Assignment 6

Winter 2017-2018

Due at 23:59 on Sunday 14–01–2018

Requirements: Solutions should be submitted using Canvas. Please include your name and student number on the first page. All solutions should be typed **individually**—copying any part of your text is strictly forbidden. When you are asked to give an algorithm, it is generally allowed to describe the algorithm in natural text. When you are explicitly asked for pseudocode, you must supply pseudocode. Always **explain** any pseudocode that you present.

Exercises for NP-hardness

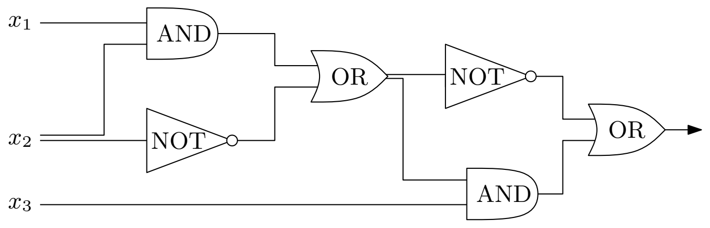


Figure 1: A Boolean circuit.

1. (1 point) Consider the Boolean circuit depicted in Fig. 1. Suppose we transform this circuit into a Boolean formula with the reduction algorithm used in the proof to show that CIRCUIT-SAT $\leq_P^T$ FORMULA-SAT. What is the resulting Boolean formula? Only the final answer is required.

   (NB: In your formula you are allowed to use the logical connectives $\vee$, $\wedge$, $\rightarrow$, $\leftrightarrow$, and $\neg$.)

2. (1 point) A Boolean formula is in *Eindhoven normal form* (ENF) if it is a disjunction (big OR) of a list of clauses, where each clause is a conjunction (big AND) of literals. A literal is a variable x_i or its negation $\neg x_i$. For example: the formula

   $$(x_1 \wedge x_3 \wedge \neg x_4) \vee (\neg x_1 \wedge x_2) \vee (x_1 \wedge \neg x_3 \wedge \neg x_4)$$

   is in ENF, but the formula $x_1 \rightarrow (x_2 \wedge (x_3 \leftrightarrow \neg x_4))$ is not. Consider the following problem:

   ENF-SAT
   Input: A Boolean ENF formula.
   Output: YES, if there is an assignment to the variables that makes the formula evaluate to *true*. NO, otherwise.

   You face this problem on your first day at work for a consulting company. Your boss would like you to find an efficient algorithm to solve it, but you realize that it is similar to CIRCUIT-SAT, for which no polynomial-time algorithm is believed to exist. Investigate the complexity of the problem, and do exactly one of the following two things to satisfy your boss:

   - Show that ENF-SAT can be solved in polynomial time, by describing an efficient algorithm and why it is correct. (Not longer than half a page.)
   - Show that ENF-SAT is NP-hard, by establishing that CIRCUIT-SAT $\leq_P^T$ ENF-SAT. (Not longer than half a page.)

Hint: In such a situation, start by evaluating some examples to get a feeling for the problem. Make some examples with the answer YES, and some examples with the answer NO.

3. ($1 + 1$ points) Consider the following two computational problems:

   HAMILTONIAN CYCLE CONSTRUCTION
   Input: An undirected graph $G = (V, E)$.
   Output: A simple cycle in G that visits each vertex exactly once, which is called a *Hamiltonian cycle*, or the answer NO if such a cycle does not exist.

   LONGEST SIMPLE PATH
   Input: An undirected graph $G = (V, E)$.
   Output: A longest simple path in G.

   The goal of this exercise is to prove that HAMILTONIAN CYCLE CONSTRUCTION $\leq_P^T$ LONGEST SIMPLE PATH: a polynomial-time algorithm to find a *longest simple path*, can be used to find a *Hamiltonian cycle* in polynomial time (if one exists).

   (a) Describe how the HAMILTONIAN CYCLE CONSTRUCTION problem can be solved in polynomial time, if one has a hypothetical subroutine available that solves LONGEST SIMPLE PATH in constant time.

   Hint: Starting from an input $G = (V, E)$ to HAMILTONIAN CYCLE CONSTRUCTION, turn G into a graph G' by making some local changes. Then run the hypothetical algorithm for LONGEST SIMPLE PATH on G', to learn something about the existence of Hamiltonian cycles in G. Keep in mind that G may have a simple *path* that visits every vertex exactly once, but may not have a simple *cycle* that visits every vertex exactly once.

   (b) Prove the correctness of your reduction, in two steps. (i) Show that whenever G has a Hamiltonian cycle, then your algorithm will output such a cycle; and (ii) prove that whenever your algorithm gives an output (other than NO), then that output is indeed a Hamiltonian cycle in G.

4. ($\frac{1}{2} + \frac{1}{2}$ point) Consider the reduction from 3-SAT to SUBSET SUM explained in the slides. In the reduction, two $(n + m)$-digit numbers are generated for each clause C_j: a number $c_{j,1}$ where the $(n + j)$-th digit (counted from the left) is 1 and all other digits are 0, and a number $c_{j,2}$ where the $(n + j)$-th digit is 2 and all other digits are 0.

   (a) Person A claims that the proof would also have worked if we had generated only one number for C_j, namely the number where the $(n + j)$-th digit is 3 and all other digits are 0. Is Person A right? Explain your answer in at most 5 lines.

   (b) Person B claims that the proof would also have worked if we had generated three numbers for C_j, each equal to the number c_j where the $(n + j)$-th digit is 1 and all other digits are 0. (Note that this generates an instance of SUBSET SUM where the set X contains certain numbers three times, so X is actually a multi-set. This is, in fact, allowed in SUBSET SUM.) Is Person B right? Explain your answer in at most 5 lines.

5. (1 point) In the lecture, we proved that MAXIMUM CLIQUE is NP-hard by showing that 3-SAT $\leq_P^T$ MAXIMUM CLIQUE. The reduction transforms a 3-SAT formula F with k clauses into a graph $G = (V, E)$, such that F is satisfiable if and only if G has a clique of size k. In this exercise we study the properties of the reduction in more detail.

   Complete the following lemma and prove it.

   Lemma 1 *Consider a 3-SAT formula F, and the graph $G = (V, E)$ that is constructed by the reduction. Then the maximum size of a clique in G equals the $(\ldots)$ of formula F.*

6. ($\frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \frac{1}{2}$ points) For each of the following problems, state whether they are NP-hard or polynomial-time solvable. Explain your answer in at most five lines.

   (a) Clique in Bipartite Graphs
   Input: An undirected graph $G = (V, E)$ that is guaranteed to be bipartite: the vertices V can be partitioned into two disjoint sets L and R, such that each edge connects one vertex in L to one vertex in R.
   Output: A maximum clique in G.

   (b) Shortest Simple $s - t$ Path in Unweighted Graphs
   Input: An unweighted undirected graph $G = (V, E)$ and distinguished vertices s and t.
   Output: A simple path from s to t with a minimum number of edges, or No if there is no path from s to t.

   (c) Shortest Simple $s - t$ Path in Weighted Graphs
   Input: An undirected graph $G = (V, E)$ with integer weights on the edges (potentially negative), and two distinguished vertices s and t.
   Output: A simple path from s to t with minimum total weight, or No if there is no path from s to t.

   Hint: Use the results of other exercises in this homework set.

   (d) Cheapest Disconnection
   Input: A directed graph $G = (V, E)$ with non-negative integer weights on the edges, and distinguished vertices s and t.
   Output: A minimum-weight subset of the edges $S \subseteq E$ whose removal disconnects s from t: in the graph $G' = (V, E \setminus S)$ there is no path from s to t.

7. (2 points) Consider the following problem, which is also illustrated in Fig. 2.

   Rectangle Packing
   Input: A set $R = \{r_1, r_2, \ldots, r_n\}$ of rectangles and another rectangle b. Each rectangle is given by its width and height, which are integral.
   Output: Yes, if there is a subset $R' \subseteq R$ such that the rectangles from R' can be placed inside b (without overlap) such that they exactly fill up b. (It is not allowed to rotate the rectangles.) No, otherwise.

(i)	$r_1 = 4 \times 2$	$r_6 = 4 \times 4$	$b = 16 \times 8$
	$r_2 = 6 \times 4$	$r_7 = 9 \times 2$	
	$r_3 = 1 \times 7$	$r_8 = 5 \times 8$	
	$r_4 = 3 \times 8$	$r_9 = 5 \times 2$	
	$r_5 = 7 \times 4$	$r_{10} = 9 \times 6$	

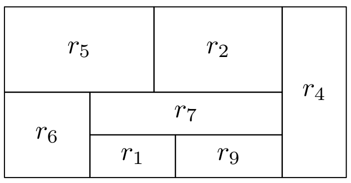
(ii)


Figure 2: (i) An instance of Rectangle Packing. (ii) A packing of a subset (namely $\{r_1, r_2, r_4, r_5, r_6, r_7, r_9\}$) showing that the instance shown in (i) is a "yes"-instance.

Prove that Rectangle Packing $\in$ P or prove that Rectangle Packing is NP-hard. (If you prove that Rectangle Packing is NP-hard, you should do so by giving a reduction from one of the problems on the summary slides, and proving the correctness of your reduction.)