Course 02249, DTU Compute
Carsten Witt

Computationally Hard Problems – Fall 2016
Assignment 1

Date: 30.08.2016, **Due date**: 05.09.2016, 21:00

The following exercises are **not** mandatory:

Exercise 1.1: Consider the language L_{graphs} from Example 2.3 in the lecture notes. Let $w \in \{0,1\}^k$ be a word of length k over $\{0,1\}$.

a) Describe how one can check that it encodes an undirected graph.

b) If w encodes an undirected graph, describe how one can check whether a certain edge $\{i,j\}$, where $j > i$, is present.

End of Exercise 1

Exercise 1.2: Design a language L_{dgraphs} for directed graphs. Do not forget to specify the alphabet Σ_{dgraphs} you use.

End of Exercise 2

Exercise 1.3: Design a language for sequences of fractions. A fraction is the quotient of two integers.

a) Specify the alphabet.

b) Specify how the language is defined.

c) Show how the sequence 1/3, 234/39, −55/16 is coded in your language.

End of Exercise 3

Continued on next page.

Exercise 1.4: A *disjunctive form* is a formula over n boolean variables $x_1, \ldots, x_n$ such that

- the formula is a boolean disjunction of k so-called monomials $m_1, \ldots, m_k$,
- where each monomial m_i, $1 \leq i \leq k$, is a boolean conjunction of variables or their negations.

For example, a disjunctive form for $n = 5$ and $k = 4$ is

$$\underbrace{(\overline{x}_1 \wedge x_2 \wedge x_3)}_{m_1} \vee \underbrace{(\overline{x}_1 \wedge x_4)}_{m_2} \vee \underbrace{(x_1 \wedge \overline{x}_3 \wedge x_4 \wedge x_5)}_{m_3} \vee \underbrace{(x_2 \wedge \overline{x}_3 \wedge \overline{x}_5)}_{m_4}.$$

Design a language $L_{\text{disj-form}}$ for the set of disjunctive forms. Do not forget to specify the alphabet $\Sigma_{\text{disj-form}}$ you use.

End of Exercise 4

The following exercise is **mandatory**:

Exercise 1.5: A *chess game configuration* describes the current state in the well-known chess game. Such a state includes the positions of the up to 32 available pieces on the 8×8-chess board and the color of the player whose turn it is next. The aim is to design a formal language L_{chess} for chess game configurations.

a) Specify the alphabet Σ_{chess} you use.

b) Specify how a chess game configuration is encoded in the language L_{chess}.

c) Describe how one can check whether a given word $w \in \Sigma^*_{\text{chess}}$ is in L_{chess}, and, if so, how the game configuration can be reconstructed.

d) How would the initial game configuration (i. e., the configuration the game is set up to begin with) be encoded in your language?

Note: a game configuration only describes how pieces are arranged on the board and whose turn it is. We do not check whether the arrangement can be reached in a real chess game.

End of Exercise 5