INPUT INDEPENDENCE AND COMPUTATIONAL COMPLEXITY

KOJI KOBAYASHI

Abstract. This paper describes about complexity of PH problems by using problem input independence, and provide new approach to solve P vs NP problem.

Circuit family that emulate Deterministic Turing machine (DTM) are almost monotone circuits except input variables (like negation normal form (NNF)). Therefore, we can find out DTM limitation by using this "NNF circuit family".

To clarify NNF circuit limitation, we pay attention to AND-gate and OR-gate of NNF circuit DAG network. Because each pair of positive and negative variables do not become 1 in same input, these pair join at OR-gate to output computation result as 1. If some accept inputs does not include another accept input between these input with Hamming distance, OR-gate which join "Neighbor input" as accept input have to meet AND-gate to exclude between "Boundary input" as reject input. This means that these different variables of neighbor input are finally meet AND-gate, and NNF circuit have to use unique AND-gate to differentiate such different variables and boundary input variables.

The other hand, we can make neighbor input problem "Neighbor Tautology DNF problem (NTD)" from DNF Tautology problem. NTD is in PH, and NTD consist of neighbor input and number of these input is over polynomial size of input length. Therefore NNF circuit family that compute NTD are over polynomial size, and NTD that include PH is not in P.

Date: 2018-03-10.

1. NNF circuit family

Explained in [Sipser] Circuit Complexity section, Circuit family can emulate DTM computation only using NOT-gate in separating input values $\{0, 1\}$ to $\{01, 10\}$. In this paper, we use this "almost all monotone circuit" to clarify DTM limitation.

Definition 1.1.

We use term as following;

NNF : Negation Normal Form.

DTM : Deterministic Turing Machine

NTM : Nondeterministic Turing Machine

DAG : Direct Acyclic Graph

DNF : Disjunctive Normal Form.

DNFTAUT : DNF TAUTology problem.

MTD: Minimal Tautology DNF. That is, tautology DNF which become non tautology if any clause delete from the DNF. (Negation of Minimal Unsatisfiable Core of CNF)

In this paper, we will use words and theorems of References [Sipser].

Definition 1.2.

We will use the term;

"NNF Circuit Family" as circuit family that have no NOT-gate except connecting input gates directly (like negation normal form). DTM emulator which mentioned [Sipser] Circuit Complexity section are included in NNF Circuit family. To simplify, circuit can compute shorter input from circuit input (such shorter input have filler with concrete input).

"Input variable pair" as output pair of input gate and NOT-gate $\{01, 10\}$ that correspond to an input variable $\{0, 1\}$.

"Accept input" as input that circuit family output 1.

"Reject input" as input that circuit family output 0.

“Partial input” as subset of input which connect target gate and decide the gate output as 1.

“Neighbor input”as another accept input that no accept inputs exists between these target input and neighbor input with Hamming distance.

“Boundary input of neighbor input” as reject input that exist between neighbor inputs with Hamming distance.

“Different Variables” as subset of input variables that difference each other in neighbor input.

“Same Variables” as subset of input variables that same each other in neighbor input.

“Effective circuit of input t” as one of minimal sub circuit that decide circuit output as 1 with input t. Effective circuit do not include gate even if gate change output 0 and effective circuit keep output 1. To simplify, effective circuit do not include NOT-gate (monotone circuit).

“Effective sub circuit of partial input t / gate g” as one of minimal sub circuit that decide gate g output as 1 with partial input t.

Confirm NNF circuit family behavior. Mentioned in [Sipser], NNF circuit family can emulate DTM with polynomial size of DTM computation time. All effective circuit become DAG that leaves are input variables and root is an output gate. All gates that include effective circuit become 1 if output is 1. Especially, all different variables of input cannot overlay in same time, so all different effective circuit are join at OR-gate to connect output gate as root.

This NNF circuit behavior clarify problem structure symmetry and independence of each inputs.

Theorem 1.3.

All input variable pair of different variables join OR-gate in effective circuit.

Proof. It is trivial because input variable pair does not become 1 in same input and it is necessary to join OR-gate and output 1 to connect output gate in effective circuit. □

Theorem 1.4.

NNF circuit have to use at least one AND-gate to differentiate neighbor input and boundary input.

Proof. Mentioned above 1.3, all input variable pair of different variables join OR-gate which output 1 in effective circuit. Because NNF circuit is monotone circuit except input, there is two case to join OR-gate;

a) all different variables meet at AND-gate, and join at OR-gate after meeting AND-gate,

b) some partial different variables meet at AND-gate, and join OR-gate after meeting AND-gate, and meet at AND-gate all output of OR-gate.

a) case, some AND-gate become 1 if input include one side of different variables to differentiate these different variables from boundary input. Therefore, root of AND-gate is also unique gate.

b) case, because no boundary input become accept input, some OR-gate which join neighbor input become 0 with boundary input. That is, effective gate become 1 if these OR-gate become 1. Therefore, it is necessary that effective gate include AND-gate that meet all OR-gate that join different variables (and other same variables) to output 1 if input include different variables, and output 0 if input include boundary input of these different variables. □

That is to say, neighbor input cannot permutate proper partial input of different variables. Input of OR-gate can permutate each other, so NNF circuit have to use all OR-gate and AND-gate that fix different variables as neighbor input (and not boundary inputs). It is necessary to use unique AND-gate to identify all fixing different variables.

2. Neighbor Tautology DNF

Let clarify number of neighbor input. To consider DNF tautology problem, some input have neighbor input by changing one literals positive / negative. So we define new partial problem from these DNF tautology.

Definition 2.1.

We will use the term "Neighbor Tautology DNF problem" or "NTD" as partial Minimal Tautology DNF problem which input also tautology if one literal x change positive / negative $\{x, \overline{x}\} \to \{\overline{x}, x\}$ and not tautology if proper subset of one type literal change positive / negative.

$$NTD = \left\{ f \mid f \equiv \top, f \begin{pmatrix} \cdots & x & \overline{x} & \cdots \\ \cdots & \overline{x} & x & \cdots \end{pmatrix} \equiv \top, g = f \begin{pmatrix} \cdots & x & \overline{x} & \cdots \\ \cdots & \{x, \overline{x}\} & \{x, \overline{x}\} & \cdots \end{pmatrix} \not\equiv \top \right\}$$

$\begin{pmatrix} \cdots & x & \overline{x} & \cdots \\ \cdots & \overline{x} & x & \cdots \end{pmatrix}$: changing of all literal x to $\overline{x}$ ($\overline{x}$ to x).

$\begin{pmatrix} \cdots & x & \overline{x} & \cdots \\ \cdots & \{x, \overline{x}\} & \{\overline{x}, x\} & \cdots \end{pmatrix}$: (any) changing of proper subset of literal x to $\overline{x}$ ($\overline{x}$ to x).

Theorem 2.2.

If $f \in NTD$ *then* $f \begin{pmatrix} \cdots & x & \overline{x} & \cdots \\ \cdots & \overline{x} & x & \cdots \end{pmatrix} \in NTD$*, and* $f, f \begin{pmatrix} \cdots & x & \overline{x} & \cdots \\ \cdots & \overline{x} & x & \cdots \end{pmatrix}$ *are neighbor input.*

Proof. It is trivial because of $x, \overline{x}$ symmetry with tautology, and NTD definition;

$$f \begin{pmatrix} \cdots & x & \overline{x} & \cdots \\ \cdots & \overline{x} & x & \cdots \end{pmatrix} \begin{pmatrix} \cdots & x & \overline{x} & \cdots \\ \cdots & \overline{x} & x & \cdots \end{pmatrix} = f \equiv \top$$

$$f \begin{pmatrix} \cdots & x & \overline{x} & \cdots \\ \cdots & \overline{x} & x & \cdots \end{pmatrix} \begin{pmatrix} \cdots & x & \overline{x} & \cdots \\ \cdots & \{x, \overline{x}\} & \{x, \overline{x}\} & \cdots \end{pmatrix}$$

$$= f \begin{pmatrix} \cdots & x & \overline{x} & \cdots \\ \cdots & \{\overline{x}, x\} & \{\overline{x}, x\} & \cdots \end{pmatrix} \not\equiv \top$$ □

Theorem 2.3.

Minimal Tautology DNF (MTD) correspond to NTD.

Proof. Proof this theorem by constructing NTD from MTD.

If $f \in MTD$ and $f \notin NTD$, then there are some variable x that keep tautology to change proper subset of x.

$$f \in MTD \wedge f \notin NTD \to \exists x \left(f \begin{pmatrix} \cdots & x & \overline{x} & \cdots \\ \cdots & \{x, \overline{x}\} & \{x, \overline{x}\} & \cdots \end{pmatrix} \equiv \top \right)$$

Let attach y to $\overline{x}$. y have some relation g with x.

$$f \in MTD \wedge f \notin NTD \to \exists x \left(\left(f \begin{pmatrix} \cdots & x & \overline{x} & \cdots \\ \cdots & \{x, \overline{y}\} & \{x, \overline{y}\} & \cdots \end{pmatrix} \equiv \top \right) \wedge (g(x,y) \equiv \top) \right)$$

However, from $f \in MTD$ then

$(x, y) \to (1,1), (0,0)$

and from $f \begin{pmatrix} \cdots & x & \overline{x} & \cdots \\ \cdots & \{x, \overline{y}\} & \{x, \overline{y}\} & \cdots \end{pmatrix} \equiv \top$ then

$(x, y) \to (1,0), (0,1)$

So

$(x, y) \to (1,1), (0,0), (1,0), (0,1)$

and g is no bind. So

$$f \in MTD \wedge f \notin NTD \to \exists x \left(f \begin{pmatrix} \cdots & x & \overline{x} & \cdots \\ \cdots & \{x, \overline{y}\} & \{x, \overline{y}\} & \cdots \end{pmatrix} \equiv \top \right)$$

This means

$$f \in MTD \wedge f \notin NTD \to \exists x \left(f \begin{pmatrix} \cdots & x & \overline{x} & \cdots \\ \cdots & \{x, \overline{y}\} & \{x, \overline{y}\} & \cdots \end{pmatrix} \in MTD \right)$$

y: free variable.

On the other hand, each MTD have limitation of length and number of variables type. So we can repeat this operation to any proper subset of variables cannot change another free variable. Such MTD satisfy NTD condition. □

x, y of NTD that made by 2.3 is independent each other, but we can modify easily to depend x, y each other.

Theorem 2.4.

There is some DNF f which;

a) become 1 at one of any set of truth value assignment T

$\forall T \forall t \in T \left(f \left(t \right) = 1 \right)$

b) each clauses have any one of 3 variables conbination. We can only decide these literal become positive or negative.

c) number of clauses is atmost polynomial size of variables type.

Proof. Let 3-clauses $c_1, c_2, \cdots c_n$ that variables is $x_1, x_2, \cdots x_k$become true at truth value assignment $\{t\}$, and c_1 include variables x_1, x_2, x_3. Because we can decide positive / negative of x_1, x_2, x_3 in c_1, so c_1 is possible 8 petterns;

$x_1 \wedge x_2 \wedge x_3$, $\overline{x_1} \wedge x_2 \wedge x_3$, $x_1 \wedge \overline{x_2} \wedge x_3$,

$\overline{x_1} \wedge \overline{x_2} \wedge x_3$,$x_1 \wedge x_2 \wedge \overline{x_3}$, $\overline{x_1} \wedge x_2 \wedge \overline{x_3}$,

$x_1 \wedge \overline{x_2} \wedge \overline{x_3}$, $\overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3}$

$\overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3}$

These possible c_1 become partition of trueth value assignment, one of above c_1 become true at least $\frac{1}{8}$ of truth value assignment $\{t\}$. So we can reduce number of $\{t\}$ at most $\frac{7}{8}$ to decide suitable positive / negative pettern as c_1.

Above condition is applicatable another clauses $c_2, \cdots c_n$, so we can decide positive / negative of variables $x_1, x_2, \cdots x_k$ in $c_2, \cdots c_n$ one by one to reduce $\{t\}$ at most $\frac{7}{8}$. Number of $\left|\{t\}\right|$ is at most 2^k, therefore some constant $d$that $2^k \times \left({}^7\!/_8 \right)^{n^d} \rightarrow 0$, and $\{t\}$ of $x_1, x_2, \cdots x_k$ become 1 at least one of clauses $c_1, c_2, \cdots c_n$ that n is polinomial size of k. □

Theorem 2.5.

Any NTD can convert some NTD that have all pair of variables in clauses atmost polynomial number of variables types.

Proof. If NTD f does not have clauses which include both x and y, we can make another NTD f' that include x, y in same clause with following step;

1) add literal y or $\overline{y}$ to some clauses c that include $x, \overline{x}$.

2) add new clauses d which include x, y and complement all truth value assignment $\{t\}$ that $c\left(t\right) = 1$ and $\left(c \wedge \{y, \overline{y}\}\right)\left(t\right) = 0$.

Mentioned above 2.4, clauses which include any variables and which number is polynomial of variables type can complement any truth value assignment. So $|f'|$ is polynomial size of $|f|$ because number of variables type in f is linear size of $|f|$. □

Theorem 2.6.

If NTD f keeps same clauses to permutate literal $x, \overline{x}$, there are some NTD f' that does not keep same clauses to permutate literal $x, \overline{x}$.

Proof. To modify methods mentioned above proof 2.5, we can easily make f' from f. In 2) step, we choose some variables set that do not same variables set in any clauses of f (and also another clauses of f'), these clauses does not become symmetory. □

Theorem 2.7.

$NTD \in PH$

Proof. We can solve NTD by computing;

a) input as TAUT problem, and

b) all input that change any proper subset of one type literal as non TAUT problem.

b) can compute that choice changing literal as existence, and compute them as non TAUT problem. coNP Oracle machine with TAUT oracle can compute this problem. Therefore NTD is in PH. □

Theorem 2.8.

If input of NTD have some clauses which include variables x, y, the input that change variables y to x (and reduce all $x \wedge x \rightarrow x$, $x \wedge \overline{x} \rightarrow 0$ to become indistinguishable what variables changed) also in NTD.

$$\forall p \in NTD \left(\exists x, y \in p \, (x, y \in c \in p) \rightarrow q \in NTD \mid q = p \begin{pmatrix} \cdots & x & \overline{x} & y & \overline{y} & \cdots \\ \cdots & x & \overline{x} & x & \overline{x} & \cdots \end{pmatrix} \right)$$

$x, y \in c \in p$: DNF p have some clauses c that include variable x, y.

Proof. (Proof by contradiction.) Assume to the contrary that

$$\exists p \in NTD \left(\exists x, y \in p \left(x, y \in c \in p \right) \wedge q \notin NTD \mid q = p \begin{pmatrix} \cdots & x & \overline{x} & y & \overline{y} & \cdots \\ \cdots & x & \overline{x} & x & \overline{x} & \cdots \end{pmatrix} \right)$$

Because of $p \equiv \top$, it is trivial that $q \equiv \top$ and $q \begin{pmatrix} \cdots & x & \overline{x} & \cdots \\ \cdots & \overline{x} & x & \cdots \end{pmatrix} \equiv \top$. So some $q \begin{pmatrix} \cdots & x & \overline{x} & \cdots \\ \cdots & \{x, \overline{x}\} & \{x, \overline{x}\} & \cdots \end{pmatrix} \equiv \top$ from assumption $q \notin NTD$.

However,

$p \in NTD \rightarrow p \begin{pmatrix} \cdots & x & \overline{x} & \cdots \\ \cdots & \{x, \overline{x}\} & \{x, \overline{x}\} & \cdots \end{pmatrix} \not\equiv \top, p \begin{pmatrix} \cdots & y & \overline{y} & \cdots \\ \cdots & \{y, \overline{y}\} & \{y, \overline{y}\} & \cdots \end{pmatrix} \not\equiv \top$

So following are only tautology of changing positive / negative variables

$p \begin{pmatrix} \cdots & x & \overline{x} & y & \overline{y} & \cdots \\ \cdots & \overline{x} & x & y & \overline{y} & \cdots \end{pmatrix} \equiv \top, p \begin{pmatrix} \cdots & x & \overline{x} & y & \overline{y} & \cdots \\ \cdots & x & \overline{x} & \overline{y} & y & \cdots \end{pmatrix} \equiv \top$

then q satisfy following conditions.

$q \begin{pmatrix} \cdots & x & \overline{x} & x & \overline{x} & \cdots \\ \cdots & \overline{x} & x & x & \overline{x} & \cdots \end{pmatrix} \equiv \top, q \begin{pmatrix} \cdots & x & \overline{x} & x & \overline{x} & \cdots \\ \cdots & x & \overline{x} & \overline{x} & x & \cdots \end{pmatrix} \equiv \top$

This means that we have to treat each x, y in $q = p \begin{pmatrix} \cdots & x & \overline{x} & y & \overline{y} & \cdots \\ \cdots & x & \overline{x} & x & \overline{x} & \cdots \end{pmatrix}$ separately. That is, $q = p \begin{pmatrix} \cdots & x & \overline{x} & y & \overline{y} & \cdots \\ \cdots & x & \overline{x} & x & \overline{x} & \cdots \end{pmatrix}$ is irreducible about $x \wedge x \rightarrow x, x \wedge \overline{x} \rightarrow \bot$, so $\forall x, y \in p \left(x, y \notin c \in p \right)$. This is contradict assumption $\exists x, y \in p \left(x, y \in c \in p \right)$. □

Theorem 2.9.

Size of neighbor input in NTD is over polynomial size of input length.

Proof. Mentioned above 2.8, if $p \in NTD$ and exists $x, y \in c \in p$ then $q \in NTD \mid q = p \begin{pmatrix} \cdots & x & \overline{x} & y & \overline{y} & \cdots \\ \cdots & x & \overline{x} & x & \overline{x} & \cdots \end{pmatrix}$. Because of symmetry of $y, \overline{y}$ in tautology, $q' \in NTD \mid q' = p \begin{pmatrix} \cdots & x & \overline{x} & y & \overline{y} & \cdots \\ \cdots & x & \overline{x} & \overline{x} & x & \cdots \end{pmatrix}$ also true. Let $p_{xy} =$

$p\begin{pmatrix} \cdots & x & \overline{x} & y & \overline{y} & \cdots \\ \cdots & x & \overline{x} & x & \overline{x} & \cdots \end{pmatrix}$ and $p_{x\overline{y}} = p\begin{pmatrix} \cdots & x & \overline{x} & y & \overline{y} & \cdots \\ \cdots & x & \overline{x} & \overline{x} & x & \cdots \end{pmatrix}$, and these formula is adjusted input length equal $|p|$ with filler to simplify following proof.

Mentioned above 2.52.6, some NTD have all pair of variables in clauses, and each literal pair is asymmetry, so we can repeat $p_{xy\cdots}$ to all variables. Number of such variables type is over logarithm size. So total number of $\{p_{xy\cdots}\}$ is over polynomial size. Mentioned above 2.2, each of $\{p_{xy\cdots}\}$ is also neighbor input. Therefore size of neighbor input in NTD also over polynomial size. □

Theorem 2.10.

Size of NNF circuit family that compute NTD is over polynomial size of input length.

Proof. Mentioned above 1.4, NNF have to unique gate which Different variables of neighbor input. Mentioned above 2.9, size of NTD is over polynomial size of input length. Therefore size of NNF circuit family that compute NTD is over polynomial size. □

Theorem 2.11.

$P \subsetneq PH$

Proof. Mentioned in [Sipser], NNF circuit family can emulate DTM with polynomial size of DTM computation time. However mentioned above 2.10, NNF have to use gates which number is over polynomial size of input length, and mentioned above 2.7, $NTD \in PH$. Therefore P does not include NTD, and PH is not in P. □

References

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