E-Test 1

Question 1

Which of the following formulas are not tautologies but satisfiable?

Wählen Sie eine oder mehrere Antworten:

- [ ] $((b \wedge (a \wedge a)) \rightarrow ((a \rightarrow a) \leftrightarrow (a \wedge b)))$
- [ ] $((\neg b) \rightarrow (\neg (b \vee b)))$
- [x] $((b \vee (b \wedge b)) \vee ((a \wedge a) \wedge (a \rightarrow a)))$ ✔
- [ ] $((\neg b) \rightarrow (\neg (b \vee b)))$
- [x] $((\neg (b \vee b)) \leftrightarrow ((a \wedge b) \wedge (a \wedge a)))$ ✔
- [x] $(((a \leftrightarrow b) \rightarrow (\neg a)) \vee ((\neg b) \leftrightarrow (b \wedge b)))$ ✔

Note: Closely watch the formulation of the task. It can also be "Which of the following formulas ARE tautologies" or something similar!

For this task we can simplify the formulas to obtain whether they are tautologies, satisfiable, or unsatisfiable. Additionally, we can check the results using: https://www.erpelstolz.at/cgi-bin/cgi-logik

a)

$(b \wedge \underbrace{(a \wedge a)}_{=a}) \rightarrow (\underbrace{(a \rightarrow a)}_{=\text{True}} \leftrightarrow (a \wedge b))$

iff $\big((a \wedge b) \rightarrow (\text{True} \leftrightarrow (a \wedge b))\big)$

iif $\neg a \vee \neg b \vee \Big(\big(\text{True} \rightarrow (a \wedge b)\big) \wedge \big((a \wedge b) \rightarrow \text{True}\big)\Big)$

iif $\neg a \vee \neg b \vee \Big(\big(\text{False} \vee (a \wedge b)\big) \wedge (\neg a \vee \neg b \vee \text{True})\Big)$

iif $(a \wedge b) \vee \neg a \vee \neg b$

Tautology. $\Rightarrow$ FALSE

b)

$(\neg b) \rightarrow (\neg \underbrace{(b \vee b)}_{=b})$

iif $\neg b \rightarrow \neg b$

iif $b \vee \neg b$

Tautology. $\Rightarrow$ FALSE

c)

$\underbrace{\big(b \vee (b \wedge b)\big)}_{=b} \vee (\underbrace{(a \wedge a)}_{=a} \wedge \underbrace{(a \rightarrow a)}_{=\text{True}})$

iif $b \vee a$

No tautology ($a \mapsto 0, b \mapsto 0$), but satisfiable ($a \mapsto 1, b \mapsto 0$)

d)

$(\neg b) \rightarrow \underbrace{\big(\neg (b \vee b)\big)}_{=\neg b}$

iif $b \vee \neg b$

Tautology. $\Rightarrow$ FALSE

e)

$\underbrace{\big(\neg (b \vee b)\big)}_{=\neg b} \leftrightarrow ((a \wedge b) \wedge \underbrace{(a \wedge a)}_{=\text{True}})$

iif $\neg b \leftrightarrow (a \wedge b)$

iif $\big(\neg b \rightarrow (a \wedge b)\big) \wedge \big((a \wedge b) \rightarrow \neg b\big)$

iif $(b \vee (a \wedge b)) \wedge (\neg a \vee \neg b)$
iif $(a \vee b) \wedge b \wedge (\neg a \vee \neg b)$
iif $\neg a \wedge b$
No tautology ($a \mapsto 0, b \mapsto 0$), but satisfiable ($a \mapsto 0, b \mapsto 1$)

f)
$\left(((a \leftrightarrow b) \rightarrow (\neg a)) \vee ((\neg b) \leftrightarrow (b \wedge b))\right)$
iif $\left(((\neg a \vee b) \wedge (\neg b \vee a)) \rightarrow \neg a\right) \vee \underbrace{(\neg b \leftrightarrow b)}_{\text{False}}$
iif $\neg((\neg a \vee b) \wedge (\neg b \vee a)) \vee \neg a$
iif $(a \wedge \neg b) \vee (b \wedge \neg a) \vee \neg a$
No tautology ($a \mapsto 1, b \mapsto 1$), but satisfiable ($a \mapsto 0, b \mapsto 0$)

Question 2

Assume the following propositional logic formula in CNF:
$(\neg A \vee \neg C \vee E \vee F) \wedge (A \vee B \vee \neg F) \wedge (B \vee E) \wedge (\neg D \vee \neg F) \wedge (A \vee \neg B \vee \neg F) \wedge (D \vee F) \wedge (\neg A \vee B \vee \neg D \vee E) \wedge (A \vee C \vee \neg D \vee \neg E \vee \neg F)$
Apply the DPLL+CDCL algorithm until it detects either a conflict or a complete solution. For decision, always take the smallest unassigned variable in the order $A < B < C < D < \ldots$ and assign false to it.
At which decision level is the first conflict or a full solution detected? Please answer by writing the number using digits without whitespaces.

Antwort: 3 ✔

DL0: –
Nothing, as there is no clause with only one single literal.

DL1: $\neg \boldsymbol{A}: \boldsymbol{nil}$
Due to our variable ordering, we assign $A \coloneqq False$. As there is no clause with only a single unassigned literal, we cannot propagate A any further.

DL2: $\neg \boldsymbol{B}: \boldsymbol{nil}, \neg \boldsymbol{F}: \boldsymbol{nil}, \boldsymbol{D}: \boldsymbol{nil}$
After assigning $B \coloneqq False$, we propagate it:

- $(A \vee B \vee \neg F)$: $\neg F: nil$
  Propagate $F \coloneqq False$:
  - $(D \vee F)$: $D: nil$
    Propagate $D \coloneqq True$: Nothing to do.
- $(B \vee E)$: $E: nil$
  Propagate $E \coloneqq True$: Nothing to do.

DL3: $\neg \boldsymbol{C}: \boldsymbol{nil}$ DL2后，式子已经satisfiable，但是此时C没有赋值或通过BCP赋值，full solution需要对每一个literal 赋值
After assigning $C \coloneqq False$, we have found a satisfying assignment as we have no conflicts.

It might be helpful to keep track of the assignments using green & red colors:
$(\cancel{\neg A} \vee \cancel{\neg C} \vee \cancel{E} \vee \cancel{F}) \wedge (\cancel{A} \vee \cancel{B} \vee \cancel{\neg F}) \wedge (\cancel{B} \vee E) \wedge (\cancel{\neg D} \vee \cancel{\neg F}) \wedge (\cancel{A} \vee \cancel{\neg B} \vee \cancel{\neg F}) \wedge (\cancel{D} \vee \cancel{F}) \wedge (\cancel{\neg A} \vee \cancel{B} \vee \cancel{\neg D} \vee \cancel{E}) \wedge$
$(\cancel{A} \vee \cancel{C} \vee \cancel{\neg D} \vee \cancel{\neg E} \vee \cancel{\neg F})$

Question 3

Assume the following propositional logic formula in CNF:
$c_0 : (A \vee B \vee E) \wedge c_1 : (A \vee \neg C) \wedge c_2 : (B \vee \neg D) \wedge c_3 : (B \vee E) \wedge c_4 : (A \vee B \vee C \vee D \vee \neg E)$
Assume furthermore the following trail:
DL0: -
DL1 : $\neg A : nil \; \neg C : c_1$
DL2: $\neg B : nil \; E : c_0 \; \neg D : c_2$
Apply conflict resolution to c_4 till the first unique implication point as presented in the lecture. How many literals are in the derived clause? Please answer by writing the number using digits without whitespaces.

Antwort: 3 ✔

We always apply conflict resolution using the last assigned variable. We repeat the resolution until we only have one literal left from the current decision level (thus, only one of B, E, D)

$$\dfrac{\dfrac{c_4: (A \vee B \vee C \vee D \vee \neg E) \quad c_2: (B \vee \neg D)}{c_5: (A \vee B \vee C \vee \neg E) \quad c_0: (A \vee B \vee E)}}{c_6: (A \vee B \vee C)}$$

<u>Additional explanation:</u>

- In the first step, the last assigned literal is c_2 and $\neg D$. Thus, we resolve c_4 and c_2
  This results in c_5, where D and $\neg D$ “vanishes” due to the resolution.
- Since we have still two literals from the current DL, we need to resolve using the second-last assigned literal, namely c_0 and E. Thus, we resolve c_5 and c_0
  Doing so eliminates the literal $\neg A$ and yields c_6, which only has one literal left from the current DL. Thus, we’ve finished.

As we only have one literal left from the current DL, we can stop here.
The derived clause has 3 literals.

Question 4

Assume two bitvectors a and b, both of length $l > 2$, in the following bitvector arithmetic expressions:
E0: $a << 1$
E1: $a >> 1$
E2: $a >> 2$
E3: $a << 2$
E4: $a \& b$
Which of the above expressions is encoded by the formula $\neg c_0 \wedge \neg c_1 \wedge \bigwedge_{\{} i = 2\}^{\{} l - 1\}(c_i \leftrightarrow a_{\{} i - 2\})$?

Wählen Sie eine oder mehrere Antworten:

- [ ] E0
- [ ] E1
- [ ] E2
- [x] E3 ✔
- [ ] E4

Recall, that the bit-vector is encoded as $x_{l-1} \dots x_0$. So, the Least-Significant Bit is x_0.

$\neg c_0 \wedge \neg c_1$ implies, that the two least-significant bits (right-most bits) must be “0”
The conjunction indicates, that all the other c_i’s are the same as a_{i-2} – thus, shifted to the left by two.
Subsequently, i.e., if we have $a = 1101$, we get $c = 110100$
Obviously, this is equivalent to a left bitshift by 2, and $E3$ is the correct answer.

Question 5

Which of the following formulas are true?

Wählen Sie eine oder mehrere Antworten:

- [ ] Tseitin's transformation takes a propositional logic formula and transforms it into a validity-equivalent propositional logic formula in CNF.
- [x] Each implication graph with a conflict has at least one unique implication point.✔
- [ ] Each transitive relation is an equivalence relation.
- [ ] None of the statements is true.
- [ ] DPLL+CDCL SAT solving works in polynomial time.
- [x] Tseitin's transformation increases the number of propositions.✔

1. No. The resulting formula is not validity-equivalent but only equi-satisfiable. This means, that if the original formula is satisfiable, the converted formula is also satisfiable (but not
2. Yes. The root decision is always a UIP.
3. No. Transitive means $a \sim b \wedge b \sim c \rightarrow a \sim c$. Equivalence relations are reflexive, symmetric, and transitive.
   For example, take the default "less than" comparison on the natural numbers.
   Then: $a < b \wedge b < c \rightarrow a < c$, thus it is transitive.
   However, $a < b \nleftrightarrow b < a$. Thus, it is not an equivalence relation (as it is not symmetric)
4. –
5. No. The algorithm works in exp-time.
6. Yes. The Tseitin's transformation introduces new propositions. Recall from the lecture:
   - Let's go back to
     $\varphi_n = (x_1 \wedge y_1) \vee (x_2 \wedge y_2) \vee \cdots \vee (x_n \wedge y_n)$
   - With Tseitin's encoding we need:
     - n auxiliary variables $h_1, \ldots, h_n$.
     - Each adds 3 constraints.
     - Top clause: $(h_1 \vee \cdots \vee h_n)$
   - Hence, we have
     - $3n + 1$ clauses, instead of 2^n.
     - $3n$ variables rather than $2n$.