Mock Exam "Algorithms and Computability" F24

Question 1 — 10 Points

Construct a 2-tape DTM accepting the language $\{a^n b^n a^{2n} \mid n \in \mathbb{N} \setminus \{0\}\}$ over the alphabet $\{a, b\}$. To this end, first explain informally how the machine processes an input and then define all components of the machine formally.

Hint: You can use the "stay"-direction.

Question 2 — 15 Points

Is $L = \{\ulcorner M \urcorner \mid 1101 \in L(M)\}$ computable? Give a formal argument.

Question 3 — 5 + 15 + 1 Points

Recall that

$$\mathrm{VC} = \{(G, k) \mid G \text{ is a graph that has a } k\text{-vertex cover}\}.$$

An independent set of a graph $G = (V, E)$ is a set $I \subseteq V$ of vertices such that no two vertices in I are connected by an edge in G. We define

$$\textsc{IndSet} = \{(G, k) \mid G \text{ is a graph that has an independent set with } k \text{ vertices}\}.$$

1. Show $\textsc{IndSet} \in \mathrm{NP}$.
2. Show $\mathrm{VC} \leq_p \textsc{IndSet}$.

   Hint: Draw some small graphs, some vertex cover in it, and find an independent set in the graph. Then, draw some small graphs, some independent set in it, and find a vertex cover in the graph.
3. Is $\textsc{IndSet}$ NP-complete? Explain your answer.

Question 4 — 10 Points

The partition problem

$$\text{PARTITION} = \{(n_1, n_2, \ldots, n_k) \in \mathbb{N}^* \mid \text{ there exists } S \subseteq \{1, 2, \ldots, k\} \text{ such that } \sum_{j \in S} n_j = \sum_{j' \notin S} n_{j'}\}$$

asks whether a sequence $(n_1, n_2, \ldots, n_k)$ of natural numbers can be split into two parts that sum up to the same value.

For example $(3, 1, 1, 2, 2, 1) \in \text{PARTITION}$, as $1 + 1 + 1 + 2 = 3 + 2$, but $(2, 3, 4) \notin \text{PARTITION}$.

Show that PARTITION is expressible as a 0-1 Integer Linear Program.

Question 5 — 10 Points

Apply the backtracking algorithm for the knapsack problem to the following instance: Size $W = 50$, threshold $T = 200$, and the following items:

item	1	2	3	4
weight	30	10	20	10
value	80	60	100	50

Show the tree of partial solutions until a valid solution is found or the algorithm terminates.

Question 6 — 7 + 7 Points

Apply the DPLL algorithm to the following formulas and determine whether they are satisfiable or unsatisfiable.

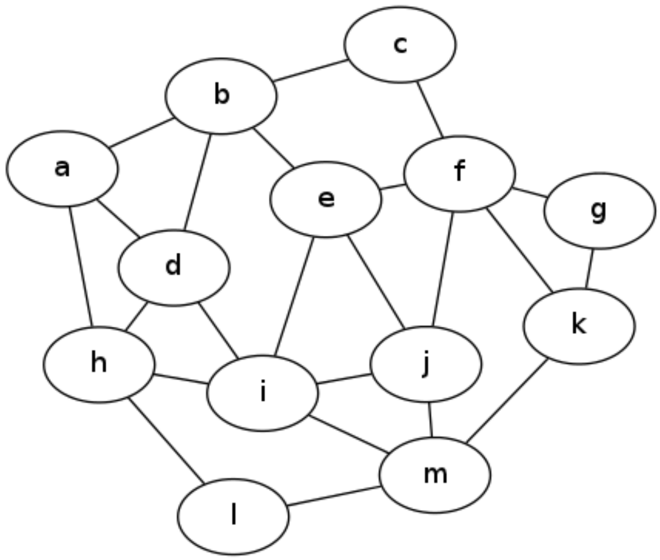
Illustrate the execution of the algorithm using a tree structure, i.e., showing the simplified formula and the rule applied.

1. $\varphi_1 = (\neg A \vee C \vee \neg D) \wedge (A \vee B \vee C \vee \neg D) \wedge (\neg A \vee \neg E) \wedge \neg C \wedge (A \vee D) \wedge (A \vee C \vee E) \wedge (D \vee E)$
2. $\varphi_2 = (A \vee B \vee \neg D) \wedge (\neg B \vee A) \wedge (\neg A \vee B \vee C) \wedge (\neg B \vee \neg A \vee C) \wedge \neg C \wedge (C \vee D)$

Hint: You can use & for $\wedge$, | for $\vee$, and ! for $\neg$ if it is more convenient.

Question 7 10 Points

Apply the approximation algorithm for vertex cover to the following graph.



Show the intermediate value of the tentative cover for each iteration.