Automated Theorem Proving

Prof. Dr. Jasmin Blanchette, Yiming Xu, PhD,
Tanguy Bozec, and Lydia Kondylidou
based on exercises by Dr. Uwe Waldmann

Winter Term 2025/26

Exercises 3: Propositional Logic Continued

Exercise 3.1 ($*$)**:** Let $N = \{C_1, \ldots, C_n\}$ be a finite set of propositional clauses without duplicated literals or complementary literals such that for every $i \in \{1, \ldots, n\}$ the clause C_i has exactly i literals. Prove or refute: N is satisfiable.

Proposed solution. The property holds. We show by induction that for every $j \in \{0, \ldots, n\}$ there is a partial valuation $\mathcal{A}_j$ that satisfies $C_1, \ldots, C_j$ and in which exactly j atoms are defined.

If $j = 0$ the statement is trivial: Define $\mathcal{A}_0$ as the valuation that is undefined for every propositional variable. If $0 < j \leq n$, we assume by induction that the statement holds for $j - 1$; so there exists a partial valuation $\mathcal{A}_{j-1}$ that satisfies $C_1, \ldots, C_{j-1}$ and in which exactly $j - 1$ atoms are defined. As C_j contains j literals that are different and noncomplementary, C_j must contain j different atoms. Since only $j - 1$ atoms are defined in $\mathcal{A}_{j-1}$, there exists at least one atom P in C_j that is undefined in $\mathcal{A}_{j-1}$. Now define $\mathcal{A}_j$ as the valuation that maps P to 1 if P occurs positively in C_j, or to 0 if P occurs negatively in C_j, and that interprets every other atom Q in the same way as $\mathcal{A}_{j-1}$. Since all atoms that are defined in $\mathcal{A}_{j-1}$ are defined in the same way in $\mathcal{A}_j$, $\mathcal{A}_j$ satisfies $C_1, \ldots, C_{j-1}$; moreover $\mathcal{A}$ satisfies C_j since it interprets P appropriately.

Exercise 3.2: Let N be a set of propositional clauses. Prove or refute the following statement: If N contains clauses $C_i \vee D_i$ ($i \in \{1, \ldots, n\}$) such that $\{C_i \mid i \in \{1, \ldots, n\}\} \models \bot$, then $N \models \bigvee_{i \in \{1,\ldots,n\}} D_i$.

Proposed solution. The statement holds. Proof: Suppose that N contains clauses $C_i \lor D_i$ ($i \in \{1,\dots,n\}$) such that $\{C_i \mid i \in \{1,\dots,n\}\} \models \bot$. Let $\mathcal{A}$ be an arbitrary model of N. We have to show that $\mathcal{A} \models \bigvee_{i\in\{1,\dots,n\}} D_i$. Assume otherwise. Then $\mathcal{A}(\bigvee_{i\in\{1,\dots,n\}} D_i) = 0$ and therefore $\mathcal{A}(D_i) = 0$ for every $i \in \{1,\dots,n\}$. On the other hand, $\mathcal{A}$ is a model of every clause in N, and therefore $\mathcal{A}(C_i \lor D_i) = 1$ for every $i \in \{1,\dots,n\}$. Consequently, $\mathcal{A}(C_i) = 1$ for every $i \in \{1,\dots,n\}$. This is impossible, however, since $\{C_i \mid i \in \{1,\dots,n\}\}$ is unsatisfiable.

Exercise 3.3: A partial Π-valuation $\mathcal{A}$ under which all clauses of a clause set N are true is called a partial Π-model of N.

Do the following clause sets over $\Pi = \{P, Q, R\}$ have partial Π-models that are not total Π-models (that is, models in the sense of Sect. 2.3)? If yes, give such a partial Π-model.

(1)
$$\begin{array}{ccccc} P & & & & \\ \neg P & \lor & Q & & \\ \neg P & \lor & \neg Q & \lor & \neg R \end{array}$$

(2)
$$\begin{array}{ccccc} P & & & & \\ \neg P & \lor & Q & & \\ & & \neg Q & \lor & R \\ \neg P & \lor & \neg Q & \lor & \neg R \end{array}$$

(3)
$$\begin{array}{ccccc} P & & & \lor & R \\ \neg P & \lor & Q & \lor & \neg R \\ & & \neg Q & \lor & \neg R \end{array}$$

(4)
$$\begin{array}{ccccc} \neg P & \lor & Q & & \\ & & \neg Q & \lor & R \\ P & & & \lor & \neg R \end{array}$$

Proposed solution. (1) No. The only model of the clauses is the total model $\mathcal{A}$ with $\mathcal{A}(P) = \mathcal{A}(Q) = 1$ and $\mathcal{A}(R) = 0$.

(2) No. The clause set is unsatisfiable.

(3) Yes. The partial valuation $\mathcal{A}$ with $\mathcal{A}(P) = 1$ and $\mathcal{A}(R) = 0$ is a partial model.

(4) No. The only models are total valuations $\mathcal{A}$ such that $\mathcal{A}(P) = \mathcal{A}(Q) = \mathcal{A}(R)$.

Exercise 3.4: For any propositional formula F, let $\mathit{negvar}(F)$ be the formula obtained from F by replacing every propositional variable by its negation. Formally:

$$\begin{aligned} \mathit{negvar}(P) &= \neg P \\ \mathit{negvar}(\neg G) &= \neg \mathit{negvar}(G) \\ \mathit{negvar}(G_1 \wedge G_2) &= \mathit{negvar}(G_1) \wedge \mathit{negvar}(G_2) \end{aligned}$$

and so on. For example, $\mathit{negvar}(P \vee (\neg Q \rightarrow (\neg P \wedge \top))) = \neg P \vee (\neg\neg Q \rightarrow (\neg\neg P \wedge \top))$.

Prove or refute: If a formula F is satisfiable, then $\mathit{negvar}(F)$ is satisfiable. (It is sufficient if you consider the boolean connectives $\neg$ and $\wedge$; the others are treated analogously.)

Proposed solution. Assume F is satisfiable. Let the valuation $\mathcal{A}$ be a model of F. We define a valuation $\mathcal{A}'$ by $\mathcal{A}'(P) = 1 - \mathcal{A}(P)$ for every propositional variable P.

Now we can show by induction over the structure of formulas that $\mathcal{A}'(\mathit{negvar}(G)) = \mathcal{A}(G)$ for every formula G:

Case 1: G is a propositional variable. If $G = P$, then $\mathcal{A}'(\neg P) = 1 - \mathcal{A}'(P) = 1 - (1 - \mathcal{A}(P)) = \mathcal{A}(P)$.

Case 2: G is a negation $\neg G_1$. We must show $\mathcal{A}'(\mathit{negvar}(\neg G_1)) = \mathcal{A}(\neg G_1)$. We have $\mathcal{A}'(\mathit{negvar}(\neg G_1)) = \mathcal{A}'(\neg \mathit{negvar}(G_1)) = 1 - \mathcal{A}'(\mathit{negvar}(G_1))$. By induction, this equals $1 - \mathcal{A}(G_1)$, i.e., $\mathcal{A}(\neg G_1)$, as desired.

Case 3: G is a conjunctive formula $G_1 \wedge G_2$. We must show $\mathcal{A}'(\mathit{negvar}(G_1 \wedge G_2)) = \mathcal{A}(G_1 \wedge G_2)$. We have $\mathcal{A}'(\mathit{negvar}(G_1 \wedge G_2)) = \mathcal{A}'(\mathit{negvar}(G_1) \wedge \mathit{negvar}(G_2)) = \min\{\mathcal{A}'(\mathit{negvar}(G_1)), \mathcal{A}'(\mathit{negvar}(G_2))\}$. By induction, this equals $\min\{\mathcal{A}(G_1), \mathcal{A}(G_2)\}$, i.e., $\mathcal{A}(G_1 \wedge G_2)$, as desired.

The remaining cases are handled analogously.

Since $\mathcal{A}(F) = 1$, we conclude that $\mathcal{A}'(\mathit{negvar}(F)) = 1$, so $\mathcal{A}'$ is a model of F.

Exercise 3.5: Let N be the following set of propositional clauses over $\Pi = \{P, Q, R\}$:

$$\begin{array}{ccccccc} P & \vee & \neg Q & & & \qquad & (1) \\ & & Q & \vee & \neg R & & (2) \\ \neg P & & & \vee & R & & (3) \end{array}$$

(a) Use the DPLL procedure to compute a (total) model of N.

(b) Use the DPLL procedure to prove that $N \models R \rightarrow P$. Before you can invoke the procedure, you will first need to transform the entailment into a suitable set of clauses.

Proposed solution. (a) We start with the literal set $M := \emptyset$ and the clause set $N := \{(1), (2), (3)\}$. We arbitrarily pick the undefined variable P and set $M := \{P\}$. Then clause (3) contains the unit literal R, so we set $M := \{P, R\}$. Next, clause (2) contains the unit literal Q, so we set $M := \{P, R, Q\}$. At this point, all clauses in N are true in M, so we stop with $M = \{P, R, Q\}$ as the model.

(b) We use the fact that $N \models R \to P$ if and only if $N' = N \cup \{\neg(R \to P)\}$ is unsatisfiable. To use the DPLL procedure, we transform N' into a set of clauses and obtain the new clauses R (4) and $\neg P$ (5). Since (4) contains the unit literal R, we set $M := \{R\}$. Since (5) contains the unit literal $\neg P$, we set $M := \{R, \neg P\}$. Since (1) contains the unit literal $\neg Q$, we set $M := \{R, \neg P, \neg Q\}$ At this point, (2) is false in M, and there is nowhere to backtrack to, so the clause set N' is unsatisfiable.

Exercise 3.6 ($*$)**:** A friend asks you to proofread her bachelor thesis. On page 14 of the thesis, she writes the following:

> **Definition 11.** Let N be a set of propositional formulas. The set $\mathit{poscomb}(N)$ of positive combinations of formulas in N is defined inductively by
> (1) $N \subseteq \mathit{poscomb}(N)$;
> (2) if $F, F' \in \mathit{poscomb}(N)$, then $F \vee F' \in \mathit{poscomb}(N)$; and
> (3) if $F, F' \in \mathit{poscomb}(N)$, then $F \wedge F' \in \mathit{poscomb}(N)$.
>
> **Lemma 12.** If N is a satisfiable set of formulas, then every positive combination of formulas in N is satisfiable.
>
> **Proof.** The proof proceeds by induction over the formula structure. Let $G \in \mathit{poscomb}(N)$. If $G \in N$, then it is obviously satisfiable, since N is satisfiable. Otherwise, G must be a disjunction or a conjunction of formulas in $\mathit{poscomb}(N)$. If G is a disjunction $F \vee F'$ with $F, F' \in \mathit{poscomb}(N)$, we know by the induction hypothesis that F is satisfiable. So F has a model. Since this is also a model of $G = F \vee F'$, the formula G is satisfiable. Analogously, if G is a conjunction $F \wedge F'$, with $F, F' \in \mathit{poscomb}(N)$, then both F and F' are satisfiable by induction, so $G = F \wedge F'$ is satisfiable as well.

(1) Is the "proof" correct?

(2) If the "proof" is not correct:

   (a) Which step is incorrect?

   (b) Does the "lemma" hold? If yes, give a correct proof; otherwise, give a counterexample.

Proposed solution. The "proof" is not correct. The flaw is in the passage "both F and F' are satisfiable by induction, so $G = F \wedge F'$ is satisfiable as well." A counterexample is $F := P$ and $F' := \neg P$. Both F and F' are satisfiable, but their conjunction is not.

Nevertheless, the lemma does hold. Proof: Since N is satisfiable, it has a model $\mathcal{A}$. We claim that $\mathcal{A}$ is also a model of any formula in $\mathit{poscomb}(N)$.

Our proof proceeds by induction over the formula structure. Let $G \in \mathit{poscomb}(N)$. If $G \in N$, then $\mathcal{A}$ is obviously a model of G. Otherwise, G must be a disjunction or a conjunction of formulas in $\mathit{poscomb}(N)$. If G is a disjunction $F \vee F'$ with $F, F' \in \mathit{poscomb}(N)$, we know by the induction hypothesis that $\mathcal{A}$ is a model of F. This is also a model of $G = F \vee F'$. If G is a conjunction $F \wedge F'$, with $F, F' \in \mathit{poscomb}(N)$, then $\mathcal{A}$ is a model of both F and F' by induction, so $\mathcal{A}$ is a model of $G = F \wedge F'$ as well.

Exercise 3.7: The sudoku puzzle presented in the first lecture has a unique solution.

	1	2	3	4	5	6	7	8	9
1								1	
2	**4**								
3		2							
4					5		4		7
5			8				3		
6			1		9				
7	3			4			2		
8		5		1					
9				8		6			

If we replace the 4 in column 1, row 2 by some other digit, this need no longer hold. Use a SAT solver to find out for which values in column 1, row 2 the puzzle has no solution.

Hint: The Perl script at

```
https://rg1-teaching.mpi-inf.mpg.de/autrea-ws23/gensud
```

produces an encoding of the sudoku above in DIMACS CNF format, which is accepted by most SAT solvers.

Proposed solution. The only value in column 1, row 2 for which the puzzle has no solution is 3.

Exercise 3.8 ($*$)**:** Given a sudoku puzzle, briefly describe a set of propositional clauses that is satisfiable if and only if the puzzle has more than one solution.

Proposed solution. The trick is to encode two copies of the sudoku problem, each working on their own sets of propositional variables, and to add constraints specifying that the two copies are not identical. These constraints will need to be converted to CNF.

Exercise 3.9: A finite graph is a pair (V, E), where V is a finite nonempty set and $E \subseteq V \times V$. The elements of V are called vertices or nodes; the elements of E are called edges. A graph has a 3-coloring if there exists a function $\phi : V \to \{0, 1, 2\}$ such that for every edge $(v, v') \in E$ we have $\phi(v) \neq \phi(v')$.

Give a linear-time translation from finite graphs (V, E) to propositional clause sets N such that (V, E) has a 3-coloring if and only if N is satisfiable and such that every model of N corresponds to a 3-coloring ϕ and vice versa.

Proposed solution. Let (V, E) be given, let $C = \{0, 1, 2\}$ be the set of "colors." Let $\Pi = \{P_v^c \mid v \in V,\ c \in C\}$, where P_v^c is supposed to be true in a model if and only if $\phi(v) = c$. Then N is the following set of clauses over Π:

- $\bigvee_{c \in C} P_v^c$ for every $v \in V$ (that is, v is mapped to some $c \in C$ by ϕ).
- $\neg P_v^c \vee \neg P_v^{c'}$ for every $v \in V$ and all $c, c' \in C$ with $c < c'$ (that is, v is not mapped to both c and c').
- $\neg P_v^c \vee \neg P_{v'}^c$ for every edge $(v, v') \in E$ and every $c \in C$ (that is, v and v' are not both mapped to c).

Exercise 3.10 ($*$)**:** A finite graph is a pair (V, E), where V is a finite nonempty set and $E \subseteq V \times V$. The elements of V are called vertices or nodes; the elements of E are called edges. A graph has a 3-coloring if there exists a function $\phi : V \to \{0, 1, 2\}$ such that for every edge $(v, v') \in E$ we have $\phi(v) \neq \phi(v')$. A 3-coloring is called complete if for every pair $(c, c') \in \{0, 1, 2\} \times \{0, 1, 2\}$ with $c \neq c'$ there exists an edge $(v, v') \in E$ such that $\phi(v) = c$ and $\phi(v') = c'$ or $\phi(v) = c'$ and $\phi(v') = c$.

Give a linear-time translation from finite graphs (V, E) to propositional clause sets N such that (V, E) has a complete 3-coloring if and only if N is satisfiable and such that every model of N corresponds to a complete 3-coloring ϕ and vice versa.

Proposed solution. There are several possible translations. We can for instance extend Π and N from the answer to Ex. 3.9 in the following way: Let $\Pi' = \Pi \cup \{Q_{v,v'}^{c,c'} \mid (v,v') \in E,\ c, c' \in C,\ c < c'\}$, where the propositional variable $Q_{v,v'}^{c,c'}$ is supposed to be true in a model only if $\phi(v) = c$ and $\phi(v') = c'$ or $\phi(v) = c'$ and $\phi(v') = c$. Then N' adds the following clauses to N:

- $\bigvee_{(v,v') \in E} Q_{v,v'}^{c,c'}$ for all $c, c' \in C$ with $c < c'$ (that is, at least one edge connects two vertices with colors c and c').
- $\neg Q_{v,v'}^{c,c'} \vee P_v^c \vee P_{v'}^c$ for every edge $(v,v') \in E$ and all $c, c' \in C$ with $c < c'$ (that is, one of v and v' is mapped to c).
- $\neg Q_{v,v'}^{c,c'} \vee P_v^{c'} \vee P_{v'}^{c'}$ for every edge $(v,v') \in E$ and all $c, c' \in C$ with $c < c'$ (that is, one of v and v' is mapped to c').

Exercise 3.11: Give OBDDs for the following three formulas:

(a) $\neg P$

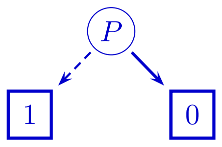
(b) $P \leftrightarrow Q$

(c) $(P \wedge Q) \vee (Q \wedge R) \vee (R \wedge P)$

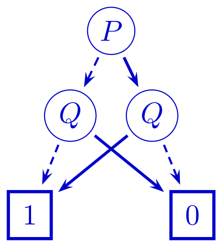
Consider the ordering $P < Q < R$.

Proposed solution.

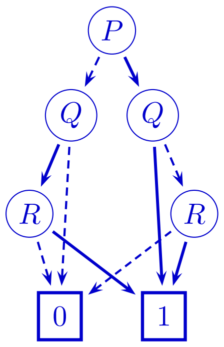
(a) $\neg P$



(b) $P \leftrightarrow Q$



(c) $(P \wedge Q) \vee (Q \wedge R) \vee (R \wedge P)$



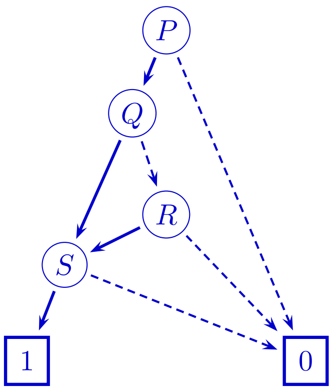
Exercise 3.12: Let F be the propositional formula $P \wedge (Q \vee R) \wedge S$.

(a) Give the reduced OBDD for F w.r.t. the ordering $P < Q < R < S$.

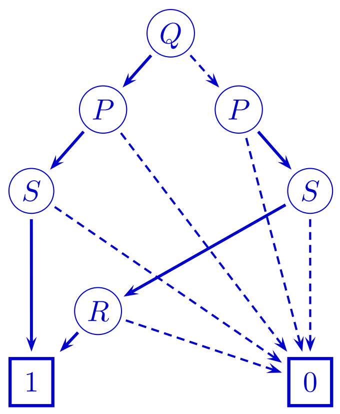
(b) Find a total ordering over $\{P, Q, R, S\}$ such that the reduced OBDD for F has six nonleaf nodes. Give the resulting reduced OBDD.

(c) For how many total orderings over $\{P, Q, R, S\}$ does the reduced OBDD for F have six nonleaf nodes?

Proposed solution. (a) With the ordering $P < Q < R < S$, we obtain the following OBDD for F:

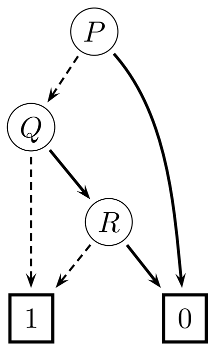


(b) If the two variables that form the inner disjunction, namely Q and R, are the largest and the smallest element of the ordering, the OBDD needs two P-nodes and two S-nodes. E.g., for $Q < P < S < R$, we obtain



(c) By symmetry, we get a similar OBDD as in part (b) if we swap Q and R in the ordering and/or if we swap P and S in the ordering. This yields four different orderings with a six-node OBDD.

Exercise 3.13: (a) Give a propositional formula F that is represented by this reduced OBDD:



(b) How many different reduced OBDDs over the propositional variables $\{P, Q, R\}$ have exactly one nonleaf node?

Proposed solution.

(a) The propositional formula F is $F = \neg P \wedge (\neg Q \vee \neg R)$.

(b) A reduced OBDD with exactly one nonleaf node means there is only one decision node for one of the variables, and it directly connects to the terminal nodes. Given variables $\{P, Q, R\}$, for each variable being the decision node, there are two possible edges from this node, leading to the terminal nodes in two distinct ways. Therefore, for each variable, there are two ways to construct such an OBDD. Since there are 3 variables, and each can be the single nonleaf node, the total number of reduced OBDDs is 6.

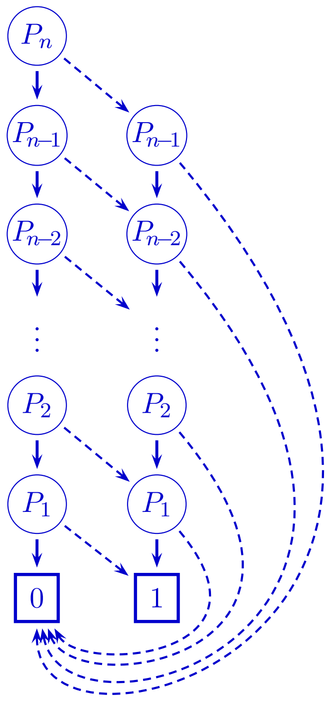
Exercise 3.14 (∗)**:** Let F_n be a propositional formula over $\{P_1, \dots, P_n\}$ such that $\mathcal{A}(F_n) = 1$ if and only if $\mathcal{A}$ maps exactly one of the propositional variables $P_1, \dots, P_n$ to 1 and the others to 0. How many nodes does a reduced OBDD for F_n have (including the leaf nodes $\boxed{0}$ and $\boxed{1}$)?

Proposed solution. To give a recursive definition for F_n, we need an auxiliary formula G_n over $\{P_1, \dots, P_n\}$ such that $\mathcal{A}(G_n) = 1$ if and only if $\mathcal{A}$ maps all propositional variables $P_1, \dots, P_n$ to 0. Then we have

$$\begin{aligned}
F_0 &\models\mid \bot \\
G_0 &\models\mid \top \\
F_n &\models\mid \text{if } P_n \text{ then } G_{n-1} \text{ else } F_{n-1} \\
G_n &\models\mid \text{if } P_n \text{ then } \bot \text{ else } G_{n-1}
\end{aligned}$$

for $n \geq 1$. (The if–then–else construct can be encoded using the usual boolean connectives as shown in the lecture notes.)

The recursive definition can be translated directly into a reduced OBDD: The OBDD has $2n+1$ nodes: one node labeled with P_n (corresponding to the formula F_n), two nodes labeled with P_i for every $i \in \{1, \dots, n-1\}$ (corresponding to F_i and G_i), and two leaf nodes:


P_n
P_{n-1}
P_{n-1}
P_{n-2}
P_{n-2}
P_2
P_2
P_1
P_1
0
1