BRICS

Basic Research in Computer Science

BRICS RS-99-4 Fröschle & Hildebrandt: On Plain and Hereditary History-Preserving Bisimulation

On Plain and Hereditary History-Preserving Bisimulation

Sibylle B. Fröschle
Thomas Troels Hildebrandt

BRICS Report Series **RS-99-4**

ISSN 0909-0878 **February 1999**

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK–8000 Aarhus C
Denmark
Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@brics.dk

BRICS publications are in general accessible through the World Wide Web and anonymous FTP through these URLs:

`http://www.brics.dk`
`ftp://ftp.brics.dk`
This document in subdirectory `RS/99/4/`

On Plain and Hereditary History-Preserving Bisimulation*

Sibylle B. Fröschle

LFCS,† University of Edinburgh, Scotland, sib@dcs.ed.ac.uk,

Thomas T. Hildebrandt

BRICS,‡ University of Aarhus, Denmark, hilde@brics.dk.

February, 1999

Abstract

We investigate the difference between two well-known notions of independence bisimilarity, *history-preserving bisimulation* and *hereditary history-preserving bisimulation*. We characterise the difference between the two bisimulations in *trace-theoretical* terms, advocating the view that the first is (just) a bisimulation for *causality*, while the second is a bisimulation for *concurrency*. We explore the frontier zone between the two notions by defining a *hierarchy* of bounded backtracking bisimulations. Our goal is to provide a stepping stone for the solution to the intriguing open problem of whether hereditary history-preserving bisimulation is decidable or not. We prove that each of the bounded bisimulations is decidable. However, we also prove that the hierarchy is strict. This rules out the possibility that decidability of the general problem follows directly from the special case. Finally, we give a non trivial reduction solving the general problem for a restricted class of systems and give pointers towards a full answer.

---

*Full version of an extended abstract appearing in Proc. of MFCS '99, © Springer-Verlag.

†**L**aboratory for **F**oundations of **C**omputer **S**cience.

‡**B**asic **R**esearch **i**n **C**omputer **S**cience, Centre of the Danish National Research Foundation. Part of this work was done while the second author was at LFCS.

1 Introduction

Bisimulation equivalence for concurrent systems was introduced by Park and Milner [14, 9] as a way of describing when two systems can be considered to denote the same process. The idea was to identify systems that could not be distinguished by interaction with an environment, and notably, this took into account the branching structure of systems. It was defined for models for process algebras like e.g. CCS and CSP in which concurrency is treated as *non-deterministic interleaving* of actions. However, for some situations, a more detailed description of the causal ordering between actions is needed. One example is when *action refinement* is considered, as studied by e.g. Vogler [19], Glabbeek and Goltz [16]. Models of this kind, that do not abstract from concurrency, are commonly referred to as *independence*, *partial order* or *true concurrency* models. Examples of these are labelled event structures, Petri nets and asynchronous transition systems, e.g. see [21].

Many attempts have been made to answer the question what the appropriate generalisation of the interleaving bisimulation to independence models is. Two interesting bisimulations for independence models are history-preserving bisimulation (HPB) and hereditary history-preserving bisimulation (HHPB). HPB was introduced in [15] and [5] under the name of *behaviour structure* bisimulation, and *mixed ordering* (mo) bisimulation respectively. The term *history-preserving* originates from [16], where Goltz and vanGlabbeek define the notion for event structures and prove the key property of HPB, namely that it is preserved under action refinement. This result has given history-preserving bisimulation its prominent place among independence bisimulations. In [2] the notion is introduced as fully concurrent bisimulation. There it is independently shown that HPB preserves action refinement for the more general model of Petri nets.

The notion of HHPB first appears in [1], where Bednarczyk studies several history-preserving bisimulations with a downwards closure condition. He calls sets that satisfy this condition *hereditary*. HHPB has also been introduced in [8] under the name of *strong* history-preserving bisimulation. This paper describes a uniform way of defining a bisimulation equivalence across a wide range of different models by applying category-theoretical ideas. For many concrete models, the abstract bisimulation specializes to already known equivalences [4]. In particular, one gets classical bisimulation for standard transition systems. For independence models, the abstract bisimulation specializes to HHPB suggesting that this notion is a very natural independence bisimulation. This is further confirmed by the results of [12]. Relational, logical and game-theoretical characterizations are found which come as con-

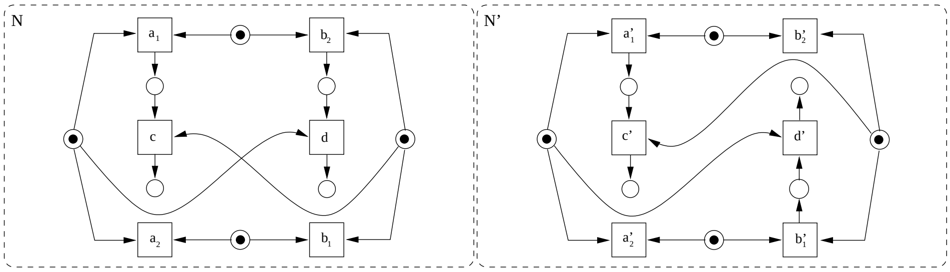

Figure 1: Two labelled nets N and N' that are HP bisimilar but not HHP bisimilar. The transitions are labelled by the actions $\{a, b, c, d\}$ as the names suggest, e.g. a_1 is labelled by a

servative extensions of the corresponding characterizations of classical bisimulation.

Altogether a fair amount of work has been done already in studying both, HPB and HHPB. However, very few attempts [17] have been made to demarcate the two notions from each other. Moreover, an intriguing question remains unsolved: Is HHPB decidable for a reasonable class of systems? In contrast, HPB has been shown to be decidable for finite 1-safe Petri nets by Vogler [18], DEXPTIME-complete by Jategaonkar and Meyer [7] and decidable for n-safe nets by Montanari and Pistore [10]. But there is no straightforward adaption of these proofs to HHPB, and it seems that the hereditary condition brings about new dimensions. This justifies a deeper investigation of the difference between plain and hereditary HPB, which is the goal of this paper.

One statement we want to put forward is that hereditary HPB is a bisimulation for *concurrency* as opposed to plain HPB (just) being a bisimulation for *causality*. Intuitively, HPB is an equivalence notion that relates systems with the same *causal* branching structure. It extends the classical notion of bisimulation with the requirement, that any two related runs must have the same causal dependency between actions, that is the same *history*. Hereditary HPB additionally imposes a backtracking condition: for any two related runs, the runs obtained by *backtracking* a pair of related transitions, must be related, too. We allow backtracking not only in the order which is laid down by the related runs; as long as no other transitions depend on a particular transition, it can be backtracked. Thereby it is ensured that the matching is not dependent on the order in which independent actions are linearized. Intuitively this is what we expect from a bisimulation for concurrency.

Figure 1 shows the standard example from [12] of two systems that are plain but not hereditary HP bisimilar. Both systems have an a-action (b-action) that can be followed by a dependent c-action (d-action) or an independent (not competing on any places) b-action (a-action). And both have an a-action (a b-action) which can be followed by an independent b-action (a-action). Consequently, the two systems are HP bisimilar. However, observe that in any HPB we can find, the matching of the parallel a- and b-transitions depends on the order in which they appear in the runs to match. So, the systems are not hereditary HP bisimilar. Note that the c transition dictates that we have to match a_1 to a'_1, and so $a_1.b_1$ to $a'_1.b'_1$. Then the backtracking condition requires that b_1 and b'_1 are related. But from this point, the system N' can make a d transition which N cannot match, so b_1 and b'_1 can clearly not be related runs.

After stating the necessary definitions in Sec. 2, we present a trace-theoretical characterisation of the difference between HHPB and HPB in Sec. 3. This will confirm our view of HHPB as a bisimulation for concurrency as opposed to HPB as a bisimulation for causality. In Sec. 4, we consider the effect of restricting HHPB, by bounding how far back in two related runs one can pick transitions to backtrack. Remarkably, we prove in Sec. 4.1 that for a fixed bound, each such bisimulation is decidable. However, in Sec. 4.2 we find that the bounded bisimulations form a *strict* hierarchy, all trivially stronger than HPB but also strictly weaker than hereditary HPB. In Sec. 5 we apply our results to approach the decidability of HHPB (for finite-state systems). After noting that decidability follows almost immediately for the class of bounded asynchronous nets, we present a non-trivial reduction in Sec. 5.2 showing that HHPB is decidable for systems with transitive independence relation. In the end, we remark on other partial results and give directions for further progress.

Let us note that one can also consider hidden actions in the context of HPB and HHPB. To avoid confusion with this standard use of strong and weak in the context of bisimulation, we prefer the name *hereditary* HPB over *strong* HPB. The weak version of HPB has been proved to be decidable in [7] and [20]. Here we will restrict our attention to (hereditary) HPB without hidden actions.

As our model of computation we choose 1-safe Petri nets. However, e.g. by using the results of [21], our results can equally be formulated for other suitable independence models, as for example transition systems with independence or labelled asynchronous transition systems.

2 Preliminaries

The following definitions are standard and/or can be found in [7], [11], or [18], perhaps in a slightly varied form.

Petri nets. A labelled Petri net N is a tuple $(S_N, T_N, F_N, init_N, l_N)$, where

- S_N is the set of places,
- T_N is the set of transitions,
- $F_N : (S_N \times T_N) \cup (T_N \times S_N) \rightarrow \{0, 1\}$ is the flow relation,
- $init_N : S_N \rightarrow \mathbb{N}_0$ is the initial marking, and
- $l_N : T_N \rightarrow Act$ is the labelling function, where Act is a set of actions.

A net N is *finite* iff S_N and T_N are finite sets.

The pre-set of an element $x \in S_N \cup T_N$, ${}^\bullet x$, is defined by $\{y \mid F_N(y, x) > 0\}$, the post-set of x, $x^\bullet$, similarly is $\{y \mid F_N(x, y) > 0\}$.

A marking M of N is a map $S_N \rightarrow \mathbb{N}_0$. We say M enables a transition $t \in T_N$ if $M(s) \geq F(s, t)$ for every $s \in S_N$. If t is enabled at M it can occur. The resulting marking M' is defined by $M'(s) = M(s) - F(s, t) + F(t, s)$ for all $s \in S_N$. We denote this by $M \xrightarrow{t} M'$.

We say that $w = t_1 \dots t_n$, is a *transition-sequence* of N. We write $|w|$ for the length of w, that is $|w| = n$. If $M \xrightarrow{t_1} \cdots \xrightarrow{t_n} M'$ we use $M \xrightarrow{w} M'$ as short notation. For any transition t we write $w.t$ for the sequence $t_1 \dots t_n t$.

A net N is *1-safe* if for every marking M that is reachable from $init_N$, we have: $M(s) \leq 1$ for every $s \in S_N$. Thus, in 1-safe nets a marking can be viewed as a set of places. We say $s \in S_N$ holds at marking M iff $s \in M$. We will always refer to this net class whenever we speak of ‘nets’ or ‘Petri nets’ in the following.

Runs. A *run* of a net N is a possibly empty transition-sequence r such that $init_N \xrightarrow{r} M'$ for some M'. Let $Runs(N)$ denote the set of all runs of a net N. When we have $r \in Runs(N)$, $t \in T_N$, and two markings M, M', such that $init_N \xrightarrow{r} M$ and $M \xrightarrow{t} M'$, then we write $r \xrightarrow{t} r.t$.

Independence of Transitions. We say two transitions t and t' of a net N are *independent* in N, denoted by $t\ I_N\ t'$, iff their neighbourhoods of places do not intersect, i. e. iff $({}^\bullet t \cup t^\bullet) \cap ({}^\bullet t' \cup t'^\bullet) = \emptyset$.

Pomsets. A *pomset* is a labelled partial order.[1] It is a tuple $p = (E_p, <_p, L_p, l_p)$, where E_p is a set of events, $<_p$ a partial order relation on E_p, L_p is a set of labels, and l_p a labelling function $l_p : E_p \rightarrow L_p$. A function f is an $isomorphism$ between pomset p and pomset q iff $f : E_p \rightarrow E_q$ is a bijection, such that we have $l_p = l_q \circ f$, and $e <_p e'$ iff $f(e) <_q f(e')$ for all $e, e' \in E_p$.

Transition-pomsets. The *transition-pomset* of a run $r = t_1 \dots t_n$, denoted by $trPom(r)$, has as events the integers from 1 to n, where the label of event i is t_i and the partial ordering is the transitive closure of the following "proximate cause" relation: event i *proximately causes* event j iff $i < j$ and t_i and t_j are *not* independent in N. The *pomset* of r, denoted by $pom(r)$, is the transition-pomset of r, where the label of each event i is $l_N(t_i)$, the *label* of t_i, rather than t_i itself.

Trace Theory. A *trace alphabet* is a pair (Σ, I), where the alphabet Σ is a finite set, and $I \subseteq \Sigma \times \Sigma$ is an irreflexive and symmetric independence relation. Let Σ^* be the set of finite words over Σ, and let r, r' range over Σ^*. For $T \subseteq \Sigma$, let $r \uparrow T$ denote the projection of r onto T, i. e. the sequence obtained by erasing all occurrences of letters which are not in T. The independence relation I induces a relation $\sim_I \subseteq \Sigma^* \times \Sigma^*$ defined by $r \sim_I r'$ iff $r \uparrow \{a, b\} = r' \uparrow \{a, b\}$ for all $a, b \in \Sigma$ such that $\neg(a\ I\ b)$. Clearly, $\sim_I$ is an equivalence relation. The $\sim_I$ equivalence classes are usually referred to as *(Mazurkiewicz's) traces*. For $r \in \Sigma^*$, $[\, r\,]$ stands for the trace containing r. $\Sigma^*/\sim_I$ represents the set of all traces over (Σ, I).

Petri nets and Trace Theory. We can associate the trace alphabet (Σ_N, I_N) to a Petri net N, where $\Sigma_N = T_N$, and I_N is as defined above. Transition-pomsets of a net N correspond one-to-one to traces in $Runs(N)/\sim_{I_N} \subseteq \Sigma_N^*/\sim_{I_N}$. A trace $[r] \in Runs(N)/\sim_{I_N}$ corresponds to $trPom(r)$ and a transition-pomset p of N corresponds to the trace $\{r \mid r \text{ is a linearization of } p\}$.

3 (Hereditary) History-Preserving Bisimulation and Trace Theory

We are now ready for the two notions which are central to this paper, *HPB* and *HHPB*. Originally, these bisimulations have been defined on structures

[1]This is not the original definition, but the convention used in [7].

that represent the partial order explicitly. By employing the notion of *synchronous runs* from [7], and the notion of *backwards enabled transitions* introduced in [12] we can define (hereditary) HPB on runs, instead. This gives a characterization closely related to work in [5] and [12].

Definition 1 *Let r_1 and r_2 be runs of nets N_1 and N_2, respectively. We say that r_1 and r_2 are* synchronous *iff the identity function on $\{1, 2, \ldots, |r_1|\}$ is an isomorphism between the pomset of r_1 and the pomset of r_2.*

Intuitively, two runs are synchronous if their induced pomsets are isomorphic, and both runs correspond to the same linearization of the associated pomset isomorphism class.

Definition 2 *Let N be a net, and (Σ_N, I_N) the associated trace alphabet. Let $r = t_1 \ldots t_n \in Runs(N)$. For $t \in \Sigma_N$, we say t is* backwards enabled *in r, written $t \in BEn(r)$, iff there is $i \in \{1, \ldots, n\}$ s. t. $t_i = t$, and $\forall j \in \{i+1, \ldots, n\}.\ t_j\ I_N\ t_i$. This means that i is a maximal element in $pom(r)$. If $t \in BEn(r)$ we define $\delta(r, t)$ to be the result of deleting the last occurrence of t in r, i. e. $\delta(r, t) = t_1 \ldots t_{i-1} t_{i+1} \ldots t_n$ iff $last(r, t) = i$, where $last(r, t)$ denotes the position of the last occurrence of t in r. That is $last(r, t) = i$ iff $t_i = t$ and $t_j \neq t$ for all $j \in \{i+1, \ldots, n\}$.*

Definition 3 *A* HPB *between two nets N_1 and N_2 consists of a set $\mathcal{H} \subseteq Runs(N_1) \times Runs(N_2)$ of pairs (r_1, r_2) such that*

- *(i) Whenever $(r_1, r_2) \in \mathcal{H}$, then r_1 and r_2 are synchronous.*
- *(ii) $(\varepsilon, \varepsilon) \in \mathcal{H}$.*
- *(iii) Whenever $(r_1, r_2) \in \mathcal{H}$ and $r_1 \xrightarrow{t_1} r_1.t_1$ for some t_1, then there exists t_2, such that $r_2 \xrightarrow{t_2} r_2.t_2$ and $(r_1.t_1, r_2.t_2) \in \mathcal{H}$.*
- *(iv) Vice versa.*

A HPB is hereditary *when it further satisfies*

- *(v) Whenever $(r_1, r_2) \in \mathcal{H}$ and $t_1 \in BEn(r_1)$ and $t_2 \in BEn(r_2)$ for some t_1, t_2 such that $last(r_1, t_1) = last(r_2, t_2)$, then $(\delta(r_1, t_1), \delta(r_2, t_2)) \in \mathcal{H}$.*

We say two nets are (hereditary) HP bisimilar iff there is a (hereditary) HPB relating them.

It is trivial that one can regard a relation $R \subseteq \{ (r_1, r_2) \in T^*_{N_1} \times T^*_{N_2} \mid |r_1| = |r_2| \}$ as a language over the alphabet $T_{N_1} \times T_{N_2}$, and vice versa. With this in mind, we can regard a (hereditary) HPB $\mathcal{H}$ as a language over the *trace alphabet* $\mathcal{T}_{N_1,N_2}$. We define $\mathcal{T}_{N_1,N_2}$ as $\mathcal{T}_{N_1,N_2} = (\Sigma, I)$, where $\Sigma = T_{N_1} \times T_{N_2}$, and I is defined as $(t_1, t_2)\ I\ (t'_1, t'_2)$ iff $t_1\ I_{N_1}\ t'_1\ \wedge\ t_2\ I_{N_2}\ t'_2$.

We will now characterize the difference between HPB and HHPB in trace-theoretical terms. For this we consider two properties of languages.

Definition 4 *We say a language $L \subseteq \Sigma^*$ is* prefix-closed *iff $r.t \in L$ implies $r \in L$.*

We say L is trace-consistent w. r. t. an independence relation I on Σ *iff $r \sim_I r' \in L$ implies $r \in L$. For $L \subseteq \Sigma^*$, let $L_{\sim_I}$ denote the smallest trace language including L, i. e. $L_{\sim_I} = \{r \in \Sigma^* \mid \exists r' \in L.\ r' \sim_I r\}$.*

By definition every HHPB is prefix-closed. This does not generally apply for HPBs. But as prefix-closed HPBs correspond to bisimulations that have been built up inductively from $(\varepsilon, \varepsilon)$ without adding "any redundant tuples", we can extract from any given HPB one that is prefix-closed.

Proposition 1 *Two nets are (hereditary) HP bisimilar iff there exists a* prefix-closed *(hereditary) HPB language relating them.*

A HPB language $\mathcal{H}$ is not necessarily trace-consistent, neither is a HHPB. But this can always be obtained.

Observation 1 *Let $\mathcal{H}$ be a (hereditary) HPB language between two nets N_1 and N_2. Let $\mathcal{T}_{N_1,N_2} = (\Sigma, I)$, then $\mathcal{H}_{\sim_I}$ is a (hereditary) HPB too.*

Prop. 1 ensures, that it is safe to consider only prefix-closed HPBs. Note that if this property is fixed, an analogue to Obs. 1 is no longer possible. In general, if $\mathcal{H}$ is a prefix-closed HPB, $\mathcal{H}_{\sim_I}$ is not necessarily prefix-closed. However, if $\mathcal{H}$ is hereditary, this will still be true.

Interestingly, if a prefix-closed HPB is also trace-consistent, it is in fact hereditary. So, if one takes as part of the definition that a HPB is prefix-closed, one can regard hereditary HPBs as the class of trace-consistent HPBs.

Proposition 2 *Two nets are hereditary HP bisimilar iff there exists a* trace-consistent *prefix-closed HPB relating them.*

PROOF: "$\Rightarrow$" By Obs. 1, we can extend every prefix-closed HPB $\mathcal{H}$ to the trace-consistent HPB $\mathcal{H}_{\sim_I}$. If $\mathcal{H}$ is hereditary we have that $\mathcal{H}_{\sim_I}$ is still prefix-closed.

"$\Leftarrow$" Let $\mathcal{H}$ be a trace-consistent and prefix-closed HPB relating the two nets N_1, N_2. We only need to check property (v) of definition 3. Note that we can use BEn and δ for joint runs and transitions of N_1 and N_2 in the obvious way. Then to prove property (v) we assume $r \in \mathcal{H}$ and $t \in BEn(r)$, and have to show that $\delta(r,t) \in \mathcal{H}$.

So assume $r \in \mathcal{H}$, and $t \in BEn(r)$. As $\mathcal{H}$ is trace-consistent, we have $r' \in \mathcal{H}$ such that r' corresponds to r with the last occurrence of t reshuffled to last position. As $\mathcal{H}$ is prefix-closed, we get $\delta(r',t) = \delta(r,t) \in \mathcal{H}$. □

Remark: Conversely, from Obs. 1 it follows that one could take as part of the definition that a HPB is trace-consistent. Then HHPBs become the class of prefix-closed HPBs. This is exactly the approach taken in the original definition of HHPB, since HPBs defined on partial orders correspond precisely to the class of trace-consistent HPBs defined on runs. We find the view we have put forward more natural. Taking trace-consistency as part of the definition disguises how the linearized runs of the two systems are matched to each other. Since in HPBs the matching can be dependent on the order in which independent actions are linearized, this is information we do not want to hide away in a HPB.

With the property of prefix-closure we merely restrict our attention to HPBs that have been inductively built up. Hence, defining HPB on synchronous runs and fixing prefix-closure as part of the definition seems very natural. The interpretation of HHPBs as the class of (prefix-closed) HPBs that are *trace languages* expresses then nicely that in HHPB the matching does not depend on the order of how independent transitions are linearized.

It is not difficult to capture the conditions (i)-(iv) of the definition of HPB in terms of languages as well. Together with the results above, this gives a purely language-theoretical characterisation of HPB and HHPB, which can be found in [6].

4 History-Preserving Bisimulation and Bounded Backtracking

We define a hierarchy of backtracking bisimulations by bounding the number of transitions which one can backtrack over to an arbitrary number n.

Definition 5 *A HPB $\mathcal{H}$ is (n)-hereditary when it further satisfies*

(v) Whenever $(r_1, r_2) \in \mathcal{H}$ and $t_1 \in BEn(r_1)$ and $t_2 \in BEn(r_2)$ for some t_1, t_2 such that $last(r_1, t_1) = last(r_2, t_2) \geq |r_1| - n$, then $(\delta(r_1, t_1), \delta(r_2, t_2)) \in \mathcal{H}$.

Note that (0)-hereditary HPBs are exactly the prefix-closed HPBs.

It is easy to give a dynamic condition on nets, which guarantees that (n)-hereditary HP bisimilarity coincides with hereditary HP bisimilarity.

Definition 6 *Let N be a net. We say that N is* (n)-bounded asynchronous *if for any $r = t_1 t_2 \ldots t_k \in Runs(N)$ such that $t_i \in BEn(r)$, it holds that $k - i \leq n$.*

Proposition 3 *Let N and N' be two (n)-bounded asynchronous nets. Then N and N' are hereditary HP bisimilar iff N and N' are (n)-hereditary HP bisimilar.*

4.1 Decidability of (n)-Hereditary History-Preserving Bisimilarity

For any fixed n, (n)-HHP bisimilarity is decidable for finite systems. The idea behind our proof is that we can define HHPB and (n)-HHPB in a 'forward fashion'. At each tuple we keep a matching directive that prescribes how transitions are going to be matched from this point onwards. The matching directive allows us to express the backtracking requirement as a property of the matching directives of two connected tuples.

To characterize HHPB in this manner we need to record the matching of the entire future. Because of this the forwards characterization merely shifts the difficulty of the decidability of HHPB from the past to the future: now we are confronted with an infinite amount of possible futures. This is not the case for (n)-HHPB. But we shall see that it is sufficient to record future matchings of length n. Our proof builds on this fact and insights gained in the proofs of the decidability of HPB [18, 7].

Below is the definition of (n)-D HPB, our forwards characterization of (n)-HHPB.

Convention. For a pair of synchronous runs (r_1, r_2) of two nets N_1 and N_2, we use r as a short notation. Similarly, we write t for a pair of transitions (t_1, t_2) when t_1 and t_2 correspond to each other in a pair of synchronous runs (r_1, r_2). We also write $r \stackrel{t}{\rightarrow} r'$ when we have two pairs of synchronous runs (r_1, r_2), (r_1', r_2'), and a pair of transitions (t_1, t_2), such that $r_1 \stackrel{t_1}{\rightarrow} r_1'$ and $r_2 \stackrel{t_2}{\rightarrow} r_2'$.

Definition 7 *A (n)-D HPB between two nets N_1 and N_2 consists of a set $\mathcal{H}_D$ of triples (r_1, r_2, D) such that*

(i) *r_1 is a run of N_1, r_2 is a run of N_2, and r_1 and r_2 are synchronous. The matching directive D is a* non-empty *and* prefix-closed *set of pairs of words (w_1, w_2), such that w_1 is a transition-sequence of N_1, w_2 of N_2 respectively, and $|w_1| = |w_2| \leq n$.*

(ii) *For some D, $(\varepsilon, \varepsilon, D) \in \mathcal{H}_D$.*

(iii) *Whenever $(r_1, r_2, D) \in \mathcal{H}_D$, and $w \in D$ for some w, such that $|w| < n$, and for some t_1, $r_1.w_1 \stackrel{t_1}{\rightarrow} r_1.w_1.t_1$, then there is some t_2 such that $(w_1.t_1, w_2.t_2) \in D$.*

Note that $(\varepsilon, \varepsilon) \in D$ because D is prefix-closed and non-empty.

(iv) *Vice versa.*

(v) *Whenever $(r_1, r_2, D) \in \mathcal{H}_D$, and $(t_1, t_2) \in D$, then there is some D', such that $(r_1.t_1, r_2.t_2, D') \in \mathcal{H}_D$ and*

(a) $\forall w$ *s. t.* $|w| < n.\ tw \in D \Leftrightarrow w \in D'$.

(b) $\forall w'.\ w' \in D' \wedge t\ I\ t'$ *for all* $t' \in w' \Rightarrow w' \in D$.

We now prove that (n)-D HPB is indeed equivalent to (n)-HHPB. As in Sec. 3 it is sufficient to consider only prefix-closed (n)-D HPBs since they correspond to bisimulations that are built up inductively from the empty runs without adding any "redundant tuples". Prefix-closure for (n)-D HPB is defined as follows.

Definition 8 *We say a (n)-D HPB $\mathcal{H}_D$ is* prefix-closed *iff whenever $(r_1.t_1, r_2.t_2, D') \in \mathcal{H}_D$, then there is $(r_1, r_2, D) \in \mathcal{H}_D$ for some D such that $t \in D$ and*

1. $\forall w$ *s. t.* $|w| < n.\ tw \in D \Leftrightarrow w \in D'$.

2. $\forall w'.\ w' \in D' \wedge t\ I\ t'$ *for all* $t' \in w' \Rightarrow w' \in D$.

Lemma 1 *Two nets are (n)-hereditary HP bisimilar iff they are (n)-D HP bisimilar.*

PROOF: For one direction let $\mathcal{H}$ be a (n)-HHPB relating N_1 and N_2. It is also safe to assume prefix-closure of $\mathcal{H}$. We define $\mathcal{H}_D$ by assigning a matching directive D to every pair (r_1, r_2). We take $D = \{w \mid |w| \leq n \wedge r.w \in \mathcal{H}\}$. Prefix-closure of D is given by prefix-closure of $\mathcal{H}$, hence property (i) of definition 7 clearly holds. Properties (ii), (iii), and (iv) are also trivial.

To see that property (v) holds, let $(r_1, r_2, D) \in \mathcal{H}_D$ and $(t_1, t_2) \in D$. Then, due to the way D is defined there is D' such that $(r.t, D') \in \mathcal{H}_D$. Condition (a) is also immediate by the way matching directives are added to the tuples. To check condition (b) assume we have $w' \in D' \wedge t\ I\ t'$ for all $t' \in w'$. But then we have $r.t.w' \in \mathcal{H}$ with t being backtrack enabled. The fact that $|w'| \leq n$ together with property (v) of definition 5 implies that $r.w' \in \mathcal{H}$. Hence, by definition of D we have $w' \in D$ as required.

For the other direction assume $\mathcal{H}_D$ to be a prefix-closed (n)-D HPB. Define $\mathcal{H}$ by simply ignoring the matching directive D of triples $(r_1, r_2, D) \in \mathcal{H}_D$. It is clear that properties (i), (ii), (iii) and (iv) of the definition of (n)-HHPB are satisfied. To prove property (v), let $r.t.w \in \mathcal{H}$ such that t is backtrack enabled, and $|w| \leq n$. By prefix-closure of $\mathcal{H}_D$ we have $(r, D), (r.t, D') \in \mathcal{H}_D$ for some D, D' such that $t \in D$, $w \in D'$, and the two conditions of property (v) of definition 7 are satisfied. But then we have $w \in D$ by condition (b), and thus $(r.w, D'') \in \mathcal{H}_D$ for some D'' as required.
□

Now that we have expressed the backtracking condition in a forwards fashion, we can proceed along the lines of the decidability proofs for HPB [18, 7]. We will sketch these proofs, and thereby explain the remaining steps of our decidability proof.

For this we need a further definition from [7].

Definition 9 *Let $p = (E_p, <_p, L_p, l_p)$ be a pomset and $e, e' \in E_p$. Event e' is a* maximal cause *of event e in p iff $e' <_p e$ and there is no event $e'' \in E_p$ such that $e' <_p e'' <_p e$.*

The key insight of the proofs of the decidability of HPB is the following fact: two isomorphic pomsets stay isomorphic after the addition of a pair of transitions iff the maximal causes of the new events are the same (up to isomorphism) in the resulting pomsets. This means that we do not need to keep the entire history, but it is sufficient to record only those events that can act as maximal causes.

The next step is to find a notion that contains this most-recent history, but is finite in the sense that there are only finitely many instances of it. In any partial order run the events that can act as maximal causes correspond to distinct transitions. This is so because a transition cannot be independent of itself. Thus, as one possibility we can take pomsets whose events have distinct transitions as labels. As we consider only finite nets there are clearly only finitely many such pomsets. What we have just described is the notion of growth-sites defined by Jategaonkar and Meyer. Vogler develops a different concept called ordered markings (OM), where the most-recent history is captured by imposing an order on the markings of a net.

Instead of defining HPB on runs we can now base HPB on growth-sites or OMs. The resulting bisimulations are called gsc-bisimulation, and OM-bisimulation, respectively. Jategaonkar and Meyer show that gsc-bisimulation is indeed equivalent to HPB. Vogler proves the analogue for OM-bisimulation. As there are only finitely many growth-sites or OMs for a system, these bisimulations can be decided by exhaustive search. The decidability of HPB is then immediate.

We can define a growth-sites or OM bisimulation that corresponds to (n)-D HPB just as well, and call the resulting notions (n)-D gsc-bisimulation and (n)-D OM-bisimulation. The proof that (n)-D gsc- and (n)-D OM-bisimulation indeed coincide with (n)-D HPB is a straightforward adaptation of the proofs in [7] and [18]. Since there are only finitely many matching directives of size n, (n)-D gsc- and (n)-D OM-bisimilarity can also be decided by exhaustive search. Consequently, (n)-D HP bisimilarity is decidable and with it (n)-HHP bisimilarity.

Theorem 1 *For any fixed n, it is decidable whether two finite nets are (n)-HHP bisimilar.*

4.2 Strictness of the Hierarchy

It is a simple consequence of the definition, that HHP bisimilarity implies (n)-HHP bisimilarity for any n, which again implies (n')-HHP bisimilarity for $n' < n$. Given the result of the previous section, an obvious question to ask is whether HHP bisimilarity coincides with (n)-HHP bisimilarity for some fixed bound n. The example of Fig. 1 shows that (0)-HHP bisimilarity is weaker

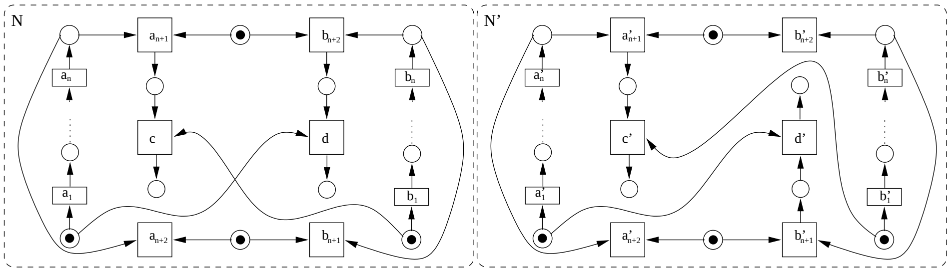

Figure 2: Two nets N and N' that are (n)-HHP bisimilar but not (n+1)-HHP bisimilar. Note that for $n = 0$ one gets the two systems given in Fig. 1

than (1)-HHP bisimilarity. Fig. 2 shows an elegant generalisation, which discriminates (n)-hereditary from (n+1)-hereditary HP bisimilarity. Despite its simple appearance, it was not at all trivial to find.

Let us first argue why no HHPB relates N and N'. In any HHPB we must match a_i with a'_i, and b_i with b'_i for $1 \leq i \leq n$. Then one option in N' is to perform a'_{n+1} and b'_{n+1}. These transitions have to be matched with either a_{n+1} and b_{n+1}, or a_{n+2} and b_{n+2} respectively. Suppose we choose the match a_{n+1}, b_{n+1}. We can now backtrack all the a-transitions such that d becomes enabled in N'. But no d action is possible in N. If we choose a_{n+2}, b_{n+2} as our match, we can backtrack all the b-transitions. Then c becomes possible in N', but not in N. The systems are clearly (n+1)-bounded asynchronous, so by Prop. 3 N and N' are not (n+1)-HHP bisimilar either.

The above counter-strategy does not apply for (n)-HHPB, but we can use the following strategy to match the critical $n + 1$ transitions. Say we have to match a'_{n+1}, and b'_{n+1} has not been fired yet, i. e. we can still choose between a_{n+1} and a_{n+2} as a match. We make our match dependent on the first transition in the history. Assume it is an a-transition. Then it is safe to match a'_{n+1} with a_{n+1}, which determines that b'_{n+1} is later matched with b_{n+1}. For d to become enabled in N', we need to backtrack all the a-transitions, however there will be $n + 1$ b-transitions following the first a, so this is not possible. Similar, it is safe to match a'_{n+2} with a_{n+2}. A symmetrical argument applies if the first action was a b-action, and similar for the remaining cases.

Lemma 2 *For all $n \in \mathbb{N}_0$, there exist two finite nets that are (n)- but not (n+1)-HHP bisimilar.*

Theorem 2 *For all $n \in \mathbb{N}_0$, (n)-HHP bisimilarity is strictly weaker than (n+1)-HHP bisimilarity, and hence (unbounded) HHP bisimilarity.*

5 Applications to the Decidability Problem of Hereditary History-Preserving Bisimulation

In the previous section we have shown that the hierarchy of (n)-HHPBs is strict. However, for any two fixed finite systems the hierarchy collapses, and so the decidability of the general problem would follow immediately, if the bound can be effectively computed for any two given systems. That this might not be possible in general is indicated by the fact, that the problem of hereditary history-preserving *simulation* has recently been shown to be undecidable [13]. Though, even if the general problem turns out to be undecidable, it is interesting to investigate for which classes of systems deciding HHPB does reduce to deciding (n)-HHPB. Below, we will give some restricted classes of systems, for which this is indeed the case.

5.1 Bounded Asynchronous Systems

We say that a net N is bounded asynchronous, if there exists some natural number n such that N is (n)-bounded asynchronous. It is easy to see, that a finite 1-safe net fails to be bounded asynchronous if and only if there is a reachable marking M and a *loop*, $M \stackrel{t_1}{\to} M_1 \cdots \stackrel{t_n}{\to} M_n = M$ such that every marking M_i in the loop enables a transition t which is independent of all transitions in the loop, i.e. $t \; I_N \; t_i$ for all i. Since finite 1-safe nets have only finitely many markings we get the following lemma.

Lemma 3 *It is decidable if a finite 1-safe net is (n)-bounded asynchronous for some n, and the bound n can be computed.*

With Prop. 3 the decidability of HHPB for bounded asynchronous systems follows immediately.

Proposition 1 *HHP bisimilarity is decidable for bounded asynchronous nets.*

5.2 Systems with Transitive Independence Relation

Definition 10 *An independence relation I over an alphabet Σ is transitive if, for every* distinct $t, t', t'' \in \Sigma$, $t \; I \; t' \wedge t' \; I \; t''$ *implies* $t \; I \; t''$.

Let N be a net. A transition $t \in T_N$ is a self-loop *iff ${}^\bullet t = t^\bullet$. Intuitively, a self-loop is a transition that can be repeated immediately, i. e. independently of the occurrence of other transitions. Note that the existence of a run $r = r'.t.t$ implies that t is a self-loop (in our context of 1-safe nets).*

Let us first draw our attention to systems with transitive independence relation that do not contain any self-loops. It is easy to see that for such systems the number of transitions over which can be backtracked is bound by the size of the maximal independence clique. In other words, a system with maximal independence clique of size k is (k)-bounded asynchronous, and hence decidability for finite systems of this subclass is immediate.

If a system contains a self-loop that can occur concurrently with another transition, then this system is clearly not bounded asynchronous. However, we can transfer the decidability result to the full class of finite systems with transitive independence relation with the help of another key observation. In every (H)HPB between two systems with transitive independence relation, concurrently occurring self-loop transitions have always to be matched to self-loops. Hence, we do not need to consider the unfoldings of such self-loops. It is sufficient to match the first occurrence of such a transition, when we make sure that the match is indeed a self-loop. But then the number of transitions over which one can backtrack is again bound by the size of the maximal independence clique, and so we have established decidability. The precise definition of what it means for a self-loop to occur concurrently in a given context, and the details of the proof can be found in the appendix.

Theorem 3 *For finite systems with transitive independence relation, HHP bisimilarity is decidable.*

6 Final Remarks

There is still undiscovered land in the zone between plain and hereditary HPB. One possibility to advance the frontier is to identify system classes for which the two notions coincide. Several classes of such systems have already been found. The most interesting one is the system class of BPP in full standard form [6]. Plain and hereditary HPB for the class of *free-choice* nets have recently been shown not to coincide by the first author, disproving a conjecture in [3].

The trace-theoretical characterization looks promising for approaching the decidability problem of HHPB, see [6] for more details.

Acknowledgements. The first author would like to thank Julian Bradfield and Anca Muscholl for helpful discussions and Walter Vogler for valuable comments on an earlier draft of this paper. The second author would like to acknowledge the lot of people who have contributed with their view on this problem, in particular Marcin Jurdziński.

References

[1] M. Bednarczyk. Hereditary history preserving bisimulation or what is the power of the future perfect in program logics. Technical report, Polish Acad. of Sciences, Gdansk, 1991.

[2] E. Best, R. Devillers, A. Kiehn, and L. Pomello. Concurrent bisimulations in Petri nets. *Acta Inform.*, 28(3):231–264, 1991.

[3] A. Cheng. *Reasoning About Concurrent Computational Systems.* PhD thesis, BRICS, Dep. of Computer Science, University of Aarhus, 1996.

[4] A. Cheng and M. Nielsen. Open maps (at) work. Research Series RS-95-23, BRICS, Dep. of Computer Science, University of Aarhus, Apr. 1995.

[5] P. Degano, R. De Nicola, and U. Montanari. Partial orderings descriptions and observations of nondeterministic concurrent processes. In *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, volume 354 of *LNCS*, pages 438–466, 1989.

[6] S. Fröschle. On the decidability problem of strong history-preserving bisimulation. Progress Report, 1998.

[7] L. Jategaonkar and A. R. Meyer. Deciding true concurrency equivalences on safe, finite nets. *Theor. Comp. Sci.*, 154(1):107–143, 1996.

[8] A. Joyal, M. Nielsen, and G. Winskel. Bisimulation from open maps. *Inform. and Comput.*, 127(2):164–185, 1996.

[9] R. Milner. A calculus of communicating systems. volume 92 of *LNCS*. Springer, 1980.

[10] U. Montanari and M. Pistore. Minimal transition systems for history-preserving bisimulation. In *STACS '97*, volume 1200 of *LNCS*, pages 413–425. Springer, 1997.

[11] M. Mukund and P. S. Thiagarajan. Linear time temporal logics over Mazurkiewicz traces. In *POMIV '96*, volume 29 of *DIMACS*, pages 171–201. AMS, 1997.

[12] M. Nielsen and C. Clausen. Bisimulations, games and logic. Research Series RS-94-6, BRICS, Dep. of Computer Science, University of Aarhus, April 1994.

[13] M. Nielsen and M. Jurdziński. Hereditary history preserving simulation is undecidable. Research Series RS-99-1, BRICS, 1999.

[14] D. Park. Concurrency and automata on infinite sequences. In *Theoretical Computer Science: 5th GI-Conference*, volume 104 of *LNCS*. Springer, 1981.

[15] A. Rabinovich and B. A. Trakhtenbrot. Behavior structures and nets. *Fund. Inform.*, 11(4):357–403, 1988.

[16] R. van Glabbeek and U. Goltz. Equivalence notions for concurrent systems and refinement of actions. In *MFCS '89*, volume 379 of *LNCS*, pages 237–248. Springer, 1989.

[17] R. van Glabbeek and U. Goltz. Refinement of actions and equivalence notions for concurrent systems, 1998. http://theory.stanford.edu/ rvg/abstracts.html.

[18] W. Vogler. Deciding history preserving bisimilarity. In *Automata, languages and programming '91*, volume 510 of *LNCS*, pages 495–505. Springer, 1991.

[19] W. Vogler. Bisimulation and action refinement. *Theor. Comp. Sci.*, 114(1):173–200, 1993.

[20] W. Vogler. Generalized OM-bisimulation. *Inform. and Comput.*, 118(1):38–47, 1995.

[21] G. Winskel and M. Nielsen. Models for concurrency. In *Handbook of logic in computer science, Vol. 4*, Oxford Sci. Publ., pages 1–148. Oxford Univ. Press, New York, 1995.

A Systems with Transitive Independence Relation

Here we will give the detailed proof of the decidability of HHPB for the full class of finite systems with transitive independence relation. As described in Sec. 5.2 the essence of the proof is the observation that concurrently occurring self-loops have always to be matched to self-loops. We will first give the precise definition of what it means for a self-loop to occur concurrently, and then formulate and prove the corresponding lemma.

Definition 11 *Assume a given net N. Let t be a self-loop transition of N, and let r be some run of N. We say the self-loop t is* concurrently occurring at r *iff*

- t *is enabled at* r*, and*
- *there exists* t'*, s. t.* $t \; I \; t'$ *and we have* $r \stackrel{t'}{\to} r.t'$ *or* $BEn(r, t')$.

Lemma 4 *Let $\mathcal{H}$ be a history-preserving bisimulation relating two nets with transitive independence relation, N_1, N_2.*

- *Whenever* $(r_1.t_1, r_2.t_2) \in \mathcal{H}$*, and* t_1 *is a concurrently occurring self-loop at* r_1*, then* t_2 *is a self-loop as well.*
- *Vice versa.*

PROOF: To prove the first part of the lemma let $(r_1.t_1, r_2.t_2) \in \mathcal{H}$ and let t_1 be a concurrently occurring self-loop at r_1. First assume we have $t_1' \; I \; t_1$, such that $r_1 \stackrel{t_1'}{\to} r_1.t_1'$. Clearly we have $(r_1.t_1.t_1, r_2.t_2.t_2^*) \in \mathcal{H}$ for some $t_2^* \; D \; t_2$, and $(r_1.t_1.t_1.t_1', r_2.t_2.t_2^*.t_2') \in \mathcal{H}$ for some t_2', s. t. $t_2' \; I \; t_2$ and $t_2' \; I \; t_2^*$. With transitivity of independence the latter leads to a contradiction with the requirement $t_2^* \; D \; t_2$, unless $t_2^* = t_2$. But if $t_2^* = t_2$, then t_2 must be a self-loop because it can occur twice consecutively.

Secondly, assume we have $t_1' \; I \; t_1$, such that $BEn(r_1, t_1')$. A similar argument shows that t_2 must be a self-loop, too.

The second part of the lemma can be proved by a symmetric argument. □

This lemma ensures that we do not need to consider the unfoldings of concurrently occurring self-loops. It is sufficient to match one instance of a concurrently occurring self-loop transition, and to make sure it is really matched to a self-loop.

This idea is translated into what we shall call 'No Self-loop Unfolding' (NSU) HPB. After giving the definition we will show that for systems with transitive independence relation this new kind of bisimilarity indeed coincides with (hereditary) history-preserving bisimilarity.

Note that in the following we will make use of the convention introduced in Sec. 4.1.

Definition 12 *A NSU (No Self-loop Unfolding) history-preserving bisimulation between two nets N_1 and N_2 consists of a set $\mathcal{H}_{NSU}$ of pairs (r_1, r_2) such that*

(i) *Whenever* $(r_1, r_2) \in \mathcal{H}_{NSU}$*, then* r_1 *is a run of* N_1*,* r_2 *is a run of* N_2*, and* r_1 *and* r_2 *are synchronous.*

(ii) $(\varepsilon, \varepsilon) \in \mathcal{H}_{NSU}$.

(iii) *Whenever* $(r_1, r_2) \in \mathcal{H}_{NSU}$ *and* $r_1 \stackrel{t_1}{\rightarrow} r_1.t_1$ *for some* t_1*, such that* t_1 *is* not *a concurrently occurring self-loop at* r_1*, then there exists* t_2*, such that* $r_2 \stackrel{t_2}{\rightarrow} r_2.t_2$ *and* $(r_1.t_1, r_2.t_2) \in \mathcal{H}_{NSU}$.

(iv) *Vice versa.*

(v) *Whenever* $(r_1, r_2) \in \mathcal{H}_{NSU}$ *and* $r_1 \stackrel{t_1}{\rightarrow} r_1.t_1$ *for some* t_1*, such that* t_1 *is a concurrently occurring self-loop at* r_1*, and there exists no* x_2 *such that* $(t_1, x_2) \in BEn(r)$*, then there exists* t_2*, such that* t_2 *is a self-loop,* $r_2 \stackrel{t_2}{\rightarrow} r_2.t_2$*, and* $(r_1.t_1, r_2.t_2) \in \mathcal{H}_{NSU}$.

(vi) *Vice versa.*

A NSU history-preserving bisimulation is hereditary when it further satisfies

(vii) *Whenever* $(r_1, r_2) \in \mathcal{H}_{NSU}$ *and* $t_1 \in BEn(r_1)$ *and* $t_2 \in BEn(r_2)$ *for some* t_1, t_2 *such that* $last(r_1, t_1) = last(r_2, t_2)$*, then* $(\delta(r_1, t_1), \delta(r_2, t_2)) \in \mathcal{H}_{NSU}$.

We say two nets are (hereditary) NSU history-preserving bisimilar iff there is a (hereditary) NSU HPB relating them.

Lemma 5 *Two nets with transitive independence relation are (hereditary) history-preserving bisimilar iff they are (hereditary) NSU history-preserving bisimilar.*

PROOF: With lemma 4 it is easy to check that every (hereditary) HPB is also a (hereditary) NSU HPB.

For the non-trivial direction let $\mathcal{H}_{NSU}$ be a (hereditary) NSU HPB. Define $\mathcal{H}$ by unfolding self-loop matches inductively as follows:

Base Step $\mathcal{H} = \mathcal{H}_{NSU}$,

Inductive Step Whenever $rr' \in \mathcal{H}$ and t_1, t_2 is a pair of concurrently occurring self-loops at r_1, r_2, s. t. $(t_1, t_2) \in BEn(r)$ then $r.t.r' \in \mathcal{H}$.

It is easy to check that $\mathcal{H}$ is a (hereditary) HPB. □

We can restrict our attention to the special class of *minimal* (hereditary) NSU HPBs, which strictly do not contain any unfoldings of concurrently occurring self-loops.

Definition 13 *A (hereditary) NSU HPB $\mathcal{H}_{NSU}$ is* minimal *iff*

- *Whenever $r.t.r' \in \mathcal{H}_{NSU}$ and t_1 is a concurrently occurring self-loop at r_1, then there exists* no *x_2 such that $(t_1, x_2) \in BEn(r)$.*
- *Vice versa.*

Lemma 6 *Two nets are (hereditary) NSU history-preserving bisimilar iff there exists a minimal (hereditary) NSU history-preserving bisimulation.*

PROOF: We can simply 'collapse' any given (hereditary) NSU HPB $\mathcal{H}_{NSU}$ to a minimal one: erase all tuples that violate the above conditions from $\mathcal{H}_{NSU}$. Clearly, the result is still a (hereditary) NSU HPB. □

Minimal (hereditary) NSU HPBs between systems of our subclass look exactly like (hereditary) HPBs of systems with transitive independence relation and no self-loops. They meet all characterisics that made it possible to find a decision procedure for the latter subclass. In particular, the number of joint transitions which one can backtrack over is bound by the size of the maximal independence clique. So, we get the following result.

Lemma 7 *Hereditary NSU HP bisimilarity is decidable for finite systems with transitive independence relation.*

PROOF: By lemma 6 it is sufficient to check whether there exists a minimal hereditary NSU history-preserving bisimulation. But this is clearly decidable for our subclass. We only need to adapt the steps of the proof of the decidability of (n)-hereditary HPB to show that the corresponding notion of (n)-hereditary NSU HPB is decidable for our subclass. □

With this and lemma 5 we immediately get decidability for the whole class of finite systems with transitive independence relation.

Recent BRICS Report Series Publications

RS-99-4 **Sibylle B. Fröschle and Thomas Troels Hildebrandt.** ***On Plain and Hereditary History-Preserving Bisimulation*****. February 1999. 21 pp.**

RS-99-3 **Peter Bro Miltersen.** ***Two Notes on the Computational Complexity of One-Dimensional Sandpiles*****. February 1999. 8 pp.**

RS-99-2 **Ivan B. Damgård.** ***An Error in the Mixed Adversary Protocol by Fitzi, Hirt and Maurer*****. February 1999. 4 pp.**

RS-99-1 **Marcin Jurdziński and Mogens Nielsen.** ***Hereditary History Preserving Simulation is Undecidable*****. January 1999. 15 pp.**

RS-98-55 **Andrew D. Gordon, Paul D. Hankin, and Søren B. Lassen.** ***Compilation and Equivalence of Imperative Objects (Revised Report)*****. December 1998. iv+75 pp. This is a revision of Technical Report 429, University of Cambridge Computer Laboratory, June 1997, and the earlier BRICS report RS-97-19, July 1997. Appears in Ramesh and Sivakumar, editors,** ***Foundations of Software Technology and Theoretical Computer Science: 17th Conference*****, FST&TCS '97 Proceedings, LNCS 1346, 1997, pages 74–87.**

RS-98-54 **Olivier Danvy and Ulrik P. Schultz.** ***Lambda-Dropping: Transforming Recursive Equations into Programs with Block Structure*****. December 1998. 55 pp. To appear in** ***Theoretical Computer Science*****.**

RS-98-53 **Julian C. Bradfield.** ***Fixpoint Alternation: Arithmetic, Transition Systems, and the Binary Tree*****. December 1998. 20 pp.**

RS-98-52 **Josva Kleist and Davide Sangiorgi.** ***Imperative Objects and Mobile Processes*****. December 1998. 22 pp. Appears in Gries and de Roever, editors,** ***IFIP Working Conference on Programming Concepts and Methods*****, PROCOMET '98 Proceedings, 1998, pages 285–303.**

RS-98-51 **Peter Krogsgaard Jensen.** ***Automated Modeling of Real-Time Implementation*****. December 1998. 9 pp. Appears in** ***The 13th IEEE Conference on Automated Software Engineering*****, ASE '98 Doctoral Symposium Proceedings, 1998, pages 17–20.**