Theory of Computer Science

G. Röger
F. Pommerening
Spring Term 2020

University of Basel
Computer Science

Exercise Sheet 11 — Solutions

Exercise 11.1 (Undecidability; 1+3+2 marks)

(a) Show that for every alphabet Σ there is a context-free grammar $G_{\text{palindrome}}(\Sigma)$ that generates exactly the non-empty palindromes over Σ.

Solution:

The grammar $G_{\text{palindrome}}(\Sigma) = \langle \Sigma, \{S\}, P, S \rangle$ with $P = \{S \to x \mid x \in \Sigma\} \cup \{S \to xx \mid x \in \Sigma\} \cup \{S \to xSx \mid x \in \Sigma\}$ generates exactly the non-empty palindromes over Σ and is context-free.

(b) Consider a PCP-instance $\mathcal{I} = \langle (x_1, y_1), \ldots, (x_k, y_k) \rangle$ with $x_i, y_i \in \Sigma^+$. Let $\Sigma' = \Sigma \cup \{\#\}$ where $\#$ is a symbol that does not occur in Σ. Specify a context-free grammar $G_{\mathcal{I}}$ over Σ', that generates the following language (with $^{-1}$ we denote the reverse of a word here):

$$\mathcal{L}(G_{\mathcal{I}}) = \{x_{i_1} \ldots x_{i_n} \# y_{i_n}^{-1} \ldots y_{i_1}^{-1} \mid n \geq 1 \text{ and } i_1, \ldots, i_n \in \{1, \ldots, k\}\}$$

Moreover, show that $\mathcal{I}$ has a solution iff $\mathcal{L}(G_{\mathcal{I}})$ contains a palindrome.

Solution:

The grammar $G_{\mathcal{I}} = \langle \Sigma', \{S, Z\}, P, S \rangle$ with $P = \{S \to x_i Z y_i^{-1} \mid 1 \leq i \leq k\} \cup \{Z \to x_i Z y_i^{-1} \mid 1 \leq i \leq k\} \cup \{Z \to \#\}$ generates the desired language and is context-free.

- If $i_1, \ldots i_n$ is a solution of $\mathcal{I}$ then we can derive a palindrome in $G_{\mathcal{I}}$:
  We first use the rule $S \to x_1 Z y_1^{-1}$ and then the rules $Z \to x_i Z y_i^{-1}$ for $i = i_2, \ldots, i = i_k$. Finally, we use $Z \to \#$. The word generated in this way is $x_{i_1} \ldots x_{i_n} \# y_{i_n}^{-1} \ldots y_{i_1}^{-1}$. It satisfies $x_{i_1} \ldots x_{i_n} = y_{i_1} \ldots y_{i_n} = (y_{i_n}^{-1} \ldots y_{i_1}^{-1})^{-1}$ (because $i_1, \ldots i_n$ is a solution of $\mathcal{I}$) and is thus a palindrome.
- If we can derive a palindrome in $G_{\mathcal{I}}$ then there exists a solution of $\mathcal{I}$:
  We first note that there is only one rule that can derive $\#$ and this rule must be used exactly once. Hence, the symbol $\#$ occurs once in every derived word. If this word is a palindrome, this occurrence must be in the middle of the word and the word has form $x\#y$ with $x, y \in \Sigma^+$ and $x = y^{-1}$. Moreover, there must be indices $i_1, \ldots, i_n$ such that $x = x_{i_1} \ldots x_{i_n}$ and $y = y_{i_n}^{-1} \ldots y_{i_1}^{-1}$ because otherwise this word would not be in the language. Together, we get that $x_{i_1} \ldots x_{i_n} = x = y^{-1} = (y_{i_n}^{-1} \ldots y_{i_1}^{-1})^{-1} = y_{i_1} \ldots y_{i_n}$, so $i_1, \ldots, i_n$ is a solution for $\mathcal{I}$.

(c) Use the results from parts (a) and (b) to prove that the intersection problem of *context-free* languages is undecidable.

INTERSECTION$_{\text{CF}}$: Given two context-free grammars G_1 and G_2, is $\mathcal{L}(G_1) \cap \mathcal{L}(G_2) = \emptyset$?

Note: of course you can use the statements of parts (a) and (b) even if you did not solve the exercises.

Solution:

We show that $\overline{\text{PCP}} \leq \text{INTERSECTION}_{\text{CF}}$. As PCP is undecidable its complement $\overline{\text{PCP}}$ is also undecidable.

We use reduction function $f(\mathcal{I}) = \langle G_{\mathcal{I}}, G_{\text{palindrome}}(\Sigma')\rangle$. This function is total and computable because we can create the alphabet Σ' and the grammars $G_{\mathcal{I}}$ and $G_{\text{palindrome}}(\Sigma')$ for any instance $\mathcal{I}$.

We know from part (a) that the context-free grammar $G_{\text{palindrome}}(\Sigma')$ generates exactly the language of all (non-empty) palindromes over Σ'. From part (b) we know that the context-free grammar $G_{\mathcal{I}}$ for an instance $\mathcal{I}$ of the PCP generates a language that contains a palindrome iff there is a solution for $\mathcal{I}$. Thus, $G_{\mathcal{I}} \cap G_{\text{palindrome}}(\Sigma') \neq \emptyset$ iff $\mathcal{I}$ has a solution.

Overall, we get

$$\mathcal{I} \notin \text{PCP iff } G_{\mathcal{I}} \cap G_{\text{palindrome}}(\Sigma') = \emptyset \text{ iff } f(\mathcal{I}) \in \text{INTERSECTION}_{\text{CF}}$$

Hence, f is a reduction from $\overline{\text{PCP}}$ to $\text{INTERSECTION}_{\text{CF}}$. Since $\overline{\text{PCP}}$ is undecidable, this implies that $\text{INTERSECTION}_{\text{CF}}$ is also undecidable.

Exercise 11.2 (Non-deterministic Algorithms; 2+2 marks)

Specify a non-deterministic polynomial algorithm for each of the following problems. This shows that the problems are in NP, a complexity class we will introduce next week.

(a) HITTINGSET:

- *Given:* finite set M, set of sets $\mathcal{S} = \{S_1, \ldots, S_n\}$ with $S_i \subseteq M$ for all $i \in \{1, \ldots, n\}$, natural number $k \in \mathbb{N}_0$
- *Question:* Is there a set H with at most k elements, which contains at least one element from each set in $\mathcal{S}$.
  Formally: Is there a set H with $|H| \leq k$ and $H \cap S_i \neq \emptyset$ for all $i \in \{1, \ldots, n\}$?

Solution:

Input: set M, set of sets $\mathcal{S}$, number k

hittingset := $\emptyset$
remaining := M
not_hit := $\mathcal{S}$
WHILE $|$*hittingset*$| < k$:
    GUESS *next* $\in$ *remaining*
    remaining := *remaining* $\setminus$ {*next*}
    hittingset := *hittingset* $\cup$ {*next*}
    not_hit_prev := copy of *not_hit*
    FOR $s \in$ *not_hit_prev* :
        IF *next* $\in s$
            not_hit := *not_hit* $\setminus$ {s}
    IF *not_hit* $= \emptyset$:
        ACCEPT
REJECT

(b) SETPACKING:

- *Given:* finite set M, set of sets $\mathcal{S} = \{S_1, \ldots, S_n\}$ with $S_i \subseteq M$ for all $i \in \{1, \ldots, n\}$, natural number $k \in \mathbb{N}_0$

- *Question:* Is there a set $\mathcal{S}' \subseteq \mathcal{S}$ with $|\mathcal{S}'| \geq k$, such that all sets in $\mathcal{S}'$ are pairwise disjoint, i.e., for all $S_i, S_j \in \mathcal{S}'$ with $S_i \neq S_j$ it holds that $S_i \cap S_j = \emptyset$?

Solution:

Input: set M, set of sets $\mathcal{S}$, number k

$chosen := \emptyset$
$remaining := \mathcal{S}$
WHILE $|chosen| < k$:
    IF $remaining = \emptyset$:
        REJECT
    GUESS $next \in remaining$
    FOR $set \in chosen$:
        IF $next \cap set \neq \emptyset$:
            REJECT
    $chosen := chosen \cup \{next\}$
    $remaining := remaining \setminus \{next\}$
ACCEPT