LOTOS-Like Composition of Boolean Nets and Causal Set Construction

Tommaso Bolognesi(✉)

CNR-ISTI, Pisa, Italy
t.bolognesi@isti.cnr.it

Abstract. In the context of research efforts on causal sets as discrete models of physical spacetime, and on their derivation from simple, deterministic, sequential models of computation, we consider *boolean nets*, a transition system that generalises cellular automata, and investigate the family of causal sets that derive from their computations, in search for interesting emergent properties. The choice of boolean nets is motivated by the fact that they naturally support compositions via a LOTOS-inspired parametric parallel operator, with possible interesting effects on the emergent structure of the derived causal sets.

More generally, we critically reconsider the whole issue of algorithmic causet construction and expose the limitations suffered by these structures w.r.t. to the requirements of Lorentz invariance that even *discrete* models of physical spacetime, as recently shown, can and should satisfy. We conclude by hinting at novel ways to add momentum to the bold research programme that attempts to identify the natural with the computational universe.

Keywords: Boolean nets · Causal sets · Discrete spacetime · Parallel composition · LOTOS

1 Introduction

This paper is dedicated to Ed Brinksma and is largely motivated by a desire to explore possible bridges between the topics investigated by Ed and friends in 'those good-old LOTOS days' – process-algebraic languages and operators, formal specification styles and structuring principles, etc.[1] – and the new research field that the author has joined after quitting the formal methods community, around 2005. This new area of activity deals with the emergent properties of the

[1] Of course the range of Ed's activities is broader, as suggested by the Festschrift title 'ModelEd, TestEd, TrustEd'. Indeed, the addition of 'randomisEd' wouldn't be completely inappropriate, in light of an episode which involved a small group of 'LOTOS-eaters' during a relaxing late-evening walk in a forgotten European city. On that occasion Prof. Brinksma, dissatisfied with the manipulations performed on the Rubik Magic Rings puzzle by the author – dismissed as insufficiently random – gave a public, truly brilliant demonstration of his unexpected randomisation skills.

J.-P. Katoen et al. (Eds.): Brinksma Festschrift, LNCS 10500, pp. 27–47, 2017.
DOI: 10.1007/978-3-319-68270-9_2

computations of simple programs, and with discrete models of physical spacetime. The occasion gives us also the opportunity for a critical assessment of some of the steps taken in these directions.

Causal Sets. Causality among events in spacetime is regarded by many theoretical physicists as a most fundamental aspect of nature, and represents a key notion in the *continuum* spacetime of Special and General Relativity. When revisited under the assumption of spacetime discreteness – a feature often imagined to manifest itself at the Plank scale (10^{-35} m, 10^{-44} s) – the idea of causality finds a simple realisation in terms of causal sets [10,28,30]. A *causal set*, or *causet*, is a partially ordered set of events $(S, \preceq)$ with the additional property of being *finitary*, which means that all *order intervals* $I[s,t] = \{x | s \preceq x \preceq t\}$, for any pair of elements s and t (the *source* and the *sink*) must be finite. A causet can be represented by a directed acyclic graph. Most efforts in the Causal Set Programme are concerned with identifying adequate counterparts, in the discrete setting, of concepts and features of continuous spacetime [29], such as lightcones, Lorentz invariance [12], dimensionality [22], curvature [1].

The most direct way to obtain a causet of solid, physical realism is to directly derive it, using the *stochastic sprinkling technique* to be introduced later, from a solid, continuous, Lorentzian manifold (e.g. flat Minkowski or positively-curved De Sitter space-time) guaranteed to satisfy the Einstein field equations. However, an attractive challenge for those who support the conjecture of an ultimately discrete, computational and deterministic nature of the universe, is to derive realistic causets directly from the computations of simple, discrete, deterministic models, without resorting to predefined continuum solutions, as *sprinkling* does. This idea has been first proposed by Wolfram [34], under the whimsical name of 'universe hunting', and has been further investigated by the author since 2010, often referred to as 'algorithmic causet construction' [3–5].

Bridges. How can we establish a bridge between algorithmic causal sets and process algebra? A possible link is suggested by the observation that partially ordered structures of events are such general and flexible mathematical objects that, unsurprisingly, they find application in a number of diverse fields of science and technology, including Computer Science. For example, roughly in the same period during which the Causal Set Programme for Quantum Gravity started to use these structures as discrete models of spacetime, in Theoretical Computer Science, in the areas of Concurrency Theory and Formal Methods, the 'true concurrency semantics' research effort started to devise mappings from process algebras to event structures somewhat analogous to causets [32,33].

An important difference between the two types of event set, from Physics and from Computer Science, is that in the former *all* events are expected to take place, while in the latter, as a reflection of choice operators in the syntax, special relations indicate that some events are in conflict. In Bundle Event Structures [20], for example, relation $e \# e'$ means that there is no system run in which both e and e' occur: some portions of the event structure remains 'unvisited'.

However, a difference between the two structures that is more interesting for our purposes here is that we cannot hope to detect emergent, macroscopic, possibly regular patterns in stochastic causets - by the very definition of stochastic process - while in the event structures from Computer Science this is certainly possible, since structure and order are inherited from the syntax itself. This fact, abstractly represented in Fig. 1, makes event structures from process algebraic specifications potentially interesting under a quantum gravity or computational universe perspective.

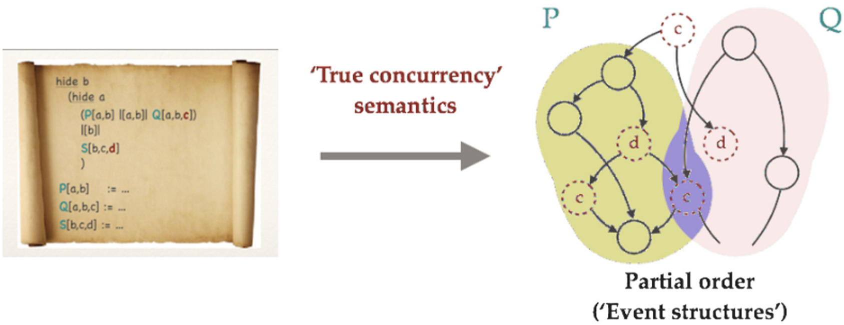

Fig. 1. The true-concurrency semantics of a process-algebraic specification maps a formal piece of syntax into a highly structured set of partially ordered events. The structure found in the semantic object on the r.h.s. is inherited from the structure in the syntactic object on the l.h.s.

Indeed, our initial plan was to investigate the emergent properties of the event structures obtained by the 'true concurrency semantics' of LOTOS [20] for large corpora of specifications, possibly generated at random. The unavailability of a fully automated semantics for a sufficiently large subset of the language – one including recursion – and the limited resources at our disposal, prevented us from following this path. We have therefore opted for a conceptually simpler labelled transition system – boolean nets – and have studied the causets that originate from their computations. Interestingly, it is straightforward to export to this state transition model the LOTOS parametric parallel composition operator, that represents a key structuring construct of the language: this is attractive, in light of the importance that we attribute to the emergence of macro/structures in causets.

Paper Plan. In Sect. 2 we introduce boolean nets, their synchronous and asynchronous executions, and the global graphs derived in the two cases. In Sect. 3 we contrast stochastic vs. deterministic causet construction techniques, recalling the main technique of the first type – manifold *sprinkling* – and mentioning two alternative approaches – *indirect* and *direct* – for building causets of the second type. In Sect. 4 we address the derivation of causets from (unstructured) asynchronous computations of boolean networks, under three different execution policies, and we study a peculiar property of the obtained graphs that has to do with Lorentz invariance. This leads us to critically reconsider the *indirect* approach to algorithmic causet construction in its generality. In Sect. 5 we consider the parametric parallel composition of (asynchronous) boolean nets, and take a preliminary look at the associated causets. In Sect. 6 we mention a few aspects in which, in our opinion, research and experimentation on 'universe hunting' could find new momentum and better results in term of emergent complexity.

2 Boolean Networks: Sync and Async Execution

Boolean networks, abbreviated *bool nets* in the sequel, are a sequential dynamical system based on a finite set of boolean variables, each controlled by a different boolean function. Random bool nets have been originally developed by Stuart Kauffman for modelling genetic regulatory networks [19], and have found application, more recently, in *Integrated Information Theory*, as abstract models of neural networks [25].

2.1 The Model

An (N, k)-bool net is a pair $(G(B, E), F)$ where:

- $G(B, E)$ is a directed graph with N vertices $B = \{b_1, \ldots, b_N\}$, and $N \cdot k$ edges E that specify the k input arguments: $b_{i,1} \to b_i, \ldots, b_{i,k} \to b_i$ for every $b_i \in B$.
- $F = \{f_1, \ldots f_N\}$ is a set of N boolean functions of k arguments.

Each vertex $b_i \in B$ is a boolean variable controlled by boolean function $f_i(b_{i,1} \ldots b_{i,k}) \in F$. The ordered k-tuple of arguments $(b_{i,1} \ldots b_{i,k})$ identifies the bits in B that f_i reads, and corresponds to k directed edges in E, namely edges $b_{i,1} \to b_i, \ldots, b_{i,k} \to b_i$. Thus, there is a total of $|E| = N * k$ edges.[2] The $G(B, E)$ graph of a $(N5, k3)$-bool net is provided, as an example, in Fig. 2-left. The numeric codes of the boolean functions associated to each node are indicated in parentheses. Note that there are 2^{2^k} boolean functions of k variables – 256 for this example.

There are two ways in which a boolean net can be executed:

Synchronous execution. This criterion naturally combines with a *discrete time* assumption: at each time step, or clock tick $t = 0, 1, 2 \ldots$, each function f_i reads the values of its k arguments – the bits $b_{i,1}, \ldots, b_{i,k}$ identified by the incoming edges of node b_i – and fires, assigning the computed value to its controlled node, i.e. to variable b_i. All functions fire together. Under this synchronous firing policy, boolean nets are a generalisation of cellular automata [17].
We assume deterministic functions $F = \{f_1, \ldots f_N\}$, thus synchronous evolution is itself *deterministic*: each global state has only one successor. Thus, placing deterministic bool nets with synchronous evolution in the wider context of probabilistic system - viewing them as special, limit cases of that

[2] Note that, unless decorated with appropriate edge priority assignments, the graph is not sufficient for correctly identifying the order of function arguments: this is disambiguated in F.

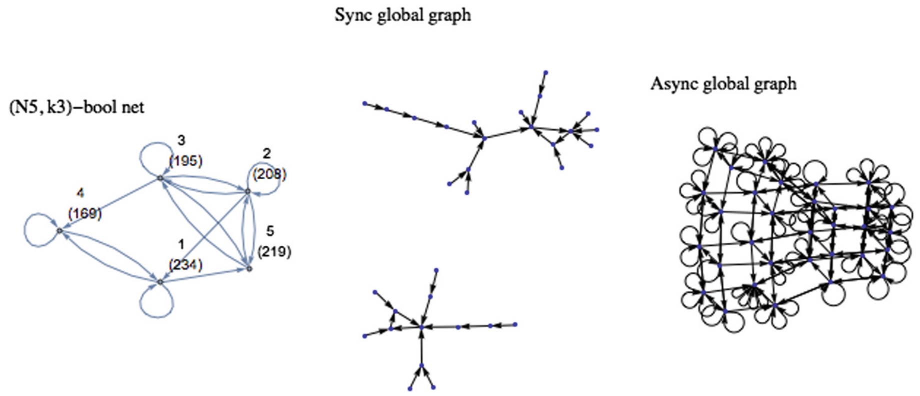

Fig. 2. $G(B, E)$ directed graph for a $(N5, k3)$-bool net (left). Numbers in parentheses near each node identify the boolean function of three variables that controls the node bit. Each node has three incoming arcs, identifying the argument bits. Global graphs for the same net under sync (middle) and async (right) execution.

family - we can represent them by *causal graphs* [26], since they fulfill the requirement that the state at time $t+1$ of each variable b_i, denoted b_i^{t+1}, is conditionally independent of b_j^{t+1}, for every other variable b_j, given the global state B^t of the system at time t: $prob(b_i^{t+1}|B^t, b_j^{t+1}) = prob(b_i^{t+1}|B^t)$. Each bit/function $b_i = f_i(b_{i,1} \ldots b_{i,k})$ can thus be interpreted as an individual causal element within the system [26].

Asynchronous execution. Although we assume deterministic functions $F = \{f_1, \ldots f_N\}$, asynchronous evolution admits both *deterministic* and *nondeterministic* variants.

The *nondeterministic* form naturally combines with a *continuous time* assumption: we imagine function firing to be an instantaneous random event, occurring independently from other firings. In continuous time, the probability of two firings to occur simultaneously is zero, thus we assume that all these events occur one at a time, in an interleaving fashion. Correspondingly, each global state may have multiple successors – as many as the number N of bits. The continuous time postulate that no two firings occur simultaneously actually introduces a causal dependency between the individual bits $b_i \in B$. While the global transitions of the system from its current state B_t to its next state B_{t+1} may still be interpreted as a global mechanism, the states of individual bits within system B are not conditionally independent on the past. The next state of an individual bit may depend on the next state of other bits b_j in addition to the current state of its parents $\{b_{i,1} \ldots b_{i,k}\}$. For this reason, asynchronously updated bool nets cannot be interpreted as a system composed of individual causal elements [25], and thus do not readily fit into the framework of causal graphs [26].

Two types of *deterministic*, asynchronous bool nets shall be introduced later.

2.2 Global Graphs

Bool nets, either sync or async, are *finite* transition systems, thus we can capture their behaviour by a directed, global, state transition graph in which each node is a global state, i.e. a tuple of bits. For a complete characterisation of the bool net behaviour we do not refer to a specific initial global state, but create all global states and find all transitions that emanate from each of them. In general the graph may be disconnected.

Two global graphs, for the same boolean net but with sync or async execution, are illustrated in Fig. 2. The sync graph in the middle has two connected components, each featuring a three-node cyclic attractor. The layout of the async graph on the right exhibits some degree of symmetry, and may give the impression of a 3-D assembly of cubic frames: this is a consequence of the transition interleaving policy, by which a group of transitions may fire in all possible orderings. (In a simple setting, with just two transitions, interleaving yields the typical diamond shape.)

3 Stochastic vs. Deterministic Causets

Started in the late 1980's [10], the Causal Set Programme has always been concerned with techniques for building realistic causets, able to reproduce or approximate features of physical, continuous spacetime. Invariably, all the considered techniques have been of *stochastic* nature. The primary technique, in this group, is *sprinkling*.

By the *sprinkling technique* one can derive a causet from a Lorentzian manifold M provided with a volume measure, in two steps. First one creates a uniform, Poisson distribution of points - to become the causet nodes - in a finite region of M, with density δ, so that the expected number of points in a volume V is δV, and the probability to find exactly n points in that portion is:

$$P(n) = \frac{(\delta V)^n e^{-\delta V}}{n!}. \tag{1}$$

Then the causet edges are created by letting the sprinkled points inherit the causal structure of M: in M two points/events are causally related when their squared Lorentz distance L^2 is positive (time-like relation) or null (light-like relation), and are causally unrelated when L^2 is negative (space-like relation).[3] In the sequel we shall conveniently call these objects *sprinkled causets*. Sprinkled causets can be regarded as the most direct discrete versions of 'real', continuous forms of physical spacetime (e.g. Minkowski and De Sitter).

As mentioned in the introduction, a challenging goal of causet-based quantum gravity research is to build causets of physical significance without resorting to an underlying continuum, with the manifold obtained a posteriori, as an asymptotic approximation.

[3] In four dimensional, Minkowski spacetime M^{1+3}, with time dimension t and spatial dimensions x, y, z, the squared Lorentz distance between events $e_1(t_1, x_1, y_1, z_1)$ and $e_2(t_2, x_2, y_2, z_2)$ is $L^2(e_1, e_2) = +(t_2 - t_1)^2 - (x_2 - x_1)^2 - (y_2 - y_1)^2 - (z_2 - z_1)^2$.

The first experiments with *deterministic* techniques for causet construction, as opposed to *stochastic* techniques, have been carried out by Wolfram [34], although some preliminary ideas can be found in [16].

The rationale for this alternative approach is, in our opinion, quite strong, although still controversial in the community of theoretical physicists: as widely shown by Wolfram with cellular automata and other simple models [34], the richness and variety of patterns that emerge from suitable (and, typically, graphical) representation of deterministic computations, ranging from regular, periodic behaviours to fractal structures, from pseudo-randomness to 'digital particles', is far beyond the reach of purely stochastic models. Furthermore, the assumption of a physical universe fundamentally fuelled, at its lowest spacetime scales, by a digital computation rather than by differential equations, has appealed several physicists (and non-physicists) in the last decades [15,21,31,36] and may be, at present, the best candidate for explaining the peculiar mix of order and disorder found in nature. (For a comprehensive collection of papers on these issues, see [13].)

Following Wolfram's pioneering steps, we have carried out a number of additional experiments with algorithmic causet construction [3–8], investigating properties such as dimensionality, curvature, and Lorentz invariance in the discrete setting.

We distinguish two main techniques for algorithmic causet construction.

Indirect. Reflecting Wolfram's original ideas, a causet is obtained by considering the computation of a sequential model (e.g. an n-dimensional Turing machine), by viewing the computation steps as the events (nodes) of the causet, and by inferring the causal relations among events from the write and read operations carried out at each step on the state variables of the model (e.g. the tape cells and the state of the Turing machine head). A concrete example of application of this technique is provided in Sect. 4.

Direct. In this case we devise an algorithm that directly creates and manipulates the graph representing the causet.

As the reader may have already realised, there is an abundant degree of arbitrariness in these constructions, and no clear guiding principle for their choice, other than, perhaps, conceptual simplicity. The exploration of the computational universe, as conceived by Wolfram, is fundamentally a blind experimental activity: run virtually all instances of the model at hand and see what happens. One may also object that, under the assumption of a computational universe, the choice of a specific Turing-universal model from which to derive causets is irrelevant, since all of them are equivalent, at least in terms of computing power. In practice, however, different models perform quite differently when it comes to concretely spotting interesting properties. Cellular automata diagrams, for example, are more convenient than many other models for detecting, by direct visual inspection, interesting emergent patterns such as digital particles. Thus, we still consider it interesting to explore families of causets derived from different models of computation.

4 Causets from Async Bool Nets

What type of causet can be derived from the computation of a bool net, following the 'indirect' construction approach of Sect. 3?

Let $A = (G(B, E), F)$ be a (N, k)-bool net, and let us conveniently restrict here to the case of *asynchronous* execution (see Subsect. 2.1), in which step w of the sequential computation corresponds to the application of just one boolean function $f_i \in F$, that we write $f_i^w(b_{i,1}, \ldots b_{i,k})$ for stressing its position w in the sequence of steps. We shall directly say that *event w has written bit b_i after reading bits $b_{i,1}, \ldots b_{i,k}$*. We have chosen to address asynchronous rather than synchronous bool nets because they are closer to the LOTOS execution model, and because the derivation of causets from them is easier and more in line with our past experiments.

Following the general technique described in [4], to which the reader is referred for more details, a causet $C = (S, \preceq)$ is readily obtained from the selected computation of bool net A as follows:

- S is the set of computation steps, identified only by their temporal order of occurrence w – a natural number;
- $v \preceq w$, where $v, w \in S$, whenever one of the arguments $b_{i,1}, \ldots b_{i,k}$ of function $f_i^w(b_{i,1}, \ldots b_{i,k})$, say $b_{i,j}$, sees event v as its *most recent writer event*, meaning that no other event between v and w has written $b_{i,j}$. We say that $b_{i,j}$ is the *causality mediator* between v and w.

For obtaining and comparing multiple causet types from the same basic model we consider three different bool net (async) execution policies:

Nondeterministic - At each step the choice of which function to fire it taken uniformly at random.

Deterministic - bit cycling - The N bits of the net are updated one after the other, from left to right, in cycles.

Deterministic - label cycling - Function firings are enriched by labels, which turn out to be particularly useful when used in conjunction with LOTOS-like parallel composition of bool nets. These labels are assigned by a deterministic mechanism: each function $f_i(b_{i,1}, \ldots b_{i,k})$, controlling bit b_i of net A has an associated one-to-one *labelling function* $\alpha : \{0, 1\}^k \to L$, which returns a different symbol of alphabet L for each different k-tuple of bits read by f_i. L thus includes 2^k symbols; furthermore, it is ordered.[4] A pointer scans L from left to right, and stops at the first label that is represented in one of the transitions: this is the transition to be fired. When multiple transitions share that label, the one corresponding to the bit with lowest index is chosen. This labelling policy is just a simple implementation of the idea that a transition label should depend on the current state of the system, but there are

[4] For $k = 3$, for example, we typically set $\alpha(0, 0, 0) = 0$, $\alpha(0, 0, 1) = 1, \ldots, \alpha(1, 1, 1) = 7$, with $L = \{0 \ldots 7\}$ ordered in the natural way. In the sequel we shall also create a different labelling function for each different bool net bit by considering different rotations of the range tuple $(0 \ldots 7)$.

clearly many other ways to reflect this requirement, or even to dismiss it. Our choice has been, admittedly, quite arbitrary, and we cannot exclude that other labelling techniques might yield more interesting causets; indeed, a certain degree of arbitrariness seems unavoidable, in 'Wolfram-style' explorations of the huge universe of deterministic computations.

Which causet properties are we going to analyse?

In [8] we have considered several statistical indicators meant to characterise causets obtained from various techniques, and to measure their closeness to the ideal Lorentzian causets – the sprinkled causets mentioned in Sect. 3. Here we focus on just one indicator, which refers to the out-degrees of causet nodes.[5]

The importance of looking at the growth rate of causet node out-degrees is well explained by Rideout [27]:

> "*The 'usual' discrete structures which we encounter, e.g. as discrete approximations to spatial geometry, have a 'mean valence' of order 1. e.g. each 'node' of a Cartesian lattice in three dimensions has six nearest neighbors. [...] Such discrete structures cannot hope to capture the noncompact Lorentz symmetry of spacetime. Causal sets, however, have a 'mean valence' which grows with some finite power of the number of elements in the causal set. It is this 'hyper-connectivity' that allows them to maintain Lorentz invariance in the presence of discreteness.*"

Thus, an important requirement for a causet to support Lorentz invariance is that the number of outgoing links from the generic causet node should grow with the size of the causet (see also [9]).

In [11] Bombelli et al. mention that, considering the causet $C[s,t]$ obtained from uniformly sprinkling points in an order interval $I[s,t]$ of height T of d-dimensional Minkowski space (T being the Lorentz distance between s and t), the number of nearest neighbors of the root node s – the number of outgoing links – grows like $Log(T)$ for $d = 2$, and like T^{d-2} for $d \geq 3$, provided that the sprinkling density is kept constant. Again, the essential feature here is that the out-degree of *each* node in a sprinkled causet will grow, possibly slowly, but unbounded, as new nodes are added.

Can we expect this feature to be satisfied by the causets derived from the three variants of bool net computation just introduced? The answer depends on whether we adopt a nondeterministic or deterministic execution and, in the second case, it depends on the type of algorithm.

Let us clarify the issue in the wider context of causets derived from the sequential steps of virtually any model of computation, following the 'indirect' technique in which causality is induced by the mediation of state variables.

Consider some generic sequential model of computation and let X be the possibly dynamic set of state variables that can be read, written, created or

[5] Recall that we always consider the causet in its transitively reduced form, or Hasse diagram, whose edges are often called 'links'.

destroyed by the computation steps. Recall that we establish a direct correspondence between the steps and the causet nodes, so that we can sloppily attribute read/write or create/destroy operations to the ones or the others.

Here is what may happen in terms of causet link creation and node out-degree growth:

– If event w *reads* variable $\boldsymbol{x} \in X$, then a new edge $v \to w$ is created, where event v is the most recent writer of $\boldsymbol{x}$.
– If event w *creates* variable $\boldsymbol{x}$, it acquires the opportunity to see its own outgoing edges grow in number, thanks to all future events, if any, that read $\boldsymbol{x}$ before some other event writes or destroys $\boldsymbol{x}$.
– If event w *writes* or *destroys* $\boldsymbol{x}$, it permanently prevents node v – the most recent writer of $\boldsymbol{x}$ until w – to collect further outgoing edges.

If follows that the only circumstance in which a causet node v can see its out-degree grow unbounded is when v creates/writes $\boldsymbol{x}$, and in the subsequent events, $\boldsymbol{x}$ is read infinitely often but never rewritten.

As a consequence, any *fair* sequential model in which each state variable is always eventually updated yields causets in which *all* nodes exhibit an O(1) growth of their out-degrees. This is clearly the case of our *nondeterministic*, async bool net computations, that behave fairly by definition!

How about *deterministic* bool net computations? The 'advantage' of these computations is that they may behave unfairly in a number of creative ways!

Let us then consider Fig. 3. In these plots we compare four types of causets in terms of their node degree growth. They are: causets from sprinkling in 2D space (see Sect. 3), causets from nondeterministic bool nets, and causets from deterministic bool nets with bit-cycling or label-cycling.

Each plot refers to the growth of a single 300-node causet, and collects 300 function plots, each describing the out-degree growth of a different node as the causet develops.

The two plots in the upper row, with their random-like traits, reflect the nondeterministic nature of the computations from which they originate, but differ in a fundamental aspect: in a sprinkled causet (upper-left) – the structure of reference for Lorentz invariance – each node out-degree grows slowly but unbounded; in a causet from an infinite, nondeterministic async bool net computation, each node reaches, with probability 1, a constant, permanent out-degree, for the reasons we have discussed, although ever bigger constants may be achieved, as the computation unfolds.

The two plots in the lower row are from the two deterministic variants of async bool net computations: bit-cycling and label-cycling. Not surprisingly, the bit-cycling policy, being maximally fair with respect to bit choice, prevents the growth of node out-degrees beyond N, the number of bits in the net, since, in the causet construction process, no event can play the role of most recent bit writer for more than N steps. On the contrary, the label-cycling policy, exerting only an indirect control on the choice of the bit to update, leaves room, in a few cases, to 'unfair' behaviours, thus to unbounded node out-degree growth, as clearly visible in the lower-right plot of Fig. 3.

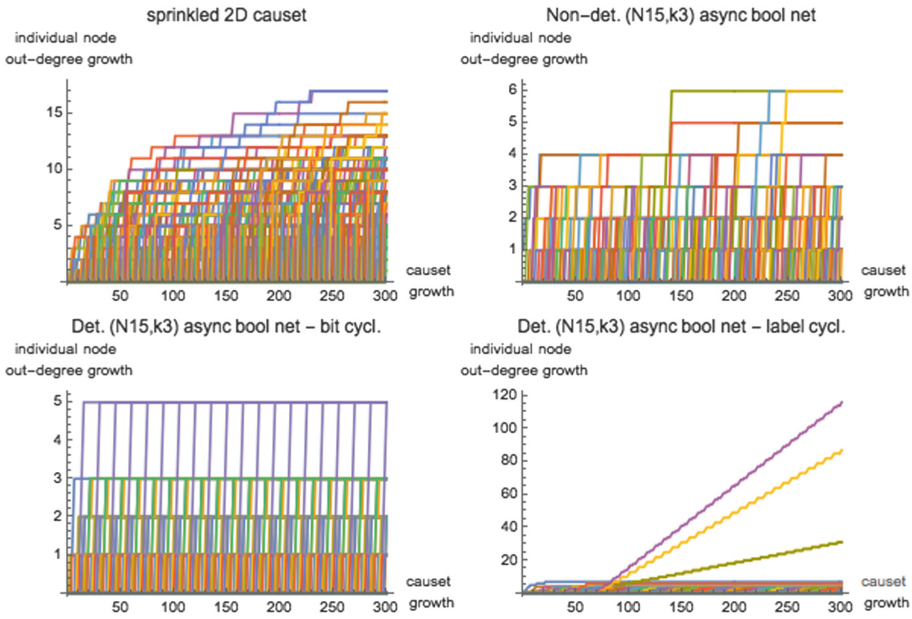

Fig. 3. Individual growth rates, as a function of causet size, of the out-degree of *all* nodes in causets of four types: sprinkled causet of dimensions $d = 2$ (upper-left), causet from nondeterministic async bool net (upper-right), from deterministic async bool net with bit cycling (lower-left), and from deterministic async bool net with label cycling (lower-right).

Indeed, we can cheaply establish a limitation also for the deterministic causets from label-cycling bool net computations: the number of causet nodes with unbounded node degree must be *finite*, and smaller than N, the total number of bits. The reason is as follows. Recalling that each event writes precisely one bit of the net, if we had more than N events exhibiting unbounded out-degree growth, by the pigeon-hole principle at least two of them, e_x and e_y, would be in charge of the same bit, implying that the occurrence of the most recent of them in the sequential computation, say e_y, would obscure e_x as most recent writer of that bit, stopping permanently the out-degree growth of e_x.

This circumstance indicates that these toy causets cannot compete with sprinked causets in terms of physical realism. Still, the examples above seem to demonstrate that, when focusing on the much desired property of unbounded out-degree growth, a deterministic approach to causet construction may offer advantages w.r.t. a nondeterministic one – excluding of course sprinkling itself, which directly derives its good properties from an underlying manifold of guaranteed physical significance.

Discussion on the limits of causets obtained by the 'indirect' technique shall be resumed in the closing section.

Although it might be interesting to study further properties of the class of causets derived from *unstructured* bool nets, we turn now to LOTOS-like *compositions* of bool nets, hoping to spot interesting effects on the derived causets in terms of macroscopic emergent properties.

5 Causets from Parallel Compositions of Bool Nets

This section deals with deterministic computations of LOTOS-like compositions of bool nets. In this case, for moving from potentially nondeterministic computations to deterministic ones we disregard the bit-cycling technique and concentrate on the label-cycling policy, given the fundamental role played by transition labels in the LOTOS parametric parallel composition operator.

5.1 Composing Bool Nets

When two LOTOS processes P and Q are composed in a *parallel composition expression* '$P|[syncLabs]|Q$', where *syncLabs* is a set of labels, the resulting labelled transition system is obtained by forcing the processes to proceed jointly – in synchrony – on the transitions with labels in *syncLabs*, while proceeding independently on their other transitions – in an interleaving fashion. This is what established by the inference rules of transition for the parametric parallel composition operator:

$$\frac{P \xrightarrow{x} P' \wedge x \notin syncLabSet}{P|[syncLabSet]|Q \xrightarrow{x} P'|[syncLabSet]|Q} \quad (left\ \ interleaving) \tag{2}$$

$$\frac{Q \xrightarrow{x} Q' \wedge x \notin syncLabSet}{P|[syncLabSet]|Q \xrightarrow{x} P|[syncLabSet]|Q'} \quad (right\ \ interleaving) \tag{3}$$

$$\frac{P \xrightarrow{x} P' \wedge Q \xrightarrow{x} Q' \wedge x \in syncLabSet}{P|[syncLabSet]|Q \xrightarrow{x} P'|[syncLabSet]|Q'} \quad (synchronisation) \tag{4}$$

The derivation of a *global transition graph* from an async bool net (unstructured) was discussed in Subsect. 2.2. Since the semantics rules (2)–(4) for LOTOS parallel composition are applicable to *labelled transition systems*, it is perfectly feasible to apply them to the composition of boolean nets P and Q. The only missing elements are transitions labels!

For assigning labels to the individual transitions of async bool nets P and Q, we use the deterministic labelling policy described in Sect. 4, based on a labelling function α. On this basis, the application of rules (2)–(4) becomes possible, and the expression $P|[syncLabs]|Q$ formally identifies all possible transitions of the composite system also when P and Q are bool nets.

Thus, expression $P|[syncLabs]|Q$ denotes a composite system in which P and Q may execute their respective transitions independently form each other, or jointly, when both are labelled by an element of *syncLabs*, thus involving a mix of synchrony and asynchrony. We shall use the LOTOS notation '$P|||Q$' for the case $syncLabs = \phi$ ('pure interleaving'). Of course, the composite transition system can be further composed with additional async nets.

5.2 Derived Causets

For exploring the class of causets associated with deterministic computations of composite bool nets, we start from most elementary instances of the model. We shall therefore consider bool net compositions of form $P[N3, k3]|||Q[N3, k3]$.

Figure 4 illustrates four typical pairs of graphs for this type of composition. Each pair consists of a *raw* causet and its transitively reduced form, next to each other. All four *reduced* causets – the proper causets of interest here – collapse to a trivial tree form that basically washes away the structure of the raw graph.

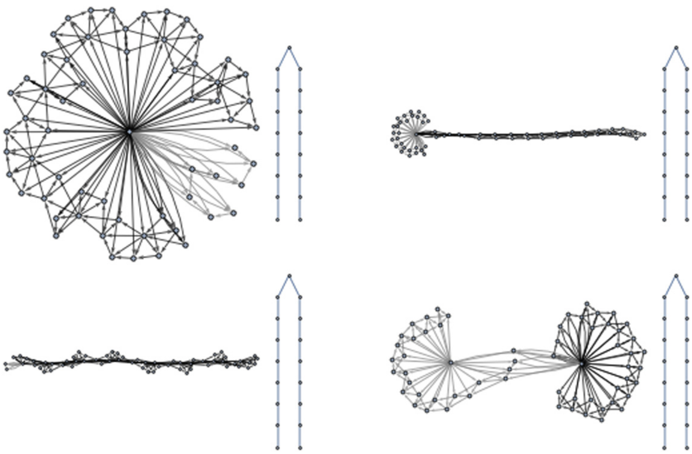

Fig. 4. Causets from deterministic computations of randomly generated bool nets $P[N3, k3]$ and $Q[N3, k3]$ composed by disjoint (interleaving) parallel composition $P|||Q$. The computation is made deterministic by the cycling label policy described in the text. The four (transitively reduced) causets share the same trivial tree-like form, and are shown, for clarity, with fewer nodes than the corresponding raw graph. In the raw graphs, the causal edges created by transitions of P and Q are rendered, respectively, in black and grey. The central node with high out-degree in the upper-left graph corresponds to the initial event of the computation, which initialises all 6 bits of P and Q. The different shapes of the raw graphs essentially depend on the number of nodes that succeed to permanently keep the role of last writer for some bit of P or Q.

It is easy to realise that, by using the $P|||Q$ composition, the causet events generated by the independent transitions of P and Q can only be arranged in two independent total orders: when $N = k$, as in this case, each event e of P reads *all* bits of P, thus it causally depends on the immediately preceding event e' of P, no matter which bit e' has written. Likewise for Q.

It is important not to confuse global transition graphs, not shown here, with causets. First, the global transition graph contains all transitions among all possible global system states, while a causet corresponds to one particular execution path on the global graph, and reveals its intrinsic partial order, if any.

Second, a causet *node* represents an event corresponding to a *transition* in the global graph.[6]

Figure 5 refers to two bool net compositions of form $P[N3, k3]|\{a\}|Q[N3, k3]$, in which P and Q must synchronise on just one label ('a'), out of the $2^k = 8$ labels of alphabet L. This simple change is sufficient to induce a change in the derived causets, which appear on the r.h.s. of the figure: now some causet events correspond to synchronisations between P and Q, and the causets appear as two separate causal paths that periodically share these events. In spite of the extreme simplicity of these patterns, we may regard them as a first, rudimentary demonstration of how to promote the appearance of macrostructures – the intertwined causal paths of P and Q – in a causet.

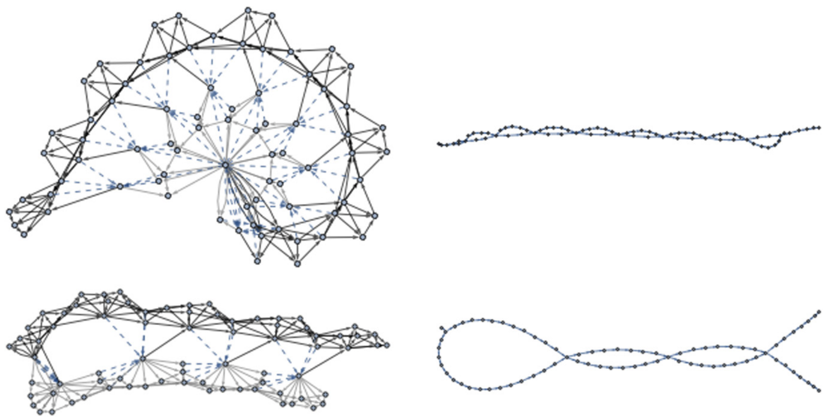

Fig. 5. Causets from deterministic computations of randomly generated bool nets $P[N3, k3]$ and $Q[N3, k3]$ composed by parallel composition $P|\{a\}|Q$. Raw and reduced causet forms are shown next to each other. Again, the different growth patterns of the raw causets (circular vs. linear) are not retained in the reduced causets, which appear essentially equivalent.

The inspection of causets from $P(N3, k3)|[syncLabs]|Q(N3, k3)$ bool net compositions with increasing coupling, i.e. larger set *syncLab*, does not reveal any qualitatively different causet structure, except for an increased probability of deadlock – a bool net waiting to synchronise its transition on a label not available in the transitions of the other. (Note that talking about deadlock *probability* corresponds to the fact that we are creating *random* instances of bool nets, for the given parameter settings, i.e. random sets of boolean functions.)

6 In the graphical rendering of causets, we may render differently (black/gray/dashed) the edges that point to a node, depending on whether that node corresponds to a transition from P, from Q, or from both. This is the criterion adopted for Figs. 4 and 5. As an alternative, we may directly paint the causets node differently, as done for the subsequent figures.

Moving now to higher parameter values we find causets (in their transitively reduced form!) of higher complexity. A curious phenomenon observed here is the dependence of the causet overall shape, or growth symmetry – again, circular or linear – on the coupling factor $|syncLabs|$. Figure 6 shows the causets obtained from $P[N15, k3]||[syncLabs]||Q[N15, k3]$ compositions of a fixed pair of bool nets, with $|syncLabs|$ ranging from 0 to 8, the alphabet size.

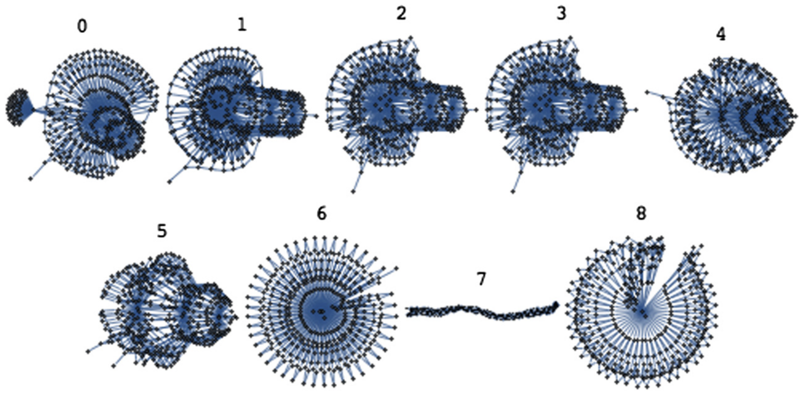

Fig. 6. Causets from parallel compositions $P[N15, k3]||[syncLabs]||Q[N15, k3]$ of two bool nets, with coupling factor $|syncLabs|$ ranging from 0 to 8, as indicated on top of each graph. Only for coupling factor 7 does the causet assume a linear shape.

Curiously, with 7 synchronisation labels the causet develops linearly rather than circularly. Note that with the $(N15, k3)$ parameter setting, deadlocks end up being much less frequent, due to the increased offer of labeled transitions by the two interacting nets.

Our primary motivation for exploring causets from structured, *composite* bool net systems was to spot the emergence of corresponding macro structures in the causets themselves, and the possibility to identify a partition into distinguishable regions. Graph and network theories certainly provide an abundance of tools that might prove useful for formally characterising or measuring 'interesting' causet partitions, but their consideration is beyond the scope of this paper. On the other hand, a simple way to help identifying regions by direct visual inspection is to paint causet nodes with different colors, following some predefined criterion. In the case of bool net parallel compositions an obvious criterion, implicitly suggested already in Fig. 1, is to differentiate among nodes corresponding to independent transitions of P, of Q, and of P and Q jointly. In Fig. 7 we provide two examples (one of them is the second element in the upper row of Fig. 6) in which the three types of causet node are painted, respectively, in white, black and pink.

With respect to the construction of realistic discrete models of physical spacetime, the experiments we have carried out with interacting bool nets, some

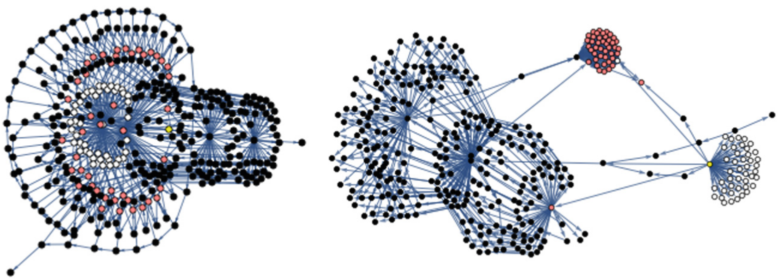

Fig. 7. Painting causet nodes. White and black nodes correspond to transitions performed in interleaving by bool nets P and Q, respectively. Pink nodes correspond to joint transitions of the two bool nets. (Color figure online)

of which have been illustrated here, suggest that the value of the algorithmic causets obtained in this way is mainly metaphorical. The case of cellular automata is somewhat analogous: very few people believe that those automata can, in themselves, fully explain or actually *generate* the whole physical universe, but more people may be convinced that the surprising emergent properties they exhibit, often resembling patterns and processes found in nature, hint at the existence of some deep connection between the computational and the natural universe. Our causets from bool nets indicate that it is possible to obtain some form of macro-structure from deterministic, sequential computations. However, an important limitation of this approach is that the macrostructure is not genuinely *emergent*, in the same way as digital particles unpredictably emerge in some cellular automata, but is an expected consequence of the structure built into the process by us.

This and other limitations of the 'algorithmic causet' effort are discussed in the next, closing section, where we also hint at possible solutions and mention some promising developments.

6 Conclusions

The limitations affecting the causets discussed in this paper represent only a part (and not the most problematic!) of the difficulties that experiments with algorithmic causet construction have faced in the last 15 years. Let us briefly summarise the issues, placing them in a temporal perspective.

The first derivation of causets (or 'causal networks') from the computations of a sequential model, namely a *mobile automaton* operating on a one-dimensional array of cells, has been proposed by Wolfram (see [34], p. 489, and the interactive demo [24]). The crucial limitation of those graphs is that the transitively reduced causet is *totally ordered*, corresponding to a sequence of nodes with both in-degree 1 (except for the root) and out-degree 1. According to the findings on Lorentz invariance in the discrete setting [9,11,27], this is bad.

The total order of events is a direct consequence of the short steps taken by the automaton on the cell array: some of the three (or more) cells read at each step must have been written in the immediately preceding step, yielding events that are causally linked one after the other. (A way to avoid the total order is to consider 'jumping mobile automata' [4].)

The same total-order limitation is suffered by causets derived from Turing Machines (TM), also investigated by Wolfram and others (see demo [35]). A jumping policy would not be effective here, since it is now also the state of the machine head that inevitably plays the role of causality mediator between any pair of adjacent events.

Some of the features exhibited by the *raw* causets derived from the above models appear potentially interesting, even under a Physics perspective. In [4], for example, we have classified the raw causets from the computations of the 4096 elementary TM's – those with 2 states and binary tape. The large majority of them cannot escape the dull fate of a trivial, one-dimensional growth, as in case 7 of Fig. 6, but, interestingly, the toy spacetime produced in 12 cases is two-dimensional and planar: it is flat (Euclidean) for 8 of them, and negatively curved (hyperbolic) for the remaining 4.

It is frustrating to discover elaborate patterns in algorithmic, raw causet, and see them vanish completely after transitive reduction. To mitigate the problem, in [3] we have shown that when transitive reduction is applied to a local area of the raw causet, rather than globally, some patterns in that area may survive, notably digital particles.

Another important limitation that affects the raw and, a fortiori, the reduced causets from several simple models of computation is *planarity*. In [4] it is proved that the causets from general, one-dimensional TMs, from two variants of mobile automata on tape, from string rewrite systems, and from tag systems and cyclic tag systems are all planar, a feature that conflicts with the four dimensions of conventional spacetime.

One way to obtain causets of higher dimensionality consists in increasing the dimensionality of the support on which their parent computation operates, based on the idea that a d-dimensional support should yield a $(d+1)$-dimensional causet, due to the expected contribution of the intervened time dimension. For this reason, causets from two-dimensional TM's and from *network mobile automata* [2] – a model analogous to mobile automata on tape, but operating by rewrite rules on planar, trivalent networks – have been widely investigated in [3], yielding examples of three-dimensional causets. Some improved results were obtained by dropping the planarity requirement for the support network, and by using genetic algorithms [6].[7]

In light of the Occam's razor principle, however, a technique by which the desired causet features (high dimensionality or macro-regions) emerge

[7] Several techniques are available for measuring the dimension of a graph [23]. Unfortunately their estimates may disagree! For the mentioned example we have used the 'node shell growth rate' technique, which provided a dimension 3 estimate but only relative to the node shells centered a the causet root.

spontaneously from a simple and abstract computation should be preferred over ones by which they are built explicitly into the process.

In this paper we have focused on a specific feature of algorithmic causets related to Lorentz invariance: node out-degree growth, as a function of causet size. In this respect, the status of the various classes of algorithmic causets can be summarised as follows, in order of increasing interest.

- Out-degree = 1 for all causet nodes. This is the trivial case of totally ordered causets, which would not even deserve mention if it were not the norm for most of the initial experiments in the field.
- $O(1)$ out-degree growth for all nodes. Causet nodes are related by a proper partial order, but none of their out-degrees can grow beyond a constant value. Graphs assume a typical, uninteresting 'polymer-like', linear structure. When using the indirect causet construction technique in which causality is induced by read-write operations on state variables, an $O(1)$ growth is observed whenever these operations interest all variables in a fair manner. Examples include the discussed causets from bool net computations using the cycling-bit policy, but also many causets from our past experiments with network mobile automata.
- The out-degree grows unbounded (polynomially) for a *finite* number of nodes. An example, referring to bool net computations using the cycling-label policy, was illustrated in Fig. 3, in the lower-right plot.
- The out-degree grows unbounded for an unboundedly growing number of nodes, although not for *all* of them. This feature cannot be observed with causets derived from computations involving a finite and constant number of state variables, like the bool nets considered in the paper, but may be satisfied by dropping that limitation: when new state variables are constantly created, some of them may end up being read infinitely often without being rewritten, thus inducing unbounded growth in the out-degree of their last writer event. Clearly this privilege cannot be enjoyed by *all* causet nodes, since any write operation (we assume they never cease) will permanently stop the out-degree growth of some causet node, as explained in the paper. In network mobile automata, state variables – represented by the faces of the dynamic, planar network – are constantly created by one of the two employed rewrite rules, thus the feature in question can be potentially observed, although in the referenced papers we have not investigated it. Note that the above limitations may affect the causets from virtually any sequential model of computation, given the general validity of the arguments we have provided.
- The out-degree grows unbounded for *all* causet nodes. This is the ideal case observed with stochastic, sprinkled causets, consistent with the Lorentz-invariance requirement. With the *indirect* causet construction technique of concern in this paper, this feature cannot be achieved. On the contrary, it can with an algorithmic, *direct* causet construction technique such as the one described in [8].

Coming now to a more general assessment of the results obtained so far with algorithmic causets and other analogous efforts, in our opinion the most serious problem that this research faces today is that the powerful phenomenon of emergence in computation has never succeeded to ignite a multi-layered cascade of hierarchical levels of emergence, beyond the first level. With Wolfram's ECA's [34], for example, the ground level 0 consists of the boolean functions defining the automata, and a level 1 may emerge from it, e.g. with the digital particles of ECA 110: no level 2 in turn emerges from the interactions of these particles.

This persistent failure to achieve a multi-layered architecture of emergence from simple models of computation seems to indicate that radically new and even more 'creative' ingredients must be involved in the process. We tentatively list three of them, not completely disjoint from one another.

Self-modifying code. Rather that being static, the algorithm that fuels the simulated universe from the bottom could modify itself as it evolves. This is certainly a substantial paradigmatic change, and a well-known concept in Computer Science, but we are not aware of any successful experiments with it in the area of interest here.

Top-down causation. This is regarded as one of the key factors for boosting complexity and variety in the biosphere: the upper level – e.g. a collectivity – induces changes back to the lower level – e.g. the individuals. George Ellis has recently shown how effective and pervasive top-down causation can be, beyond the realm of biology [14].

Emergent causality. Under the usual, reductionist interpretation of the natural world, or of complex artificial systems, all the causal power resides at the lowest, most reduced level of description, leaving no room for causation at the upper levels. But very recent work has shown [18], precisely in the context of the boolean networks described in Subsect. 2.1, and using a formal notion of *Effective Information* based on relative entropy (or 'Kullback-Leibler divergence'), that in some cases the upper levels can supersede the lower ones in causal power.

We hope to witness, in the near future, a new wave of experiments on algorithmic causet construction and 'universe hunting' able to fruitfully incorporate some of the concepts listed above. The task is demanding, its potential results groundbreaking: not only might they reveal a whole new generation of emergent, computation-based phenomena of relevance for complexity studies and fundamental physics, but they could also shed light on the mechanisms at the roots of agency and, ultimately, of consciousness [25].

Acknowledgements. The author expresses his warmest gratitude to Larissa Albantakis for many stimulating exchanges and discussions on notions of causality, Effective Information, Integrated Information. Lack of time has prevented us from completing our joint investigation of the possible applications of these recently proposed informational measures to the synchronous/asynchronous, deterministic/nondeterministic, unstructured/composite boolean networks considered here. This will be the subject of a forthcoming paper.

This work has been partially funded by FQXi Mini-Grant number: FQXi-MGA-1702.

References

1. Benincasa, D.M.T., Dowker, F.: The scalar curvature of a causal set. Phys. Rev. Lett. **104**, 181301 (2010). http://arxiv.org/abs/1001.2725
2. Bolognesi, T.: Planar trinet dynamics with two rewrite rules. Complex Syst. **18**(1), 1–41 (2008)
3. Bolognesi, T.: Algorithmic causets. In: Space, Time, Matter - Current Issues in Quantum Mechanics and Beyond - Proceedings of DICE 2010. IOP (2011). J. Phys. - Conf. Ser
4. Bolognesi, T.: Causal sets from simple models of computation. Int. J. Unconvn. Comput. (IJUC) **7**, 489–524 (2011)
5. Bolognesi, T.: Algorithmic causal sets for a computational spacetime. In: Zenil, H. (ed.) A Computable Universe. World Scientific, Singapore (2013)
6. Bolognesi, T.: Do particles evolve? In: Zenil, H. (ed.) Irreducibility and Computational Equivalence. Springer, Heidelberg (2013). doi:10.1007/978-3-642-35482-3_12
7. Bolognesi, T.: Spacetime computing: towards algorithmic causal sets with special-relativistic properties. In: Adamatzky, A. (ed.) Advances in Unconventional Computing. ECC, vol. 22, pp. 267–304. Springer, Cham (2017). doi:10.1007/978-3-319-33924-5_12
8. Bolognesi, T.: Simple indicators for lorentzian causets. Class. Quantum Gravity **33**(18), 185004 (2016). (41 p.)
9. Bombelli, L., Henson, J., Sorkin, R.D.: Discreteness without symmetry breaking: a theorem May 01 2006. Mod. Phys. Lett. A **24**, 2579–2587 (2009). doi:10.1142/S0217732309031958
10. Bombelli, L., Lee, J., Meyer, D., Sorkin, R.D.: Space-time as a causal set. Phys. Rev. Lett. **59**(5), 521–524 (1987)
11. Bombelli, L., Lee, J., Meyer, D., Sorkin, R.D.: Reply to comment on 'space-time as a causal set'. Phys. Rev. Lett. **60**(7), 656 (1988)
12. Dowker, F., Henson, J., Sorkin, R.: Quantum gravity phenomenology, Lorentz invariance and discreteness. Mod. Phys. Lett. A **19**, 1829–1840 (2004)
13. Zenil, H. (ed.): A Computable Universe. World Scientific, Singapore (2013)
14. Ellis, G. (ed.): How Can Physics Underlie the Mind?. TFC. Springer, Heidelberg (2016). doi:10.1007/978-3-662-49809-5
15. Fredkin, E.: Five big questions with pretty simple answers. IBM J. Res. Dev. **48**(1), 31–45 (2004)
16. Gacs, P., Levin, L.A.: Causal nets or what is a deterministic computation? Inf. Control **51**, 1–19 (1981)
17. Gardner, M.: Mathematical games: the fantastic combinations of John Conway's new solitaire game 'Life'. Sci. Am. **223**(4), 120–123 (1970)
18. Hoel, E.P., Albantakis, L., Tononi, G.: Quantifying causal emergence shows that macro can beat micro. PNAS **110**(49), 19790–19795 (2013)
19. Kauffman, S.A.: Metabolic stability and epigenesis in randomly constructed genetic nets. J. Theoret. Biol. **22**, 437–467 (1969)
20. Langerak, R.: Bundle event structures: a non-interleaving semantics for LOTOS. In: Diaz, M. Groz, R. (eds) FORTE. IFIP Transactions, vol. C-10, pp. 331–346. North-Holland, Amsterdam (1992)
21. Lloyd, S.: Universe as quantum computer. Complexity **3**(1), 32–35 (1997)
22. Meyer, D.A.: The dimension of causal sets. Ph.D. thesis, MIT (1989)
23. Nowotny, T., Requardt, M.: Dimension theory of graphs and networks, July 1997. http://arxiv.org/abs/hep-th/9707082

24. Nussey, A., Tafjord, O.: Causal network generated by a mobile automaton. The Wolfram Demonstrations Project. http://demonstrations.wolfram.com/CausalNetworkGeneratedByAMobileAutomaton/
25. Masafumi, O., Larissa, A., Giulio, T.: From the phenomenology to the mechanisms of consciousness: integrated information theory 3.0. PLoS Comput. Biol. **10**(5), e1003588 (2014)
26. Pearl, J.: Causality: Models, Reasoning and Inference, vol. 29. Cambridge University Press, Cambridge (2000)
27. Rideout, D.P.: HomePage. University of California, San Diego, valid March 2016. http://www.math.ucsd.edu/~drideout/
28. Rideout, D.P., Sorkin, R.D.: A classical sequential growth dynamics for causal sets. Phys. Rev. D **61**, 024002 (1999). http://arxiv.org/abs/gr-qc/9904062 [gr-qc]
29. Saravani, M., Aslanbeigi, S.: On the Causal Set-Continuum Correspondence, 25 May 2014. arXiv:1403.6429v1 [hep-th]
30. Sorkin, R.D.: Causal sets: discrete gravity. In : Gomberoff, A. Marolf, D. (eds.) Proceedings of the Valdivia Summer School, September 2003. http://arxiv.org/abs/gr-qc/0309009
31. Hooft, G.: The cellular automaton interpretation of quantum mechanics, June 2014. http://arxiv.org/abs/1405.1548 [quant-ph]
32. Winskel, G.: Event structures. In: Brauer, W., Reisig, W., Rozenberg, G. (eds.) ACPN 1986. LNCS, vol. 255, pp. 325–392. Springer, Heidelberg (1987). doi:10.1007/3-540-17906-2_31
33. Winskel, G.: An introduction to event structures. In: de Bakker, J.W., de Roever, W.-P., Rozenberg, G. (eds.) REX 1988. LNCS, vol. 354, pp. 364–397. Springer, Heidelberg (1989). doi:10.1007/BFb0013026
34. Wolfram, S.: A New Kind of Science. Wolfram Media Inc., Champaign (2002)
35. Zeleny, E.: Turing machine causal networks. The Wolfram Demonstrations Project. http://demonstrations.wolfram.com/TuringMachineCausalNetworks/
36. Zuse, K.: Calculating space. Technical report, Proj, MAC, MIT, Cambridge, Mass (1970). Technical Translation AZT-70-164-GEMIT. Original title: "Rechnender Raum"

http://www.springer.com/978-3-319-68269-3

ModelEd, TestEd, TrustEd
Essays Dedicated to Ed Brinksma on the Occasion of His 60th Birthday
Katoen, J.-P.; Langerak, R.; Rensink, A. (Eds.)
2017, XII, 391 p. 115 illus., Softcover
ISBN: 978-3-319-68269-3