

# Automata, Computability and Complexity

Jean Gallier

## Solutions of the Practice Final Exam

April 26, 2020

**Problem 1 (10 pts).** Let  $\Sigma$  be an alphabet.

(1) What is an ambiguous context-free grammar? What is an inherently ambiguous context-free language?

(2) Is the following context-free grammar ambiguous, and if so demonstrate why?

$$E \longrightarrow E + E$$

$$E \longrightarrow E * E$$

$$E \longrightarrow (E)$$

$$E \longrightarrow a.$$

*Solution.* (1) See Definition 7.5 of the notes.

(2) Yes, this grammar is ambiguous. For example, the string  $a + a * a$  has two distinct leftmost derivations

$$\begin{aligned} \mathbf{E} &\Longrightarrow \mathbf{E} * E \Longrightarrow \mathbf{E} + E * E \\ &\Longrightarrow a + \mathbf{E} * E \Longrightarrow a + a * \mathbf{E} \Longrightarrow a + a * a, \end{aligned}$$

and

$$\begin{aligned} \mathbf{E} &\Longrightarrow \mathbf{E} + E \Longrightarrow a + \mathbf{E} \\ &\Longrightarrow a + \mathbf{E} * E \Longrightarrow a + a * \mathbf{E} \Longrightarrow a + a * a. \end{aligned}$$

**Problem 2 (5pts).** Given any trim DFA  $D = (Q, \Sigma, \delta, q_0, F)$  accepting a language  $L = L(D)$ , if  $D$  is a minimal DFA, then prove that its Myhill-Nerode equivalence relation  $\simeq_D$  is equal to  $\rho_L$ .

*Solution.* We proved in Proposition 6.6 of the notes that for any trim DFA  $D$  accepting a language  $L$ , we have

$$\simeq_D \subseteq \rho_L.$$

We also proved that the number of equivalence classes of  $\rho_L$  is the size of the minimal DFA's for  $L$  (again, see Proposition 6.6). Therefore, if  $D$  is a minimal DFA, then  $\simeq_D$  and  $\rho_L$  have the same number of classes, which implies that  $\simeq_D = \rho_L$ .

**Problem 3 (10 pts).**

Consider the following DFA  $D_0$  with start state  $A$  and final state  $D$  given by the following transition table:

	$a$	$b$
$A$	$B$	$A$
$B$	$C$	$A$
$C$	$D$	$A$
$D$	$D$	$A$

Reverse all the arrows of  $D_0$ , obtaining the following NFA  $N$  (without  $\epsilon$ -transitions) with start state  $D$  and final state  $A$  given by the following table:

	$a$	$b$
$A$	$\emptyset$	$\{A, B, C, D\}$
$B$	$\{A\}$	$\emptyset$
$C$	$\{B\}$	$\emptyset$
$D$	$\{C, D\}$	$\emptyset$

This NFA accepts the language  $\{aaa\}\{a, b\}^*$ .

- (1) Use the subset construction to convert  $N$  to a DFA  $D$  (with 5 states).
- (2) Prove that  $D$  is a minimal DFA.

*Solution.* (1) Applying the subset construction, we obtain the following DFA with start state 0 and final state 4:

		$a$	$b$
0	$\{D\}$	1	2
1	$\{C, D\}$	3	2
2	$\emptyset$	2	2
3	$\{B, C, D\}$	4	2
4	$\{A, B, C, D\}$	4	4

(2) Let's apply the method for propagating inequivalence described in Section 6.3 of the notes. Since 4 is the only final state, the initial table is the following (where  $\times$  means inequivalent, and  $\square$  means don't know yet):

1	$\square$			
2	$\square$	$\square$		
3	$\square$	$\square$	$\square$	
4	$\times$	$\times$	$\times$	$\times$
	0	1	2	3

Let us proceed from the bottom up and from right to left (as opposed to the top down). At the end of the first round, we get

1	$\times$			
2	$\times$	$\times$		
3	$\times$	$\times$	$\times$	
4	$\times$	$\times$	$\times$	$\times$
	0	1	2	3

Nothing changes during the second round, so we conclude that there are no pairs of equivalent states, which means that the DFA is minimal.

**Problem 4 (10 pts).** Given any context-free grammar  $G = (V, \Sigma, P, S')$ , with special starting production  $S' \rightarrow S$  where  $S'$  only appears in this production, the set of characteristic strings  $C_G$  is defined by

$$C_G = \{ \alpha\beta \in V^* \mid S' \xRightarrow{rm}^* \alpha B v \xRightarrow{rm} \alpha\beta v, \\ \alpha, \beta \in V^*, v \in \Sigma^*, B \rightarrow \beta \in P \}.$$

Consider the grammar  $G$  with nonterminal set  $\{S, A, C\}$  and terminal set  $\{a, b, c\}$  given by the following productions:

$$\begin{aligned} S' &\rightarrow S \\ S &\rightarrow AC \\ A &\rightarrow aAb \\ A &\rightarrow ab \\ C &\rightarrow c. \end{aligned}$$

Describe all *rightmost* derivations and the set  $C_G$ .

*Solution.* Rightmost derivations are of the form

$$S' \xRightarrow{rm} S$$

or

$$\begin{aligned} S' &\xRightarrow{rm} S \\ S &\xRightarrow{rm} AC \end{aligned}$$

or

$$\begin{aligned} S' &\xRightarrow{rm} S \\ S &\xRightarrow{rm} AC \\ AC &\xRightarrow{rm} Ac \end{aligned}$$

or

$$\begin{aligned} S' &\xRightarrow{rm} S \\ S &\xRightarrow{rm} AC \\ AC &\xRightarrow{rm} Ac \\ Ac &\xRightarrow{rm}^* a^n Ab^n c \\ a^n Ab^n c &\xRightarrow{rm} a^{n+1} b^{n+1} c = a^{n+1} b b^n c \end{aligned}$$

or

$$\begin{aligned} S' &\xRightarrow{rm} S \\ S &\xRightarrow{rm} AC \\ AC &\xRightarrow{rm} Ac \\ Ac &\xRightarrow{rm}^* a^n Ab^n c \\ a^n Ab^n c &\xRightarrow{rm} a^{n+1} Ab^{n+1} c = a^{n+1} A b b^n c \end{aligned}$$

with  $n \geq 0$ . It follows that

$$C_G = \{S, AC, Ac, a^n b, a^n Ab \mid n \geq 1\}.$$

**Problem 5 (20 pts).**

(i) Give a context-free grammar for the language

$$L_2 = \{a^m b^n c^p \mid n \neq p, m, n, p \geq 1\}.$$

(ii) Prove that the language  $L_2$  is not regular.

*Solution.* Let  $G_2$  be the grammar whose productions are

$$\begin{aligned} S &\longrightarrow ABY \mid AYC \\ Y &\longrightarrow bYc \mid bc \\ A &\longrightarrow aA \mid a \\ B &\longrightarrow bB \mid b \\ C &\longrightarrow cC \mid c. \end{aligned}$$

It is easy to check (by induction on the length of derivations) that  $L(G_2) = L_2$ .

(ii) We proceed by contradiction using Myhill-Nerode. If  $L_2$  is regular, then there is a right-invariant equivalence relation  $\simeq$  of finite index such that  $L_2$  is the union of classes of  $\simeq$ . Consider the infinite sequence

$$ab, ab^2, \dots, ab^n, \dots$$

Since  $\simeq$  has a finite number of classes, there are two distinct strings  $ab^i$  and  $ab^j$  in the above sequence such that

$$ab^i \simeq ab^j$$

with  $1 \leq i < j$ . By right-invariance, by concatenating on the right with  $c^i$ , we obtain

$$ab^i c^i \simeq ab^j c^i$$

and since  $i < j$ , we have  $ab^j c^i \in L_2$  and  $ab^i c^i \notin L_2$ , a contradiction.

**Problem 6 (10 pts).**

(i) Give a context-free grammar for the language

$$L_2 = \{a^m b^n \mid n < 3m, m > 0, n \geq 0\}.$$

*Solution.*

$$\begin{aligned} S &\longrightarrow aSXXX \\ S &\longrightarrow aXX \\ X &\longrightarrow b \mid \epsilon \end{aligned}$$

By induction, every leftmost derivation is of the form

$$S \xrightarrow[lm]{m} a^m S X^{3m} \xrightarrow[lm]{} a^{m+1} X^{3m+2} \xrightarrow[lm]{*} a^{m+1} b^n,$$

with  $m \geq 0$  and  $n < 3(m+1)$ .

**Problem 7 (10 pts).**

Prove that if the language  $L_1 = \{a^n b^n c^n \mid n \geq 1\}$  is not context-free (which is indeed the case), then the language  $L_2 = \{w \mid w \in \{a, b, c\}^*, \#(a) = \#(b) = \#(c)\}$  is not context-free either.

*Solution.* We know from Homework 8 that the context-free languages are closed under intersection with the regular languages. Assume by contradiction that  $L_2$  is context-free. The language  $R = \{a\}^* \{b\}^* \{c\}^*$  is regular (for example, an NFA can be easily constructed), and

$$L_1 = L_2 \cap R.$$

Since  $L_2$  is context-free and  $R$  is regular, then  $L_1$  is context-free, a contradiction.

**Problem 8 (10 pts).** Give an algorithm deciding whether a context-free grammar generates the empty language.

*Answer.* Let  $G$  be a context-free grammar. To test whether  $L(G) = \emptyset$ , compute the set of nonterminals that derive some terminal string,  $T(G) = \{A \in N \mid \exists w \in \Sigma^*, A \xRightarrow{+} w\}$ , using the iterative method given in the notes (Section 7.5). Then, test whether  $S \in T(G)$  (where  $S$  is the start symbol of  $G$ ).

**Problem 9 (10 pts).**

Let  $f: \mathbb{N} \rightarrow \mathbb{N}$  be a total computable function. Prove that if  $f$  is a bijection, then its inverse  $f^{-1}$  is also (total) computable.

*Solution.* Since the subtraction operation on natural numbers (monus) is primitive recursive, and since  $f$  is computable, the functions  $g_1, g_2: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$  given by

$$g_1(x, y) = f(x) - y, \quad g_2(x, y) = y - f(x)$$

are computable. Then the function  $h: \mathbb{N} \rightarrow \mathbb{N}$  defined by minimization by

$$h(y) = \min x [\text{abs}(f(x), y) = 0] = \min x [\text{add}(f(x) - y, y - f(x)) = 0]$$

is partial computable. However, since  $f$  is bijective, for any  $y \in \mathbb{N}$ , there is a *unique*  $x \in \mathbb{N}$  such that  $f(x) = y$ , namely  $x = f^{-1}(y)$ , so in fact  $h = f^{-1}$ . This shows that  $f^{-1}$  is partial computable, but since it is a total function, it is computable.

**Problem 10 (15 pts).** Recall that the **Clique Problem** for undirected graphs is this: Given an undirected graph  $G = (V, E)$  and an integer  $K \geq 2$ , is there a set  $C$  of nodes with  $|C| \geq K$  such that for all  $v_i, v_j \in C$ , there is *some* edge  $\{v_i, v_j\} \in E$ ? Equivalently, does  $G$  contain a complete subgraph with at least  $K$  nodes?

Give a direct polynomial reduction from the **Clique Problem** for undirected graphs to the **Satisfiability Problem**.

Assuming that the graph  $G = (V, E)$  has  $n$  nodes and that the budget is an integer  $K$  such that  $2 \leq K \leq n$ , create  $nK$  boolean variables  $x_{ik}$  with intended meaning that  $x_{ik} = \mathbf{T}$  if node  $v_i$  is chosen as the  $k$ th element of a clique  $C$ , with  $1 \leq k \leq K$ , and write clauses asserting that  $K$  nodes are chosen to belong to a clique  $C$ .

*Solution.* We want to assert that there is an injection  $\kappa: \{1, \dots, K\} \rightarrow \{1, \dots, n\}$  such that for all  $h, k$  with  $1 \leq h < k \leq K$ , there is an edge between  $v_i$  and  $v_j$ , with  $\kappa(h) = i$  and  $\kappa(k) = j$ . Since  $\kappa(k) = i$  iff  $x_{ik} = \mathbf{T}$ , this is equivalent to saying that if  $x_{ih} = \mathbf{T}$  and  $x_{jk} = \mathbf{T}$ , then  $\{v_i, v_j\} \in E$ .

To assert that  $K$  choices of nodes are made, equivalently that  $\kappa(k)$  is defined for all  $k \in \{1, \dots, K\}$ , write the  $K$  clauses

$$(x_{1k} \vee x_{2k} \vee \dots \vee x_{nk}), \quad k = 1, \dots, K.$$

To assert that at most one node is chosen as the  $k$ th node in  $C$ , equivalently that  $\kappa$  is a functional relation, write the clauses

$$(\overline{x_{ik}} \vee \overline{x_{jk}}) \quad 1 \leq i < j \leq n, \quad k = 1, \dots, K.$$

To assert that no node is picked twice, equivalently that  $\kappa$  is injective, write the clauses

$$(\overline{x_{ih}} \vee \overline{x_{ik}}) \quad 1 \leq h < k \leq K, \quad i = 1, \dots, n.$$

To assert that any two distinct nodes in  $C$  are connected by an edge, we say that for all  $h, k$  with  $1 \leq h < k \leq k$ , if  $x_{ih} = \mathbf{T}$  and  $x_{jk} = \mathbf{T}$ , namely  $x_{ih} \wedge x_{jk} = \mathbf{T}$ , then  $\{v_i, v_j\} \in E$ . The contrapositive says that if  $\{v_i, v_j\} \notin E$ , then  $\overline{x_{ih} \wedge x_{jk}} = \mathbf{T}$ , or equivalently  $(\overline{x_{ih}} \vee \overline{x_{jk}}) = \mathbf{T}$ . Thus we have the clauses

$$(\overline{x_{ih}} \vee \overline{x_{jk}}) \quad \text{if } \{v_i, v_j\} \notin E, \quad 1 \leq h < k \leq K,$$

which assert that if there is no edge between  $v_i$  and  $v_j$ , then  $v_i$  and  $v_j$  should not be chosen to be in  $C$ . Let  $S$  be the above set of clauses,

If the graph  $G$  has a clique with at least  $K$  nodes, then it has a clique  $C = \{v_{i_1}, \dots, v_{i_K}\}$  with  $K$  nodes, and then it is clear that the clauses in  $S$  are satisfied by the truth assignment  $v$  such that

$$v(x_{jk}) = \begin{cases} \mathbf{T} & \text{if } j = i_k, 1 \leq k \leq K, 1 \leq j \leq n \\ \mathbf{F} & \text{otherwise.} \end{cases}$$

Conversely, if the clauses in  $S$  are satisfied by a truth assignment  $v$ , then we obtain the clique of size  $K$  given by  $C = \{v_{i_1}, \dots, v_{i_K}\}$  with

$$i_k = j \quad \text{iff} \quad v(x_{jk}) = \mathbf{T}.$$

Therefore,  $G$  has a clique of size at least  $K$  iff the set of clauses  $S$  is satisfiable.