Monash University
Faculty of Information Technology
2^{nd} Semester 2023

FIT2014
Exercises 11
Complexity: NP-completeness

ASSESSED PREPARATION: Question 1.
You must provide a serious attempt at this entire question at the start of your class.

---

1. A **singleton clause** in a CNF expression is a clause containing only one literal.

Consider the following decision problem.

SINGLETON CLAUSE SAT (abbreviated SCS)
INPUT: a Boolean expression φ in CNF which has at least one singleton clause.
QUESTION: Is φ satisfiable?

(a) Give a polynomial-time reduction from SAT to SINGLETON CLAUSE SAT.

Your main task here is to work out how you can construct, from any CNF expression φ, a new CNF expression φ' which has a singleton clause, such that φ is satisfiable if and only if φ' is satisfiable.

Remember that φ does not necessarily have a singleton clause. The new expression will need to be related to φ but needs to *always* have a singleton clause.

(b) Prove that, for every CNF expression φ,

$$\varphi \in \text{SAT} \iff \varphi' \in \text{SINGLETON CLAUSE SAT}.$$

(c) What else do you need to prove about your reduction from (a), in order to prove that SAT $\leq_P$ SINGLETON CLAUSE SAT?

(d) What else do you need to prove, in order to prove that SINGLETON CLAUSE SAT is NP-complete?

2. Recall Exercise Sheet 10, Q8, which was about the following problem.

NEARLY SAT
INPUT: Boolean expression φ in CNF.
QUESTION: Is there a truth assignment that satisfies all, or all except one, of the clauses of φ?

In that exercise, we proved that (a) NEARLY SAT belongs to NP, and (b) SAT $\leq_m^p$ NEARLY SAT.

(a) What can we say about NEARLY SAT, given parts (a) and (b) of that exercise?

(b) Suppose we alter the Question in the problem definition so that the truth assignment has to satisfy *all except* k of the clauses, where k is some *fixed* number (i.e., it's fixed in advance, and *not* part of the input). Would the problem still be in NP? Could you still give a polynomial-time reduction from SAT to this new problem? What can we say about the complexity classification of this problem now?

3. This question is from the final exam in 2014.

A **Hamiltonian path** in a graph G is a path that includes every vertex of G. All the vertices on the path must be distinct.

A **Hamiltonian circuit** in a graph G is a circuit that includes every vertex of G. All the vertices on the circuit must be distinct.

The decision problems HAMILTONIAN PATH and HAMILTONIAN CIRCUIT are defined as follows:

HAMILTONIAN PATH
INPUT: Graph G.
QUESTION: Does G have a Hamiltonian path?

HAMILTONIAN CIRCUIT
INPUT: Graph G.
QUESTION: Does G have a Hamiltonian circuit?

(a) Prove that HAMILTONIAN CIRCUIT belongs to NP.

(b) Give a polynomial-time reduction (a.k.a. polynomial transformation) from HAMILTONIAN PATH to HAMILTONIAN CIRCUIT.

(c) Assuming you know that HAMILTONIAN PATH is NP-complete, what can you say about HAMILTONIAN CIRCUIT from your solutions to parts (a) and (b)?

4. Recall:

- GRAPH COLOURING $:= \{(G,k) : G$ is a graph, $k \in \mathbb{N}$, and G is k-colourable $\}$. It is NP-complete.
- A graph is **bipartite** if its vertex set can be partitioned into two subsets A and B such that every edge of the graph joins a vertex in A to a vertex in B.

Consider the following variants of GRAPH COLOURING.

(a) BIPARTITE GRAPH COL. $:=$ $\{$ (G,k) : G is a **bipartite** graph, $k \in \mathbb{N}$, and G is k-colourable $\}$

(b) BAD GRAPH COLOURING $:=$ $\{$ (G,k,b) : G is a graph, $k,b \in \mathbb{N}$, and it is possible to assign colours to the vertices of G so that $\le k$ different colours are used and $\le b$ edges are **bad** $\}$

(Here, an edge is **bad** if its endpoints get the same colour.)

(c) GRAPH ODD-COLOURING $:=$ $\{$ (G,k) : G is a graph, $k \in \mathbb{N}$, k is **odd**, and G is k-colourable $\}$

(d) ODD GRAPH COLOURING $:=$ $\{$ (G,k) : G is a graph, $|V(G)|$ is **odd**, $k \in \mathbb{N}$, and G is k-colourable $\}$

For each of (a), (b), (c), (d), determine if it is in P or NP-complete, and justify your answers.

Supplementary exercises

5.

(a) Prove that MOSTLY SAT belongs to NP.

MOSTLY SAT
INPUT: Boolean expression φ in CNF.
QUESTION: Is there a truth assignment that satisfies at least three-quarters of the clauses of φ?

(b) Give a polynomial-time reduction from SAT to MOSTLY SAT, and prove that it is such a polynomial-time reduction.

(c) What can we say about MOSTLY SAT, given (a) and (b)?

6.

Consider the following variants of SATISFIABILITY.

(a)	MONOTONE SAT	:=	{ satisfiable Boolean expressions in Conjunctive Normal Form in which no variables are negated }
(b)	ODD SAT	:=	{ satisfiable Boolean expressions in Conjunctive Normal Form in which every clause has an <u>odd</u> number of literals }
(c)	EVEN SAT	:=	{ satisfiable Boolean expressions in Conjunctive Normal Form in which every clause has an <u>even</u> number of literals }
(d)	DNF-SAT	:=	{ satisfiable Boolean expressions in **Disjunctive** Normal Form }

For each of (a), (b), (c), (d), determine if it is in P or NP-complete, and justify your answers.

The following information relates to Questions 9–12.

Before attempting these questions, it is recommended that you revise previous Tutorial exercises about reduction to SATISFIABILITY. These are: Tutorial 1, Q10; Tutorial 5, Q9; Tutorial 6, Q9. If your solutions to any of these were not in Conjunctive Normal Form (CNF), then think about how you would convert them to CNF. See also the notes by FIT2014 tutor Chris Monteith (available under week 3).

Suppose you have a set $T \subseteq A \times A \times A$ of triples, with each element of each triple belonging to some set A of n elements. A *three-dimensional matching* is a subset, $S \subseteq T$, consisting of exactly n triples from T, such that all members of S are disjoint (i.e., if (a, b, c) and (d, e, f) are both in S then $a \neq d$, $b \neq e$ and $c \neq f$).

These requirements ensure that every member of A appears as a *first* member of some triple in S, and also as the *second* member of some triple in S, and also as the *third* member of some triple in S.

For example, suppose that $A = \{1, 2\}$, and $T = \{(1, 1, 1), (1, 1, 2), (1, 2, 1), (2, 2, 1)\}$. Then T has the 3D matching $S = \{(1, 1, 2), (2, 2, 1)\}$.

If, instead, $T = \{(1, 1, 1), (1, 1, 2), (1, 2, 1), (2, \mathbf{1}, 1)\}$, then T has no 3D matching.

Let 3DM be the language consisting of every set T (consisting of triples) that has a 3D matching.

7. Prove that 3DM belongs to NP.

8.

Give a polynomial-time reduction (a.k.a. polynomial transformation) from 3DM to SATISFIABILITY.

This should be described as an algorithm.

Show that it runs in polynomial-time.

9.

Now suppose that $T \subseteq A \times A \times A \times A$ is a set of *4-tuples*, where again $|A| = n$. A *four-dimensional matching* is a subset, $S \subseteq T$, consisting of exactly n *4-tuples* from T, such that all members of S are disjoint.

Let 4DM be the language consisting of those sets T of 4-tuples that have a 4D matching.

Give a polynomial-time reduction from 3DM to 4DM.

Prove that it is a polynomial-time reduction.

10.

For any fixed positive integer k, we can define the language kDM along similar lines: T is a set of k-tuples $(x_1, x_2, \ldots, x_k)$, where each $x_i \in A$ and $|A| = n$; a k-*dimensional matching* is a set of n disjoint members of T; and kDM is the set of all such T that have a k-dimensional matching.

Suppose you have proved that, for all $k \geq 1$, there is a polynomial-time reduction from kDM to $(k + 1)$DM. Use induction to prove that, for all $k \geq 1$ and all $\ell \geq k$, there is a polynomial-time reduction from kDM to $\ell\,$DM.