The definition of sequential machine by **PSC**

Tadao Ishii*

Abstract

In this paper we will show how to define a sequential machine by using **PSC** system with Γ as a finite set of pair-sentence formulas. A pair-sentence form (A^0, B^1) means that if we assume each stage number i specifies the delay time of the link operation between pair-sentences, then the pair-sentence form (A^i, B^{i+1}) shows (A^i, A^{i+1}) (where $A^{i+1} := B^i$) that is the propagation of truth value of a sentence A at one delay time. So, we can define the one delay truth signal circuit by using a pair-sentence form. Moreover, to combine the plural pair-sentences in Γ, we can define $\boldsymbol{M_S} = \langle \boldsymbol{TV_\Gamma}, \boldsymbol{I_\Gamma}, \boldsymbol{O}, v^0, \delta_\Gamma, \lambda_\Gamma \rangle$ as a sequential machine generated from Γ, where $\boldsymbol{TV_\Gamma}$ a set of truth value products, $\boldsymbol{I_\Gamma}$ a set of input truth value products, $\boldsymbol{O}$ a set of output truth value products, $\delta_\Gamma : \boldsymbol{TV_\Gamma}^i \times \boldsymbol{I_\Gamma}^i \to \boldsymbol{TV_\Gamma}^{i+1}$ a truth value transition function and $\lambda_\Gamma : \boldsymbol{TV_\Gamma}^i \times \boldsymbol{I_\Gamma}^i \to \boldsymbol{O}^i$ a truth value output function.

Keywords: **SCI**, Liar paradox, revision theory, sequential machine, 3-valued Łukasiewicz logic.

1 Introduction

The Liar sentence has been studied in connection with several theories of truth [12, 8, 5]. As a typical such theory, Tarski had proposed *T-biconditionals* for L^0 in L^1 such that X is true in $L^1 \iff p$ in L^0, where X is replaced by the standard name of a sentence in L^0 and p is replaced by the sentence that is being true in L^0. The language L^0 for which the definition is constructed is called the *object language*, and the language L^1 in which the definition is given is called the *metalanguage*. For example, let's consider a simple Liar sentence:

(This sentence) : "This sentence is not true".

Then, at first we have the following identity. (1) This sentence = "This sentence is not true", and by applying the Tarski's biconditionals to this sentence, we get (2) "This sentence is not true" is true $\iff$ This sentence is not true. So, by substitution of (1) to (2), we get (3) This sentence is true $\iff$ This sentence is not true, which immediately yields a contradiction. The Liar paradox appears to show that the fundamental intuition is incoherent. In general, a central problem in the theory of truth is to resolve the paradox without damaging the fundamental intuition in any essential way.

To deal with the Liar paradox normally, we have introduced a system **PSC** [6, 7] that just rejects the principle of identity "A is A", one of the third Aristotelian principles for thinking, as a conservative extension of R. Suszko's non-Fregean logic **SCI** [2, 11]. **PSC** was obtained from the classical sentential calculus by adding a new pair-sentence connective $((\_)^i, (\_)^j)$, where i, j are some stage numbers. Frege, Ramsey and others have made the observation that the sentence '"A" is true' has the same meaning as A itself, and the addition of the truth predicate does not contribute any new content to the sentence A. So, our **PSC** does not include the truth predicate against the ordinary truth theory. If we consider a simple Liar sentence in **PSC**, and define A="This sentence is true", then we get a pair-sentence form $(A^0, \neg A^1)$ with intent to mean that a *situation* of A on stage 0 is *referential* to the *situation* of $\neg A$ on stage 1. More

*Department of Information Systems, School of Information and Culture, Niigata University of Information and International Studies

precisely speaking, we assume that any formulas A appear in the pair-sentence $(A^0, \neg A^1)$ has a situation with superscript i and if some situation A^i holds then its next linked situation A^{i+1} is referred by $A^{i+1} := \neg A^i$. Hence the referential relation is similar to identity connective $\equiv$, but more general notion just as a mutual link relation between sentences A and $\neg A$, and even that can be established between contradict ones if we introduce the stage notion i on which each sentence is valid.

In this paper we will show how to define a sequential machine by using **PSC** system with Γ as a finite set of pair-sentence formulas. For example, a pair-sentence form (A^0, B^1) means that if we assume each stage number i specifies the delay time of the link operation between pair-sentences A and B, then the pair-sentence form (A^i, B^{i+1}) shows (A^i, A^{i+1}) (where $A^{i+1} := B^i$) that is the propagation of truth value of a sentence A from A^i to A^{i+1} at one delay time. So, we can define the one delay truth signal circuit by using a pair-sentence form. Moreover, in the case of Liar sentence we have a pair-sentence form $(A^0, \neg A^1)$ which can show the self-referential negative feedback of the truth value of A. Let $\Gamma = \{(A_1^0, B_1^1), (A_2^0, B_2^1), \ldots, (A_m^0, B_m^1)\}$ be a finite set of pair-sentence formulas, $X^0 = \{A_1, A_2, \ldots, A_m\}$ and $X^1 = \{B_1, B_2, \ldots, B_m\}$ are sets of all formulas appeared in the first stage 0 and the second stage 1, respectively. If we introduce that $\boldsymbol{TV_\Gamma} = \{\Pi_{j=1}^m v(A^j); A^j \in (Sub(X^0) \cap Sub(X^1))\}$ a set of all truth value assignment products where $Sub(X^i)$ a set of all subformulas of each element of X^i $(i \in \{0, 1\})$, $\boldsymbol{I_\Gamma} = \{\Pi_{j=1}^k v(C^j); C^j \in Prim(X)\}$ a set of all input truth value assignment products where $X = Sub(X^1)/Sub(X^0)$ a subtraction of each $Sub(X^i)$ and $Prim(X)$ a set of all primitive formulas of X, $\boldsymbol{O} = \{\Pi_{j=1}^l v(D^j); D^j \in Y\}$ a set of all output truth value assignment products where Y a set of all output formulas, $v^0 \in \boldsymbol{TV_\Gamma}$ an initial truth value assignment product, $\delta_\Gamma : \boldsymbol{TV_\Gamma}^i \times \boldsymbol{I_\Gamma}^i \to \boldsymbol{TV_\Gamma}^{i+1}$ a truth value assignment transition function, $\lambda_\Gamma : \boldsymbol{TV_\Gamma}^i \times \boldsymbol{I_\Gamma}^i \to \boldsymbol{O}^i$ a truth value assignment output function and $\boldsymbol{F}$ a set of all final(accepted) truth value assignment, then we can define both $\boldsymbol{M_S} = \langle \boldsymbol{TV_\Gamma}, \boldsymbol{I_\Gamma}, \boldsymbol{O}, v^0, \delta_\Gamma, \lambda_\Gamma \rangle$ as a sequential machine and also $\boldsymbol{M_A} = \langle \boldsymbol{TV_\Gamma}, \boldsymbol{I_\Gamma}, v^0, \delta_\Gamma, \boldsymbol{F} \rangle$ as a finite automaton, generated from Γ.

2 PSC Logic

2.1 Formal System of PSC

Let $\mathcal{L}_P =< \boldsymbol{FOR_P}, \neg, \wedge, \vee, \to, ((\_)^i,\ (\_)^j), \top, \bot >$ be a language of the sentential calculus with pair-sentence connective. The formulas $\boldsymbol{FOR_P}$ of a language $\mathcal{L}_P$ are generated in the usual way from an infinite set $\boldsymbol{VAR_P}$ of sentential variables, constants $\top$(true) and $\bot$(false) by the standard truth functional connectives $\neg$(negation), $\wedge$ (conjunction), $\vee$ (disjunction) and $\to$ (material implication) as well as the pair-sentence constructor $((\_)^i,\ (\_)^j)$, where $i, j \in \mathbf{N}$ are some stage numbers. In our language $\mathcal{L}_P$, we assume that every sentential variables are defined on an initial stage number $0 \in \mathbf{N}$. So, we have:
(1) $\boldsymbol{VAR_P} = \boldsymbol{VAR}^0 = \{p^0, q^0, r^0, \ldots\}$
(2) $\boldsymbol{VAR_P} \subseteq \boldsymbol{FOR_P}$
(3) $\forall A, B \in \boldsymbol{FOR_P} \Longrightarrow \neg A, A \wedge B, A \vee B, A \to B, (A, B) \in \boldsymbol{FOR_P}$

Also we may use the same parentheses as auxiliary symbols even assume that the priority of each connective is weak as $\neg$, $\wedge$, $\vee$, $\to$, $(\_\,,\_)$ in order. Throughout this paper the letters p, q, r, p^0, p^1, p^2, ... will be used to denote any variables, the letters A, B, C, A^0, A^1, A^2, ... formulas of a language $\mathcal{L}_P$, the letters X, Y, Z, ... sets of formulas, and Greek letters $\Gamma, \Sigma, \Delta, \ldots$ sets of pair-sentence formulas. Moreover, two constants $\top$ and $\bot$ are defined as $p^0 \vee (\neg p)^0$ and $p^0 \wedge (\neg p)^0$ for some $p^0 \in \boldsymbol{VAR}^0$, respectively. At first we will introduce several terminology for pair-sentences as the following.

Definition 2.1 (Pair-sentence) (1) *For any sentence $A \in \boldsymbol{FOR_P}$, if there exist some sentence $B \in \boldsymbol{FOR_P}$ such that "A is B" is also a new sentence, then we assume that there exists a sentence $(A^0, B^1) \in \boldsymbol{FOR_P}$, which means that there exists A^1 on the next linked stage of A^0 such that A^1 is referential to B^0, and call (A^0, B^1) a* pair-sentence formula *of*

A^0 and B^1. Otherwise, we assume that there exists a senetnce $(A^0, A^0) \in \boldsymbol{FOR_P}$, and call (A^0, A^0) a unit of pair-sentence form *for A^0.*

(2) *The referential stage numbering of composed formulas is the following: for any stage numbers $i, j, k \in \mathbf{N}$,*
(i) $(\neg A^i)^j \iff \neg(A^{i+j})$
(ii) $(A^i \,\%\, B^j)^k \iff A^{i+k} \,\%\, B^{j+k}$ *where* $\% \in \{\wedge, \vee, \rightarrow\}$
(iii) $(A^i, B^j)^k \iff (A^{i+k}, B^{j+k})$

(3) *If some sentence A has only a unit of pair-sentence form, then we assume that: $A^i \rightarrow (A^i)^{\pm n}$ for every $i, n \in \mathbf{N}$.*

The superscript of each formula shows the referential stage number on which the formula is valid. The referential stage number will start from 0 and increase with depending on the referential frequency like $0, 1, 2, 3, \ldots$. If we will interpret "A is B" as an identical sentence (A^0, B^0), that is $A^0 \equiv B^0$ in Suszko's notation, then we will get **SCI** system.

Example 2.2 (1) *For any sentence $A \in \boldsymbol{FOR_P}$, "A is A" if and only if there exist A^0, A^1, $(A^0, A^1) \in \boldsymbol{FOR_P}$ by Definition 2.1 (1). So, we have* $\{$"A is A"$\} \iff \Gamma_1 = \{(A^0, A^1)\}$.

(2) *Similarly, for any $A, B, C \in \boldsymbol{FOR_P}$,*
(i) $\{$"A is not A"$\} \iff \Gamma_2 = \{(A^0, \neg A^1)\}$
(ii) $\{$"A is not B", "B is not C", "C is A"$\}$
$\iff \Gamma_3 = \{(A^0, \neg B^1), (B^0, \neg C^1), (C^0, A^1)\}$
(iii) $\{$"C is $(A \vee (B \wedge \neg C))$"$\} \iff \Gamma_4 = \{(C^0, (A \vee (B \wedge \neg C))^1)\}$
(iv) $\{$"B is C", "C is $(\neg A \wedge \neg B \wedge C) \vee (A \wedge \neg B)$"$\}$
$\iff \Gamma_5 = \{(B^0, C^1), (C^0, ((\neg A \wedge \neg B \wedge C) \vee (A \wedge \neg B))^1)\}$

Definition 2.3 *Let Γ be a set of pair-sentence formulas $\{(A^0, B_1^1), (B_1^0, B_2^1), (B_2^0, B_3^1), \ldots, (B_{n-1}^0, B_n^1)\}$ $(\exists n \in \mathbf{N})$. Then we get $\Gamma = \{(A^0, B_1^1), (B_1^1, B_2^2), (B_2^2, B_3^3), \ldots, (B_{n-1}^{n-1}, B_n^n)\}$ by Definition 2.1(3). So,*

(1) *We say that a sequence of formulas $A^0 B_1^1 B_2^2 \cdots B_n^n$ is a* referential pattern *of formula A generated from Γ.*

(2) *If A is belong to a set of formulas $\{B_1^1, B_2^2, \ldots, B_n^n\}$, we say that A has a* circular referential relation *with respect to Γ. Otherwise, A has a* non-circular referential relation *with respect to Γ.*

(3) *The* referential cycle number *of A with respect to Γ, $\tau(A, \Gamma)$ in symbol, is defined as follows:*
(i) $\tau(A, \Gamma) = 0$ *if* $A \notin \{B_1^1, B_2^2, \ldots, B_n^n\}$,
(ii) $\tau(A, \Gamma) = n$ *if* $A \in \{B_1^1, B_2^2, \ldots, B_n^n\}$ *and* $A = B_n^n$.
So, if A has a circular referential relation with respect to Γ, $\tau(A, \Gamma) \geq 1$. Otherwise, $\tau(A, \Gamma) = 0$.

(4) *If $\tau(A, \Gamma) \leq 1$, we say that A is* categorical *with respect to Γ. Otherwise, A is* paradoxical *with respect to Γ.*

Definition 2.4 (PSC system) *The axiomatic system* **PSC** *for the language $\mathcal{L}_P$ consists of two sets of schema $\boldsymbol{TFA}$ (truth functional axioms) and $\boldsymbol{PSA}$ (pair-sentence axioms) below:*
$(A1) - (A10)$ *classical truth functional axioms*
$(E1)$ (A, A)
$(E2)$ $(A, B) \rightarrow (B, A)$
$(E3)$ $(A, B) \wedge (B, C) \rightarrow (A, C)$
$(C1)$ $(A, B) \rightarrow (\neg A, \neg B)$

$(C2)$ $(A, B) \wedge (C, D) \rightarrow ((A \wedge C), (B \wedge D))$
$(C3)$ $(A, B) \wedge (C, D) \rightarrow ((A \vee C), (B \vee D))$
$(C4)$ $(A, B) \wedge (C, D) \rightarrow ((A \rightarrow C), (B \rightarrow D))$
$(C5)$ $(A, B) \wedge (C, D) \rightarrow ((A, C), (B, D))$
$(P1)$ $(A, B) \rightarrow (A \rightarrow B)$
$(P2)$ $A \rightarrow A^{\pm n}$ $(\forall n \in \mathbf{N})$ *if A has only a unit of pair-sentence form*
(Mp) $\dfrac{A \quad A \rightarrow B}{B}$

The axioms in $\boldsymbol{TFA}$ with modus ponens as a single rule give an axiomatic system $\boldsymbol{CL}$ for the classical sentential logic. If we define a system $\mathbf{PSC}^0$ by restricting the stage number as $0 \in \mathbf{N}$ only in a language $\mathcal{L}_P$, i.e., "A is B" $\Longleftrightarrow$ there exist $A^0,\ B^0,\ (A^0, B^0) \in \boldsymbol{FOR_P}^0$, and hence, eliminating axioms (P2) from $\mathbf{PSC}$, then the system $\mathbf{PSC}^0$ is collapsed into systems $\mathbf{SCI}$ because in this case we can regard every pair-sentence formula $(A, B)^0$ as an identity formula $(A \equiv B)^0$ in $\mathbf{SCI}$ on stage 0.

Definition 2.5 (Derivability) *Let Γ be a finite set of pair-sentence formulas in a language $\mathcal{L}_P$, X a finite set of formulas, A a formula and* $\mathbf{PSC}$ *a system in $\mathcal{L}_P$. Then we say that:*

(1) *A^j is* derivable *from X based on Γ in* $\mathbf{PSC}$*,* $\mathbf{PSC}, X \vdash^{\Gamma} A^j$ *in symbol, if there is a sequence of formulas $B_1^{i_1}, B_2^{i_2}, \ldots, B_{n-1}^{i_{n-1}}, B_n^{i_n} (n \geq 1)$ such that $B_n^{i_n} = A^j$ and every formula in the sequence $B_1^{i_1}, B_2^{i_2}, \ldots, B_{n-1}^{i_{n-1}}, A^j$ is either an axiom of* $\mathbf{PSC}$*, or belongs to $X \cup \Gamma$, or is obtained by (Mp) rule from formulas occurring before it in the sequence. n is a* length of derivation *A^j from X based on Γ in* $\mathbf{PSC}$*.*

(2) *A is* derivable *from X based on Γ in* $\mathbf{PSC}$*,* $\mathbf{PSC}, X \vdash^{\Gamma} A$ *in symbol, if there is a sequence of formulas $B_1^0, B_2^0, \ldots, B_{n-1}^0, B_n^0 (n \geq 1)$ such that $B_n^0 = A^0$ and every formula in the sequence $B_1^0, B_2^0, \ldots, B_{n-1}^0$, A^0 is either an axiom of* $\mathbf{PSC}$*, or belongs to $X \cup \Gamma$, or is obtained by (Mp) rule from formulas occurring before it in the sequence.*

(3) *If $X = \emptyset$,* $\mathbf{PSC} \vdash^{\Gamma} A$ *in symbol, A is a* theorem *of* $\mathbf{PSC}$ *based on Γ.*

Proposition 2.6 *Let $\Gamma_1 = \{(A^0, \neg A^1)\}$, $\Gamma_2 = \{(A^0, \neg A^1), (A^0, A^3)\}$ and $\Gamma_3 = \{(A^0, (B \vee (C \wedge \neg A))^1)\}$. Then,*
(1) $\mathbf{PSC}, A^0 \vdash^{\Gamma_1} \neg A^1$
(2) $\mathbf{PSC} \vdash^{\Gamma_1} (A^0, \neg\neg A^2)$
(3) $\mathbf{PSC} \vdash^{\Gamma_2} \bot$
(4) $\mathbf{PSC} \vdash^{\Gamma_3} B \rightarrow A$

Definition 2.7 (Elementary extensions of PSC) *Let us assume the following additional axioms:* $(P3)$ $(A^i, B^j) \wedge (B \leftrightarrow C)^j \rightarrow (A^i, C^j)$ $(\forall i, j \in \mathbf{N})$
$(P4)$ $(A, A^{\pm n})$ $(\exists n \geq 1)$ $(n - reflexivity)$
Then, some elementary extensions of $\mathbf{PSC}$ *are defined as follows:*
(1) $\mathbf{PSC_B} := \mathbf{PSC} \oplus (P3)$
(2) $\mathbf{PSC_n} := \mathbf{PSC} \oplus (P4)$
(3) $\mathbf{PSC_{Bn}} := \mathbf{PSC} \oplus (P3) \oplus (P4)$

2.2 Semantices of PSC

We interpret $\mathcal{L}_P$ by using a classical truth assignment function $v : \boldsymbol{VAR_P} \rightarrow \{0, 1\}$ where $\boldsymbol{VAR_P}$ is a set of sentencical variables $\boldsymbol{VAR}^0 = \{p^0, q^0, r^0, \ldots\}$ on stage $0 \in \mathbf{N}$. Then we can easily extend the function v to the domain of all formulas $\boldsymbol{FOR_P}$ in a language $\mathcal{L}_P$. The assignment for all logical connectives $\{\neg, \wedge, \vee, \rightarrow\}$ are as usual way, but we will use the truth transition function $\delta^{j-i} : \boldsymbol{TV}^i \times \hat{\boldsymbol{I}}^{j-i} \rightarrow \boldsymbol{TV}^j$ to interpret a pair-sentence formula (A^i, B^j) where $\boldsymbol{TV}^i = \{v(A^i); A^i \in \boldsymbol{FOR_P}\}$ and $\hat{\boldsymbol{I}}^{j-i} = \boldsymbol{I}^i \times \boldsymbol{I}^{i+1} \times \cdots \times \boldsymbol{I}^j$ where $\boldsymbol{I}^k = \{i_1^k, i_2^k, \ldots, i_m^k\}$ is a set of input truth value products on stage k. The n-th order of truth transition function δ^n is defined as follow:

Definition 2.8 (Truth transition function) *Let Γ be a finite set of pair-sentence formulas $\{(A_1^0, B_1^1), (A_2^0, B_2^1), \ldots, (A_n^0, B_n^1)\}$, X^0 a set of all formulas appeared in the first stage 0 $\{A_1, A_2, \ldots, A_n\}$ and X^1 a set of all formulas appeared in the second stage 1 $\{B_1, B_2, \ldots, B_n\}$.*

(1) *Let us define $\boldsymbol{TV}^i = \{v(A^i); A^i \in X^i\}$ a set of truth value assignment on stage $i \in \{0, 1\}$, $\boldsymbol{I} = \{\Pi_{j=0}^n v(A^j); A^j \in Prim(X)\}$ a set of all input truth value assignment products for all formulas which have a unit of pair-sentence form in Γ where $X = Sub(X^1)/Sub(X^0)$ a subtraction of each $Sub(X^i)$, $Sub(X^i)$ a set of all subformulas of each element of X^i and $Prim(X)$ a set of all primitive formulas of X. Then $\delta_\Gamma : \boldsymbol{TV}^0 \times \boldsymbol{I} \to \boldsymbol{TV}^1$ is a* truth transition function *determined by Γ such that $v((A_k^0, B_k^1)) = 1$ if and only if $\delta_\Gamma(v(A_k^0), \boldsymbol{I}) = v(B_k^1)$ $(1 \leq k \leq n)$.*

(2) *Moreover, the following is a* sequence of truth transition functions *determined by δ_Γ : for any initial truth value assignment $v_0 \in \boldsymbol{TV}^0$,*
$${\delta_\Gamma}^0(v_0, \hat{\boldsymbol{I}}^0) = v_0$$
$${\delta_\Gamma}^{n+1}(v_0, \hat{\boldsymbol{I}}^{n+1}) = \delta_\Gamma({\delta_\Gamma}^n(v_0, \hat{\boldsymbol{I}}^n), \boldsymbol{I})$$
where $\hat{\boldsymbol{I}}^0 =< \epsilon >$(empty input), $\hat{\boldsymbol{I}}^n = \overbrace{\boldsymbol{I} \times \cdots \times \boldsymbol{I}}^{n}$ and $n \geq 0$ is an order of truth transition function.

(3) *v_0 is* n-reflexive *with respect to Γ if ${\delta_\Gamma}^n(v_0, \hat{\boldsymbol{I}}^n) = v_0$ $(\exists n \in \mathbf{N})$.*

(4) *${\delta_\Gamma}^{-1}$ is a* reverse truth transition function *of δ_Γ if ${\delta_\Gamma}^{-1}(\delta_\Gamma(v_0, \boldsymbol{I}), \boldsymbol{I}^{-1}) = v_0$.*

We notice that 1-reflexive assignments are fixed points of δ_Γ, 2-reflexive ones have 2 as a cycle number and every initial assignment v_0 is 0-reflexive. Then we can easily extend this function δ_Γ to the domain of all elements in an Boolean algebra as follows.

Definition 2.9 *Let Γ be a finite set of pair-sentence formulas, $\mathcal{A}_P = \langle \boldsymbol{A_P}, \sim, \cap, \cup, \supset, (\_ : \_), 1, 0\rangle$ an* **PSC***-algebra and $\boldsymbol{D_P}$ a subset of $\boldsymbol{A_P}$.*

(1) *An assignment of $\mathcal{A}_P$ is a homomorphism $v : \mathcal{L}_P \to \mathcal{A}_P$ such that the following hold: for any $A, B \in \boldsymbol{FOR_P}$,*
(*i*) $v(A^i) \Longleftrightarrow (v(A))^i$ $(\forall i \in \mathbf{N})$
(*ii*) $v(\neg A) \Longleftrightarrow \sim v(A)$
(*iii*) $v(A\%B) \Longleftrightarrow v(A)\dot{\%}v(B)$ *where* $\% \in \{\wedge, \vee, \to\}$ *and* $\dot{\%} \in \{\cap, \cup, \supset\}$ *is an algebraic counterpart of* $\%$ *in order*
(*iv*) $v((A, B)) \Longleftrightarrow (v(A) : v(B))$
(*v*) $v(\top) = 1$ *and* $v(\bot) = 0$

(2) *$\delta_\Gamma : {\mathcal{A}_P}^0 \times \boldsymbol{I} \to {\mathcal{A}_P}^1$ is a Boolean transition function determined by Γ, where ${\mathcal{A}_P}^i$ is an Boolean algebra on order i $(i = 0, 1)$ and $\boldsymbol{I}$ is an Boolean input elements list.*

(3) *The ordering of composed elements is the following: for every elements $a^m, b^n \in \boldsymbol{A_P}$ and number $l \in \mathbf{N}$,*
(*i*) $(\sim a^m)^l \Longleftrightarrow \sim a^{m+l}$
(*ii*) $(a^m \,\dot{\%}\, b^n)^l \Longleftrightarrow (a^{m+l} \,\dot{\%}\, b^{n+l})$ *where* $\dot{\%} \in \{\cap, \cup, \supset, :\}$

(4) (*i*) *$\boldsymbol{D_P}$ is* closed *if for every elements $a^m, b^n \in \boldsymbol{A_P}$, $a^m \in \boldsymbol{D_P}$ and $a^m \supset b^n \in \boldsymbol{D_P}$ imply $b^n \in \boldsymbol{D_P}$.* (*ii*) *$\boldsymbol{D_P}$ is* proper *if $\boldsymbol{D_P} \neq \boldsymbol{A_P}$.* (*iii*) *$\boldsymbol{D_P}$ is* admissible *if for every assignment v of $\mathcal{A}_P$ and formula $A \in \boldsymbol{TFA} \sqcup \boldsymbol{PSA}$, $v(A) \in \boldsymbol{D_P}$.* (*iv*) *$\boldsymbol{D_P}$ is* prime *if for every element $a^m \in \boldsymbol{A_P}$, $a^m \in \boldsymbol{D_P}$ or $\sim a^m \in \boldsymbol{D_P}$.* (*v*) *$\boldsymbol{D_P}$ is* transit *if for every elements*

$a^m, b^n \in \boldsymbol{A_P}$ *and some input elements list* $\hat{\boldsymbol{I}}^{n-m} \subseteq \overbrace{\boldsymbol{A_P} \times \cdots \times \boldsymbol{A_P}}^{n-m}$, $(a^m : b^n) \in \boldsymbol{D_P} \iff \delta_\Gamma{}^{n-m}(a^m, \hat{\boldsymbol{I}}^{n-m}) = b^n$. *(vi)* $\boldsymbol{D_P}$ *is* normal *if for every elements* $a^m, b^m \in \boldsymbol{A_P}$ *and an empty input list* $\hat{\boldsymbol{I}}^0 =< \epsilon >$, $(a^m : b^m) \in \boldsymbol{D_P} \iff \delta_\Gamma{}^0(a^m, \hat{\boldsymbol{I}}^0) = b^m \iff a^m = b^m$.

(5) $\boldsymbol{D_P}$ *is* filter *if* $\boldsymbol{D_P}$ *is proper, closed and admissible.*

Definition 2.10 *Let* Γ *be a finite set of pair-sentence formulas,* X *a finite set of formulas,* A *a formula and* $\mathcal{A}_P$ *an* **PSC**-*algebra.*

(1) $\mathcal{M}_P = \langle \mathcal{A}_P, \boldsymbol{D_P} \rangle$ *is a* **PSC**-*matrix if* $\boldsymbol{D_P}$ *is a filter in* $\mathcal{A}_P$.

(2) *Moreover,* $\mathcal{M}_P$ *is a* **PSC**-*model if* $\boldsymbol{D_P}$ *is a prime* ($1 \in \boldsymbol{D_P}$ *and* $0 \notin \boldsymbol{D_P}$)*, transit filter.*

(3) A *is* true *in a* **PSC**-*model* $\mathcal{M}_P$ *under the assumption of* X *based on* Γ, $\mathcal{M}_P, X \models^\Gamma A$ *in symbol, if for every assigmnent* v *of* $\mathcal{A}_P$, $v(X \cup \Gamma) \subseteq \boldsymbol{D_P}$ *implies* $v(A) \in \boldsymbol{D_P}$.

(4) A *is* valid *under the assumption of* X *based on* Γ, $X \models^\Gamma A$ *in symbol, if for every* **PSC**-*model,* $\mathcal{M}_P, X \models^\Gamma A$.

Definition 2.11 *Let* $\mathcal{M}_P = \langle \mathcal{A}_P, \boldsymbol{D_P} \rangle$ *be a* **PSC**-*matrix. Then we define the following:* $\forall a^m, b^n \in \boldsymbol{A_P}$,

(1) $\approx$ *is a binary relation on* $\boldsymbol{A_P}$ *such that* $a^m \approx b^n \Leftrightarrow (a^m : b^n) \in \boldsymbol{D_P}$.

(2) $|a^m|$ *is the congruence class of element* a^m*, i.e.,* $|a^m| = \{b^n; a^m \approx b^n\}$.

(3) $\boldsymbol{A_P}/\!\approx$ *is the set of congruence classes of elements of* $\boldsymbol{A_P}$*, i.e.,* $\boldsymbol{A_P}/\!\approx = \{|a^m|; a^m \in \boldsymbol{A_P}\}$.

(4) $\mathcal{A}_P/\!\approx = \langle \boldsymbol{A_P}/\!\approx, \sim, \cap, \cup, \supset, (\_ : \_), |1|, |0| \rangle$ *is an* **PSC**-*algebra with the following definitions: for every* $|a^m|, |b^n| \in \boldsymbol{A_P}/\!\approx$,
(i) $\sim |a^m| \iff |\sim a^m|$
(ii) $|a^m| \dot{\%} |b^n| \iff |a^m \dot{\%} b^n|$ *where* $\dot{\%} \in \{\cap, \cup, \supset, :\}$

Proposition 2.12 *Let* Γ *be a finite set of pair-sentence formulas,* X *a finite set of formulas,* A *a formula and* $\mathcal{M}_P = \langle \mathcal{A}_P, \boldsymbol{D_P} \rangle$ *a* **PSC**-*matrix. Then we have the following:*

(1) $\boldsymbol{D_P}/\!\approx$ *is a filter in* $\boldsymbol{A_P}/\!\approx$*. So,* $\mathcal{M}_P/\!\approx = \langle \mathcal{A}_P/\!\approx, \boldsymbol{D_P}/\!\approx \rangle$ *is a* **PSC**-*matrix.*

(2) *Moreover,* $\boldsymbol{D_P}/\!\approx$ *is a transit filter in* $\boldsymbol{A_P}/\!\approx$.

(3) $\boldsymbol{D_P}/\!\approx$ *is prime if and only if* $\boldsymbol{D_P}$ *is prime in* $\boldsymbol{A_P}$.

(4) *The mapping* $a^m \mapsto |a^m|$ *is a matrix homomorphism from* $\mathcal{M}_P$ *onto* $\mathcal{M}_P/\!\approx$*. So,* $\mathcal{M}_P, X \models^\Gamma A$ *if and only if* $\mathcal{M}_P/\!\approx, X \models^\Gamma A$.

Theorem 2.13 (Completeness) *Let* Γ *be a finite set of pair-sentence formulas,* X *a finite set of formulas,* A *a formula and* $\mathcal{M}_P = \langle \mathcal{A}_P, \boldsymbol{D_P} \rangle$ *a* **PSC**-*model.*

(1) X *is consistent if and only if there exists a model* $\mathcal{M}_P$ *and an assignment* v *of* $\mathcal{A}_P$ *such that* $X \subseteq v^{-1}(\boldsymbol{D_P})$.

(2) $\mathbf{PSC}, X \vdash^\Gamma A$ *if and only if for every* **PSC**-*model* $\mathcal{M}_P$, $\mathcal{M}_P, X \models^\Gamma A$.

(3) $\mathbf{PSC} \vdash^\Gamma A$ *if and only if for every* **PSC**-*model* $\mathcal{M}_P$, $\mathcal{M}_P \models^\Gamma A$.

(4) $\mathbf{PSC} \vdash^\emptyset A$ *if and only if for every* **PSC**-*model* $\mathcal{M}_P$, $\mathcal{M}_P \models^\emptyset A$.

3 Sequential machine by PSC

We will consider a sequential machine by using **PSC** system.

Definition 3.1 *Let* $\Gamma = \{(A^0_1, B^1_1), (A^0_2, B^1_2), \ldots, (A^0_m, B^1_m)\}$ *be a finite set of pair-sentence formulas,* $X^0 = \{A_1, A_2, \ldots, A_m\}$ *and* $X^1 = \{B_1, B_2, \ldots, B_m\}$ *are sets of all formulas appeared in the first stage 0 and the second stage 1, respectively.* $X = Sub(X^1)/Sub(X^0)$ *a subtraction of each* $Sub(X^i)$ *where* $Sub(X^i)$ *a set of all subformulas of each element of* X^i $(i \in \{0, 1\})$, $Prim(X) = \{C_1, C_2, \ldots, C_k\}$ *a set of all primitive formulas of* X, $Y = \{D_1, D_2, \ldots, D_l\}$ *a set of all output formulas. Then we define sequential machine and finite automaton as follows [1, 4]:*

(1) $\boldsymbol{M_S} = \langle \boldsymbol{TV_\Gamma}, \boldsymbol{I_\Gamma}, \boldsymbol{O}, v^0, \delta_\Gamma, \lambda_\Gamma\rangle$ *is a* sequential machine *generated from* Γ*, where* $\boldsymbol{TV_\Gamma} = \{\Pi^m_{j=1} v(A^j); A^j \in (Sub(X^0) \cap Sub(X^1))\}$ *is a set of all truth value assignment products* $(m = |Sub(X^0) \cap Sub(X^1)|)$, $\boldsymbol{I_\Gamma} = \{\Pi^k_{j=1} v(C^j); C^j \in Prim(X)\}$ *a set of all input truth value assignment products* $(k = |Prim(X)|)$, $\boldsymbol{O} = \{\Pi^l_{j=1} v(D^j); D^j \in Y\}$ *a set of all output truth value assignment products* $(l = |Y|)$, $v^0 \in \boldsymbol{TV_\Gamma}$ *an initial truth value assignment product,* $\delta_\Gamma : \boldsymbol{TV_\Gamma}^i \times \boldsymbol{I_\Gamma}^i \to \boldsymbol{TV_\Gamma}^{i+1}$ *a truth value assignment transition function and* $\lambda_\Gamma : \boldsymbol{TV_\Gamma}^i \times \boldsymbol{I_\Gamma}^i \to \boldsymbol{O}^i$ *a truth value assignment output function.*

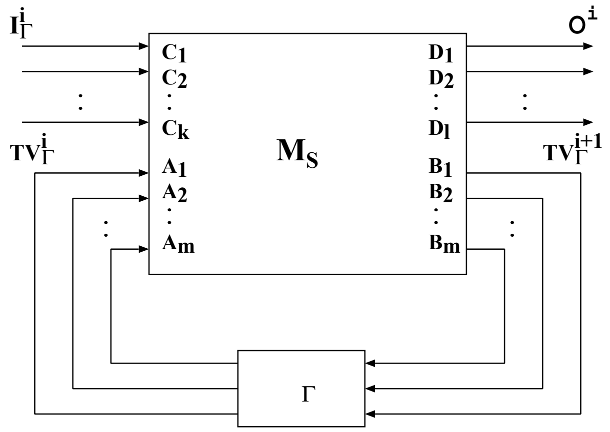

Figure 1: Sequential machine $\boldsymbol{M_S}$ generated from Γ

(2) *Moreover, if we restrict to* $\boldsymbol{O} = \{1(yes), 0(no)\}$ *in* $\boldsymbol{M_S}$ *such that* $\boldsymbol{F} = \{v \in \boldsymbol{TV_\Gamma}, i \in \boldsymbol{I_\Gamma}; \lambda_\Gamma(v, i) = 1\}$*, then* $\boldsymbol{M_A} = \langle \boldsymbol{TV_\Gamma}, \boldsymbol{I_\Gamma}, v^0, \delta_\Gamma, \boldsymbol{F}\rangle$ *is a* finite automaton *generated from* Γ*, where* $\boldsymbol{F}$ *is a set of all final(accepted) truth value assignment.*

We will investigate several sets of pair-sentence formulas.

Example 3.2 (Dialogue for Socrates, Plato and Aristoteles)

Socrates : "Plato's remarks are not true".

Plato : "Aristoteles's remarks are not true".

Aristoteles : "Socrates' remarks are true".

Let's A = "Socrates's remarks are true", B = "Plato's remarks are true" and C = "Aristoteles's remarks are true". Then we define $\Gamma_3 = \{(A^0, \neg B^1), (B^0, \neg C^1), (C^0, A^1)\}$ as a set of pair-sentence formulas. Since we have $X^0 = \{A, B, C\}$, $X^1 = \{A, \neg B, \neg C\}$ and $X = Sub(X^1)/Sub(X^0) = \{\neg B, \neg C\}$, so $\boldsymbol{I_{\Gamma 3}} = Prim(X) = \emptyset$. If we define $\delta_{\Gamma 3} : \boldsymbol{TV_{\Gamma 3}}^i \times \boldsymbol{I_{\Gamma 3}}^i \cup \{\epsilon\} \to \boldsymbol{TV_{\Gamma 3}}^{i+1}$ where

Table 1: Truth transition table of Γ_3

$\boldsymbol{TV}_{\boldsymbol{\Gamma}_3}^{i}$	A^i	B^i	C^i	$A^{i+1} \stackrel{\text{def}}{=} \neg B^i$	$B^{i+1} \stackrel{\text{def}}{=} \neg C^i$	$C^{i+1} \stackrel{\text{def}}{=} A^i$	$\boldsymbol{TV}_{\boldsymbol{\Gamma}_3}^{i+1}$
v_1	1	1	1	0	0	1	v_7
v_2	1	1	0	0	1	1	v_5
v_3	1	0	1	1	0	1	v_3
v_4	1	0	0	1	1	1	v_1
v_5	0	1	1	0	0	0	v_8
v_6	0	1	0	0	1	0	v_6
v_7	0	0	1	1	0	0	v_4
v_8	0	0	0	1	1	0	v_2

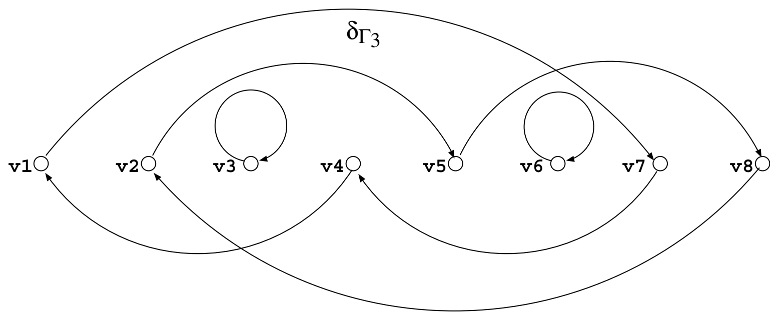

Figure 2: Boolean transition figure of $\delta_{\Gamma 3}$

ϵ is an empty input, and let's $\boldsymbol{TV}_{\boldsymbol{\Gamma}3} = \{v(A) \times v(B) \times v(C); v : Sub(X^0) \cap Sub(X^1) \to \{0,1\}\} = \{v_1, v_2, \ldots, v_8\}$, $v^0 \in \boldsymbol{TV}_{\boldsymbol{\Gamma}3}$ and $\boldsymbol{F} = \{v_3, v_6\} \subseteq \boldsymbol{TV}_{\boldsymbol{\Gamma}3}$, then we get $\boldsymbol{M}_{\boldsymbol{A}} = \langle \boldsymbol{TV}_{\boldsymbol{\Gamma}3}, \boldsymbol{I}_{\boldsymbol{\Gamma}3}, v^0, \delta_{\Gamma 3}, \boldsymbol{F} \rangle$ as a finite automaton generated from Γ_3 (see Table 1).

Example 3.3 (Circular definition in Gupta's book [3])

$$C \stackrel{\text{def}}{=} (A \vee (B \wedge \neg C))$$

In this case, we define $\Gamma_4 = \{(C^0, (A \vee (B \wedge \neg C))^1)\}$ as a set of pair-sentence formulas. Since we have $X^0 = \{C\}$, $X^1 = \{A \vee (B \wedge \neg C)\}$ and $X = Sub(X^1)/Sub(X^0) = \{A, B, \neg C, B \wedge \neg C, A \vee (B \wedge \neg C)\}$, so $Prim(X) = \{A, B\}$. If we define $\delta_{\Gamma 4} : \boldsymbol{TV}_{\boldsymbol{\Gamma}_4}^{i} \times \boldsymbol{I}_{\boldsymbol{\Gamma}_4}^{i} \to \boldsymbol{TV}_{\boldsymbol{\Gamma}_4}^{i+1}$, and let's $\boldsymbol{TV}_{\boldsymbol{\Gamma}4} = \{v(C); v : Sub(X^0) \cap Sub(X^1) \to \{0,1\}\} = \{v_1, v_2\}$, $\boldsymbol{I}_{\boldsymbol{\Gamma}4} = \{v(A) \times v(B); v : Prim(X) \to \{0,1\}\} = \{i_1, \ldots, i_4\}$, $v^0 \in \boldsymbol{TV}_{\boldsymbol{\Gamma}4}$ and $\boldsymbol{F} = \{v_1, v_2\} \subseteq \boldsymbol{TV}_{\boldsymbol{\Gamma}4}$, then we get $\boldsymbol{M}_{\boldsymbol{A}} = \langle \boldsymbol{TV}_{\boldsymbol{\Gamma}4}, \boldsymbol{I}_{\boldsymbol{\Gamma}4}, v^0, \delta_{\Gamma 4}, \boldsymbol{F} \rangle$ as a finite automaton generated from Γ_4 (see Table 2).

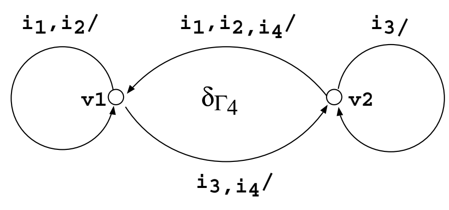

Figure 3: Boolean transition figure of $\delta_{\Gamma 4}$

Example 3.4 (Two type of traffic signals with a push button [9])

(*a*) *There exist two type of traffic signals. One signal A is for cars, and the other B with a push button is for pedestrians.*

Table 2: Truth transition table of Γ_4

$\boldsymbol{I}_{\boldsymbol{\Gamma}_4}^i$	A^i	B^i	$\boldsymbol{TV}_{\boldsymbol{\Gamma}_4}^i$	C^i	$C^{i+1} \stackrel{\text{def}}{=} A^i \vee (B^i \wedge \neg C^i)$	$\boldsymbol{TV}_{\boldsymbol{\Gamma}_4}^{i+1}$
i_1	1	1	v_1	1	1	v_1
	1	1	v_2	0	1	v_1
i_2	1	0	v_1	1	1	v_1
	1	0	v_2	0	1	v_1
i_3	0	1	v_1	1	0	v_2
	0	1	v_2	0	1	v_1
i_4	0	0	v_1	1	0	v_2
	0	0	v_2	0	0	v_1

(b) *If push a button Off, then A turns on Blue light and B on Red light.*

(c) *If push a button On at a time t, then A turns on Yellow at $t+1$ and also both A on Red and B on Blue at $t+2$.*

(d) *Moreover, at $t+3$ keep the previous state (both A on Red and B on Blue). And at $t+4$, both signals A and B return to the inital state (both A on Blue and B on Red).*

(e) *While A turns on Yellow or Red, it is invalid to push the button.*

We will show the time table of each signal A and B (see Table 3). From this table, we need the following sets $\boldsymbol{I}, \boldsymbol{D}$ and $\boldsymbol{O}$ to define the sequential machine.

Table 3: Time table of each signal A and B

	t	t+1	t+2	t+3	t+4	
Time	1	2	3	4	5	6
Buttom	Off	On	--	--	--	
Signal A	Blue	Blue	Yellow	Red	Red	Blue
Signal B	Red	Red	Red	Blue	Blue	Red
State	d1		d2	d3	d4	

Let $\boldsymbol{I} = \{\text{Off}, \text{On}\} = \{0, 1\}$ be a set of input push button states, $\boldsymbol{D} = \{d_1, d_2, d_3, d_4\} = \{00, 01, 11, 10\}$ a set of signal states and $\boldsymbol{O} = \{\text{Blue}, \text{Yellow}, \text{Red}\} = \{00, 01, 11(10)\}$ a set of output signals. Here each binary numeral is a coding of each element in sets. Next we define the state transition function δ and the signal output function λ of Signal A based on Table 3 as follows:

Table 4: Binary coding of transition, output and Boolean function tables (Signal A)

δ		D			
		d1 (00)	d2 (01)	d3 (11)	d4 (10)
I	Off (0)	d1 (00)	d3 (11)	d4 (10)	d1 (00)
	On (1)	d2 (01)	d3 (11)	d4 (10)	d1 (00)

λ		D			
		d1 (00)	d2 (01)	d3 (11)	d4 (10)
I	Off (0)	Bl (00)	Ye (01)	Re (11)	Re (10)
	On (1)	Bl (00)	Ye (01)	Re (11)	Re (10)

From Table 4, we can define every output formulas B_1, B_2, D_1, D_2 by using input formulas C, A_1, A_2 as follows:
$B_1 = (\neg A_1 \wedge A_2 \wedge \neg C) \vee (A_1 \wedge A_2 \wedge \neg C) \vee (\neg A_1 \wedge A_2 \wedge C) \vee (A_1 \wedge A_2 \wedge C)$

	C	A1	A2		B1	B2		D1	D2	
Off	0	0	0	(d1)	0	0	(d1)	0	0	(Bl)
	0	0	1	(d2)	1	1	(d3)	0	1	(Ye)
	0	1	1	(d3)	1	0	(d4)	1	1	(Re)
	0	1	0	(d4)	0	0	(d1)	1	0	(Re)
On	1	0	0	(d1)	0	1	(d2)	0	0	(Bl)
	1	0	1	(d2)	1	1	(d3)	0	1	(Ye)
	1	1	1	(d3)	1	0	(d4)	1	1	(Re)
	1	1	0	(d4)	0	0	(d1)	1	0	(Re)

$= (\neg A_1 \wedge A_2) \vee (A_1 \wedge A_2) = A_2.$
$B_2 = (\neg A_1 \wedge A_2 \wedge \neg C) \vee (\neg A_1 \wedge \neg A_2 \wedge C) \vee (\neg A_1 \wedge A_2 \wedge C) = (\neg A_1 \wedge A_2 \wedge \neg C) \vee (\neg A_1 \wedge C)$
$D_1 = (A_1 \wedge A_2 \wedge \neg C) \vee (A_1 \wedge \neg A_2 \wedge \neg C) \vee (A_1 \wedge A_2 \wedge C) \vee (A_1 \wedge \neg A_2 \wedge C)$
$= (A_1 \wedge A_2) \vee (A_1 \wedge \neg A_2) = A_1$
$D_2 = (\neg A_1 \wedge A_2 \wedge \neg C) \vee (A_1 \wedge A_2 \wedge \neg C) \vee (\neg A_1 \wedge A_2 \wedge C) \vee (A_1 \wedge A_2 \wedge C)$
$= (\neg A_1 \wedge A_2) \vee (A_1 \wedge A_2) = A_2$

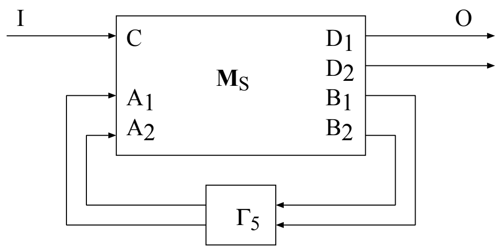

Figure 4: Sequential machine generated ftom Γ_5 (Signal A)

We define $\Gamma_5 = \{(A_1^0, A_2^1), (A_2^0, ((\neg A_1 \wedge A_2 \wedge \neg C) \vee (\neg A_1 \wedge C))^1)\}$ as a set of pair-sentence formulas. Since we have $X^0 = \{A_1, A_2\}$, $X^1 = \{A_2, (\neg A_1 \wedge A_2 \wedge \neg C) \vee (\neg A_1 \wedge C)\}$ and $X = Sub(X^1)/Sub(X^0) = \{C, \neg A_1, \neg C, \neg A_1 \wedge A_2, \ldots\}$, so $Prim(X) = \{C\}$. If we define $\delta_{\Gamma 5} : \boldsymbol{TV}_{\boldsymbol{\Gamma}5}^{i} \times \boldsymbol{I}_{\boldsymbol{\Gamma}5}^{i} \rightarrow \boldsymbol{TV}_{\boldsymbol{\Gamma}5}^{i+1}$, and let's $\boldsymbol{TV}_{\boldsymbol{\Gamma}5} = \{v(A_1) \times v(A_2); v : Sub(X^0) \cap Sub(X^1) \rightarrow \{0,1\}\} = \{v_1(d_3), v_2(d_4), v_3(d_2), v_4(d_1)\}$, $\boldsymbol{I}_{\boldsymbol{\Gamma}5} = \{v(C); v : Prim(X) \rightarrow \{1,0\}\} = \{i_1(\text{On}), i_2(\text{Off})\}$, $\boldsymbol{O} = \{v(D_1) \times v(D_2); v : \{D_1, D_2\} \rightarrow \{0,1\}\} = \{00(\text{Blue}), 01(\text{Yellow}), 11(\text{Red}), 10(\text{Red})\}$, $v^0 \in \boldsymbol{TV}_{\boldsymbol{\Gamma}5}$ and $\lambda_{\Gamma 5} : \boldsymbol{TV}_{\boldsymbol{\Gamma}5}^{i} \times \boldsymbol{I}_{\boldsymbol{\Gamma}5}^{i} \rightarrow \boldsymbol{O}^i$ then we get $\boldsymbol{M_S} = \langle \boldsymbol{TV}_{\boldsymbol{\Gamma}5}, \boldsymbol{I}_{\boldsymbol{\Gamma}5}, \boldsymbol{O}, v^0, \delta_{\Gamma 5}, \lambda_{\Gamma 5} \rangle$ as a sequential machine generated from Γ_5 (see Table 5).

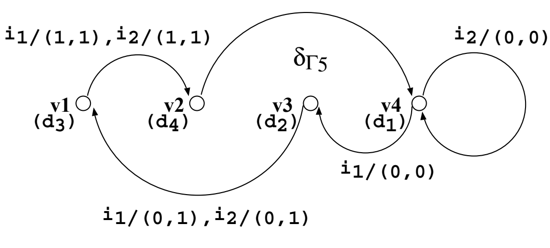

Figure 5: Boolean transition figure of $\delta_{\Gamma 5}$ (Signal A)

Table 5: Truth transition table of Γ_5 (Signal A)

$\boldsymbol{I}_{\boldsymbol{\Gamma}_5}^i$	C^i	$\boldsymbol{TV}_{\boldsymbol{\Gamma}_5}^i$	A_1^i	A_2^i	$A_1^{i+1} \stackrel{\text{def}}{=} A_2^i$	A_2^{i+1}	$\boldsymbol{TV}_{\boldsymbol{\Gamma}_5}^{i+1}$	D_1^i	D_2^i
i_1	1	v_1	1	1	1	0	v_2	1	1
	1	v_2	1	0	0	0	v_4	1	0
	1	v_3	0	1	1	1	v_1	0	1
	1	v_4	0	0	0	1	v_3	0	0
i_2	0	v_1	1	1	1	0	v_2	1	1
	0	v_2	1	0	0	0	v_4	1	0
	0	v_3	0	1	1	1	v_1	0	1
	0	v_4	0	0	0	0	v_4	0	0

where we assume that $A_2^{i+1} \stackrel{\text{def}}{=} (\neg A_1^i \wedge A_2^i \wedge \neg C^i) \vee (\neg A_1^i \wedge C^i)$ and every output formulas are $D_1^i \stackrel{\text{def}}{=} A_1^i$ and $D_2^i \stackrel{\text{def}}{=} A_2^i$.

4 3-valued Łukasiewicz logic on PSC

Let $\mathcal{L}_3 =< \boldsymbol{FOR_3}, \neg_3, \wedge_3, \vee_3, \rightarrow_3, \equiv_3, \perp, \mathrm{N}, \top >$ be a sentential language constructed from sentential variables, constants $\perp$(strictly false), $\top$(strictly true) and N(possible) by means of Łukasiewicz connectives: $\neg_3$ (Ł-negation), $\rightarrow_3$ (Ł-implication), $\wedge_3$ (Ł-conjunction), $\vee_3$ (Ł-disjunction) and $\equiv_3$ (Ł-equivalence). $\boldsymbol{FOR_3}$ is a set of all formulas of $\mathcal{L}_3$. Here each connective is defined by the following truth tables [10]:

Table 6: Truth table of each connective in $\mathcal{L}_3$

A	$\neg_3$A
0	1
1/2	1/2
1	0

$\rightarrow_3$	0	1/2	1
0	1	1	1
1/2	1/2	1	1
1	0	1/2	1

$\vee_3$	0	1/2	1
0	0	1/2	1
1/2	1/2	1/2	1
1	1	1	1

$\wedge_3$	0	1/2	1
0	0	0	0
1/2	0	1/2	1/2
1	0	1/2	1

$\equiv_3$	0	1/2	1
0	1	1/2	0
1/2	1/2	1	1/2
1	0	1/2	1

Where $\mathcal{A}_3 = \langle \boldsymbol{A_3}, \sim_3, \cap_3, \cup_3, \supset_3, \circ, \{0, 1/2, 1\}\rangle$ is the well-known Ł3-algebra and $\boldsymbol{D_3} = \{1\}$ a designated subset of $\boldsymbol{A_3}$, then the pair $\mathcal{M}_3 = \langle \mathcal{A}_3, \boldsymbol{D_3}\rangle$ is the Łukasiewicz's 3-valued model. For any formula $A \in \boldsymbol{FOR_3}$ is true in $\mathcal{M}_3$ model under the assumption of $X \subseteq \boldsymbol{FOR_3}$, $\mathcal{M}_3, X \models A$ in symbol, if for every assignment $h : \boldsymbol{FOR_3} \rightarrow \boldsymbol{A_3}$, $h(X) \subseteq \boldsymbol{D_3}$ implies $h(A) = 1$, where $h(\perp) = 0, h(\mathrm{N}) = 1/2$ and $h(\top) = 1$. Moreover,

$h(\neg_3 A) \stackrel{\text{def}}{=} (1 - h(A))$

$h(A \rightarrow_3 B) \stackrel{\text{def}}{=} min\{1, 1 - h(A) + h(B)\}$

$h(A \vee_3 B) \stackrel{\text{def}}{=} ((h(A) \rightarrow_3 h(B)) \rightarrow_3 h(B)) = max\{h(A), h(B)\}$

$h(A \wedge_3 B) \stackrel{\text{def}}{=} \neg_3(\neg_3 A \vee_3 \neg_3 B) = min\{h(A), h(B)\}$

To interpret the tautology set of $\mathcal{L}_3$ on **PSC** logic, we define the truth assignment $v_h : \boldsymbol{FOR_3} \rightarrow \{0, 1\}$ such that for any algebraic assignment $h : \boldsymbol{FOR_3} \rightarrow \boldsymbol{A_3}$, $v_h(A) \stackrel{\text{def}}{=} h(\neg_3(A \rightarrow_3 \neg_3 A))$.

This means that for any formula $A \in \boldsymbol{FOR_3}$, $v_h(A) = 1$ if $h(A) = 1$ and otherwise, $v_h(A) = 0$. Then we can define the Łukasiewicz's 3-valued logic on **PSC** as follows:
Let $\mathcal{L}_P =< \boldsymbol{FOR_P}, \neg, \wedge, \vee, \rightarrow, ((\_),\ (\_)), \top, \bot >$ be a **PSC** language, where

$\neg A \stackrel{\text{def}}{=} (A \rightarrow_3 \neg_3 A)$

$A \rightarrow B \stackrel{\text{def}}{=} (A \rightarrow_3 (A \rightarrow_3 B))$

$A \vee B \stackrel{\text{def}}{=} ((A \rightarrow_3 B) \rightarrow_3 B)$

$A \wedge B \stackrel{\text{def}}{=} \neg_3(\neg_3 A \vee_3 \neg_3 B)$

$(A, B) \stackrel{\text{def}}{=} ((A \rightarrow_3 B) \wedge_3 (B \rightarrow_3 A)) = (A \equiv_3 B)$

Every connectives $\{\neg, \wedge, \vee, \rightarrow, ((\_),\ (\_))\}$ are classical and have one of two truth values $(\top, \bot)$. Also in this case, a pair-sentence connective $((\_),\ (\_))$ is an identity, so $v_h((A, B)) = 1$ if h(A)=h(B) and otherwise, $v_h((A, B)) = 0$ (see Table 6 and 7).

Table 7: Truth table of each connective in $\mathcal{L}_P$

A	¬3A	¬A = (A→3 ¬3A)	
0	1	1	= min{1, 1-0+1}
1/2	1/2	1	= min{1, 1-1/2+1/2}
1	0	0	= min{1, 1-1+0}

A	B	A→3B	A→B = (A→3(A→3B))	A∨B = ((A→3B)→3B)
0	0	1	1	0
0	1/2	1	1	1/2 (0)
0	1	1	1	1
1/2	0	1/2	1	1/2 (0)
1/2	1/2	1	1	1/2 (0)
1/2	1	1	1	1
1	0	0	0	1
1	1/2	1/2	1/2 (0)	1
1	1	1	1	1

A	B	¬3A	¬3B	¬3A ∨3 ¬3B	A∧B = ¬3(¬3A ∨3 ¬3B)
0	0	1	1	1	0
0	1/2	1	1/2	1	0
0	1	1	0	1	0
1/2	0	1/2	1	1	0
1/2	1/2	1/2	1/2	1/2	1/2 (0)
1/2	1	1/2	0	1/2	1/2 (0)
1	0	0	1	1	0
1	1/2	0	1/2	1/2	1/2 (0)
1	1	0	0	0	1

Here if we interpret the Łukasiewicz's connectives $\{\neg_3, \rightarrow_3\}$ by $\neg_3 A \stackrel{\text{def}}{=} (A, \neg(A, A))$ and $A \rightarrow_3 B \stackrel{\text{def}}{=} (A \wedge B, A)$ on **PSC** logic, then we can get the 2-valued (classical) system **PSC** with $\Gamma_6 = \{(\neg_3 A, (A, \neg(A, A))), (A \rightarrow_3 B, (A \wedge B, A))\}$ as a set of pair-sentence formulas, and which has the same tautology set as Łukasiewicz's 3-valued logic (see Table 8 and Figure 6).

Table 8: Interpretation of each Łukasiewicz's connective in $\mathcal{L}_P$

A	$\neg_3$A	(A, A)	$\neg$(A, A)	(A, $\neg$(A, A))
0	1	1	0	1
1/2	1/2	1	0	(1/2, 0) (=1/2)
1	0	1	0	0

A	B	A$\rightarrow_3$B	A$\wedge$B	(A$\wedge$B, A)
0	0	1	0	1
0	1/2	1	0	1
0	1	1	0	1
1/2	0	1/2	0	(1/2, 0) (=1/2)
1/2	1/2	1	1/2	1
1/2	1	1	1/2	1
1	0	0	0	0
1	1/2	1/2	1/2	(1, 1/2) (=1/2)
1	1	1	1	1

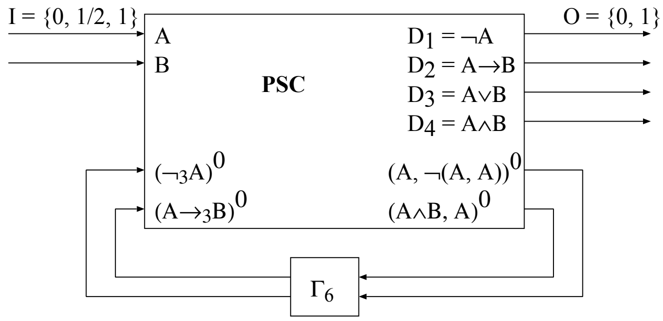

Figure 6: 2-valued (classical) system of Łukasiewicz's 3-valued logic on **PSC**

where $D_1 \stackrel{\text{def}}{=} (A \rightarrow_3 \neg_3 A)$, $D_2 \stackrel{\text{def}}{=} (A \rightarrow_3 (A \rightarrow_3 B))$, $D_3 \stackrel{\text{def}}{=} ((A \rightarrow_3 B) \rightarrow_3 B)$ and $D_4 \stackrel{\text{def}}{=} \neg_3(\neg_3 A \vee \neg_3 B)$.

5 Conclusion

In this paper we have defined a sequential machine $\boldsymbol{M_S} = \langle \boldsymbol{TV_\Gamma}, \boldsymbol{I_\Gamma}, \boldsymbol{O}, v^0, \delta_\Gamma, \lambda_\Gamma \rangle$ and also a finite automaton $\boldsymbol{M_A} = \langle \boldsymbol{TV_\Gamma}, \boldsymbol{I_\Gamma}, v^0, \delta_\Gamma, \boldsymbol{F} \rangle$ by using **PSC** system with Γ as a finite set of pair-sentence formulas. Here $\Gamma = \{(A_1^0, B_1^1), (A_2^0, B_2^1), \ldots, (A_m^0, B_m^1)\}$ is a finite set of pair-sentence formulas, $X^0 = \{A_1, A_2, \ldots, A_m\}$ and $X^1 = \{B_1, B_2, \ldots, B_m\}$ are sets of all formulas appeared in the first stage 0 and the second stage 1, respectively. Moreover, $\boldsymbol{TV_\Gamma} = \{\Pi_{j=1}^m v(A^j); A^j \in (Sub(X^0) \cap Sub(X^1))\}$ is a set of all truth value assignment products where $Sub(X^i)$ a set of all subformulas of each element of X^i $(i \in \{0, 1\})$, $\boldsymbol{I_\Gamma} = \{\Pi_{j=1}^k v(C^j); C^j \in Prim(X)\}$ a set of all input truth value assignment products where $X = Sub(X^1)/Sub(X^0)$ a subtraction of each $Sub(X^i)$ and $Prim(X)$ a set of all primitive formulas of X, $\boldsymbol{O} = \{\Pi_{j=1}^l v(D^j); D^j \in Y\}$ a set of all output truth value assignment products where Y a set of all output formulas, $v^0 \in \boldsymbol{TV_\Gamma}$ an initial truth value assignment product, $\delta_\Gamma : \boldsymbol{TV_\Gamma}^i \times \boldsymbol{I_\Gamma}^i \rightarrow \boldsymbol{TV_\Gamma}^{i+1}$ a truth value assignment transition function, $\lambda_\Gamma : \boldsymbol{TV_\Gamma}^i \times \boldsymbol{I_\Gamma}^i \rightarrow \boldsymbol{O}^i$ a truth value assignment output function and $\boldsymbol{F}$ a set of all final(accepted) truth value assignment.

We have also explained some concrete examples of Γ:

$\Gamma_1 = \{(A^0, A^1)\}$: The principle of identity "A is A"

$\Gamma_2 = \{(A^0, \neg A^1)\}$: A simple Liar sentence

$\Gamma_3 = \{(A^0, \neg B^1), (B^0, \neg C^1), (C^0, A^1)\}$: Dialogue for Socrates, Plato and Aristoteles

$\Gamma_4 = \{(C^0, (A \vee (B \wedge \neg C))^1)\}$: Circular definition in Gupta's book [3]

$\Gamma_5 = \{(B^0, C^1), (C^0, ((\neg A \wedge \neg B \wedge C) \vee (A \wedge \neg B))^1)\}$: Two type of traffic signals with a push button [9]. Finally we have constructed the 2-valued classical system of Łukasiewicz's 3-valued logic on **PSC** with $\Gamma_6 = \{(\neg_3 A, (A, \neg(A, A))), (A \to_3 B, (A \wedge B, A))\}$ as a set of pair-sentence formulas, and which has the same tautology set as Łukasiewicz's 3-valued logic.

References

[1] W. Burks, *Essays on Cellular Automata*, University of Illinois Press, Urbana, Chicago, London, 1970.

[2] S. L. Bloom and R. Suszko, Investigations into the sentential calculus with identity, *Notre Dame Journal of Formal Logic*,vol.XIII, No. 3, (1971), pp.289–308.

[3] A. Gupta and N. Belnap, *The Revision Theory of Truth*, MIT Press, Cambridge, 1993.

[4] M. A. Harrison, *Introduction to Switching Theory and Automata Theory*, McGraw-Hill, 1965.

[5] H. G. Herzberger, Naive semantics and the Liar paradox, *Journal of Philosophy 79*, 1982, pp.479–497.

[6] T. Ishii, A syntactical comparison between pair sentential calculus **PSC** and Gupta's definitional calculus $\mathbf{C}_n$, *Bulletin of NUIS*, Niigata University of International and Information Studies, 2016.

[7] T. Ishii, SCI for pair-sentence and its completeness, *Non-Classical Logics, Theory and Applications*, Vol. 8, 2016, pp.61–65.

[8] S. Kripke, Outline of a theory of truth, *Journal of Philosophy 72*, 1975, pp.690–716.

[9] U. Sousichi, *Sinnri-syoumei-keisann*, Minerva-shobo, Japan, 1989.

[10] R. Suszko, Remarks on Łukasiewicz's three-valued logic, *Bulletin of the Section of Logic*, University of Łódź, vol.4, Nr.3, 1975, pp.87–90.

[11] R. Suszko, Abolition of the Fregean axiom, *Logic Colloquium*, eds. by R. Parikh, Springer, Berlin, 1975, pp.169–239.

[12] A. Tarski, The concept of truth in formalized language, in *Logic, Semantics, Metamathematics*, trans. by J. H. Woodger, Oxford University Press, 1956, pp.152–278.