Christian-Albrechts-Universität zu Kiel
Technische Fakultät

Dependable Systems Group
D. Nowotka, P. Fleischmann

Example solution for Series #6

Exercise 1 — 10 Points

You get for each correct answer 1 point, but you will lose 1 point for an incorrect answer.

a) If $\varphi \in \Phi$ is a formula in CNF and C is a clause in φ that is a contradiction then φ is a contradiction. ☒ true ○ false

b) If a clause containts $p \vee \neg p$ for an atom $p \in A$ then the clause is a tautology. ☒ true ○ false

c) For checking the validity of a formula in CNF it suffices to check one clause for validity. ○ true ☒ false

d) The naïve algorithm for transforming a formula into CNF has polynomial run-time in the number of atoms. ○ true ☒ false

e) The CNF of a formula is unique. ○ true ☒ false

f) The Horn-SAT algorithm has exponential run-time. ○ true ☒ false

g) The Horn-SAT algorithm is non-deterministic. ○ true ☒ false

h) An atom $p \in A$ is a clause. ☒ true ○ false

i) An atom $p \in A$ is a formula in CNF. ☒ true ○ false

j) Two clauses of a formula in CNF are not allowed to have the same clauses. ○ true ☒ false

Exercise 2 — 3.5 Points

Give the following definitions and notations:

a) Clause. (1P)
b) Formula $\varphi \in \Phi$ in CNF. (1P)
c) Horn clause. (1.5P)

Solution:

a) A formula is a clause if it is a disjunction of literals. (0.5+0.5P)
b) φ is a conjunction of clauses. (0.5+0.5P)
c) A clause is a Horn clause if exactly one literal is a positive atom. (0.5+0.5+0.5P)

Exercise 3 — 17 Points

a) Prove the following remaining cases from the theorem that if η is a tautology then η is a theorem.
   1) For formulae $\varphi_1, \varphi_2 \in \Phi$ it holds $\neg\varphi_1 \wedge (\varphi_2 \vee \neg\varphi_2) \vdash \varphi_1 \rightarrow \varphi_2$. (4.5P)
   2) For formulae $\varphi_1, \varphi_2 \in \Phi$ it holds $\varphi_1 \wedge \neg\varphi_2 \vdash \neg(\varphi_1 \rightarrow \varphi_2)$. (4P)

b) Transform the following formula into CNF by using the naïve algorithm: $\varphi = \neg(p \rightarrow (s \wedge q))$. (4.5P)

c) Apply the Horn-SAT algorithm on $\varphi = (p \vee \neg q \vee \neg s) \wedge (q \vee \neg s \vee \neg r) \wedge r$. (4P)

Solution:

a) 1) We have $\neg\varphi_1 \wedge (\varphi_2 \vee \neg\varphi_2)$ as premise. (1) (0.5P)
The $\wedge$ elimination rules on (1) give $\neg\varphi_1$ (2) (0.5P)
and $(\varphi_2 \vee \neg\varphi_2)$ (3) (0.5P)
For proving $\varphi_1 \to \varphi_2$ the $\to$-introduction rule needs to be applied. (0.5P)
Thus assume φ_1. (4) (0.5P)
The $\wedge$-introduction rule on (2) and (4) gives $\neg\varphi_1 \wedge \varphi_1$. (5) (0.5P)
(5) is a contradiction and by (cd) we get $\perp$. (6) (0.5P)
By ex falso quodlibet we get φ_2. (7) (0.5P)
By $\to$-introduction we get finally $\varphi_1 \to \varphi_2$. (0.5P)

2) We have $\varphi_1 \wedge \neg\varphi_2$ as premise. (1) (0.5P)
Suppose $\varphi_1 \to \varphi_2$. (2) (0.5P)
The first $\wedge$-elimination rule on (1) gives us φ_1. (3) (0.5P)
By modus ponens on (2) and (3) we get φ_2. (4) (0.5P)
The second $\wedge$-elimination rule on (1) delivers $\neg\varphi_2$. (5) (0.5P)
The $\wedge$-introduction rule on (4) and (5) gives us $\varphi_2 \wedge \neg\varphi_2$. (6) (0.5P)
Thus we have a contradiction. (0.5P)
By (tnd) we finally get $\neg(\varphi_1 \to \varphi_2)$. (0.5P)

b) For applying the naïve algorithm we need to build the truth table for φ.

p	q	s	φ
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

(1P)

Now we have to consider the rows where φ is evaluated to false and to negate them:

p	q	s	clause
0	0	0	$p \vee q \vee s$
0	0	1	$p \vee q \vee \neg s$
0	1	0	$p \vee \neg q \vee s$
0	1	1	$p \vee \neg q \vee \neg s$
1	1	1	$\neg p \vee \neg q \vee \neg s$

(2.5P)

This delivers

$$(p \vee q \vee s) \wedge (p \vee q \vee \neg s) \wedge (p \vee \neg q \vee s) \wedge (p \vee \neg q \vee \neg s) \wedge (\neg p \vee \neg q \vee \neg s).$$

(1P)

c) The given formula is of the form

$$\varphi \equiv ((q \wedge s) \to p) \wedge ((s \wedge r) \to q) \wedge \top \to r.$$

(1P)

Since we have to mark atoms, we have to maintain the following array

p	q	r	s	$\top$	$\perp$

(0.5P)

In the first step we mark $\top$, i.e.

p	q	r	s	$\top$	$\perp$
				x	

(0.5P)

The while loop is executed with $\top \to r$ and r is marked

p	q	r	s	$\top$	$\perp$
		x		x	

(0.5P)

Now the while-loop constraint is not satisfied. (0.5P)
Since $\perp$ is not marked, (0.5P)
the algorithm returns SAT. (0.5P)