Algorithmics

Spring ’20 - Tutorial #4

Alexander Moch

May 11, 2020

Exercise 4.1

Task

Given two languages $L_1 \subseteq \Sigma_1^*$ and $L_2 \subseteq \Sigma_2^*$ with $L_1 \in \mathcal{P}$ and $\emptyset \neq L_2 \neq \Sigma_2^*$, show: $L_1 \leq_p L_2$.
Would $\mathcal{P} = \mathcal{NP}$ imply that all languages in $\mathcal{P}$ are $\mathcal{NP}$-complete?

Solution

According to the definition of L_2, there exist (arbitrarily chosen) $w_{yes} \in L_2$ and $w_{no} \in \Sigma_2^* \setminus L_2$. We define $f : \Sigma_1^* \to \Sigma_2^*$ as:

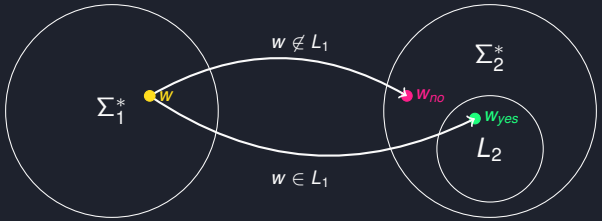
$$f(w) = \begin{cases} w_{yes} & \text{if } w \in L_1, \\ w_{no} & \text{otherwise.} \end{cases}$$

Obviously, for all $w \in \Sigma_1^*$ it holds that: $w \in L_1 \Leftrightarrow f(w) \in L_2$. Now we show that there exists a polynomial-time algorithm M which computes f: For a given input w, M first decides whether $w \in L_1$. This can be done in polynomial time because $L_1 \in \mathcal{P}$. Based on the result, M then computes $f(w)$ according to the definition above.

Solution

According to the definition of L_2, there exist (arbitrarily chosen) $w_{yes} \in L_2$ and $w_{no} \in \Sigma_2^* \setminus L_2$. We define $f : \Sigma_1^* \to \Sigma_2^*$ as:

$$f(w) = \begin{cases} w_{yes} & \text{if } w \in L_1, \\ w_{no} & \text{otherwise.} \end{cases}$$

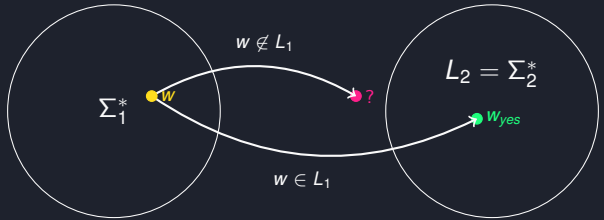


Remember: $L_1 \in \mathcal{P}$.

Solution

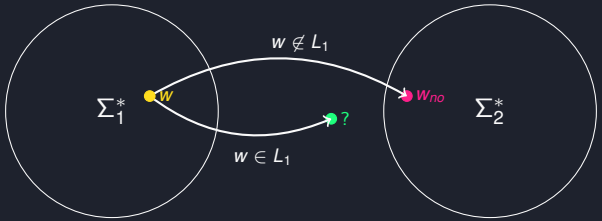
It is now also straightforward to see that each language in $\{L \in \mathcal{P} \mid L \neq \emptyset \wedge \overline{L} \neq \emptyset\}$ is $\mathcal{P}$-complete w.r.t. $\leq_p$ (and, for this reason, also $\mathcal{NP}$-complete if $\mathcal{P} = \mathcal{NP}$).

This is not true for L with $\emptyset \in \{L, \overline{L}\}$. Let $L_1 \subseteq \Sigma_1^*$ with $\emptyset \neq L_1 \neq \Sigma_1^*$ and let $L_2 = \Sigma_2^*$. Then, there cannot exist a polynomial transformation $f : \Sigma_1^* \to \Sigma_2^*$ from L_1 to L_2 as the case $f(w) \notin L_2$ for $w \notin L_1$ would not be available because $\Sigma_2^* \setminus L_2 = \emptyset$. (Analogously for $L_2 = \emptyset$.)



Solution

This is not true for L with $\emptyset \in \{L, \overline{L}\}$. Let $L_1 \subseteq \Sigma_1^*$ with $\emptyset \neq L_1 \neq \Sigma_1^*$ and let $L_2 = \Sigma_2^*$. Then, there cannot exist a polynomial transformation $f : \Sigma_1^* \to \Sigma_2^*$ from L_1 to L_2 as the case $f(w) \notin L_2$ for $w \notin L_1$ would not be available because $\Sigma_2^* \setminus L_2 = \emptyset$. (Analogously for $L_2 = \emptyset$.)



$L_2 = \emptyset$

Exercise 4.2

Task

Show that $\mathcal{NP}$ is closed w.r.t. $\leq_p$, i.e., from $L_1 \leq_p L_2$ and $L_2 \in \mathcal{NP}$ follows $L_1 \in \mathcal{NP}$.

Efficiently Verifiable Proofs

Definition 1

We say that a given decision problem Π has **Efficiently Verifiable Proofs** if there is a proof scheme $(Proof_\Pi, V_\Pi)$ for Π, which is defined as follows

- **Proofs:** $Proof_\Pi$ assigns to each input x for Π a set $Proof_\Pi(x)$ of **possible proofs**.
- **Efficient Verification:** $V_\Pi = V_\Pi(x, y)$ denotes a **decision algorithm** of running time polynomially bounded in $|x|$, which decides for each input x for Π and each possible proof $y \in Proof_\Pi(x)$ if y is a proof for the claim that $\Pi(x) = 1$.
- **Correctness:** It holds that $\Pi(x) = 1$ if and only if there is some $y \in Proof_\pi(x)$ such that $V_\Pi(x, y) = 1$.

Solution

To show: $L_1 \leq_p L_2 \wedge L_2 \in \mathcal{NP} \Longrightarrow L_1 \in \mathcal{NP}$.

- There is a polynomial reduction $f : \Sigma_1^* \to \Sigma_2^*$ with $x \in L_1 \Leftrightarrow f(x) \in L_2$.
- There exist efficiently verifiable proofs for L_2.

Definition 2 (Proof Scheme for L_1)

- **Proofs:** $Proof_{L_1}$ assigns to each input $x \in \Sigma_1^*$ the set $Proof_{L_2}(f(x))$ of **possible proofs**.
- **Efficient Verification:** $V_{L_1}(x, y)$ denotes the following decision algorithm: first compute $f(x)$ in polynomial time then run $V_{L_2}(f(x), y)$, i.e. $V_{L_1}(x, y) = V_{L_2}(f(x), y)$.
- **Correctness:** As f is a polynomial reduction and $L_2 \in \mathcal{NP}$, the scheme is correct.

Exercise 4.3

Task

We call two languages $L_1 \subseteq \Sigma_1^*$ and $L_2 \subseteq \Sigma_2^*$ polynomial-time isomorphic if there is a bijective function $f : \Sigma_1^* \to \Sigma_2^*$ such that f is a polynomial reduction from L_1 to L_2 and f^{-1} is a polynomial reduction from L_2 to L_1.
Show that the following decision problems (treated as languages) are polynomial-time isomorphic. The respective instances consist of an undirected graph $G = (V, E)$ and a number $K \in \{0, \ldots, |V|\}$.

Independent Set: G contains an independent set (i.e., a set of vertices such that for every two vertices in the set, there is no edge connecting the two) of size at least K.

Clique: G contains a clique of size at least K.

Vertex Cover: G contains a vertex cover (i.e., a set of vertices such that each edge of the graph is incident to / "contains" at least one vertex of the set) of size at most $|V| - K$.

Solution

It is easy to see that the following statements are equivalent:

I) W is an independent set in G.

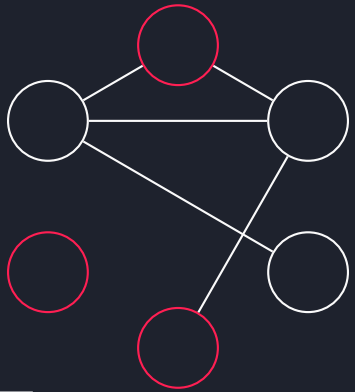
II) W is a clique in the complement graph G^c of G, where $G^c := (V, E^c)$ and $E^c := \{\{u, v\} \subseteq V \mid u \neq v,\ \{u, v\} \notin E\}$.

III) $V \setminus W$ is a vertex cover for G.

Solution

Example to illustrate the equivalence of (I) Independent Set and (II) Clique:

(I) An independent set[1] (red nodes) in G

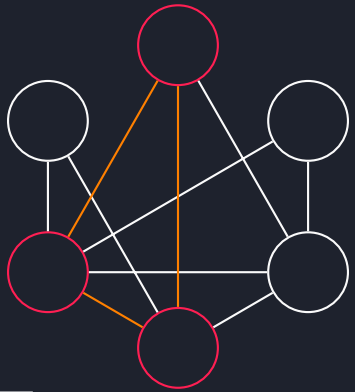


[1] The independent set is not maximal in this case.

Solution

Example to illustrate the equivalence of (I) Independent Set and (II) Clique:

(II) The corresponding clique[2] (red nodes) in G^c

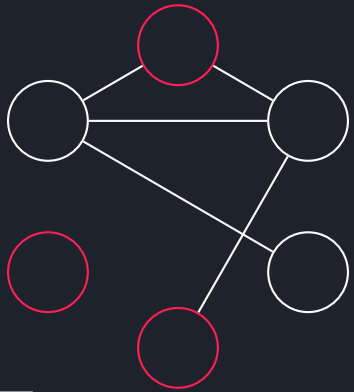


[2]The clique is not maximal in this case.

Solution

Example to illustrate the equivalence of (I) Independent Set and (III) Vertex Cover:

(I) An independent set[3] (red nodes) in G

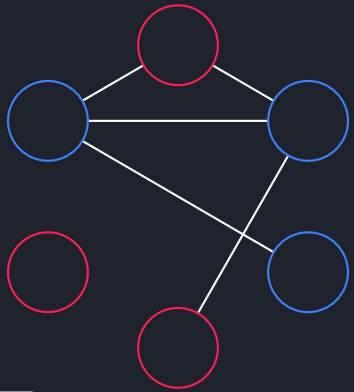


[3]The independent set is not maximal in this case.

Solution

Example to illustrate the equivalence of (I) and (III):

(III) The corresponding vertex cover[4] (blue nodes) in G



[4]The vertex cover is not minimal in this case.

Solution

It is easy to see that the following statements are equivalent:

I) W is an independent set in G.

II) W is a clique in the complement graph G^c of G, where $G^c := (V, E^c)$ and $E^c := \{\{u, v\} \subseteq V \mid u \neq v,\ \{u, v\} \notin E\}$.

III) $V \setminus W$ is a vertex cover for G.

Now the transformations are clear:

- $f_{I \to II}$ (from *Independent Set* to *Clique*) maps (G, K) to (G^c, K).
- $f_{I \to III}$ (from *Independent Set* to *Vertex Cover*) maps (G, K) to $(G, |V| - K)$.
- $f_{II \to III}$ (from *Clique* to *Vertex Cover*) can be constructed using composition:

$$f_{II \to III} := f_{I \to III} \circ f_{I \to II}^{-1}.$$

Exercise 4.4

Task

In this exercise, the inputs are DNF-formulas (i.e., disjunctions $M_1 \vee \cdots \vee M_n$ of conjunctions $L_1 \wedge \cdots \wedge L_s$ of literals L_i). Assume $\mathcal{P} \neq \mathcal{NP}$ and decide for each of the following two problems whether it is in $\mathcal{P}$:

a) Is there an assignment to the variables of the DNF-formula given as an input such that it evaluates to **1**?

b) Is there an assignment to the variables of the DNF-formula given as an input such that it evaluates to **0**?

Solution

a) Is there an assignment to the variables of the DNF-formula given as an input such that it evaluates to **1**?

This problem is in $\mathcal{P}$.
A monomial (i.e., a conjunction of literals) is satisfiable iff it does not contain a variable and its negation. (For each monomial, this can obviously be checked in polynomial time w.r.t. the size of the monomial.)
A DNF-formula is satisfiable iff at least one of its monomials is satisfiable. (Hence, the monomials of the DNF-formula can be checked independently, one after another for satisfiability. The search terminates once a satisfiable monomial has been found or if no monomials are left to check.)

Example for two variables x_1 and x_2:

$($	x_1	$\wedge$	$\neg x_1$	$)$	$\vee$	$($	x_1	$\wedge$	x_2	$)$	$\vee$	$($	$\neg x_1$	$\wedge$	$\neg x_2$	$)$
$($	1	$\wedge$	0	$)$	$\vee$	$($	1	$\wedge$	1	$)$	$\vee$	$($	0	$\wedge$	0	$)$

Solution

b) Is there an assignment to the variables of the DNF-formula given as an input such that it evaluates to **0**?

This problem is $\mathcal{NP}$-complete and, hence, not in $\mathcal{P}$ if $\mathcal{P} \neq \mathcal{NP}$.
Through negation and the use of De Morgan's laws, each instance of SAT (which is known to be $\mathcal{NP}$-complete) can be transformed (in polynomial time) into an instance of this problem.

$$
\begin{aligned}
1 &= (\neg x_1 \vee x_1) \wedge (\neg x_1 \vee \neg x_2) \wedge (x_1 \vee x_2) \\
\Leftrightarrow 0 &= \neg((\neg x_1 \vee x_1) \wedge (\neg x_1 \vee \neg x_2) \wedge (x_1 \vee x_2)) \\
&= \neg(\neg x_1 \vee x_1) \vee \neg(\neg x_1 \vee \neg x_2) \vee \neg(x_1 \vee x_2) \\
&= (x_1 \wedge \neg x_1) \vee (x_1 \wedge x_2) \vee (\neg x_1 \wedge \neg x_2)
\end{aligned}
$$

Exercise 4.5

Task

The Directed Hamiltonian Circuit Problem (DHC) for directed graphs is defined analogously to the Hamiltonian Circuit Problem (HC) for undirected graphs, which has already been introduced in the lecture.

Prove that *HC* is $\mathcal{NP}$-complete by showing that $DHC \leq_p HC$. $HC \in \mathcal{NP}$ and $DHC \in \mathcal{NPC}$ can be assumed as known facts in this exercise.

Solution

We know that $HC \in \mathcal{NP}$. In order to show $DHC \leq_p HC$, we use the following local replacement:

A directed Hamiltonian circuit in the given directed graph can be translated straightforwardly into an undirected Hamiltonian circuit in the new undirected graph.
If there is an undirected Hamiltonian circuit in the new graph, we have two possible “directions” to walk it. We fix a direction by picking an edge in this Hamiltonian circuit which has a corresponding edge in the given directed graph and assigning the respective direction.This yields a directed Hamiltonian circuit; because if we reached the left node in the undirected component of the above image from the left side and left this node to the left side as well, the node in the middle would not be passable as part of a Hamiltonian circuit any longer.


That's all Folks!