Completeness on Multiple-valued Logical Functions Realized by Asynchronous Sequential Circuits

Hisashi SATO

Completeness on Multiple-valued Logical Functions Realized by Asynchronous Sequential Circuits

非同期式順序回路によって定義される多値論理関数の完全性について

Hisashi SATO

Contents

1 Introduction **2**

2 Asynchronous Circuit **8**

3 Output Stability **11**

4 Realization of Logical Functions by Asynchronous Sequential Circuits **13**

4.1 Realization 13

4.2 Initialization 14

5 Classical Completeness Problem **15**

5.1 Functional Completeness 15

5.2 Examples of Complete sets 15

5.3 Completeness Criteria 20

5.4 Completeness under "loop-free" 23

6 Completeness under the General Initialization Assumption **25**

7 Completeness for Restricted Sequential Circuit under the General Initialization Assumption **32**

7.1 GR-completeness 32

7.2 GR-completeness criterion for the binary case 34

8 Completeness under the Initialization-by-Input Assumption **40**

8.1 NS-completeness 40

8.2 Examples of NS-complete and NS-incomplete sets 42

8.3 NS-completeness criterion for the binary case 46

References **53**

1 Introduction

In this paper, we are concerned with "completeness problem" for logical functions realized by asynchronous circuits.

Let $k \geq 2$ and $X_k = \{0, 1, \ldots, k-1\}$. We denote by Ω_k the set of all functions with arbitrary numbers of variables defined over X_k. We call elements of Ω_k (k-valued) logical functions.

The classical completeness problem in logic theory deals with logical functions. A set of logical functions is said to be functionally complete (or simply,complete) if it generates all logical functions by composition. Several criteria have been given by E.L. Post [21] for the binary case, and J. Slupecki [28] and J.W. Butler [1] for the general case.

S.V. Jablonskiĭ [9] determined all maximal incomplete sets in Ω_3. I.Rosenberg [22, 23] have refined these results by characterizing maximal incomplete sets for general cases.

A switching gate is an switching gate which receives input signals $x_1, \ldots, x_n$ and transforms the input signals into the output signal y uniquely determined by the input signals. The relation between input $x_1, \ldots, x_n$ and output y is represented by a logical function f as $y = f(x_1, \ldots, x_m)$.

But, in operations of switching gates there is a certain time-lag between the input and the output. This "delay" is usually very small compared with the duration time of input signals, and is often disregarded. In classical switching theory, "delay-less" functions are considered as basic elements to construct circuits. In such circuits, feedback loops are not allowed, since a loop of "delay-less elements" works contradictorily.

Admittance of "logical gate with delay" has a strong influence on the completeness. In fact, J. von Neumann showed that NAND gate with unit delay can not generate all logical functions[16]. Although the NAND function can generate all logical functions. V.B. Kudryavcev [11, 12] considered the classical completeness problem by introducing a "function with time delay", i.e. a pair (f, d) of a logical function f and a nonnegative integer d such a pair is called an indexed operator, which may be considered as a model of a switching gate which carries out an operation in a definite time delay.

A set Σ of indexed operators is said to be $\sim$-complete if for any logical function f there exists a nonnegative integer d such that an indexed operator (f, d) can be obtained by composing the elements of the set Σ "combinationally." Let Φ be a set of indexed operators. We denote by $\hat{\Phi}$ the minimum set of indexed operators defined as follows:

1. $(I, 0) \in \hat{\Phi}$

2. If $(f, d) \in \hat{\Phi}$ and $(g_1, d'), \ldots, (g_m, d') \in \hat{\Phi}$ then $(h, d+d') \in \hat{\Phi}$ for $\forall h = f(g_{i_1}, \ldots, g_{i_n})$.

V.B. Kudryavcev gave a completeness criterion for the binary case by giving all the maximal incomplete sets of these pairs explicitly. A.Nozaki [17] redefined this problem in mathematically clear form and gave a criterion for the general case. T. Hikita [2] obtained an effective criterion for $\sim$-completeness and charactrizations of the maximal incompleteness and he determined all the maximal incomplete sets in the ternary case.

We regard any circuit as a direct graph, i.e. a switching gate in a circuit is a vertex and for two switching gates f_1, f_2 in a circuit where an output of f_1 is one of inputs for f_2, (f_1, f_2) is a edge. A circuit C is said to be a sequential circuit if the graph defined by the circuit C admits cycles in graph theory. We call such cycles "feedback loops."

However actual gates have always delay. Admittance of "feedback loops" has a strong influence on the completeness [6, 7, 20]. For instance, a simple circuit shown in Figure 1 supplies the stable output "0", only if it has received the input signal "0" in advance, once for all. It should be noted that any "loop-free" circuit consisting of "AND" gates cannot realize the constant function. This is why we consider the realization of logical functions by sequential circuits, i.e., the circuits which may contain "feedback loops".

K.Inagaki [6, 7] introduced several types of completeness of binary logical gates with unit-time delay based on sequential circuits(t_6-completeness,weakly t-completeness and f-completeness). He gave completeness criterion for each type of completeness. A.Nozaki [17] redefined this problem in mathematically clear form(S-completeness) and gave a criterion for S-completeness.

A sequential circuit can be represented by simultaneous equations. For instance, the sequential circuit shown in Figure 2 is represented as follows.

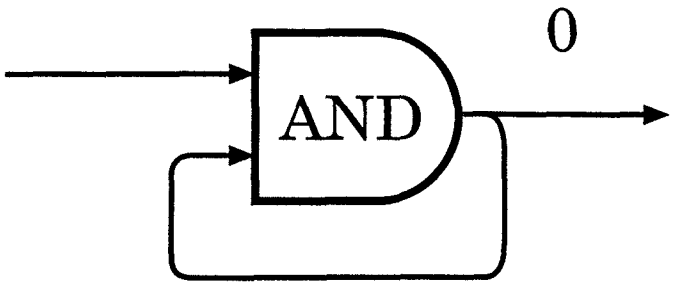

Figure 1: Realization of the constant function c_0 by AND

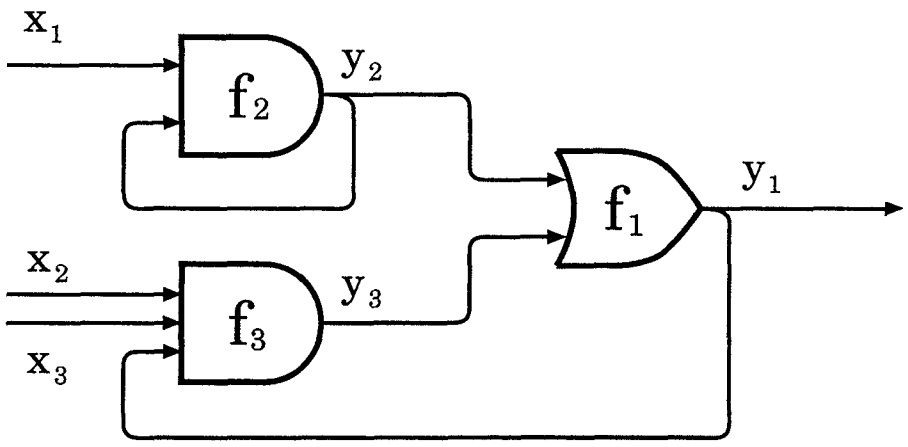

Figure 2: Example of sequential circuit

$$\begin{cases} y_1(t+e_1) &= f_1(y_2(t), y_3(t)) \\ y_2(t+e_2) &= f_2(x_1, y_2(t)) \\ y_3(t+e_3) &= f_3(x_2, x_3, y_1(t)), \end{cases}$$

where the symbol $y_i(t)$ stands for the signal y_i at the moment t, and e_i is a certain non-negative integer representing the delay of the i-th element. Although the input signals are usually given synchronously at the same moment, the operations of the gates in the circuit may not be synchronized. So we have to study asynchronous circuits.

Indeed, asynchronous circuits have been investigated from the viewpoint of electronics. Huffman[3, 4] introduced a model of asynchronous circuits, which we call feedback-delay model or Huffman model. Concepts related to his model, such as flow of states, hazard and critical race have been studied. For example, . Main interests in Huffman model are to explain characteristics peculiar to asynchronous circuits and the logical design for asynchronous circuits. Muller introduced another model of asynchronous circuit, which we call speed independent model or Muller model[15]. A circuit in Muller model dose not have input and output signals. So main interests in Muller model are transitions of states in an asynchronous circuit and "uniqueness of final state" in an asynchronous circuit.

So far completeness problem has not been investigated for asynchronous circuits. In this paper, we shall give mathematical definitions of asynchronous circuits and the realization of logical functions by means of asynchronous circuits. Our main purpose here is to solve the completeness problem for multiple-valued logical functions realized asynchronous circuits.

First of all, we define several concepts such as an admissible sequence, stable output. Our formulation for the realization of a logical function based on an asynchronous circuit relies on these concepts. As regards an input assumption, we assume that input signals do not change until a state of a circuit become stable. This formulation may be considered to be a model of asynchronous sequential circuits, in which each switching gate carries out an operation in a finite time delay. Introducing four types of completeness(LF-,GS-,GR-,NS-completeness) of a set of logical functions, we give a completeness criterion of each type of completeness. A set of logical function F is said to be LF-complete if any logical function g is realized by a "loop-free circuit" over F. A set of logical functions F is said to be GS-complete if any logical function g is realized by a sequential circuit over F with respect to some non-empty set of states. A set of logical function F is said to be GR-complete if any logical function g is realized by a "short feedback" circuit over F with respect to some non-empty set of states. In GS-completeness and GR-completeness, we assume that we can change the initial state of a circuit by some means. On the other hand, we may assume that we can change the initial state of a circuit only by feeding a certain inputs to the circuit. We call such inputs an initial sequence. A set of logical function F is said to

be NS-complete if any logical function g is realized by a sequential circuit over F by some initial sequence.

We prove that LF-completeness is equivalent to functionally completeness. We determine all maximal GS-incomplete sets for the binary case and the ternary case. Further we obtain charactrizations of the maximal GS-incomplete set for the general case. We determine all maximal GR-incomplete sets for the binary case. A completeness criterion for GR-completeness is given under a strong condition for the general case. Finally, we determine all maximal NS-incomplete sets for the binary case.

Acknowledgment

I wish to express my thanks to Professor S. Iitaka and Professor A. Nozaki for their warm guidance, suggestions and encouragement. I am grateful to Professor S. Shimada and Professor K. Kondo for their warm guidance and encouragement. I am also grateful to Professor T. Mishima for his encouragement.

2 Asynchronous Circuit

Let $M(X, Y)$ be the set of all mappings from a set X to a set Y.

Definition 2.1 *For the set* $X_k = \{0, 1, \cdots, k-1\}$, *let*

(i) $\Omega_{k,n} = \mathrm{Map}(X_k^n, X_k)$

(ii) $\Omega_k = \cup_{n=1}^{\infty} \Omega_{k,n}$

We call an elements of Ω_k **(k-valued) logical functions**. *We denote by* $\mathbf{Const}_k$ *the set of constant functions on* X_k.

A **sequential circuit** C is a system of simultaneous equations as follows:

$$(C) \quad \begin{cases} y_1' &= f_1(y_1, \cdots, y_m, x_1, \cdots, x_n) \\ y_2' &= f_2(y_1, \cdots, y_m, x_1, \cdots, x_n) \\ & \vdots \\ y_m' &= f_m(y_1, \cdots, y_m, x_1, \cdots, x_n) \end{cases}$$

where f_i represents the input-output behavior of the i-th element,whose current output is denoted by y_i, and $x_1, \cdots, x_n$ represent the "external" inputs given from outside of the circuit. The first output y_1 is taken as the external output of the circuit C.

Let F be a set of logical functions. The circuit C is called circuit **over** F iff all component functions f_i's are in F.

Definition 2.2 *A circuit is said to be* **loop-free** *iff the following condition is satisfied.*

The i-th function f_i depends only on $y_{i+1}, \cdots, y_m$ and $x_1, \cdots, x_n$ for all i.

A loop-free circuit can be represented as follows:

$$\begin{cases} y_1' &= f_1(y_2, \cdots, y_m, x_1, \cdots, x_n) \\ y_2' &= f_2(y_3, \cdots, y_m, x_1, \cdots, x_n) \\ & \vdots \\ y_m' &= f_m(x_1, \cdots, x_n) \end{cases}$$

Definition 2.3 *A circuit is said to be* **restricted sequential circuit** (*simply,* **restricted circuit**) *or* **short-loop circuit** *iff the following condition is satisfied.*

The output of an element may be connected to its own input terminal, but no other feed-back loops are allowed.

A restricted sequential circuit can be represented as follows:

$$\begin{cases} y_1' &= f_1(y_1, \cdots, y_m, x_1, \cdots, x_n) \\ y_2' &= f_2(y_2, \cdots, y_m, x_1, \cdots, x_n) \\ & \vdots \\ y_m' &= f_m(y_m, x_1, \cdots, x_n). \end{cases}$$

An m-tuple $\mathbf{y} = (y_1, \cdots, y_m)$ is called **state** of the circuit C. An n-tuple $\mathbf{x} = (x_1, \cdots, x_n)$ is called **input** of the circuit C.

Definition 2.4 *Let* $\mathbf{x} = (x_1, \cdots, x_n)$ *be an arbitrary input of the circuit* C, *let* $\mathbf{y} = (y_1, \cdots, y_m)$ *and* $\mathbf{z} = (z_1, \cdots, z_m)$ *be arbitrary states of the circuit* C. *We say that the state of the circuit* C *can be transferred from* $\mathbf{y}$ *to* $\mathbf{z}$ *by the input* $\mathbf{x}$, *iff*

$$z_i = \begin{cases} y_i \\ \ or \\ f_i(y_1, \cdots, y_m, x_1, \cdots, x_n) \end{cases}$$

for all i $(1 \leq i \leq n)$. *This transition of states is denoted as follows.*

$$\mathbf{y} \xrightarrow{\mathbf{x}} \mathbf{z}.$$

If $z_i = f_i(y_1, \cdots, y_m, x_1, \cdots, x_n)$ *for all* i $(1 \leq i \leq n)$, *then we say the transition is* **synchronous** *and denote it by* $\mathbf{z} = [\mathbf{y}(t)]_{\mathbf{x}}$.

A circuit $\mathcal{C}$ is said to be a synchronous circuit if any transitions of the circuit $\mathcal{C}$ are synchronous. Otherwise we call the circuit $\mathcal{C}$ an asynchronous circuit.

Definition 2.5 *An infinite sequence of state transitions*

$$\mathbf{y}(0) \xrightarrow{\mathbf{x}} \mathbf{y}(1) \xrightarrow{\mathbf{x}} \mathbf{y}(2) \xrightarrow{\mathbf{x}} \cdots \xrightarrow{\mathbf{x}} \mathbf{y}(t) \xrightarrow{\mathbf{x}} \cdots$$

is said to be **admissible** *for an input* $\mathbf{x}$ *iff the following condition satisfied.*

Let t_0 be a non-negative integer and i a positive integer not greater than m. Let c be an element of X_k. If $f_i(\mathbf{y}(t), \mathbf{x}) = c$ for all $t \geq t_0$, then there exists an integer $T \geq t_0$ such that $\mathbf{y}_i(t) = c$ for any $t \geq T$, where $\mathbf{y}_i(t)$ denotes the i-th component of $\mathbf{y}(t)$.

Remark 2.6 *In actual circuit, we can assume the existence of a positive constant K satisfying the following condition:*

$$\text{If } f_i(\mathbf{y}(t), \mathbf{x}) = c \text{ for } T \leq t < T + K \text{ , then } y_i(T + k) = c.$$

Proposition 2.7 *Let*

$$\mathbf{y}(0) \xrightarrow{\mathbf{x}} \mathbf{y}(1) \xrightarrow{\mathbf{x}} \mathbf{y}(2) \xrightarrow{\mathbf{x}} \cdots$$

be an admissible sequence.

(i) *Any subsequence of the form:*

$$\mathbf{y}(T) \xrightarrow{\mathbf{x}} \mathbf{y}(T+1) \xrightarrow{\mathbf{x}} \mathbf{y}(T+2) \xrightarrow{\mathbf{x}} \cdots$$

is also admissible.

(ii) *If* $\mathbf{z}(0) \xrightarrow{\mathbf{x}} \mathbf{z}(1) \xrightarrow{\mathbf{x}} \mathbf{z}(2) \xrightarrow{\mathbf{x}} \cdots \xrightarrow{\mathbf{x}} \mathbf{z}(t)$ *and* $\mathbf{z}(t) = \mathbf{y}(t')$ *then the combined sequence*

$$\mathbf{z}(0) \xrightarrow{\mathbf{x}} \mathbf{z}(1) \xrightarrow{\mathbf{x}} \mathbf{z}(2) \xrightarrow{\mathbf{x}} \cdots \xrightarrow{\mathbf{x}} \mathbf{z}(t) \xrightarrow{\mathbf{x}} \mathbf{y}(t'+1) \xrightarrow{\mathbf{x}} \mathbf{y}(t'+2) \xrightarrow{\mathbf{x}} \cdots$$

is also admissible.

Proof

Obvious by Definition 2.5. □

3 Output Stability

Definition 3.1 *A* **synchronous sequence** *for an input* x *is a sequence of the form:*

$$\mathrm{y}(0) \xrightarrow{\mathrm{x}} \mathrm{y}(1) \xrightarrow{\mathrm{x}} \mathrm{y}(2) \xrightarrow{\mathrm{x}} \cdots$$

where $\mathrm{y}(t+1) = [\mathrm{y}(t)]_{\mathrm{x}}$ *for all non-negative integer* t.

Remark 3.2 *A synchronous sequence is always admissible.*

Definition 3.3 *An infinite sequence*

$$\mathrm{y}(0) \xrightarrow{\mathrm{x}} \mathrm{y}(1) \xrightarrow{\mathrm{x}} \mathrm{y}(2) \xrightarrow{\mathrm{x}} \cdots$$

is said to be **output stable** *iff there exists a non-negative integer* t_0 *such that*

$$\mathrm{y}_1(t_0) = \mathrm{y}_1(t_0+1) = \mathrm{y}_1(t_0+2) = \cdots .$$

The state $\mathrm{y}(t_0)$ *is called the* **stable state** *of the sequence and the output* $\mathrm{y}_1(t_0)$ **stable output** *of the sequence.*

Definition 3.4 *Let* t_0 *be a non-negative integer. Let*

$$\mathrm{y}(0) \xrightarrow{\mathrm{x}} \mathrm{y}(1) \xrightarrow{\mathrm{x}} \mathrm{y}(2) \xrightarrow{\mathrm{x}} \cdots$$

be an admissible sequence. If a state $\mathrm{y}(t_0)$ *in the sequence satisfies the following simultaneous equation*

$$\begin{cases} y_1 &= f_1(y_1, \cdots, y_m, x_1, \cdots, x_n) \\ y_2 &= f_2(y_1, \cdots, y_m, x_1, \cdots, x_n) \\ &\vdots \\ y_m &= f_m(y_1, \cdots, y_m, x_1, \cdots, x_n) \ , \end{cases}$$

then we have $\mathrm{y}(t_0) = \mathrm{y}(t_0+1) = \mathrm{y}(t_0+2) = \cdots$. *We call such a sequence to be* **terminating**, *and the state* $\mathrm{y}(t_0)$ **stable state.**

Remark 3.5 *A terminating sequence is always output stable.*

Proposition 3.6 *Every admissible sequence of a loop-free circuit is terminating.*

Proof

Let

$$\mathbf{y}(0) \xrightarrow{\mathbf{x}} \mathbf{y}(1) \xrightarrow{\mathbf{x}} \mathbf{y}(2) \xrightarrow{\mathbf{x}} \cdots \xrightarrow{\mathbf{x}} \mathbf{y}(t) \xrightarrow{\mathbf{x}} \cdots$$

be an admissible sequence of a loop-free circuit. By definition of admissible sequence, for each $1 \leq m \leq m$ there exists t_m such that

$$y_m(t) = f_m(x_1, \ldots, x_n) \text{ for all } t > t_m.$$

Assume that there exists t_{k+1} such that

$$\begin{cases} y_{k+1}(t) &= f(y_{k+2}(t_{k+1}), \ldots, y_m(t_{k+1}), x_1, \ldots, x_n) \\ & \vdots \\ y_m(t) &= f(x_1, \ldots, x_n) \end{cases}$$

for all $t > t_{k+1}$. Then there exists t_k such that

$$\begin{cases} y_k(t) &= f(y_{k+1}(t_k), y_{k+2}(t_k), \ldots, y_m(t_k), x_1, \ldots, x_n) \\ y_{k+1}(t) &= f(y_{k+2}(t_k), \ldots, y_m(t_k), x_1, \ldots, x_n) \\ & \vdots \\ y_m(t) &= f(x_1, \ldots, x_n) \end{cases}$$

for all $t > t_k$. By induction, we have

$$\mathbf{y}(t_0) = \mathbf{y}(t_0 + 1) = \mathbf{y}(t_0 + 2) = \cdots$$

for some t_0. □

4 Realization of Logical Functions by Asynchronous Sequential Circuits

4.1 Realization

Definition 4.1 *Let Y be a non-empty set of states. The circuit C* **realizes** *a logical function f in Ω_k, with respect to the set Y iff the following conditions are satisfied for any input* $\mathbf{x} = (x_1, \ldots, x_n)$.

(i) *Every admissible sequence starting from a state in Y for the input* $\mathbf{x}$ *is stable state , and its stable state is uniquely determined by the input* $\mathbf{x}$ *,and its stable output is always identical with* $f(x_1, \ldots, x_n)$.

(ii) *Let* $\mathbf{y}$,$\mathbf{z}$ *be states. If* $\mathbf{y} \in Y$ *and* $\mathbf{y} \xrightarrow{\mathbf{x}} \mathbf{z}$ *for some input* $\mathbf{x}$, *then the state* $\mathbf{z}$ *is also in Y.*

Lemma 4.2 *We consider two circuits C and C', which realize the logical functions f and g, with respect to the state sets Y and Y', respectively:*

$$(C)\begin{cases} y_1' &= f_1(y_1, \ldots, y_m, x_1, \ldots, x_n) \\ &\vdots \\ y_m' &= f_m(y_1, \ldots, y_m, x_1, \ldots, x_n), \end{cases}$$

$$(C')\begin{cases} y_1' &= g_1(y_1, \ldots, y_r, x_1, \ldots, x_s) \\ &\vdots \\ y_r' &= g_r(y_1, \ldots, y_r, x_1, \ldots, x_s). \end{cases}$$

Then the circuit (C'') defined below realizes the function $f(g(x_1, \ldots, x_s), x_{s+1}, \ldots, x_{s+n-1})$ with respect to the set:

$$Y'' = \{(a_1, \ldots, a_m, b_1, \ldots, b_r) \mid (a_1, \ldots, a_m) \in Y \text{ and } (b_1, \ldots, b_r) \in Y'\}.$$

$$(C'') \begin{cases} y'_1 &= f_1(y_1, \ldots, y_m, y_{m+1}, x_{s+1}, \ldots, x_{s+n-1}) \\ & \vdots \\ y'_m &= f_m(y_1, \ldots, y_m, y_{m+1}, x_{s+1}, \ldots, x_{s+n-1}) \\ y'_{m+1} &= g_1(y_{m+1}, \ldots, y_{m+r}, x_1, \ldots, x_s) \\ & \vdots \\ y'_{m+r} &= g_r(y_{m+1}, \ldots, y_{m+r}, x_1, \ldots, x_s). \end{cases}$$

Proof

After the output $y_{m+1}(t)$ of the circuit C' becomes stable, the circuit C in C'' evaluates the value of

$$h(y_{m+1}, x_{s+2}, \ldots, x_{s+n}),$$

which is equal to

$$h(h(x_1, \ldots, x_s,), x_{s+1}, \ldots, x_{s+n-1}).$$

□

4.2 Initialization

There are two contrastive initialization assumptions.

1. **General Initialization Assumption**

   We can set the initial state of a circuit to an arbitrary state by some means.

2. **Initialization-by-Input Assumption**

   We can change the initial state of a circuit only by feeding a certain input sequence to the circuit.

We can use a circuit C for evaluating a function h if it realizes the function with respect to some non-empty set Y of states under the general initialization assumption. Let us consider a circuit C which realizes a function h with respect to a set Y of states under the initialization-by-input assumption. The circuit C can not be used for evaluating the function h practically, unless there is a finite sequence of inputs which can convert all state of the circuit to some states in Y.

5 Classical Completeness Problem

5.1 Functional Completeness

Definition 5.1 *Let F,G be subsets of Ω_k. We donate by $F \otimes G$ the set of all functions h, each of which can be defined by a function $f \in F$ and functions $g_1, \ldots, g_n \in G$ as follows:*

$$h = f(g_1, \ldots, g_n).$$

Definition 5.2 *Let F be a subset of Ω_k. We denote:*

$$\begin{aligned} F^{(0)} &= F \otimes \{I\} \\ F^{(n)} &= F^{(n-1)} \otimes F^{(1)} \\ F^{*} &= \cup_{n=1}^{\infty} F^{(n)} \\ \tilde{F} &= (F \cup \{I\})^{*}, \end{aligned}$$

where the symbol I stands for the identity mapping:$I(x) = x$.

Definition 5.3 *Let F be a set of logical functions.*

(i) *F is said to be* complete *iff $\tilde{F} = \Omega$.*

(ii) *F is said to be* incomplete *iff it is not complete.*

(iii) *F is said to be* maximal incomplete *iff the following conditions are satisfied.*

- *F is incomplete.*
- *$\forall F' : F' \supset F \Longrightarrow F'$: functional complete*

5.2 Examples of Complete sets

Example 5.4 *In binary case, the following sets are known to be complete:*

$$\begin{aligned} H_0 &= \{\mathrm{OR}, \mathrm{NOT}\} \\ H_1 &= \{\mathrm{NAND}\}. \end{aligned}$$

Example 5.5 ([29]) *In general case, Webb showed that the function* webb *can generate all logical functions in* Ω_k.

$$\mathrm{webb}(x,y) \;=\; \max(x,y) \oplus 1.$$

Definition 5.6 *Let* S *be a subset of* X_k. *We denote by* $\mathrm{sel}_S(x,y,z)$ *a k-valued logical function defined as follows:*

$$\mathrm{sel}_S(x,y,z) = \begin{cases} x & \text{if } z \in S \\ y & \text{otherwise} \end{cases}$$

We call a function $\mathrm{sel}_S(x,y,z)$ *a selecting function. We denote by* **Sel** *the set of all selecting functions.*

Proposition 5.7

$$\mathrm{sel}_S(x,y,z) = \mathrm{sel}_{S^c}(y,x,z)$$

In this subsection, we shall explain logical functions generated by selecting functions and constant functions.

Theorem 5.8 *For any* $S = \{x_1, \ldots, x_m\}$, $\mathrm{sel}_S(x,y,z)$ *can be composed by*

$$\mathrm{sel}_{\{x_1\}}(x,y,z), \ldots, \mathrm{sel}_{\{x_m\}}(x,y,z).$$

Proof

We consider the following equations:

$$\begin{cases} o_i \;=\; \mathrm{sel}_{\{x_i\}}(x, o_{i+1}, z) & (1 \le i \le m) \\ o_{m+1} \;=\; y. \end{cases}$$

We can easily verify that $o_1 = \mathrm{sel}_S(x,y,z)$. □

Theorem 5.9 *The set of all selecting functions* **Sel** *is not complete.*

Proof

We denote by $f(x_1, \ldots, x_n)$ a logical function which satisfies the following condition:

$$\exists (a_1, \ldots, a_n) \in X_k^n \text{ such that } f(a_1, \ldots a_n) \notin \{a_1, \ldots, a_n\}.$$

As an output value of a selecting function is equal to one of input values to it, $f(x_1, \ldots, x_n)$ can not be composed by **Sel**. □

Proposition 5.10 *For any* $\mathbf{a} = (a_1, \ldots, a_n) \in X_k^n$ *and* $g, h \in \Omega_k$, *we define* $f_{\mathbf{a}}^{(g,h)} : X_k^n \to X_k$ *by*

$$f_{\mathbf{a}}^{(g,h)}(x_1, \ldots, x_n) = \begin{cases} g & \text{if } \mathbf{a} = (x_1, \ldots, x_n) \\ h & \text{otherwise.} \end{cases}$$

$f_{\mathbf{a}}^{(g,h)}(x_1, \ldots, x_n)$ *can be composed by* **Sel**, g *and* h.

Proof

We consider the following equations:

$$\begin{cases} o_i = f_{\{a_i\}}(o_{i-1}, h, x_i) & (1 \leq i \leq n) \\ o_0 = g. \end{cases}$$

We can easily verify that $o_1 = f_{\mathbf{a}}^{(g,h)}(x_1, \ldots, x_n)$. □

Proposition 5.11 $\min(x_1, \ldots, x_m)$ *and* $\max(x_1, \ldots, x_m)$ *can be composed by* **Sel**.

Proof

We define a logical function $f_i^g(x_1, \ldots, x_m)$ by

$$f_i^g(x_1, \ldots, x_m) = \begin{cases} i & \text{if } i \in \{x_1, \ldots, x_m\} \\ g & \text{otherwise.} \end{cases}$$

The following equations show that $f_i^g(x_1, \ldots, x_m)$ can be composed by **Sel**.

$$\begin{cases} o_i = \mathrm{sel}_{\{i\}}(x_i, o_{i+1}, x_i) & (1 \leq i \leq m) \\ o_{m+1} = g \end{cases}$$

$l = \min(x_1, \ldots, x_m)$ and $l' = \max(x_1, \ldots, x_m)$ show that

- $l \in \{x_1, \ldots, x_m\}$
- $l_0 < l \Rightarrow l_0 \notin \{x_1, \ldots, x_m\}$.
- $l' \in \{x_1, \ldots, x_m\}$
- $l < l_0 \Rightarrow l_0 \notin \{x_1, \ldots, x_m\}$.

The following equations show that $o_0 = \min(x_1, \ldots, x_m)$ and $o'_0 = \max(x_1, \ldots, x_m)$. Therefore the logical functions $\min(x_1, \ldots, x_m)$ and $o'_0 = \max(x_1, \ldots, x_m)$ can be composed by **Sel**.

$$\begin{cases} o_i & = & f_i^{o_{i+1}}(x_1, \ldots, x_m) \quad (0 \leq i \leq m) \\ o_{m+1} & = & x_1 \\ o'_i & = & f_{m-i}^{o'_{i+1}}(x_1, \ldots, x_m) \quad (0 \leq i \leq m) \\ o'_{m+1} & = & x_1. \end{cases}$$

□

Theorem 5.12

$$[\mathbf{Sel}] = \{f \in \Omega_k \mid f(x_1, \ldots, x_m) \in \{x_1, \ldots, x_m\}\}$$

Proof

For $f \in \{f \in \Omega_k \mid f(x_1, \ldots, x_m) \in \{x_1, \ldots, x_m\}\}$, we represent $f(x_1, \ldots, x_m)$ in Table 1. The following equations show that $o(x_1, \ldots, x_m) = f(x_1, \ldots, x_m)$.

$$\begin{cases} o_{\mathbf{a}} & = & f_{\mathbf{a}}^{(c_{\mathbf{a}}, \min(x_1, \ldots, x_m))}(x_1, \ldots, x_m) \quad \mathbf{a} \in X_k^m \\ o(x_1, \ldots, x_m) & = & \max\{o_{\mathbf{a}} \mid \mathbf{a} \in X_k^m\} \end{cases}$$

□

Theorem 5.13 *Let* $C = \{c_{i_1}, \ldots, c_{i_l}\} \subset \mathbf{Const}_k$.

$$[\mathbf{Sel} \cup C] = \{f \in \Omega_k \mid f(x_1, \ldots, x_m) \in \{c_{i_1}, \ldots, c_{i_l} x_1, \ldots, x_m\}\}.$$

Proof

x_1	x_2	$\cdots$	x_m	$f(x_1,\ldots,x_m)$
0	0	$\cdots$	0	$c_{(0,0,\ldots,0)} = 0$
0	0	$\cdots$	1	$c_{(0,0,\ldots,1)} \in \{0,1\}$
		$\vdots$		
0	0	$\cdots$	$k-1$	$c_{(0,0,\ldots,k-1)} \in \{0,k-1\}$
		$\vdots$		
0	$k-1$	$\cdots$	$k-1$	$c_{(0,k-1,\ldots,k-1)} \in \{0,k-1\}$
1	0	$\cdots$	0	$c_{(1,0,\ldots,0)} \in \{0,1\}$
1	0	$\cdots$	1	$c_{(1,0,\ldots,1)} \in \{0,1\}$
		$\vdots$		
$k-1$	0	$\cdots$	0	$c_{(k-1,0,\ldots,0)} \in \{0,k-1\}$
$k-1$	0	$\cdots$	1	$c_{(k-1,0,\ldots,1)} \in \{0,1,k-1\}$
		$\vdots$		
$k-1$	$k-1$	$\cdots$	$k-1$	$c_{(k-1,k-1,\ldots,k-1)} = k-1$

Table 1: Truth table of f

Now we use the same notation as in the proof of Theorem 5.12. For

$$f(x_1,\ldots,x_m) \in \{f \in \Omega_k \mid f(x_1,\ldots,x_m) \in \{c_{i_1},\ldots,c_{i_l}x_1,\ldots,x_m\}\},$$

the following equations show that $o(x_1,\ldots,x_m) = f(x_1,\ldots,x_m)$.

$$\begin{cases} o_{\mathbf{a}} & = & f_{\mathbf{a}}^{(c_{\mathbf{a}},\min(x_1,\ldots,x_m))}(x_1,\ldots,x_m) & f(\mathbf{a}) \in \{x_1,\ldots,x_m\}, \mathbf{a} \in X_k^m \\ o_{\mathbf{a}} & = & f_{\mathbf{a}}^{(c_{i_j},\min(c_{i_j},x_1,\ldots,x_m))}(x_1,\ldots,x_m) & f(\mathbf{a}) = i_j, \mathbf{a} \in X_k^m \\ o(x_1,\ldots,x_m) & = & \max\{o_{\mathbf{a}} \mid \mathbf{a} \in X_k^m\} & \end{cases}$$

□

Now we obtain the following examples.

Example 5.14 *1.* **Sel** *is incomplete.*

2. $\mathbf{Sel} \cup \{c_i \mid i \in X_k\}$ *is complete.*

5.3 Completeness Criteria

For the binary case, E. L. Post gave a powerful criterion as follows.

Theorem 5.15 ([21]) *Let F be a set of logical functions. F is functional complete iff it is* not *contained in any set of the following five sets :*

$$M_0, M_1, S, M^{\leq} \text{ and } L,$$

where

$$\begin{aligned} M_0 &= \{f \in \Omega_2 \mid f(0,\ldots,0) = 0\} \\ M_1 &= \{f \in \Omega_2 \mid f(1,\ldots,1) = 1\} \\ S &= \{f \in \Omega_2 \mid f(x_1,\ldots,x_n)' = f(x_1',\ldots,x_n')\} \\ M^{\leq} &= \textit{the set of monotone non-decreasing functions} \\ L &= \textit{the set of linear functions.} \end{aligned}$$

Definition 5.16 *Let ρ be an h-ary relation on X_k, i.e. a subset of the set X_k^h. We say that $f \in MAP(X_k^m, X_k)$ preservers ρ if*

$$(f(a_{11},\ldots,a_{1m}),\ldots,f(a_{h1},\ldots,a_{hm})) \in \rho$$

whenever $\forall(a_{1j},\ldots,a_{hj}) \in \rho$.

For every h-ary relation ρ on X_k, we denote by $\mathrm{Pol}\,\rho$ the set of all $f \in \Omega_k$ preserving ρ.

For the ternary case, Jablonskiĭ determined all maximal incomplete sets.

Theorem 5.17 ([8, 9, 23]) *All the maximal incomplete sets in Ω_3 are the following sets:*

(1) $D(i) = \mathrm{Pol}(\{i\})$, *where* $i \in \{0,1,2\}$,

(2) $D(i,j) = \mathrm{Pol}(\{i,j\})$, *where* $(i,j) = (0,1),(1,2),(2,0)$,

(3) $C(0) = \mathrm{Pol}\left(\left\{\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}2\\2\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}0\\2\end{pmatrix}, \begin{pmatrix}2\\0\end{pmatrix}\right\}\right)$,

(4) $C(1) = \mathrm{Pol}\left(\left\{\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}2\\2\end{pmatrix}, \begin{pmatrix}1\\2\end{pmatrix}, \begin{pmatrix}2\\1\end{pmatrix}, \begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}\right\}\right)$.

(5) $C(2) = \mathrm{Pol}\left(\left\{\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}2\\2\end{pmatrix}, \begin{pmatrix}2\\0\end{pmatrix}, \begin{pmatrix}0\\2\end{pmatrix}, \begin{pmatrix}2\\1\end{pmatrix}, \begin{pmatrix}1\\2\end{pmatrix}\right\}\right)$,

(6) $E(0) = \mathrm{Pol}\left(\left\{\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}2\\2\end{pmatrix}, \begin{pmatrix}1\\2\end{pmatrix}, \begin{pmatrix}2\\1\end{pmatrix}\right\}\right)$,

(7) $E(1) = \mathrm{Pol}\left(\left\{\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}2\\2\end{pmatrix}, \begin{pmatrix}2\\0\end{pmatrix}, \begin{pmatrix}0\\2\end{pmatrix}\right\}\right)$,

(8) $E(2) = \mathrm{Pol}\left(\left\{\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}2\\2\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}1\\0\end{pmatrix}\right\}\right)$,

(9) $O(0) = \mathrm{Pol}\left(\left\{\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}2\\2\end{pmatrix}, \begin{pmatrix}2\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}2\\1\end{pmatrix}\right\}\right)$,

(10) $O(1) = \mathrm{Pol}\left(\left\{\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}2\\2\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}1\\2\end{pmatrix}, \begin{pmatrix}0\\2\end{pmatrix}\right\}\right)$,

(11) $O(2) = \mathrm{Pol}\left(\left\{\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}2\\2\end{pmatrix}, \begin{pmatrix}1\\2\end{pmatrix}, \begin{pmatrix}2\\0\end{pmatrix}, \begin{pmatrix}1\\0\end{pmatrix}\right\}\right)$,

(12) $Perm = \mathrm{Pol}\left(\left\{\begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}1\\2\end{pmatrix}, \begin{pmatrix}2\\0\end{pmatrix}\right\}\right)$,

(13) $L = \mathrm{Pol}\left(\left\{\begin{pmatrix}a\\b\\c\end{pmatrix} \middle| c \equiv 2(a+b) \pmod 3\right\}\right)$,

(14) $S = \mathrm{Pol}\left(\left\{\left(\begin{array}{c} a \\ b \\ c \end{array}\right) \middle| \ |\{a,b,c\}| \leq 2\right\}\right)$.

For general case($k \geq 3$), G.Rosenberg gave a powerful criterion as follows.

Theorem 5.18 ([23]) *Each maximal incomplete set in Ω_k is described in the form* $\mathrm{Pol}\rho$ *where ρ is one of the following relations on X_k:*

(1) *Every partial order of X_k with least and greatest elements.*

(2) *Every relation $\{(x, sx) \mid x \in X_k\}$ where s is a permutation of X_k with $\frac{k}{p}$ cycles of the same prime length p.*

(3) *Every quaternary relation $\{(a_1, a_2, a_3, a_4) \in X_k^4 \mid a_1 +' a_2 = a_3 +' a_4$ where $< X_k, +' >$ is a p-elementary abelian group, where p is a prime.*

(4) *Every non-trivial equivalence relation on X_k.*

(5) *Every central relation on X_k.*

(6) *Every relation λ_T determined by an h-regular family T of equivalence relations on X_k $(h > 2)$.*

A subset F of Ω_k is complete iff for every relation described under (1)-(6) above there exists an $f \in F$ not preserving ρ.

Definition 5.19 *Suppose that $k = p^m$, where p is a prime. An Abelian group $G = (X_k, \oplus_G)$ is an elementary p-group iff $pG = \{0\}$.*

Definition 5.20 *Let $1 \leq h \leq k$. An h-ary relation ρ on X_k is totally reflexive iff ρ contains any $(a_1, \ldots, a_h)$ such that $a_1, \ldots, a_h \in X_k$ are not all distinct. ρ is totally symmetric iff $(a_1, \ldots, a_h) \in \rho$ implies $(a_{p(1)}, \ldots, a_{p(h)}) \in \rho$ for any permutation p of the set $\{1, 2, \ldots, h\}$. For a totally reflexive and totally symmetric relation ρ, the center of ρ is the set of all $c \in X_k$ such that $(c, a_2, \ldots, a_h) \in \rho$ for any $a_2, \ldots, a_h$. ρ is central iff $\rho \neq X^h$ and its center is non-empty.*

Definition 5.21 *Let $2 < h \leq k$ and let $m \leq 1$. The family $T = \{\theta_0, \ldots, \theta_{m-1}\}$ of equivalence relations on X_k is h-regular iff*

1. *each θ_j has h equivalence classes $(j = 0, \ldots, m-1)$.*
2. *the intersection $\cap_{j=0}^{m-1} \epsilon_j$ of arbitrary equivalence classes ϵ_j of θ_j $(j = 0, \ldots, m-1)$ is nonempty.*

The relation determined by T is the relation λ_T of all $a_1, \ldots, a_{h-1}$ having the property that for $0 \leq \forall j < m$ at least two elements among $a_0, \ldots, a_{h-1}$ are equivalent in θ_j.

5.4 Completeness under "loop-free"

Definition 5.22 *Let F be a set of logical functions. We denote by $\overline{F}$ the set of all functions realized by loop-free circuits over F.*

Definition 5.23 *Let F be a set of logical functions.*

(i) *F is said to be* LF-complete *iff $\overline{F} = \Omega$.*

(ii) *F is said to be* LF-incomplete *iff it is not complete.*

(iii) *F is said to be* maximal LF-incomplete *iff the following conditions are satisfied.*

- *F is LF-incomplete.*
- *$\forall F' : F' \supset F \Longrightarrow F' :$ LF-complete*

We can show the following theorem immediately.

Theorem 5.24 *Let F be a set of logical functions. F is said to be LF-complete iff F is complete.*

Proof

Let F and G be subsets of Ω_k. By Lemma 4.2, all $h \in F \circledcirc G$ are realized by loop-free circuits over $F \cup G$. Since each $f \in F$ is realized by a loop-free circuit over F. Each $f \in (F \cup \{I\})^{(n)}$ is realized by a loop-free circuit over F. Therefore any $f \in \hat{F}$ is realized by a loop-free circuit over F. It follows that functional completeness and LF-completeness are equivalent. □

6 Completeness under the General Initialization Assumption

Definition 6.1 *Let F be a set of logical functions. We denote by $[F]$ the set of functions, each of which can be realized by a circuit over F, with respect to some non-empty set Y of states.*

Definition 6.2 *Let F be a set of logical functions.*

(i) *F is said to be* GS-complete *iff* $[F] = \Omega$.

(ii) *F is said to be* GS-incomplete *iff it is not GS-complete.*

(iii) *F is said to be* maximal GS-incomplete *iff the following conditions are satisfied.*

- *F is GS-incomplete.*
- $\forall F' : F \supset F \Longrightarrow F'$ *: GS-complete*

Proposition 6.3 *Let F and F' be sets of logical functions.*

(i) $F \subseteq \overline{F} \subseteq [F]$

(ii) $F \subseteq F' \Longrightarrow [F] \subseteq [F']$

(iii) $F' \subseteq [F] \Longrightarrow [F \cup F'] = [F]$

(iv) $[[F]] = [F]$

Lemma 6.4 *Let f be a surjective logical function. We can realize all constant functions over $\{f\}$.*

Proof

Since f is a surjection, $\forall i \in X_k \exists a_{ij} \in X_k$ such that $f(a_{i1}, \ldots, a_{im}) = i$.

The constant function c_l is realized by the following circuit with respect to the states set $Y = \{(l, l \oplus 1, \ldots, l \oplus m-1)\}$:

$$\begin{cases} y'_0 & = & f(y_{a_{11} \oplus l}, \ldots, y_{a_{1m} \oplus l}) \\ y'_1 & = & f(y_{a_{21} \oplus l}, \ldots, y_{a_{2m} \oplus l}) \\ & \vdots & \\ y'_{k-1} & = & f(y_{a_{k1} \oplus l}, \ldots, y_{a_{km} \oplus l}), \end{cases}$$

where $\oplus$ denotes the addition in the ring Z/kZ. □

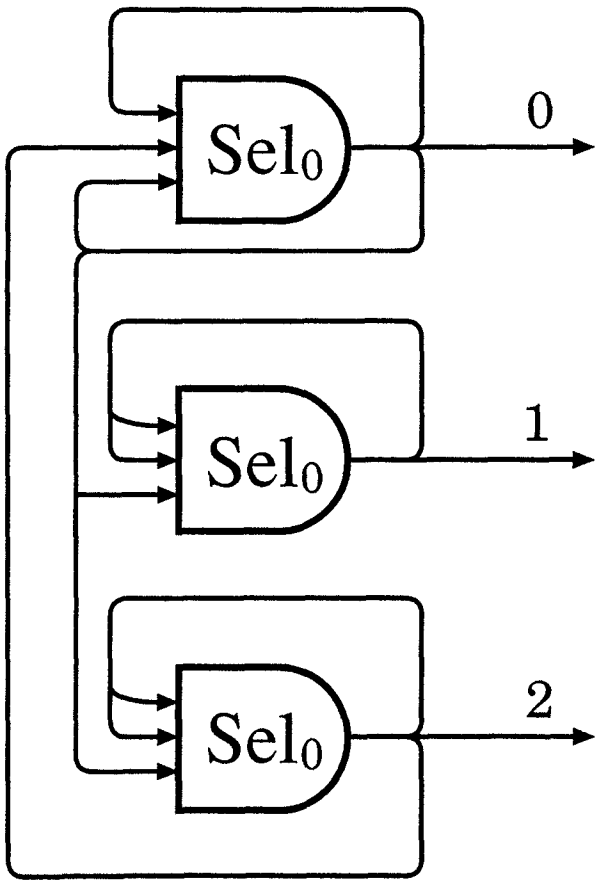

Figure 3: Realization of constant functions by sel

Example 6.5 *Since each function in* **Sel** *is a surjection, all constant functions are constructed by* **Sel**. *For instance, any constant function in* Ω_3 *is realized by the following circuit(Figure 3):*

$$\begin{cases} y_0' &= \mathrm{sel}_0(y_0, y_2, y_1) \\ y_1' &= \mathrm{sel}_0(y_1, y_1, y_0) \\ y_2' &= \mathrm{sel}_0(y_2, y_2, y_0). \end{cases}$$

By Example 5.14, **Sel** *is GS-complete.*

Lemma 6.6 *Let* f *be a non-constant logical function. We can realize a constant function over* $\{f\}$.

Proof

We consider the following function:$f^*(x) = f(x, \ldots, x)$.

(Case 1)

Suppose that f^* is surjective. There exists $m \in N$ such that $(f^*)^m$ is the identity map. Then the constant function c_l is realized by the following circuit with respect to the states set $Y = \{(l, \ldots, l)\}$:

$$\begin{cases} y_1' &= f^*(y_2) \\ y_2' &= f^*(y_3) \\ &\vdots \\ y_{m-1}' &= f^*(y_m) \\ y_m' &= f^*(y_1) \end{cases}$$

(Case 2)

Suppose that f^* is not surjective. We denote:

$$\begin{cases} A_0 &= X_k \\ A_n &= \mathrm{Im}(f^*)^n \quad n \geq 1 \end{cases}$$

Since X_k is a finite set, there exist $m_1, m_2 \in N$ such that $A_{m_1} = A_{m_2}$ and $m_1 > m_2$. Let $A = A_{m_1}$ and $m = m_1 - m_2$. The function $(f^*)^m \mid A$ is a surjection on A. By the same arguments of **Case1**, we can realize constant functions c_l $(l \in A)$. □

Theorem 6.7 *Let ρ be an h-ary relation on X_k. If $\{(x,\ldots,x) \mid x \in X_k\} \subset \rho$ then* $\mathrm{Pol}\,\rho$ *is GS-incomplete.*

Proof

We consider a logical function $f(x_1,\ldots,x_m)$ realized by a circuit $\mathcal{C}$ over $\mathrm{Pol}\,\rho$, with respect to a state set Y.

$$\begin{cases} \mathrm{y}(0) \xrightarrow{\mathrm{x}_1} \mathrm{y}^{(1)}(1) \xrightarrow{\mathrm{x}_1} \mathrm{y}^{(1)}(2) \xrightarrow{\mathrm{x}_1} \cdots \xrightarrow{\mathrm{x}_1} \mathrm{y}^{(1)}(t) \xrightarrow{\mathrm{x}_1} \cdots \\ \mathrm{y}(0) \xrightarrow{\mathrm{x}_2} \mathrm{y}^{(2)}(1) \xrightarrow{\mathrm{x}_2} \mathrm{y}^{(2)}(2) \xrightarrow{\mathrm{x}_2} \cdots \xrightarrow{\mathrm{x}_2} \mathrm{y}^{(2)}(t) \xrightarrow{\mathrm{x}_2} \cdots \\ \qquad\qquad\qquad\qquad\qquad\qquad \vdots \\ \mathrm{y}(0) \xrightarrow{\mathrm{x}_h} \mathrm{y}^{(h)}(1) \xrightarrow{\mathrm{x}_h} \mathrm{y}^{(h)}(2) \xrightarrow{\mathrm{x}_h} \cdots \xrightarrow{\mathrm{x}_h} \mathrm{y}^{(h)}(t) \xrightarrow{\mathrm{x}_h} \cdots \end{cases}$$

are admissible sequences of the circuit $\mathcal{C}$ over $\mathrm{Pol}\,\rho$ for inputs $\mathrm{x}_i = (x_{i1},\ldots,x_{im})$. We assume that these admissible sequences are synchronous.

Assume that $(x_{i1},\ldots,x_{ih}) \in \rho$ for all i. Since

$$\{(x,\ldots,x) \mid x \in X_k\} \subset \rho,$$

$(y_i,\ldots,y_i)$ is an element of ρ for all i. Suppose that $(y_i^{(1)}(t),\ldots,y_i^{(h)}(t)) \in \rho$. Since all component functions f_i' are in $\mathrm{Pol}\,\rho$ and $(x_{i1},\ldots,x_{ih}) \in \rho$,

$$(y_i^{(1)}(t+1),\ldots,y_i^{(h)}(t+1)) \in \rho.$$

By induction on t, $(y_i^{(1)}(t),\ldots,y_i^{(h)}(t)) \in \rho$ for $\forall t$. So $f(x_1,\ldots,x_m)$ preserves ρ. Therefore a logical function which can be realized by the circuit $\mathcal{C}$ over $\mathrm{Pol}\,\rho$ is in $\mathrm{Pol}\,\rho$. □

We can show the following proposition immediately.

Proposition 6.8 *Let ρ be a relation. $\{(x,\ldots,x) \mid x \in X_k\} \subset \rho$ iff* $\mathbf{Const}_k \subset \mathrm{Pol}\,\rho$.

Theorem 6.9 *Let F be a maximal incomplete set. F is GS-incomplete iff $F \supset \mathbf{Const}_k$.*

Proof

Obvious by Theorem 6.7 and Lemma 6.4. □

Example 6.10 *For the binary case, the following two sets are GS-incomplete.*

1. *The set L of linear functions*
2. *The set* $M^{\leq} = \mathrm{Pol}\left(\left\{\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}1\\1\end{pmatrix}\right\}\right)$ *of all monotone non-decreasing functions*

For the binary case, we obtain a powerful criterion.

Theorem 6.11 ([26]) *Let F be a set of logical functions in Ω_2. F is GS-complete iff it is not contained in neither the set of linear functions nor the set of monotone non-decreasing functions.*

Proof

Suppose that F is not contained in L. Then there exists $f \in \Omega_2 \setminus L$ such that f is a surjection. By Lemma 6.4, we can realize both $c_0(x)$ and $c_1(x)$.

$$c_0(x), c_1(x) \in [F].$$

On the other hand, the set $F \cup \mathbf{Const}_2$ is not contained in any of the following sets:

$$M_0, M_1, S, M^{\leq} \text{ and } L.$$

From Theorem 5.15 and Proposition 6.3,

$$\begin{aligned} [F] &= [F \cup \mathbf{Const}_2] \\ &\supseteq \overline{F \cup \mathbf{Const}_2} \\ &= \Omega_2 \end{aligned}$$

□

Example 6.12 *For the general case($k > 2$), the following sets are GS-incomplete.*

1. *Let ρ be a partial order on X_k. Since $x \leq x$, $\mathbf{Const}_k \subset \mathrm{Pol}\,\rho$.*

   $\mathrm{Pol}\,\rho$ *is GS-incomplete.*

2. *Let ρ be a quaternary relation $\{(x_1, \ldots, x_4) \mid x_1 \oplus_G x_2 = x_3 \oplus_G x_4\}$. Since $x \oplus_G x = x \oplus_G x$ for $\forall x \in X_k$, $\mathbf{Const}_k \subset \mathrm{Pol}\,\rho$.*

   $\mathrm{Pol}\,\rho$ *is GS-incomplete.*

3. *Let ρ be a equivalence relation. $(x, x) \in \rho$ for $\forall x \in X_k$.*

   $\mathrm{Pol}\,\rho$ *is GS-incomplete.*

4. *Let ρ be an h-ary relation on X_k. Suppose that $h \geq 2$. If ρ is a central relation, then $\{(x, \ldots, x) \in X_k^h\} \subset \mathrm{Pol}(\rho)$.*

   $\mathrm{Pol}\,\rho$ *is GS-incomplete.*

5. *Let ρ_T be a relation determined by T. For $\forall x \in X_k$, x and x are equivalent in some $\theta \in T$. Then $\mathbf{Const}_k \subset \mathrm{Pol}\,\rho$.*

   $\mathrm{Pol}\,\rho$ *is GS-incomplete.*

For the general case($k > 2$), we obtain a powerful criterion.

Theorem 6.13 ([27]) *Let F be a set of k-valued logical functions. F is GS-complete iff there exists an $f \in F$ not preserving ρ for every relation ρ described under $(1) - (5)$ below.*

(1) *Every partial order of X_k with least and greatest elements.*

(2) *Every quaternary relation $\{(a_1, a_2, a_3, a_4) \in X_k^4 \mid a_1 +' a_2 = a_3 +' a_4$ where $< X_k, +' >$ is a p-elementary abelian group,where p is a prime.*

(3) *Every non-trivial equivalence relation on X_k.*

(4) *Every non-unary central relation on X_k*

(5) *Every relation λ_T determined by an h-regular family T of equivalence relations on X_k.*

Proof

Let θ be the the trivial equivalence relation on X_k, i.e. $\theta = \{(x, x) \mid x \in X_k\}$. Hence the family $T = \{\theta\}$ is k-regular. The relation λ_T determined by T is

$$\left\{(a_1, \ldots, a_k) \in X_k^k | \sharp\{a_1, \ldots, a_k\} < k\right\}.$$

Since $F \not\subset \mathrm{Pol}\,\lambda_T$, there exists $f \in \Omega_k$ such that f is a surjection. By lemma 6.4, we can realize all constant functions.

$$\mathbf{Const}_k \subset [F].$$

On the other hand, there exists an $f \in F \cup \mathbf{Const}_k$ not reserving ρ for every relation ρ described in Theorem 5.18. Therefore

$$\begin{aligned} [F] &= [F \cup \mathbf{Const}_k] \\ &\supseteq \overline{F \cup \mathbf{Const}_k} \\ &= \Omega_k. \end{aligned}$$

□

Corollary 6.14 *Let F be a set of logical functions in Ω_3.*

F is GS-complete iff it is not contained in any set of the following eleven sets:

$$\begin{array}{ccc} C(0) & C(1) & C(2) \\ E(0) & E(1) & E(2) \\ O(0) & O(1) & O(2) \\ L & S. & \end{array}$$

7 Completeness for Restricted Sequential Circuit under the General Initialization Assumption

7.1 GR-completeness

Definition 7.1 *Let F be a set of logical functions. We denote by $\lfloor F \rfloor$ the set of functions, each of which can be realized by a restricted sequential circuit over F.*

Definition 7.2 *Let F be a set of logical functions.*

(i) *F is said to be* GR-complete *iff $\lfloor F \rfloor = \Omega$.*

(ii) *F is said to be* GR-incomplete *iff it is not GR-complete.*

(iii) *F is said to be* maximal GR-incomplete *iff the following conditions are satisfied.*

- *F is GR-incomplete.*
- *$\forall F' : F' \supset F \Longrightarrow F'$: GR-complete*

Proposition 7.3 *Let F and F' be sets of logical functions.*

(i) $F \subseteq \overline{F} \subseteq \lfloor F \rfloor \subseteq [F]$

(ii) $F \subseteq F' \Longrightarrow \lfloor F \rfloor \subseteq \lfloor F' \rfloor$

(iii) $F' \subseteq \lfloor F \rfloor \Longrightarrow \overline{F \cup F'} \subseteq \lfloor F \rfloor$

(iv) $\lfloor F \rfloor \subseteq \lfloor \lfloor F \rfloor \rfloor$

(v) $\lfloor F \rfloor = \overline{[F]}$

Example 7.4 *Both of the constant functions c_0 and c_1 are realized by circuits over* {NOT}. *On the other hand, both of the constant functions are not realized by restricted circuits over* {NOT}.

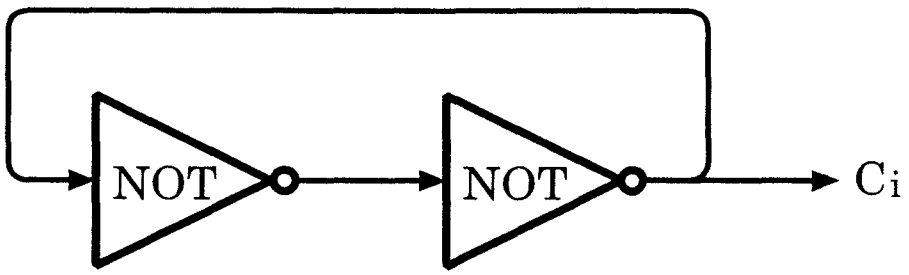

Figure 4: Realization of constant function by NOT

Proposition 7.5 *Let f be a k-valued logical function. Define $f^*(x)$ to be $f(x,\ldots,x)$. If there exists $x_0 \in X_k$ such that $f^*(x_0) = x_0$ then we can realize the constant function c_{x_0}.*

Proof

The constant function c_{x_0} is realized by the following circuit with respect to the state set $Y = \{(x_0)\}$(Figure 5).

$$\left\{ y_1' = f(y_1,\ldots,y_1) \right.$$

□

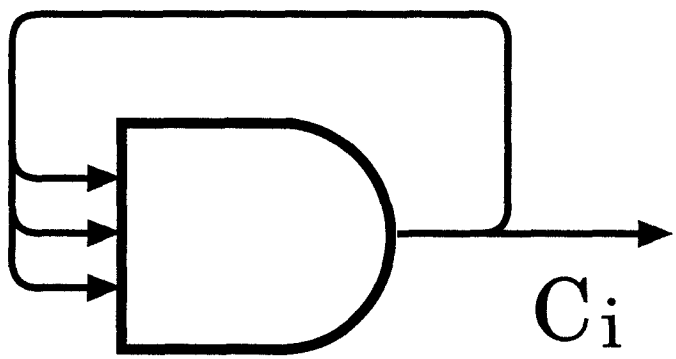

Figure 5: Realization of a constant function by f^*

Proposition 7.6 *Let F be a subset of Ω_k. If F is GS-incomplete then F is GR-incomplete.*

Proof

Obvious, by Proposition 7.3 and Theorem 6.9. □

Example 7.7 *In the binary case, the sets $M^{\leq}$ and L are GR-incomplete, and the set S is GR-complete.*

Theorem 7.8 *For every relation ρ described defined in the following* (1) *and* (2)*, $\mathrm{Pol}(\rho)$ is GR-complete.*

(1) *Every relation $\{(x, sx) \mid x \in X_k\}$ where s is a permutation of X_k with $\frac{k}{p}$ cycles of the same prime length p.*

(2) *Every unary central relation on X_k*

Proof

Let ρ be a relation described under $(1) - (2)$ above. Since the identity map $I(x) \in \mathrm{Pol}(\rho)$, we can realize all constant functions by restricted circuits over $\mathrm{Pol}(\rho)$. Then $\mathbf{Const}_k \subset \lfloor \mathrm{Pol}(\rho) \rfloor$. There exists a constant function c_i such that $c_i \notin \mathrm{Pol}(\rho)$. From Theorem 5.18, $\mathrm{Pol}(\rho)$ is a maximal incomplete set. Therefore, $\mathrm{Pol}(\rho)$ is GR-complete. □

Theorem 7.9 *Let F be a subset of Ω_k. Suppose that $\mathbf{Const}_k \subset F$. F is GR-complete iff F is complete.*

Proof

Suppose that F is incomplete. Then $F \subset \mathrm{Pol}\,\rho$ for some relation ρ described in Theorem 5.18. Since $\mathbf{Const}_k \subset F$ and arguments in Example 6.12,F is GR-incomplete. □

7.2 GR-completeness criterion for the binary case

We define $W = \{f \in \Omega_2 \mid f(0, \ldots, 0) = 1, f(1, \ldots, 1) = 0\}$. By H we denote the set $S \cap W$.

Proposition 7.10 *The set $S \cap W$ is GR-incomplete.*

Proof

We consider a one-variable function $h(x)$ realized by a restricted sequential circuit $\mathcal{C}$ over H, with respect to a state set Y.

We will show that $h(x)$ is either $I(x)$ or NOT(x), and cannot be $C_0(x)$ or $C_1(x)$. Assume that the circuit $\mathcal{C}$ is the "minimum" one, having the smallest number of elements among the circuits which realize $h(x)$.

$$(\mathcal{C}) \begin{cases} y_1' &= f_1(y_1, y_2, \cdots, y_{m-1}, y_m, x) \\ y_2' &= f_1(y_2, y_3, \cdots, y_{m-1}, y_m, x) \\ & \vdots \\ y_m' &= f_m(y_m, x). \end{cases}$$

(Step1) We shall show that the value of the function $f_m(y_m, x)$ of the m-th element is equal to NOT(x). Two cases are possible.

Case 1 $f_m(0,0) = f_m(1,0)$

Since f_m is self-dual, we have:

$$f_m(1,1) = f_m(0,1).$$

Therefore, the first variable of f_m is dummy, and we have:

$$f_m(y, x) = \text{NOT}(x).$$

Case 2 $f_m(0,0) = 1, f_m(1,0) = 0$

Then we have:

$$f_m(1,1) = 0, f_m(0,1) = 1.$$

Obviously, the output f_m is "unstable" for any input 0 or 1. This is a contradiction.

(Step 2) Let us examine the function f_{m-1}. Since f_m is identical with NOT, three cases are possible.

Case 1 $f_{m-1}(0,1,0) = f_{m-1}(1,1,0) = a$

Then we have:

$$f_{m-1}(1,0,1) = f_{m-1}(0,0,1) = \overline{a}.$$

Hence the output of y_{m-1} will eventually become a for input 0,and $\overline{a}$ for input 1.

- $a = 0$

  The stable output of f_{m-1} is identical with $I(x)$.

- $a = 1$

  The stable output of f_{m-1} is identical with NOT(x). Since NOT(x) is given by y_m, the $(m-1)$-th element is redundant, against the assumption.

Case 2 $f_{m-1}(0,1,0) = 1, f_{m-1}(1,1,0) = 0$

Then we have:

$$f_{m-1}(1,0,1) = 0, f_{m-1}(0,0,1) = 1.$$

In this case, the output of f_{m-1} is "unstable" for any input. This is a contradiction.

Case 3 $f_{m-1}(0,1,0) = 0, f_{m-1}(1,1,0) = 1$

We can show the following fact.

We can assume without loss of generality that y_{m-1} is identical with NOT(x).

So $(m-1)$-th element is redundant, against the assumption.

Up to now, we have shown the following facts.

(i) $f_m(y,x) = \text{NOT}(x)$

(ii) $f_{m-1}(y, \text{NOT}(x), x) = I(x)$

(Step 3) If m is greater than two, then we can replace the variable y_{m-1} by x. This is against the minimality assumption of the circuit $\mathcal{C}$. Thus the number of elements is not greater than two. Therefore,

$$h(x) = \begin{cases} \text{NOT}(x) & \text{if } m = 1 \\ I(x) & \text{if } m = 2 \end{cases}$$

This completes the proof of the proposition.

□

Remark 7.11 *By Theorem 6.11, the set H is GS-complete.*

Lemma 7.12 *Suppose that F is not contained in L and H. Then we can realize both of the constant functions C_0 and C_1, by the circuits over F.*

Proof

(Step1) Let f be a function in $\Omega_2 \setminus H$. Suppose that f is a non-constant function. Then f is a surjection. We define $f'(x) = f(x, x, \cdots, x)$. If $f'(x)$ is the constant function $C_a(x)$ then, we can realize the constant function $C_a(x)$. So we can assume that f' is a non-constant function.

Case 1 $f'(x) = I(x)$

By the same arguments as in the proof of Lemma 6.4, we can realize the constant functions C_0 and C_1.

Case 2 $f'(x) = \mathrm{NOT}(x)$

Then we have:

$$f(0, 0, \cdots, 0) = 1, f(1, 1, \cdots, 1) = 0.$$

Since $f \notin H$, f is not in S. Then there exists a binary sequence

$$\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_n)$$

such that

$$f(\alpha_1, \alpha_2, \cdots, \alpha_n) = f(\overline{\alpha_1}, \overline{\alpha_2}, \cdots, \overline{\alpha_n}) = a.$$

We denote:

$$f''(x_1, x_2) = f(x_{2-\alpha_1}, x_{2-\alpha_2}, \cdots, x_{2-\alpha_n}).$$

Then the constant function $C_a(x)$ is realized by the following circuit $\mathcal{C}'$ with respect to the state set

$$Y = \{(a, 1), (a, 0)\}$$

(see Figure 6) :

$$(\mathcal{C}') \begin{cases} y_1' &= f''(y_2, x) \\ y_2' &= f'(x). \end{cases}$$

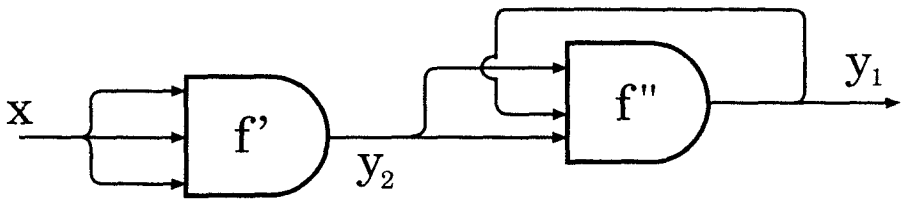

Figure 6: The circuit C'

(**Step 2**) Let g be a function in $F \setminus L$. Then g is a surjection. We denote:

$$g^*(x) = g(x, x, \cdots, x).$$

Case 1 $g^*(x) = I(x)$

By the same arguments as in the proof of Lemma 6.4, we can realize the constant functions $C_0(x)$ and $C_1(x)$.

Case 2 $g^*(x) = \mathrm{NOT}(x)$

In Step 1, we have shown that a constant function $C_a(x)$ is realized by the circuit over F. Therefore we can realize the another constant function $C_{\overline{a}}(x)$(see Figure 7).

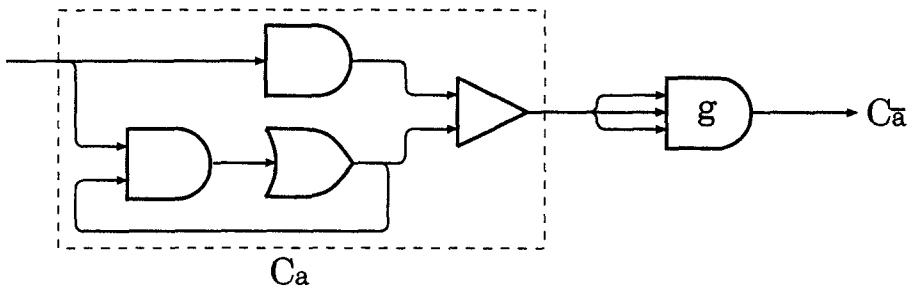

Figure 7: Realization of $C_{\overline{a}}$ by g and C_a

Case 3 $g^*(x) = C_0(x)$ or $C_1(x)$

By the same arguments as in the proof of Lemma 6.4, we can realize the constant functions $C_0(x)$ and $C_1(x)$.

In this way, we can realize both of the constant functions $C_0(x)$ and $C_1(x)$. □

Theorem 7.13 *Let F be a set of logical functions. F is GR-complete iff it is not contained in any three sets:*

$$M^{\leq}, L \text{ and } H.$$

Proof

(only-if part) Obvious by Proposition 7.7 and Proposition 7.10.

(if part) Suppose that F is not contained in L and H. By Lemma 7.12, we can realize both of $C_0(x)$ and $C_1(x)$.

$$C_0(x), C_1(x) \in \lfloor F \rfloor$$

On the other hand, the set $F \cup \{C_0(x), C_1(x)\}$ is not contained in any five sets:

$$M_0, M_1, S, M^{\leq} \text{ and } L.$$

Therefore,

$$\begin{aligned} \lfloor F \rfloor &\supseteq \overline{F \cup \{C_0(x), C_1(x)\}} && \text{(by Proposition 7.3)} \\ &= \Omega_2. && \text{(by Theorem 5.24)} \end{aligned}$$

□

8 Completeness under the Initialization-by-Input Assumption

8.1 NS-completeness

Definition 8.1 *A circuit C is said to be* terminating *if and only if every admissible sequence of the circuit C for every input, starting from any state is always terminating, and its stable state is uniquely determined by the initial state and the input.*

Definition 8.2 *Let C be a terminating circuit. We denote by*

$$C(\mathbf{y}, \mathbf{a}(1), \mathbf{a}(2), \mathbf{a}(3), \cdots, \mathbf{a}(t))$$

the final output of the circuit for the input sequence $\mathbf{a}(1), \mathbf{a}(2), \mathbf{a}(3), \cdots, \mathbf{a}(t)$, which is given to the circuit in the following way.

(i) *The first input $\mathbf{a}(1)$ is given to the circuit which is in the state $\mathbf{y}$.*

(ii) *The i-th input $\mathbf{a}(i)$ is feed to the circuit after the circuit reaches the stable state for the input $\mathbf{a}(i-1)$.*

The final output means the first component of the stable state for the final input $\mathbf{a}(t)$.

Definition 8.3 *The circuit C* evaluates *a logical function h by an* initial sequence $\mathbf{x}(-d), \mathbf{x}(-d+1), \cdots, \mathbf{x}(-1)$ *if and only if the following conditions are satisfied.*

(i) *The circuit C is terminating.*

(ii) *For any input sequence $\mathbf{x}(0), \mathbf{x}(1), \cdots, \mathbf{x}(t)$ and any state $\mathbf{y}$ of the circuit C,*

$$h(\mathbf{x}(t)) =$$
$$C(\mathbf{y}, \mathbf{x}(-d), \mathbf{x}(-d+1), \cdots, \mathbf{x}(-1), \mathbf{x}(0), \mathbf{x}(1), \cdots, \mathbf{x}(t)).$$

Lemma 8.4 *Let C be a circuit. If the circuit C evaluates a logical function h by an initial sequence*

$$\mathrm{x}(-d), \mathrm{x}(-d+1), \cdots, \mathrm{x}(-1),$$

then

$$h(\mathrm{x}(t)) = C(\mathrm{y}, \mathrm{x}(-T), \mathrm{x}(-T+1), \cdots, \mathrm{x}(-d-1), \mathrm{x}(-d), \mathrm{x}(-d+1), \cdots, \mathrm{x}(-1), \mathrm{x}(0), \mathrm{x}(1), \cdots, \mathrm{x}(t))$$

for any state y *and any input sequence* $\mathrm{x}(-T), \mathrm{x}(-T+1), \cdots, \mathrm{x}(-d-1)$ *and* $\mathrm{x}(0), \mathrm{x}(1), \cdots, \mathrm{x}(t)$.

Proof

Obvious by Definition 8.3. □

Definition 8.5 *Let F be a set of logical functions. We denote by $\langle F \rangle$ the set of all logical functions, each of which is evaluated by a circuit C over F by some initial sequence.*

Definition 8.6 *Let F be a set of logical functions.*

(i) *F is said to be* NS-complete *if and only if $\langle F \rangle = \Omega$.*

(ii) *F is said to be* NS-incomplete *if and only if it is not NS-complete.*

(iii) *F is said to be* maximal NS-incomplete *if and only if the following conditions are satisfied.*

- *F is NS-incomplete.*
- *$\forall F' : F' \supset F \Longrightarrow F'$: NS-complete*

Proposition 8.7 *Let F and F' be sets of logical functions.*

(i) $F \subseteq \overline{F} \subseteq \langle F \rangle \subseteq [F]$

(ii) $F' \subseteq F \Longrightarrow \langle F' \rangle \subseteq \langle F \rangle$

We can show the following lemma immediately.

Lemma 8.8 *Let F be a subset of Ω_k. Suppose that* $\mathbf{Const}_k \subset \langle F \rangle$. *Then*

$$\overline{F \cup \mathbf{Const}_k} \subset \langle F \rangle.$$

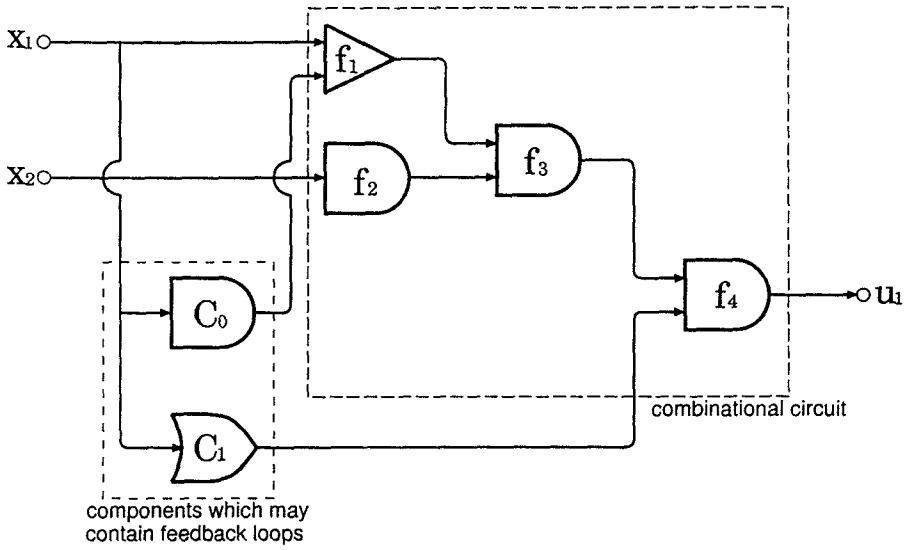

Figure 8: A circuit easily initializable

8.2 Examples of NS-complete and NS-incomplete sets

Example 8.9 *Let $f \in$* **Sel**. *We consider the following circuit:*

$$(C)\,\{y' = f_i(y, x, y).$$

A constant function c_i is evaluated by (C) by initial sequence (i). So **Sel** *is NS-complete. Since* **Sel** $\subset \mathrm{Pol}(\rho)$ *for every unary central relation ρ,* $\mathrm{Pol}(\rho)$ *is NS-complete.*

Example 8.10 *For the binary case, the sets $M^{\leq}$ and L are NS-incomplete because of Proposition 8.7.*

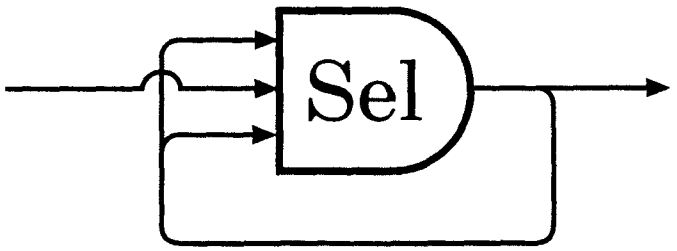

Figure 9: Realization of constant function by sel

Example 8.11 *Let ρ be a unary central relation. Since* **Sel** $\subset$ Pol(ρ), Pol(ρ) *is NS-complete.*

Theorem 8.12 Pol(ρ) *for every relation ρ described under* (1) – (5) *is NS-incomplete.*

(1) *Every partial order of X_k with a least and greatest elements.*

(2) *Every quaternary relation $\{(a_1, a_2, a_3, a_4) \in X_k^4 \mid a_1 +' a_2 = a_3 +' a_4$ where $< X_k, +' >$ is a p-elementary abelian group (p prime).*

(3) *Every non-trivial equivalence relation on X_k.*

(4) *Every non-unary central relation on X_k*

(5) *Every relation λ_T determined by an h-regular family T of equivalence relations on X_k.*

(6) *Every relation $\{(x, sx) \mid x \in X_k\}$ where s is a permutation of X_k with $\frac{k}{p}$ cycles of the same prime length p.*

Proof

By Theorem 6.13, Pol(ρ) for ρ described above (1) – (5) is GS-complete. Since Proposition 8.7, Pol(ρ) is NS-complete.

Now we prove that Pol(ρ) for ρ described above (6) is NS-incomplete.

Let ρ_σ be a relation determined by a permutation σ with $\frac{k}{p}$ cycles of the same prime length p. Let C be a circuit over Pol(ρ_σ).

For $(x_1,\ldots,x_m) \in X_k^m$ and a permutation σ, we define

$$(x_1,\ldots,x_m)^\sigma = (\sigma(x_1),\ldots,\sigma(x_m)).$$

The logical function h is evaluated by the circuit C with initial sequences $\mathbf{x}(-d).\ldots.\mathbf{x}(-1)$.

$$\mathbf{y}_1(0) \xrightarrow{\mathbf{x}} \cdots \xrightarrow{\mathbf{x}} \mathbf{y}_1(t) \xrightarrow{\mathbf{x}} \cdots$$

is a synchronous sequence for the input $\mathbf{x}$.

We consider the synchronous sequence starting $\mathbf{y}_2(0) = \mathbf{y}_1(0)^\sigma$ for the input $\mathbf{x}^\sigma$.

$$\mathbf{y}_2(0) = \mathbf{y}_1(0)^\sigma \xrightarrow{\mathbf{x}^\sigma} \cdots \xrightarrow{\mathbf{x}^\sigma} \mathbf{y}_2(t) \xrightarrow{\mathbf{x}^\sigma} \cdots$$

Since the circuit C is over $\mathrm{Pol}(\rho_\sigma)$, we can show that $\mathbf{y}_2(t) = \mathbf{y}_1^\sigma(t)$ for $\forall t$. Let m be an order of σ. From Lemma 8.4, we obtain

$$\begin{aligned} & C(\mathbf{y}^\sigma, \mathbf{x}(-d)^\sigma, \ldots, \mathbf{x}(-1)^\sigma, \mathbf{x}(-d), \ldots, \mathbf{x}(-1), \mathbf{x}(1)^\sigma, \ldots, \mathbf{x}(t)^\sigma) \\ = \;& C(\mathbf{y}, \mathbf{x}(-d), \ldots, \mathbf{x}(-1), \mathbf{x}(-d)^{\sigma^{m-1}}, \ldots, \mathbf{x}(-1)^{\sigma^{m-1}}, \mathbf{x}(1), \ldots, \mathbf{x}(t))^\sigma, \end{aligned}$$

i.e.

$$\begin{pmatrix} h(\mathbf{x}(t)) \\ h(\mathbf{x}(t)^\sigma) \end{pmatrix}$$

$$= \begin{pmatrix} C(\mathbf{y}, \mathbf{x}(-d), \ldots, \mathbf{x}(-1), \mathbf{x}(-d)^{\sigma^{m-1}}, \ldots, \mathbf{x}(-1)^{\sigma^{m-1}}, \mathbf{x}(1), \ldots, \mathbf{x}(t)) \\ C(\mathbf{y}^\sigma, \mathbf{x}(-d)^\sigma, \ldots, \mathbf{x}(-1)^\sigma, \mathbf{x}(-d), \ldots, \mathbf{x}(-1), \mathbf{x}(1)^\sigma, \ldots, \mathbf{x}(t)^\sigma) \end{pmatrix} \in \rho_\sigma.$$

Therefor $\mathrm{Pol}(\rho_\sigma)$ is NS-incomplete. □

Theorem 8.13 *Let A be a proper subset of X_k. We define*

$$\rho_A = \{(x,x') \in X_k^2 \mid x \neq x'\} \cup \{(x,x) \mid x \in A\}.$$

$\mathrm{Pol}(\rho_A)$ *is NS-incomplete.*

Proof

Let C be a circuit over $\mathrm{Pol}(\rho_A)$. Let σ be a random permutation such that $\sigma(\sigma(x)) = x \mathrm{for} \forall x in X_k$.

$$\mathbf{y}_1(0) \xrightarrow{\mathbf{x}} \cdots \xrightarrow{\mathbf{x}} \mathbf{y}_1(t) \xrightarrow{\mathbf{x}} \cdots$$

is a sequence for the input $\mathbf{x}$. We consider the sequence starting $\mathbf{y}_2(0) = \mathbf{y}_1(0)^\sigma$ for the input $\mathbf{x}^\sigma$.

$$\mathbf{y}_2(0) = \mathbf{y}_1(0)^\sigma \xrightarrow{\mathbf{x}^\sigma} \cdots \xrightarrow{\mathbf{x}^\sigma} \mathbf{y}_2(t) \xrightarrow{\mathbf{x}^\sigma} \cdots$$

Since the circuit C is over $\mathrm{Pol}(\rho_{\sigma,A})$,

$$\begin{pmatrix} \mathbf{y}_1(i) \\ \mathbf{y}_2(i) \end{pmatrix} \in \mathrm{Pol}(\rho_{\sigma,A}),$$

where $\mathbf{y}_i = (\mathbf{y}_i(1), \ldots, \mathbf{y}_i(m))$. Therefore, we can show that

$$\begin{pmatrix} C(\mathbf{y}, \mathbf{a}(1), \ldots, \mathbf{a}(t)) \\ C(\mathbf{y}^\sigma, \mathbf{a}(1)^\sigma, \ldots, \mathbf{a}(t)^\sigma) \end{pmatrix} \in \rho_A.$$

We assume that the circuit C evaluates the constant function c_v such that $v \notin A$ by initial sequence $x(-d), \ldots, x(-1)$. Then

$$\begin{aligned} C(\mathbf{y}, x(-d), \ldots, x(-1), \sigma(x(-d)), \ldots, \sigma(x(-1)), x(1), \ldots, x(t)) &= v \\ C(\mathbf{y}^\sigma, \sigma(x(-d)), \ldots, \sigma(x(-1)), x(-d), \ldots, x(-1), \sigma(x(1)), \ldots, \sigma(x(t))) &= v. \end{aligned}$$

Therefore

$$\begin{pmatrix} C(\mathbf{y}, x(-d), \ldots, x(-1), \sigma(x(-d)), \ldots, \sigma(x(-1)), x(1), \ldots, x(t)) \\ C(\mathbf{y}^\sigma, \sigma(x(-d)), \ldots, \sigma(x(-1)), x(-d), \ldots, x(-1), \sigma(x(1)), \ldots, \sigma(x(t))) \end{pmatrix} \in \mathrm{Pol}(\rho_A),$$

i.e.

$$\begin{pmatrix} v \\ v \end{pmatrix} \in \mathrm{Pol}(\rho_A).$$

This is a contradiction to the definition of ρ_A. □

Remark 8.14 *Let A be a proper subset of X_k. We define $\tau_A = \{(x) \mid x \in A\}$. Then $\mathrm{Pol}(\sigma_A) \subset \mathrm{Pol}(\tau_A)$ and $\mathrm{Pol}(\sigma_A) \neq \mathrm{Pol}(\tau_A)$. The unary relation τ_A is a central relation. So $\mathrm{Pol}(\tau_A)$ is NS-complete but $\mathrm{Pol}(\sigma_A)$ is NS-incomplete.*

8.3 NS-completeness criterion for the binary case

We define

$$N_0 = \mathrm{Pol}\left(\left\{\begin{pmatrix}0\\0\end{pmatrix},\begin{pmatrix}0\\1\end{pmatrix},\begin{pmatrix}1\\0\end{pmatrix}\right\}\right)$$

$$N_1 = \mathrm{Pol}\left(\left\{\begin{pmatrix}1\\1\end{pmatrix},\begin{pmatrix}0\\1\end{pmatrix},\begin{pmatrix}1\\0\end{pmatrix}\right\}\right).$$

The sets N_0 and N_1 are NS-incomplete by Theorem 8.13. The set S is NS-incomplete by Theorem 8.12.

In the following,we denote by F a set of logical functions.

Proposition 8.15 *If $F \not\subset S$ then $c_0 \in \langle F\rangle$ or $c_1 \in \langle F\rangle$.*

Proof

Let $f \in F \setminus S$. Then there exists $(a_1,\ldots,a_n) \in X_2^m$ such that

$$f(a_1,\ldots,a_n) = f(a_1',\ldots,a_n').$$

We define $g(x_0,x_1) = f(x_{a_1},\ldots,x_{a_n})$. Then $g(0,1) = g(1,0)$. Three cases are possible.

Case 1$(g(0,0) = g(1,1) = a)$

In this case,$g'(x) = g(x,x)$ is a constant function c_a. So $c_a \in \langle F\rangle$.

Case 2$(g(0,1) = g(0,0) \neq g(1,1))$

In this case, $g(x,y)$ is either $\mathrm{AND}(x,y)$ or $\mathrm{NAND}(x,y)$.

If $g(x,y) = \mathrm{AND}(x,y)$ then the constant function c_0 is evaluated by the following circuit $\mathcal{C}_0$ by initial sequence (0).

$$(\mathcal{C}_0)\,\{y' = \mathrm{AND}(x,y)\,.$$

If $g(x,y) = \mathrm{NAND}(x,y)$ then the constant function c_0 is evaluated by the following circuit $\mathcal{C}_1$.

$$(\mathcal{C}_1)\begin{cases} y_1' &= \mathrm{NAND}(y_2,y_2)\\ y_2' &= \mathrm{NAND}(x,y_3)\\ y_3' &= \mathrm{NAND}(x,x)\end{cases}.$$

So $c_0 \in \langle F \rangle$.

Case 3$(g(0,1) = g(1,1) \neq g(0,0))$

In this case, $g(x,y)$ is either $\mathrm{OR}(x,y)$ or $\mathrm{NOR}(x,y)$.

If $g(x,y) = \mathrm{OR}(x,y)$ then the constant function c_1 is evaluated by the following circuit $\mathcal{C}_2$ by initial sequence (1).

$$(\mathcal{C}_2)\ \{y' = \mathrm{OR}(x,y)\ .$$

If $g(x,y) = \mathrm{NOR}(x,y)$ then the constant function c_1 is evaluated by the following circuit $\mathcal{C}_3$.

$$(\mathcal{C}_3)\begin{cases} y_1' &= \mathrm{NOR}(y_2,y_2) \\ y_2' &= \mathrm{NOR}(x,y_3) \\ y_3' &= \mathrm{NOR}(x,x) \end{cases}.$$

So $c_1 \in \langle F \rangle$. □

Remark 8.16 *The circuits $\mathcal{C}_i$ $(i = 1,2,3)$ in proof of Proposition 8.15 are restricted sequential circuits.*

Proposition 8.17 *Suppose that $F \not\subset L$. A two-variable non-linear function f belongs in $\overline{F \cup \{c_a\}}$ for any constant a.*

Proof

Case 1 $(a = 0)$

Let f be an n-variable non-linear function in F. If $n = 2$ then the f is two-variable non-linear function. So we suppose that $n \geq 3$. By Galois's expansion theorem(See [25]),

$$\begin{aligned} & f(x_1,\ldots,x_n) \\ &= a_0 \oplus (a_1 \cdot x_1 \oplus \cdots \oplus a_n \cdot x_n) \\ &\quad \oplus (a_{12} \cdot x_1 \cdot x_2 \oplus a_{13} \cdot x_1 \cdot x_3 \oplus \cdots a_{n-1,n} \cdot x_{n-1} \cdot x_n) \\ &\quad \oplus \cdots \oplus a_{12\ldots n} \cdot x_1 \cdot \cdots \cdot x_n, \end{aligned}$$

where $\oplus$ and $\cdot$ denote the addition in modulo 2 and logical AND, respectively.

We chose the minimum degree term $x_{i_1} \cdots x_{i_p}$ whose degree is greater than two. We define:

$$\begin{aligned} \omega_i &= \begin{cases} x & \text{if } i = i_1 \\ y & \text{if } i = i_j (2 \leq j \leq p) \\ 0 & \text{otherwise,} \end{cases} \\ g(x,y) &= f(\omega_1, \ldots, \omega_n). \end{aligned}$$

Then the logical function g is two-variable non-linear. Furthermore $g(x,y)$ belongs to $\overline{F \cup \{a\}}$.

Case 2 $(a = 1)$

After replacing x_i by $X_i \oplus 1$, apply the same arguments in **Case 1** to $f'(X_1, \ldots, X_n) = f(X_1 \oplus 1, \ldots, X_n \oplus 1)$. Then we can show that there exists a two-variable non-linear function which belongs to $\overline{F \cup \{1\}}$. □

Proposition 8.18 *If $F \not\subset N_0$ then either c_1 or $g(x,y) = x \oplus y$ is contained in $\langle F \cup \{c_0\} \rangle$.*

Proof

Let $f \in \Omega_2 \setminus F$. Then there exists

$$\begin{pmatrix} u_i \\ v_i \end{pmatrix} \in \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$$

such that

$$\begin{pmatrix} f(u_1, \ldots, u_n) \\ f(v_1, \ldots, v_n) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

We define:

$$\begin{aligned} \omega_i &= \begin{cases} x & \text{if } u_i = 1, v_i = 0 \\ y & \text{if } u_i = 0, v_i = 1 \\ 0 & \text{if } u_i = v_i = 0, \end{cases} \\ g(x,y) &= f(\omega_1, \ldots, \omega_n). \end{aligned}$$

Thereby $g(x,y)$ belongs to $\overline{F \cup \{c_0\}}$ and $g(0,1) = g(1,0) = 1$.

If $g(0,0) = 1$ then the circuit $\mathcal{C}_1$ evaluates the constant function c_1 by initial sequence (0). If $g(1,1) = 1$ then the circuit $\mathcal{C}_1$ evaluates the constant function c_1 by initial sequence (1).

$$(\mathcal{C}_1)\left\{ \; y' \;=\; g(x,y) \right.$$

If $g(0,0) = g(1,1) = 0$ then $g(x,y) = x \oplus y$. □

Proposition 8.19 *If $F \not\subset N_1$ then either c_0 or $g(x,y) = x \oplus y \oplus 1$ is contained in $\langle F \cup \{c_1\}\rangle$.*

Proof

Let $f \in \Omega_2 \setminus F$. Then there exists

$$\begin{pmatrix} u_i \\ v_i \end{pmatrix} \in \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$$

such that

$$\begin{pmatrix} f(u_1,\ldots,u_n) \\ f(v_1,\ldots,v_n) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

We define:

$$\begin{aligned} \omega_i &= \begin{cases} x & \text{if } u_i = 1, v_i = 0 \\ y & \text{if } u_i = 0, v_i = 1 \\ 1 & \text{if } u_i = v_i = 1, \end{cases} \\ g(x,y) &= f(\omega_1,\ldots,\omega_n). \end{aligned}$$

Thereby $g(x,y)$ belongs to $\overline{F \cup \{c_1\}}$ and $g(0,1) = g(1,0) = 0$.

If $g(0,0) = 0$ then the circuit $\mathcal{C}_1$ evaluates the constant function c_1 by initial sequence (0). If $g(1,1) = 1$ then the circuit $\mathcal{C}_1$ evaluates the constant function c_1 by initial sequence (1).

$$(\mathcal{C}_1)\left\{ \; y' \;=\; g(x,y) \right.$$

If $g(0,0) = g(1,1) = 1$ then $g(x,y) = x \oplus y \oplus 1$. □

We define $f_{a,b,c}(x,y) = x \cdot y \oplus a \cdot x \oplus b \cdot y \oplus c$ and $g(x,y) = x \oplus y$.

Remark 8.20 *The circuits $\mathcal{C}_i$ in proof of Proposition 8.18 and Proposition 8.19 are restricted sequential circuits.*

Proposition 8.21 *Suppose that $f, g \in \langle F \cup \{c_0\}\rangle$. Then $c_1 \in \langle F \cup \{c_0\}\rangle$.*

Proof

Case 1 $(a = 0, b = 0)$

In this case, the circuit $\mathcal{C}_1$ evaluates the function $h(x,y) = x \cdot y \oplus x \oplus y \oplus c$. If $c = 0$ then $h(x,y) = \mathrm{OR}(x,y)$. And if $c = 1$ then $h(x,y) = \mathrm{NOR}(x,y)$. As a result, we can construct the circuit over $F \cup \{c_0\}$ which evaluates constant function c_1.

$$(\mathcal{C}_1)\begin{cases} y_1' &= g(y_2, y_3) \\ y_2' &= f(x_1, x_2) \\ y_3' &= g(x_1, x_2) \end{cases}$$

Case 2 $(a = 1, b = 0)$

In this case, the circuit $\mathcal{C}_2$ evaluates the function $h(x,y)$. If $c = 0$ then $h(x,y) = \mathrm{OR}(x,y)$. And if $c = 1$ then $h(x,y) = \mathrm{NOR}(x,y)$. As a result, we can construct the circuit over $F \cup \{c_0\}$ which evaluates constant function c_1.

$$(\mathcal{C}_2)\begin{cases} y_1' &= g(y_2, y_3) \\ y_2' &= f(x_1, x_2) \\ y_3' &= g(y_4, x_2) \\ y_4' &= c_0 \end{cases}$$

Case 3 $(a = 0, b = 1)$

In this case, the circuit $\mathcal{C}_3$ evaluates the function $h(x,y)$. If $c = 0$ then $h(x,y) = \mathrm{OR}(x,y)$. And if $c = 1$ then $h(x,y) = \mathrm{NOR}(x,y)$. As a result, we can construct the circuit over $F \cup \{c_0\}$ which evaluates constant function c_1.

$$(\mathcal{C}_3)\begin{cases} y_1' &= g(y_2, y_3) \\ y_2' &= f(x_1, x_2) \\ y_3' &= g(x_1, y_4) \\ y_4' &= c_0 \end{cases}$$

Case 4 $(a = 1, b = 1)$

In this case, the circuit C_4 evaluates the function $h(x, y)$. If $c = 0$ then $h(x, y) = \mathrm{OR}(x, y)$. And if $c = 1$ then $h(x, y) = \mathrm{NOR}(x, y)$. As a result, we can construct the circuit over $F \cup \{c_0\}$ which evaluates constant function c_1.

$$(C_4)\begin{cases} y_1' &= g(y_2, y_3) \\ y_2' &= f(x_1, x_2) \\ y_3' &= g(y_4, y_4) \\ y_4' &= c_0 \end{cases}$$

□

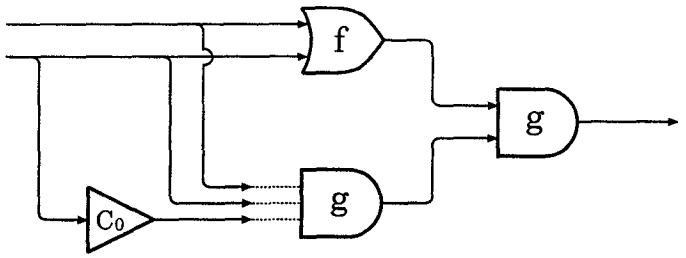

Figure 10: Realization of OR or NOR by f, g and c_0

By the same arguments in above proof, we can show the following proposition.

Proposition 8.22 *Suppose that* $f, g \oplus 1 \in \langle F \cup \{c_1\}\rangle$. *Then* $c_0 \in \langle F \cup \{c_0\}\rangle$.

Theorem 8.23 *Let* F *be a set of logical functions.* F *is NS-complete iff it is not contained in any of the five sets* $M^{\leq}, L, S, N_0$ *and* N_1.

Proof

Assume that F is not contained in any of the five sets $M^{\leq}, L, S, N_0$ and N_1. We can realize either c_0 or c_1 because of Proposition 8.15.

If we can realize the constant function c_0 then we can realize another constant function c_1 because of Proposition 8.17,Proposition 8.19 and Proposition 8.21. On the other hand,

if we can realize the constant function c_1 then we can realize another constant function c_2 because of Proposition 8.17,Proposition 8.19 and Proposition 8.22. As a result, we can realize both of c_0 and c_1,i.e. $\mathbf{Const}_2 \subset \langle F \rangle$.

Hence $F \cup \mathbf{Const}_2$ is not contained in any of the five sets $M^{\leq}$, L, S, M_0 and M_1. It shows F is NS-complete by virtue of Lemma 8.8. □

References

[1] J. W. Butler, *On complete and independent sets of operations in finite algebra*, Pacific J. Math., vol. 10 (1960), pp. 1169-1179.

[2] T. Hikita, *A Completeness Criterion for Specra*, SIAM J. Comput., Vol. 6, No. 2 (1977), pp. 285–297.

[3] D.A. Huffman, *The Synthesis of Sequential Switching Circuits*, J. Franklin Inst., Vol. 257(1954), pp. 161-190, pp. 275-303.

[4] D.A. Huffman, *A Study of the Memory Requirements of Sequential Switching Circuits*, Tech. Report No. 293(1955), Res. Lab. of Electronics, MIT.

[5] 伊吹公夫, 万能性を有する論理回路の研究, 通研成果報告, 3747 (1968-03)

[6] K. Inagaki, *t_6-completeness of Delayed Gates* (in Japanese), Trans. IECE Japan 1980/10, 63-D, pp. 827-834.

[7] K. Inagaki, *Weak t-Completeness of Sets of Delayed Logic Elements* (in Japanese), Trans. IECE Japan 1980/10, 63-D, pp. 835-842.

[8] S.V. Jablonskiĭ, "On functional completeness in the three-valued calculus," Dokl. Akad. Nauk SSSR, 95, pp.1153–1155, 1954.

[9] S.V. Jablonskiĭ, "Functional constructions in a k-valued logic," Trudy Mat. Inst. Steklov, 51, pp.5–142, 1958.

[10] I. Kimura, *Extensions of asynchronous circuits and the delay problem I*, J. Comput. System Sci. vol. 2 (1968), pp.251-287.

[11] V.B. Kudryavcev, *Completeness theorem for a class of automata without feedback couplings*, Problemy Kibernet., 8 (1959), pp.91–115.

[12] V.B. Kudryavcev, *Completeness theorem for a class of automata without feedback couplings*, Dokl. Akad. Nauk SSSR, 132(1969), pp. 272–274=Soviet Math. Dokl., 1 (1960), pp. 537–539.

[13] F. L. Luconi, Completely functional asynchronous computational structures, in "IEEE Conference Record of the Eighth Annual Symposium on Switching Theory and Automata

[14] R.E. Miller, *"Switching theory, Vol II"*,John Wiley & Sons, 1965.

[15] D. E. Muller and W. S. Bartky. A theory of asynchronous circuits,in "Proceedings of an International Symposium on the Switching Theory" vol. 29 of the Annals of the Computation Laboratory of Harvard University, pp. 204-243. Harvard University Press, Cambridge,Mass, 1959.

[16] J. von Neumann , *Probablistic Logics and The Synthesis of Reliable Organisms From Unreliable Components*, in "AUTOMATA STUDIES", C.E Shannon and J. McCarthy ed. ,Princeton University Press, 1956.

[17] A. Nozaki, *Réalisation des fonctions definies dans un ensemble fini á l'aide des organes élémentaires d'entrée-sortie*, Proc. Japan Acad., 46 (1970), pp. 478–482.

[18] A.Nozaki, *Hazard Analysis of Asynchronous Circuits in Muller-Bartky's Sense*, J.Comput. System Sci. Vol. 13, No. 2 (1976), pp. 161-171.

[19] A. Nozaki, *Functional completeness of multi-valued logical functions under uniform compositions*, Rep. Fac. Eng., Yamanashi Univ., 29 (1978), pp. 61–67.

[20] A. Nozaki, *Completeness of Logical Gates based on Sequential Circuits* (in Japanese), Trans. IECE Japan, 1982/2, J65-D, 171-178.

[21] E. L. Post, *The Two-valued Iterative Systems of Mathematical Logic*, Princeton Annals of Math. Studies, 5, Princeton University Press,Princeton, N.J, 1941.

[22] I. Rosenberg, *La structure des fonctions de plusieurs variables sur un ensemble fini*, C.R. Acad. Sci. Paris Sér. A-B, 260 (1965), pp. 3817-3819

[23] I. Rosenberg, *Über die funktionale Vollständigkeit in den mehrwertigen Logiken*, Pozpravy Ceskoslovensken Akad. Ved., 80 no. 3 (1970), pp.3-93.

[24] I. Rosenberg, Completeness properties of multiple-valued algebra, in "Computer Science and Multiple-Valued Logic, Theory and Applications", D.C. Rine ed., 114-186, North-Holland, Amsterdam, 1977.

[25] T.Sasao, "*Ronri Sekkei*"(In Japanese), KindaiKagakuSha, (1995).

[26] H. Sato, A. Nozaki and G. Pogosyan *Completeness Logical Functions Realized by Asynchronous Sequential Circuits*, J. of Info. Processing, Vol. 14, No. 2, (1991), pp. 164-171

[27] H. Sato, *On Multiple-valued Logical Functions Realized by Asynchronous Sequential Circuits*, to appear in IEICE Trans. Fundamentals, Japan.

[28] J. Slupecki, *Completeness criterion in many-valued logical system* (In Polish), Comptes Rendus des séances de la Société des Sciences et des Lettres de Varsovie, CI.III, 32(1932), pp. 102-109.

[29] D. Webb, *Definition of Post's generalized negative and maximum in terms of one binary operation*, American J. of Math., vol. 58, (1936), pp. 193-194.