Exercise 1

> Prove, using splitting, that the formulas $p \leftrightarrow \neg q$ and $\neg p \leftrightarrow q$ are equivalent.

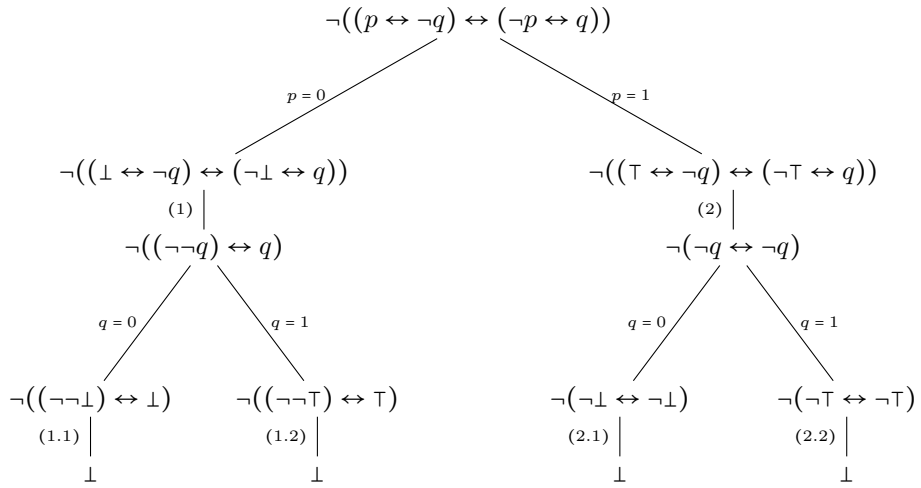
In order to show this one has to show that

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

Unfortunately, splitting can only decide whether a formula is satisfiable or not. Hence, the problem

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

has to be considered. That is, if splitting applied to the formula above returns "unsatisfiable", it can be concluded that $(p \leftrightarrow \neg q) \leftrightarrow (\neg p \leftrightarrow q)$ is valid.



With the respective transformations:

- for (1)

$$\neg((\bot \leftrightarrow \neg q) \leftrightarrow (\neg\bot \leftrightarrow q)) \overset{\neg\bot \Rightarrow \top}{\longmapsto} \neg((\bot \leftrightarrow \neg q) \leftrightarrow (\top \leftrightarrow q))$$
$$\overset{\bot \leftrightarrow A \Rightarrow \neg A}{\longmapsto} \neg((\neg\neg q) \leftrightarrow (\top \leftrightarrow q)) \overset{\top \leftrightarrow A \Rightarrow A}{\longmapsto} \neg((\neg\neg q) \leftrightarrow q)$$

- for (1.1)

$$\neg((\neg\neg\bot) \leftrightarrow \bot) \overset{A \leftrightarrow \bot \Rightarrow \neg A}{\longmapsto} \neg(\neg\neg\neg\bot) \overset{\neg\bot \Rightarrow \top}{\longmapsto} \neg(\neg\neg\top) \overset{\neg\top \Rightarrow \bot}{\longmapsto} \neg(\neg\bot)$$
$$\overset{\neg\bot \Rightarrow \top}{\longmapsto} \neg\top \overset{\neg\top \Rightarrow \bot}{\longmapsto} \bot$$

- for (1.2)

$$\neg((\neg\neg\top) \leftrightarrow \top) \overset{A \leftrightarrow \top \Rightarrow A}{\longmapsto} \neg(\neg\neg\top) \overset{\neg\top \Rightarrow \bot}{\longmapsto} \neg(\neg\bot) \overset{\neg\bot \Rightarrow \top}{\longmapsto} \neg\top \overset{\neg\top \Rightarrow \bot}{\longmapsto} \bot$$

- for (2)

$$\neg((\top \leftrightarrow \neg q) \leftrightarrow (\neg\top \leftrightarrow q)) \overset{\neg\top \Rightarrow \bot}{\longmapsto} \neg((\top \leftrightarrow \neg q) \leftrightarrow (\bot \leftrightarrow q))$$
$$\overset{\top \leftrightarrow A \Rightarrow A}{\longmapsto} \neg(\neg q \leftrightarrow (\bot \leftrightarrow q)) \overset{\bot \leftrightarrow A \Rightarrow \neg A}{\longmapsto} \neg(\neg q \leftrightarrow \neg q)$$

- for (2.1)

$$\neg(\neg\bot \leftrightarrow \neg\bot) \overset{\neg\bot \Rightarrow \top}{\longmapsto} \neg(\top \leftrightarrow \neg\bot) \overset{\top \leftrightarrow A \Rightarrow A}{\longmapsto} \neg(\neg\bot) \overset{\neg\bot \Rightarrow \top}{\longmapsto} \neg\top \overset{\neg\top \Rightarrow \bot}{\longmapsto} \bot$$

- for (2.2)

$$\neg(\neg\top \leftrightarrow \neg\top) \overset{\neg\top \Rightarrow \top}{\longmapsto} \neg(\bot \leftrightarrow \neg\bot) \overset{\bot \leftrightarrow A \Rightarrow \neg A}{\longmapsto} \neg(\neg\neg\top) \overset{\neg\top \Rightarrow \bot}{\longmapsto} \neg(\neg\bot)$$
$$\overset{\neg\bot \Rightarrow \top}{\longmapsto} \neg\top \overset{\neg\top \Rightarrow \bot}{\longmapsto} \bot$$

Exercise 2

> 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.

By applying the optimised definitional clausal transformation algorithm one obtains the following

labels	subformulas	definitions	clauses
-	-	n_1	n_1
n_1	$\neg((p \leftrightarrow q) \leftrightarrow (r \leftrightarrow q))$	$n_1 \rightarrow \neg n_2$	see (1)
n_2	$(p \leftrightarrow q) \leftrightarrow (r \leftrightarrow q)$	$(n_3 \leftrightarrow n_4) \rightarrow n_2$	see (2)
n_3	$(p \leftrightarrow q)$	$n_3 \leftrightarrow (p \leftrightarrow q)$	see (3)
n_4	$(r \leftrightarrow q)$	$n_4 \leftrightarrow (r \leftrightarrow q)$	see (4)

(Note that n_1 only occurs positively, n_2 only negatively and the other have no polarity)

From there the definitions must be transformed into CNF (some steps are skipped to decrease the amount of depicted transformations).

- (1) is $\neg n_1 \vee \neg n_2$, because $n_1 \rightarrow \neg n_2 \mapsto \neg n_1 \vee \neg n_2$
- (2) is $(n_3 \vee n_4 \vee n_2) \wedge (n_3 \vee \neg n_3 \vee n_2) \wedge (\neg n_4 \vee n_4 \vee n_2) \wedge (\neg n_4 \vee \neg n_3 \vee n_2)$, because $(n_3 \leftrightarrow n_4) \rightarrow n_2 \mapsto \neg(n_3 \leftrightarrow n_4) \vee n_2 \mapsto \neg((\neg n_3 \vee n_4) \wedge (\neg n_4 \vee n_3)) \vee n_2 \mapsto \neg(\neg n_3 \vee n_4) \vee \neg(\neg n_4 \vee n_3) \vee n_2 \mapsto (n_3 \wedge \neg n_4) \vee (n_4 \wedge \neg n_3) \vee n_2 \mapsto (n_3 \vee (n_4 \wedge \neg n_3)) \wedge (\neg n_4 \vee (n_4 \wedge \neg n_3)) \vee n_2 \mapsto ((n_3 \vee n_4) \wedge (n_3 \vee \neg n_3) \wedge (\neg n_4 \vee n_4) \wedge (\neg n_4 \vee \neg n_3)) \vee n_2 \mapsto (n_3 \vee n_4 \vee n_2) \wedge (n_3 \vee \neg n_3 \vee n_2) \wedge (\neg n_4 \vee n_4 \vee n_2) \wedge (\neg n_4 \vee \neg n_3 \vee n_2)$
- (3) is $(\neg n_3 \vee \neg p \vee q) \wedge (\neg n_3 \vee \neg q \vee p) \wedge (p \vee q \vee n_3) \wedge (p \vee \neg p \vee n_3) \wedge (\neg q \vee q \vee n_3) \wedge (\neg q \vee \neg p \vee n_3)$, because $n_3 \leftrightarrow (p \leftrightarrow q) \mapsto (\neg n_3 \vee (p \leftrightarrow q)) \wedge (\neg(p \leftrightarrow q) \vee n_3) \mapsto (\neg n_3 \vee ((\neg p \vee q) \wedge (\neg q \vee p)) \wedge (\neg(p \leftrightarrow q) \vee n_3) \mapsto (\neg n_3 \vee \neg p \vee q) \wedge (\neg n_3 \vee \neg q \vee p) \wedge (\neg(p \leftrightarrow q) \vee n_3) \mapsto (\neg n_3 \vee \neg p \vee q) \wedge (\neg n_3 \vee \neg q \vee p) \wedge (\neg((\neg p \vee q) \wedge (\neg q \vee p)) \vee n_3)$ from there after a sequence of transformations as in (2) one obtains $(\neg n_3 \vee \neg p \vee q) \wedge (\neg n_3 \vee \neg q \vee p) \wedge (p \vee q \vee n_3) \wedge (p \vee \neg p \vee n_3) \wedge (\neg q \vee q \vee n_3) \wedge (\neg q \vee \neg p \vee n_3)$
- (4) is $(\neg n_4 \vee \neg r \vee q) \wedge (\neg n_4 \vee \neg q \vee r) \wedge (r \vee q \vee n_4) \wedge (r \vee \neg r \vee n_4) \wedge (\neg q \vee q \vee n_4) \wedge (\neg q \vee \neg r \vee n_4)$, because of the same sequence of transformations as in (3).

Hence, we obtain the following set of clauses.

(n_1),	$(\neg n_3 \vee \neg p \vee q)$,	$(\neg n_4 \vee \neg r \vee q)$,
$(\neg n_1 \vee \neg n_2)$,	$(\neg n_3 \vee \neg q \vee p)$,	$(\neg n_4 \vee \neg q \vee r)$,
$(n_3 \vee n_4 \vee n_2)$,	$(p \vee q \vee n_3)$,	$(r \vee q \vee n_4)$,
$(n_3 \vee \neg n_3 \vee n_2)$,	$(p \vee \neg p \vee n_3)$,	$(r \vee \neg r \vee n_4)$,
$(\neg n_4 \vee n_4 \vee n_2)$,	$(\neg q \vee q \vee n_3)$,	$(\neg q \vee q \vee n_4)$,
$(\neg n_4 \vee \neg n_3 \vee n_2)$,	$(\neg q \vee \neg p \vee n_3)$,	$(\neg q \vee \neg r \vee n_4)$

Taking this set of clauses as input for DPLL one can observe the following sequence of events. However, this is not the only possible sequence of events, as the choice of literal is non-deterministic. That is, while the algorithm normally creates a tree, we were lucky and the initial choices of literals already showed satisfiability. Hence, due to the depth first search tactic of DPLL the algorithm only had to investigate a path within the search space.

1. (a) The clause set is:

(n_1),	$(\neg n_3 \vee \neg p \vee q)$,	$(\neg n_4 \vee \neg r \vee q)$,
$(\neg n_1 \vee \neg n_2)$,	$(\neg n_3 \vee \neg q \vee p)$,	$(\neg n_4 \vee \neg q \vee r)$,
$(n_3 \vee n_4 \vee n_2)$,	$(p \vee q \vee n_3)$,	$(r \vee q \vee n_4)$,
$(n_3 \vee \neg n_3 \vee n_2)$,	$(p \vee \neg p \vee n_3)$,	$(r \vee \neg r \vee n_4)$,
$(\neg n_4 \vee n_4 \vee n_2)$,	$(\neg q \vee q \vee n_3)$,	$(\neg q \vee q \vee n_4)$,
$(\neg n_4 \vee \neg n_3 \vee n_2)$,	$(\neg q \vee \neg p \vee n_3)$,	$(\neg q \vee \neg r \vee n_4)$

(b) Unit clause exists. Propagate:

i. Unit Clause: n_1

$$
\begin{array}{lll}
(n_1), & (\neg n_3 \vee \neg p \vee q), & (\neg n_4 \vee \neg r \vee q), \\
(\neg n_1 \vee \neg n_2), & (\neg n_3 \vee \neg q \vee p), & (\neg n_4 \vee \neg q \vee r), \\
(n_3 \vee n_4 \vee n_2), & (p \vee q \vee n_3), & (r \vee q \vee n_4), \\
(n_3 \vee \neg n_3 \vee n_2), & (p \vee \neg p \vee n_3), & (r \vee \neg r \vee n_4), \\
(\neg n_4 \vee n_4 \vee n_2), & (\neg q \vee q \vee n_3), & (\neg q \vee q \vee n_4), \\
(\neg n_4 \vee \neg n_3 \vee n_2), & (\neg q \vee \neg p \vee n_3), & (\neg q \vee \neg r \vee n_4)
\end{array}
$$

ii. Unit Clause: $\neg n_2$

$$
\begin{array}{lll}
 & (\neg n_3 \vee \neg p \vee q), & (\neg n_4 \vee \neg r \vee q), \\
(\neg n_2), & (\neg n_3 \vee \neg q \vee p), & (\neg n_4 \vee \neg q \vee r), \\
(n_3 \vee n_4 \vee n_2), & (p \vee q \vee n_3), & (r \vee q \vee n_4), \\
(n_3 \vee \neg n_3 \vee n_2), & (p \vee \neg p \vee n_3), & (r \vee \neg r \vee n_4), \\
(\neg n_4 \vee n_4 \vee n_2), & (\neg q \vee q \vee n_3), & (\neg q \vee q \vee n_4), \\
(\neg n_4 \vee \neg n_3 \vee n_2), & (\neg q \vee \neg p \vee n_3), & (\neg q \vee \neg r \vee n_4)
\end{array}
$$

(c) The clause set is not empty.

(d) The clause set does not contain the empty clause.

(e) Select literal: n_3

2. (a) The clause set is:

$$
\begin{array}{lll}
 & (\neg n_3 \vee \neg p \vee q), & (\neg n_4 \vee \neg r \vee q), \\
(n_3) & (\neg n_3 \vee \neg q \vee p), & (\neg n_4 \vee \neg q \vee r), \\
(n_3 \vee n_4), & (p \vee q \vee n_3), & (r \vee q \vee n_4), \\
(n_3 \vee \neg n_3), & (p \vee \neg p \vee n_3), & (r \vee \neg r \vee n_4), \\
(\neg n_4 \vee n_4), & (\neg q \vee q \vee n_3), & (\neg q \vee q \vee n_4), \\
(\neg n_4 \vee \neg n_3), & (\neg q \vee \neg p \vee n_3), & (\neg q \vee \neg r \vee n_4)
\end{array}
$$

(b) Unit clause exists. Propagate:

i. Unit Clause: n_3

$$
\begin{array}{lll}
 & (\neg n_3 \vee \neg p \vee q), & (\neg n_4 \vee \neg r \vee q), \\
(n_3) & (\neg n_3 \vee \neg q \vee p), & (\neg n_4 \vee \neg q \vee r), \\
(n_3 \vee n_4), & (p \vee q \vee n_3), & (r \vee q \vee n_4), \\
(n_3 \vee \neg n_3), & (p \vee \neg p \vee n_3), & (r \vee \neg r \vee n_4), \\
(\neg n_4 \vee n_4), & (\neg q \vee q \vee n_3), & (\neg q \vee q \vee n_4), \\
(\neg n_4 \vee \neg n_3), & (\neg q \vee \neg p \vee n_3), & (\neg q \vee \neg r \vee n_4)
\end{array}
$$

ii. Unit Clause: $\neg n_4$

$$
\begin{array}{ll}
(\neg p \vee q), & (\neg n_4 \vee \neg r \vee q), \\
(\neg q \vee p), & (\neg n_4 \vee \neg q \vee r), \\
 & (r \vee q \vee n_4), \\
 & (r \vee \neg r \vee n_4), \\
(\neg n_4 \vee n_4), & (\neg q \vee q \vee n_4), \\
(\neg n_4), & (\neg q \vee \neg r \vee n_4)
\end{array}
$$

(c) The clause set is not empty.

(d) The clause set does not contain the empty clause.

(e) Select literal: $\neg p$

3. (a) The clause set is:

$$
\begin{array}{l}
(\neg p), \\
(\neg p \vee q), \\
(\neg q \vee p), \\
(r \vee q), \\
(r \vee \neg r), \\
(\neg q \vee q), \\
(\neg q \vee \neg r)
\end{array}
$$

(b) Unit clause exists. Propagate:

i. Unit Clause: $\neg p$

$$
\begin{array}{l}
(\neg p), \\
(\neg p \vee q), \\
(\neg q \vee p), \\
(r \vee q), \\
(r \vee \neg r), \\
(\neg q \vee q), \\
(\neg q \vee \neg r)
\end{array}
$$

ii. Unit Clause: $\neg q$

$$
\begin{array}{l}
(\neg q), \\
(r \vee q), \\
(r \vee \neg r), \\
(\neg q \vee q), \\
(\neg q \vee \neg r)
\end{array}
$$

iii. Unit Clause: r

$$(r),$$
$$(r \vee \neg r)$$

(c) The clause set is empty. Return *satisfiable*.

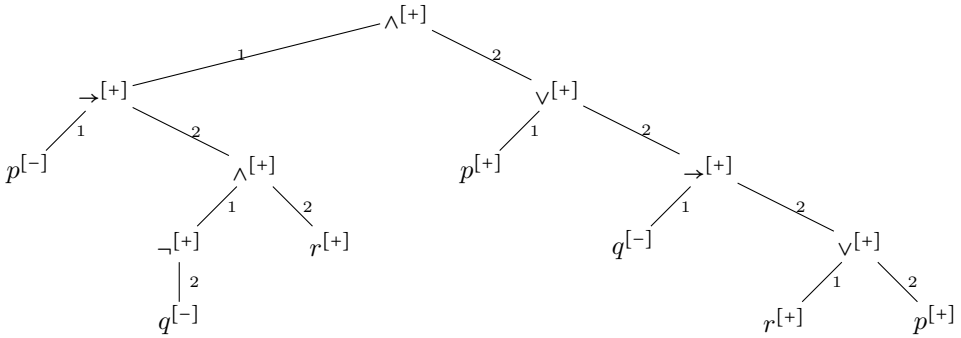
This is only one path in the whole DPLL tree. If this selection of literals would have resulted in the empty clause one would need to consider further branches of the DPLL tree. Given this path the following literals were chosen during unit propagation $n_1, \neg n_2, n_3, \neg n_4, \neg p, \neg q, r$. Hence, resulting in the interpretation $\{n_1 \mapsto 1, n_2 \mapsto 0, n_3 \mapsto 1, n_4 \mapsto 0, p \mapsto 0, q \mapsto 0, r \mapsto 1\}$. Lastly, having found one interpretation, further investigation of the DPLL tree is not required.

If one chooses to apply the two optimisations before applying DPLL, one has to remove the tautologies

(n_1)	$(\neg n_3 \vee \neg p \vee q)$,	$(\neg n_4 \vee \neg r \vee q)$,
$(\neg n_1 \vee \neg n_2)$,	$(\neg n_3 \vee \neg q \vee p)$,	$(\neg n_4 \vee \neg q \vee r)$,
$(n_3 \vee n_4 \vee n_2)$,	$(p \vee q \vee n_3)$,	$(r \vee q \vee n_4)$,
$(n_3 \vee \neg n_3 \vee n_2)$,	$(p \vee \neg p \vee n_3)$,	$(r \vee \neg r \vee n_4)$,
$(\neg n_4 \vee n_4 \vee n_2)$,	$(\neg q \vee q \vee n_3)$,	$(\neg q \vee q \vee n_4)$,
$(\neg n_4 \vee \neg n_3 \vee n_2)$,	$(\neg q \vee \neg p \vee n_3)$,	$(\neg q \vee \neg r \vee n_4)$

as well as the clauses with pure literals. However, in this case there are none. From there this reduced set of clauses serves as input for DPLL, producing a computation similar to the one above.

Exercise 3



Since, q only has negative polarity it is anti-monotonic. Moreover, the only occurrence r is positive, thus r is monotonic. Lastly, p occurs positively as well as negatively, thus p is neither monotonic nor anti-monotonic.

labels	subformulas	definitions	clauses
-	-	n_1	n_1
n_1	$(p \to (\neg q \wedge r)) \wedge (p \vee (q \to (r \vee p)))$	$n_1 \to (n_2 \wedge n_4)$	see (1)
n_2	$(p \to (\neg q \wedge r))$	$n_2 \to (p \to n_3)$	see (2)
n_3	$(\neg q \wedge r)$	$n_3 \to (\neg q \wedge r)$	see (3)
n_4	$(p \vee (q \to (r \vee p)))$	$n_4 \to (p \vee n_5)$	see (4)
n_5	$(q \to (r \vee p))$	$n_5 \to (q \to n_6)$	see (5)
n_6	$(r \vee p)$	$n_6 \to (r \vee p)$	see (6)

(Note that all subformulas occur positively)

- (1) is $(\neg n_1 \vee n_2) \wedge (\neg n_1 \vee n_4)$, because $n_1 \to (n_2 \wedge n_4) \mapsto \neg n_1 \vee (n_2 \wedge n_4) \mapsto \neg n_1 \vee n_2 \wedge \neg n_1 \vee n_4$;
- (2) is $\neg n_2 \vee \neg p \vee n_3$, because $n_2 \to (p \to n_3) \mapsto \neg n_2 \vee (p \to n_3) \mapsto \neg n_2 \vee \neg p \vee n_3$;
- (3) is $(\neg n_3 \vee \neg q) \wedge (\neg n_3 \vee r)$, because $n_3 \to (\neg q \wedge r) \mapsto \neg n_3 \vee (\neg q \wedge r) \mapsto \neg n_3 \vee \neg q \wedge \neg n_3 \vee r$;
- (4) is $\neg n_4 \vee p \vee n_5$, because $n_4 \to (p \vee n_5) \mapsto \neg n_4 \vee p \vee n_5$;
- (5) is $\neg n_5 \vee \neg q \vee n_6$, because $n_5 \to (q \to n_6) \mapsto \neg n_5 \vee (q \to n_6) \mapsto \neg n_5 \vee \neg q \vee n_6$;
- (6) is $\neg n_6 \vee r \vee p$, because $n_6 \to (r \vee p) \mapsto \neg n_6 \vee r \vee p$

Hence, one obtains the following set of clauses.

$$
\begin{array}{ll}
(n_1) & \\
(\neg n_1 \vee n_2), & (\neg n_1 \vee n_4), \\
(\neg n_2 \vee \neg p \vee n_3), & \\
(\neg n_3 \vee \neg q), & (\neg n_3 \vee r), \\
(\neg n_4 \vee p \vee n_5), & \\
(\neg n_5 \vee \neg q \vee n_6), & \\
(\neg n_6 \vee r \vee p) &
\end{array}
$$

1. (a) The clause set is:

$$
\begin{array}{ll}
(n_1) & \\
(\neg n_1 \vee n_2), & (\neg n_1 \vee n_4), \\
(\neg n_2 \vee \neg p \vee n_3), & \\
(\neg n_3 \vee \neg q), & (\neg n_3 \vee r), \\
(\neg n_4 \vee p \vee n_5), & \\
(\neg n_5 \vee \neg q \vee n_6), & \\
(\neg n_6 \vee r \vee p) &
\end{array}
$$

(b) Unit clause exists. Propagate:

i. Unit Clause: n_1

$$
\begin{array}{ll}
(n_1) & \\
(\neg n_1 \vee n_2), & (\neg n_1 \vee n_4), \\
(\neg n_2 \vee \neg p \vee n_3), & \\
(\neg n_3 \vee \neg q), & (\neg n_3 \vee r), \\
(\neg n_4 \vee p \vee n_5), & \\
(\neg n_5 \vee \neg q \vee n_6), & \\
(\neg n_6 \vee r \vee p) &
\end{array}
$$

ii. Unit Clause: n_2

$$
\begin{array}{ll}
(n_2), & (n_4), \\
(\neg n_2 \vee \neg p \vee n_3), & \\
(\neg n_3 \vee \neg q), & (\neg n_3 \vee r), \\
(\neg n_4 \vee p \vee n_5), & \\
(\neg n_5 \vee \neg q \vee n_6), & \\
(\neg n_6 \vee r \vee p) &
\end{array}
$$

iii. Unit Clause: n_4

$$
\begin{array}{ll}
(\neg p \vee n_3), & (n_4), \\
(\neg n_3 \vee \neg q), & (\neg n_3 \vee r), \\
(\neg n_4 \vee p \vee n_5), & \\
(\neg n_5 \vee \neg q \vee n_6), & \\
(\neg n_6 \vee r \vee p) &
\end{array}
$$

(c) The clause set is not empty.

(d) The clause set does not contain the empty clause.

(e) Select literal: n_3

2. (a) The clause set is:

$$(n_3),$$
$$(\neg p \vee n_3),$$
$$(\neg n_3 \vee \neg q), \qquad (\neg n_3 \vee r),$$
$$(p \vee n_5),$$
$$(\neg n_5 \vee \neg q \vee n_6),$$
$$(\neg n_6 \vee r \vee p)$$

(b) Unit clause exists. Propagate:

i. Unit Clause: n_3

$$(n_3),$$
$$(\neg p \vee n_3),$$
$$(\neg n_3 \vee \neg q), \qquad (\neg n_3 \vee r),$$
$$(p \vee n_5),$$
$$(\neg n_5 \vee \neg q \vee n_6),$$
$$(\neg n_6 \vee r \vee p)$$

ii. Unit Clause: $\neg q$

$$(\neg q), \qquad (r),$$
$$(p \vee n_5),$$
$$(\neg n_5 \vee \neg q \vee n_6),$$
$$(\neg n_6 \vee r \vee p)$$

iii. Unit Clause: r

$$(r),$$
$$(p \vee n_5),$$
$$(\neg n_6 \vee r \vee p)$$

(c) The clause set is not empty.

(d) The clause set does not contain the empty clause.

(e) Select literal: p

3. (a) The clause set is:

$$(p)$$
$$(p \vee n_5)$$
$$(\neg n_6 \vee p)$$

(b) Unit clause exists. Propagate:

i. Unit Clause: p

$$(p)$$
$$(p \vee n_5)$$
$$(\neg n_6 \vee p)$$

(c) The clause set is empty. Return *satisfiable*.

Given this path the following literals were chosen during unit propagation $n_1, n_2, n_3, n_4, p, \neg q, r$. Since, n_5 and n_6 were not fixed, its truth value is inconsequential (given the fixed atoms). Hence, resulting in the interpretation $\{n_1 \mapsto 1, n_2 \mapsto 1, n_3 \mapsto 1, n_4 \mapsto 1, n_5 \mapsto 0, n_6 \mapsto 0, p \mapsto 1, q \mapsto 0, r \mapsto 1\}$. Lastly, having found one interpretation, further investigation of the DPLL tree is not required.

Alternatively, applying the optimisation rules. That is, remove all tautologies and all clauses with pure literals. There are no tautologies. Leaving the removal of clauses with pure literals.

1. r and $\neg q$ are pure.

$$(n_1)$$
$$(\neg n_1 \vee n_2), \qquad (\neg n_1 \vee n_4),$$
$$(\neg n_2 \vee \neg p \vee n_3),$$
$$(\neg n_3 \vee \neg q), \qquad (\neg n_3 \vee r),$$
$$(\neg n_4 \vee p \vee n_5),$$
$$(\neg n_5 \vee \neg q \vee n_6),$$
$$(\neg n_6 \vee r \vee p)$$

2. n_3 and n_5 are pure.

$$(n_1)$$
$$(\neg n_1 \vee n_2), \qquad (\neg n_1 \vee n_4),$$
$$(\neg n_2 \vee \neg p \vee n_3),$$
$$(\neg n_4 \vee p \vee n_5)$$

3. n_2 and n_4 are pure.

$$(n_1)$$
$$(\neg n_1 \vee n_2), \qquad (\neg n_1 \vee n_4)$$

4. n_1 is pure.

$$(n_1)$$

Since there are no clauses left, DPLL is not required. Resulting in the interpretation $\{n_1 \mapsto 1, n_2 \mapsto 1, n_3 \mapsto 1, n_4 \mapsto 1, n_5 \mapsto 1, n_6 \mapsto 0, p \mapsto 0, q \mapsto 0, r \mapsto 1\}$, where n_6 can be chosen freely.

Exercise 4

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

To show polynomial membership of DNF satisfiability, it suffices to devise a polynomial time algorithm $\mathcal{A}$ and to check its validity. To that end, let $\mathcal{A}$ be an algorithm taking a DNF formula $\varphi \in \mathcal{L}^0$ and returning 1 if φ is satisfiable and 0 otherwise. Moreover, let $C(\varphi)$ be the set of clauses in φ, let $L(\varphi)$ be the set of literals in φ and $A(\varphi)$ be the set of atoms in φ.

```
A(φ):
for c ∈ C(φ):
  if L(c) = {⊤} or L(c) = {¬⊥} or L(c) = {⊤, ¬⊥}:
    return 1

X := ∅
for c ∈ C(φ):
  for l ∈ L(c):
    if l ≠ ⊥ and l ≠ ¬⊤:
      X := X ∪ {c}

X' := {}
for c ∈ X:
  add := true
  for l_i ∈ L(c):
    for l_j ∈ L(c):
      if l_j = ¬l_i:
        add := false
  if add:
    X' := X' ∪ c

if X' = ∅:
  return 0
else:
  return 1
```

Firstly, it is easy to see that this algorithm runs in polynomial time. Let n be the length of the formula φ with respect to some reasonable measure of length, e.g. number of connectives. Since, the number of clauses is bound by the length of the formula the first loop runs in $O(n)$. Moreover, the second loop also runs in $O(n)$, as one has to check every literal in φ. As for the third loop, there a runtime of $O(n^2)$ can be observed, i.e. in the worst case every literal has to be check against every other literal. Hence, $\mathcal{A}$ runs in $O(n^2)$.
Secondly, the validity of $\mathcal{A}$. That is,

$$\varphi \; sat. \iff \mathcal{A}(\varphi) = 1$$

Starting with "$\Rightarrow$". If φ is satisfiable, it can not be empty. Furthermore, there must be an interpretation $\mathcal{I}$ such that for at least one clause $c \in C(\varphi)$ $\mathcal{I} \vDash c$ (by semantics of $\vee$). If $L(c) = \{\top\}$ or $L(c) = \{\neg\bot\}$ or $L(c) = \{\top, \neg\bot\}$, φ is a tautology, thus justifying the first loop of $\mathcal{A}$. Moreover, if $\bot \in L(c)$ or $\neg\top \in L(c)$ a contradiction is obtained, as those clauses can never by satisfied. Hence, their removal in loop two of $\mathcal{A}$ is justified. Thereby, the special cases are dealt with. By semantics of $\wedge$ it must therefore hold that for all $l \in L(c)$ $\mathcal{I} \vDash l$. However, this can not be the case if $l, \bar{l} \in L(c)$. However, by assumption $\mathcal{I} \vDash c$, thus the dual of a literal in $L(c)$ can not be in $L(c)$ as well. Therefore, c not be removed during the third loop, thus X' is not empty and $\mathcal{A}(\varphi) = 1$.
As for "$\Leftarrow$". If $\mathcal{A}(\varphi) = 1$ then in *Case 1* it must be that $\{\top\} \in C(\varphi)$ or $\{\neg\bot\} \in C(\varphi)$ or $L(c) = \{\top, \neg\bot\}$, which implies by semantics of $\vee$ that φ is satisfiable by every interpretation. While in *Case 2* it holds that X' is not empty. That is, for at least one $c \in X'$ it must hold that if $l \in L(c)$ then $\bar{l} \notin L(c)$ and that $\neg\top \notin L(c)$ and $\bot \notin L(c)$. Now consider the following assignment for all $a \in A(\varphi)$

$$\mathcal{I}(a) := \begin{cases} 1 & \text{if } a \in L(c) \\ 0 & \text{otw.} \end{cases}$$

It is easy to see that $\mathcal{I}$ satisfies all literals in $L(c)$. Hence, $\mathcal{I} \vDash c$ and thus $\mathcal{I} \vDash \varphi$ (by semantics of $\vee$).

Exercise 5

> 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.

Before formalising the problem at hand a small observation. Consider the four propositional variables a, b, c and d. The aim, transform the statement

$$\begin{aligned}&(a \wedge \neg b \wedge \neg c \wedge \neg d) \vee \\ &(\neg a \wedge b \wedge \neg c \wedge \neg d) \vee \\ &(\neg a \wedge \neg b \wedge c \wedge \neg d) \vee \\ &(\neg a \wedge \neg b \wedge \neg c \wedge d)\end{aligned}$$

into an equivalent statement in CNF. Consider

$$\begin{aligned}&(a \vee b \vee c \vee d) \wedge \\ &(\neg a \vee \neg b) \wedge \\ &(\neg a \vee \neg c) \wedge \\ &(\neg a \vee \neg d) \wedge \\ &(\neg b \vee \neg c) \wedge \\ &(\neg b \vee \neg d) \wedge \\ &(\neg c \vee \neg d)\end{aligned}$$

It is easy to observe, that at least one variable has to be set to true. However, due to the binary disjunctions at most one variable can be set to true. Therefore, only one variable can be set to true. However, the choice of variable is arbitrary. For example,set c to true, c occurs in negated form in three binary disjunctions. Hence, the respective other literal has to hold, i.e. $\neg a, \neg b$ and $\neg d$ have to be true.

Let c_{ijn} be a proposition which is true iff the cell in row i and in column j has the value n where $i, j, n \in \{1, 2, \dots 9\}$. To make things more explicit let $R = C = N := \{1, 2, \dots 9\}$, let $B := \{1, 2, 3\}$ and let $R_1 = C_1 := \{1, 2, 3\}$, $R_2 = C_2 := \{4, 5, 6\}$ and $R_3 = C_3 := \{7, 8, 9\}$.
Firstly, *"(S1) every cell of the 9x9 grid contains exactly one digit from 1 to 9;"*, by using the idea developed above, consider

$$(\bigwedge_{i \epsilon R} \bigwedge_{j \epsilon C} (\bigvee_{n \epsilon N} c_{ijn})) \wedge$$
$$(\bigwedge_{i \epsilon R} \bigwedge_{j \epsilon C} \bigwedge_{n \epsilon N} \bigwedge_{\substack{m \epsilon N \\ m > n}} (\neg c_{ijn} \vee \neg c_{ijm}))$$

The set of big clauses, ensures that for each cell $(R \times C)$ at least one colour is chosen. While the short clauses ensure that for each cell only one number can be chosen. That is, for one cell every number predicate is in its negated form in a disjunction with the negated form of every other number predicate. Hence, by choosing one number predicate, all other number predicates for this cell have to be set to false.

Secondly, *"(S2) every row of the 9x9 grid must contain one of each digit from 1 to 9;"* this will be done in a similar fashion as above.

$$\bigwedge_{i \epsilon R} \bigwedge_{n \epsilon N} (\bigvee_{j \epsilon C} c_{ijn})$$

The set of big clauses, ensures that for each row r and for each number n there must be at least one cell in r that holds the number n. Moreover, as formalised in the previous point only one assignment per cell is possible. Hence, by the pigeon hole principle a number can only occur once.

Thirdly, *"(S3) every column of the 9x9 grid must contain one of each digit from 1 to 9;"* is formalised in analogue to the previous one.

$$\bigwedge_{j \epsilon C} \bigwedge_{n \epsilon N} (\bigvee_{i \epsilon R} c_{ijn})$$

The set of big clauses, ensures that for each column c and for each number n there must be at least one cell in c that holds the number n. Moreover, as formalised in the previous point only one assignment per cell is possible. Hence, by the pigeon hole principle a number can only occur once.

Lastly, again the statement *"(S4) every 3x3 block of the 9x9 grid must contain one of each digit from 1 to 9."* is formalised in a similar fashion to the cases above.

$$\bigwedge_{b_r \epsilon B} \bigwedge_{b_c \epsilon B} \bigwedge_{n \epsilon N} (\bigvee_{i \epsilon R_{b_r}} \bigvee_{j \epsilon C_{b_c}} c_{ijn})$$

The first two conjunctions loop through the blocks. The rest states that for every block there must be at least one cell that contains n. By the same argument as above this suffices.

Hence, allowing the construction of

$$\begin{aligned}\Gamma :=&(\bigwedge_{i\in R}\bigwedge_{j\in C}(\bigvee_{n\in N} c_{ijn}))\wedge\\ &(\bigwedge_{i\in R}\bigwedge_{j\in C}\bigwedge_{n\in N}\bigwedge_{\substack{m\in N\\ m>n}}(\neg c_{ijn}\vee\neg c_{ijm}))\wedge\\ &\bigwedge_{i\in R}\bigwedge_{n\in N}(\bigvee_{j\in C} c_{ijn})\wedge\\ &\bigwedge_{j\in C}\bigwedge_{n\in N}(\bigvee_{i\in R} c_{ijn})\wedge\\ &\bigwedge_{b_r\in B}\bigwedge_{b_c\in B}\bigwedge_{n\in N}(\bigvee_{i\in R_{b_r}}\bigvee_{j\in C_{b_c}} c_{ijn})\end{aligned}$$

The number of clauses in Γ is $(9\cdot 9+9\cdot 9\cdot\frac{9\cdot(9-1)}{2}))+9\cdot 9+9\cdot 9+3\cdot 3\cdot 9 = 4\cdot 9^2+4\cdot 9^3 = 3240$.

A small comment with respect to this formalisation. While being very concise with respect to the amount of clauses, it is probably not the most time efficient choice of formalisation, due to abundance of long clauses. An alternative formalisation would use the concept of maximality rather than minimality. That is, it would express that there can be at most one occurrence of a number per row, column, box. For example, for the rows

$$(\bigwedge_{i\in R}\bigwedge_{n\in N}\bigwedge_{j\in C}\bigwedge_{\substack{k\in C\\ k>j}}(\neg c_{ijn}\vee\neg c_{ikn}))$$

which would suffice, as it is already required that each cell has to be filled. While continuing this formalisation would drastically increase the amount of clauses, those clauses would be of size two. Hence, I would conjecture that this would result in a more time efficient formalisation of the problem at hand.

Moving on towards formalising the sudoku in the exercise as SAT.

$$\begin{aligned}\Gamma' :=&c_{114}\wedge c_{138}\wedge c_{147}\wedge c_{159}\wedge\\ &c_{239}\wedge c_{248}\wedge c_{252}\wedge c_{275}\wedge\\ &c_{322}\wedge\\ &c_{419}\wedge c_{432}\wedge c_{466}\wedge c_{481}\wedge c_{497}\wedge\\ &c_{536}\wedge c_{545}\wedge c_{558}\wedge c_{567}\wedge c_{579}\wedge\\ &c_{617}\wedge c_{628}\wedge c_{642}\wedge c_{674}\wedge c_{696}\wedge\\ &c_{784}\\ &c_{835}\wedge c_{854}\wedge c_{868}\wedge c_{872}\wedge\\ &c_{929}\wedge c_{957}\wedge c_{962}\wedge c_{973}\wedge c_{995}\end{aligned}$$

Hence, it has to be tested whether $\Gamma \wedge \Gamma'$ is sat. After applying "minisat". The following result is obtained (i.e. it is satisfiable).

4	1	8	7	9	5	6	2	3
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	3	2
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

Regarding the encoding of into a *minisat* instance. The following encoding is chosen.

$$\begin{array}{ccc} \tau(c_{111}) = 1 & \dots & \tau(c_{191}) = 9 \\ \vdots & \ddots & \vdots \\ \tau(c_{911}) = 72 & \dots & \tau(c_{991}) = 81 \\ \tau(c_{112}) = 82 & \dots & \tau(c_{192}) = 91 \\ \vdots & \ddots & \vdots \\ \tau(c_{919}) = 720 & \dots & \tau(c_{999}) = 729 \end{array}$$

Exercise 6

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

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

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

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

The current assignment is

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

Given this assignment the following clauses are satisfied (blue) and not satisfied (red)

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

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

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

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

WSAT randomly selects a non satisfied clause c. Then given c it selects a variable from c at random and flips its assignment. For the sake of this exercise uniform probability is assumed. There are three unsatisfied clauses, i.e. $c_4 = \neg p_0 \vee \neg p_1 \vee \neg p_2$, $c_{11} = \neg p_1 \vee \neg p_2 \vee \neg p_3$ and $c_{15} = \neg p_0 \vee \neg p_2 \vee \neg p_3 \vee \neg p_4$.

- p_0 occurs in c_4 and c_{15}, with the prior having 3 and the latter having 4 variables. Hence, $P(p_0) = \frac{1}{3} \cdot \frac{1}{3} + \frac{1}{3} \cdot \frac{1}{4} = \frac{7}{36}$.
- p_1 occurs in c_4 and c_{11}, with the prior having 3 and the latter having 3 variables. Hence, $P(p_1) = \frac{1}{3} \cdot \frac{1}{3} + \frac{1}{3} \cdot \frac{1}{3} = \frac{2}{9}$.
- p_2 occurs in c_4, c_{11} and c_{15}, with the first two having 3 and the last having 4 variables. Hence, $P(p_2) = \frac{1}{3} \cdot \frac{1}{3} + \frac{1}{3} \cdot \frac{1}{3} + \frac{1}{3} \cdot \frac{1}{4} = \frac{11}{36}$.
- p_3 occurs in c_{11} and c_{15}, with the prior having 3 and the latter having 4 variables. Hence, $P(p_3) = \frac{1}{3} \cdot \frac{1}{3} + \frac{1}{3} \cdot \frac{1}{4} = \frac{7}{36}$.
- p_4 occurs only in c_{15}, a clause having 4 variables. Hence, $P(p_4) = \frac{1}{3} \cdot \frac{1}{4} = \frac{1}{12}$.

GSAT chooses the variable which selection maximises the amount of satisfied clauses. Hence, the first step is to count the amount of clauses satisfied by flipping the respective variable.

- Flipping p_0 to $p_0 \mapsto 0$ leaves two clauses unsatisfied. That is,

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

- Flipping p_1 to $p_1 \mapsto 0$ leaves three clauses unsatisfied. That is,

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

- Flipping p_2 to $p_2 \mapsto 0$ leaves zero clauses unsatisfied. That is,

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

- Flipping p_3 to $p_3 \mapsto 0$ leaves one clauses unsatisfied. That is,

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

- Flipping p_4 to $p_4 \mapsto 0$ leaves three clauses unsatisfied. That is,

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

There exists a single optimal choice

$$p_1 = p_4 < p_0 < p_3 < p_2$$
$$13 = 13 < 14 < 15 < 16$$

Hence, the following probabilities can be computed. $P(p_0) = P(p_1) = P(p_3) = P(p_4) = 0$ and $P(p_2) = 1$.