

Efficient Algorithms and Programming

Autumn 1999

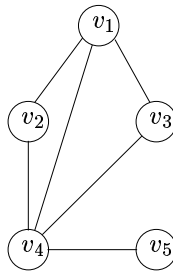
Solution to Homework 11

Problem 1 (E96, 30%)

An *independent set* of a graph $G = (V, E)$ is a subset $V' \subseteq V$ of vertices such that each edge in E is incident on at most one vertex in V' . The *independent-set problem* is to determine a maximum-size independent set in G .

Question 1.1

An independent set of size two for the graph below is $V' = \{v_1, v_5\}$. Find a larger independent set for this graph.



Solution: $V' = \{v_2, v_3, v_5\}$

Question 1.2

Formulate the related decision problem, INDEPENDENT-SET, for the independent-set problem.

Solution:

INDEPENDENT-SET = $\{\langle G, k \rangle : G \text{ is a graph with an independent set of size } k\}$.

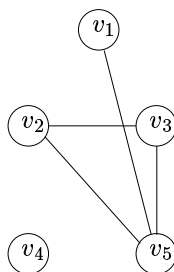
Question 1.3

Show that INDEPENDENT-SET \in NP.

Solution: Given an instance $\langle G, k \rangle$ and a certificate V' , we can verify whether V' is a subset of V , whether each edge of E is incident on at most one vertex in V' , and whether the cardinality of V' is k . These checks can clearly all be performed in polynomial time, thus INDEPENDENT-SET is in NP.

Question 1.4

Given a graph $G = (V, E)$, construct a graph H as the *complement* of G , that is, let H be a graph with vertex set V and edge set $\{(u, v) \mid u, v \in V, u \neq v, \text{ and } (u, v) \notin E\}$. The complement of the above graph is:



Show that G has a clique of size k if and only if H has an independent set of size k .

Solution:

- \Rightarrow : Assume G has a clique V' of size k . From the definition of a clique we know that for all $u, v \in V' : (u, v) \in E$ and thus for all u, v , H will not contain the edge (u, v) . Since there are no edges between any of the vertices in V' , V' is an independent set of size k .
- \Leftarrow : (analogous) Assume H has an independent set V' of size k . From the definition of an independent set we know that for all $u, v \in V'$ there is no edge (u, v) in H and thus for all $u, v \in V'$, $(u, v) \in E$. Clearly V' is a clique of G of size k .

Question 1.5

Complete the proof that INDEPENDENT-SET is NP-complete.

Solution: From (b) we know INDEPENDENT-SET is in NP. From (c) it is clear that we can reduce CLIQUE to INDEPENDENT-SET in polynomial time. Since CLIQUE is known to be NP-complete, it follows from Lemma 36.8 that INDEPENDENT-SET is NP-complete.

Question 1.6

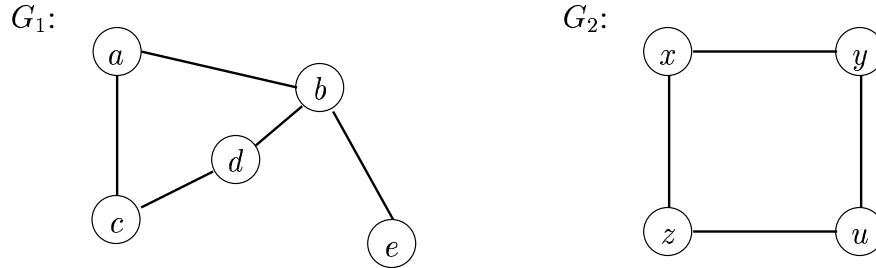
Assume each vertex of G has degree two. Construct a linear time algorithm to find a maximal independent set in G .

Solution: Graphs where all vertices have degree two consists of disjoint sets of sub-graphs where each sub-graph consists of a closed “chain” of vertices. For a chain with n vertices there is an independent set of size $\lfloor \frac{n}{2} \rfloor$. A simple adoption of depth-first search can be used to construct the independent set.

Problem 2 (E97, 20%)

Given two undirected graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, the subgraph-isomorphism problem is to determine whether G_1 contains a subgraph isomorphic to G_2 , that is, to determine whether there exists a subset $V \subseteq V_1$ and a subset $E \subseteq E_1$ such that $|V| = |V_2|$ and $|E| = |E_2|$ and there exists an injective function $f : V_2 \rightarrow V$ satisfying $(u, v) \in E_2$ if and only if $(f(u), f(v)) \in E$.

As an example, consider the two graphs G_1 and G_2 below:



In this example, G_1 contains subgraphs isomorphic to G_2 . One possible subgraph is given by $V = \{a, b, c, d\}$, $E = \{(a, b), (b, d), (d, c), (c, a)\}$, and f given by $f(x) = a$, $f(y) = b$, $f(z) = c$, $f(u) = d$.

The related decision problem, SUBGRAPH-ISOMORPHISM, for the subgraph-isomorphism problem is

SUBGRAPH-ISOMORPHISM = $\{\langle G_1, G_2 \rangle : G_1 \text{ contains a subgraph isomorphic to } G_2\}$.

Question 2.1

Show that SUBGRAPH-ISOMORPHISM is in NP.

Solution: Given an instance $\langle G_1, G_2 \rangle$ and a certificate (V, E, f) , we can verify whether the following conditions are satisfied:

- $V \subseteq V_1$ and $|V| = |V_2|$
- $E \subseteq E_1$ and $|E| = |E_2|$
- $\forall (u, v) \in E_2 : (f(u), f(v)) \in E$
- f is injective.

Each of these checks can be performed in polynomial time, and thus SUBGRAPH-ISOMORPHISM is in NP.

Question 2.2

Show that SUBGRAPH-ISOMORPHISM is NP-hard.

Hint: Consider restricting SUBGRAPH-ISOMORPHISM to cases where G_2 is a complete graph and use that CLIQUE is NP-complete.

Solution: We use a reduction from CLIQUE. Let $\langle G, k \rangle$ be an instance of CLIQUE. The reduction algorithm constructs G_1 and G_2 as follows:

- G_1 is a copy of G .
- G_2 is a complete graph consisting of k vertices.

It is observed that G has a clique of size k if and only if G_1 contains a subgraph isomorphic to G_2 . Since the above construction can be performed in polynomial time, we have that

CLIQUE \leq_P SUBGRAPH-ISOMORPHISM.

Since CLIQUE is NP-complete, it follows from Lemma 36.8 that SUBGRAPH-ISOMORPHISM is NP-hard.

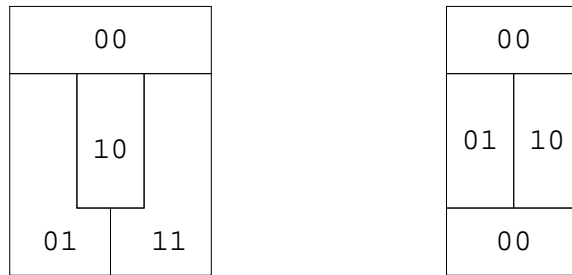
Question 2.3

Prove that SUBGRAPH-ISOMORPHISM is NP-complete.

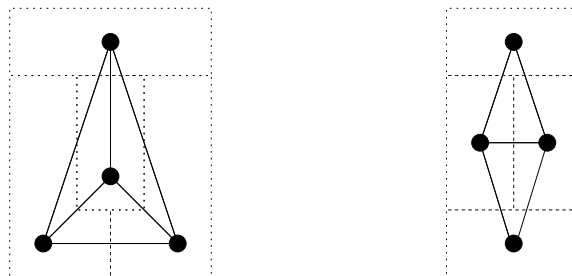
Solution: Follows immediately from 5.1, 5.2, and Lemma 36.8.

Problem 3 (E98, 40%)

In this problem we consider the colouring of maps of countries. It is a famous theorem that any map of countries can be coloured with only *four* colours so that no two adjacent countries have the same colour. Below are shown two maps. The first requires four colours, the second only three. We use the values $\{00, 01, 10, 11\}$ for the four colours.



For the purpose of finding an assignment of colours to the countries, we can model a map as an undirected graph $G = (V, E)$, where the vertices $V = \{1, \dots, n\}$ represent the countries, and the edges $E \subseteq V \times V$ represent the neighbouring relation. The two maps above have the following associated graphs:



We take a colouring of a graph G to be a vector $\vec{c} = \langle c_1, \dots, c_n \rangle$ assigning a colour $c_i \in \{00, 01, 10, 11\}$ to each of the vertices. A *valid colouring* is a colouring \vec{c} of G such that

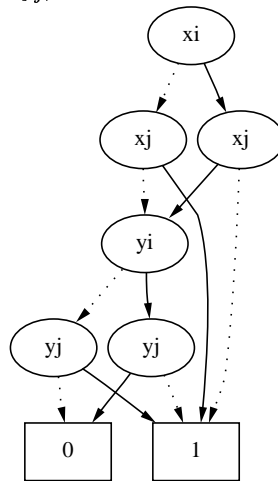
$$\forall (i, j) \in E. c_i \neq c_j.$$

ROBDDs can be used to find valid colourings. For each country i , we use two Boolean variables (x_i, y_i) representing the colour of that country. For instance, if $c_i = 01$ then $(x_i, y_i) = (0, 1)$.

Question 3.1

Write down an expression over the Boolean variables expressing the condition that two given countries i and j have different colours. Draw an ROBDD corresponding to this expression.

Solution: $\neg(x_i \Leftrightarrow x_j \wedge y_i \Leftrightarrow y_j)$:



Question 3.2

Write down a Boolean expression over the variables $x_1, y_1, \dots, x_n, y_n$ expressing that the variables represent a valid colouring of a graph G .

Solution: $\bigwedge_{(i,j) \in E} \neg(x_i \Leftrightarrow x_j \wedge y_i \Leftrightarrow y_j)$

Question 3.3

Describe an algorithm that efficiently finds a colouring and counts the number of possible different colourings in an undirected graph.

Solution: Construct an ROBDD for $\bigwedge_{(i,j) \in E} \neg(x_i \Leftrightarrow x_j \wedge y_i \Leftrightarrow y_j)$ using a number of APPLY calls. Extract a solution with ANYSAT and compute the count with SAT-COUNT.

Another problem associated with colourings is to determine whether three colours is enough, i.e., whether there exists a valid colouring using only three colours. The decision problem can be formulated as follows:

$$3\text{-COL} = \{ \langle G \rangle \mid G \text{ admits a valid colouring } \vec{c} \text{ with only three colours} \} .$$

Question 3.4

Show that 3-COL is in NP.

Solution: Use \vec{c} as certificate, check it by running through each edge in the graph and for each edge check the colouring condition. This is clearly implementable as a polynomial time algorithm.

Professor C. Lever has shown the following three polynomial-time reductions:

- $\text{CIRCUIT-SAT} \leq_P 3\text{-COL}$
- $3\text{-COL} \leq_P 3\text{-CNF-SAT}$
- $2\text{-CNF-SAT} \leq_P 3\text{-COL}$

Question 3.5

Use one of his reductions to show that 3-COL is NP-complete.

Solution: Since 3-COL is in NP, $\text{CIRCUIT-SAT} \leq_P 3\text{-COL}$, and CIRCUIT-SAT is NP-complete, the result follows from lemma 36.8 in [CLR].

Question 3.6

Describe how to use ROBDDs to decide whether there exists a valid colouring using three colours.

Solution: Take the expression for a four-colouring. Conjoin with the expression $\bigwedge_{i=1}^n \neg(x_i \wedge y_i)$ disallowing the use of the fourth colour. If the result is the terminal 0 if and only if there is no three-colouring.