1 Adversarial Search and Games

1.1 Minimax Search

a) Compute Minimax-Value

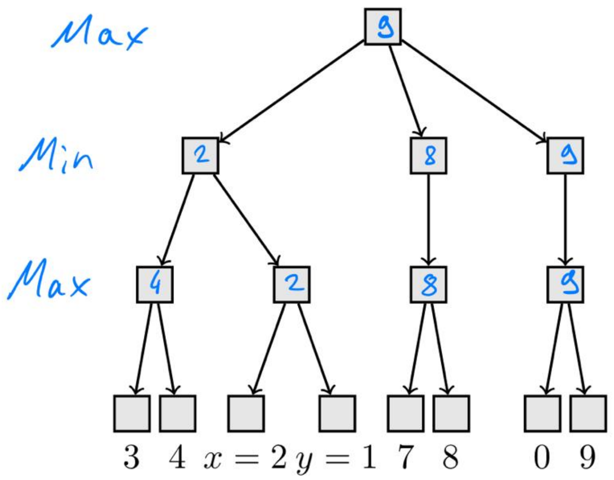
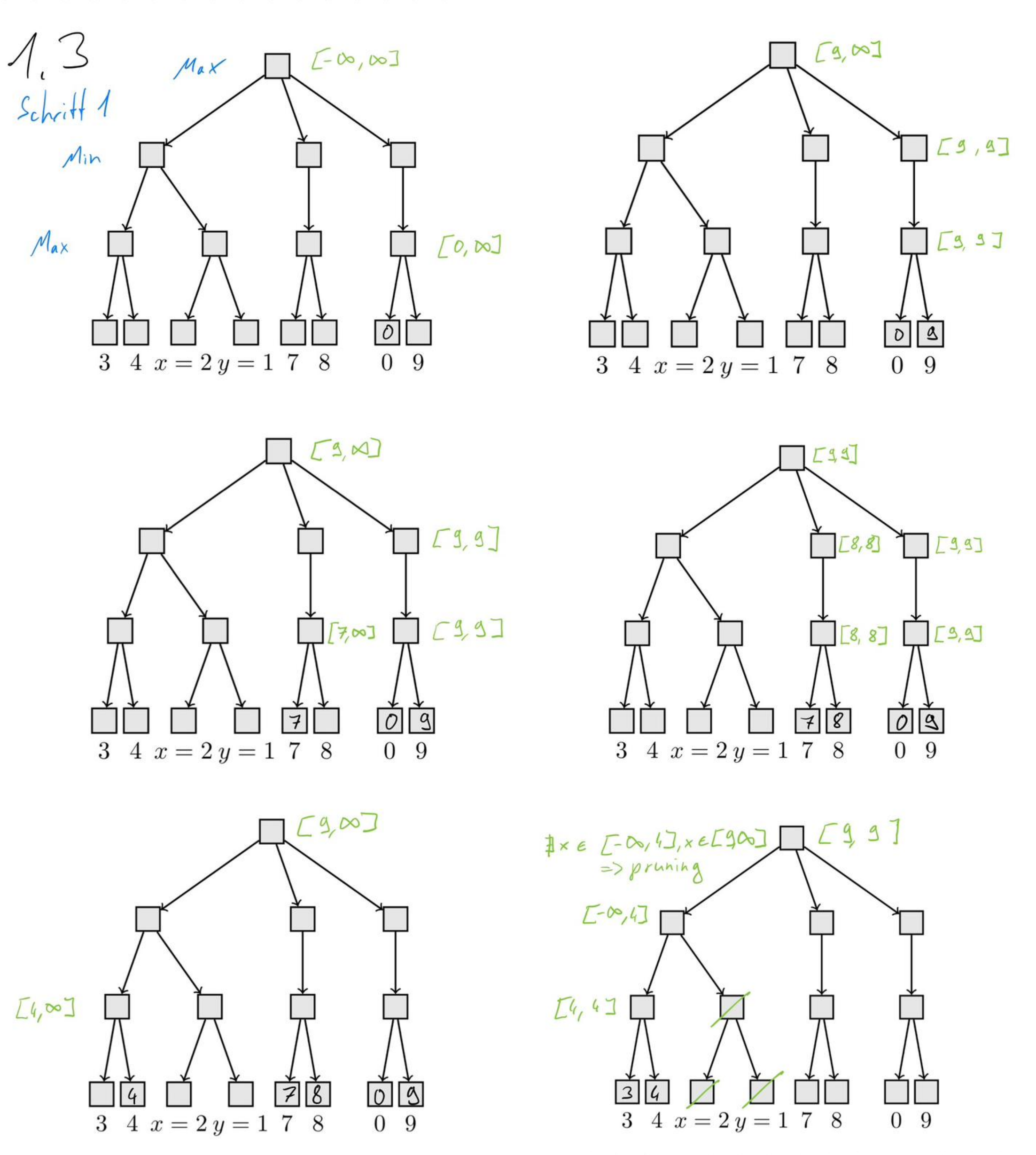


Figure 1: Minimax Search Tree

b) Independency of value from x and y

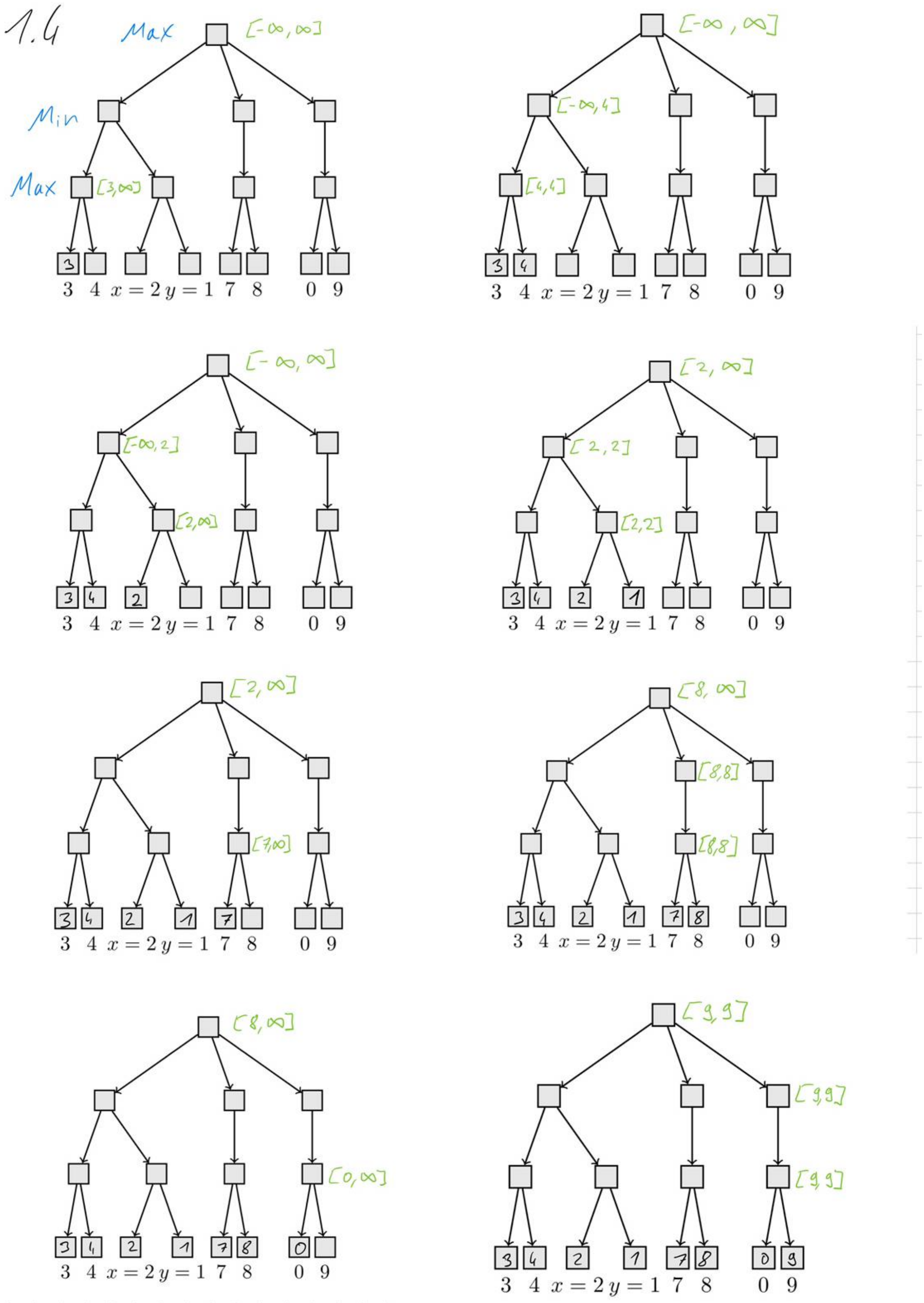
The leafs of x and y don't have any impact on the root node's value, because of the structure of the tree. To change the value of the tree node, a child of the root must have a higher value than 9, otherwise thats the chosen max value. But the left child of the root will always evaluate to 4, if the parent of x and y doesn't have a lower value than that. To have an impact on the root node, x or y would have to be a value which is greater than 9 und lower than 4, which is impossible.

c) Pruning with optimal move order



Using this order will result in 3 nodes being pruned, with two of them being leaf nodes.

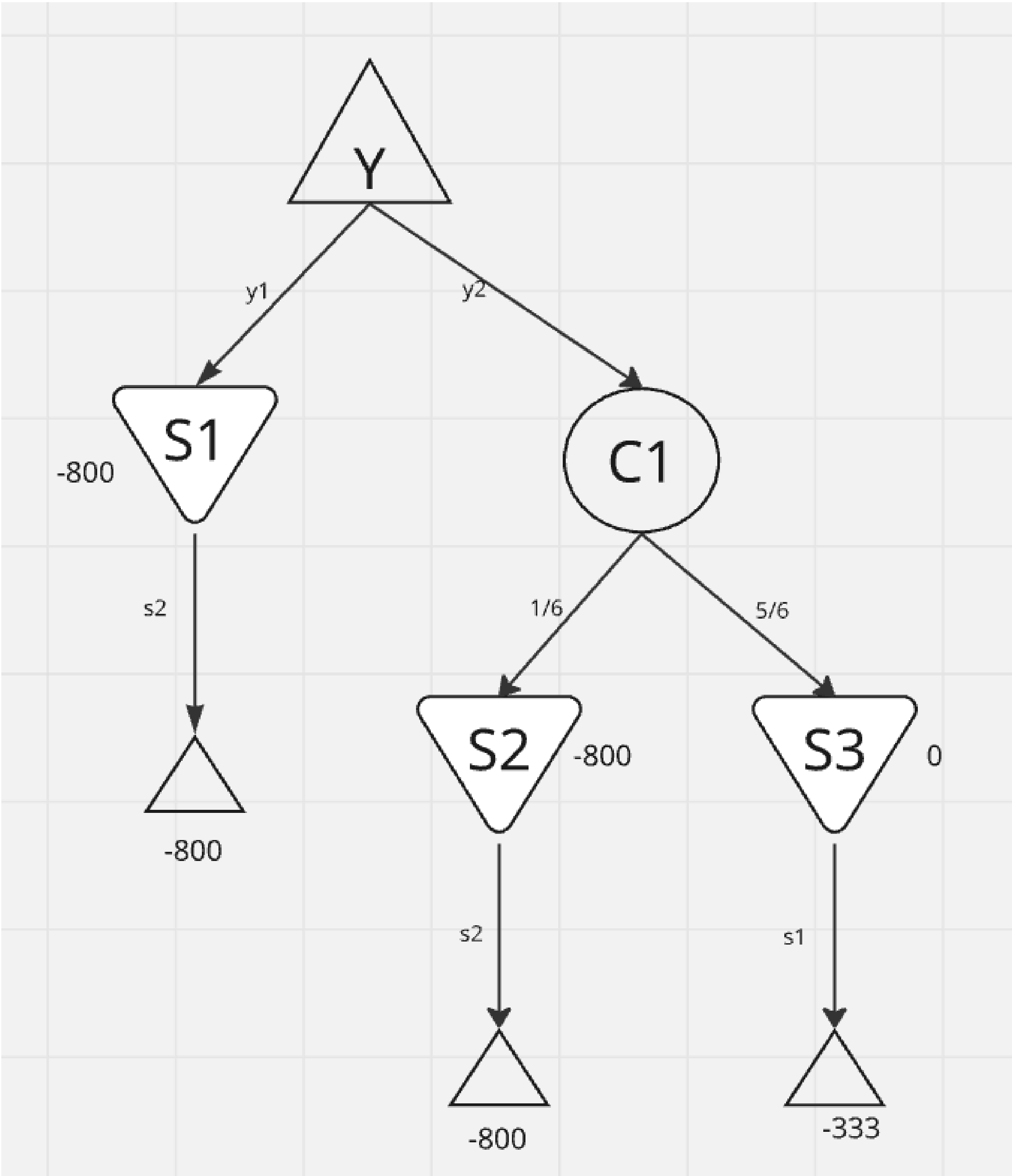
d) Pruning with worst move order



Using this order, none of the nodes can be pruned.

1.2 Games of Chance

a) Resulting Search Tree



b) Expected Minimax Value

y1s1 -> Y = -800 S = -666.67
y1s2 -> S = 0
Therefore, if Y choses y1, then S will choose s2.
y2s1 -> Y = 0 S = -25
y2s2 -> S = 0

Oliver Löhr (3311903) Bachelor Informatik
Lara Aziz (3720604) Bachelor Informatik
Joel Thomas (3814387) Master Computer Science

If, Y choses y2, then S will likely choose s1, as there is a possibility that Y will land in Park Lane in the next round, hence gaining much more.
Then, Y = -333.33
For Y, y1 gives -800, while y2 gives -333.33. So y2 is selected.
Therefore, y2s1
Expected minimax value = -333.33 - (-25) = -308.33

c) Number of Visits and Upper Confidence Bound

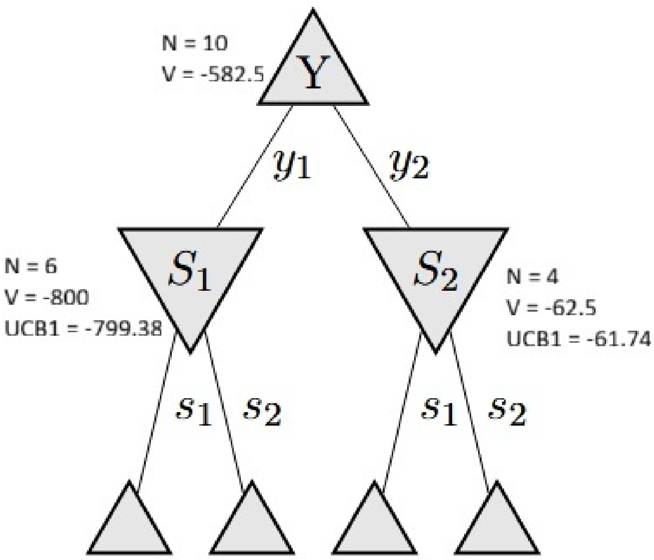
To find: the number of visits(N) and the UCB1 value for each node.
The UCB1 formula for each node i is:

$$UCB1_i = V_i + C\sqrt{lnN_p/N_i}$$

where,
V_i: average observed outcome
n_p: total visit count of parent node
n_i: total visit count of node i

Path	Outcomes	N_{path}	V_{path}(Average)
y_1s_1	-1250, -1300, -1175	3	-1241.67
y_1s_2	-800,-800, -800	3	-800
y_2s_1	-400, +50	2	-175
y_2s_2	-50, -75	2	-62.5

$N_y = 10$
Intermediate node values and visits
Node S_1: $V_{S_1} = -800$ $N_{S_1} = 6$
Node S_2: $V_{S_2} = -62.5$ $N_{S_2} = 4$
Node Y: $V_Y = -582.5$ $N_Y = 10$
$UCB_{y_1} = -799.38$
$UCB_{y_2} = -61.74$



1.3 Correctness of $\alpha - \beta$-pruning

a)

For a max-node the true minimax value is the maximum over all children. If the best value seen so far is γ, then the final minimax value val(n_0) satisfies,

$$val(n_0) \geq \gamma$$

b)

For each node n_i its siblings are denoted $\{n_{i1}$,.....,$n_{ib_i}\}$ and the parent relation is $n_i \in children(\mathrm{n}_{i-1})$. The type (min/max) alternates along the path.
If n_i is a min-node:

$$n_i = min(n_{i+1}, n_{i1}, n_{i2}, ..., n_{ib_i})$$

If n_i is a max-node:

$$n_i = max(n_{i+1}, n_{i1}, n_{i2}, ..., n_{ib_i})$$

c)

If n_i is a min-node:

$$n_i = min(l_i, n_{i+1}, r_i)$$

If n_i is a max-node:

$$n_i = max(l_i, n_{i+1}, r_i)$$

d)

Because n_1 is a min-node, one immediate upper bound on the final n_1 comes from its already-seen left-children, l_1.

$$n_i = min(l_i, (otherchildren))$$

So if the value contributed by the path is $\leq l_1$, then the minimum will remain l_1 and the path cannot lower n_1.
Because the path alternates min/max and n_j is a max-nod, the min-ancestors along the path are exactly 1,3,5,...,j-1.
Therefore if n_j is to affect n_1, then:

$$n_j < min(l_1, l_3, l_5, ..., l_{j-1})$$

2 Propositional Logic

2.1 Who is lying?

We define the variables:

- J: John tells the truth.
- P: Peter tells the truth.
- E: Emma tells the truth.

(a)

The statements:
John says: 'Peter always lies.'
This means J is true if and only if P is false. Formula: $(J) \leftrightarrow \neg(P)$
Peter says: 'Either John is a liar or Emma is a liar, but not both.'
There are two possibilities for this statement to be true:

- Possibility 1: John lies ($\neg J$) AND Emma tells the truth (E).
- Possibility 2: John tells the truth (J) AND Emma lies ($\neg E$).
- We connect these two possibilities with an OR.

Formula: $P \leftrightarrow ((\neg J \wedge E) \vee (J \wedge \neg E))$
Emma says: 'If John is a liar, then Peter is also a liar.'
This is a classical implication.
Formula: $E \leftrightarrow (\neg J \rightarrow \neg P)$

(b)

The set of formulae is:

$$1. \quad J \leftrightarrow \neg P \tag{1}$$
$$2. \quad P \leftrightarrow ((\neg J \vee \neg E) \wedge \neg(\neg J \wedge \neg E)) \tag{2}$$
$$3. \quad E \leftrightarrow (\neg J \rightarrow \neg P) \tag{3}$$

Goal: Find a truth assignment that satisfies all three.

Step 1: Analysis of John's formula

Formula: $J \leftrightarrow \neg P$
Using the definition $A \leftrightarrow B \equiv (A \rightarrow B) \wedge (B \rightarrow A)$, we have:

$$(J \leftrightarrow \neg P) \equiv (J \rightarrow \neg P) \wedge (\neg P \rightarrow J)$$

Now rewrite the implications:

$$J \rightarrow \neg P \equiv \neg J \vee \neg P$$
$$\neg P \rightarrow J \equiv P \vee J$$

So John's statement becomes:

$$(\neg J \vee \neg P) \wedge (P \vee J)$$

From this, we see the required relation:

$$\boxed{J = \neg P}$$

Step 2: Analysis of Emma's formula

Formula: $E \leftrightarrow (\neg J \rightarrow \neg P)$
Rewrite the implication using $A \rightarrow B \equiv \neg A \vee B$:

$$\neg J \rightarrow \neg P \equiv J \vee \neg P$$

Now substitute John's relation $J = \neg P$:

$$J \vee \neg P \equiv (\neg P) \vee \neg P \equiv \neg P$$

$$E \leftrightarrow \neg P$$

We now know:

$$J = \neg P \quad \text{and} \quad E = \neg P$$

Therefore:

$$\boxed{J = E}$$

Interim result: John and Emma must have the same truth value.

Step 3: Analysis of Peter's formula

Peter states his 'one lies but not both' condition:

$$P \leftrightarrow ((\neg J \vee \neg E) \wedge \neg(\neg J \wedge \neg E))$$

3.1 Substitute $J = E$
Since we know $J = E$, then $\neg J = \neg E$. Thus the formula becomes:

$$P \leftrightarrow ((\neg J \vee \neg J) \wedge \neg(\neg J \wedge \neg J))$$

Simplify step by step:

(a) $\neg J \vee \neg J = \neg J$

(b) $\neg J \wedge \neg J = \neg J$

(c) $\neg(\neg J) = J$

Therefore Peter's statement reduces to:

$$P \leftrightarrow (\neg J \wedge J)$$

3.2 Simplify the contradiction
The term $(\neg J \wedge J)$ is a contradiction, meaning it is always false:

$$\neg J \wedge J \equiv \text{false}$$

So:

$$P \leftrightarrow \text{false}$$

Thus:

$$\boxed{P = \text{false}}$$

Step 4: Determine J and E

We previously established:

$$J = \neg P \quad \text{and} \quad E = \neg P$$

Since $P = \text{false}$, we conclude:

$$J = \neg\text{false} = \text{true}$$
$$E = \neg\text{false} = \text{true}$$

Final answer

- **John tells the truth**
- **Peter lies**
- **Emma tells the truth**

2.2 Knowledge Bases

We are given the knowledge base:

$$K = \{\, A \vee (B \vee \neg C),\ A \Leftrightarrow B,\ (C \wedge A) \Rightarrow D \,\}.$$

(a)

1. The formula $A \vee (B \vee \neg C)$ simplifies to:

$$A \vee B \vee \neg C.$$

2. The equivalence $A \Leftrightarrow B$ is rewritten as:

$$(A \rightarrow B) \wedge (B \rightarrow A),$$

which becomes:

$$(\neg A \vee B) \ \wedge \ (\neg B \vee A).$$

3. The implication $(C \wedge A) \Rightarrow D$ becomes:

$$\neg(C \wedge A) \vee D = \neg C \vee \neg A \vee D.$$

Thus, the CNF of the entire knowledge base is:

$$(A \vee B \vee \neg C) \ \wedge \ (\neg A \vee B) \ \wedge \ (\neg B \vee A) \ \wedge \ (\neg C \vee \neg A \vee D).$$

(b)

The clauses in the CNF are:

$$\{A \vee B \vee \neg C,\ \neg A \vee B,\ \neg B \vee A,\ \neg C \vee \neg A \vee D\}.$$

As sets of literals:

$$\Big\{\{A, B, \neg C\},\ \{\neg A, B\},\ \{\neg B, A\},\ \{\neg C, \neg A, D\}\Big\}.$$

(c)

A definite clause contains exactly one positive literal.
Checking each clause:

- $A \lor B \lor \neg C$: two positive literals $\to$ not definite.
- $\neg A \lor B$: one positive literal $B \to$ definite.
- $\neg B \lor A$: one positive literal $A \to$ definite.
- $\neg C \lor \neg A \lor D$: one positive literal $D \to$ definite.

Thus, the definite clauses are:

$$\neg A \lor B, \qquad \neg B \lor A, \qquad \neg C \lor \neg A \lor D.$$

2.3 Models

Consider the four propositions A, B, C, and D. The total number of possible truth assignments (interpretations) for 4 binary variables is $2^4 = 16$. We need to find how many of these 16 assignments make the following formulae true.

(a) $B \lor \neg C$

This formula depends only on B and C. The variables A and D are 'don't cares' (irrelevant to the truth value, but must be counted).

- **Analyze $B \lor \neg C$:** A disjunction is FALSE only if both parts are false.
  - B is False (0)
  - $\neg C$ is False $\Rightarrow C$ is True (1)

  Combinations for (B, C):
  - $(0,0) \to 0 \lor 1 = 1$ (True)
  - $(0,1) \to 0 \lor 0 = 0$ (False)
  - $(1,0) \to 1 \lor 1 = 1$ (True)
  - $(1,1) \to 1 \lor 0 = 1$ (True)

  So, there are **3** valid assignments for the pair (B, C).
- **Account for A and D:** For each valid combination of B and C, the variables A and D can be either True or False ($2 \times 2 = 4$ variations).
- **Calculation:**
$$\text{Models} = 3 \times 2^2 = 3 \times 4 = 12$$

Answer: 12 models.

(b) $A \land \neg(\neg B \lor C) \land D$

This is a conjunction. For the formula to be True, **every** part must be True.

- **Analyze the components:**
  - Part 1: A must be **True** (1).
  - Part 2: D must be **True** (1).
  - Part 3: $\neg(\neg B \lor C)$ must be **True**.
- **Simplify Part 3 (De Morgan's Law):**

$$\neg(\neg B \lor C) \equiv \neg(\neg B) \land \neg C \equiv B \land \neg C$$

  For this to be True:
  - B must be **True** (1).
  - C must be **False** (0).
- **Calculation:** We have fixed values for all four variables:

$$A = 1, \quad B = 1, \quad C = 0, \quad D = 1$$

  There is only 1 specific assignment that satisfies this.

Answer: 1 model.

(c) $((B \to D) \lor (A \to D)) \land C$

This is a conjunction of a complex term and C.

- **Constraint on C:** Since it is a conjunction $(\cdots \land C)$, C must be **True** (1). This reduces our search space. We effectively assume $C = 1$ and look for satisfying assignments of A, B, D ($2^3 = 8$ possibilities).
- **Analyze the complex term:** Formula: $(B \to D) \lor (A \to D)$
  Rewrite implication ($X \to Y \equiv \neg X \lor Y$):

$$(\neg B \lor D) \lor (\neg A \lor D)$$

  Simplify (Associativity and Idempotence of $\lor$):

$$\neg A \lor \neg B \lor D \lor D \equiv \neg A \lor \neg B \lor D$$

  This formula is a disjunction. It is **True** in all cases *except* when all literals are False.
- **Find the failing case:** The expression $\neg A \lor \neg B \lor D$ is False if:
  - $\neg A$ is False $\Rightarrow A = 1$
  - $\neg B$ is False $\Rightarrow B = 1$

  - D is False $\Rightarrow D = 0$

  There is exactly **1** failing combination for (A, B, D) out of 8 possibilities.

- **Calculation:**
  - Total combinations for (A, B, D): 8
  - Failing combinations: 1
  - Valid combinations: $8 - 1 = 7$

  Since C is fixed to True, we do not multiply further.

Answer: 7 models.