Final Exam 2ITX0 Applied Logic

December 2019

This examination is intended to be representative to what may be expected at the real examination of Applied Logic 2ITX0.

This real examination will be supported by ANS: then you have to write the answers in indicated boxes.

Problem 1.

Give a propositional formula in the boolean variables a, b, c, d describing that b is true and exactly one of the variables a, c, d is false.

Problem 2.

Consider the CNF consisting of the following eight clauses

(1)	$\neg p \vee t$	(5)	$p \vee \neg r$
(2)	$\neg q \vee r$	(6)	$\neg q \vee s \vee \neg t$
(3)	$p \vee q$	(7)	$r \vee \neg s \vee \neg t$
(4)	$\neg r \vee \neg s$	(8)	$\neg p \vee q \vee s$

(a) For each of the values $i = 1, 2, 3$ establish whether a resolution step is possible on clause (i) and clause (6), and if so, give the result of this resolution step.

(b) Prove that this CNF is unsatisfiable by DPLL; indicate for every step the number of the clause that is used.

Problem 3.

Compute the Tseitin transformation of the formula $(p \wedge \neg q) \vee (r \rightarrow (p \wedge q))$.

Problem 4.

An information source produces symbols from the set $\{A, B, C\}$, independently distributed with probabilities $\Pr(A) = 0.1$, $\Pr(B) = 0.3$, and $\Pr(C) = 0.6$.

1. Calculate the entropy of this source, measured in bits. Show your calculation, and round your answer to two decimal places. The following binary logarithms are given:

p	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$-\log_2 p$	3.32	2.32	1.74	1.32	1.00	0.737	0.515	0.322	0.152

2. Using Huffman's algorithm, construct a binary prefix-free encoding tree for this source that uses, on average, less than 1.34 bits per symbol, by encoding *pairs* of symbols. For these pairs we have:

pair	AA	AB	AC	BA	BB	BC	CA	CB	CC
probability	0.01	0.03	0.06	0.03	0.09	0.18	0.06	0.18	0.36

3. Show by calculation that, on average, pairs are then encoded in less than $2 \times 1.34 = 2.68$ bits.

Problem 5.

Consider the code that encodes the two data bits $s_1 s_2$ into the 5-bit codeword $s_1 s_2 s_1 s_2 p$, where $p = s_1 \oplus s_2$, and $\oplus$ is addition modulo 2.

1. What is the rate (compression ratio) of this code?
2. List all the codewords in this code.
3. What is the maximum number of bit errors that can always be *detected* with this code? Why?
4. What is the maximum number of bit errors that can always be *corrected* with this code? Why?

Problem 6.

1. Considering symmetric and asymmetric encryption, which one is — without additional measures to prevent it— vulnerable to the man-in-the-middle attack, and which one is not?
2. Explain how such an attack would work.
3. Why does it not apply to the other type of encryption?

Problem 7.

Start with two red marbles and two blue marbles. In every step either

- the number of blue marbles is doubled and the number of red marbles is decreased by one, or
- all marbles swap colors, so the blue marbles are made red and conversely.

Give an SMT formula (either in SMT syntax or standard logical syntax) specifying whether it is possible to end up with exactly 10 red marbles after doing exactly 4 such steps.

Problem 8.

Compute $wp(\langle$ if $a < b$ then $a := a + b\rangle; b := a + b, a = b)$ and
$wp($ if $a < b$ then $\langle a := a + b; b := a + b\rangle, a = b)$.
Simplify your results as much as possible.

Problem 9.

Prove partial correctness of

$\{a = A \wedge b = B \wedge a \geq 0\}$
while $a \neq 0$ do
if $a > 12$ then $\langle a := a - 10;\ b := b + 30\rangle$
else $\langle a := a - 1;\ b := b + 3\rangle$
$\{b = 3A + B\}$