Technische Universität Wien
Fakultät für Informatik
Uwe Egly
Antonius Weinzierl

WS 2013
Übungsblatt Block 2
Matrikelnummer: XXXXXXXX

Exercises on Formal Methods in Computer Science

If you would like to receive feedback *in the exercise sessions*, you should submit your solutions to TUWEL no later than *November 11th 2013*. You will get feedback in electronic form if you upload you exercises no later than *November 29th 2013*.

This exercise sheet is divided into two parts: first algorithms and techniques, second proofs and properties. Note that the questions of Block 2 in the final exam are going to have a strong emphasis on understanding and proofs. All exercises are relevant for the final exam. We strongly recommend solving *at least* the following exercises until the presentation of the solutions: Exercise 2 (a), Exercise 3 (a), (b), Exercise 6, Exercise 7 (b), Exercise 8 (a), (c), and Exercise 9 (a).

1 Algorithms and Techniques

Exercise 1 First-Order Theories

To get an intuition, what a formula means, it often helps to visualize an example instantiation of the occuring relations. That is, one visualizes a model (or interpretation) of the formula by drawing the respective relations. Binary relations can be visualized very easily as directed graphs: let $R \subseteq U \times U$ be a relation on the universe (domain) U, then the corresponding directed graph G is $G = (U, R)$. So, whenever two elements u_1 and u_2 of the universe of an interpretation are related by R, then the corresponding graph contains an edge between u_1 and u_2.
Consider the formula $\forall x \forall y \forall z : xRy \wedge yRz \rightarrow xRz$ and an interpretation I on the universe $U = \{u, v, w, t\}$ such that $I(R) = \{(u, v), (v, w), (u, w), (v, t), (u, t)\}$. Now $I(R)$ can be seen as a directed graph over U and this graph is shown in Figure 1. Since I is a model of the above formula, the shown graph is a visualization of this model.

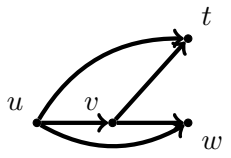


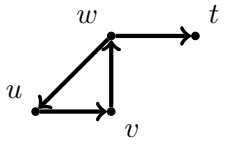
Figure 1: A graph visualizing a model of the formula $\forall x \forall y \forall z : xRy \wedge yRz \rightarrow xRz$.

(a) Let T_1 be a theory consisting of the following fomulae:

$$\forall x : xRx$$
$$\forall x \forall y : xRy \rightarrow yRx$$

i) Pick a domain of size at least 5, pick any model of T_1 based on your chosen domain, and visualize R.

ii) Consider the following graph, and extend it such that it corresponds to a model of T_1.



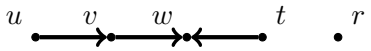
iii) Visualize a relation, which violates T_1.

(b) Visualize the theory T_2, which consists of the formula:

$$\forall x \exists y : xRy$$

i) Pick a domain of size at least 5 and visualize a chosen model of T_2.

ii) Consider the following graph, and extend it such that it corresponds to a model of T_2.



Solution:

This exercise shall provide students with a way to develop an intuitive understanding of formulas, as such it is not part of the subject matter.

Exercise 2 Tseitin Transformation

(a) For the formula $\psi = \big(a \rightarrow (b \rightarrow \neg a)\big)$ use Tseitin translation to compute a sat-equivalent CNF.

Solution:
The formula tree and the assigned labels for ψ are given in Figure 2.

The resulting equivalences are:

$$l_1 \leftrightarrow a$$
$$l_2 \leftrightarrow b$$
$$l_3 \leftrightarrow a$$
$$l_4 \leftrightarrow \neg l_3$$
$$l_5 \leftrightarrow (l_2 \rightarrow l_4)$$
$$l_6 \leftrightarrow (l_1 \rightarrow l_5)$$

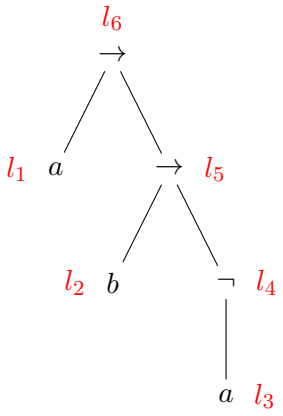


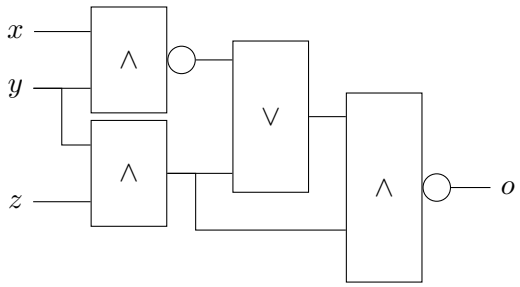
Figure 2: Formula tree for ψ and assigned labels in red.

Transforming those to CNF yields:

$$\begin{array}{lll}
\neg l_1 \vee a & l_1 \vee \neg a & \\
\neg l_2 \vee b & l_2 \vee \neg b & \\
\neg l_3 \vee a & l_3 \vee \neg a & \\
\neg l_4 \vee \neg l_3 & l_4 \vee l_3 & \\
\neg l_5 \vee \neg l_2 \vee l_4 & l_5 \vee l_2 & l_5 \vee \neg l_4 \\
\neg l_6 \vee \neg l_1 \vee l_5 & l_6 \vee l_1 & l_6 \vee \neg l_5
\end{array}$$

If we add the single clause l_6 to the above set of clauses, then the resulting set of clauses is sat-equivalent to ψ.

(b) Given the circuit below with AND, NAND, and OR gates, use Tseitin translation to obtain a linear-size (and sat-equivalent) CNF.



Solution:

We directly label the circuit dag with labels as in Figure 3. Observe that we do not assign labels to input lines here and use NAND-gates directly (instead of decomposing them into AND followed by NOT).

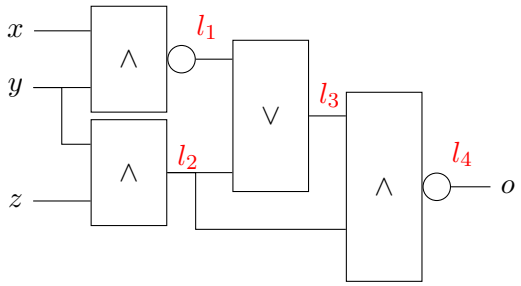


Figure 3: Labelled circuit.

So the corresponding equivalences are:

$$
\begin{aligned}
l_1 &\leftrightarrow x \uparrow y \\
l_2 &\leftrightarrow y \wedge z \\
l_3 &\leftrightarrow l_1 \vee l_2 \\
l_4 &\leftrightarrow l_3 \uparrow l_2
\end{aligned}
$$

where $\uparrow$ is the Sheffer stroke, a binary logical connective that is equivalent to a NAND gate, i.e., $x \uparrow y \equiv \neg(x \wedge y)$.

Corresponding to those equivalences are the following clauses:

$$
\begin{array}{lll}
\neg l_1 \vee \neg x \vee \neg y & l_1 \vee x & l_1 \vee y \\
\neg l_2 \vee y & \neg l_2 \vee z & l_2 \vee \neg y \vee \neg z \\
\neg l_3 \vee l_1 \vee l_2 & l_3 \vee \neg l_1 & l_3 \vee \neg l_2 \\
\neg l_4 \vee \neg l_3 \vee \neg l_2 & l_4 \vee l_3 & l_4 \vee l_2
\end{array}
$$

We add the single clause l_4 to the above set and obtain a set of clauses corresponding to the above circuit, whose size is linear in the size of the circuit.

Exercise 3 Implication Graphs

Let $\mathcal{C}$ be a clause set consisting of the following clauses:

$$
\begin{aligned}
c_1 &: \quad (\neg A \vee B) \\
c_2 &: \quad (\neg A \vee \neg B \vee C) \\
c_3 &: \quad (A \vee B) \\
c_4 &: \quad (\neg F \vee \neg B \vee \neg G) \\
c_5 &: \quad (G \vee \neg E) \\
c_6 &: \quad (G \vee D) \\
c_7 &: \quad (C \vee E \vee \neg D) \\
c_8 &: \quad (\neg A \vee C)
\end{aligned}
$$

(a) Draw an implication graph for $\mathcal{C}$. Use the decision $C = 0@1$, and $F = 1@2$ until you reach a conflict.

Solution:
The resulting conflict graph is given in Figure 5.

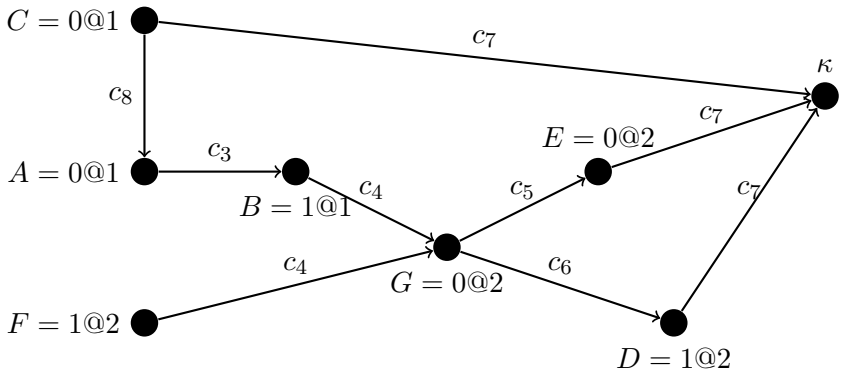


Figure 4: Implication Graph for $\mathcal{C}$ with decisions $C = 0@1$ and $F = 1@2$.

(b) Determine all UIPs in the implication graph, find the first UIP and use resolution to learn a conflict clause corresponding to the first UIP.

Solution:
UIPs (nodes through which all paths from the current decision to the conflict go through) are the nodes $F = 1@2$ and $G = 0@2$ where the latter is the first UIP (closest to the conflict).

We resolve c_7, c_5, and c_6 and obtain:

$$
\begin{aligned}
r_1 := res(c_7, c_5, E) &= (C \vee G \vee \neg D) \\
r_2 := res(r_1, c_6, D) &= (C \vee G \vee G) \\
fac(r_2) &= (C \vee G)
\end{aligned}
$$

So the learned clause according to the first UIP scheme is $c_9\colon (C \vee G)$.

(c) Add the learned clause, apply conflict-driven backtracking and draw the resulting implication graph.

Solution:
For conflict-driven backtracking, we backtrack to the second highest DL in the learned clause, i.e., we backtrack to $DL = 1$. For this kind of backtracking, we keep all decisions on $DL = 1$ but delete all others with $DL > 1$. After BCP the resulting implication graph is as in Figure 5.

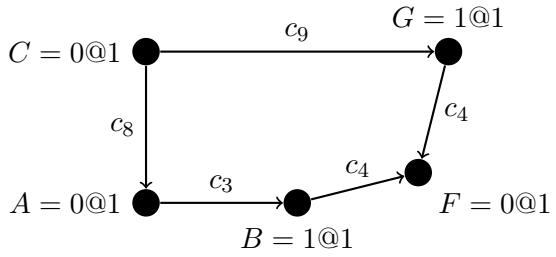


Figure 5: Implication Graph for $\mathcal{C}$ with learned clause c_9 after conflict-driven backtracking and BCP.

Exercise 4 Sparse Method

Apply the Sparse Method including preprocessing on the formula φ below to obtain a propositional formula. Note that φ is not yet in NNF (Negation Normal Form).

$$(x_1 = x_2 \rightarrow x_2 = x_3) \wedge \big[\neg(x_2 = x_4 \vee x_3 \neq x_4 \vee x_4 \neq x_5) \vee (x_6 \neq x_5 \wedge x_6 = x_7 \wedge x_7 = x_3)\big]$$

Solution:
In the first step, we transform φ into NNF. We substitute $\rightarrow$ and apply DeMorgan to obtain φ^E, which now is in NNF:

$$(x_1 \neq x_2 \vee x_2 = x_3) \wedge \big[(x_2 \neq x_4 \wedge x_3 = x_4 \wedge x_4 = x_5) \vee (x_6 \neq x_5 \wedge x_6 = x_7 \wedge x_7 = x_3)\big]$$

Then, we draw the equality graph $G^E(\varphi^E)$ of φ^E, given in Figure 6. Dashed lines represent equality edges while solid lines represent disequality edges.
The edge (x_1, x_2) is not part of a simple contradictory cycle, therefore we set $(x_1 \neq x_2)$ to true and obtain φ_2^E:

$$(\mathit{true} \vee x_2 = x_3) \wedge \big[(x_2 \neq x_4 \wedge x_3 = x_4 \wedge x_4 = x_5) \vee (x_6 \neq x_5 \wedge x_6 = x_7 \wedge x_7 = x_3)\big]$$

Propositional simplification yields φ_3^E:

$$\big[(x_2 \neq x_4 \wedge x_3 = x_4 \wedge x_4 = x_5) \vee (x_6 \neq x_5 \wedge x_6 = x_7 \wedge x_7 = x_3)\big]$$

The equality graph $G^E(\varphi_3^E)$ then is as shown in Figure 7.

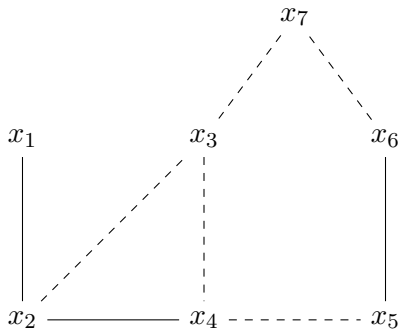


Figure 6: Equality graph $G^E(\varphi^E)$, dashed lines represent equality, solid lines disequality.

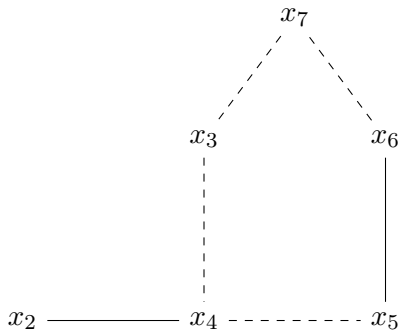


Figure 7: Equality graph $G^E(\varphi_3^E)$, dashed lines represent equality, solid lines disequality.

Edge (x_2, x_4) now is not in a simple contradictory cycle, therefore we set $(x_2 \neq x_4)$ to true and apply propositional simplification to obtain φ_4^E:

$$\big[(x_3 = x_4 \wedge x_4 = x_5) \vee (x_6 \neq x_5 \wedge x_6 = x_7 \wedge x_7 = x_3)\big]$$

The equality graph $G^E(\varphi_4^E)$ is given in Figure 8.
All edges of $G^E(\varphi_4^E)$ are part of a simple contradictory cycle, so we stop with preprocessing and build the propositional skeleton $e(\varphi_4^E)$:

$$(e_{3,4} \wedge e_{4,5}) \vee (\neg e_{5,6} \wedge e_{6,7} \wedge e_{3,7})$$

For transitivity contraints B_t we make the nonpolar equality graph $G_{NP}^E(\varphi_4^E)$ chordal as shown in Figure 9. Observe that edges (x_4, x_7) and (x_5, x_7) are introduced.
The according transitivity constraints B_t are then:

$$\begin{aligned}
&(e_{3,4} \wedge e_{4,7} \rightarrow e_{3,7}) \wedge (e_{4,7} \wedge e_{3,7} \rightarrow e_{3,4}) \wedge (e_{3,7} \wedge e_{3,4} \rightarrow e_{4,7}) \wedge \\
&(e_{4,5} \wedge e_{5,7} \rightarrow e_{4,7}) \wedge (e_{5,7} \wedge e_{4,7} \rightarrow e_{4,5}) \wedge (e_{4,7} \wedge e_{4,5} \rightarrow e_{5,7}) \wedge \\
&(e_{5,6} \wedge e_{6,7} \rightarrow e_{5,7}) \wedge (e_{6,7} \wedge e_{5,7} \rightarrow e_{5,6}) \wedge (e_{5,7} \wedge e_{5,6} \rightarrow e_{6,7})
\end{aligned}$$

The resulting formula in propositional logic then is $e(\varphi_4^E) \wedge B_t$.

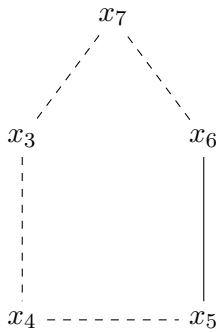


Figure 8: Equality graph $G^E(\varphi_4^E)$, dashed lines represent equality, solid lines disequality.

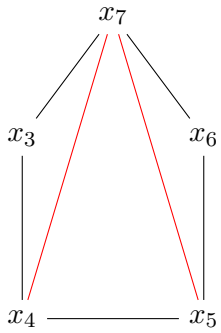


Figure 9: Nonpolar equality graph $G^E_{NP}(\varphi_4^E)$, made chordal by additional edges (in red).

Exercise 5 Ackermann's Reduction

Apply Ackermann's reduction on the following EUF-formula φ to obtain an E-formula:

$$F(F(x_1)) \neq F(x_1) \wedge G(x_1, x_2) = F(x_2) \wedge F(G(x_2, F(x_2))) \neq F(F(x_1))$$

Solution:
We first number the instances of the UFs inwards-to-outwards, left-to-right:

$$F_2(F_1(x_1)) \neq F_1(x_1) \wedge G_1(x_1, x_2) = F_3(x_2) \wedge F_4(G_2(x_2, F_3(x_2))) \neq F_2(F_1(x_1))$$

This already gives $\mathcal{T}$ for the numbered instances. For example:

$$\begin{aligned}
\mathcal{T}(F_1(x_1)) &= f_1 \\
\mathcal{T}(F_2(F_1(x_1))) &= f_2 \\
\mathcal{T}(F_3(x_2)) &= f_3 \\
\mathcal{T}(F_4(G_2(x_2, F_3(x_2)))) &= f_4 \\
\mathcal{T}(G_1(x_1, x_2)) &= g_1 \\
\mathcal{T}(G_2(x_2, F_3(x_2))) &= g_2
\end{aligned}$$

So $flat^E := f_2 \neq f_1 \wedge g_1 = f_3 \wedge f_4 \neq f_2$.
Based on $\mathcal{T}$ we construct $FC^E :=$

$$\begin{aligned}
(x_1 = f_1 \rightarrow & f_1 = f_2) \wedge \\
(x_1 = x_2 \rightarrow & f_1 = f_3) \wedge \\
(x_1 = g_2 \rightarrow & f_1 = f_4) \wedge \\
(f_1 = x_2 \rightarrow & f_2 = f_3) \wedge \\
(f_1 = g_2 \rightarrow & f_2 = f_4) \wedge \\
(x_2 = g_2 \rightarrow & f_3 = f_4) \wedge \\
((x_1 = x_2 \wedge x_2 = f_3) \rightarrow & g_1 = g_2)
\end{aligned}$$

Finally $\varphi^E := FC^E \rightarrow flat^E$.

2 Proofs and Properties

Exercise 6 First-Order Theories

In the lecture, we discussed reasoning under different theories. Here we are concerned with LISP-like lists and the theory $\mathcal{T}^E_{cons} = \mathcal{T}_{cons} \cup \mathcal{T}_E$. In a verification attempt of some program, we have to prove the following:

> *For non-atomic lists ℓ_1, ℓ_2, if the "car" of both lists are equal and the "cdr" of both lists are equal, then ℓ_1 is equal to ℓ_2.*

We formalize the above statement as follows:

$$\varphi\colon \quad \big[\, \neg atom(\ell_1) \wedge \neg atom(\ell_2) \wedge car(\ell_1) \doteq car(\ell_2) \wedge cdr(\ell_1) \doteq cdr(\ell_2) \,\big] \;\rightarrow\; \ell_1 \doteq \ell_2$$

Prove the statement $\mathcal{T}^E_{cons}$-valid, i.e., show that $\mathcal{T}^E_{cons} \models \varphi$.

Hint: Besides the equality axioms reflexivity, symmetry and transitivity, the following axioms from $\mathcal{T}^E_{cons}$ are sufficient for a proof:

(1) Substitution axioms (functional congruence) for *cons*:

$$\forall x_1 \forall x_2 \forall y_1 \forall y_2 \, [(x_1 \doteq x_2 \wedge y_1 \doteq y_2) \rightarrow cons(x_1, y_1) \doteq cons(x_2, y_2)]$$

(2) Construction:

$$\forall x \, [\neg atom(x) \rightarrow cons(car(x), cdr(x)) \doteq x]$$

Solution:
The proof is by contradiction. Suppose there exists a $\mathcal{T}^E_{cons}$-interpretation I with $I \not\models \varphi$.

1.	$I \not\models \varphi$	assumption
2.	$I \models car(\ell_1) \doteq car(\ell_2)$	1., $\rightarrow$, $\wedge$
3.	$I \models cdr(\ell_1) \doteq cdr(\ell_2)$	1., $\rightarrow$, $\wedge$
4.	$I \models \neg atom(\ell_1)$	1., $\rightarrow$, $\wedge$
5.	$I \models \neg atom(\ell_2)$	1., $\rightarrow$, $\wedge$
6.	$I \not\models \ell_1 \doteq \ell_2$	1., $\rightarrow$
7.	$I \models cons(car(\ell_1), cdr(\ell_1)) \doteq cons(car(\ell_2), cdr(\ell_2))$	2., 3., functional congruence
8.	$I \models cons(car(\ell_1), cdr(\ell_1)) \doteq \ell_1$	4., construction
9.	$I \models cons(car(\ell_2), cdr(\ell_2)) \doteq \ell_2$	5., construction
10.	$I \models \ell_1 \doteq \ell_2$	7., 8., 9., symmetry + transitivity
11.	$I \models \bot$	6., 10.

The assumption is false: φ is therefore $\mathcal{T}^E_{cons}$-valid.

Exercise 7 Tseitin Transformation

In the first part of this exercise, we consider a restriction of the Tseitin transformation where the input formula is only composed of propositional variables, negation, and conjunction. In the second part, we consider a simplified transformation whose output is not in CNF.

(a) Let ψ be a propositional formula and let $\delta(\psi)$ be the set of clauses resulting from Tseitin's transformation on ψ. Prove that the following holds:

If ψ is satisfiable then $\delta(\psi)$ is satisfiable.

You only need to prove this for the connectives $\wedge$ and $\neg$. Use the below clause schemes, which introduce a new label for every boolean variable.

$$\begin{array}{rrrr} L_a \leftrightarrow a & (\neg L_a \vee a) & (L_a \vee \neg a) & \\ L_\phi \leftrightarrow (L_1 \wedge L_2) & (\neg L_\phi \vee L_1) & (\neg L_\phi \vee L_2) & (L_\phi \vee \neg L_1 \vee \neg L_2) \\ L_\phi \leftrightarrow \neg L_1 & (\neg L_\phi \vee \neg L_1) & (L_\phi \vee L_1) & \end{array}$$

Solution:
Let $\delta(\psi)$ be the set of all clauses from the labelling of ψ and the additional clause (L_ψ), i.e., $\delta(\psi) = \hat{\delta}(\psi) \cup \{(L_\psi)\}$.

We have to show: If ψ is satisfiable then $\delta(\psi)$ is satisfiable. In other words: If there exists $I \in Mod(\psi)$ then there exists $I' \in Mod(\delta(\psi))$, that is for every $C \in \delta(\psi)$ holds $I'(C) = 1$.

To prove this statement, we assume that ψ is satisfiable. Then we have to show that for some interpretation I' of $\delta(\psi)$ it holds that all clauses $C \in \delta(\psi)$ evaluate to true, i.e., $\forall C \in \delta(\psi) : I'(C) = 1$.

As we assumed ψ to be satisfiable, there exists a model I of ψ. We extend I to an interpretation I' for $\delta(\psi)$ as follows:

i) $I'(a) = I(a)$ for every propositional variable a occuring in ψ.

ii) $I'(L_\phi) = I(\phi)$ for every subformula occurrence ϕ of ψ, i.e., $\phi \in \Sigma(\psi)$, where L_ϕ is the label assigned to ϕ.

It remains to show that I' is also a model of $\delta(\psi)$.

For the following proof we assume without further notice that ϕ is a subformula occurrence of ψ, i.e., $\phi \in \Sigma(\psi)$.

As every clause in $\delta(\psi) \setminus \{(L_\psi)\}$ results from the translation of one subformula occurrence ϕ of ψ, we first show by structural induction on ψ that, for all $C \in \delta(\psi) \setminus \{(L_\psi)\}$, it that holds $I'(C) = 1$. The Induction Hypothesis (IH) which we use is as follows:

IH: If ϕ' is a proper subformula of ϕ (i.e, $\phi' \neq \phi$) then I' satisfies all clauses in $\hat{\delta}(\psi) = \delta(\psi) \setminus \{(L_\psi)\}$ that stem from the translation of ϕ'.

- Base case: $\phi = a$ where a is propositional variable. The clauses in $\delta(\psi)$ steming from the translation of ϕ are $(\neg L_a \vee a)$ and $(L_a \vee \neg a)$. To show that they evaluate to true under I' consider all cases for $I(a)$:
  - $I(a) = 1$: then $I'(a) = 1$ by i) and $I'(L_a) = 1$ by ii), thus $I'(\neg L_a \vee a) = 1$ and $I'(L_a \vee \neg a) = 1$.

- $I(a) = 0$: then $I'(a) = 0$ by i) and $I'(L_a) = 0$ by ii), thus $I'(\neg L_a \vee a) = 1$ and $I'(L_a \vee \neg a) = 1$.

  Therefore all clauses for $\phi = a$ are satisfied by I'.

- Induction step: case $\phi = \phi_1 \wedge \phi_2$. The clauses are $(\neg L_\phi \vee L_1)$, $(\neg L_\phi \vee L_2)$, $(L_\phi \vee \neg L_1 \vee \neg L_2)$ where the label for ϕ_1 is L_1, respectively for ϕ_2 is L_2.
  We consider all cases for $I(\phi)$:
  - $I(\phi) = 1$: thus $I(\phi_1) = I(\phi_2) = 1$ by the semantics of $\wedge$, so $I'(L_1) = I'(L_2) = 1$ by ii) as well as $I'(L_\phi) = 1$. Therefore $I'(\neg L_\phi \vee L_1) = I'(\neg L_\phi \vee L_2) = I'(L_\phi \vee \neg L_1 \vee \neg L_2) = 1$.
  - $I(\phi) = 0$: thus $I(\phi_1) = 0$ or $I(\phi_2) = 0$. Without loss of generality we assume $I(\phi_1) = 0$. Thus $I'(L_\phi) = I'(L_2) = 0$. Therefore $I'(\neg L_\phi \vee L_1) = I'(\neg L_\phi \vee L_2) = I'(L_\phi \vee \neg L_1 \vee \neg L_2) = 1$.

  As all clauses for ϕ_1 and ϕ_2 are satisfied by I' according IH, it follows that all clauses for $\phi = \phi_1 \wedge \phi_2$ are satisfied by I'.

- Induction step: case $\phi = \neg\phi_1$. The clauses are $(\neg L_\phi \vee \neg L_1), (L_\phi \vee L_1)$ where L_1 is the label for ϕ_1.
  We consider all cases for $I(\phi)$:
  - $I(\phi) = 1$: thus $I(\phi_1) = 0$ and by ii) is $I'(L_\phi) = 1$ and $I'(L_1) = 0$. Therefore $I'(\neg L_\phi \vee \neg L_1) = I'(L_\phi \vee L_1) = 1$.
  - $I(\phi) = 0$: thus $I(\phi_1) = 1$ and by ii) is $I'(L_\phi) = 0$ and $I'(L_1) = 1$. Therefore $I'(\neg L_\phi \vee \neg L_1) = I'(L_\phi \vee L_1) = 1$.

  As all clauses for ϕ_1 are satisfied by I' according to IH, all clauses for $\phi = \neg\phi_1$ are satisfied by I'.

The only remaining clause not covered by structural induction is (L_ψ) where L_ψ is the label assigned to ψ. As $I \in Mod(\psi)$ holds $I(\psi) = 1$ and thus by ii) holds $I'(L_\psi) = 1$.

Therefore all clauses are satisfied by I' and we have proven: if ψ is satisfiable then $\delta(\psi)$ is satisfiable. □

Shorter Alternative: One can show that the clauses for the cases $\phi = a$ and $\phi = \neg\phi_1$ evaluate to true in shorter terms. Instead of the case distinction for $I(\phi)$, directly use the relationship between ϕ and its assigned label, as shown in the following:

- Case $\phi = a$: by ii) $I'(a) = I'(L_a)$ therefore $I'(\neg L_a \vee a) = 1 - I'(L_a) + I'(L_a) = 1$ and $I'(L_a \vee \neg a) = I'(L_a) + 1 - I'(L_a) = 1$, so all clauses are satisfied.
- Case $\phi = \neg\phi_1$: by ii) and the semantics of negation it holds that $I'(L_\phi) = 1 - I'(L_1)$ therefore $I'(\neg L_\phi \vee \neg L_1) = 1 - (1 - I'(L_1)) + 1 - I'(L_1) = 1$ and $I'(L_\phi \vee L_1) = 1 - I'(L_1) + I'(L_1) = 1$. As the clauses for ϕ_1 are satisfied by IH, all clauses for ϕ are satisfied.

Notice: The rest of the proof (assumption I that is a model, induction hypothesis, etc.) remains the same.

(b) Consider a simplified variant of Tseitin's transformation: let φ be a propositional formula, let $\Sigma(\varphi)$ be the set of all subformulas of φ, and let ℓ_φ be the label for φ. Then, the result of simplified Tseitin's transformation is the formula:

$$\lambda = \left(\bigwedge_{\psi \in \Sigma(\varphi)} (\ell_\psi \leftrightarrow \psi) \right) \rightarrow \ell_\varphi$$

Prove: λ is valid if and only if φ is valid.

Solution:

"$\varphi \Rightarrow \lambda$": We show, if φ is valid then λ is valid. If φ is valid, then for every interpretation I holds that $I(\varphi) = 1$.

Assume for contradiction that there exists an interpretation I' such that $I'(\lambda) = 0$. By the semantics of $\rightarrow$ it follows that $I'(\bigwedge_{\psi \in \Sigma(\varphi)} (\ell_\psi \leftrightarrow \psi)) = 1$ and $I'(\ell_\varphi) = 0$. By the semantics of $\wedge$ it holds for every $\psi \in \Sigma(\varphi)$ that $I'(\ell_\psi \leftrightarrow \psi) = 1$, specifically for $\psi = \varphi$. Hence, $I'(\ell_\varphi) = 0 = I'(\varphi)$.

Consider the projection I'' of I' to those propositional variables occurring only in φ, then $I''(\varphi) = 0$. This contradicts that φ is valid. Therefore, no such interpretation I' with $I'(\lambda) = 0$ exists, i.e., λ is valid.

"$\lambda \Rightarrow \varphi$": We show, if λ is valid then φ is valid.

Assume for contradiction that there exists an interpretation I such that $I(\varphi) = 0$. Consider the extension of I' of I such that $I'(\psi) = I'(\ell_\psi)$ for all $\psi \in \Sigma(\varphi)$. Obviously, I' is an interpretation over λ, hence $I'(\lambda) = 1$. Since $I(\varphi) = 0$, it holds that $I'(\ell_\varphi) = 0$, therefore by the semantics of $\rightarrow$ it must hold that $I'(\bigwedge_{\psi \in \Sigma(\varphi)} (\ell_\psi \leftrightarrow \psi)) = 0$.

Since $I'(\psi) = I'(\ell_\psi)$ for all $\psi \in \Sigma(\varphi)$, it holds that $I'(\bigwedge_{\psi \in \Sigma(\varphi)} (\ell_\psi \leftrightarrow \psi)) = 1$, which is a contradiction. Therefore, no such interpretation I with $I(\varphi) = 0$ exists, i.e., φ is valid.

Exercise 8 Implication Graphs

(a) Show that in a conflict graph the first UIP is uniquely defined, i.e., there is exactly one node in the implication graph which is a first UIP.

Solution:

Proof by contradiction: Assume there are two nodes v, v' where both v and v' are first-UIPs. Let d be the node of the last decision and k the conflict node.

A UIP is by definition a node where all paths from d to k go through. As v and v' are first-UIPs, they both are UIPs, so all paths from d to k go through v and also through v'.

Therefore there either is a path $d, \ldots, v, \ldots, v', \ldots, k$ from v to v' or there is a path from v' to v. Without loss of generality, let the path be from v to v'. As all paths from d to k go through v and v', all paths are of form $d, \ldots, v, \ldots, v', \ldots, k$, because the implication graph is acyclic.

As $v \neq v'$ the distance $d(v', k)$ between v' and k is smaller than the distance $d(v, k)$, i.e., $d(v', k) < d(v, k)$. But this contradicts the assumption that v is a first-UIP, because v' is closer to the conflict k than v.

Therefore there can be only one first UIP.

As d, the current decision node, is always a UIP, there always exists a at least one UIP, hence there also exists a UIP closest to the conflict, i.e., there exists a first UIP.

(b) Let $\mathcal{C}$ be a set of clauses and G a conflict graph with respect to $\mathcal{C}$. Prove: if C_l is the first clause that is learned following the first-UIP scheme, then C_l is a consequence of $\mathcal{C}$.

Bonus questions: how can this statement be used to show that all clauses that are learned (following the first-UIP scheme) are a consequence of $\mathcal{C}$?

Solution:
Consider how a new clause is learnt: Find the first-UIP u and resolve with clauses from the conflict k to u. Let $S \subseteq \mathcal{C}$ denote those clauses that occur as edge-labels in the implication graph G from the first UIP u to the conflict node k.

As C_l is learnt following the first UIP schema, there is a resolution derivation $K_1, K_2, \ldots, K_n$ of C_l from S where $K_n = C_l$ and for each K_ℓ holds: either $K_\ell \in S$ or K_ℓ is the resolvent of two K_i and K_j with $i, j < \ell$ and $1 \leq \ell \leq n$. As resolution is correct it follows that $S \models C_l$.

By monotonicity of propositional logic it then follows that $F \cup S \models C_l$ for any set of formulas F, specifically $\mathcal{C} \cup S \models C_l$. And as $S \subseteq \mathcal{C}$ it follows that $\mathcal{C} \models C_l$.

Bonus question: For a sequence $C_1, \ldots, C_n$ of learned clauses, we can inductively apply the statement as follows. C_1 is a consequence of $\mathcal{C}$ by the above proof. C_2 is a consequence of $\mathcal{C} \cup \{C_1\}$ again by the above proof; since C_1 is a consequence of $\mathcal{C}$ it therefore holds by transitivity of the consequence relation that C_2 also is a consequence of $\mathcal{C}$. The proof sketch for C_3 to C_n then is analogous. Note that this is not a full inductive proof, but just a quick and incomplete proof sketch.

(c) Prove: *During the run of a SAT solver, the implication graph G_k at step k is acyclic.*

Hints:

1) Perform a proof by induction over k.
2) Consider the following events that can occur:
   (i) making a decision,

(ii) unit propagation (one step of BCP),

(iii) a clause is unsatisfiable,

(iv) backtracking.

Solution:
Let $P(n)$ be the property that G_n is acyclic.

Base case, $n = 0$: G_0 is the empty graph, hence it is acyclic. Therefore $P(0)$ holds.

Induction Hypothesis: Let n be an integer ≥ 0 and suppose $P(0), \dots P(n)$ is true.

Step: Consider $P(n+1)$ which we have to show to be true. By (IH) it holds that $G_i = (V_i, E_i)$ is acyclic $(0 \leq i \leq n)$. We continue by a case distinction wrt. all possible steps of the SAT solver.

(i) "Making a decision": wlog. let the decision be $X = f@d$. Then $V_{n+1} = V_n \cup \{X\}$ and $E_{n+1} = E_n$. Since G_n is acyclic, $G_{n+1} = (V_{n+1}, E_{n+1})$ also is acyclic.

(ii) "A clause is unsatisfiable": wlog. let the unsatisfiable clause be $(\ell_1 \vee \dots \vee \ell_m)$, then $V_{n+1} = V_n \cup \{\kappa\}$ and $E_{n+1} = E_n \cup \{(\ell_j, \kappa) \mid 1 \leq j \leq m\}$. Note that no new edge has been added other than those going to κ. Hence G_{n+1} only contains a cycle, if G_n contains a cycle. Therefore it follows by (IH) that G_{n+1} is acyclic.

(iii) "Unit propagation": wlog. let the unit clause be $(\ell_1 \vee \dots \vee \ell_m)$ and let ℓ_m be the unassigned variable. Then $V_{n+1} = V_n \cup \{\ell_i\}$ and $E_{n+1} = E_n \cup \{(\ell_j, \ell_m) \mid 1 \leq j \leq m-1\}$. Since all added edges go to the newly added node ℓ_m, it follows by (IH) that G_{n+1} is acyclic.

(iv) "Backtracking": since backtracking only removes a part of the implication graph $G_n = (V_n, E_n)$, we have that $G_{n+1} = (V_n \setminus V', E_n \setminus E')$ for some sets $V' \subset V_n, E' \subset E_n$. Removing edges can not make a graph cyclic. Therefore, by (IH) follows that G_{n+1} is acyclic.

Since $P(n+1)$ holds for any step under the assumption that (IH) holds, it therefore follows that $P(n)$ holds for any $n \geq 0$. Consequently, the implication graph G_k at step k is acyclic.

Exercise 9 Ackermann's Reduction

(a) The removal of Boolean variables from an E-formula is defined as follows:

Definition. Let φ^E be any E-formula with Boolean variables $b_1, \dots, b_n$. Construct an E-formula ψ^E without any Boolean variable by replacing each b_i by $v_{b_{i,1}} \doteq v_{b_{i,2}}$ where $v_{b_{i,1}}$, $v_{b_{i,2}}$ are two new term variables (identifiers).

Prove that φ^E is E-satisfiable iff ψ^E is E-satisfiable.

Solution:
For a full proof see the extra sheet (extra-sheet4-1a.pdf) in the BACKGROUND directory in TUWEL. In the exam, a proof sketch as follows is also sufficient:

Let $NV = \bigcup_{i=1}^{n} \{v_{b_{i,1}}, v_{b_{i,2}}\}$ be the set of term variables in ψ^E that do not occur in φ^E.

($\varphi^E \to \psi^E$): Let $M \in Mod(\varphi^E)$. Consider the interpretation M' where $M'(x \doteq y) = M(x \doteq y)$ if $x, y \notin NV$, and $M'(v_{b_{i,1}} \doteq v_{b_{i,2}}) = M(b_i)$ otherwise. Observe that ψ^E contains no $x \doteq y$ where $x \in NV$ and $y \notin NV$ or where $x \notin NV$ and $y \in NV$, since $v_{b_{i,1}}$ and $v_{b_{i,2}}$ are new term variables. Then, it follows by the Equivalent Replacement Lemma that $M' \in Mod(\psi^E)$.

($\varphi^E \leftarrow \psi^E$): Let $M' \in Mod(\psi^E)$, consider the interpretation M where $M'(x \doteq y) = M(x \doteq y)$ if $x, y \notin NV$ and $M(b_i) = M'(v_{b_{i,1}} \doteq v_{b_{i,2}})$. Again, by the Equivalent Replacement Theorem, we conclude that $M \in Mod(\varphi^E)$.

(b) Transform the EUF-formula φ^{EUF} below to an E-formula φ^E using Ackermann's reduction. Note that φ^{EUF} contains an uninterpreted predicate, which requires special treatment first.

$$\varphi^{EUF}: \quad F(F(x_1)) \doteq G(x_2, G(x_1, x_3, x_4), F(x_2)) \to p(x_1, y).$$

Solution:
First, we replace the predicate and obtain:

$$\varphi': \quad F(F(x_1)) \doteq G(x_2, G(x_1, x_3, x_4), F(x_2)) \to H_p(x_1, y) \doteq x_p.$$

Labelling function occurrences inside-out yields:

$$\varphi'': \quad F_2(F_1(x_1)) \doteq G_2(x_2, G_1(x_1, x_3, x_4), F_3(x_2)) \to H_p(x_1, y) \doteq x_p.$$

The propositional skeleton is:

$$flat^E: \quad f_2 \doteq g_2 \to h_p \doteq x_p.$$

The functional constraints FC^E are:

$$\begin{aligned} x_1 \doteq f_1 &\to f_1 \doteq f_2 \\ x_1 \doteq x_2 &\to f_1 \doteq f_3 \\ f_1 \doteq x_2 &\to f_2 \doteq f_3 \\ (x_2 \doteq x_1 \wedge g_1 \doteq x_3 \wedge f_3 \doteq x_4) &\to g_1 \doteq g_2 \end{aligned}$$

Note that there are no constraints for h_p since it only occurs once in φ'. Finally, φ^E is $FC^E \to flat^E$.