Automated Deduction - SS 2019

Problem Set 1
Due April 2, 2019, 12 o'clock

Problem 1.1. (10 points) Prove, using splitting, that the formulas $p \leftrightarrow \neg q$ and $\neg p \leftrightarrow q$ are equivalent.

Solution:
We need to prove validity of $\big(p \leftrightarrow \neg q\big) \leftrightarrow \big(\neg p \leftrightarrow q\big)$, using splitting. We do so by proving unsatisfiability of:

$$\neg\Big(\big(p \leftrightarrow \neg q\big) \leftrightarrow \big(\neg p \leftrightarrow q\big)\Big).$$

We start with splitting on p.

Case p **is** $true$. We obtain $\neg\Big(\big(\top \leftrightarrow \neg q\big) \leftrightarrow \big(\neg\top \leftrightarrow q\big)\Big)$, which is simplified to $\neg\Big(\neg q \leftrightarrow \neg q\Big)$.
We split on q. If q is $true$, we get $\neg\Big(\neg\top \leftrightarrow \neg\top\Big)$ which is simplified to $\bot$. We therefore backtrack and check if q is $false$. We then get $\neg\Big(\neg\bot \leftrightarrow \neg\bot\Big)$ and simplify to $\bot$. Hence, the formula is unsat on this splitting path.
Case p **is** $false$. We obtain $\neg\Big(\big(\bot \leftrightarrow \neg q\big) \leftrightarrow \big(\neg\bot \leftrightarrow q\big)\Big)$, which is simplified to $\neg(q \leftrightarrow q)$. Similarly to the previous case, we split on q and derive $\bot$ on both splitting paths, yielding that the formula is unsat also on this splitting path.

In conclusion, $\neg\Big(\big(p \leftrightarrow \neg q\big) \leftrightarrow \big(\neg p \leftrightarrow q\big)\Big)$ is unsat and thus $\big(p \leftrightarrow \neg q\big) \leftrightarrow \big(\neg p \leftrightarrow q\big)$ is valid. That is, $\big(p \leftrightarrow \neg q\big)$ and $\big(\neg p \leftrightarrow q\big)$ are equivalent.

Problem 1.2. (10 points) Apply the optimized definitional clausal transformation algorithm to the formula:

$$\neg((p \leftrightarrow q) \leftrightarrow (r \leftrightarrow q))$$

Apply the DPLL algorithm to the resulting set of clauses. If the resulting set of clauses is satisfiable, give a model of the formula above.

Solution:

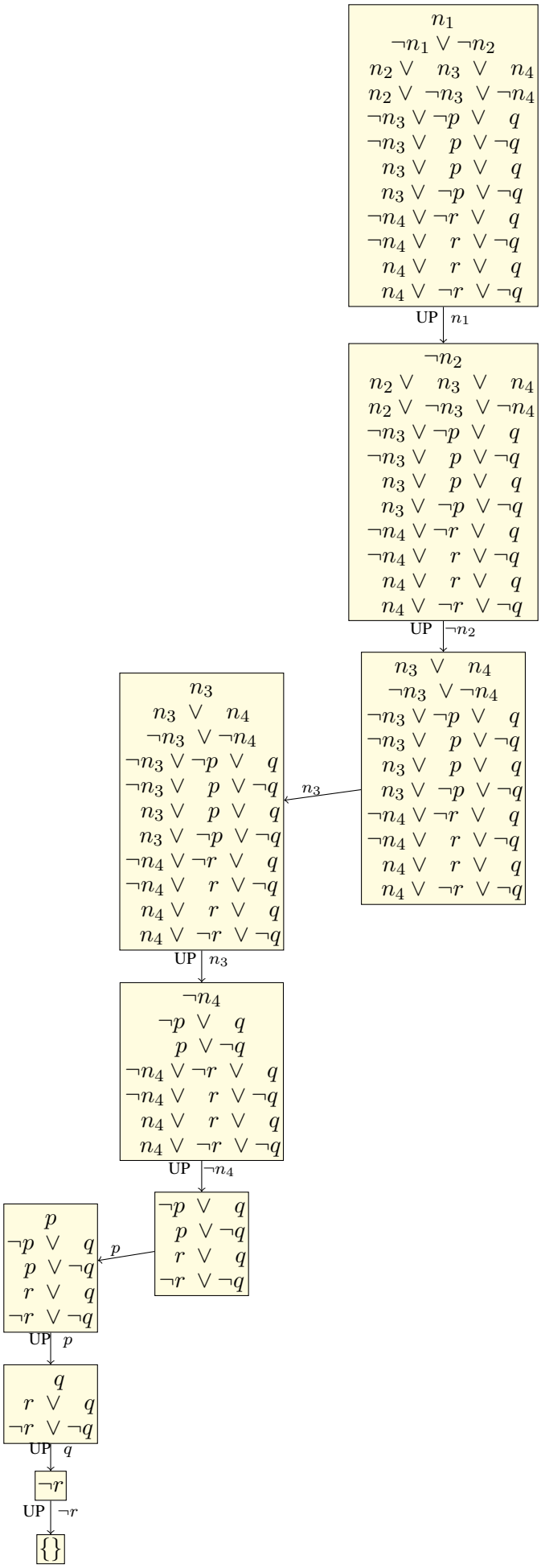
By applying the algorithm of definitional clausal form transformation over the given formula, we obtain the following clauses:

	subformula	definition	set C of clauses
			n_1
n_1	$\neg((p \leftrightarrow q) \leftrightarrow (r \leftrightarrow q))$	$n_1 \leftrightarrow \neg n_2$	$\neg n_1 \vee \neg n_2$ $n_1 \vee n_2$
n_2	$(p \leftrightarrow q) \leftrightarrow (r \leftrightarrow q)$	$n_2 \leftrightarrow (n_3 \leftrightarrow n_4)$	$\neg n_2 \vee \neg n_3 \vee n_4$ $\neg n_2 \vee n_3 \vee \neg n_4$ $n_2 \vee n_3 \vee n_4$ $n_2 \vee \neg n_3 \vee \neg n_4$
n_3	$p \leftrightarrow q$	$n_3 \leftrightarrow (p \leftrightarrow q)$	$\neg n_3 \vee \neg p \vee q$ $\neg n_3 \vee p \vee \neg q$ $n_3 \vee p \vee q$ $n_3 \vee \neg p \vee \neg q$
n_4	$r \leftrightarrow q$	$n_4 \leftrightarrow (r \leftrightarrow q)$	$\neg n_4 \vee \neg r \vee q$ $\neg n_4 \vee r \vee \neg q$ $n_4 \vee r \vee q$ $n_4 \vee \neg r \vee \neg q$

Observe that in the table above, n_1 has only positive polarity and n_2 has only negative polarity. By applying the algorithm of optimized definitional clausal form transformation over the given formula, we thus obtain the following clauses:

	subformula	definition	set C' of clauses
			n_1
n_1	$\neg((p \leftrightarrow q) \leftrightarrow (r \leftrightarrow q))$	$n_1 \rightarrow \neg n_2$	$\neg n_1 \vee \neg n_2$
n_2	$(p \leftrightarrow q) \leftrightarrow (r \leftrightarrow q)$	$n_2 \leftarrow (n_3 \leftrightarrow n_4)$	$n_2 \vee n_3 \vee n_4$ $n_2 \vee \neg n_3 \vee \neg n_4$
n_3	$p \leftrightarrow q$	$n_3 \leftrightarrow (p \leftrightarrow q)$	$\neg n_3 \vee \neg p \vee q$ $\neg n_3 \vee p \vee \neg q$ $n_3 \vee p \vee q$ $n_3 \vee \neg p \vee \neg q$
n_4	$r \leftrightarrow q$	$n_4 \leftrightarrow (r \leftrightarrow q)$	$\neg n_4 \vee \neg r \vee q$ $\neg n_4 \vee r \vee \neg q$ $n_4 \vee r \vee q$ $n_4 \vee \neg r \vee \neg q$

We apply DPLL on C'. We use the notation UP for a unit propagation step.


n_1
$\neg n_1 \vee \neg n_2$
$n_2 \vee n_3 \vee n_4$
$n_2 \vee \neg n_3 \vee \neg n_4$
$\neg n_3 \vee \neg p \vee q$
$\neg n_3 \vee p \vee \neg q$
$n_3 \vee p \vee q$
$n_3 \vee \neg p \vee \neg q$
$\neg n_4 \vee \neg r \vee q$
$\neg n_4 \vee r \vee \neg q$
$n_4 \vee r \vee q$
$n_4 \vee \neg r \vee \neg q$
UP n_1
$\neg n_2$
$n_2 \vee n_3 \vee n_4$
$n_2 \vee \neg n_3 \vee \neg n_4$
$\neg n_3 \vee \neg p \vee q$
$\neg n_3 \vee p \vee \neg q$
$n_3 \vee p \vee q$
$n_3 \vee \neg p \vee \neg q$
$\neg n_4 \vee \neg r \vee q$
$\neg n_4 \vee r \vee \neg q$
$n_4 \vee r \vee q$
$n_4 \vee \neg r \vee \neg q$
UP $\neg n_2$
$n_3 \vee n_4$
$\neg n_3 \vee \neg n_4$
$\neg n_3 \vee \neg p \vee q$
$\neg n_3 \vee p \vee \neg q$
$n_3 \vee p \vee q$
$n_3 \vee \neg p \vee \neg q$
$\neg n_4 \vee \neg r \vee q$
$\neg n_4 \vee r \vee \neg q$
$n_4 \vee r \vee q$
$n_4 \vee \neg r \vee \neg q$
n_3
n_3
$n_3 \vee n_4$
$\neg n_3 \vee \neg n_4$
$\neg n_3 \vee \neg p \vee q$
$\neg n_3 \vee p \vee \neg q$
$n_3 \vee p \vee q$
$n_3 \vee \neg p \vee \neg q$
$\neg n_4 \vee \neg r \vee q$
$\neg n_4 \vee r \vee \neg q$
$n_4 \vee r \vee q$
$n_4 \vee \neg r \vee \neg q$
UP n_3
$\neg n_4$
$\neg p \vee q$
$p \vee \neg q$
$\neg n_4 \vee \neg r \vee q$
$\neg n_4 \vee r \vee \neg q$
$n_4 \vee r \vee q$
$n_4 \vee \neg r \vee \neg q$
UP $\neg n_4$
p
$\neg p \vee q$
$p \vee \neg q$
$r \vee q$
$\neg r \vee \neg q$
p
$\neg p \vee q$
$p \vee \neg q$
$r \vee q$
$\neg r \vee \neg q$
UP p
q
$r \vee q$
$\neg r \vee \neg q$
UP q
$\neg r$
UP $\neg r$
{}

A branch of the DPLL tree returns the empty set, hence C' is satisfiable. A model of C' is:

$$\{n_1 \to 1, n_2 \to 0, n_3 \to 1, n_4 \to 0, p \to 1, q \to 1, r \to 0\},$$

from which we get a model of original input formula as:

$$\{p \to 1, q \to 1, r \to 0\}.$$

Problem 1.3. (25 points) Consider the formula:

$$(p \to (\neg q \wedge r)) \ \wedge \ (p \vee (q \to (r \vee p)))$$

(a) For each atom in the above formula, decide whether it is monotonic, anti-monotonic, or neither.

(b) Compute a clausal normal form C of the above formula by applying the optimized definitional clausal transformation algorithm.

(c) Decide the satisfiability of the computed CNF formula C by applying the DPLL method on C. If C is satisfiable, give an interpretation which satisfies it.

Solution:

(a) From the parse tree of the formula, we have:

- p both with polarity -1 and 1, hence p is not monotonic, nor anti-monotonic (that is, p is not pure);
- q has only polarity -1, hence it is anti-monotonic (that is, q is pure negative);
- r has only polarity 1, hence it is monotonic (that is, r is pure positive).

(b) By By applying the algorithm of optimized definitional clausal form transformation over the given formula, we obtain the following clauses:

	subformula	definition	set C of clauses
			n_1
n_1	$(p \to (\neg q \wedge r)) \ \wedge \ (p \vee (q \to (r \vee p)))$	$n_1 \to (n_2 \wedge n_3)$	$\neg n_1 \vee n_2$ $\neg n_1 \vee n_3$
n_2	$p \to (\neg q \wedge r)$	$n_2 \to (p \to n_4)$	$\neg n_2 \vee \neg p \vee n_4$
n_3	$p \vee (q \to (r \vee p))$	$n_3 \to (p \vee n_5)$	$\neg n_3 \vee \ p \ \vee n_5$
n_4	$\neg q \wedge r$	$n_4 \to (\neg q \wedge r)$	$\neg n_4 \vee \neg q$ $\neg n_4 \vee \ r$
n_5	$q \to (r \vee p)$	$n_5 \to (q \to n_6)$	$\neg n_5 \vee \neg q \vee n_6$
n_6	$r \vee p$	$n_6 \to (r \vee p)$	$\neg n_6 \vee \ r \ \vee p$

(c) Similarly to **Problem 1.2**, we apply the DPLL method on C. First, we do unit propagation with n_1, followed by unit propagation with n_2 and n_3. We then split on p and do unit propagation with p, followed by unit propagation with n_4, $\neg q$ and r, resulting then in the empty set. Hence C is satisfiable. A model of C is given by:

$$\{n_1 \to 1, n_2 \to 1, n_3 \to 1, n_4 \to 1, n_5 \to 1, n_6 \to 1, p \to 1, q \to 0, r \to 1\}.$$

From the above model of C we get a model of the original formulas as:

$$\{p \to 1, q \to 0, r \to 1\}.$$

Problem 1.4. (10 points) A formula A is in disjunctive normal form, or simply DNF, if it is either $\top$, or $\bot$, or it is a disjunction of conjunction of literals:

$$A = \bigvee_i \bigwedge_j L_{i,j},$$

where $L_{i,j}$ are literals.

Show that satisfiability of formulas in DNF can be checked in polynomial time.

Solution:
Consider a DNF formula A:

$$A = \bigvee_i^k \bigwedge_j L_{i,j},$$

where $L_{i,j}$ are literals.
A is satisfiable iff $\bigwedge_j L_{i,j}$ is satisfiable for some i. Thus, the following algorithm can be used to check satisfiability of A:

```
(1) for i = 1...k do:
(2)     If ⋀_j L_{i,j} is sat, then return A is sat
(3) return A is unsat
```

Recall that $\bigwedge_j L_{i,j}$ is unsat iff it contains complementary literals. Hence checking satisfiability of $\bigwedge_j L_{i,j}$ in line (2) reduces to check that $\bigwedge_j L_{i,j}$ does not contain complementary literals. This check can be done in the worst-case in quadratic time in the size of $\bigwedge_j L_{i,j}$ (that is, in the number of literals of $\bigwedge_j L_{i,j}$). As the above algorithm requires at most k iterations of the for-loop in line (1), all together the above algorithm decides satisfiability of A in polynomial time in the size of A.

Problem 1.5. (25 points) Consider the Sudoku puzzle for placing digits from 1 to 9 into a 9x9 grid. The grid is additionally divided into 9 blocks, each block representing a 3x3 grid. In the Sudoku puzzle, each cell in the 9x9 grid is filled with one digit from 1 to 9, by satisfying the following constraints:

(S1) every cell of the 9x9 grid contains exactly one digit from 1 to 9;

(S2) every row of the 9x9 grid must contain one of each digit from 1 to 9;

(S3) every column of the 9x9 grid must contain one of each digit from 1 to 9;

(S4) every 3x3 block of the 9x9 grid must contain one of each digit from 1 to 9.

Consider an instance of the Sudoku puzzle, as given in Figure 1:
Does this Sudoku instance has a solution? That is, is it possible to assign digits to the empty cells of Figure 1 by satisfying the Sudoku constraints? Provide your solution by solving the following tasks:

(a) Express the Sudoku constraints (S1)-(S4) in CNF, using propositional clauses. How many clauses does your CNF formalization use?

(b) Formalize the Sudoku instance of Figure 1 as an instance of SAT.

(c) Encode the Sudoku instance of Figure 1 as an input to the MiniSat solver and evaluate MiniSat on your encoding. Is Figure 1 satisfiable? If yes, fill-in the empty cells of Figure 1! Provide the electronic version of your MiniSat encoding together with your solution.

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

Figure 1: A Sudoku instance

Solution:

(a). We introduce 729 propositional variables v_{rcd}, where $r, c, d \in \{1, \ldots, 9\}$. The variable v_{rcd} is true iff the cell in the row number r and column number c contains the digit d. For example, the Sudoku instance of Figure 1 satisfies the formula $v_{129} \wedge v_{268} \wedge \neg v_{691}$.

We next express the Sudoku constraints (S1)-(S4), as follows:
(S1) Every cell contains exactly one digit:

$$
\begin{aligned}
& v_{rc1} \vee v_{rc2} \vee \ldots \vee v_{rc8} \vee v_{rc9} \\
& \neg v_{rc1} \vee \neg v_{rc2} \\
& \neg v_{rc1} \vee \neg v_{rc3} \\
& \ldots \\
& \neg v_{rc8} \vee \neg v_{rc9}
\end{aligned}
$$

This way, (S1) yields 2,997 clauses and 6,561 literals.

(S2) Every row must contain one of each digit:

$$\{\neg v_{r,c,d} \vee \neg v_{r,c',d} \mid r, c, c', d \in \{1, ..., 9\}, c < c'\}.$$

This way, (S2) yields 2,916 clauses and 5,832 literals.

(S3) Similarly, we encode the constraint that every column must contain one of each digit, yielding 2,916 clauses and 5,832 literals.

(S4) Similarly, we encode the constraint that every 3x3 block must contain one of each digit, yielding 2,916 clauses and 5,832 literals.

As a results, the constraints (S1)-(S4) yield 11,745 clauses and 24,057 literals, with nearly all clauses being binary clauses.

(b) In addition to the above constraints expressing (S1)-(S4), we add unit clauses corresponding to the initial configuration of the Sudoku instance from Figure 1. For example, we add v_{129} and v_{268} as unit clauses. The resulting set of clauses formalize the Sudoku instance of Figure 1.

(c) The Sudoku instance of Figure 1 is satisfiable, as shown in Figure 2.

Problem 1.6. (20 points)

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

Figure 2: A solution of the Sudoku instance of Figure 1.

Consider the set consisting of the following clauses:

$p_0 \vee \neg p_1 \vee p_2$ $\quad$ $p_0 \vee \neg p_1 \vee p_2 \vee p_4$ $\quad$ $\neg p_0 \vee p_1 \vee \neg p_2$ $\quad$ $\neg p_0 \vee \neg p_1 \vee \neg p_2$

$p_0 \vee \neg p_1 \vee p_4$ $\quad$ $p_3 \vee p_2 \vee p_4 \vee \neg p_0$ $\quad$ $\neg p_2 \vee \neg p_2 \vee p_4 \vee p_3$ $\quad$ $\neg p_2 \vee \neg p_0 \vee p_4 \vee p_4$

$p_0 \vee p_3 \vee \neg p_4$ $\quad$ $p_0 \vee \neg p_1 \vee \neg p_2 \vee \neg p_3$ $\quad$ $\neg p_1 \vee \neg p_2 \vee \neg p_3$ $\quad$ $p_1 \vee \neg p_2 \vee \neg p_3 \vee \neg p_4$

$p_1 \vee p_2$ $\quad$ $p_2 \vee p_3 \vee \neg p_4 \vee p_3$ $\quad$ $\neg p_0 \vee \neg p_2 \vee \neg p_3 \vee \neg p_4$ $\quad$ $p_0 \vee p_2 \vee p_4$

(a) For each of the variables p_0, p_1, p_2, p_3, p_4 find the probability that WSAT will choose this variable for flipping when the current interpretation is:

$$\{p_0 \mapsto 1, p_1 \mapsto 1, p_2 \mapsto 1, p_3 \mapsto 1, p_4 \mapsto 1\}.$$

(b) Answer the same question as above, but using GSAT instead of WSAT.

Solution:

(a) WSAT will first choose a clause that is false in this interpretation. That is, it will choose one clause from

$$\neg p_1 \vee \neg p_2 \vee \neg p_3$$
$$\neg p_0 \vee \neg p_2 \vee \neg p_3 \vee \neg p_4$$
$$\neg p_0 \vee \neg p_1 \vee \neg p_2,$$

each with the equal probability $\frac{1}{3}$. Then a variable in the clause will be selected for flipping, with an equal probability among the variables of the chosen clause. Hence,

- if $\neg p_1 \vee \neg p_2 \vee \neg p_3$ is chosen, then WSAT will choose a variable from this clause with probability $\frac{1}{3} * \frac{1}{3}$, that is with $\frac{1}{9}$.
- if $p_0 \vee \neg p_2 \vee \neg p_3 \vee \neg p_4$ is chosen, then WSAT will choose a variable from this clause with probability $\frac{1}{3} * \frac{1}{4}$, that is with $\frac{1}{12}$.
- if $\neg p_0 \vee \neg p_1 \vee \neg p_2$ is chosen, then WSAT will choose a variable from this clause with probability $\frac{1}{3} * \frac{1}{3}$, that is with $\frac{1}{9}$.

This gives us the following probabilities for a variable to be chosen by WSAT for flipping from the input set of clauses:

- p_0 will be chosen with probability $\frac{1}{12} + \frac{1}{9} = \frac{7}{36}$;
- p_1 will be chosen with probability $\frac{1}{9} + \frac{1}{9} = \frac{2}{9}$;
- p_2 will be chosen with probability $\frac{1}{9} + \frac{1}{12} + \frac{1}{9} = \frac{11}{36}$;
- p_3 will be chosen with probability $\frac{1}{9} + \frac{1}{12} = \frac{7}{36}$;
- p_4 will be chosen with probability $\frac{1}{12}$.

(b) In the current interpretation $I = \{p_0 \mapsto 1, p_1 \mapsto 1, p_2 \mapsto 1, p_3 \mapsto 1, p_4 \mapsto 1\}$, 13 clauses out of 16 are satisfied. GSAT will select a variable $x \in \{p_0, p_1, p_2, p_3, p_4\}$ for flipping, such that $flip(I, x)$ satisfies the maximal number of clauses. To this end, we have:

- $flip(I, p_0)$ satisfies 14 clauses;
- $flip(I, p_1)$ satisfies 13 clauses;
- $flip(I, p_2)$ satisfies 16 clauses;
- $flip(I, p_3)$ satisfies 15 clauses;
- $flip(I, p_4)$ satisfies 13 clauses.

Hence, GSAT will select p_2 for flipping with probability 1, and all other variables will be chosen with probability 0.