottom row, where the most significant bit is read first. For instance the sequence of dominoes on the left forms $42 = 2^5 + 2^3 + 2^1$ on the top row and $15 = 2^3 + 2^2 + 2^1 + 2^0$ on the bottom row.

★ 2.3 Show that the set of sequences for which the number for the top row is three times the number for the bottom row is regular.

3 The Dichotomy Property

Let L be a regular language accepted by a deterministic finite automaton with states Q.

3.1. Show that L is non-empty if and only if $\exists w \in L \ : \ |w| \leq |Q|$.

3.2. Show that L is infinite if and only if $\exists w \in L \ : \ |Q| < |w| \leq 2|Q|$. (*Hint: Consider the proof of the pumping lemma with a single cycle.*)

4 Intersection of Regular and Context-Free Languages

4.1. Let L be a regular language and L' be a context-free language, both over a common alphabet Σ. Is $L \cap L'$ context-free? (*Hint: consider both languages as languages accepted by automata.*)

5 Boolean Expressions

We define the language of *well-formed boolean expression* via grammar $G := (\{Be, St\}, \Sigma, R, Be)$, where $\Sigma = \{\wedge, \neg, \top, \bot, (,)\}$ and the production rules R are $Be \to St \wedge St \mid \neg St \mid St$ and $St \to \top \mid \bot \mid (Be)$. This grammar can be shown to be unambiguous, and hence there is a unique parse-tree for each generated expression. Thus a well-formed boolean expression can be evaluated to a unique value, which is either true ($\top$) or false ($\bot$). (As usual $\wedge$ denotes logical AND, and $\neg$ negation.)

5.1. Give a context-free grammar generating all well-formed boolean expressions that evaluate to *true*.

5.2. Give a pushdown automaton accepting all well-formed boolean expressions that evaluate to *false*.

★ 5.3 Show that this grammar is unambiguous. (*Hint: Show that words produced by Be (and St) are well-parenthesized. Then look at the first derivation of the smallest word with 2 different derivations.*)

6 Finite Context-Free Languages

6.1. Show that if every subset of a language L is context-free then L is finite. (*Hint: Recursively build a subset of words whose lengths grow fast. Apply the pumping lemma.*)

7 Universal Automata

A DFA D an be unambiguously encoded as a string. (For instance, a DFA of the form $D := (Q, \Sigma, \delta, q_0, F)$ with $\Sigma := \{0, 1\}$, $F \subseteq Q \subset \{0, 1\}^*$ and $q_0 = 0$ can be encoded over the alphabet $\{0, 1, \#, (,)\}$ as a list of strings $(q\#b\#q')$ where $\delta(q, b) = q'$. The specific details of an encoding are not important here; only that one exist.)

7.1. Does there exist a universal DFA? More precisely, fix an arbitrary encoding f for DFAs and consider the language $L_U := \{f(D)\#w : w \in L(D)\}$. Is this regular?

8 Unary Languages

8.1. Find all unary regular languages. To this end, show that any regular $L \subseteq 0^*$ can be written as the union of a finite language and languages $L_{b_i,c} := \{0^{b_i + cn} : n \geq 0\}$ for $b_1, \ldots, b_k, c \in \mathbb{N}$. Conversely, show any such language is regular.

★★ 8.2 Find all unary context-free languages. For this part show that a unary language is context-free iff it is regular. (*Suggestion: Apply pumping to find the form of the long words. Then write them as a union of regular languages.*)