MEASURING COMPLEXITY BY USING REDUCTION TO SOLVE P VS NP AND NC & PH

KOBAYASHI KOJI

1. Abstract

This article prove that NC and PH is proper (especially P is not NP) by using reduction difference. We can prove that NC is proper by using AL0 is not NC. This means L is not P. We can prove P is not NP by using reduction difference between L and P. And we can also prove that PH is proper by using P is not NP.

2. NC is proper

We use circuit problem as follows;

Definition 1. We will use the term "AC^i", "NC^i" as each complexity decision problems classes. "FAC^i" as function problems class of AC^i. These complexity classes also use uniform circuits family set that compute target complexity classes problems. "$f \circ g$" as composite circuit that output of g are input of f. In this case, we also use complexity classes to show target circuit. For example, $A \circ BB$ when A is circuits family and BB is circuits family set mean that $a \circ b \mid a \in A, b \in B \in BB$. "$R(A)$" as subset of reversible NC that include A. Reversible mean that $\left(R(A) \circ (R(A))^{-1}\right)(x) = x$. Circuits family uniformity is that these circuits can compute FAC^0.

Theorem 2. $NL \leq_{AC^0} NC^2$

Proof. Mentioned [1] Theorem 10.40, all NC^2 are closed by FL reduction. This reduction is validity of (c_1, c_2) transition function. Transition function change $O(1)$ memory and keep another memory. Therefore this validity can compute AC^0 and we can replace FL to FAC^0. □

Theorem 3. *AC^i has Universal Circuits Family that can emulate all AC^i circuits family. That is, every AC^i has $AC^i - Complete$ under FAC^0.*

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 partial circuit $v_{p-q,d}$ that emulate all wires w_{p-q} from g_p output to g_q input in every depth d. U_j use three value $\{\top, \bot, \emptyset\}$. $\emptyset$ is special value that all g_k ignore this value. All gate in a depth d is u_d, all wires that input connected k in a depth d is $v_{k-,d}$, output connected k in a depth d is $v_{-k,d}$.

$v_{p-q,d}$ input connected each $u_{p,d}$ output and w_{p-q}. $v_{p-q,d}$ output connected each $u_{q,d+1}$ input. If w_{p-q} does not exist, $v_{p-q,d}$ output $\emptyset$. Else if w_{p-q} have negative then $v_{p-q,d}$ output $u_{k,d}$ negative value. Else $v_{p-q,d}$ output $u_{k,d}$ positive value.

$u_{k,d}$ input connected each $v_{-k,d-1}$ output and g_k. $u_{k,d}$ output connected each $v_{k-,d}$ input. If g_k is one of C_j input value, $u_{k,d}$ output the input value. Else (g_k is And / Or gate) $u_{k,d}$ output the gate value that compute from all $v_{-k,d-1}$ output values. In this computation, $u_{k,d}$ ignore all $\emptyset$. If all value are $\emptyset$, $u_{k,d}$ output $\emptyset$.

This U_j that consists of u, v emulate C_j. We can make every u, v in FAC^0 because C_j is uniform circuit1. Therefore, A^i in AC^i and this theorem was shown. □

Theorem 4. *$NC^i = NC^{i+1} \rightarrow NC^i - Complete = AC^i - Complete = NC^{i+1} - Complete$.*

Proof. If $NC^i = NC^{i+1}$, all $NC^i - Complete, AC^i - Complete, NC^{i+1} - Complete$ can reduce each other and $NC^i - Complete, AC^i - Complete, NC^{i+1} - Complete$ in NC^i. Therefore, this theorem was shown. □

Theorem 5. *$NC^i \subsetneq NC^{i+1}$*

Proof. To prove it using reduction to absurdity. We assume that $NC^i = NC^{i+1}$. It is trivial that $NC^i = AC^i = NC^{i+1} = AC^{i+1} = \cdots$.

Because $NC^i = NC^{i+1}$ and mentioned above 4, $R\left(FAC^i - Complete\right) \subset FAC^i - Complete$. Therefore

$NC^i = NC^{i+1} \rightarrow \forall A, B \in R\left(FAC^i - Complete\right) \exists C \in FAC^0 \left(A \circ B = A \circ C\right)$

A is reversible circuits family. Therefore A have A^{-1}.

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

$\rightarrow \forall A, B \in R\left(FAC^i - Complete\right) \exists C \in FAC^0 \left(A^{-1} \circ A \circ B = A^{-1} \circ A \circ C\right)$

$\rightarrow \forall B \in R\left(FAC^i - Complete\right) \exists C \in FAC^0 \left(B = C\right)$

This means $FAC^0 = FAC^i$. But this contradict $AC^0 \subsetneq NC^1 \subset AC^i$.

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

Theorem 6. *$AC^i \subsetneq AC^{i+1}$*

Proof. If $AC^i = AC^{i+1}$ then $AC^i = NC^{i+1} = AC^{i+1} = NC^{i+2} = AC^{i+2}$ and contradict mentioned above 5 $NC^i \subsetneq NC^{i+1}$. Therefore, this theorem was shown than reduction to absurdity. □

Theorem 7. *$NC = AC \subsetneq P$*

Proof. To prove it using reduction to absurdity. We assume that $NC = P$. It is trivial that we can reduce some $A \in P - Complete$ to $B \in NC$. But B is also in NC^i. Therefore, this mean that $NC^i = NC^i$ and contradict mentioned above 5 $NC^i \subsetneq NC^{i+1}$. Therefore, this theorem was shown than reduction to absurdity. □

3. PH is proper

Definition 8. We will use the term "L", "P", "$P - Complete$", "NP", "$NP - Complete$", "FL", "FP" as each complexity classes. These complexity classes also use Turing Machine (TM) set that compute target complexity classes problems. We will use the term "Δ_k", "Σ_k", "Π_k" as each Polynomial hierarchy classes. "$f \circ g$" as composite problem that output of g are input of f. "$R(A)$" as "reversible TM" that equal A. Reversible mean that $\left(R(A) \circ (R(A))^{-1}\right)(x) = x$.

Theorem 9. *$R(\Sigma_k) \subset \Sigma_k$, $R(\Pi_k) \subset \Pi_k$.*

Proof. We can reduce Σ_k and Π_k to another Σ_k and Π_k that have tree graph of computation history. (if all configuration keep input, computation history become tree graph.) These Σ_k,Π_k are $R(\Sigma_k)$, $R(\Pi_k)$ because each computation history of each output only reach one input. Therefore $\left(R(A) \circ (R(A))^{-1}\right)(x) = x$. We can compute these reduction in FP. Therefore, this theorem was shown. □

Theorem 10. $P \subsetneq NP$

Proof. To prove it using reduction to absurdity. We assume that $P = NP$.

As we all know that if $P = NP$ then all NP can reduce $P - Complete$ under FL. And all $NP \circ FP \subset NP$. Therefore

$P = NP \rightarrow \forall A \in NP - Complete \forall B \in FP \exists C \in FL (A \circ B = A \circ C)$

Mentioned above9, $R(NP - Complete) \subset NP - Complete$. Therefore

$P = NP \rightarrow \forall D \in R(NP - Complete) \forall B \in FP \exists C \in FL (D \circ B = D \circ C)$

D is reversible function. Therefore D have D^{-1}.

$P = NP$

$\rightarrow \forall D \in R(P - Complete) \forall B \in FP \exists C \in FL \left(D^{-1} \circ D \circ B = D^{-1} \circ D \circ C\right)$

$\rightarrow \forall D \in R(P - Complete) \forall B \in FP \exists C \in FL (B = C)$

This means $FP = FL$. But this contradict $FL \subsetneq FP$ mentioned above5. Therefore, this theorem was shown than reduction to absurdity. □

Theorem 11. $\Pi_k = \Pi_{k+1} \rightarrow \Pi_k - Complete = \Pi_{k+1} - Complete$

Proof. If $\Pi_k = \Pi_{k+1}$, all $\Pi_k - Complete, \Pi_{k+1} - Complete$ can reduce each other and $\Pi_k - Complete, \Pi_{k+1} - Complete$ in Π_k. Therefore, this theorem was shown. □

Theorem 12. $\Pi_k \subsetneq \Pi_{k+1}$

Proof. To prove it using reduction to absurdity. We assume that $\Pi_k = \Pi_{k+1}$. It is trivial that $\Pi_k = \Pi_{k+1} = \Pi_{k+2} = \cdots$.

Mentioned [2] Theorem 6.26, $\Pi_k - Complete$ under polynomial time reduction exist. Therefore all $\Pi_{k+1} - Complete$ can reduce $\Pi_k - Complete$ under FP. Because $\Pi_k = \Pi_{k+1}$ and mentioned above 11, $R(\Pi_k - Complete) \subset \Pi_k - Complete$. Therefore

$\Pi_k = \Pi_{k+1} \rightarrow \forall A, B \in R(\Pi_k - Complete) \exists C \in FP (A \circ B = A \circ C)$

A is reversible function. Therefore A have A^{-1}.

$\Pi_k = \Pi_{k+1}$

$\rightarrow \forall A, B \in R(\Pi_k - Complete) \exists C \in FP \left(A^{-1} \circ A \circ B = A^{-1} \circ A \circ C\right)$

$\rightarrow \forall B \in R(\Pi_k - Complete) \exists C \in FP (B = C)$

This means $\Pi_k = FP$. But this contradict mentioned above10. Therefore, this theorem was shown than reduction to absurdity. □

Theorem 13. $\Delta_k \subsetneq \Sigma_k, \Sigma_k \neq \Pi_k$

Proof. Mentioned [2] Theorem 6.12,

$\Sigma_k = \Pi_k \rightarrow \Sigma_k = \Pi_k = PH$

$\Delta_k = \Sigma_k \rightarrow \Delta_k = \Sigma_k = \Pi_k = PH$

This contraposition is,

$(\Sigma_k \subsetneq PH) \vee (\Pi_k \subsetneq PH) \rightarrow \Sigma_k \neq \Pi_k$

$(\Delta_k \subsetneq PH) \vee (\Sigma_k \subsetneq PH) \vee (\Pi_k \subsetneq PH) \rightarrow \Delta_k \neq \Sigma_k$

From mentioned above 12,

$\Sigma_k \subsetneq \Pi_{k+1} \subset PH$

Therefore, $\Delta_k \neq \Sigma_k, \Sigma_k \neq \Pi_k$.
Mentioned [2] Theorem 6.10,
$\Sigma_k \subset \Sigma_{k+1}$, $\Pi_k \subset \Pi_{k+1}$,$\forall k \geq 1 \left(\Delta_k \subset (\Sigma_k \cap \Pi_k) \subset (\Sigma_k \cup \Pi_k) \subset \Delta_{k+1}\right)$
Therefore, $\Delta_k \subsetneq \Sigma_k, \Sigma_k \neq \Pi_k$. □

Theorem 14. $\Pi_k \not\subset \Sigma_k, \Sigma_k \not\subset \Pi_k$

Proof. To prove it using reduction to absurdity. We assume that $\Pi_k \subset \Sigma_k$. This means that all $\overline{\Sigma_k} = \Pi_k$ is also Σ_k.
$\Pi_k \subset \Sigma_k \rightarrow \forall A \in \Sigma_k \left(\overline{A} \in \Pi_k \subset \Sigma_k\right)$
Mentioned [2] Theorem 6.21, all Σ_k are closed under polynomial time conjunctive reduction. We can emulate these reduction by using Π_1. That is,
$\exists B \in \Sigma_k \forall C \in \Sigma_k \exists D \in \Pi_1 \left(B \circ D = C\right)$
Therefore,
$\Pi_k \subset \Sigma_k$
$\rightarrow \exists B \in \Sigma_k \forall C \in \Sigma_k \exists D \in \Pi_1 \forall A \in \Sigma_k \left(B \circ D = C\right) \wedge \left(\overline{A} \in \Pi_k \subset \Sigma_k\right)$
$\rightarrow \exists B \in \Sigma_k \forall C \in \Sigma_k \exists D \in \Pi_1 \left(B \circ D = C\right) \wedge \left(\overline{B} \in \Sigma_k\right)$
$\rightarrow \exists B \in \Sigma_k \forall C \in \Sigma_k \exists D \in \Pi_1 \left(B \circ D = C\right) \wedge \left(B \in \Pi_k\right)$
Therefore $\Sigma_k \subset \Pi_k$ because $B \circ D \in \Pi_k$. But this means $\Sigma_k = \Pi_k$ and contradict $\Sigma_k \neq \Pi_k$ mentioned above 13. Therefore $\Pi_k \not\subset \Sigma_k$.
We can prove $\Sigma_k \not\subset \Pi_k$like this.
Therefore, this theorem was shown than reduction to absurdity. □

Theorem 15. $\Delta_k \subsetneq \Pi_k$

Proof. To prove it using reduction to absurdity. We assume that $\Delta_k = \Pi_k$.
Mentioned [2] Theorem 6.10,
$\Sigma_k \subset \Sigma_{k+1}$, $\Pi_k \subset \Pi_{k+1}$,$\forall k \geq 1 \left(\Delta_k \subset (\Sigma_k \cap \Pi_k) \subset (\Sigma_k \cup \Pi_k) \subset \Delta_{k+1}\right)$
Therefore
$\Delta_k = \Pi_k$
$\rightarrow \Delta_k = \Pi_k \subset (\Sigma_k \cap \Pi_k) \subset \Sigma_k \subset (\Sigma_k \cup \Pi_k) \subset \Delta_{k+1}$
$\rightarrow \Pi_k \subset \Sigma_k$
But this result contradict mentioned above 14.
Therefore, this theorem was shown than reduction to absurdity. □

Theorem 16. $\Sigma_k \subsetneq \Delta_{k+1}, \Pi_k \subsetneq \Delta_{k+1}$

Proof. To prove it using reduction to absurdity. We assume that $\Sigma_k = \Delta_{k+1}$.
Mentioned [2] Theorem 6.10,
$\forall k \geq 1 \left(\Delta_k \subset (\Sigma_k \cap \Pi_k) \subset (\Sigma_k \cup \Pi_k) \subset \Delta_{k+1}\right)$
Therefore
$\Sigma_k = \Delta_{k+1}$
$\rightarrow \Delta_k \subset (\Sigma_k \cap \Pi_k) \subset \Pi_k \subset (\Sigma_k \cup \Pi_k) \subset \Sigma_k = \Delta_{k+1}$
$\rightarrow \Pi_k \subset \Sigma_k$
But this result contradict mentioned above 14. Therefore $\Sigma_k \subsetneq \Delta_{k+1}$.
We can prove $\Pi_k \subsetneq \Delta_{k+1}$ like this.
Therefore, this theorem was shown than reduction to absurdity. □

Theorem 17. $PH \subsetneq PSPACE$

Proof. To prove it using reduction to absurdity. We assume that $PH = PSPACE$. It is trivial that we can reduce some $A \in PSPACE - Complete$ to $B \in PH$. But

B is also in Δ_k. Therefore, this mean that $\Delta_k = \Delta_{k+1}$ and contradict mentioned above 1516 $\Delta_k \subsetneq \Delta_{k+1}$. Therefore, this theorem was shown than reduction to absurdity. $\square$

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