Universität Koblenz-Landau

FB 4 Informatik

Prof. Dr. Viorica Sofronie-Stokkermans
Dipl. Inf. Markus Bender

January 24, 2013

Exercises for
"Advances in Theoretical Computer Science"
Exercise sheet 13
Due on 29.1.13, 09:00 s.t.

Exercise 13.1:

Give a function $f : \Sigma^* \to \Sigma^*$ which polynomially reduces L_1 to L_2, or explain why this is not possible:

(1) $\Sigma = \{0, 1, 2\}$;
$L_1 = \{w \in \{0, 1\}^* \mid w$ is the binary representation of a prime number$\}$;
$L_2 = \{w \in \{0, 1, 2\}^* \mid w$ is the representation of a prime number in base $3\}$.

(2) $\Sigma = \{0, 1, 2, 3, 4\}, L_1$ as in (1);
$L_2 = \{w \in \{0, 1, 2, 3, 4\}^* \mid w$ is the representation of a prime number in base $5\}$.

(3) $\Sigma = \{0, 1\}, L_1$ as in (1);
$L_2 = \{w \in \{1\}^* \mid w$ is the representation of a prime number in base $1\}$.

(4) $\Sigma = \{a, b, 0, 1\}$; $L_1 = \{a^n b^n \mid n \geq 0\}$; $L_2 = \{1\}$.
(In (4) we require that for all $w \in \Sigma^* \backslash L_1$, $f(w) = 0$.)

Exercise 13.2:

Definitions. Assume we are in propositional logic with propositional variables Π.

- A *literal* L is a propositional variable P or the negation of a propositional variable $\neg P$.
- A propositional formula is in *disjunctive normal form (DNF)* if it has the form $(L_1^1 \wedge \cdots \wedge L_{n_1}^1) \vee \cdots \vee (L_1^m \wedge \cdots \wedge L_{n_m}^m)$.
- A propositional formula is a *clause* if it is of the form $L_1 \vee \cdots \vee L_n$ (i.e. is a disjunction of literals). A *Horn clause* is a clause which contains at most one positive literal. (For instance $P \vee \neg Q \vee \neg R$ and $\neg Q \vee \neg R$ are Horn clauses but $P \vee Q \vee \neg R$ is not a Horn clause.)

(1) Let DNF-SAT $= \{F \mid F$ is a satisfiable formula of propositional logic in disjunctive normal form$\}$.
Show that DNF-SAT is in P.

(2) Let Horn-SAT $= \{F \mid F$ is a satisfiable conjunction of Horn clauses$\}$.

Is Horn-SAT in NP? Is Horn-SAT NP-complete? Is Horn-SAT in P? Justify your answer.

Remark: For solving these problems you do not need to construct Turing machines. You can use results on propositional logic presented e.g. in the lecture "Logik für Informatiker" (e.g. the existence of algorithms for checking the satisfiability of sets (= conjunctions) of Horn clauses).

Exercise 13.3:
We know that SAT is NP-complete. In the previous exercise we saw that DNF-SAT is in P.

If we could construct a polynomial reduction of SAT to DNF-SAT (i.e. if we could prove that SAT $\prec_{\mathsf{pol}}$ DNF-SAT) then we could show that P = NP.

Formulae in propositional logic can be transformed to DNF using distributivity:

$$A \land (B_1 \lor \cdots \lor B_k) \equiv (A \land B_1) \lor \cdots \lor (A \land B_k).$$

Why does this not lead to a polynomial reduction?

The submission of the solutions is not compulsory. If you want to submit your solutions, please do so until 29.1.13, 09:00 s.t.. Joint solutions prepared by up to three persons are allowed. Please do not forget to write your name on your solution.
Submission possibilities:

- By e-mail to mbender@uni-koblenz.de with the keyword "Homework ACTCS" in the subject.
- Put it in the box in front of Room B 222.