Spring 2020 CIS 262

Automata, Computability and Complexity
Jean Gallier

Homework 9

April 16, 2020; Due April 29, 2020, beginning of class

Problem B1 (40 pts). Give a context-free grammar for the language over the alphabet $\{a, b, c\}$ given by

$$L = \{xcy \mid x \neq y,\ x, y \in \{a, b\}^*\}.$$

Hint. At first glance, this seems impossible. Think nondeterministically. You need to figure out how to express that $x \neq y$ in such a way that you can write grammar rules that enforce this condition. Obviously, this is the case if $|x| < |y|$ or $|y| < |x|$. Another possibility is that x and y differ by some symbol in the same position (scanning from left ro right).

If you do it "right," you choice of productions should yield a justification of the correctness of your grammar.

Problem B2 (10 pts). Prove that the extended pairing function $\langle x_1, \ldots, x_n\rangle_n$ defined in the notes (see Section 2.1 of the notes, page 44) satisfies the equation

$$\langle x_1, \ldots, x_n, x_{n+1}\rangle_{n+1} = \langle x_1, \langle x_2, \ldots, x_{n+1}\rangle_n\rangle.$$

Compute $\langle 2, 5, 7, 17\rangle_4$ (this integer has 10 digits).

Problem B3 (30 pts). Prove that the function, $f\colon \Sigma^* \to \Sigma^*$, given by

$$f(w) = www$$

is RAM computable by constructing a RAM program ($\Sigma = \{a, b\}$).

Problem B4 (30 pts). Give context-free grammars for the following languages:

(a) $L_5 = \{wcw^R \mid w \in \{a, b\}^*\}$ (w^R denotes the reversal of w)

(b) $L_6 = \{a^m b^n \mid 1 \leq m \leq n \leq 2m\}$

(c) $L_8 = \{xcy \mid |x| = 2|y|,\ x, y \in \{a, b\}^*\}$

In each case, give a (very) brief justification of the fact that your grammar generates the desired language.

Problem B5 (60 pts). Given a context-free language L and a regular language R, prove that $L \cap R$ is context-free.

Do not use PDA's to solve this problem!

Use the following method. Without loss of generality, assume that $L = L(G)$, where $G = (V, \Sigma, P, S)$ is in Chomsky normal form, and let $R = L(D)$, for some DFA $D = (Q, \Sigma, \delta, q_0, F)$. Use a kind of cross-product construction as described below. Construct a CFG G_2 whose set of nonterminals is $Q \times N \times Q \cup \{S_0\}$, where S_0 is a new nonterminal, and whose productions are of the form:

$$S_0 \to (q_0, S, f),$$

for every $f \in F$;

$$(p, A, \delta(p, a)) \to a \quad \text{iff} \quad (A \to a) \in P,$$

for all $a \in \Sigma$, all $A \in N$, and all $p \in Q$;

$$(p, A, s) \to (p, B, q)(q, C, s) \quad \text{iff} \quad (A \to BC) \in P,$$

for all $p, q, s \in Q$ and all $A, B, C \in N$;

$$S_0 \to \epsilon \quad \text{iff} \quad (S \to \epsilon) \in P \text{ and } q_0 \in F.$$

Prove that for all $p, q \in Q$, all $A \in N$, all $w \in \Sigma^+$, and all $n \geq 1$,

$$(p, A, q) \overset{n}{\underset{lm}{\Longrightarrow}}_{G_2} w \quad \text{iff} \quad A \overset{n}{\underset{lm}{\Longrightarrow}}_{G} w \quad \text{and} \quad \delta^*(p, w) = q.$$

Conclude that $L(G_2) = L \cap R$.

Problem B6 (50 pts). Given an undirected graph $G = (V, E)$ and a set $C = \{c_1, \ldots, c_p\}$ of p colors, a *coloring* of G is an assignment of a color from C to each node in V such that no two adjacent nodes share the same color, or more precisely such that for evey edge $\{u, v\} \in E$, the nodes u and v are assigned different colors. A k-coloring of a graph G is a coloring using at most k-distinct colors. For example, the graph shown in Figure 1 has a 3-coloring (using green, blue, red).

The **graph coloring problem** is to decide whether a graph G is k-colorable for a given integer $k \geq 1$.

(1) Give a polynomial reduction from the **graph 3-coloring problem** to the **3-satisfiability problem for propositions in CNF**.

If $|V| = n$, create $n \times 3$ propositional variables x_{ij} with the intended meaning that x_{ij} is true iff node v_i is colored with color j. You need to write sets of clauses to assert the following facts:

1. Every node is colored.

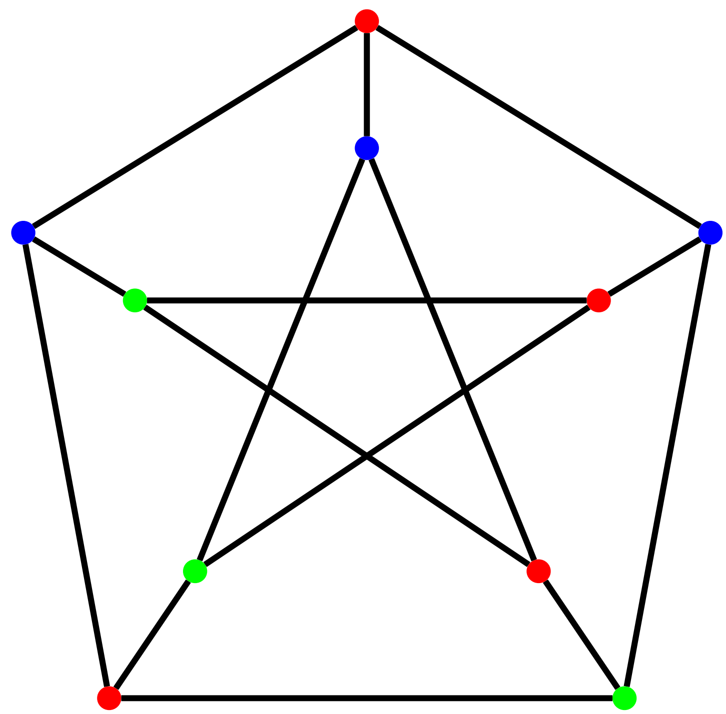

Figure 1: Petersen graph.

2. No two distinct colors are assigned to the same node.

3. For every edge $\{v_i, v_j\}$, nodes v_i and v_j cannot be assigned the same color.

Beware that it is possible to assert that every node is assigned one and only one color using a proposition in disjunctive normal form, but this is not a correct answer; we want a proposition in conjunctive normal form.

(2) Prove that 2-coloring can be solved deterministically in polynomial time.

Remark: It is known that a graph has a 2-coloring iff it is bipartite, but **do not** use this fact to solve B3(2). *Only use material covered in the notes for CIS262.*

The problem of 3-coloring is actually $\mathcal{NP}$-complete, but this is a bit tricky to prove.

Problem B7 (60 pts). Let A be any $p \times q$ matrix with integer coefficients and let $b \in \mathbb{Z}^p$ be any vector with integer coefficients. The 0-1 *integer programming problem* is to find whether a system of p linear equations in q variables

$$\begin{aligned} a_{11}x_1 + \cdots + a_{1q}x_q &= b_1 \\ \vdots \qquad\quad & \quad \vdots \\ a_{i1}x_1 + \cdots + a_{iq}x_q &= b_i \\ \vdots \qquad\quad & \quad \vdots \\ a_{p1}x_1 + \cdots + a_{pq}x_q &= b_p \end{aligned}$$

with $a_{ij}, b_i \in \mathbb{Z}$ has any solution $x \in \{0,1\}^q$, that is, with $x_i \in \{0,1\}$. In matrix form, if we let

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1q} \\ \vdots & \ddots & \vdots \\ a_{p1} & \cdots & a_{pq} \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_p \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ \vdots \\ x_q \end{pmatrix},$$

then we write the above system as

$$Ax = b.$$

(i) Prove that the 0-1 integer programming problem is in $\mathcal{NP}$.

(ii) Prove that the restricted 0-1 integer programming problem in which the coefficients of A are 0 or 1 and all entries in b are equal to 1 is $\mathcal{NP}$-complete by providing a polynomial-time reduction from the bounded-tiling problem. **Do not try to reduce any other problem to the 0-1 integer programming problem**.

Hint. Given a tiling problem, $((\mathcal{T}, V, H), \widehat{s}, \sigma_0)$, create a 0-1-valued variable, x_{mnt}, such that $x_{mnt} = 1$ iff tile t occurs in position (m,n) in some tiling. Write equations or inequalities expressing that a tiling exists and then use "slack variables" to convert inequalities to equations. For example, to express the fact that every position is tiled by a single tile, use the equation

$$\sum_{t \in \mathcal{T}} x_{mnt} = 1,$$

for all m, n with $1 \leq m \leq 2s$ and $1 \leq n \leq s$. Also, if you have an inequality such as

$$2x_1 + 3x_2 - x_3 \leq 5 \tag{$*$}$$

with $x_1, x_2, x_3 \in \mathbb{Z}$, then using a new variable y_1 *taking its values in* $\mathbb{N}$, that is, *nonnegative values*, we obtain the equation

$$2x_1 + 3x_2 - x_3 + y_1 = 5, \tag{$**$}$$

and the inequality $(*)$ has solutions with $x_1, x_2, x_3 \in \mathbb{Z}$ iff the equation $(**)$ has a solution with $x_1, x_2, x_3 \in \mathbb{Z}$ and $y_1 \in \mathbb{N}$. The variable y_1 is called a *slack variable* (this terminology comes from optimization theory, more specifically, linear programming). For the 0-1-integer programming problem, all variables, including the slack variables, take values in $\{0, 1\}$.

Conclude that the 0-1 integer programming problem is $\mathcal{NP}$-complete.

TOTAL: 280 points.