Theoretical Computer Science (Bridging Course)

Dr. G. D. Tipaldi
F. Boniardi
Winter Semester 2014/2015

University of Freiburg
Department of Computer Science

Revision Sheet

Question 1 (Finite Automata, $8 + 6$ points)

(a) Give a regular expression for each of the following languages:

- (i) all strings over $\{0, 1\}$ that are at least three symbols long and have a 0 at their resp. 3rd positions
- (ii) all strings over $\{0, 1\}$ that have odd length, if starting with a 0, and even length otherwise
- (iii) all strings over $\{a, b\}$ that contain the substrings aa or bab
- (iv) all strings over $\{a, b\}$ that do not contain the substring ba

(b) Draw a DFA equivalent to each of the following regular expressions:

- (i) $a(a \cup b)^*b$
- (ii) $(ab)^*$

Question 2 (Regular languages, 14 points)
Let $\Sigma = \{a, b\}$. Use the pumping lemma to prove that:

$$L = \{a^n b^{2n} a^{3n} \mid n \geq 0\}$$

is not regular.

Any other proof techniques will **not** receive any points.

Question 3 (Context-free languages, 7+7 points)

(a) Give the state diagram of a PDA recognizing the language

$$A = \{a^i b^j \mid i > 0 \text{ and } j = i + 1\}.$$

(b) Let $G = \langle \{S, X, Y, Z\}, \{a, b, c, d\}, R, S\rangle$ be the CFG with rules:

$$\begin{aligned} S &\to XYZ \\ X &\to Xa \mid b \mid \varepsilon \\ Y &\to b \mid c \\ Z &\to cd \end{aligned}$$

Specify a CFG G_0 in Chomsky Normal Form such that $L(G_0) = L(G)$.

Question 4 (NP-completeness, $7 + 7$ points)

Let $\mathcal{G} := \langle V, E \rangle$ be an undirected graph. A *vertex cover* of $\mathcal{G}$ is a vertex set $C \subseteq V$ such that for all $\langle u, v \rangle \in E$, $u \in C$ or $v \in C$.

Let $\mathcal{S} := \langle S, \mathcal{C} \rangle$ be a subset collection, i.e., S is a finite set and $\mathcal{C} = \{C_1, \ldots, C_n\}$ where $C_i \subseteq S$ for all $i \in \{1, \ldots, n\}$. A *hitting set* of $\mathcal{S}$ is a subset $H \subseteq S$ such that $H \cap C_i \neq \emptyset$ for all $i \in \{1, \ldots, n\}$.

The **VertexCover** and **HittingSet** decision problems are defined as:

$$\textbf{VertexCover} = \{\langle \mathcal{G}, n \rangle \mid \mathcal{G} \text{ is a graph which has a vertex cover of size at most } n \in \mathbb{N}_1\}$$
$$\textbf{HittingSet} = \{\langle \mathcal{S}, m \rangle \mid \mathcal{S} \text{ is a subset collection with a hitting set of size at most } m \in \mathbb{N}_1\}$$

(a) Prove that $\textsc{VertexCover} \leq_{\mathrm{p}} \textsc{HittingSet}$.

(b) Prove that $\textsc{HittingSet}$ is NP-complete. (You may use your result from part (a) and that it is known that VertexCover is NP-complete.)

Question 5 (Decidability, $4 + 10$ points)

Consider the problem of testing whether a given single-tape Turing machine ever writes a blank symbol over a non-blank symbol during the course of its computation, for any input string.

(a) Formulate this problem as a language.

(b) Show that the problem is undecidable.

Question 6 (Propositional Logic, $5 + 9$ points)

(a) Resolution is not a complete proof method. However, the *contradiction theorem* can be used to obtain a sound and complete method based on resolution for answering queries of the form "Does $\mathrm{KB} \models \varphi$?".

Describe how this is done in general, i.e., to which set of clauses the resolution method is applied, and which outcome of the resolution method means that $\mathrm{KB} \models \varphi$.

You may assume that KB is given as a set of clauses and φ as a conjunction of literals.

(b) Use the method described in part (a) to prove $\mathrm{KB} \models P \wedge R$ for

$$\mathrm{KB} = \{P \vee \neg Q, \quad P \vee Q \vee \neg R, \quad P \vee R, \quad Q \vee S, \quad R, \quad \neg R \vee S\}.$$

Question 7 (Propositional logic, $9 + 5$ points)

(a) Which of the following formulae are *satisfiable*? Which ones are *valid*? Which ones are *unsatisfiable*? For formulas belonging to several of these categories, please list *all* of them.

For all *satisfiable* cases, also provide a satisfying truth assignment. For the questions about validity and unsatisfiability, you do *not* need to justify your answers.

(i) $(A \vee \neg B) \rightarrow (A \wedge C)$

(ii) $(A \leftrightarrow B) \wedge (B \leftrightarrow \neg A)$

(iii) $(A \wedge B) \vee (\neg A \wedge \neg B)$

(*iv*) $(A \leftrightarrow B) \wedge (B \rightarrow \neg A)$

(b) Prove that

$$(A \wedge B) \rightarrow C \equiv A \rightarrow (B \rightarrow C)$$

by providing a sequence of logical equivalences that transforms the left-hand side into the right-hand side.

Question 8 (Example of multiple choice question)

In which of the following cases is the logical formula to the left a *reasonable formalization* of the natural-language sentence to the right?

- ☐ $\forall x \forall y ((\mathit{LivesIn}(x, y) \wedge \neg \mathit{EatsUp}(x)) \rightarrow \mathit{BadWeatherIn}(y))$ "Whenever someone who lives in some place does not eat up, the weather in that place will be bad."
- ☐ $\forall x \forall y (\mathit{Friend}(x, y) \wedge \mathit{Friend}(y, x))$ "Whenever A is a friend of B, B is a friend of A."
- ☐ $\forall x \forall y (\mathit{FatherOf}(x, \mathrm{me}) \wedge \mathit{DaughterOf}(y, x) \wedge \mathit{Female}(\mathrm{me})) \rightarrow (y = \mathrm{me})$ "If my father has a daughter and I am female, then that daughter is me."
- ☐ $\exists x \forall y \mathit{Father}(x, y)$ "Everybody has at least one father."
- ☐ $\mathit{DaughterOf}(\mathrm{me}, \mathit{Friend})$ "I am the daughter of my friend."