Tutorial 1 : Decision problems and Turing machines

Computational complexity theory, 5th semester.

2022

Exercice 1 — *Describe a decision problem*

Describe as a decision problem the following problems. Give the set $\mathcal{L}$ of instances and the question that should be answered.

1. (BIP ?) : Determine if a is graph bipartite.
2. (SAT ?) : Determine how to make a boolean formula true.
3. (3SAT ?) : SAT restricted to conjonctive normal forms where each clause has 3 variables.
4. (ADD ?) : Add two integers.
5. (TAU ?) : Determine if a boolean formula is always true.
6. (3-TAU ?) : TAU restricted to disjonctive normal forms where each clause has 3 variables.
7. (QBF ?) : Determine if a quantified boolean formula is true.
8. (CONNECTIVITY ?) : Connectivity of a graph.
9. (PLANARITY ?) : Planarity of a graph.
10. (k-COL ?) : k-colorability of a graph.
11. (CHROMA) : chromatic number of a graph.
12. (SIZE) : number of nodes in a graph.
13. (INS) : Size of a maximum independant set in a graph.
14. (WINS) : weight of a maximum weighted independant set in a graph.
15. (MATCHING) : Find a maximum matching in a graph.
16. (VERTEX COVER) : Find a set of nodes of minimum size covering all the edges of a graph.
17. (HAM ?) : determine if a graph is hamiltonian.
18. (COHAM ?) : determine if a graph is not hamiltonian.
19. (SHP) : Find a shortest elementary path between two nodes, with non-negative weights.
20. (LOP) : Find a longest elementary path between two nodes, with non-negative weights.
21. (TSP) : In a complete, weighted and undirected graph, find a minimum cost cycle going through each node.
22. (MSPT) : Find a minimum spanning tree in a graph.
23. (SET COVER) : Using sets of a subset of $\mathcal{P}(X)$, find a minimum cover of a set X.
24. (UST) : Find a tree spanning a subset of nodes in a graph and of minimum cost.
25. (DST) : Find a directed tree spanning a subset of nodes in a directed graph rooted at a specific node and of minimum cost.
26. (GRUNDY ?) : determine if a graph has a grundy function.
27. (SUBSET SUM ?) : Determine if a value can be obtained by summing some of the integers of a set.
28. (PARTITION ?) : Determine if a set of integers can be parted into two sets with same sum.
29. (KNAPSACK) : The knapsack problem.
30. (LP) : Solve a linear program with real variables.

31. (ILP) : Solve a linear program with integer variables.
32. (FLOW) : Find a maximum flow in a network.
33. (MREG ?) : Determine if a Markov chain is regular.
34. (CHESS ?) : Determine if the white can force a win from a given position on a chess board.
35. (REVERSI ?) : Determine if the white can force a win from a given position on a reversi board.
36. (SUDOKU ?) : Solve a sudoku.
37. (U-SUDOKU ?) : Unicity of the solution of a sudoku.
38. (MIN SUDOKU) : Determine the minimum number of necessary clues to solve a sudoku.

Exercice 2 — ***Turing machines***

For this exercise, assume that every result you must demonstrate in exercise 3 is true.

1. Describe a deterministic Turing machine that replaces the input integer by a sequence of 1.
2. Describe a deterministic Turing machine that checks if a word is a palindrome.
3. Describe a deterministic Turing machine that solves the problem (ADD ?).
4. Describe a non-deterministic Turing machine that solves the problem (SAT ?).
5. Describe a deterministic Turing machine that solves the problem (SAT ?).
6. Describe a deterministic Turing machine that solves the problem (3-COL ?).
7. Describe a non-deterministic Turing machine that solves the problem (W-INS).

Exercice 3 — ***Equivalent Turing machines***

1. Let $\mathcal{M}$ be a Turing machine with an alphabet Σ, simulate $\mathcal{M}$ with a classical machine $\mathcal{M}'$.
2. Let $\mathcal{M}$ be a Turing machine, simulate $\mathcal{M}$ with a Turing machine $\mathcal{M}'$ that can only read and write 1 and 0.
3. Explain how we can encode the YES and NO answers of a decision problem by a Turing machine with no acceptation or rejection states.
4. Let $\mathcal{M}$ be a Turing machine, simulate $\mathcal{M}$ with a Turing machine $\mathcal{M}'$ that has only a half-tape (infinite on the right, finite on the left).
5. Let $\mathcal{M}$ be a Turing machine with 2 tapes, simulate $\mathcal{M}$ with a classical Turing machine $\mathcal{M}'$
6. Let $\mathcal{M}$ be a Turing machine with a 2D tape, simulate $\mathcal{M}$ with a classical Turing machine $\mathcal{M}'$.
7. Let $\mathcal{M}$ be a non-deterministic Turing machine, simulate $\mathcal{M}$ with a deterministic Turing machine $\mathcal{M}'$.