
Collaborating on homework is encouraged, but you must write your own solutions in your own words and list your collaborators. Copying someone else's solution will be considered plagiarism and may result in failing the whole course.

Please answer clearly and concisely. Explain your answers. Unexplained answers will get lower scores or even no credits.

- (1) (20 points) For each of the following problems, say whether it is decidable or not. Justify your answer by describing an appropriate Turing machine, or by reducing from ALL_{CFG} which was shown undecidable in class. Assume that the alphabet of CFG G contains the symbol a .
 - (a) $L_1 = \{\langle G \rangle \mid \text{CFG } G \text{ generates at least two strings that contain the symbol } a\}$.
 - (b) $L_2 = \{\langle G \rangle \mid \text{CFG } G \text{ generates all strings that contain the symbol } a\}$.
- (2) (20 points) For each of the following problems, show that it is NP-complete (namely, (1) it is in NP and (2) some NP-complete language reduces to it.) When showing NP-completeness, you can start from any language that was shown NP-complete in class or tutorial.
 - (a) $L_1 = \{\langle G' \rangle \mid \text{Graph } G' \text{ on } n' \text{ vertices contains a vertex cover of size } n'/2\}$
Hint: Reduce from VERTEX COVER. When reducing from an instance $\langle G, k \rangle$ of VERTEX COVER, consider the following three cases separately (here n denotes the number of vertices of G): (1) $k = n/2$, (2) $k < n/2$, (3) $k > n/2$.
 - (b) $L_2 = \{\langle G, k \rangle \mid \text{Graph } G \text{ contains (at least) two independent sets, each of size } k\}$.
- (3) (30 points) Throughout the semester, we looked at various models of computation and we came up with the following “hierarchy” of classes of languages:

$$\text{regular} \subseteq \text{context-free} \subseteq \text{P} \subseteq \text{NP} \quad \text{decidable} \subseteq \text{recognizable}$$

We also gave examples showing that the containments are strict (e.g., a context-free language that is not regular), except for the containment $\text{P} \subseteq \text{NP}$, which is not known to be strict.

There is one gap in this picture between NP languages and decidable languages. In this problem you will fill this gap.

- (a) Show that 3SAT is decidable, and the decider has running time $2^{O(n)}$. (Unlike a *verifier* for 3SAT which is given a 3CNF formula φ together with a potential satisfying assignment for φ , a *decider* for 3SAT is only given a 3CNF formula but not an assignment for it.)
- (b) Argue that for every NP-language L there is a constant c such that L is decidable in time $2^{O(n^c)}$. (Use the Cook–Levin Theorem.)

(c) Let L' be the following language:

$$L' = \{\langle M, w \rangle \mid \text{Turing machine } M \text{ does not accept input } \langle M, w \rangle \text{ within } 2^{2^{|w|}} \text{ steps}\}.$$

It is not hard to see that L' can be decided in time $O(2^{2^n})$.

Show that L' cannot be decided in time $2^{O(n^c)}$ for any constant c , and therefore it is not in NP.

Hint: Assume that L' can be decided by a Turing machine D in time $2^{O(n^c)}$. What happens when D is given input $\langle D, w \rangle$, where w is a sufficiently long string?

(4) (30 points) A *heuristic* is an algorithm that often works well in practice, but it may not always produce the correct answer. In this problem, we will consider a heuristic for IS (independent set).

Recall that the *degree* of a vertex is the number of edges incident to it. In the following, we assume the vertices in the input graph G are labelled from 1 to n . Consider the following heuristic H for IS:

On input $\langle G, k \rangle$, where G is a graph:

Let v_i be the vertex of minimum degree (if there are more than one choice for v_i , break ties by choosing the vertex with the smallest label)

Set $S = \{v_i\}$

Let $N = \{u \in V \setminus S \mid (u, v) \notin E \text{ for all } v \in S\}$ be the set of vertices outside S that are not adjacent to any vertex in S

While N is not empty

Let v_i be the vertex in N of minimum degree (breaking ties by choosing the smallest label)

Update $S = S \cup \{v_i\}$

Update $N = \{u \in V \setminus S \mid (u, v) \notin E \text{ for all } v \in S\}$ to be the set of vertices outside S that are not adjacent to any vertex in S

End While

accept if and only if $|S| \geq k$.

(a) Show that H runs in polynomial time.

(b) Show that if H accepts $\langle G, k \rangle$, then $\langle G, k \rangle \in \text{IS}$.

(c) Show that it is possible that H rejects $\langle G, k \rangle$, even though $\langle G, k \rangle \in \text{IS}$.

Give such an instance $\langle G, k \rangle$ where the graph G contains at most 5 vertices.