

ALMOST ALL MONOTONE CIRCUIT FAMILY

KOJI KOBAYASHI

ABSTRACT. This paper describes about “Almost all monotone circuit family” and introduce its interesting advantages to measuring problem complexity.

Explained in Michael Sipser “Introduction to the Theory of COMPUTATION”, circuit family that emulate Deterministic Turing machine (DTM) are almost all monotone circuit family except some NOT-gate that connect input variables (like negation normal form (NNF)).

This “NNF Circuit family” have good characteristic which present each accept input exclusivity and symmetry. Each input make some INPUT-gate and NOT-gate output 1 which set is different from another input, and meet these output in OR-gate and finally connect (specified) OUTPUT-gate. That is, NNF circuit start accept input that exclusive each other, and meet each input as symmetry input step by step and finally goal same output. Especially, some accept input which sandwich reject inputs correspond with unique gate. That is, we can measure problem complexity by using NNF circuit and sandwich inputs. For examples, such sandwich inputs include DNF tautology problem.

1. NNF CIRCUIT FAMILY

Explained in [Sipser] Circuit Complexity section (Theorem 9.30), Circuit family can emulate DTM only using NOT-gate in changing input values $\{0, 1\}$ to $\{01, 10\}$. This “almost all monotone circuit family” have simple structure like monotone circuit family.

Definition 1.1.

We will use the term;

“NNF” as Negation Normal Form.

“DTM” as Deterministic Turing Machine.

Date: 2018-03-24.

“DNF” as Disjunctive Normal Form.

“CNF” as Conjunctive Normal Form.

“NNF Circuit Family” as circuit family that have no NOT-gate except connecting INPUT-gates directly (like negation normal form).

“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.

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

“Boundary input of neighbor input” as reject inputs 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.

“Sandwich structure” as input structure of neighbor input and boundary input in Hamming space.

“Effective circuit of input t ” as one of minimal sub circuit of NNF 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.

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

To simplify, we treat DNF as set of clauses, and also treat clause as set of literals.

Theorem 1.2.

Let $t : N \rightarrow N$ be a function where $t(n) \geq n$.

If $A \in TIME(t(n))$ then NNF circuit family can emulate DTM that compute A with $O(t^2(n))$ gate.

Proof. This Proof is based on [Sipser] proof (Theorem 9.30).

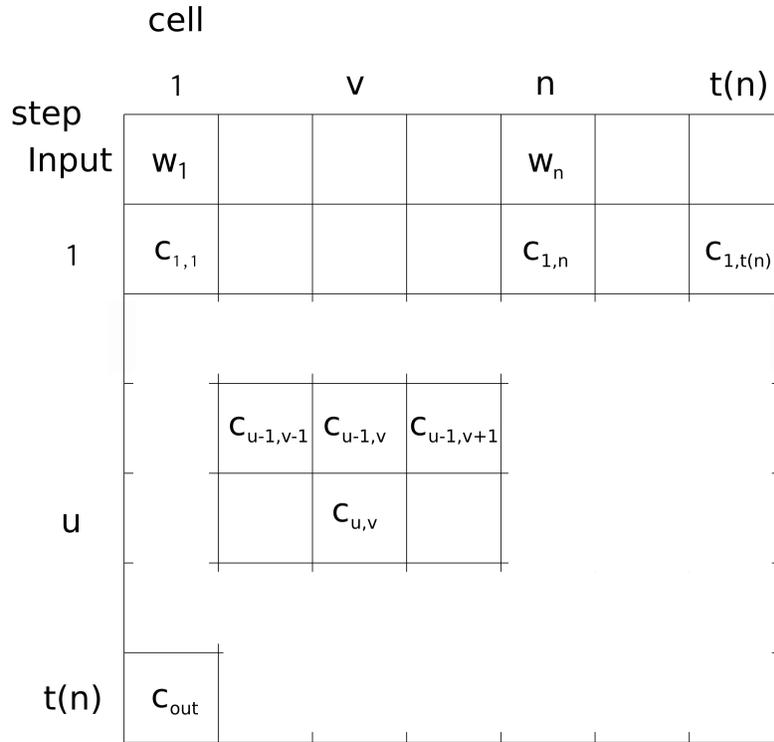


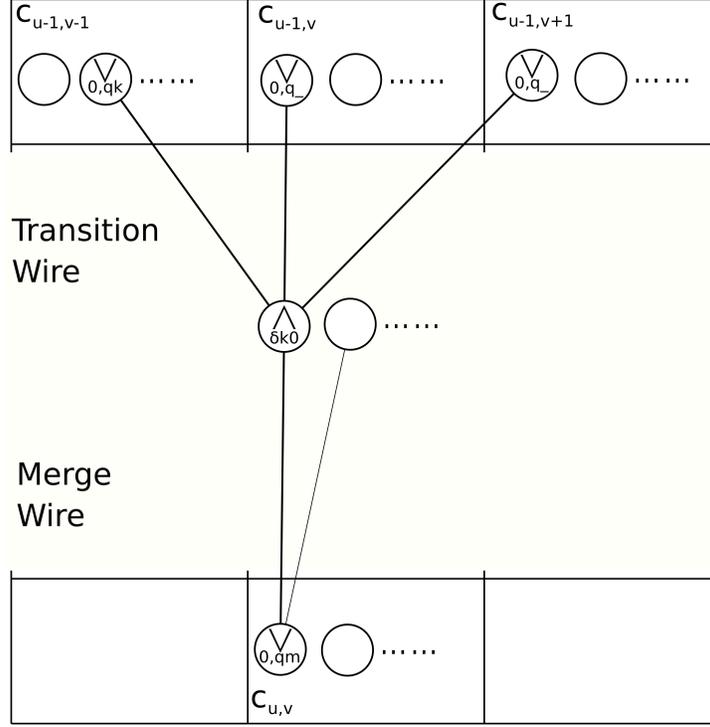
FIGURE 1.1. NNF circuit block diagram

NNF circuit family can emulate DTM by computing every step's cell values (and head state if head on the cell). Figure 1.1 shows part of a NNF circuit block diagram.

Input of this circuit is modified $w_1 \cdots w_n$ to $c_{1,1} \cdots c_{1,n}$, and finally output result at $c_{out} = c_{t(n),1}$ cell. This circuit emulate DTM behavior, so $c_{u,v}$ compute cell's state of step u from previous step cell $c_{u-1,v}$ and each side cells $c_{u-1,v-1}, c_{u-1,v+1}$ (because head affect atmost side cells in each step).

Figure 1.2 shows part of $c_{u,v}$ circuit that transition function is "if state is q_k and tape value is 0, then move +1 and change state to q_m ". This circuit shows one of transition configuration which $(c_{u-1,v}, c_{u-1,v+1}) = (0, 0)$. q_- means "no head on the cell".

Each OR-gate $\vee_{w,q}$ in $c_{u,v}$ correspond to every step's cell condition (cell value w , and head status q if head exist on the $c_{u,v}$ cell), and output 1 if and only if

FIGURE 1.2. $c_{u,v}$ circuit

$c_{u,v}$ cell satisfy corresponding condition. Previous step's \vee output in $c_{u-1,v-1}$, $c_{u-1,v}$, $c_{u-1,v+1}$ are connected to next step's AND-gate \wedge_{δ} in $c_{u,v}$ with transition wire. Each \wedge_{δ} correspond to transition function δ , and each \wedge_{δ} output correspond to each transition function's result of $c_{u,v}$. So \wedge_{δ} in $c_{u,v}$ output 1 if and only if previous step's \vee output in $c_{u-1,v-1}$, $c_{u-1,v}$, $c_{u-1,v+1}$ satisfy transition function δ condition. Each transition functions affect (or do not affect) next step's condition, so \wedge_{δ} output is connected to $\vee_{w,qm}$ in $c_{u,v}$ and decide $c_{u,v}$ condition. Because DTM have constant number of transition functions, NNF can compute each step's cell by using constant number of AND-gates and OR-gates (without NOT-gate).

First step's cells are handled in a special way. Input is $\{0, 1\}^*$ and above monotone circuit cannot manage 0 value. So NNF circuit compute $\{0, 1\}^* \rightarrow \{01, 10\}^*$ by using NOT-gate.

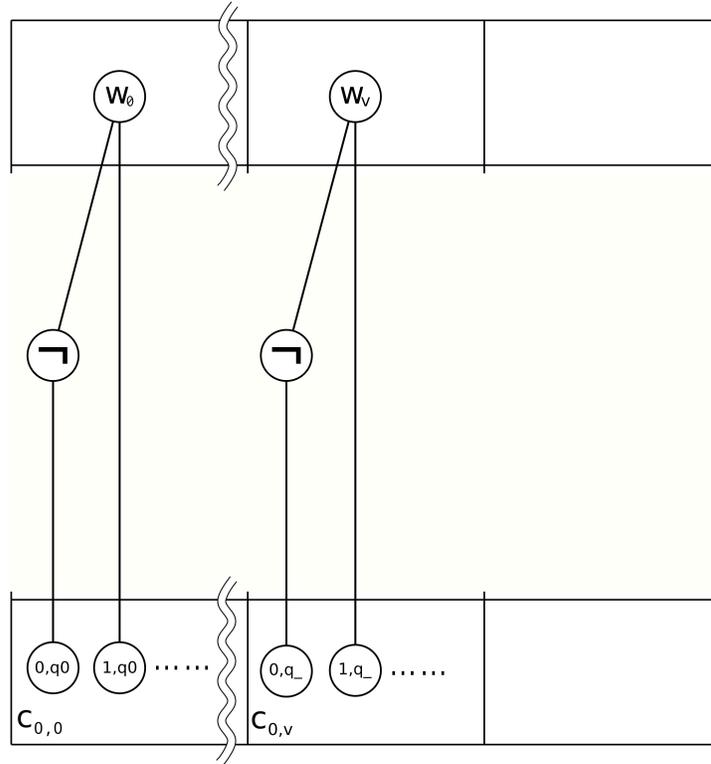


FIGURE 1.3. First step

□

Corollary 1.3.

NNF circuit family can compute P problem with polynomial number of gates of input length.

Confirm NNF circuit family behavior. NNF circuit family can emulate DTM with polynomial number of gate of DTM computation time. All effective circuit become DAG that root is specified OUTPUT-gate. All gates that include effective circuit become 1 if OUTPUT-gate become 1. Especially, all different variables of input cannot overlay in same input, so all effective circuits (with different inputs) are join at OR-gate to connect OUTPUT-gate. This NNF circuit behavior clarify input exclusivity and symmetry.

Theorem 1.4.

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

Proof. Because input variable pair does not become 1 in same input, so it is necessary to join OR-gate and output 1 to connect OUTPUT-gate in effective circuit. \square

Theorem 1.5.

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

Proof. Mentioned above 1.4, all accept input variable pair of different variables join at OR-gate. Because NNF circuit is almost all monotone circuit, there is two case of joining at 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 at OR-gate these AND-gate output, and meet at AND-gate all OR-gate output.

Case a), some AND-gate become 1 if and only if input include one side of different variables. Therefore, trunk of these AND-gate does not become 1 if AND-gate does not include these different variables as input of sub circuit that connect trunk.

Case b), because no boundary input become accept input, some OR-gate which join neighbor input become 0 if input include boundary input. That is, effective circuit become 0 if some of these OR-gate become 0, and become 1 if all of these OR-gate become 1. Therefore, it is necessary that effective circuit include AND-gate that meet all these OR-gate which join different variables (and other same variables). Such AND-gate correspond to each different variables pair. So AND-gate is differ from each different variables pair. \square

That is to say, neighbor input with sandwich structure cannot permutate proper partial input of different variables . NNF circuit have to use unique AND-gate to differentiate neighbor input and boundary inputs.

2. NEIGHBOR PROBLEMS

Let analyze sandwich structure of problem. Some P problem have specific characteristic that exist neighbor input atmost logarithm distance which reach to trivial accept input. For example, each accept input of Horn DNF Tautology problem (negation of Horn CNF SAT problem $\overline{HornSAT}$) have sandwich structure which maximum distance is logarithm size and we can trace to tautology 1 step by step. However some input of DNF Tautology probelm (negation of CNF SAT problem \overline{CNFSAT}) have over logarithm distance.

Theorem 2.1.

$t \in \overline{HornSAT}$ have neighbor input which reach to tautology 1 within logarithm distance in Hamming space.

Proof. $t \in \overline{HornSAT}$ can reduce another $t_i \in \overline{HornSAT}$ that delete some literal \bar{x} which $x = c \in t$. If t_i is overwrite \bar{x} in t with filler, size of different variables between t, t_i is atmost $O(\log |[t]|)$ because $|\bar{x}| < O(\log |[t]|)$. We can repeat each overwriting to reach tautology 1. Therefore we obtain theorem. \square

Theorem 2.2.

$f \in \overline{CNFSAT}$ have neighbor input that permutate all of one variables x, \bar{x} to \bar{x}, x .

$$\exists f \in \overline{CNFSAT}, x \in f \left(f \begin{pmatrix} \cdots & x & \bar{x} & \cdots \\ \cdots & \bar{x} & x & \cdots \end{pmatrix} = 1, f \begin{pmatrix} \cdots & \{x, x\} & \{\bar{x}, \bar{x}\} & \cdots \\ \cdots & \{x, \bar{x}\} & \{\bar{x}, x\} & \cdots \end{pmatrix} = 0 \right)$$

$$\begin{pmatrix} \cdots & x & \bar{x} & \cdots \\ \cdots & \bar{x} & x & \cdots \end{pmatrix} : \text{permutate all } x, \bar{x} \text{ to } \bar{x}, x.$$

$$\begin{pmatrix} \cdots & \{x, x\} & \{\bar{x}, \bar{x}\} & \cdots \\ \cdots & \{x, \bar{x}\} & \{\bar{x}, x\} & \cdots \end{pmatrix} : \text{permutate (any) proper subset of } x, \bar{x} \text{ to } \bar{x}, x.$$

Proof. We can make f by applying $g \in \overline{CNFSAT}$ following step;

a) Delete all clauses $d \in g$ which redundant tautology

$$g \rightarrow g' \mid (d \notin g') \wedge (d \vee g' = g)$$

$$d(t) = 1 \rightarrow \exists d_k \in g (d_k(t) = 1)$$

b) Change all proper subset of literal x, \bar{x} to another literal of free variables y, \bar{y}

which become tautology when permutate x, \bar{x} to \bar{x}, x .

$$g \rightarrow g' = g \begin{pmatrix} \cdots & \{x, x\} & \{\bar{x}, \bar{x}\} & \cdots \\ \cdots & \{x, \bar{y}\} & \{\bar{x}, y\} & \cdots \end{pmatrix}$$

$$g \begin{pmatrix} \cdots & \{x, x\} & \{\bar{x}, \bar{x}\} & \cdots \\ \cdots & \{x, \bar{x}\} & \{\bar{x}, x\} & \cdots \end{pmatrix} = 1$$

a) is trivial that keep tautology.

b) also keep tautology because

$$g = g \begin{pmatrix} \cdots & \{x_p, x_q\} & \{\bar{x}_p, \bar{x}_q\} & \cdots \\ \cdots & \{x_p, \bar{x}_q\} & \{\bar{x}_p, x_q\} & \cdots \end{pmatrix} = 1$$

x_p, x_q : any partition of x

means

$$(x_p, x_q) = (1, 1), (0, 0) \text{ (from } g = 1)$$

$$(x_p, x_q) = (1, 0), (0, 1) \text{ (from } g \begin{pmatrix} \cdots & \{x_p, x_q\} & \{\bar{x}_p, \bar{x}_q\} & \cdots \\ \cdots & \{x_p, \bar{x}_q\} & \{\bar{x}_p, x_q\} & \cdots \end{pmatrix} = 1)$$

So

$$(x_p, x_q) = (1, 1), (0, 0), (1, 0), (0, 1)$$

and x_q can permutate free variable y .

Each a) b) step reduce number of x in g . So finally these a) b) step stop and we get f that do not become tautology when we permutate proper subset of literal x, \bar{x} . Therefore we obtain theorem. \square

In other words, there is some difference between $\overline{HornSAT}$ and \overline{CNFSAT} which indicated sandwich structure in Hamming space, and these sandwich structure correspond to NNF circuit family.

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