CAS 701
Logic and Discrete Mathematics
Fall 2017

Exercise Group 3

Dr. William M. Farmer
McMaster University

Revised: October 18, 2017

You are required to submit solutions to 10 of the following 15 exercises. If you submit more than 10, only the first 10 will be marked. Your solutions should begin with a table of contents that indicates which of the 15 exercises you have chosen to do. Each question is worth 10 points. **The solutions must be written using LaTeX and are due November 3, 2017 at the beginning of class.**

1. Show that the following functions on the natural numbers are primitive recursive:

   a. Addition.

   b. Multiplication.

   c. Exponentiation.

2. Show that the version of the Ackermann function presented in the lecture notes is an instance of well-founded recursion.

3. Give a natural example of a well-founded relation that is not a partial order.

4. Let $f : \mathbb{N} \rightarrow \mathbb{N}$ generate the Fibonacci sequence $0, 1, 1, 2, 3, 5, \ldots$.

   a. Show that f is a primitive recursive function.

   b. Define f by well-founded recursion.

   c. Define f by recursion via a monotone functional.

5. Let PROP be the set of propositional formulas defined in the lecture notes. For $A \in \mathsf{PROP}$, let $p(A)$ be the number of propositional symbols occurring in A and $c(A)$ be the number of propositional connectives occurring in A. Prove by structural induction that, for all $A \in \mathsf{PROP}$,

$$p(A) \leq c(A) + 1.$$

6. Let $A \in \mathsf{PROP}$. A *subformula* of A is a member of PROP that is a substring of A. Define this notion recursively and then prove that, if $A \in \mathsf{PROP}$ contains n propositional connectives, A contains at most $2n + 1$ subformulas.

7. Construct a truth table for each of the following propositional formulas:

   a. $P \equiv \neg P$.
   b. $(P \wedge Q) \supset (P \vee Q)$.
   c. $(Q \supset \neg P) \equiv (P \equiv Q)$.
   d. $(P \supset Q) \wedge (\neg P \supset Q)$.
   e. $((P \supset Q) \supset R) \supset S$.
   f. $(P \wedge Q) \supset (P \supset Q)$.

   Or, alternatively for double marks, implement a program in your programming language of choice that, given a propositional formula A as input, returns a truth table for A, and then apply your program to the six propositional formulas above.

8. Which of the formulas in the previous question are

   a. Valid?
   b. Invalid (falsifiable)?
   c. Satisfiable?
   d. Unsatisfiable?

9. The logician Raymond Smullyan liked to talk about an island whose inhabitants are knights and knaves. The knights always tell the truth and the knaves always lie. Suppose you encounter two people, A and B. A says “The two of us are both knights” and B says “A is a knave”. What can you determine about A and B?

10. Suppose you encounter two people on Smullyan’s island, A and B. Both A and B say “I am a knight”. What can you determine about A and B?

11. Show that the propositional connective nand (Sheffer stroke) is not associative.

12. Assuming that $\{\neg, \vee\}$ is a complete set of propositional connectives, show that the set $\{\mathsf{nand}\}$ is also complete.

13. Suppose A is a propositional formula with the following truth table:

P	Q	R	A
T	**T**	**T**	**F**
T	**T**	**F**	**F**
T	**F**	**T**	**F**
T	**F**	**F**	**T**
F	**T**	**T**	**T**
F	**T**	**F**	**F**
F	**F**	**T**	**F**
F	**F**	**F**	**T**

a. Construct a formula B in conjunctive normal form such that A and B are logically equivalent.

b. Construct a formula C in disjunctive normal form such that A and C are logically equivalent.

14. Write a program in your favorite programming language that, given $A \in \mathsf{PROP}$ as input, returns $B \in \mathsf{PROP}$ as output such that B is in conjunctive normal form and $A \equiv B$ is a tautology. Use your program to solve the previous question.

15. Let $\Gamma \cup \{A, B\} \in \mathsf{PROP}$. Prove the following statements:

a. $\Gamma \cup \{A\} \vDash B$ iff $\Gamma \vDash A \supset B$.

b. $\Gamma \cup \{A\} \vDash B$ and $\Gamma \cup \{A\} \vDash \neg B$ implies $\Gamma \vDash \neg A$.

c. $\Gamma \cup \{A\} \vDash C$ and $\Gamma \cup \{B\} \vDash \neg C$ implies $\Gamma \cup \{A \vee B\} \vDash C$.