



HAL
open science

A Boolean algebra for a formal expression of events in logical systems

Bruno Denis, Jean-Jacques Lesage, Jean-Marc Roussel

► **To cite this version:**

Bruno Denis, Jean-Jacques Lesage, Jean-Marc Roussel. A Boolean algebra for a formal expression of events in logical systems. IMACS symposium on mathematical modelling, Feb 1994, Vienna, Austria. pp. 859-862. hal-03740536

HAL Id: hal-03740536

<https://hal.science/hal-03740536>

Submitted on 3 Aug 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Boolean algebra for a formal expression of events in logical systems

Bruno Denis ; Jean-Jacques Lesage ; Jean-Marc Roussel

Laboratoire Universitaire de Recherche en Production Automatisée
Ecole Normale Supérieure de Cachan
61, avenue du président Wilson
F-94235 Cachan cedex - France
e-mail : denis@lurpa.ens-cachan.fr

Abstract. The dynamic modelling of logical systems widely calls upon the event notion. In terms of Function Chart Grafcet or Petri Nets for instance, events are generally represented by “rising or falling edges” of logical variables. However, numerous ambiguities are encountered in the models because the translation of events into edges is not formal enough. In this paper, we propose a Boolean algebra the definition-set of which allows us to describe the time behavior of the inputs and the outputs of any logical system. In this algebra, we have defined two unary operations in order to formally express the events. Then, we are giving 14 properties related to these rising and falling edge operations and their composition with the operations AND, OR, and NOT.

Key Words. Boolean algebra, event, logical system, Grafcet, Interpreted Petri Nets.

1. INTRODUCTION

The notion of event is widely used in describing the dynamic behavior of logical systems, mostly when it is represented by combinatory equations, Grafcets or Interpreted Petri Nets (IPN). The use of events in the transition conditions of a graph introduces a wide variety of descriptions that turns out very useful for the representation of real time systems [15]. It actually allows for the conditioning of the clearing of transitions by the occurrence of an event (event - connected approach) and not only through the checking of a condition (condition-connected approach) [14]. Both these approaches are joint in the Interpreted Petri Nets if one associates a predicate and an event to each transition [10].

In this paper we will show that, even though the relevance of the notion of event cannot be questioned, the same cannot apply to the rigor of its definition or its use in the transition conditions of Grafcets or Interpreted Petri Nets. We will first demonstrate the limits and inaccuracies of the notion of “edges” and the interest of building an “extended” Boolean algebra so as to reach a more formal approach of events in transition conditions. This Boolean algebra will then be described, and its main properties given.

2. PROBLEMATICS OF THE TAKING INTO ACCOUNT OF EVENTS

From a theoretical point of view, the notion of event corresponds to a zero time spectrum information, i.e the information that translates the supposedly instantaneous change of state of a logical variable or of the function of logical variables [1], [7], [2]. The concept is then translated in the notation “rising edge : \uparrow ” or “falling edge : \downarrow ” of the variable or of the function of variables [11] (Fig. 1).

Yet in practice, the use of edges is often limited to the sole elementary logical variables as the evaluation of expressions that hold edges of functions of logical variables - as in equation (1) - is not mastered.

$$E_i = \uparrow(a \cdot \bar{c} + b) \cdot \downarrow(a \cdot b + c) \quad (1)$$

When the restrictive hypotheses of non-simultaneous events or of total independence between the a, b, c variables can be emitted, heuristics such as (2) or (3) may be used [4] :

$$\uparrow(a \cdot b) = \uparrow a \cdot b + a \cdot \uparrow b \quad (2)$$

$$\uparrow(a + b) = \uparrow a \cdot \bar{b} + \bar{a} \cdot \uparrow b \quad (3)$$

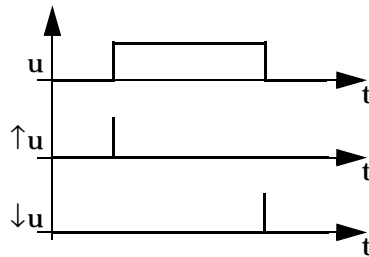


Fig. 1 Rising edge and falling edge of a boolean variable.

One only has to examine such an expression as (1) to identify the cause of the difficulties met in evaluating such logical equations. In fact, in this expression, “+” and “.” represent the two operations of composition of Boole’s Algebra and “-”, the unary complementary operation ; these operations are thus perfectly defined by their truth table. However, “↑” is only a mere notation that shows that the designer of this expression is interested in the change of state of the function (a · c + b) and not in its logical level “1”. In [3] the authors actually stress that grafcet transition conditions that hold edges of variables are not defined according to Boole’s Algebra.

So as to develop and evaluate such expressions, whatever the number of variables that make up the logical functions, and taking into account the possibility of distinct simultaneous events, we now wish to build an “extended” Boolean algebra that holds two event-connected unary operations : the “rising edge” and the “failing edge”. Thus equipped with an algebraic definition of edges, we will establish a set of properties that allows for the development and the evaluation of expressions such as (1).

3. BUILDING OF AN “EVENT-CONNECTED” BOOLEAN

3.1 Set of definition

The set of definition for the researched algebra must :

- faithfully represent the inputs and outputs of any logical system,
- allow for the temporal taking into account of events,
- hold at least two elements,
- be closed under all the operations defined on it.

Taking these four criteria into consideration, we have retained and will note \mathbb{I} the set of functions defined on \mathbb{R}^{+*} , whose range is $\mathbb{B} = \{0, 1\}$, that verify the following property.

$$\mathbb{I} = \{u : \mathbb{R}^{+*} \rightarrow \mathbb{B} \mid \forall t \in \mathbb{R}^{+*} : (\exists \varepsilon_t > 0 : (\forall (\varepsilon_1, \varepsilon_2) \in]0, \varepsilon_t[{}^2, u(t - \varepsilon_1) = u(t - \varepsilon_2)))\}$$

By definition, all u functions of \mathbb{I} are then piecewise continuous and can admit a double discontinuity at certain points. The general shape of a function of the set \mathbb{I} is represented in Fig. 2.

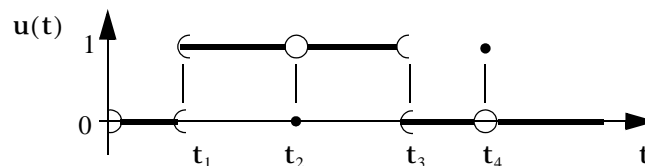


Fig. 2 Example of an element function of the set \mathbb{I} .

At the points of discontinuity, the problem of the value of the function is posed. The function can actually be considered as right-continuous or left-continuous. As J.P. Frachet, we will hold for the right - continuity, as it is more natural to the physicist, it being causal [5]. At the date of occurrence of an event, we will therefore consider that the function has already changed its value.

We admit the existence of points that present a double discontinuity ($u(t_2) = 0 ; u(t_4) = 1$) so as to ensure that \mathbb{I} is closed under the edge operations.

It may be stressed here that such a definition of the elements of \mathbb{I} is perfectly in conformity with the practice of automaticians who frequently represent the time evolution of boolean values as timing diagrams.

3.2 Convention of notation

To make the reading of this paper easier, and to avoid all possible mixing of the operations on the elements of \mathbb{I} (function $u : \mathbb{R}^{+*} \rightarrow \mathbb{B}$) and the booleans (values taken by these functions at a given time), we will from then on note “ \wedge ” the logical operation AND, “ \vee ” the logical operation OR, “ \neg ” the NOT operation on a boolean. The notations “ \cdot ”, “ $+$ ”, et “ $\bar{\cdot}$ ” will be dedicated to the operations on \mathbb{I} .

Furthermore, we have carefully distinguished the function from the boolean, i.e the value taken at a given time by this function. For instance, u, v, w are three functions element of \mathbb{I} while $u(t), v(t), w(t)$ are three booleans.

3.3 Definition of operations on \mathbb{I}

After having explained our set of definition and precised the notations, we can defined the following operations on \mathbb{I} .

the **AND** operation

$$\mathbb{I}^2 \rightarrow \mathbb{I} \quad \text{with} \quad \forall t \in \mathbb{R}^{+*}, (u \cdot v)(t) = u(t) \wedge v(t)$$

the **OR** operation

$$\mathbb{I}^2 \rightarrow \mathbb{I} \quad \text{with} \quad \forall t \in \mathbb{R}^{+*}, (u + v)(t) = u(t) \vee v(t)$$

the **NOT** operation

$$\mathbb{I} \rightarrow \mathbb{I} \\ u \rightarrow \bar{u} \quad \text{with} \quad \forall t \in \mathbb{R}^{+*}, \bar{u}(t) = \neg u(t)$$

By definition, \mathbb{I} is closed under these operations and \mathbb{I} is a boolean algebra [6] [8] [9].

3.4 Taking event into account in this algebra

All the interest of this algebra resides in the fact that we can now strictly define two extra unary operations to formally express the notion of event.

the **RE** operation (rising edge)

$$\mathbb{I} \rightarrow \mathbb{I} \\ \rightarrow \uparrow u \quad \text{with} \quad \forall t \in \mathbb{R}^{+*}, \uparrow u(t) = u(t) \wedge (\exists \varepsilon_0 > 0 : \forall \varepsilon \in]0, \varepsilon_0[, u(t - \varepsilon) = 0)$$

the **FE** operation (falling edge)

$$\mathbb{I} \rightarrow \mathbb{I} \\ \rightarrow \downarrow u \quad \text{with} \quad \forall t \in \mathbb{R}^{+*}, \downarrow u(t) = \bar{u}(t) \wedge (\exists \varepsilon_0 > 0 : \forall \varepsilon \in]0, \varepsilon_0[, u(t - \varepsilon) = 1)$$

The images of t under the function $\uparrow u$ (respectively $\downarrow u$) are thus determined at all moments as the logical AND between two booleans. The first boolean is the value of the u -function (respectively the complement of the value) at that moment, whereas the second boolean is the value of a predicate at the same moment. The truthfulness of the predicate depends on the value taken by the function u on the interval $]t - \varepsilon_0, t[$.

For the function represented on Fig. 2 for instance :

- $\uparrow u(t) = 1$ if $t \in [t_1, t_2[\cup]t_2, t_3[\cup \{t_4\}$ and if $t \in]0, t_1[\cup]t_3, t_4[\cup]t_4, \infty[$ i.e. $t = t_1$ or $t = t_4$.
- $\downarrow u(t) = 1$ si $t \in]0, t_1[\cup \{t_2\} \cup [t_3, t_4[\cup]t_4, \infty[$ and if $t \in]t_1, t_2[\cup]t_2, t_3[$ i.e. $t = t_2$ or $t = t_3$.

By definition, \mathbb{I} is closed under these two operations as the functions $\uparrow u$ and $\downarrow u$ are defined in \mathbb{R}^{+*} , with boolean values and verify the property of the elements of \mathbb{I} developed in paragraph 3.1.

3.5 FUNDAMENTAL PROPERTIES

The following properties ((4) to (17)) have been demonstrated on the set \mathbb{II} (these proofs as well as an example have been developed in [12] :

$$\mathbf{u} + \uparrow \mathbf{u} = \mathbf{u} \quad \mathbf{(4)} \qquad \mathbf{u} \cdot \uparrow \mathbf{u} = \uparrow \mathbf{u} \quad \mathbf{(5)} \qquad \uparrow \bar{\mathbf{u}} = \downarrow \mathbf{u} \quad \mathbf{(6)}$$

$$\bar{\mathbf{u}} + \downarrow \mathbf{u} = \bar{\mathbf{u}} \quad \mathbf{(7)} \qquad \bar{\mathbf{u}} \cdot \downarrow \mathbf{u} = \downarrow \mathbf{u} \quad \mathbf{(8)} \qquad \downarrow \bar{\mathbf{u}} = \uparrow \mathbf{u} \quad \mathbf{(9)}$$

$$\uparrow(\uparrow \mathbf{u}) = \uparrow \mathbf{u} \quad \mathbf{(10)} \qquad \uparrow(\downarrow \mathbf{u}) = \downarrow \mathbf{u} \quad \mathbf{(11)} \qquad \downarrow(\uparrow \mathbf{u}) = 0^* \quad \mathbf{(12)} \qquad \downarrow(\downarrow \mathbf{u}) = 0^* \quad \mathbf{(13)}$$

$$\uparrow \left(\prod_{i=1}^n \mathbf{u}_i \right) = \sum_{i=1}^n \left(\uparrow \mathbf{u}_i \cdot \prod_{(j=1), (j \neq i)}^n \mathbf{u}_j \right) \quad \mathbf{(14)} \qquad \downarrow \left(\prod_{i=1}^n \mathbf{u}_i \right) = \sum_{i=1}^n \left(\downarrow \mathbf{u}_i \cdot \prod_{(j=1), (j \neq i)}^n \downarrow \mathbf{u}_j + (\mathbf{u}_j \cdot \uparrow \bar{\mathbf{u}}_j) \right) \quad \mathbf{(15)}$$

$$\downarrow \left(\sum_{i=1}^n \mathbf{u}_i \right) = \sum_{i=1}^n \left(\downarrow \mathbf{u}_i \cdot \prod_{(j=1), (j \neq i)}^n \bar{\mathbf{u}}_j \right) \quad \mathbf{(16)} \qquad \uparrow \left(\sum_{i=1}^n \mathbf{u}_i \right) = \sum_{i=1}^n \left(\uparrow \mathbf{u}_i \cdot \prod_{(j=1), (j \neq i)}^n \uparrow \mathbf{u}_j + (\bar{\mathbf{u}}_j \cdot \downarrow \bar{\mathbf{u}}_j) \right) \quad \mathbf{(17)}$$

4. CONCLUSION

In this paper we have presented the results of a theoretical work that aims at making up for the lack of a formal definition of the notion of event as it is practiced in Grafcet. We have therefore built up an algebra on a set \mathbb{II} of functions defined in \mathbb{R}^{+*} with boolean values. In this algebra, two operations have been defined for the formal definition of events : the rising edge operation and the falling edge operation. Fourteen properties have then been given in relation with the edge operations and their combinations. This work, though it has an inner finality, is actually part of a global project that aims at the analysis of the coherence and of the dynamic behavior of complex systems. The Boole's algebra that we have presented has actually allowed us to design a module of formal calculation and of simplification of combinatory expressions. This module of formal calculation is itself used for the analysis of the dynamics of grafcets and the automatic generation of the equivalent automaton (AGGLAE Project of the LURPA [13]).

5. REFERENCES

- [1] French AFCET working group, Pour une représentation normalisée du cahier des charges d'un automatisme logique. AII, 61 (1977) 27-32 & 62 (1977) 36-40, Dunod (Ed.) France.
- [2] Blanchard M., Comprendre maîtriser et appliquer le GRAFCET. Cépaduès (Ed.), Toulouse-France, 1979.
- [3] Bouteille N. & al., Le GRAFCET. Cépaduès (Ed.), Toulouse-France, 1992.
- [4] David R. & Alla H., Petri Nets and Grafcet tools for modelling discrete event systems. Prentice Hall (Ed.), London, 1992.
- [5] Frachet J.P. & Colombari G., Elements for a semantics of the time in GRAFCET and dynamic systems using non-standard analysis. APII Hermès (Ed.), 27-1 (1993), 107-125.
- [6] Garding L. & Tambour T., Algebra for Computer Science. Springer Verlag (Ed.) New-York, 1988.
- [7] GREPA, Le GRAFCET de nouveaux concepts. Cépaduès (Ed.), Toulouse-France, 1985.
- [8] Halmos P., Lectures on Boolean Algebras. Van Nostrand Mathematical Studies (Ed.), New-York, 1967.
- [9] MacLane S. & Birkhoff G., Algebra. The MacMillan Compagny (Ed.), London, 1967.
- [10] Moalla M. & al., Synchronized Petri Nets : A model for the description of non-autonomous systems. Mathematical Foundations of Computer Sciences, Springer Verlag (Ed.), (1978) 374-383.
- [11] IEC 848 Standard, Preparation of function charts for control systems. 1988.
- [12] Roussel J.M. & Lesage J.J., Une algèbre de Boole pour l'approche événementielle des systèmes logiques. To appear in : APII Hermès (Ed.), 27-5 (1993).
- [13] Roussel J.M., Analyse de grafcets par génération logique de l'automate équivalent. PhD thesis of E.N.S. de Cachan-France , To appear, 1994.
- [14] Sayat B. & Ladet P., Control specification of a production system using GRAFCET and Petri nets. APII Hermès (Ed.), 27-1 (1993) 53-64.
- [15] Zahnd J., Machines séquentielles. Dunod (Ed.), Paris, 1987.