History Preserving Bisimulation for Contextual Nets^{*}

Paolo Baldan, Andrea Corradini, Ugo Montanari

Dipartimento di Informatica Università di Pisa

Abstract. We investigate the notion of history preserving bisimulation [15, 18, 3] for contextual P/T nets, a generalization of ordinary P/TPetri nets where a transition may check for the presence of tokens without consuming them (non-destructive read operations). A first equivalence, simply called *HP-bisimulation*, is based on Winskel's prime event structures. A finer equivalence, called *RHP-bisimulation* (where "R" stands for "read"), relies on asymmetric event structures [1], a generalization of prime event structures which gives a more faithful account of the dependencies among transition occurrences arising in contextual net computations. Extending the work in [11, 19], we show that HP-bisimulation is decidable for finite *n*-safe contextual nets. Moreover by resorting to causal automata [12] — a variation of ordinary automata introduced to deal with history dependent formalisms — we can obtain an algorithm for deciding HP-bisimulation, instead, remains an open question.

1 Introduction

Contextual nets [14], also called nets with test arcs in [5], with activator arcs in [9] or with read arcs in [20], are a generalization of classical P/T Petri nets where transitions may check for the presence of tokens in the places of the net, without consuming such tokens. More precisely, a transition of a contextual net, besides the usual preconditions and postconditions, may also have some context conditions. The transition is enabled if the current marking covers its preconditions and context conditions. Then the firing of the transition consumes its preconditions, leaves the context unchanged and produces its postconditions. The possibility of faithfully representing a "read-only" access to resources allows contextual nets to model many concrete situations more naturally than classical nets. In recent years they have been used to model concurrent accesses to shared data [16], to provide concurrent semantics to concurrent constraint (CC) programs [13] and to model priorities [8].

Several concurrent semantics for contextual nets based on processes and event structures have been defined in the literature [9, 21, 7, 1]. Relying on such concurrent descriptions of contextual net computations, the aim of this paper is to

^{*} Research partially supported by MURST project Tecniche Formali per Sistemi Software, by TMR Network GETGRATS and by Esprit WG APPLIGRAPH.



Fig. 1. A contextual net N with its AES G and PES P semantics, and a net N' with the same PES semantics.

discuss more abstract semantics by introducing suitable notions of bisimulation, inspired by the classical history preserving bisimulation [15, 18, 3]. The reason why the extension to contextual nets is not straightforward is that, as noticed in [1], a new kind of dependency between transitions, different from causality and symmetric conflict arises in contextual nets, which cannot be represented faithfully in traditional event structures. The new kind of dependency is related to the possibility of reading part of the state without changing it, and thus it is common to all computational formalisms where a step may preserve part of the state. Consider the contextual net N in Fig. 1 with two transitions t_0 and t_1 such that the same place s is a context for t_0 and a precondition for t_1 . As for a conflict of ordinary nets, the firing of t_1 prevents t_0 to be executed, so that t_0 can never follow t_1 in a computation. But the converse is not true, since t_0 can fire before t_1 . This situation can be naturally interpreted as an asymmetric conflict between the two transitions. Equivalently, since t_0 precedes t_1 in any computation where both fire, in such computations t_0 acts as a cause of t_1 , but, differently from a cause, t_0 is not necessary for the firing of t_1 . Hence we can also think of the relation between t_0 and t_1 as a weak form of causal dependency.

A reasonable way of encoding a situation of asymmetric conflict in Winskel's prime event structures (PES's) consists of representing the firing of t_1 with two distinct events (see the PES P in Fig. 1): t'_1 representing the execution of t_1 that prevents t_0 , thus mutually exclusive with t_0 , and t''_1 representing the execution of t_1 after t_0 (thus caused by t_0). Besides this phenomenon of "duplication" of events, which increases the complexity of the structures, what may be unsatisfactory in such encoding is the fact that it looses the information on weak causality. For instance, the net N' in Fig. 1 has the same PES semantics of N. In [1] we proposed a solution based on *asymmetric event structures* (AES's), a generalization of prime event structures where the symmetric binary conflict relation is replaced by a relation \nearrow modeling asymmetric conflict. In an AES a situation of asymmetric conflict as in the net N of Fig. 1 is faithfully represented simply by $t_0 \nearrow t_1$. The AES G corresponding to the net N is depicted in the same figure (asymmetric conflict is represented by a dotted arrow). History preserving bisimulation on ordinary P/T nets relies on the notion of process and of deterministic PES $ev(\pi)$ associated to a process π . Roughly speaking, two nets N_0 and N_1 are history preserving bisimilar if for any process π_0 of N_0 we can find a process π_1 of N_1 such that the underlying deterministic PES's are isomorphic. Moreover whenever π_0 can perform an action becoming a process π'_0 , also π_1 can perform the same action becoming π'_1 and vice versa. The isomorphism between $ev(\pi_0)$ and $ev(\pi_1)$ is required to be extensible to an isomorphism between $ev(\pi'_0)$ and $ev(\pi'_1)$. Informally this means that each event in N_1 can be simulated by an event in N_2 with the same causal history.

When trying to reformulate this notion for contextual nets, we have more than one possibility according to what we decide to observe as history. We can decide that the history of an event e in a process is given by the set of events that precede e in the process, with their relative ordering and without any additional information on the origin of the precedences among events. This amounts to associate to a process π a deterministic PES. The corresponding notion of bisimulation is called *history preserving bisimulation* (*HP-bisimulation*). As already noticed, the PES semantics and thus HP-bisimulation loose the information on weak causality (asymmetric conflict) between events which, working with deterministic processes, is confused with causality. For instance the nets N and N' in Fig. 1, are HP-bisimilar since they have the same PES semantics. In some situations this level of abstraction may be not appropriate: the firing sequence $t_0; t_1$ in N is simulated by $t_0; t''_1$ in N', but in N', differently from what happens in N, there is a flow of information from t_0 to t''_1 . Imagine that tokens represent pieces of data and transitions are functions operating on such data. Then changing the function computed by t_0 in the net N' we influence the output of t''_1 , while in N the output of t_1 does not depend on t_0 . A different notion of bisimulation, discriminating the nets N and N', arises by assuming that the history of an event in a process should record complete information on causality and weak causality. Formally, this is obtained by associating to a process π a deterministic AES. The corresponding finer equivalence is called *read history preserving* bisimulation (RHP-bisimulation). Both HP-bisimulation and RHP-bisimulation coincide with the classical notion when restricted to ordinary P/T nets.

Generalizing the results proved in [11, 19] for ordinary P/T nets, we show that HP-bisimulation is decidable for finite *n*-safe contextual nets. To this aim we resort to causal automata [11], a variation of ordinary automata suited to deal with history-dependent formalisms like CCS with causal or location semantics and the π -calculus with the ordinary or non-interleaving semantics. The (possibly infinite) transition system of processes of a contextual net, which is used to define HP-bisimulation, is abstracted to a causal automaton via a construction which respects (preserves and reflects) bisimilarity. The automaton is finite for finite *n*-safe contextual nets. Hence the HP-bisimilarity of any two finite *n*-safe contextual nets can be checked by applying the general algorithm proposed in [11] to the corresponding causal automata. Since such algorithm relies on the translation of causal automata into ordinary automata, the standard techniques for ordinary transition systems can be used, allowing, for instance, to obtain a minimal realization for a contextual net up to HP-bisimulation. Our attempt of extending the described approach to RHP-bisimulation instead fails. In principle, we are still able to build a (generalized) causal automaton corresponding to a contextual net in a way that respects RHP-bisimulation. However such an automaton is in general infinite also for finite n-safe nets and thus the translation does not suggest an immediate way of checking RHP-bisimulation. The decidability of RHP-bisimulation on contextual nets is left as an open question.

2 Contextual nets, processes and asymmetric event structures

In this section, after introducing the basics of marked contextual P/T nets [16], we present the notions of deterministic occurrence contextual net and process. Then we briefly review prime event structures, and asymmetric event structures as defined in [1].

2.1 Contextual nets

We first recall some notation for multisets. Let A be a set; a *multiset* of A is a function $M : A \to \mathbb{N}$. It is called finite if $\{a \in A : M(a) > 0\}$ is finite. The set of finite multisets of A is denoted by μ_*A . The usual operations and relations on multisets, like multiset union + or multiset difference -, are used. We write $M \leq M'$ if $M(a) \leq M'(a)$ for all $a \in A$. If $M \in \mu_*A$, we denote by $\llbracket M \rrbracket$ the multiset defined, for all $a \in A$, as $\llbracket M \rrbracket(a) = 1$ if M(a) > 0, and $\llbracket M \rrbracket(a) = 0$ otherwise. With abuse of notation we will write $a \in M$ for M(a) > 0 and $M \cap M'$ for the set $\{a \in A : M(a) > 0 \land M'(a) > 0\}$. A multirelation $f : A \to B$ is a multiset of $A \times B$. It is called *finitary* if $\{b \in B : f(a, b) > 0\}$ is a finite set for all $a \in A$. A finitary multirelation f induces in an obvious way a function $\mu f : \mu_*A \to \mu_*B$, defined as $\mu f(M)(b) = \sum_{a \in A} M(a) \cdot f(a, b)$ for $M \in \mu_*A$ and $b \in B$. In the sequel we will implicitly assume that all multirelations are finitary.

Definition 1 ((labelled, marked) contextual net). A (marked) contextual Petri net (c-net) is a tuple $N = \langle S, T, F, C, m \rangle$, where

- -S is a set of places and T is a set of transitions;
- $-F = \langle F_{pre}, F_{post} \rangle$ is a pair of multirelations from T to S;
- -C is a multirelation from T to S, called the context relation;
- -m is a finite multiset of S, called the initial marking.

The c-net is called finite if T and S are finite sets. Without loss of generality, we assume $S \cap T = \emptyset$. Moreover, we require that for each transition $t \in T$, there exists a place $s \in S$ such that $F_{pre}(t,s) > 0$. A labelled c-net is a pair $\langle N, l_N \rangle$ where N is a c-net and $l_N : T \to Act$ is a function from T to a set of labels Act.

In the following when considering a c-net N, we implicitly assume that $N = \langle S, T, F, C, m \rangle$ and, if N is labelled, we denote by l_N its labelling function. Furthermore, throughout the paper we consider a fixed set of labels Act.

As usual, given a finite multiset of transitions $A \in \mu_* T$ we write $\bullet A$ for its *pre-set* $\mu F_{pre}(A)$ and A^{\bullet} for its *post-set* $\mu F_{post}(A)$. Moreover, by \underline{A} we denote the *context* of A, defined as $\underline{A} = \mu C(A)$. The same notation is used to denote the functions from S to the powerset $\mathcal{P}(T)$ defined as, for $s \in S$, $\bullet s = \{t \in T : F_{post}(t,s) > 0\}$, $s^{\bullet} = \{t \in T : F_{pre}(t,s) > 0\}$, $\underline{s} = \{t \in T : C(t,s) > 0\}$.

Let N be a contextual net. For a finite multiset of transitions A to be enabled by a marking M, it is sufficient that M contains the pre-set of A and at least one *additional* token in each place of the context of A. This corresponds to the intuition that a token in a place can be used as context concurrently by many transitions and with multiplicity greater than one by the same transition. Formally, a finite multiset of transitions $A \in \mu_*T$ is *enabled* by a marking $M \in$ μ_*S if $\bullet A + \llbracket A \rrbracket \leq M$. In this case, to indicate that the execution of A in M produces the new marking $M' = M - \bullet A + A^{\bullet}$ we write $M \llbracket A
angle M'$. We call $M \llbracket A
angle M'$ a step (or a firing when it involves just one transition).

A marking M of a c-net N is called *reachable* if there is a finite step sequence in N leading to M from the initial marking, i.e., $m[A_0\rangle M_1[A_1\rangle M_2 \dots [A_n\rangle M$. A c-net is called *n*-safe if for any reachable marking M each place contains at most n tokens, namely $M(s) \leq n$ for all $s \in S$. A 1-safe net will be often called simply safe.

2.2 Deterministic occurrence c-nets and processes

As for ordinary nets a process of a c-net is defined as an occurrence c-net with a mapping to the original net. Occurrence c-nets are safe c-nets such that the dependency relations between transitions satisfy suitable acyclicity and wellfoundedness requirements.

Definition 2 (causality). Let N be a safe c-net. The causality relation $<_N$ is the least transitive relation on $S \cup T$ such that

1. if $s \in {}^{\bullet}t$ then $s <_N t$; 2. if $s \in t^{\bullet}$ then $t <_N s$; 3. if $t^{\bullet} \cap \underline{t'} \neq \emptyset$ then $t <_N t'$.

Given $x \in S \cup T$, we write $\lfloor x \rfloor$ for the set of causes of x in T, defined as $\lfloor x \rfloor = \{t \in T : t \leq_N x\} \subseteq T$, where \leq_N is the reflexive closure of $<_N$.

Note that causality is defined as for ordinary nets, with an additional clause stating that transition t causes t' if it generates a token in a context place of t'.

Definition 3 (asymmetric conflict). Let N be a safe c-net. The asymmetric conflict relation \nearrow_N is defined as

 $t \nearrow_N t'$ iff $\underline{t} \cap \bullet t' \neq \emptyset$ or $(t \neq t' \land \bullet t \cap \bullet t' \neq \emptyset)$ or $t <_N t'$.

In our interpretation, $t \nearrow_N t'$ if t' prevents t to be fired or, equivalently, if t must precede t' in each computation where both transitions fire. Informally, if $fire_C(t)$ indicates that t fires in a computation C and $prec_C(t, t')$ indicates that t precedes t' in C, then $t \nearrow t'$ means that

for all computations C, if $fire_C(t)$ and $fire_C(t')$ then $prec_C(t, t')$ (†)

The discussion in the introduction suggests that, in an acyclic safe net, (\dagger) is surely satisfied when the same place s appears as context for t and as precondition for t'. But (\dagger) is trivially true (with t and t' in interchangeable roles) when t and t' have a common precondition, since they never fire in the same computation. This corresponds to the intuition that an ordinary symmetric conflict amounts to an asymmetric conflict in both directions. Finally, (\dagger) is weaker than the condition expressing causality and thus it is satisfied when t < t'.

We are now able to introduce deterministic occurrence c-nets.

Definition 4 ((deterministic) occurrence c-nets). A deterministic occurrence c-net is a safe c-net $O = \langle S, T, F, C, m \rangle$ such that

- 1. each place $s \in S$ is in the post-set of at most one transition, i.e. $|\bullet s| \leq 1$;
- 2. \nearrow_O is well-founded and finitary; thus $(\nearrow_O)^*$ and \leq_O are finitary partial orders;¹
- 3. $m = \{s \in S : \bullet s = \emptyset\}.$

While conditions (1) and (3) are standard, condition (2) deserves some comments. First, it implies the acyclicity of \nearrow_O , which can be interpreted as a conflict freeness property. In fact, if some transitions t_0, \ldots, t_n form a \nearrow_O -cycle $t_0 \nearrow_O t_1 \nearrow_O \ldots \nearrow_O t_n \nearrow_O t_0$ then they cannot fire together in the same computation since each one should precede the others. Hence cycles of asymmetric conflict can be thought of as a kind of conflicts on sets of events. Condition (2) also requires the absence of infinite descending chains of \nearrow_O , ensuring that each transition must be preceded only by finitely many other transitions.

We will denote by $\min(O)$ and $\max(O)$ the sets of minimal and maximal places of O w.r.t. the partial order \leq_O . An occurrence c-net O will be often denoted as $\langle S, T, F, C \rangle$, not mentioning the initial marking m which is uniquely determined as $\min(O)$.

In a deterministic occurrence c-net *all* transitions can fire in a single computation, in any order compatible with the asymmetric conflict relation. Hence, as for ordinary nets, a deterministic process of a c-net, representing a concurrent computation of the net, is defined as a deterministic occurrence c-net with a mapping to the original net. Such a mapping allows one to transform each firing (step) sequence of the occurrence net into a firing sequence of the original net.

Definition 5 ((deterministic) process). A deterministic process of a c-net N is a mapping $\pi : O_{\pi} \to N$, where O_{π} is a deterministic occurrence c-net and π is a strong c-net morphism, namely a pair of total functions $\pi = \langle \pi_S : S_N \to S_{O_{\pi}}, \pi_T : T_N \to T_{O_{\pi}} \rangle$ such that the pre-set, post-set and context of each transition are preserved. A process π is called marked if π_S preserves also the initial marking of the net.

¹ A relation $r \subseteq X \times X$ is called *finitary* if for any $x \in X$ the set $\{y \in X : y r x\}$ is finite. Furthermore r^* denotes the reflexive and transitive closure of a relation r.

We will denote by $\min(\pi)$ and $\max(\pi)$ the sets of places $\min(O_{\pi})$ and $\max(O_{\pi})$. Similarly the relations $\leq_{O_{\pi}}$ and $\nearrow_{O_{\pi}}$ will be denoted simply as \leq_{π} and \nearrow_{π} . Any marked process of the net N having an empty set of transitions is called an *initial process* of N (observe that all the initial processes of a net are isomorphic). Since in this paper we will only deal with deterministic occurrence c-nets and processes, the qualification "deterministic" will be often omitted. Some examples of deterministic marked processes of a c-net can be found in Fig. 2 (for the moment ignore the fact that processes are partly dotted).

Hereinafter we will consider only finite processes, i.e., processes with a finite underlying occurrence net. It is possible to show that a marking M of a c-net N is reachable iff there is some finite marked process π such that $M = \mu \pi_S(\max(\pi))$.

2.3 Prime and asymmetric event structures

Event structures [22] are a simple event based model of concurrent computations in which events are considered as atomic and instantaneous steps, which can appear only once in a computation. Recall that a (labelled) prime event structure with binary conflict (PES) is a tuple $\langle E, \leq, \#, l \rangle$ consisting of a set E of events endowed with two binary relations: the partial order \leq modelling causality and the relation # modeling conflict, which is symmetric, irreflexive and hereditary w.r.t. causality. The last component $l: E \to Act$ is the labelling function over the fixed set of labels Act.

To faithfully model the dependencies in c-net computations, in [1] we introduced *asymmetric event structures*, a generalization of PES's where the symmetric conflict relation is replaced by a relation \nearrow representing *weak causality* or *asymmetric conflict*.

Definition 6 (asymmetric event structure). A (labelled) asymmetric event structure (AES) is a tuple $G = \langle E, \leq, \nearrow, l \rangle$, where E is a set of events, $l : E \rightarrow$ Act is the labelling function (with Act fixed set of labels) and \leq, \nearrow are binary relations on E called causality and asymmetric conflict, respectively, such that:

1. \leq is a partial order and $\lfloor e \rfloor = \{e' \in E : e' \leq e\}$ is finite for all $e \in E$; 2. $\langle \subseteq \nearrow$ and \nearrow is acyclic on $\lfloor e \rfloor$ for all $e \in E$.

We do not discuss in detail the notion of AES. It is basically a "stateless" counterpart of (possibly nondeterministic) occurrence c-nets and most of the considerations done for occurrence c-nets also apply to AES's. In particular the conflict relation is induced by cycles of asymmetric conflict.

In this paper we are interested only in finite deterministic event structures. A (deterministic) prime event structure is a conflict free PES, namely a PES $\langle E, \leq, \#, l \rangle$ such that $\# = \emptyset$. It will be always denoted simply by $\langle E, \leq, l \rangle$. Similarly a (deterministic) asymmetric event structure is an AES $\langle E, \leq, \nearrow, l \rangle$ such that the asymmetric conflict relation is well-founded and finitary (in particular the relation \nearrow is acyclic, a property corresponding to conflict freeness). Observe that for finite AES's such conditions reduce to the acyclicity of \nearrow . For both deterministic PES's and AES's the entire set E of events is a configuration of the event structure, meaning that all the events in E can be executed in a single computation of the modelled system.

The notion of isomorphism for PES's and AES's is defined in the obvious way.

History preserving bisimulation on contextual nets 3

As mentioned in the introduction, history preserving bisimulation is a behavioural equivalence which takes into account the dependencies among events. Roughly speaking it equates two ordinary nets if each action of the first net can be simulated by an action of the second net with the same history, and vice versa. This section generalizes this idea to the setting of contextual nets, where, according to the chosen notion of history, two different formulations of history preserving bisimulation arise: HP-bisimulation, relying on PES's, which observes only the precedences between events, and RHP-bisimulation, based on AES's, which instead distinguishes causality from weak causality.

A basic ingredient for the definition of history preserving bisimulation is a transition system, associated to each c-net, where states are processes.

Definition 7 (process moves). Given two processes π and π' of a labelled c-net N, we write $\pi \xrightarrow{a}{\epsilon} \pi'$, saying that π moves to π' performing action a, if

- $\begin{array}{l} -T_{\pi'} = T_{\pi} \cup \{e\}, \ \text{with} \ e \notin T_{\pi} \ \text{and} \ l_N(\pi'_T(e)) = a; \\ -S_{\pi} \subseteq S_{\pi'}; \\ -\pi_S, \ \pi_T, \ F_{\pi} \ \text{and} \ C_{\pi} \ \text{are the restrictions to} \ O_{\pi} \ \text{of the components of } \pi'. \end{array}$

Fig. 2 represents a sequence of processes of a c-net N such that each π_i moves to π_{i+1} . For example, $\pi_0 \xrightarrow[e_0]{c} \pi_1$ and $\pi_1 \xrightarrow[e_1]{b} \pi_2$.

To each process π of a c-net we can naturally associate a (deterministic) PES having the transitions of the underlying occurrence net as events and the transitive closure of the asymmetric conflict relation as causality: this corresponds to confuse the weak causality determined by the precedences induced by contexts and the "strong" causality deriving from the flow of information. Alternatively, we can associate to the process a deterministic AES, which keeps weak causality and "strong" causality distinct.

Definition 8 (event structures for processes). Let π be a process of a labelled c-net N. The PES associated to π is defined as:

$$ev(\pi) = \langle T_{\pi}, (\nearrow_{\pi})^*, l_N \circ \pi_T \rangle$$

The AES associated to π is instead

$$aev(\pi) = \langle T, \leq_{\pi}, \nearrow_{\pi}, l_N \circ \pi_T \rangle$$

Based on the notions of process and of (asymmetric) event structure associated to a process, history preserving (HP-) bisimulation and read history preserving (RHP-) bisimulation are readily defined.







Fig. 2. A contextual net N and a sequence of process moves starting from an initial process. The process mappings are specified by the labelling of the items. For any process the non dotted part represents the corresponding partial process.

Definition 9 (HP-bisimulation). Let N_1 and N_2 be labelled c-nets. An HP-simulation \mathcal{R} of N_1 in N_2 is a set of triples $\langle \pi_1, f, \pi_2 \rangle$ where π_i is a marked process of N_i for $i \in \{1, 2\}$, and $f : ev(\pi_1) \to ev(\pi_2)$ is an isomorphism of PES's, such that

- 1. $\langle \pi_0(N_1), \emptyset, \pi_0(N_2) \rangle \in \mathcal{R}$, with $\pi_0(N_i)$ initial process of N_i for $i \in \{1, 2\}$;
- 2. $\langle \pi_1, f, \pi_2 \rangle \in \mathcal{R} \land \pi_1 \xrightarrow{a} \pi'_1 \Rightarrow \pi_2 \xrightarrow{a} \pi'_2 \land \langle \pi'_1, f', \pi'_2 \rangle \in \mathcal{R} \land f'_{|ev(\pi_1)} = f.$

An HP-bisimulation between N_1 and N_2 is a set of triples \mathcal{R} such that \mathcal{R} and $\mathcal{R}^{-1} = \{ \langle \pi_2, f^{-1}, \pi_1 \rangle : \langle \pi_1, f, \pi_2 \rangle \in \mathcal{R} \}$ are HP-simulations. The labelled c-nets N_1 and N_2 are HP-bisimilar, written $N_1 \sim_{hp} N_2$, if there is an HP-bisimulation \mathcal{R} between N_1 and N_2 .

Definition 10 (RHP-bisimulation). An RHP-bisimulation \mathcal{R} between two labelled c-nets N_1 and N_2 is a set of triples $\langle \pi_1, f, \pi_2 \rangle$ satisfying the same conditions as HP-bisimulation, but with $ev(\pi_i)$ replaced by $aev(\pi_i)$. The labelled c-nets N_1 and N_2 are RHP-bisimilar, written $N_1 \sim_{rhp} N_2$, if there is an RHP-bisimulation \mathcal{R} between N_1 and N_2 .

Any RHP-bisimulation for two c-nets N_1 and N_2 is also an HP-bisimulation. In fact if π_1 and π_2 are processes of N_1 and N_2 , respectively, and $f : aev(\pi_1) \rightarrow aev(\pi_2)$ is an isomorphism of AES's then it is easy to see that f is also an isomorphism of PES's between $ev(\pi_1)$ and $ev(\pi_2)$. Therefore $N_1 \sim_{rhp} N_2$ implies $N_1 \sim_{hp} N_2$. The converse implication, instead, does not hold. For instance, if N and N' are the c-nets in Fig. 1, then $N \sim_{hp} N'$, while $N \not\sim_{rhp} N'$.

4 Causal automata

Causal automata are a generalization of ordinary automata introduced in [11] as an appropriate model for history dependent formalisms (see also [12], where more general models, called HD-automata, are presented). Causal automata extend ordinary automata by allowing sets of names to appear explicitly in the states and labels of the automata. The names are local, namely they do not have a global identity, and the correspondence between the names of the source and those of the target states of each transition is specified explicitly. This allows for a compact representation of systems since states differing only for the concrete identity of the names can be identified. Moreover causal automata provide a mechanism for the generation of new names: the problem of choosing a fresh name simply disappears in this formalism where a new name is simply a name which does not correspond to any name in the source state. In the specific case of Petri nets, names are identities of transitions in a process (events) and the correspondence between names allows to represent causal dependencies.

Definition 11 (causal automaton). Let \mathcal{N} be a fixed infinite countable set of names (event names) and let Act be a fixed set of labels. A causal automaton is a tuple $\mathcal{A} = \langle Q, n, \longmapsto, q_0 \rangle$, where

- -Q is the set of states;
- $-n: Q \to \mathcal{P}_{fin}(\mathcal{N})$ is a function associating to each state a finite set of names; $-\longmapsto$ is a set of transitions, each of the form $q \vdash_{\overline{\mathcal{M}}}^{a} q'$, with
 - q, q' the source and target states;
 - $a \in Act$ the label;
 - $M \subseteq n(q)$ the set of dependencies of the transition;
 - $\sigma: n(q') \hookrightarrow n(q) \cup \{\star\}$ the injective inverse renaming function;
- $-q_0 \in Q$ is the initial state; it is required that $n(q_0) = \emptyset$.

For each state $q \in Q$ the set of names n(q) is used to represent the past events which can (but not necessarily will) be referenced by future transitions. Conceptually, each transition $q \vdash_{M} \rightarrow_{\sigma} q'$ depends on the past events mentioned in M. Due to the local scope of names, the function $\sigma : n(q') \hookrightarrow n(q) \cup \{\star\}$ is needed to relate the names of the target state to those of the source. The event mapped to \star (if any) represents the new event generated by the considered transition. In the following the components of a causal automaton will be often denoted by using the name of the automaton as subscript.

The notion of bisimulation on causal automata (CA-bisimulation) takes into account the fact that a state has attached a set of local names. Hence a bisimulation not only relates states, but also the corresponding sets of local names.

Definition 12 (CA-bisimulation). Let \mathcal{A} and \mathcal{B} be two causal automata. A CA-simulation \mathcal{R} of \mathcal{A} in \mathcal{B} is a set of triples $\langle q, \delta, p \rangle$, where $q \in Q_{\mathcal{A}}$, $p \in Q_{\mathcal{B}}$ and δ is a partial injective function from $n_{\mathcal{A}}(q)$ to $n_{\mathcal{B}}(p)$, such that

- 1. $\langle q_{0,\mathcal{A}}, \emptyset, q_{0,\mathcal{B}} \rangle \in \mathcal{R};$ 2. if $\langle q, \delta, p \rangle \in \mathcal{R}$ and $q \xrightarrow{a}_{\mathcal{M}} q'$ in \mathcal{A} then
 - $p \xrightarrow[\delta(M)]{a} p' \text{ in } \mathcal{B} \text{ for some } p' \text{ and}$
 - $-\langle q', \delta', p' \rangle \in \mathcal{R}$ for some δ' such that $\delta^* \circ \sigma = \rho \circ \delta'$, where δ^* is defined as $\delta \cup \{(\star, \star)\}$ (see the diagram below).

$$n_{\mathcal{A}}(q) \cup \{\star\} \xrightarrow{\delta^{^{}}} n_{\mathcal{B}}(p) \cup \{\star\}$$

$$\sigma \uparrow \qquad \uparrow^{\rho}$$

$$n_{\mathcal{A}}(q') \xrightarrow{\delta'} n_{\mathcal{B}}(p')$$

A CA-bisimulation between \mathcal{A} and \mathcal{B} is a set of triples \mathcal{R} such that \mathcal{R} and $\mathcal{R}^{-1} = \{\langle p, \delta^{-1}, q \rangle : \langle q, \delta, p \rangle \in \mathcal{R} \}$ are CA-simulations. The automata \mathcal{A} and \mathcal{B} are CA-bisimilar, written $\mathcal{A} \sim_{ca} \mathcal{B}$, if there exists a bisimulation \mathcal{R} between \mathcal{A} and \mathcal{B} .

In [11] an algorithm has been proposed for checking the CA-bisimilarity of (finite) causal automata. Given a causal automaton \mathcal{A} , after removing the "unnecessary" names from the states of the automaton, the basic step of the algorithm



Fig. 3. Diagrams for abstraction homomorphisms.

constructs an ordinary labelled transition system $Unf(\mathcal{A})$, called the unfolding of \mathcal{A} , such that $\mathcal{A} \sim_{ca} \mathcal{B}$ iff the associated transition systems $Unf(\mathcal{A})$ and $Unf(\mathcal{B})$ are bisimilar. Then standard algorithms (e.g., a partition/refinement algorithm) can be used to verify bisimilarity on the ordinary transition systems or to obtain a minimal equivalent transition system.

Abstraction homomorphisms [4], also called zig-zag morphisms [17] or transition preserving homomorphisms [6], are defined in the setting of ordinary automata as morphisms which "preserve" and "reflect" transitions. The existence of an abstraction homomorphism ensures that the source an target automata are bisimilar. The next definition generalizes this idea to causal automata.

Definition 13 (abstraction homomorphism). Let \mathcal{A} and \mathcal{B} be causal automata. An abstraction homomorphism $h : \mathcal{A} \to \mathcal{B}$ is a pair $h = \langle h, \{h_q\}_{q \in Q_{\mathcal{A}}} \rangle$ where $h : Q_{\mathcal{A}} \to Q_{\mathcal{B}}$ is a function and for all $q \in Q_{\mathcal{A}}, h_q : n_{\mathcal{B}}(h(q)) \to n_{\mathcal{A}}(q)$ is an injective function, such that $h(q_{0,\mathcal{A}}) = q_{0,\mathcal{B}}$ and

- $if q \stackrel{a}{\longmapsto_{M}} q' in \mathcal{A} then h(q) \stackrel{a}{\underset{h_{q}^{-1}(M)}{\longrightarrow}} h(q') in \mathcal{B}, with \sigma \circ h_{q'} = h_{q}^{\star} \circ \rho (see Fig. 3.(a));$
- $if h(q) \xrightarrow[M]{a} \rho p' in \mathcal{B} then q \xrightarrow[h_q(M)]{\sigma} q' in \mathcal{A} for some q', with h(q') = p' and$ $\sigma \circ h_{q'} = h_a^* \circ \rho (see Fig. 3.(b)).$

Intuitively, via an abstraction homomorphism $h : \mathcal{A} \to \mathcal{B}$ several states of \mathcal{A} can collapse into a single state of \mathcal{B} , in a way that respects the behaviour and the naming. Also in this setting, the existence of an abstraction homomorphism $h : \mathcal{A} \to \mathcal{B}$ is sufficient to conclude the bisimilarity of \mathcal{A} and \mathcal{B} .

Lemma 1. Let \mathcal{A} and \mathcal{B} be causal automata. If there exists an abstraction homomorphism $h : \mathcal{A} \to \mathcal{B}$ then $\mathcal{A} \sim_{ca} \mathcal{B}$.

It is worth observing that, as for ordinary automata, the above lemma does not provide a necessary condition. Indeed, in [12], following the approach of [10], abstraction homomorphisms have been described as open maps in a category of causal automata, and CA-bisimilarity has been given a characterization by means of spans of open maps.

5 Causal automata for contextual net bisimulations

As in the ordinary case, the definition of (R)HP-bisimulation for contextual nets relies on the transition system of marked processes and process moves, which is infinite for any non-trivial net exhibiting a cyclic behaviour. In the case of ordinary nets, the solution proposed in the literature for deciding history preserving bisimulation on significant subclasses of nets consists of recording only the part of a process which is, in a sense, relevant for the current state. More precisely, the techniques in [19] for safe nets and in [11] for *n*-safe P/T nets, basically rely on the idea of restricting the attention only to the set of events which produced at least one token in the current state and to the causal ordering among them. The corresponding structures are called *configurations* in [11] and ordered markings in [19]. The bisimulation relation defined on the transition system having such configurations as states, called incremental bisimulation, is shown to coincide with history preserving bisimulation. Since a finite n-safe net only has finitely many configurations (up to isomorphism), incremental bisimulation (and thus also history preserving bisimulation) is decidable for this class of nets. Furthermore, the paper [11] shows how it is possible to associate to each finite n-safe net N a finite causal automaton $\mathcal{A}(N)$ such that two nets N_1 and N_2 are bisimilar if and only if $\mathcal{A}(N_1) \sim_{ca} \mathcal{A}(N_2)$. Then the general algorithm for causal automata mentioned in Section 4 can be used to check the bisimilarity of nets and to construct a minimal realization.

In this section we address the problem of extending such a method to contextual nets. We prove that HP-bisimulation is still decidable for finite n-safe contextual nets by showing how a finite causal automaton can be associated to such nets via a construction which respects HP-bisimilarity. Instead, when considering RHP-bisimulation some serious problems arise: the natural extension of the described approach produces a causal automaton which may be infinite also for finite safe c-nets. Hence the decidability of RHP-bisimulation remains an open question.

5.1 HP-bisimulation

As mentioned before, to construct a finite causal automaton, or, in general, a finite transition system allowing to decide history preserving bisimulation for ordinary nets the leading idea is that not all the information carried by a process is relevant for deciding bisimulation. Hence processes may be replaced by more compact structures where part of the past history is discarded.

As one would expect, when considering HP-bisimulation on c-nets, one must keep information not only about the events which produced a token in the current marking ("producers"), but also about the events which read a token in the current marking ("readers"). Fortunately, among the readers, which can be unbounded even for a safe net, only the maximal ones play a significant role, while the others can be safely discarded.

Definition 14 (producers and (maximal) readers). Given a process π of a c-net N, we define

- the set of producers
- $p(\pi) = \{t \in T_{\pi} : t^{\bullet} \cap \max(\pi) \neq \emptyset\};\$
- the set of readers $r(\pi) = \{t \in T_{\pi} : \underline{t} \cap \max(\pi) \neq \emptyset\};\$ - the set of maximal readers $mr(\pi) = \{t \in r(\pi) : \exists s \in \underline{t} \cap \max(\pi). t \text{ is } \leq_{\pi} -\text{maximal in } \underline{s}\}.$

For instance the set of producers of the process π_4 in Fig. 2 is $p(\pi_4) = \{e_0, e_3\}$, its set of readers is $r(\pi_4) = \{e_1, e_2, e_3\}$, while the set of maximal readers is $mr(\pi_4) = \{e_1, e_3\}.$

As expressed by the proposition below, for a finite n-safe c-net N the sets $p(\pi)$ and $mr(\pi)$, with π ranging over the marked processes of N, are bounded.

Proposition 1. Let N be a finite n-safe c-net. Then, for any marked process π of N, we have $|p(\pi)| \leq n \cdot |S_N|$ and $|mr(\pi)| \leq (n \cdot |S_N|)^2$.

We next define partial processes, which are aimed at representing truncations of marked processes where only a relevant part for discriminating non HP-bisimilar states is kept, namely the producers and the maximal readers.

Definition 15 (partial process). A partial process γ of a c-net N is an unmarked process $\gamma: O_{\gamma} \to N$, where $T_{\gamma} = p(\gamma) \cup mr(\gamma)$.

Observe that any initial process for a c-net N is a partial process of N, since it has an empty set of productions.

To each (marked or unmarked) process π of a net N we associate in the obvious way a partial process which is obtained by keeping only the producers and the maximal readers of π (see Fig. 2).

Definition 16. Each process π of a c-net N naturally induces a partial process, denoted $\gamma(\pi)$, such that $O_{\gamma(\pi)} = \langle S', T', F', C' \rangle$, where

- $T' = p(\pi) \cup mr(\pi);$
- $-S' = \max(\pi) \cup \bigcup \{ \bullet t : t \in T' \};$ F', C' and the mapping to N are the restrictions of F_{π} , C_{π} and π .

We next introduce the move relation on partial processes, leading to a transition system of partial processes which is then exploited in the construction of the causal automaton associated to a c-net.

Definition 17 (partial processes moves). Given two partial processes γ and γ' we write $\gamma \xrightarrow{a}{c} \gamma'$, and we say that γ moves to γ' performing the action a, iff $\gamma \xrightarrow{a}{e} \pi$ (as a generic process, according to Definition 7) and $\gamma(\pi) = \gamma'$.

The process move $\gamma \stackrel{a}{\longrightarrow} \pi$ is called the process move underlying the partial process move $\gamma \xrightarrow[e]{e} \gamma'$. Considering only the non-dotted parts, Fig. 2 represents a sequence of partial process moves starting from the initial process π_0 .

To each (partial) process move we associate the set of maximal (weak or strong) causes of the executed transition, which will play a basic role in the definition of the automaton. In fact, to observe the partial order associated to an evolving computation it is sufficient to look, step by step, only at the immediate maximal causes of each single transition (the other dependencies being implicitly given by the transitivity of the partial order).

Definition 18 (immediate and maximal causes). The set of immediate (weak or strong) causes of a process move $\pi \xrightarrow[e]{e} \pi'$ is defined as $\mathsf{IC}(\pi \xrightarrow[e]{e} \pi') = \{t \in T_{\pi} : t^{\bullet} \cap (\underline{e} \cup \bullet e) \neq \emptyset \lor \underline{t} \cap \bullet e \neq \emptyset\}$. We denote by $\mathsf{MC}(\pi \xrightarrow[e]{e} \pi')$ the set of maximal (immediate) causes, namely the subset of \nearrow_{π} -maximal elements of $\mathsf{IC}(\pi \xrightarrow[e]{e} \pi')$. The sets of immediate and maximal causes of a partial process move are defined in the obvious way by resorting to the underlying process move.

For example, for the move $\pi_4 \xrightarrow[e_4]{d} \pi_5$ we have that $\mathsf{IC}(\pi_4 \xrightarrow[e_4]{d} \pi_5) = \{e_0, e_1, e_2, e_3\}$, while $\mathsf{MC}(\pi_4 \xrightarrow[e_4]{d} \pi_5) = \{e_0, e_1, e_3\}$.

It is possible to show that if a partial process γ of a c-net N is reachable from an initial process via a finite sequence of moves, then $\gamma = \gamma(\pi)$ for some marked process π of N. When the c-net N is n-safe and finite, the definition of $\gamma(\pi)$ and Proposition 1 allow us to conclude the validity of the following result.

Lemma 2. For any n-safe finite c-net the set of partial processes reachable from the initial process (and taken up to isomorphism) is finite.

We are now ready to present the construction of the causal automaton associated to a c-net for checking HP-bisimilarity. To obtain a "compact" automaton (with a finite number of states for finite *n*-safe c-nets) we must consider partial processes up to isomorphism. To this aim we fix a standard representative in each class of isomorphic partial processes. Furthermore we consider a normalization function norm such that for any partial process γ , norm $(\gamma) = \langle \gamma', i \rangle$, where γ' is the standard representative in the isomorphism class of γ and $i : \gamma' \to \gamma$ is a chosen process isomorphism. We assume that the names of the transitions in any (partial) process γ are taken from \mathcal{N} , namely that $T_{\gamma} \subseteq \mathcal{N}$.

Definition 19 (causal automaton for HP-bisimulation). Let N be a labelled contextual net. The HP-causal automaton associated to N is the automaton $\mathcal{A}_{hp}(N) = \langle Q, n, \mapsto, q_0 \rangle$, having (standard representatives of) partial processes as states. The initial state q_0 is the standard representative γ_0 of the initial processes of N and whenever $\gamma \in Q$ then

$$\begin{aligned} &-n(\gamma) = T_{\gamma}; \\ &-if \ \gamma \ \frac{a}{e} \ \gamma' \ and \ \mathsf{norm}(\gamma') = \langle \gamma'', i \rangle \ then \ \gamma'' \in Q \ and \ \gamma \ \underset{M}{\longmapsto}_{\sigma} \ \gamma'' \ where \\ &\bullet \ \sigma : T_{\gamma''} \hookrightarrow T_{\gamma} \cup \{\star\} \ is \ defined \ as \ \sigma = (id_{T_{\gamma}} \cup \{(e, \star)\}) \circ i_{T}; \\ &\bullet \ M = \mathsf{MC}(\gamma \ \frac{a}{e} \ \gamma'). \end{aligned}$$

Observe that the renaming function in a transition of the causal automaton is obtained from the isomorphism given by the normalization function norm, simply by redirecting the new name e to \star . As anticipated, the maximal causes of a process move are used as dependencies in the automaton transition.

The states of the automaton are standard representatives of partial processes reachable from the initial partial process. Hence by Lemma 2 we derive that for any finite *n*-safe c-net the above defined automaton has a finite number of states (and clearly, being the net finite, it has also a finite number of transitions leaving from each state). Vice versa, if the net is not *n*-safe for some *n*, then the automaton will have an infinite number of states.

To effectively build the automaton we can perform an inductive construction based on Definition 19. The only thing to observe is that, given a partial process γ , there might be infinitely many moves $\gamma \xrightarrow[e]{e} \gamma'$ since the event e can be chosen arbitrarily among the unused events in \mathcal{N} . However, without loss of generality, we can limit our attention only to some partial process moves, called the *representative* moves, where the newly generated name is chosen in a canonical way. For instance we can suppose that the set of names \mathcal{N} is well-ordered and call a transition $\gamma \xrightarrow[e]{a} \gamma'$ representative if $e = \min(\mathcal{N} - T_{\gamma})$.

Proposition 2. Let N be a finite c-net. Then N is n-safe for some n iff the automaton $\mathcal{A}_{hp}(N)$ is finite.

The main result now states that there is a precise correspondence between HP-bisimulation on contextual nets and CA-bisimulation on causal automata. Hence HP-bisimilarity of contextual nets can be checked on the corresponding automata.

Theorem 1. Let N_1 and N_2 be two c-nets. Then $N_1 \sim_{hp} N_2$ if and only if $\mathcal{A}_{hp}(N_1) \sim_{ca} \mathcal{A}_{hp}(N_2)$.

Proof (sketch). The proof is organized in two steps. First observe that the transition system of marked processes of a c-net N can be seen itself as a causal automaton $\mathcal{A}_{pr}(N) = \langle Q, n, \longmapsto, q_0 \rangle$, where

- Q is the set of marked processes π of N and $n(\pi) = T_{\pi}$ for any process π ;
- $-\pi \vdash_{M} \xrightarrow{a} \pi'$ if, according to Definition 7, $\pi \xrightarrow{a} \pi'$, $M = \mathsf{MC}(\pi \xrightarrow{a} \pi')$, and the naming $\sigma: T_{\pi'} \to T_{\pi} \cup \{\star\}$ is defined as the identity for $x \in T_{\pi'} - \{e\}$, while $\sigma(e) = \star$;
- the initial state q_0 is any initial process of N.

Then, it is possible to prove that HP-bisimulation on c-nets coincides with CA-bisimulation on the causal automata of processes, namely $N_1 \sim_{hp} N_2$ iff $\mathcal{A}_{pr}(N_1) \sim_{ca} \mathcal{A}_{pr}(N_2)$.

The second step of the proof shows that, for any c-net N there exists an abstraction homomorphism $h : \mathcal{A}_{pr}(N) \to \mathcal{A}_{hp}(N)$, and thus, by Lemma 1, $\mathcal{A}_{pr}(N) \sim_{ca} \mathcal{A}_{hp}(N)$. The abstraction homomorphism $h = \langle h, \{h_{\pi}\}_{\pi} \rangle$ can be

defined as follows: for any marked process π (state of $\mathcal{A}_{pr}(N)$), if $\operatorname{norm}(\gamma(\pi)) = \langle \gamma', i \rangle$ then $h(\pi) = \gamma'$ and $h_{\pi}: T_{\gamma'} \to T_{\pi}$ is simply i_T .

Summing up, by the above considerations we have that $\mathcal{A}_{pr}(N_i) \sim_{ca} \mathcal{A}_{hp}(N_i)$ for $i \in \{1, 2\}$, and moreover $\mathcal{A}_{pr}(N_1) \sim_{ca} \mathcal{A}_{pr}(N_2)$ iff $N_1 \sim_{hp} N_2$. Hence the thesis easily follows.

By Proposition 2 and Theorem 1 we immediately conclude the desired decidability result.

Corollary 1. HP-bisimulation is decidable on finite n-safe contextual nets.

To conclude the discussion on HP-bisimulation it is worth observing that the configurations in [11] and the ordered markings in [19] are slightly more abstract than our partial processes essentially for the fact that they do not record the correspondence between events in the process and transitions in the original net. We have used partial processes only because they allow for a simpler presentation, since partial processes are still (unmarked) processes of the given net and no new notion has to be introduced. Concretely, the drawback of our choice is that dealing with configurations the number of states of the causal automaton associated to a net may result significantly smaller. Therefore, although conceptually the two approaches are very similar, when one is interested in complexity and efficiency issues working with configurations is the better choice.

5.2 RHP-bisimulation

Let us turn our attention to RHP-bisimulation, the finer equivalence introduced in Definition 10. Since asymmetric event structures and thus RHP-bisimulation distinguish (strong) causality from weak causality (asymmetric conflict), for each step of computation we must observe *separately* the maximal causes and the weak causes (namely the events in the current state which are in asymmetric conflict with the considered transition). It is easy to realize that keeping information only about the producers and maximal readers of processes is no more adequate and thus the partial processes defined in the previous section are not sufficiently informative to discriminate c-nets which differ according to RHP-bisimulation.

Consider for instance the net N and the process π_4 in Fig. 2. Since $e_2 \leq e_3$ and both transitions read the same token, the corresponding partial process records only the transition e_3 . However, in this way, when the transition labelled by d fires, the partial process allows us to recover only the weak dependency from transition e_3 , while the dependency from e_2 is lost.

To extend the technique described in the previous section to deal with RHPbisimulation we must change the notion of partial process in order to keep trace of *all* events, maximal or not, which read a token in the current marking.

Definition 20 (read partial processes). A read partial process (R-partial process) χ of a net N is an unmarked process $\chi : O_{\chi} \to N$, where $T_{\chi} = p(\chi) \cup r(\chi)$.



Fig. 4. A finite safe c-net where the number of readers in a marked process is unbounded.

The notion of R-partial process move can be defined in the obvious way, and relying on the transition system of R-partial processes of a c-net N we can construct a causal automaton $\mathcal{A}_{rhp}(N)$. The only difference with respect to the case of HP-bisimulation is that the transitions of the causal automaton must be labeled with two sets of names, i.e., they must have the form $q \mapsto_{M,W}^{a} q'$. In fact for each transition we must observe separately the set of maximal (strong) causes $\mathsf{MC}(\chi \xrightarrow[a]{e} \chi') = \{t \in T_{\chi} : t^{\bullet} \cap (\bullet e \cup \underline{e}) \neq \emptyset \land t \text{ is } \leq_{\chi}\text{-maximal}\}$ and the set of all weak causes $\mathsf{WC}(\chi \xrightarrow[a]{e} \chi') = \{t \in T_{\chi} : \underline{t} \cap \bullet e \neq \emptyset\}$. For example, in Fig. 2, for the move $\pi_4 \xrightarrow[a]{e_4} \pi_5$ we have $\mathsf{MC}(\pi_4 \xrightarrow[a]{e_4} \pi_5) = \{e_0\}$, while $\mathsf{WC}(\pi_4 \xrightarrow[a]{e_4} \pi_5) = \{e_1, e_2, e_3\}$.

Generalizing the notion of CA-bisimulation for causal automata where transitions are labelled by two sets of names, it is still possible to prove that given two c-nets N_1 and N_2 , $N_1 \sim_{rhp} N_2$ iff $\mathcal{A}_{rhp}(N_1) \sim_{ca} \mathcal{A}_{rhp}(N_2)$.

Unfortunately, as anticipated, such translation into causal automata does not help in checking RHP-bisimulation. The states of the automaton $\mathcal{A}_{rhp}(N)$, which are normalized R-partial processes reachable from the initial process, can be infinitely many also for a finite safe net. In fact, the number of readers $r(\pi)$ in a marked process of very simple finite safe c-nets is easily seen to be unbounded, as shown, for instance by the net in Fig. 4.

6 Conclusions

We have provided contextual nets with abstract semantics inspired by the classical notion of history preserving bisimulation. The presence of asymmetric conflicts (weak causalities) between events in contextual net computations naturally suggests two different formulations of history preserving bisimulation. HP-bisimulation, relying on deterministic PES's, observes only the precedences between events, while the finer RHP-bisimulation observes both causality and weak causality by exploiting deterministic AES's. We have shown that, as for ordinary nets, HP-bisimulation is decidable for finite n-safe contextual nets, and we have provided a "translation" of such kind of nets into finite state causal automata, which allows us to reuse the algorithms existing for this general formalism in order to decide the bisimulation equivalence and to obtain a minimal realization. Such a translation can be adapted to deal with RHP-bisimulation. However in this case the construction may produce an infinite state automaton also for finite n-safe nets and thus it is not helpful to conclude the decidability of RHP-bisimulation, which remains a matter of future investigation.

An alternative solution to the problem of deciding HP-bisimulation on finite n-safe c-nets could be to find a procedure which translates each finite n-safe c-net into an HP-bisimilar ordinary net, and then to exploit the corresponding algorithm for ordinary nets. It should not be difficult to see that the translation proposed in [14], which maps each 1-safe c-net into an ordinary 1-safe net, respects HP-bisimulation and thus is appropriate for this aim. However, such translation does not extends naturally to general n-safe nets with weighted contexts. A preliminary investigation reveals that, if a translation exists, probably it would transform a c-net into a significantly more complex ordinary net. Hence, the direct algorithm proposed in this paper could still be preferable.

In [10] the hereditary (or strong) version of history preserving bisimulation on PES's is given an abstract characterization in terms of open maps, by taking as experiments finite deterministic PES's. It would be interesting to investigate the notions of bisimulation on AES's arising when one takes as experiments the deterministic PES's and the deterministic AES's. We conjecture that such equivalences gives, respectively, the hereditary variations of HP- and RHP-bisimulations.

Finally, a more general direction of future research is the development of Hennessy-Milner style logics to be interpreted over formalisms endowed with a causal semantics, with particular interest in the general model of causal automata. Such logics would be particularly appropriate for concurrent systems, allowing to express properties on their causal behaviour, like the existence of a computation where an action can be executed with a given set of causes. Some preliminary studies [2] have led to a logic for causal automata, which has been shown to be adequate with respect to CA-bisimulation in the sense that two automata are bisimilar iff they satisfies the same set of formulae.

Acknowledgments We are grateful to the anonymous referees for their comments on the submitted version of this paper.

References

- P. Baldan, A. Corradini, and U. Montanari. An event structure semantics for P/T contextual nets: Asymmetric event structures. In M. Nivat, editor, *Proceedings of FoSSaCS* '98, volume 1378 of *LNCS*, pages 63–80. Springer Verlag, 1998.
- 2. R. Bartolini. Model checking di proprietà causali di reti di Petri. MSc Thesis, University of Pisa, 1999.
- E. Best, R. Devillers, A. Kiehn, and L. Pomello. Concurrent bisimulations in Petri nets. Acta Informatica, 28(3):231–264, 1991.
- I. Castellani. Bisimulations and abstraction homomorphisms. Journal of Computer and System Sciences, 34(2/3):210–235, 1987.
- S. Christensen and N. D. Hansen. Coloured Petri nets extended with place capacities, test arcs and inhibitor arcs. In M. Ajmone-Marsan, editor, *Applications and Theory of Petri Nets*, volume 691 of *LNCS*, pages 186–205. Springer Verlag, 1993.

- G. Ferrari and U. Montanari. Towards the unification of models of concurrency. In A. Arnold, editor, *Proceedings of CAAP '90*, volume 431 of *LNCS*, pages 162–176. Springer-Verlag, 1990.
- F. Gadducci and U. Montanari. Axioms for contextual net processes. In Proceedings of ICALP'98, LNCS, pages 296–308. Springer Verlag, 1998.
- R. Janicki and M. Koutny. Invariant semantics of nets with inhibitor arcs. In Proceedings of CONCUR '91, volume 527 of LNCS. Springer Verlag, 1991.
- R. Janicki and M. Koutny. Semantics of inhibitor nets. Information and Computation, 123:1–16, 1995.
- A. Joyal, M. Nielsen, and G. Winskel. Bisimulation from open maps. Information and Computation, 127(2):164–185, 1996.
- U. Montanari and M. Pistore. Minimal transition systems for history-preserving bisimulation. In 14th Annual Symposium on Theoretical Aspects of Computer Science, volume 1200 of LNCS, pages 413–425. Springer Verlag, 1997.
- U. Montanari and M. Pistore. History-dependent automata. Technical Report TR-98-11, Dipartimento di Informatica, 1998. Available as ftp://ftp.di.unipi.it/pub/techreports/TR-98-11.ps.Z.
- U. Montanari and F. Rossi. Contextual occurrence nets and concurrent constraint programming. In H.-J. Schneider and H. Ehrig, editors, *Proceedings of the Dagstuhl* Seminar 9301 on Graph Transformations in Computer Science, volume 776 of LNCS. Springer Verlag, 1994.
- 14. U. Montanari and F. Rossi. Contextual nets. Acta Informatica, 32, 1995.
- A. Rabinovich and B. A. Trakhtenbrot. Behavior Structures and Nets. Fundamenta Informaticæ, 11(4):357–404, 1988.
- G. Ristori. Modelling Systems with Shared Resources via Petri Nets. PhD thesis, Department of Computer Science - University of Pisa, 1994.
- 17. J. van Bentham. Correspondence theory. In *Handbook of Philosophical Logic*, volume II. Reidel, 1984.
- R. van Glabbeek and U. Goltz. Equivalence notions for concurrent systems and refinement of actions. In A. Kreczmar and G. Mirkowska, editors, *Proceedings of MFCS*'89, volume 39 of *LNCS*, pages 237–248. Springer Verlag, 1989.
- W. Vogler. Deciding history preserving bisimilarity. In J. Leach Albert, B. Monien, and M. Rodríguez-Artalejo, editors, *Proceedings of ICALP'91*, volume 510 of *LNCS*, pages 495–505. Springer-Verlag, 1991.
- W. Vogler. Efficiency of asynchronous systems and read arcs in Petri nets. In *Proceedings of ICALP'97*, volume 1256 of *LNCS*, pages 538–548. Springer Verlag, 1997.
- W. Vogler. Partial Order Semantics and Read Arcs. In Proceedings of MFCS'97, volume 1295 of LNCS, pages 508–518. Springer Verlag, 1997.
- G. Winskel. Event Structures. In Petri Nets: Applications and Relationships to Other Models of Concurrency, volume 255 of LNCS, pages 325–392. Springer Verlag, 1987.