MEASUREING COMPLEXITY BY USING REDUCTION TO SOLVE P VS NP AND NC HIERARCHY

KOBAYASHI KOJI

1. Abstract

This article describes about that NC hierarchy and P is not NP by using problem reduction. If L is not P, we can prove P is not NP by using difference between logarithm space reduction and polynomial time reduction. Like this, we can also prove NC hierarchy by using difference between AL0 and NC1. This means L is not P. Therefore P is not NP.

2. Preparation

In this article, we will use words and theorems of References [1, 2, 3] in this paper. And we use description as follows;

Definition 1. We will use the term "$pDTM$" as Turing Machine set that compute P, "$LDTM$" as Turing Machine set that compute L. "$RpDTM$" as Reversible $pDTM$.

And we use circuit problem as follows;

Definition 2. We will use the term "AC^i" as uniform circuits family set that compute AC^i problem, "NC^i" as uniform circuits family set that compute NC^i problem, "RC^i" as reversible circuits family that compute NC^i problem. "$f\circ g$" as connected circuit that g outputs connect to f inputs. In this case, we also use circuits family or circuits family set. For example, $A\circ BB$ of circuits family A and circuits family set BB means a circuit that $a\circ b\mid a\in A,b\in B\in BB$. Circuits family uniformity is that these circuits can compute AC^0.

3. P is not NP if L is not P

Theorem 3. $L\subsetneq P\rightarrow P\subsetneq NP$

Proof. To prove it by using contraposition $P=NP\rightarrow L=P$. If $P=NP$, then all $A,B\in NP-Complete$ have $f\in LDTM$ that reduce A to B.

$P=NP\rightarrow\forall A,B\in NP-Complete\exists f\in LDTM\left(f\left(A\right)=B\right)$

If $g\in RpDTM$ then

$A\leq_{p}g\left(A\right)$

and

$g\left(A\right)\leq_{p}g^{-1}\left(g\left(A\right)\right)=A\in NP\Longrightarrow g\left(A\right)\in NP$

Therefore

$g\left(A\right)\in NP-Complete$

That is,

$\forall A\in NP-Complete,g\in RpDTM\exists f\in LDTM\left(f\left(A\right)=g\left(A\right)\right)$

As we all know, all $pDTM$ correspond to $RpDTM$ and $LDTM$ can pick up $pDTM$ output from $RpDTM$ output. That is;

$\forall f \in pDTM\,(\exists g \in RpDTM, h \in LDTM\,(h \circ g = f))$

So that this means $L = P$.

Therefore, this theorem was shown. □

4. NC hierarchy

Theorem 4. *AC^i has Universal Circuits Family that can emulate all AC^i circuits family.*

Proof. To prove this theorem by making universal circuit family $A^i \in AC^i$ that emulate circuit family $\{C_j\} \in AC^i$ by using "depth circuit tableau". Universal circuit $U_j \in A^i$ have partial circuit $u_{k,d}$ that emulate all C_j gates $g_{k\in n}$ (include input value) and connected wires $w_{p,q}$ from g_p output to g_q input in every depth d. ($w_{p,p}$ always exist)

$u_{v\in n,d}$ have inputs from all $u_{u\in n,d-1}$ and g_u information that mean

a) validity of $u_{u,d-1}$

b) $u_{u,d-1}$ output (true if g_u output true)

c) existence of $w_{u,v}$ (true if $w_{u,v}$ is exists)

d) negation of $w_{u,v}$ (true if $w_{u,v}$ include not gate)

e) gate type of g_v (Or gate or And gate)

and outputs to $u_{w\in n,d+1}$ that mean

A) validity of $u_{v,d}$

B) $u_{v,d}$ output

These $u_{v,d}$ compute output like this;

If $u_{u,d-1}$ a) or c) input false then $u_{v,d}$ ignore $u_{u,d-1}$.

If $u_{u,d-1}$ a) and c) input true then $u_{v,d}$ A) output true and $u_{v,d}$ B) output g_k value that compute from e), b), d). b), d) include another $u_{w\in n,d-1}$ b), d).

If all a) input false then $u_{k,d}$ A) output false.

If all c) input false then $u_{k,d}$ A) output false.

And depth 0 circuit compute additional condition;

If $u_{k,0}$ is C_j input then $u_{k,0}$ A) output true and $u_{i,d}$ B) output C_j input value, else $u_{k,0}$ A) output false.

This U_j that consists of u emulate C_j. We can make every u in AC^0, so that A^i in AC^i.

Therefore, this theorem was shown. □

Definition 5. We will use the term "A^i" as universal circuits family that compute AC^i problem, "N^i" as universal circuits family that compute NC^i problem.

Theorem 6. *AC^0 can reduce all AC^i to A^i. That is, A^i is $AC^i - Complete$ with AC^0 reduction.*

Proof. Mentioned above 24, we can make all AC^i by using AC^0 and we can connect these AC^i to A^i. That is, we can emulate all AC^i circuit by using $A^i \circ AC^0$. From the view of A^i, AC^0 is input reduction from AC^i to A^i. Therefore, this theorem was shown. □

Theorem 7. $NC^i \subsetneq NC^{i+1}$

Proof. We can prove this theorem like mentioned above 3.

To prove it using reduction to absurdity. We assume that $NC^i = AC^i = NC^{i+1}$. From assumption $NC^i = AC^i$ and mentioned above 4, universal circuit $N^i \in NC^i$ also exists.

$NC^i = AC^i \rightarrow \exists N^i \in NC^i \left(N^i = A^i\right)$

As we all know, NC^{i+1} include $N^i \circ a \mid a \in RC^1$ circuits, and all NC^1 correspond to RC^1 and AC^0 can pick up NC^1 output from RC^1 output. That is;

$\forall a \in RC^1 \left(N^i \circ a \in NC^{i+1}\right)$

$\forall c \in NC^1 \left(\exists a \in RC^1, b \in AC^0, (b \circ a = c)\right)$

From assumption $AC^i = NC^{i+1}$, all $N^i \circ a$ correspond to AC^i. Therefore

$AC^i = NC^{i+1} \rightarrow \forall a \in RC^1 \left(N^i \circ a \in AC^i\right)$

Mentioned above 6, AC^i have AC^0 reduction to universal circuit A^i.

$\forall d \in AC^i \exists e \in AC^0 \left(A^i \circ e = d\right)$

That is,

$NC^i = AC^i = NC^{i+1}$

$\rightarrow \forall a \in RC^1 \exists e \in AC^0 \left(N^i \circ a = A^i \circ e\right)$

$\rightarrow \forall a \in RC^1 \exists e \in AC^0 \left(a = e\right)$

$\rightarrow \forall f \in NC^1 \exists e, h \in AC^0 \left(f = h \circ e\right)$

But this means $AC^0 = NC^1$ and contradict $AC^0 \subsetneq NC^1$.

Therefore, this theorem was shown than reduction to absurdity. □

5. P is not NP

Theorem 8. $P \neq NP$

Proof. Mentioned above 3, $L \subsetneq P \rightarrow P \subsetneq NP$. And mentioned above 7, $L \subset NC^i \subsetneq NC^{i+1} \subset P$. Therefore $P \subsetneq NP$. □

References

[1] Michael Sipser, (translation) OHTA Kazuo, TANAKA Keisuke, ABE Masayuki, UEDA Hiroki, FUJIOKA Atsushi, WATANABE Osamu, Introduction to the Theory of COMPUTATION Second Edition, 2008

[2] OGIHARA Mitsunori, Hierarchies in Complexity Theory, 2006

[3] MORITA Kenichi, Reversible Computing, 2012