HEWLETT PACKARD

A Generalized Event Structure for the Muller Unfolding of a Safe Net

Jeremy Gunawardena
External Research Program
HPL-94-42
April, 1994

jhcg@hplb.hpl.hp.com

asynchronous circuit, event structure, logics of causality, unfolding

In 1959, Muller and Bartky published a celebrated paper on *"A Theory of Asynchronous Circuits"*. Among many novel techniques in that paper was the use of lattices resembling the domains of configurations of event structures. In the light of this we present a generalization of Muller's construction to safe nets. We find, however, that this "Muller unfolding" cannot be generated as the domain of configurations of any known event structure, not even a General Event structure. (In particular, this unfolding differs from that of Nielsen, Plotkin and Winskel.) This paper attempts to fill that gap. We make use of the logical approach to causality, developed in previous work, in which a General Event Structure is interpreted as a logical automation arising from a particular logic of causality. We introduce a new causal logic and associate a corresponding logical automaton to any finite safe Petri net. Our main result is that the domain of configurations of this generalized event structure is isomorphic to the Muller unfolding of the net. The work described here was done as part of project STETSON, a joint project between HP Labs and Stanford University on asynchronous hardware design.

Internal Accession Date Only
Published in Eike Best (ed), *"CONCUR'93"*, Proceedings of the *4th International Conference on Concurrency Theory*, Hildesheim, Germany, August 1993, Springer LNCS 715, 278-292, Springer-Verlag 1993

1 Introduction

In 1959, Muller and Bartky published *"A Theory of Asynchronous Circuits"* in which they used lattice theory to study the behaviour of clock-free digital circuits, [9]. This paper is remarkable not only for its analysis of a difficult real-world problem but also for the introduction of concepts and methods which were far in advance of their time. In particular, the lattices used by Muller and Bartky are closely related to the domains of configurations of event structures, although, of course, they were not recognized as such at the time.

In this paper we define and study an analogous construction to Muller's but in a context which is more familiar to concurrency theory: 1-safe Petri nets. The intuition behind Muller's lattice construction is that the elements of the lattice not only represent states of the corresponding circuit but also count the number of times a given wire in the circuit has experienced a change of voltage level (from 0 to 1 or vice versa). This idea is quite natural from the perspective of an electronic engineer. An oscilloscope probe placed in a circuit will faithfully record the rises and falls in the voltage level on a single wire and it is straightforward for an electronic observer to count these. Our construction for nets, builds a poset, which no longer has to be a lattice, whose elements count the number of times each transition in the net fires. We refer to this poset as the Muller unfolding of the net.

Similar constructions have recently appeared in the literature on asynchronous circuit design, [16]. We hope that the treatment we present here will clarify some of the difficulties with these constructions, which we discuss further in §2.

The idea of counting firings of transitions appears to be new to concurrency theory. For example, Nielsen, Plotkin and Winskel, [10], also construct an unfolding of a safe net and we observe in §2 that this does not coincide with the Muller construction. Indeed, we shall point out in §2 that not even a General Event Structure, of the type considered by Winskel in [14], is capable of generating the Muller poset. This leads us to ask whether there is some generalized event structure which is able to capture it. That is the problem which we address in this paper. In earlier work, [4, 5], we developed a logical approach to causality which allows us to interpret Winskel's General Event Structures as arising from a specific logic of causality, [4, Theorem 4.1]. By choosing a different logic, we can build a class of generalized event structures. In this paper we suggest an appropriate logic, $\mathcal{L}_3$, which we show to be sufficient to build a generalized event structure—a so-called $\mathcal{L}_3$-automaton—which captures the Muller poset. This is our main result.

We believe this paper has two main contributions. Firstly, it introduces to concurrency theory some ideas arising from practical problems in circuit design which have not been discussed and studied before in an abstract setting. Secondly, we have argued elsewhere, [4, 3], that causality still presents some difficult unresovled questions and that we are far from a definitive understanding of it. The Muller unfolding presents us with a family of awkward examples which force us towards a deeper understanding of causality. We believe this interplay between engineering practice and mathematical theory is important for the health of both subjects.

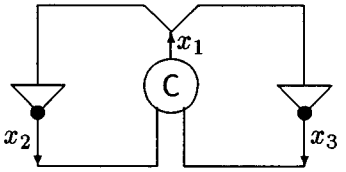
In the next section we consider Muller's construction in the context of a simple electrical circuit and use this to motivate our construction for Petri nets. In §3 we give a background sketch of the logical approach to causality developed in [4, 5] in sufficient detail to motivate our candidate logic of causality, $\mathcal{L}_3$. Finally, in §4, we explain how to build an $\mathcal{L}_3$-automaton from a safe net and give a sketch of the proof of the main result. Full details can be found in [6].

This paper arose out of questions posed by Vadim Kotov and Lucy Cherkasova during a visit by the author to the Institute of Informatics Systems in Novosibirsk in 1991. The author

gratefully acknowledges many discussions, then and subsequently, which laid the foundations of the present paper. Thanks are due to Alex Yakovlev for introducing the author to asynchronous circuits and for stimulating a detailed study of Muller's ideas. Mogens Nielsen has helped to develop the subject of Muller unfoldings and the author is grateful to him for many insightful discussions on that subject. Finally, thanks are due to four anonymous referees whose comments led to improvements in the presentation of this work. The work described here was undertaken as part of project STETSON, a joint project between Hewlett-Packard Laboratories and Stanford University on asynchronous circuit design.

2 The Muller unfolding

The diagram below shows a closed asynchronous circuit composed of two inverters and a Muller C-element. This is identical to Muller's Figure 2, [9, page 225].



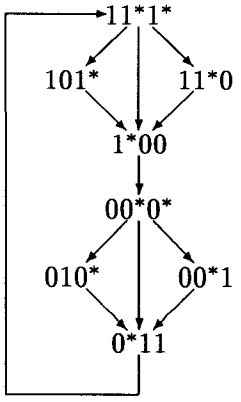
(1)

The behaviour of this circuit is given by the following Boolean circuit equations, [9, 7.8], in terms of the wires x_1, x_2 and x_3.

$$\begin{aligned} x_1' &= x_2x_3 \vee x_1x_2 \vee x_1x_3 \\ x_2' &= \neg x_1 \\ x_3' &= \neg x_1 \end{aligned}$$

For given voltage levels on the wires x_1, x_2 and x_3, these equations describe the new voltage levels, x_1', x_2', x_3', caused by the circuit components. Muller's famous C-element remembers its previous value until its inputs have both gone high or both gone low, at which point it changes value. C-elements are frequently used in asynchronous designs.

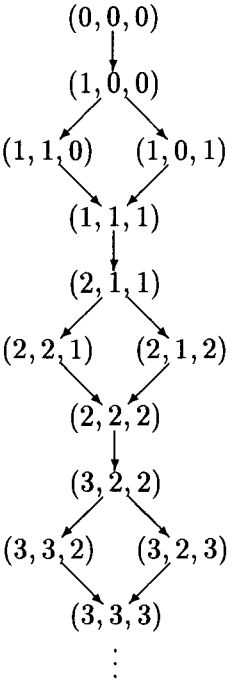
The behaviour of this circuit can be represented by the transition diagram (2). The states of this diagram are triples of Boolean values corresponding to the voltage levels on x_1, x_2 and x_3, respectively. An asterisk superscript indicates that the corresponding Boolean value is unstable and the circuit equations are tending to change that value. The transitions between the states are given by Muller's rule [9, (3.1)], which is usually referred to nowadays as the General Multiple Winner model, [1, §2.5].



(2)

(For a detailed discussion of the significance of the vertical transitions, from 111 to 100 and from 000 to 011, see [6].)

Muller's construction of the lattice of "cumulative states", [9, §7], is based on the idea of counting the number of times the voltage on each wire in the circuit changes value. A cumulative state is a triple of non-negative integers, (i, j, k), corresponding to the counts for the wires x_1, x_2 and x_3, respectively. The partial order is induced from the transitions in (2). For the circuit above we shall follow Muller's example and start counting from 011. Muller's cumulative diagram—actually the Hasse diagram of the partial order—then looks as follows, [9, Figure 3].



Although we have not formally defined Muller's construction we hope that the sketch given above will give some intuition for it. What we want to do in the remainder of this section is to take Muller's idea and look at it in the more familiar context of Petri nets. We first recall some notation and definitions.

A^* will denote the set of (finite) strings of symbols from A while A^+ will denote the multisets on A considered as functions from A to the natural numbers, $\mathbf{N}$. If $l, m \in A^+$ then $(l \pm m)(a) = l(a) \pm m(a)$ (provided $l - m$ is well-defined) and $(nl)(a) = n(l(a))$, for $n \in \mathbf{N}$. If $s \in A^*$, $[s] \in A^+$ denotes the corresponding multiset or Parikh vector, [11, Definition 13]. Note that $[-]$ is additive: $[st] = [s] + [t]$.

A labelled transition system (LTS) is a quadruple, $L = (S_L, A_L, R_L, i_L)$, where S_L is a set of states, A_L a set of actions, $R_L \subseteq S_L \times A_L \times S_L$ a set of transitions and $i_L \in S_L$ an initial state. Similarly, a transition system (TS) is a triple $U = (S_U, R_U, i_U)$. (We shall drop the suffixes when the context disambiguates the reference.) We adopt the usual notation for transitions: $p \xrightarrow{a} q$ for $(p, a, q) \in R_L$ and $p \stackrel{s}{\Rightarrow} q$ for the "transitive closure" labelled by strings $s \in A^*$. Similarly for the unlabelled versions, $p \to q$ and $p \Rightarrow q$. Every LTS, L, has an underlying TS where $p \to q$ if, and only if, $p \xrightarrow{a} q$ for some $a \in S_L$. A TS is acyclic if $\Rightarrow$ is anti-symmetric; $\Rightarrow$ is then a partial order. If U is an acyclic TS then $\wp(U) = (S_U, \Rightarrow)$ denotes the associated poset. If L is an LTS, its connected part, denoted L_c, is defined as L restricted to those states which are reachable from i_L. Similarly for a transition system. The traces of L

are $traces(L) = \{s \in A_L^* \mid i \stackrel{s}{\Rightarrow} p \text{ for some } p \in S_L\}$.

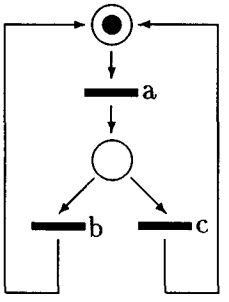
Definition 2.1 *A Petri net, $N = (B, A, F, M_0)$, is a quadruple where B is a set of places, A is a set of transitions, $F \subseteq (A \times B) \cup (B \times A)$ is a flow relation and $M_0 \in B^+$ is an initial marking.*

A marking is any element of B^+. If $x \in B \cup A$ then the preset of x, ${}^\bullet x$, and the post-set of x, $x^\bullet$, are defined by ${}^\bullet x = \{y \in B \cup A \mid (y, x) \in F\}$, $x^\bullet = \{y \in B \cup A \mid (x, y) \in F\}$.

Definition 2.2 *Let $N = (B, A, F, M_0)$ be a Petri net. Its associated LTS, denoted $\mathsf{LTS}(N)$, is defined as follows:*

- $S_{\mathsf{LTS}(N)} = B^+$;
- $A_{\mathsf{LTS}(N)} = A$;
- $R_{\mathsf{LTS}(N)} = \{(M_1, a, M_2) \mid {}^\bullet a \leq M_1,\ M_2 = M_1 - {}^\bullet a + a^\bullet\}$;
- $i_{\mathsf{LTS}(N)} = M_0$.

A net is 1-safe (or safe) if M_0 and each marking reachable from M_0 is set like: $M(b) \leq 1$, $\forall b \in B$. The traces of N are just the traces of the corresponding LTS: $traces(N) = traces(\mathsf{LTS}(N))$. The nets we deal with in this paper will all be finite (ie: both B and A will be finite) and 1-safe. The finite 1-safe net shown below will be our running example.



(3)

Muller counted changes in voltage level. We intend to count firings of transitions. The easiest way to count the transitions on a trace s is to use the multiset $[s]$ which can be thought of as a vector of numbers indexed by the names of the transitions. This is similar to Muller's cumulative state. Let $s \leq t$ denote the usual prefix ordering on strings and recall that $[s] = [t]$ if, and only if, s and t are permutations of each other.

Definition 2.3 *If $s, t \in traces(N)$ then $[s] \preceq [t]$ if, and only if, there are $s', t' \in traces(N)$ such that, $s' \leq t'$, $[s] = [s']$ and $[t] = [t']$.*

This is the *prefix ordering up to permutation* which was first introduced in [5, §2.2]. It was pointed out there that, in general, $\preceq$ is not a partial order.

Lemma 2.1 *$\preceq$ defines a partial order on the multisets of traces of N.*

Proof: It is clear that $[s] \preceq [s]$. Suppose that $[s] \preceq [t]$ and $[t] \preceq [s]$. Then there are $s' \leq t'$ with $[s'] = [s]$ and $[t'] = [t]$. There are also $t'' \leq s''$ with $[s''] = [s]$ and $[t''] = [t]$. Since the lengths of s, s' and s'' are all the same and similarly the lengths of t, t' and t'' are all the same, it must follow that s and t are of the same length. But the same must then apply to s' and t' and so $s' = t'$. Hence, $[s] = [t]$, as required.

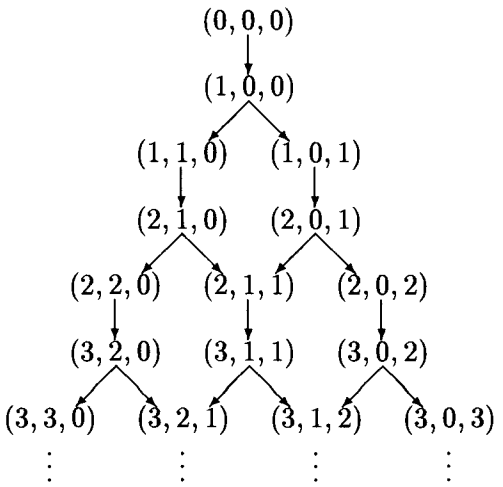
Now suppose that $[s] \preceq [t]$ and $[t] \preceq [u]$. We then have $s', t', t'', u' \in traces(N)$ such that $s' \leq t'$, $[s'] = [s]$, $[t'] = [t]$ and $t'' \leq u'$, $[t''] = [t]$, $[u'] = [u]$. At this point we have to use the fact that our traces are coming from a net. Let us suppose that $M_0 \stackrel{t'}{\Rightarrow} M'$ and $M_0 \stackrel{t''}{\Rightarrow} M''$. Since $[t''] = [t']$ it follows from the firing rule for nets that $M' = M''$: the order in which transitions are fired makes no difference to the eventual marking. Since $t'' \leq u'$ we can find a string v such that $u' = t''v$. It is then clear that v is a firing sequence from the marking M'. Hence, $t'v \in traces(N)$. Since $[t'] = [t'']$ it follows from the additivity of $[-]$ that $[t'v] = [t''v] = [u']$. So we have $s', t'v \in traces(N)$ such that $s' \leq t' \leq t'v$, $[s'] = [s]$, $[t'v] = [u'] = [u]$. Hence $[s] \preceq [u]$ and $\preceq$ is a partial order.

QED

Definition 2.4 *If $N = (B, A, F, M_0)$ is a finite 1-safe Petri net, the Muller unfolding of N is the poset $mul(N) = (X, \preceq)$ where $X = \{[s] \in A^+ \mid s \in traces(N)\}$ and $\preceq$ is the prefix ordering up to permutation.*

Landweber and Robertson use similar methods to associate a Parikh space to an arbitrary net, [8, §3], and consider the poset structure imposed by the "natural", or pointwise, order on multisets: $l \leq m$ if, and only if, $l(a) \leq m(a)$ for all $a \in A$. It is clear that this poset structure can only reflect the dynamics of the net in restricted cases. If the net is persistent then the Parikh space is a lattice, [8, Theorem 3.1], in fact, a semi-modular lattice, [15, Theorem 3.1]. If N is safe and persistent it is not hard to see that $mul(N)$ is isomorphic as a poset to the Parikh space of N. However, the poset strucutre of $mul(N)$ is based on the underlying behaviour of the net and not on the "un-natural" ordering on multisets which has, in general, absolutely no connection with the net. More recently, a Muller-style cumulative diagram for an arbitrary labelled transition system has appeared in [16, §2.4]. Unfortunately, this does not define a partial order. We give a correct construction in the full version of this paper, [6, Definition 2.1]. A cumulative diagram for a finite (not necessarily safe) Petri net is also defined in [16, §4.1.1] by different methods. This construction agrees with $mul(N)$ when N is safe but is less convenient to use. The work of [16] attempts to apply net theory to asynchronous circuit design, a difficult undertaking for reasons which are discussed further in [6]. Our agenda here is the converse, we seek to apply Muller's ideas on analysing asynchronous circuits to net theory

and, in particular, to understand the implications for event structures and causality.



(4)

The Hasse diagram of the Muller unfolding of example (3) is shown in (4). Since N has only 3 transitions, a, b and c, we have used the simplified notation (i,j,k) to denote the multiset l for which $l(a) = i$, $l(b) = j$ and $l(c) = k$. We can make some simple deductions from this example. The states $(1,1,0)$ and $(1,0,1)$ both cover the state $(1,0,0)$. But the least upper bound of $(1,1,0)$ and $(1,0,1)$ is clearly $(2,1,1)$ which does not cover either $(1,1,0)$ or $(1,0,1)$. This means that $mul(N)$ cannot be an event domain, [2], for axiom C would be violated. Hence, there can be no General Event Structure whose domain of configurations (or, at least, the compact elements thereof) is isomorphic to $mul(N)$. This implies in particular that the Nielsen, Plotkin, Winskel unfolding in [10] is different from the Muller unfolding.

This brings us to the main problem of this paper. Can we find a generalized event structure which gives the Muller unfolding as its domain of configurations?

3 A logical approach to causality

In this section we give a background sketch of the logical approach to causality. Our main purpose is to give some intuition for $\mathcal{L}_3$ and to show that it can be rigorously defined. The basis of the logical approach is that causality should be thought of as an observation on the state of a system. The observation indicates whether or not an event may occur. A logic of causality is then an appropriate language in which such observations can be stated. Following Winskel, we regard the state of a system as a configuration: *"a set of events which have occurred by some stage in a process"*, [14, §1.1], Furthermore, we interpret the principle of finite causes in a strong form as a statement of completeness for a logic of causality: any two observations which agree on all finite states are equivalent.

The mathematical ingredients are as follows. Let E be a set of events and let $Fin(E)$ denote the set of finite subsets of E. We seek logics $\mathcal{L}(E)$ which are equipped with a pairing $\models \subseteq \mathcal{L}(E) \times Fin(E)$ between formulae in the logic and states of the system. We shall write this as $s \models \rho$ for $s \in Fin(E)$ and $\rho \in \mathcal{L}(E)$. This pairing tells us when an observation ρ holds on the state s. Given such a pairing, an $\mathcal{L}$-automaton, $G = (E, \rho)$, is a pair consisting of a set of events E and a function $\rho : E \to \mathcal{L}(E)$ which associates to each event its cause, considered as an observation in the logic. The behaviour of an $\mathcal{L}$-automaton is described by the following "inference rule":

$$\frac{s \models \rho(e)}{s \xrightarrow{e} s \cup \{e\}} \quad (e \notin s), \tag{5}$$

which gives rise to a labelled transition system on the states of the automaton.

Definition 3.1 *If $G = (E, \rho)$ is an $\mathcal{L}$-automaton, its associated labelled transition system, denoted $\mathsf{LTS}(G)$, is defined as follows:*

- $S_{\mathsf{LTS(G)}} = Fin(E)$;
- $A_{\mathsf{LTS(G)}} = E$;
- $R_{\mathsf{LTS(G)}} = \{s \xrightarrow{e} s \cup \{e\} \mid e \notin s,\ s \models \rho(e)\}$;
- $i_{\mathsf{LTS(G)}} = \emptyset$.

Let $\mathsf{TS}(G)$ denote the underlying transition system of $\mathsf{LTS}(G)$. Since each transition increases the size of the state by one event, it is clear that $\mathsf{TS}(G)$ is acyclic.

Definition 3.2 *If $G = (E, \rho)$ is an $\mathcal{L}$-automaton, its domain of configurations, denoted $\wp(G)$, is defined as $\wp(G) = \wp(\mathsf{TS}(G)_c)$.*

As is customary, we consider only those states which are reachable from the initial state, $\emptyset$. Strictly speaking the domain of configurations as defined here includes only the compact elements. We can recover the full domain by taking a directed completion, [13, §9], if necessary, but we shall work only with the compact elements in this paper. The reader should note that the full domains are not Scott domains, in general.

These constructions are parametric in the choice of logic. Given any logic for which a notion of observation pairing has been defined, we get a corresponding class of automata whose behaviour is given uniformly by (5).

The remainder of this section leads up to the definition of $\mathcal{L}_3$. We shall first define $\mathcal{L}_1$ and thereby explain the close relationship between logics of causality and topology, which is an essential feature of our approach. We shall then mention $\mathcal{L}_2$, and state its relationship to Winskel's General Event Structures. This should make clear the sense in which $\mathcal{L}$-automata are generalized event structures. Finally, we shall use topological methods to define $\mathcal{L}_3$.

For any logic $\mathcal{L}$ we can define the function $\theta : \mathcal{L}(E) \to 2^{Fin(E)}$ where

$$\theta(\rho) = \{s \in Fin(E) \mid s \models \rho\}. \tag{6}$$

By the principle of finite causes, if $\theta(\rho_1) = \theta(\rho_2)$, then we should regard ρ_1 and ρ_2 as indistinguishable observations. This indicates that the logics we want to find can be regarded as collections of subsets of $Fin(E)$.

The simplest observation that we can make is that a given event, $e \in E$, has occurred. (By "occur" we mean that $e \in s$, where s is the current state of the system.) We shall denote this observation by the same symbol, e. It is characterised by the rule $s \models e$ if, and only if, $e \in s$. It follows that $\rho(e) = \{s \mid e \in s\}$. If $s \in Fin(E)$ let $s{\uparrow} = \{t \in Fin(E) \mid s \subseteq t\}$. We can then write $\rho(e) = \{e\}{\uparrow}$. A subset $x \subseteq Fin(E)$ is upwards closed if $s{\uparrow} \subseteq x$ whenever $s \in x$. In particular, $\rho(e)$ is upwards closed. Let $Fin(E){\uparrow}$ denote the set of upwards closed subsets of $Fin(E)$. It is easy to see that this is a topology; in fact, it is the smallest topology which contains the sets $\{e\}{\uparrow}$, for each $e \in E$. It is called the Alexandrov topology on $Fin(E)$, [13, Example 3.6.2], and it is our first candidate for a logic of causality: $\mathcal{L}_1(E) = Fin(E){\uparrow}$.

This definition is not very "logical"! We have not provided a syntax for our observations, nor have we explained the axiomatic basis of the logic. To do this we must use frame theory, [7, 13].

Definition 3.3 *A frame F is a poset in which* (1) *all finite meets exist;* (2) *arbitrary joins exist;* (3) *binary meets distribute over arbitrary joins,* $a \wedge \bigvee_{i\in I}\{b_i\} = \bigvee_{i\in I}\{a \wedge b_i\}$.

Frames provide a syntax with arbitrary (infinite) disjunctions and finite conjunctions as well as the constants T and F. Note that the open sets of any topology always define a frame. A homomorphism of frames, $f : F \to G$, is a function which preserves finite meets and arbitrary joins. Frames and their homomorphisms form a category which is algebraic over the category of sets, [7, II.1.2]. That is to say, despite the infinitary operation, free frames exist and frames can be constructed by generators and relations in a familiar algebraic manner, [13, Chapter 4]. Let $Fr(E)$ denote the free frame generated by E.

It is easy to define an observation pairing on $Fr(E)$; it arises naturally out of the frame theory. If $s \in Fin(E)$, define the function $v_s : E \to \{\mathsf{T}, \mathsf{F}\}$, from E to the trivial Sierpinski frame, by $v_s(e) = \mathsf{T}$ if, and only if, $e \in s$. By the universal property of a free frame, this function lifts to a frame homomorphism $v_s : Fr(E) \to \{\mathsf{T}, \mathsf{F}\}$, (for which we use the same notation). Define $s \models \rho$ if, and only if, $v_s(\rho) = \mathsf{T}$. If θ is defined as in (6) then it is proved in [4, Proposition 3.1] that $\theta : Fr(E) \to Fin(E)\uparrow$ is an isomorphism of frames. This explains the syntax and equational theory of $\mathcal{L}_1$.

An example may help to make this seem less abstract. Consider the $\mathcal{L}_1$-automaton

$$G = \boxed{\begin{array}{ll} b & \mathsf{T} \\ c & \mathsf{T} \\ a & b \vee c \end{array}}.$$

We shall always write automata in this way: the left-hand column has the events while the right-hand column has the corresponding observations. This example is isomorphic to the parallel switch in [14, Example 1.1.7] and the reader can check that the Hasse diagram of $\wp(G)$ coincides with the diagram of configurations sketched by Winskel. (For more information on $\mathcal{L}_1$-automata and their relationship to Milner's idea of confluence in CCS, see [5].)

Unfortunately, $\mathcal{L}_1$-automata are entirely conflict-free. Negation is the logical connective which seems closest to the idea of conflict and frames do have some kind of a negation, called the pseudo-complement. In fact frames have a Heyting implication, $\rho \to \sigma$, defined by the rule

$$\rho \to \sigma = \bigvee_{x \wedge \rho \leq \sigma} x.$$

The negation is then given by $\neg\rho = \rho \to \mathsf{F}$. It is not difficult to show that for $\mathcal{L}_1$ this negation is completely trivial: for any $\rho \in Fr(E)$, if $\rho \neq \mathsf{F}$ then $\neg\rho = \mathsf{F}$, [3]. This confirms the conflict-free nature of $\mathcal{L}_1$.

It is not straightforward to find causal logics with a non-trivial negation and a tractable axiomatic basis, [3, 4]. This is the central problem in the logical approach to causality. The difficulty arises because Heyting implication is a secondary operation in a frame: it is not preserved by the frame homomorphisms. If we considered it to be a primary operation and required homomorphisms to preserve it then the resulting category—of complete Heyting algebras—would no longer be algebraic and free objects would no longer exist, [7, I.4.10]. What this means for us is that we cannot simply throw in a negation and expect to generate logics by algebraic methods as we did above for $\mathcal{L}_1$.

One possibility is to push the negation out of the logic and into the semantics of the observation pairing. Let $\mathcal{L}_2(E) = Fr(E) \times Fr(E)$ consist of the language of pairs of observations from $\mathcal{L}_1$. Define the observation pairing by

$$s \models (\rho, \sigma) \text{ iff } s \models \rho \text{ and } s \not\models \sigma.$$

It is proved in [4, Theorem 4.1] that Winskel's General Event Structures correspond bijectively to $\mathcal{L}_2$-automata for which the second component of the observation is "essentially" constant. (For a precise statement, see [4].) This correspondence induces an isomorphism on the domains of configurations so that $\mathcal{L}_2$-automata are a strict generalization of General Event Structures. Unfortunately, they are still not general enough to capture the Muller unfolding. We need a logic which is capable of dealing with more complex conflicts.

Let us return to first principles. The simplest negative observation we could make is the non-occurrence of e. Let $\overline{e}$ denote this observation. It is characterised by the rule $s \models \overline{e}$ if, and only if, $e \notin s$. Guided by what we did above, let us consider the smallest topology which contains such observations.

Definition 3.4 $\mathcal{L}_3(E)$ = *the topology generated by* $\{\theta(e), \theta(\overline{e}) \mid \forall e \in E\}$.

To understand the syntax behind $\mathcal{L}_3$, and in particular to see whether it has an effective negation, we need to look at the frame theory. Let $\overline{E} = \{\overline{e} \mid e \in E\}$. Suppose that $s \in Fin(E)$. By the universal property of free frames, s defines a homomorphism of frames, $v_s : Fr(E \cup \overline{E}) \to \{\mathsf{T}, \mathsf{F}\}$, satisfying

$$\left.\begin{array}{ll} v_s(e) & = \mathsf{T} \\ v_s(\overline{e}) & = \mathsf{F} \end{array}\right\} \text{ if, and only if, } e \in s.$$

Let $\theta : Fr(E \cup \overline{E}) \to 2^{Fin(E)}$. It is clear that θ is a frame epimorphism onto $\mathcal{L}_3(E)$. Hence, frame theory tells us that there exists some set of relations in $Fr(E \cup \overline{E})$ such that

$$\theta : \frac{Fr(E \cup \overline{E})}{< \text{relations} >} \to \mathcal{L}_3(E)$$

is an isomorphism of frames.

But what are the relations? It is not hard to see that $e \wedge \overline{e} = \mathsf{F}$ and $e \vee \overline{e} = \mathsf{T}$ in $\mathcal{L}_3(E)$: an event cannot both be in s and not be in s while any event must either be in s or not be in s. Hence e and $\overline{e}$ are complements of each other, [7, I.1.6], and therefore $\neg e = \overline{e}$. We have found a non-trivial negation which does the right thing on the basic observations. If E is finite, then it is proved in [3] that

$$\theta : \frac{Fr(E \cup \overline{E})}{< e \wedge \overline{e} = \mathsf{F},\ e \vee \overline{e} = \mathsf{T} >} \longrightarrow 2^{Fin(E)} \quad (= 2^{2^E})$$

is an isomorphism of Heyting algebras. Hence, $\mathcal{L}_3(E)$ is the free Boolean algebra on E. (The frame presentation corresponds to disjunctive normal form.) Unfortunately, life is not so straightforward when E is infinite, which is the case of interest to us here. It is not hard to see that $\bigvee_{i \in I} \overline{e}_i = \mathsf{T}$ in $\mathcal{L}_3(E)$ for any infinite subset of events $\{e_i \mid i \in I\}$: any infinite subset of events must contain some event which does not occur in a given finite subset. The topology we have constructed has a surpisingly complex presentation as a frame when E is infinite.

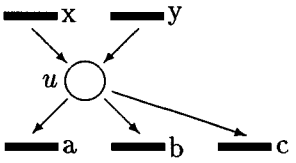
In particular, $\mathcal{L}_3(E)$ is not a classical logic when E is infinite! The negation does not obey both de Morgan laws and $\neg\neg\rho \neq \rho$. (The significance of intuitionism arising in this context is discussed further in [3].) Luckily, one of the de Morgan laws holds in any Heyting algebra: $\neg(\rho \vee \sigma) = \neg\rho \wedge \neg\sigma$, [7, I.1.11], and this is sufficient for the purposes of the next section. We shall spare the reader from any further details of the axiomatics of $\mathcal{L}_3(E)$. A full account may be found in [3].

With this background regarding $\mathcal{L}_3$ in place, we can finally embark upon the main construction.

4 Capturing the Muller unfolding

Let $N = (B, A, F, M_0)$ be a finite 1-safe Petri net. The first step is to determine the events of our automaton. It is natural to take these to represent the firings of transitions since, after all, that is what is counted in $mul(N)$. Let $\mathbf{N}^\bullet$ denote the positive natural numbers. Our events will be elements of $A \times \mathbf{N}^\bullet$, written as a_i, which should be thought of as representing the i-th firing of transition a.

The observations corresponding to these events are harder to write down. Consider a place u in the net N and those transitions which are incident on it as shown below.



(7)

Here, ${}^\bullet u = \{x, y\}$ and $u^\bullet = \{a, b, c\}$. We shall regard (7) as a generic example which will allow us to lighten the syntax considerably. It will be convenient to use the auxiliary function $\phi : B \to \mathbf{Z}$ defined by $\phi(u) = M_0(u) + |u^\bullet| - 1$.

Definition 4.1 *Given the net $N = (B, A, F, M_0)$, define the $\mathcal{L}_3$-automaton $\mathcal{A}(N) = (E, \rho)$ by $E = A \times \mathbf{N}^\bullet$ and*

$$\rho(a_n) = \bigwedge_{u \in {}^\bullet a} \bigvee_{\substack{i+j=n+k+l-\phi(u) \\ i,j \geq 0;\ k,l \geq 1}} (\overbrace{x_i \wedge y_j}^{{}^\bullet u = \{x,y\}}) \wedge \neg(\underbrace{b_k \vee c_l}_{u^\bullet = \{a,b,c\}}). \tag{8}$$

The following conventions will be used to interpret (8):

- $x_0 = \mathsf{T}$, $\forall x \in A$;
- $\bigwedge \emptyset = \mathsf{T}$ and $\bigvee \emptyset = \mathsf{F}$.

Note that, because N is finite, $\rho(a_n) \in \mathcal{L}_3$. The reader's attention is drawn to the discrepancy between the restrictions on the indices i, j—corresponding to transitions in ${}^\bullet u$—which are required to be only non-negative, and the indices k, l—corresponding to transitions in $u^\bullet$—which must always be positive. These restrictions and the discrepancy between them are important to the correct working of (8). The example net in (3) has the $\mathcal{L}_3$-automaton shown below.

a_n	$\bigvee_{i+j=n-1} b_i \wedge c_j$
b_n	$\bigvee_{i=n+j-1} a_i \wedge \neg c_j$
c_n	$\bigvee_{i=n+j-1} a_i \wedge \neg b_j$

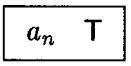
In order to understand how (8) arises, consider the behaviour of the net in the vicinity of the place illustrated in (7). Assume that this place, u, is not marked in the initial marking so that $M_0(u) = 0$ and $\phi(u) = 2$. (The reader will easily be able to supply a similar argument when u is marked initially.) In order that u becomes marked at some point it is necessary that the number of times x and y have fired, in total, should exceed by 1 the number of times a, b and c have fired, in total. (Notice that we have just made use of the fact that N is 1-safe.) Suppose that x and y have fired i and j times respectively, and that b and c are about to fire

for the k-th and l-th times, respectively. Suppose further that a is about to fire for the n-th time. If u is marked, it then follows from what was said above that

$$i + j = n - 1 + k - 1 + l - 1 + 1,$$

which we may rewrite as $i + j = n + k + l - \phi(u)$. This is exactly the constraint which appears under the disjunction in (8). The term within the disjunction tries to capture the fact that x_i and y_j have occurred while neither b_k nor c_l has. In order for a to be able to fire, all its places must be marked, which accounts for the outermost conjunction in (8).

This discussion does not prove anything; it merely suggests that (8) is a necessary consequence of the firing rule for the given interpretation of the events a_i. Note, in particular, that $\rho(a_n)$ makes no mention of a_{n-1}. It is not at all obvious from (8) that $\mathcal{A}(N)$ offers the events a_i in order of increasing i. This precedence is required by the interpretation we have given to the a_i: it would be unfortunate, to say the least, if the second firing of a took place before the first firing of a! It is instructive to consider a pathological situation where this does in fact arise, which has to be excluded from the main theorem. Consider the net with only a single transition, a, and no places. Since ${}^\bullet a = \emptyset$, the conventions above imply that the net has the automaton



which allows any of the events a_n to occur initially. This clearly does not generate the Muller unfolding. This net behaves as though there were a place in the preset of a with an infinite number of tokens which suggests that the pathology is related to what goes wrong in non-safe nets. This falls outside the scope of the present paper. Our concern here is to exclude such examples from the statement of the theorem. If N is 1-safe then a transition a with ${}^\bullet a = \emptyset$ must also have $a^\bullet = \emptyset$. Since we deal only with 1-safe nets, it is sufficient to require that no transition is isolated, [12, §1.5].

Before giving a precise statement of the main theorem we need to identify the underlying function which induces the isomorphism. The states of $mul(N)$ are elements of A^+ while the states of $\wp(\mathcal{A}(N))$ are elements of $Fin(A \times \mathbf{N}^\bullet)$. We can, of course, identify $Fin(A \times \mathbf{N}^\bullet)$ with a subset of $(A \times \mathbf{N}^\bullet)^+$ in the obvious way. Let $\nu : A^+ \to (A \times \mathbf{N}^\bullet)^+$ be defined by

$$\nu(l)(x, i) \;=\; \begin{cases} 0 & \text{if } i > l(x) \\ 1 & \text{otherwise} \end{cases} \tag{9}$$

where $l \in A^+$. It is easy to see that ν must be injective: if $\nu(l_1) = \nu(l_2)$ then $l_1 = l_2$, and, furthermore, that $\nu(l)$ is always a subset, not simply a multiset, of $(A \times \mathbf{N}^\bullet)$. If A is finite, which it will be in what follows, it is clear that $\nu : A^+ \to Fin(A \times \mathbf{N}^\bullet)$. If, for instance, $s = acbca$, then the reader can easily check that $\nu[s] = \{a_1, a_2, b_1, c_1, c_2\}$. We can now state the main result of this paper.

Theorem 4.1 *Let $N = (B, A, F, M_0)$ be a finite 1-safe Petri net with no isolated transitions. The function $\nu : A^+ \to Fin(A \times \mathbf{N}^\bullet)$ induces an isomorphism of posets from $mul(N)$ to $\wp(\mathcal{A}(N))$.*

The proof of this is more difficult than one might expect. We are only able to sketch the outlines of the argument here; full details appear in [6]. The proof falls naturally into two parts: what happens initially and what happens after some sequence of transitions have fired. The first part is straightforward and we give its proof in full to give a flavour for the kind of arguments which are used.

Proposition 4.1 *With the assumptions of Theorem 4.1, if $a \in A$ is a transition which is enabled at the marking M_0, then $\emptyset \models \rho(a_1)$ in $\mathcal{A}(N)$. Conversely, if $\emptyset \models \rho(a_n)$ in $\mathcal{A}(N)$, then $n = 1$ and a is enabled at M_0.*

Proof: Suppose that $a \in A$ is enabled at M_0. Choose $u \in {}^\bullet a$ which we may always do since N is 1-safe and no transition of N is isolated. Since N is 1-safe, $M_0(u) = 1$ and $\phi(u) = |u^\bullet|$. Suppose as usual that the vicinity of the place u is described by the picture (7). According to (8), the contribution of u to $\rho(a_1)$ looks like

$$\bigvee_{i+j=k-1+l-1} (x_i \wedge y_j) \wedge \neg(b_k \vee c_l).$$

Consider the sub-term $\sigma = (x_0 \wedge y_0) \wedge \neg(b_1 \vee c_1) = \neg(b_1 \vee c_1)$ which satisfies the restrictions imposed by (8). Clearly, $\emptyset \models \sigma$. Since we can find such a sub-term for any $u \in {}^\bullet a$, it is clear that $\emptyset \models \rho(a_1)$

Now suppose that $\emptyset \models \rho(a_n)$ in $\mathcal{A}(N)$. Choose $u \in {}^\bullet a$, which, as above, we may always do. It follows from (8) that

$$\emptyset \models (x_i \wedge y_j) \wedge \neg(b_k \vee c_l). \tag{10}$$

for some sub-term satisfying $i + j = n + k + l - \phi(u)$. Since $\emptyset \models x_i$ if, and only if, $i = 0$, this is only possible if either ${}^\bullet u = \emptyset$ or $i = j = 0$. In either case, the remaining indices must satisfy $n + k + l = \phi(u)$. We can rewrite this as

$$M_0(u) - 1 = (n-1) \; + \; (k-1) \; + \; (l-1).$$

Since $n, k, l \geq 1$ we must have that $M_0(u) \geq 1$ from which we deduce—since N is 1-safe—that $M_0(u) = 1$. Since this holds for any $u \in {}^\bullet a$, the transition a is clearly enabled. Furthermore, $(n-1)+(k-1)+(l-1) = 0$, from which we conclude, in particular, that $n = 1$. This completes the proof.

QED

Now suppose that $s \in traces(N)$ and that $M_0 \stackrel{s}{\Rightarrow} M_1$. Let $Q = (B, A, F, M_1)$ be the resulting net, which also satisfies the assumptions of Theorem 4.1. We shall use subscripts to distinguish between the nets N and Q, as in ρ_N and ρ_Q. We want to compare the observations in $\mathcal{A}(N)$ with those in $\mathcal{A}(Q)$. This will enable us to use Proposition 4.1 on Q but to refer this information back to N.

Let $\zeta : A^* \times Fin(A \times \mathbf{N}^\bullet) \to Fin(A \times \mathbf{N}^\bullet)$, which we shall write as $\zeta_s(w)$, be defined by the following rule:

$$\zeta_s(w) \models x_n \text{ if, and only if, either, } n \leq [s](x), \text{ or, } w \models x_{n-[s](x)}.$$

Since $w \models x_n$ is equivalent to $x_n \in w$, when $n \geq 1$, it is clear that this rule gives an unambiguous definition of ζ. This function has many interesting properties: $\zeta_\varepsilon(w) = w$, $\zeta_s\zeta_t(w) = \zeta_{st}(w)$ and $\zeta_s(\emptyset) = \nu[s]$, [6, Lemma 5.2]. The following examples may also help to clarify its behaviour:

$$\begin{aligned} \zeta_\varepsilon(\{a_1, b_2\}) &= \{a_1, b_2\}, \\ \zeta_a(\{a_1, b_2\}) &= \{a_1, a_2, b_2\}, \\ \zeta_{ab}(\{a_1, b_2\}) &= \{a_1, a_2, b_1, b_3\}. \end{aligned}$$

We shall be particularly concerned with subsets of $(A \times \mathbf{N}^\bullet)$ which, so to speak, have no gaps in their indices. We can identify them in the following way.

Definition 4.2 *An element $w \in Fin(A \times \mathbf{N}^\bullet)$ is said to be sequenced if $w = \zeta_s(\emptyset)$ for some $s \in A^*$.*

For example, $\{a_1, b_1, b_2\}$ is sequenced, while $\{a_1, b_2\}$ is not. More precisely, if w is sequenced and $w \models x_n$, then $w \models x_i$ for any $1 \leq i \leq n$, [6, Lemma 5.3].

The next result is the key ingredient in the proof of Theorem 4.1. Recall the definition of the net Q given above.

Proposition 4.2 *With the assumptions of Theorem 4.1, let $a \in A$ and assume that $n > [s](a)$ and that $w \in Fin(A \times \mathbf{N}^\bullet)$ is sequenced. Then,*

$$\zeta_s(w) \models \rho_N(a_n) \text{ if, and only if, } w \models \rho_Q(a_{n-[s](a)}).$$

The proof of this reduces easily to the case where $s = e$ for some $e \in A$, which then follows from a careful case analysis. Proposition 4.2, in conjunction with Proposition 4.1, allows us to work out whether or not $\mathcal{A}(N)$ will offer a_n, after N has offered s. This gives us sufficient information, inductively, to complete the proof of Theorem 4.1. We hope that this brief sketch has given the reader some idea of how the proof works.

5 Conclusion

The field of asynchronous circuit design has undergone a great resurgence in recent years, [1]. We believe that the concurrency theorist can find many interesting questions to look at in this application area and we hope that our version of Muller's construction will set a precedent for this. The logic of causality, $\mathcal{L}_3$, which we have introduced here is successful at dealing with the complexities of the Muller construction but its axiomatic basis is itself very complex. It seems unlikely that $\mathcal{L}_3$ is the optimal logic of causality, if such a thing exists at all. Is there a better or simpler one—and corresponding event structures—which could accomplish the same task? Questions like this make us realise that causality is still a largely unexplored subject, full of difficult and fascinating problems. Perhaps the results of this paper will spur others towards attacking some of them.

References

[1] J. A. Brzozowski and C.-J. H. Seger. Advances in Asynchronous Circuit Theory Part I: gate and unbounded inertial delay models. *Bulletin of the EATCS*, 42:198–249, 1990.

[2] M. Droste. Event structures and domains. *Theoretical Computer Science*, 68:37–47, 1989.

[3] J. Gunawardena. Events, Causality and Logic. In preparation.

[4] J. Gunawardena. Geometric logic, causality and event structures. In J. C. M. Baeten and J. F. Groote, editors, *CONCUR'91 - 2nd International Conference on Concurrency Theory*, pages 266–280. Springer LNCS 527, 1991.

[5] J. Gunawardena. Causal Automata. *Theoretical Computer Science*, 101:265–288, 1992.

[6] J. Gunawardena. On the causal structure of the Muller unfolding. Technical Report STAN-CS-93-1466, Department of Computer Science, Stanford University, March 1993.

[7] P. T. Johnstone. *Stone Spaces*, volume 3 of *Studies in Advanced Mathematics*. Cambridge University Press, 1982.

[8] L. H. Landweber and E. L. Robertson. Properties of Conflict-Free and Persistent Petri Nets. *Journal ACM*, 25(3):352–364, 1978.

[9] D. E. Muller and W. S. Bartky. A theory of asynchronous circuits. In *Proceedings of an International Symposium on the Theory of Switching*. Harvard University Press, 1959.

[10] M. Nielsen, G. Plotkin, and G. Winskel. Petri nets, event structures and domains. *Theoretical Computer Science*, 13:85–108, 1981.

[11] R. J. Parikh. On context-free languages. *Journal ACM*, 13(4):570–581, 1966.

[12] W. Reisig. *Petri Nets*, volume 4 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1985.

[13] S. Vickers. *Topology via Logic*, volume 5 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1989.

[14] G. Winskel. Event structures. In W. Brauer, W. Reisig, and G. Rozenberg, editors, *Advances in Petri Nets*. Springer LNCS 255, 1987.

[15] A. Yakovlev. Analysing Concurrent Systems through Lattices. Draft, 1991.

[16] A. Yakovlev, L. Lavagno, and A. Sangiovanni-Vincentelli. A Unified Signal Transition Graph Model for Asynchronous Control Circuit Synthesis. In *Proceedings ICCAD'92*, 1992.