

ICCS310: Assignment 6

Possawat Sanorkam

possawat2017@hotmail.com

March 25, 2021

1: The Meaning of Things

(1) Class NP is the problems that can be verify within polynomial time. Basically, there is no efficient algorithm to solve the problem. But, we can verify it pretty quick.

(2) Without loss of generality, we can assume that there is a certificate and a verifier that check the certificate to verify whether it is a "yes" or "no" given that the verifier answer within polynomial time.

(3) NP-complete is the problem that can only be solved within polynomial time using a NFA. Besides, we can solve it in polynomial time using a machine that compute all possibilities at once.

(4) We can find a problem that is NP, then we find a polynomial time algorithm to change the solution from one problem to the problem that we want to show that it is NP-complete.

2: Closure of NP

(i) Disprove that $A \cap B$ must be in NP

Proof:

We want to show that $A \cap B$ must not be in NP. Let $a \in A$ and $b \in B$. We were given that $A \in \text{NP}$ and $B \in \text{NP}$. So, $(\forall a \in A)[a \in \text{NP}]$ and $(\forall b \in B)[b \in \text{NP}]$. However, if $(\forall a \in A)[a \notin B]$, then we got two disjoint sets of A and B . Then, $A \cap B = \emptyset$. We have that \emptyset is a special language that is undecidable and there is no machine that can decide it, because we will not be able to find "yes" instance from nothingness. Therefore, $A \cap B$ cannot be in NP. \square

(ii) Disprove that $A \cup B$ must be in NP

Proof:

We want to show that $A \cup B$ must not be in NP. Let $a \in A$ and $b \in B$. We were given that $A \in \text{NP}$ and $B \in \text{NP}$. So, $(\forall a \in A)[a \in \text{NP}]$ and $(\forall b \in B)[b \in \text{NP}]$. However, if $(\forall a \in A)[a \notin B]$, then we got two disjoint sets of A and B . Then, $A \cup B = \Sigma^*$. We have that Σ^* is a special language that is undecidable and there is no machine that can decide it, because we will not be able to find "no" instance from everything. Therefore, $A \cup B$ cannot be in NP. \square

3: This is NP

Prove that $5\text{COLOR} \in \text{NP}$

Proof:

We want to show that $5\text{COLOR} \in \text{NP}$. So, we want to claim that there exist a polynomially-bounded certificate and a polynomially-bounded verifier.

Claim: There exists such polynomially-bounded certificate.

We claim that the certificate that is a yes instance is the list of vertices describing the color in which we colored them. The length of such a certificate is the same as the number of vertices given. So, we got the certificate.

Claim: There exists such polynomially-bounded verifier.

We claim that the verifier that verify the certificate will check through the list of vertices describing the color in which we colored them. If there exists an edge that contain same color on both ends, we have that the certificate is a "no" instance. Otherwise, it is a "yes" instance. The time complexity of this algorithm is $O(|E|)$ where E is the list of edges. So, we got the verifier.

Hence, we showed that there exist a polynomially-bounded certificate and a polynomially-bounded verifier. Therefore, 5COLOR \in NP.

4: NP-Complete

(i) Show that

$$\text{TOTAL} = \{\langle M \rangle \mid M \text{ is a Turing machine that halts on every input}\}$$

is undecidable

Proof: ($\text{ACCEPT}_{\text{TM}} \leq \text{TOTAL}_{\text{TM}}$)

Suppose that TM M_{TOTAL} decides TOTAL_{TM} and TM M_{ACCEPT} decides $\text{ACCEPT}_{\text{TM}}$, we want to show how to decide $\text{ACCEPT}_{\text{TM}}$ using M_{TOTAL} .

Given $\langle M, w \rangle$ as input:

1. Make TM M' from M where if M accepts, we accept and enter loops when M rejects.
2. Run M_{TOTAL} with $\langle M', w \rangle$.
3. If M_{TOTAL} accepts, we accept. If M_{TOTAL} rejects, we reject.

Notice that this mechanism accepts if and only if M accepts w and rejects if and only if M' rejects w .

$$M_{\text{TOTAL}} \text{ accepts } \langle M', w \rangle \iff M_{\text{ACCEPT}} \text{ accepts } \langle M, w \rangle$$

Hence, M_{ACCEPT} can correctly decide $\text{ACCEPT}_{\text{TM}}$ provided that there is a TM M_{TOTAL} . So, $\text{ACCEPT}_{\text{TM}} \leq \text{TOTAL}_{\text{TM}}$. Therefore, TOTAL is undecidable. \square

(ii) Show that

$$\text{FINITE} = \{\langle M \rangle \mid M \text{ is a Turing machine and } L(M) \text{ is a finite set}\}$$

is undecidable

Proof: ($\text{ACCEPT}_{\text{TM}} \leq \text{FINITE}_{\text{TM}}$)

Suppose that TM M_{FINITE} decides $\text{FINITE}_{\text{TM}}$ and TM M_{ACCEPT} decides $\text{ACCEPT}_{\text{TM}}$, we want to show how to decide $\text{ACCEPT}_{\text{TM}}$ using M_{FINITE} .

Given $\langle M, w \rangle$ as input:

1. Make TM M' from M where if M accepts, we accept and enter loops when M rejects.

2. Run M_{FINITE} with $\langle M', w \rangle$.
3. If M_{FINITE} accepts, we accept. If M_{FINITE} rejects, we reject.

Notice that this mechanism accepts if and only if M accepts w and rejects if and only if M' rejects w .

$$M_{\text{FINITE}} \text{ accepts } \langle M', w \rangle \iff M_{\text{ACCEPT}} \text{ accepts } \langle M, w \rangle$$

Hence, M_{ACCEPT} can correctly decide $\text{ACCEPT}_{\text{TM}}$ provided that there is a TM M_{FINITE} . So, $\text{ACCEPT}_{\text{TM}} \leq \text{FINITE}_{\text{TM}}$. Therefore, FINITE is undecidable. \square

(iii) Show that

$$\text{REGULAR} = \{ \langle M \rangle \mid M \text{ is a Turing machine and } L(M) \text{ is regular} \}$$

is undecidable

Proof: ($\text{ACCEPT}_{\text{TM}} \leq \text{REGULAR}_{\text{TM}}$)

Suppose that TM M_{REGULAR} decides $\text{REGULAR}_{\text{TM}}$ and TM M_{ACCEPT} decides $\text{ACCEPT}_{\text{TM}}$, we want to show how to decide $\text{ACCEPT}_{\text{TM}}$ using M_{REGULAR} .

Given $\langle M, w \rangle$ as input:

1. Make TM M' from M . On input x :
 - 1.1 If x has the form $0^n 1^n$, accepts.
 - 1.2 If x does not have the form $0^n 1^n$, run M on input w and accept if M accepts w .
2. Run M_{REGULAR} with $\langle M', w \rangle$.
3. If M_{REGULAR} accepts, we accept. If M_{REGULAR} rejects, we reject.

Notice that this mechanism accepts if and only if M accepts w and rejects if and only if M' rejects w .

$$M_{\text{REGULAR}} \text{ accepts } \langle M', w \rangle \iff M_{\text{ACCEPT}} \text{ accepts } \langle M, w \rangle$$

Hence, M_{ACCEPT} can correctly decide $\text{ACCEPT}_{\text{TM}}$ provided that there is a TM M_{REGULAR} . So, $\text{ACCEPT}_{\text{TM}} \leq \text{REGULAR}_{\text{TM}}$. Therefore, REGULAR is undecidable. \square

5: Silver Lining If $P = NP$

Prove that

$$\text{TOTAL} \leq_T \text{FINITE}$$

Proof:

Suppose that TM M_{TOTAL} decides TOTAL_{TM} and TM M_{FINITE} decides $\text{FINITE}_{\text{TM}}$, we want to show how to decide TOTAL_{TM} using M_{FINITE} .

Given $\langle M \rangle$ as input:

1. Run M_{FINITE} on $\langle M \rangle$.
2. If M_{FINITE} accepts, we accept. If M_{FINITE} rejects, we reject.

Refer to the fact that M_{FINITE} can determine whether M has finite set of $L(M)$ or not, M will halt on every input only if $L(M)$ is a finite set. If $L(M)$ is not a finite set, we would not be

able to determine that it will halt on every input since there would be at least one input that would not halt.

Hence, M_{TOTAL} can correctly decide TOTAL_{TM} provided that there is a TM M_{FINITE} . Therefore, $\text{ACCEPT}_{\text{TM}} \leq \text{REGULAR}_{\text{TM}}$. \square