Exercise 1

For each of the following formulas use the DPLL procedure to determine whether ϕ is satisfiable or unsatisfiable. Give a complete trace of the algorithm, showing the simplified formula for each recursive call of the DPLL function. Assume that DPLL selects variables in alphabetical order (i.e. $A, B, C, D, E, \ldots$), and that the splitting rule first attempts the value False (F) and then the value True (T).

(a) $\phi_1 = (A \vee B) \wedge (B \vee C) \wedge (\neg B \vee D) \wedge (\neg A \vee \neg D) \wedge (\neg C \vee E) \wedge (A \vee \neg B \vee \neg D)$

(b) $\phi_2 = (\neg A \vee \neg B \vee \neg C) \wedge (A \vee \neg B) \wedge (A \vee \neg D) \wedge (B \vee \neg E) \wedge (\neg C \vee D) \wedge (C \vee E)$
$\wedge (C \vee \neg E) \wedge (\neg D \vee E)$

Solution:

a) DPLL trace:

$$\{\{A, B\}, \{B, C\}, \{\neg B, D\}, \{\neg A, \neg D\}, \{\neg C, E\}, \{A, \neg B, \neg D\}\}$$

1. Splitting rule:

1a. $A \mapsto F$

$\{\{B\}, \{B, C\}, \{\neg B, D\}, \{\neg C, E\},$
$\{\neg B, \neg D\}\}$

2a. UP rule: $B \mapsto T$

$\{\{D\}, \{\neg C, E\}, \{\neg D\}\}$

3a. UP rule: $D \mapsto F$

$\{\square, \{\neg C, E\}\}$

1b. $A \mapsto T$

$\{\{B, C\}, \{\neg B, D\}, \{\neg D\},$
$\{\neg C, E\}\}$

2b. UP rule: $D \mapsto F$

$\{\{B, C\}, \{\neg B\}, \{\neg C, E\}\}$

3b. UP rule: $B \mapsto F$

$\{\{C\}, \{\neg C, E\}\}$

4b. UP rule: $C \mapsto T$

$\{\{E\}\}$

5b. UP rule: $E \mapsto T$

$\{\}$

Satisfying assignment: $A, \neg B, C, \neg D, E$

b) DPLL trace:

$$\{\{\neg A, \neg B, \neg C\}, \{A, \neg B\}, \{A, \neg D\}, \{B, \neg E\}, \{\neg C, D\}, \{C, E\}, \{C, \neg E\}, \{\neg D, E\}\}$$

1. Splitting rule:

1a. $A \mapsto F$
$\{\{\neg B\}, \{\neg D\}, \{B, \neg E\}, \{\neg C, D\}, \{C, E\}, \{C, \neg E\}, \{\neg D, E\}\}$

2a. UP rule: $B \mapsto F$
$\{\{\neg D\}, \{\neg E\}, \{\neg C, D\}, \{C, E\}, \{C, \neg E\}, \{\neg D, E\}\}$

3a. UP rule: $D \mapsto F$
$\{\{\neg E\}, \{\neg C\}, \{C, E\}, \{C, \neg E\}\}$

4a. UP rule: $C \mapsto F$
$\{\{\neg E\}, \{E\}, \{\neg E\}\}$

5a. UP rule: $E \mapsto F$
$\{\square\}$

1. Splitting rule:

   1b. $A \mapsto T$
   $\{\{\neg B, \neg C\}, \{B, \neg E\}, \{\neg C, D\}, \{C, E\}, \{C, \neg E\}, \{\neg D, E\}\}$

   2b. Splitting rule:

   2ba. $B \mapsto F$
   $\{\{\neg E\}, \{\neg C, D\}, \{C, E\}, \{C, \neg E\}, \{\neg D, E\}\}$

   3ba. UP rule: $E \mapsto F$
   $\{\{\neg C, D\}, \{C\}, \{\neg D\}\}$

   4ba. UP rule: $C \mapsto T$
   $\{\{D\}, \{\neg D\}\}$

   5ba. UP rule: $D \mapsto F$
   $\{\square\}$

   2bb. $B \mapsto T$
   $\{\{\neg C\}, \{\neg C, D\}, \{C, E\}, \{C, \neg E\}, \{\neg D, E\}\}$

   3bb. UP rule: $C \mapsto F$
   $\{\{E\}, \{\neg E\}, \{\neg D, E\}\}$

   4bb. UP rule: $E \mapsto F$
   $\{\square, \{\neg D\}\}$

   5bb. UP rule: $D \mapsto F$
   $\{\square\}$

There is no satisfying assignment.

Exercise 2

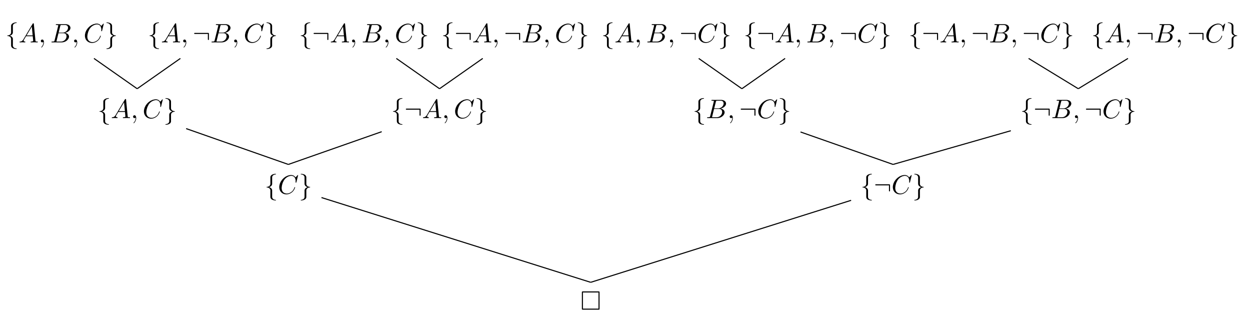
For the two formulas below, use resolution to prove that the formulas are unsatisfiable. To do so, first give a set of clauses Δ that is equivalent to the formula and second, use resolution to prove that it is unsatisfiable. Write the resolution process in the form of a tree for easier readability.

(a) $\phi_1 = (A \vee B \vee C) \wedge (\neg A \vee B \vee C) \wedge (A \vee \neg B \vee C) \wedge (A \vee B \vee \neg C) \wedge (\neg A \vee \neg B \vee C) \wedge (\neg A \vee B \vee \neg C) \wedge (A \vee \neg B \vee \neg C) \wedge (\neg A \vee \neg B \vee \neg C)$

(b) $\phi_2 = (G \vee \neg H) \wedge (\neg G \vee H) \wedge (A \rightarrow B) \wedge (\neg B \vee C) \wedge (A \vee D) \wedge (\neg C \vee \neg A \vee \neg E) \wedge (\neg D \vee A) \wedge F \wedge (F \leftrightarrow E)$

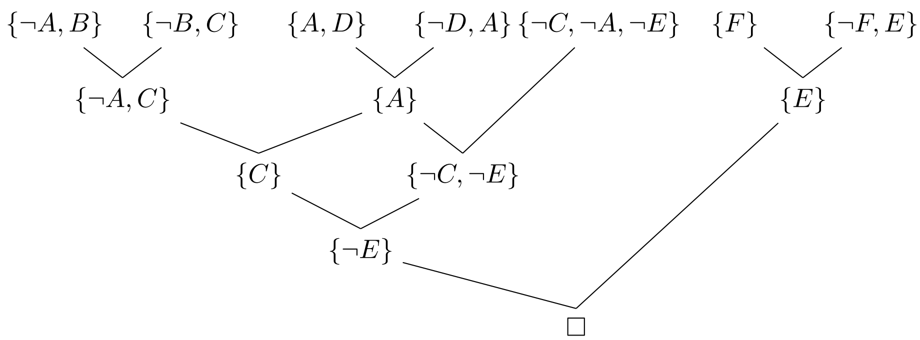
Solution:

(a) $\Delta_1 = \{\{A, B, C\}, \{\neg A, B, C\}, \{A, \neg B, C\}, \{A, B, \neg C\}, \{\neg A, \neg B, C\}, \{\neg A, B, \neg C\}, \{A, \neg B, \neg C\}, \{\neg A, \neg B, \neg C\}\}$



(b) $\Delta_2 = \{\{G, \neg H\}, \{\neg G, H\}, \{\neg A, B\}, \{\neg B, C\}, \{A, D\}, \{\neg C, \neg A, \neg E\}, \{\neg D, A\}, \{F\}, \{\neg F, E\}, \{\neg E, F\}\}$

Resolution tree:



Exercise 3

Transform the following formulas to CNF. To do so, follow the steps from the lecture (Chapter 9 slide 21) and give the intermediate results (you may skip those steps where there is nothing to do for the formula at hand). Simplify the resulting formulas where possible.

(a) $(P \to Q) \leftrightarrow (P \to R)$

(b) $\neg(P \leftrightarrow Q) \vee (Q \to R)$

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

Solution:

(a) 1. Eliminate $\leftrightarrow$: $((P \to Q) \to (P \to R)) \wedge ((P \to R) \to (P \to Q))$
   2. Eliminate $\to$: $(\neg(\neg P \vee Q) \vee (\neg P \vee R)) \wedge (\neg(\neg P \vee R) \vee (\neg P \vee Q))$
   3. Move $\neg$ inwards: $((P \wedge \neg Q) \vee \neg P \vee R) \wedge ((P \wedge \neg R) \vee \neg P \vee Q)$
   4. Distribute $\vee$ over $\wedge$: $(\neg P \vee \neg Q \vee R) \wedge (\neg P \vee Q \vee \neg R)$
   CNF: $(\neg P \vee \neg Q \vee R) \wedge (\neg P \vee Q \vee \neg R)$

(b) 1. Eliminate $\leftrightarrow$: $\neg((P \to Q) \wedge (Q \to P)) \vee (Q \to R)$
   2. Eliminate $\to$: $\neg((\neg P \vee Q) \wedge (\neg Q \vee P)) \vee \neg Q \vee R$
   3. Move $\neg$ inwards: $(P \wedge \neg Q) \vee (Q \wedge \neg P) \vee \neg Q \vee R$
   4. Distribute $\vee$ over $\wedge$: $\neg P \vee \neg Q \vee R$
   CNF: $\neg P \vee \neg Q \vee R$

(c) 1. Elimination of equivalence: nothing to do
   2. Elimination of implication: $(\neg P \vee R) \wedge (\neg Q \vee R) \wedge \neg(\neg Q \wedge (\neg R \vee P))$
   3. Move negation inwards: $(\neg P \vee R) \wedge (\neg Q \vee R) \wedge (Q \vee (R \wedge \neg P))$
   4. Distribute $\vee$ over $\wedge$: $(\neg P \vee R) \wedge (\neg Q \vee R) \wedge (Q \vee R) \wedge (Q \vee \neg P)$
   CNF: $(\neg P \vee R) \wedge (\neg Q \vee R) \wedge (Q \vee R) \wedge (Q \vee \neg P)$

Exercise 4

Consider the formulas in Exercise 3. Are they satisfiable? Draw the truth table and give the number of interpretations that satisfy the formula.

Solution:

$$\gamma = (\neg P \vee \neg Q \vee R) \wedge (\neg P \vee Q \vee \neg R)$$

P	Q	R	$(\neg P \vee \neg Q \vee R)$	$(\neg P \vee Q \vee \neg R)$	γ
$\bot$	$\bot$	$\bot$	$\top$	$\top$	$\top$
$\bot$	$\bot$	$\top$	$\top$	$\top$	$\top$
$\bot$	$\top$	$\bot$	$\top$	$\top$	$\top$
$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$	$\top$	$\top$	$\top$
$\top$	$\bot$	$\top$	$\top$	$\bot$	$\bot$
$\top$	$\top$	$\bot$	$\bot$	$\top$	$\bot$
$\top$	$\top$	$\top$	$\top$	$\top$	$\top$

$\gamma = \neg P \vee \neg Q \vee R$

P	Q	R	γ
$\bot$	$\bot$	$\bot$	$\top$
$\bot$	$\bot$	$\top$	$\top$
$\bot$	$\top$	$\bot$	$\top$
$\bot$	$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$	$\top$
$\top$	$\bot$	$\top$	$\top$
$\top$	$\top$	$\bot$	$\bot$
$\top$	$\top$	$\top$	$\top$

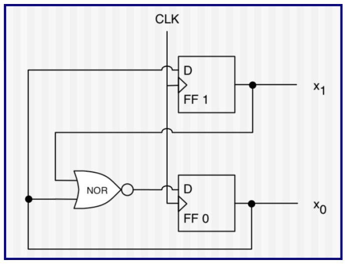
$\gamma = (\neg P \vee R) \wedge (\neg Q \vee R) \wedge (Q \vee R) \wedge (Q \vee \neg P)$

P	Q	R	$(\neg P \vee R)$	$(\neg Q \vee R)$	$(Q \vee R)$	$(Q \vee \neg P)$	γ
$\bot$	$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\top$	$\bot$
$\bot$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$
$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\top$	$\top$	$\bot$
$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$
$\top$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\bot$	$\bot$
$\top$	$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\top$	$\bot$
$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$

The formula is satisfiable and has 3 satisfying assignments.

Exercise 5

Encode the example in the slides of Chapter 7 as a logical formula to verify that if the counter is initialized in the range 0-2, it cannot transition to the value 3.



- Counter, repeatedly from $c = 0$ to $c = 2$.
- 2 bits x_1 and x_0; $c = 2 * x_1 + x_0$.
- ("FF" Flip-Flop, "D" Data IN, "CLK" Clock)

$\rightarrow$ The circuit simply repeats the operations: $x_0 \leftarrow NOR(x_0, x_1) = \neg(x_0 \vee x_1)$; and $x_1 \leftarrow x_0$. The clock is there so that both operations happen simultaneusly, so x_0 and x_1 are updated at the same time.

Hint: you need to refer to two different states of the counter (before and after an operation), so you'll need different propositions for those.

Hint: If you want to prove that given your knowledge encoded as a formula ϕ_K, some statement ϕ_S is true, then you need to encode the formula $\phi = \phi_K \wedge \neg\phi_S$. If ϕ is unsatisfiable, then it means that if we assume ϕ_K, ϕ_S must be true (as $\neg\phi_S$ leads to contradiction).

Solution:

We encode the formula in term of propositions x_0, x_1, x'_0, x'_1, which encode the state of the circuit before and after an operation.

We know that:

- The value of the circuit is initially in the range 0-2: $\neg(x_0 \wedge x_1)$
- The value of the circuit is updated with $x_0 \leftarrow \neg(x_0 \vee x_1)$: $x'_0 \leftrightarrow \neg(x_0 \vee x_1)$
- The value of the circuit is updated with $x_1 \leftarrow x_0$: $x'_1 \leftrightarrow x_0$

We want to prove that $\neg(x'_0 \wedge x'_1)$. So, our full formula is: $(\neg(x_0 \wedge x_1)) \wedge (x'_0 \leftrightarrow \neg(x_0 \vee x_1)) \wedge (x'_1 \leftrightarrow x_0) \wedge (x'_0 \wedge x'_1)$

Note: The encoding above is just one possibility. For example, one can also use propositions $c_0, \ldots, c_3, c'_0, \ldots, c'_3$ representing the value of the counter, and add clauses making sure they have the right values such as $c_0 \leftrightarrow \neg x_0 \wedge \neg x_1$). This however, would not scale if the counter can have millions of values