BRICS

Basic Research in Computer Science

BRICS RS-99-19 Jurdziński & Nielsen: Hereditary History Preserving Bisimilarity is Undecidable

Hereditary History Preserving Bisimilarity is Undecidable

Marcin Jurdziński
Mogens Nielsen

BRICS Report Series **RS-99-19**

ISSN 0909-0878 **June 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/99/19/`

Hereditary History Preserving Bisimilarity Is Undecidable

Marcin Jurdziński* Mogens Nielsen*
BRICS†
Department of Computer Science
University of Aarhus

June 1999

Abstract

We show undecidability of hereditary history preserving bisimilarity for finite asynchronous transition systems by a reduction from the halting problem of deterministic 2-counter machines. To make the proof more transparent we introduce an intermediate problem of checking domino bisimilarity for origin constrained tiling systems. First we reduce the halting problem of deterministic 2-counter machines to origin constrained domino bisimilarity. Then we show how to model domino bisimulations as hereditary history preserving bisimulations for finite asynchronous transitions systems. We also argue that the undecidability result holds for finite 1-safe Petri nets, which can be seen as a proper subclass of finite asynchronous transition systems.

1 Hereditary history preserving bisimilarity

Definition 1 (Labelled asynchronous transition system)
A *labelled asynchronous transition system* is a tuple $A = (S, s^{\text{ini}}, E, \to, L, \lambda, I)$, where S is its set of *states*, $s^{\text{ini}} \in S$ is the *initial state*, E is the set of *events*, $\to \subseteq S \times E \times S$ is the set of *transitions*, L is the set of labels, and $\lambda : E \to L$ is the *labelling function*, and $I \subseteq E^2$ is the *independence relation* which is irreflexive and symmetric. We often write $s \xrightarrow{e} s'$, instead of $(s, e, s') \in \to$. Moreover, the following conditions have to be satisfied:

1. if $s \xrightarrow{e} s'$, and $s \xrightarrow{e} s''$, then $s' = s''$,

*Address: BRICS, Department of Computer Science, University of Aarhus, Ny Munkegade, Building 540, 8000 Aarhus C, Denmark. Emails: {mju,mn}@brics.dk.
†**B**asic **R**esearch in **C**omputer **S**cience,
Centre of the Danish National Research Foundation.

2. if $(e, e') \in I$, $s \xrightarrow{e} s'$, and $s' \xrightarrow{e'} t$, then $s \xrightarrow{e'} s''$, and $s'' \xrightarrow{e} t$ for some $s'' \in S$.

An asynchronous transition system is *coherent* if it satisfies one further condition:

3. if $(e, e') \in I$, $s \xrightarrow{e} s'$, and $s \xrightarrow{e'} s''$, then $s' \xrightarrow{e'} t$, and $s'' \xrightarrow{e} t$ for some $t \in S$.

[Definition 1] $\square$

Asynchronous transition systems were introduced independently by Bednarczyk [Bed88], and Shields [Shi85]. Winskel and Nielsen [WN95, NW96] give a thorough survey and establish formal relationships between asynchronous transition systems and other models for concurrency, such as Petri nets, and event structures.

The definition of an asynchronous transition system may seem to be quite liberal, in the sense that it requires the transition system to satisfy very few properties related to its independence relation. For example, labelled asynchronous transition systems arising from finite labelled 1-safe Petri nets form a proper subclass of the class of all finite coherent asynchronous transitions systems. We want to stress, however, that we have chosen this liberal definition only for technical convenience. In fact, as we show in section 4, the proof of our undecidability result goes through even for finite labelled 1-safe Petri nets.

Let $A = (S, s^{\mathrm{ini}}, E, \to, L, \lambda, I)$ be a labelled asynchronous transition system. A sequence of events $\overline{e} = \langle e_1, e_2, \ldots, e_n \rangle \in E^n$ is a *run* of A if there are states $s_1, s_2, \ldots, s_{n+1} \in S$, such that $s_1 = s^{\mathrm{ini}}$, and for all $i \in [n]$ we have $s_i \xrightarrow{e_i} s_{i+1}$. We denote the set of runs of A by $\mathrm{Run}(A)$. We say that $k \in [n]$ is *most recent* in $\overline{e}$, and we denote it by $k \in \mathrm{MR}(\overline{e})$, if and only if $(e_k, e_\ell) \in I$ for all $k < \ell \leq n$. Note that if $k \in \mathrm{MR}(\overline{e})$ then $\overline{e} \oslash k = \langle e_1, \ldots, e_{k-1}, e_{k+1}, \ldots, e_n \rangle \in \mathrm{Run}(A)$.

Definition 2 (Hereditary history preserving bisimulation)
Let $A_i = (S_i, s_i, E_i, \to_i, L, \lambda_i, I_i)$ for $i \in \{1, 2\}$ be labelled asynchronous transition systems. A relation $B \subseteq \mathrm{Run}(A_1) \times \mathrm{Run}(A_2)$ is a *hereditary history preserving* (hhp-) bisimulation relating A_1 and A_2 if the following conditions are satisfied:

1. $(\varepsilon, \varepsilon) \in B$,

and if $(\overline{e_1}, \overline{e_2}) \in B$ then:

2. for all $e_1 \in E_1$, if $\overline{e_1} \cdot e_1 \in \mathrm{Run}(A_1)$, then there exists $e_2 \in E_2$, such that $\overline{e_2} \cdot e_2 \in \mathrm{Run}(A_2)$, and $\lambda_1(e_1) = \lambda_2(e_2)$, and $(\overline{e_1} \cdot e_1, \overline{e_2} \cdot e_2) \in B$,

3. for all $e_2 \in E_2$, if $\overline{e_2} \cdot e_2 \in \mathrm{Run}(A_2)$, then there exists $e_1 \in E_1$, such that $\overline{e_1} \cdot e_1 \in \mathrm{Run}(A_1)$, and $\lambda_1(e_1) = \lambda_2(e_2)$, and $(\overline{e_1} \cdot e_1, \overline{e_2} \cdot e_2) \in B$,

4. $k \in \mathrm{MR}(\overline{e_1})$, if and only if $k \in \mathrm{MR}(\overline{e_2})$,

5. if $k \in \mathrm{MR}(\overline{e_1}) = \mathrm{MR}(\overline{e_2})$, then $(\overline{e_1} \oslash k, \overline{e_2} \oslash k) \in B$.

A relation $B \subseteq \mathrm{Run}(A_1) \times \mathrm{Run}(A_2)$ is a *history preserving* (hp-) bisimulation if it satisfies conditions 1.–4. [Definition 2] $\square$

We say that two asynchronous transition systems A_1, and A_2 are hereditary history preserving (hhp-) *bisimilar*, if there is a hhp-bisimulation relating them; they are history preserving (hp-) bisimilar, if there is a history preserving bisimulation relating them.

Remark 1 Notice that for standard labelled transition systems, *i.e.*, asynchronous transition systems without the independence relation, (h)hp-bisimilarity coincides with the standard bisimilarity. If the independence relation is empty then only the latest event is most recent, so conditions 4. and 5. are redundant. [Remark 1] $\diamond$

Remark 2 Hhp- and hp-bisimilarities are so called *non-interleaving* notions of equivalence for concurrent programs. They are different from the standard interleaving bisimulation in that they relate behaviours of concurrent programs viewed as *partial orders* of events ordered by *causality* relation, rather than as *interleavings*, *i.e.*, sequences of events. We sketch below the standard partial order semantics for asynchronous transition systems, and we explain how notions of (h)hp-bisimilarity arise naturally in this context. The rest of the paper does not rely on any of the concepts discussed in this remark, so it can be safely omited in first reading.

Let $A = (S, s^{\mathrm{ini}}, E, \to, L, \lambda, I)$ be a labelled asynchronous transition system. For every run $\overline{e} \in E^n$ of A the independence relation induces an E-labelled partial order $\pi(\overline{e}) = ([n], \trianglelefteq, \varepsilon)$, where $\varepsilon : [n] \to E$ is the labelling function. For all $i \in [n]$ we set $\varepsilon(i) = e_i$. For $i, j \in [n]$ we define $i \lessdot j$ to hold, if $i \leq j$, and $(e_i, e_j) \notin I$. We get $\trianglelefteq$ as the transitive closure of $\lessdot$. For $\overline{e}, \overline{e'} \in \mathrm{Run}(A)$ we define $\overline{e} \cong \overline{e'}$ to hold, if the corresponding labelled partial orders $\pi(\overline{e})$ and $\pi(\overline{e'})$ are isomorphic. This is clearly an equivalence relation.

The set of *partial order runs* $\mathrm{PORun}(A)$ of A is the set $\mathrm{Run}(A)/_{\cong}$, *i.e.*, isomorphism classes of E-labelled partial orders corresponding to runs of A. Let $\tau = (|\tau|, \trianglelefteq, \varepsilon)$ be a partial order run of A, where $\varepsilon : |\tau| \to E$ is the labelling function. A *partial order behaviour* corresponding to run τ is the L-labelled partial order $\lambda(\tau) = \big(|\tau|, \trianglelefteq, \lambda \circ \varepsilon\big)$.

A non-interleaving notion of behavioural equivalence for concurrent programs should respect the partial order semantics, *i.e.*, it should relate isomorphic partial order behaviours.

Before we give an alternative definition of hereditary history preserving bisimulation we introduce some notations. If $\tau \in \mathrm{PORun}(A)$, and $\overline{e} \in \tau$, and

$\overline{e} \cdot e \in \mathrm{Run}(A)$, for some $e \in E$, then we write $\tau \oplus e$ for $[\overline{e} \cdot e]_{\cong} \in \mathrm{PORun}(A)$; otherwise $\tau \oplus e$ is undefined. Similarly, if $\overline{e} \in \tau$, and $\overline{e} = \overline{e'} \cdot e$, for some $\overline{e'} \in \mathrm{Run}(A)$, and $e \in E$, then we write $\tau \ominus e$ for $[\overline{e'}]_{\cong} \in \mathrm{PORun}(A)$; otherwise $\tau \ominus e$ is undefined. By $\mathrm{Iso}(L)$ we denote the set of isomorphisms of L-labelled partial orders. By $\mathrm{POBeh}(A_1, A_2)$ we denote the set of triples $(\tau_1, \Xi, \tau_2) \in \mathrm{PORun}(A_1) \times \mathrm{Iso}(L) \times \mathrm{PORun}(A_2)$, such that $\Xi : |\tau_1| \to |\tau_2|$ is an isomorphism of L-labelled partial orders $\lambda_1(\tau_1)$ and $\lambda_2(\tau_2)$, *i.e.*, an isomorphism of partial order behaviours corresponding to partial order runs τ_1 and τ_2.

Let $(\tau_1, \Xi, \tau_2) \in \mathrm{POBeh}(A_1, A_2)$, and let $\tau_i \oplus e_i \in \mathrm{PORun}(A_i)$ be defined for some $e_i \in E_i$ for $i = 1, 2$. If the unique extension $\Xi' : |\tau_1 \oplus e_1| \to |\tau_2 \oplus e_2|$ of Ξ (*i.e.*, $\Xi' \supset \Xi$) is an isomorphism of the L-labelled partial orders $\lambda_1(\tau_1 \oplus e_1)$, and $\lambda_2(\tau_2 \oplus e_2)$, then by $(\tau_1, \Xi, \tau_2) \oplus (e_1, e_2)$ we denote the triple $(\tau_1 \oplus e_1, \Xi', \tau_2 \oplus e_2) \in \mathrm{POBeh}(A_1, A_2)$. Otherwise $(\tau_1, \Xi, \tau_2) \oplus (e_1, e_2)$ is undefined.

Similarly, if $\tau_i \ominus e_i \in \mathrm{PORun}(A_i)$ are defined for some $e_i \in E_i$, and $i = 1, 2$, and moreover the restriction Ξ' of Ξ to $|\tau_1 \ominus e_1|$ is an isomorphism of L-labelled partial orders $\lambda(\tau_1 \ominus e_1)$, and $\lambda_2(\tau_2 \ominus e_2)$, then by $(\tau_1, \Xi, \tau_2) \ominus (e_1, e_2)$ we denote the triple $(\tau_1 \ominus e_1, \Xi', \tau_2 \ominus e_2) \in \mathrm{POBeh}(A_1, A_2)$; otherwise $(\tau_1, \Xi, \tau_2) \ominus (e_1, e_2)$ is undefined.

Now we are ready to give an alternative definition of hereditary history preserving bisimulation.

Definition 2' (Hereditary history preserving bisimulation)
Let $A_i = (S_i, s_i, E_i, \to_i, L, \lambda_i, I_i)$ for $i \in \{1, 2\}$ be labelled asynchronous transition systems. A relation $B \subseteq \mathrm{POBeh}(A_1, A_2)$ is a hereditary history preserving (hhp-) bisimulation relating A_1 and A_2 if the following conditions are satisfied:

1. $(\emptyset, \emptyset, \emptyset) \in B$,

and if $(\tau_1, \Xi, \tau_2) \in B$ then:

2. for all $e_1 \in E_1$, if $\tau_1 \oplus e_1$ is defined, then there is $e_2 \in E_2$, such that $(\tau_1, \Xi, \tau_2) \oplus (e_1, e_2)$ is defined, and $(\tau_1, \Xi, \tau_2) \oplus (e_1, e_2) \in B$,

3. for all $e_2 \in E_2$, if $\tau_2 \oplus e_2$ is defined, then there is $e_1 \in E_1$, such that $(\tau_1, \Xi, \tau_2) \oplus (e_1, e_2)$ is defined, and $(\tau_1, \Xi, \tau_2) \oplus (e_1, e_2) \in B$,

4. for all $e_i \in E_i$, and $i = 1, 2$, if $(\tau_1, \Xi, \tau_2) \ominus (e_1, e_2)$ is defined, then $(\tau_1, \Xi, \tau_2) \ominus (e_1, e_2) \in B$.

A relation $B \subseteq \mathrm{POBeh}(A_1, A_2)$ is a hp-bisimulation relating A_1 and A_2 if it satisfies conditions 1.–3. [Definition 2'] □

It can be shown that notions of bisimilarity induced by Definitions 2 and 2' are equivalent [NC95]; we have decided to make Definition 2 the primary one here, since it is technically simpler and more intuitive. [Remark 2] ◇

Hp-bisimilarity has been introduced by many authors, among others Rabinovich and Trakhtenbrot [RT88], and van Glabbeek and Goltz [vGG89]. Hhp-bisimilarity has been introduced by Bednarczyk [Bed91], and discovered independently by Joyal *et al.* [JNW96], as the open map bisimulation for observations being labelled partial orders.

Nielsen and Clausen [NC95] have established game and logic characterizations of hhp-bisimilarity. An hhp-bisimulation game is played by two players Spoiler and Duplicator on the graphs of (partial order) runs of asynchronous transition systems. Spoiler wants to show that the two asynchronous transition systems are not hhp-bisimilar. Duplicator has a winning strategy if and only if the two asynchronous transition systems are hhp-bisimilar, and an hhp-bisimulation is in fact a winning strategy for Duplicator. Hhp-bisimulation games can be seen as Ehrenfeucht-Fraïssé games for a modal logic with a backwards modality interpreted over the graphs of partial order runs of asynchronous transition systems, *i.e.*, there is a hhp-bisimulation relating two asynchronous transition systems, if and only if they are indistinguishable by formulas of the logic.

History preserving bisimilarity has been shown to be decidable for 1-safe Petri nets by Vogler [Vog91], and to be **DEXP**-complete by Jategaonkar, and Mayer [JM96]. Decidability of hereditary history preserving bisimulation for 1-safe Petri nets has remained open instead [NW96, FH99]. Fröschle and Hildebrandt [FH99] have discovered an infinite hierarchy of bisimilarity notions refining hp-bisimilarity, and coarser than hhp-bisimilarity; hhp-bisimilarity is the intersection of all the bisimilarities in the hierarchy. They have shown all the bisimilarities in the hierarchy to be decidable for 1-safe Petri nets. Fröschle [Frö99] has shown hhp-bisimilarity to be decidable for BPP-processes.

The main result of this paper is the following theorem proved in section 3.

Theorem 3 (Undecidability of hhp-bisimilarity)
Hhp-bisimilarity for finite labelled asynchronous transition systems is undecidable.

2 Domino bisimilarity is undecidable

2.1 Domino bisimilarity

Definition 4 (Origin constrained tiling system)
An *origin constrained tiling system* $T = \big(D, D^{\mathrm{ori}}, (H, H^0), (V, V^0), L, \lambda\big)$ consists of a set D of *dominoes*, its subset $D^{\mathrm{ori}} \subseteq D$ called the *origin constraint*, two *horizontal compatibility* relations $H, H^0 \subseteq D^2$, two *vertical compatibility* relations $V, V^0 \subseteq D^2$, a set L of *labels*, and a *labelling function* $\lambda : D \to L$.

[Definition 4] □

A *configuration* of T is a triple $(d, x, y) \in D \times \mathbb{N} \times \mathbb{N}$, such that if $x = y = 0$ then $d \in D^{\mathrm{ori}}$. In other words, in the "origin" position $(x, y) = (0, 0)$ of the non-negative integer grid only dominoes from the origin constraint D^{ori} are allowed.

Let (d, x, y), and (d', x', y') be configurations of T such that $|x' - x| + |y' - y| = 1$, *i.e.*, the positions (x, y), and (x', y') are neighbours in the non-negative integer grid. Without loss of generality we may assume that $x + y < x' + y'$. We say that configurations (d, x, y), and (d', x', y') are *compatible* if either of the two conditions below holds:

- $x' = x$, and $y' = y + 1$, and
  if $y = 0$, then $(d, d') \in V^0$, and if $y > 0$, then $(d, d') \in V$, or

- $x' = x + 1$, and $y' = y$, and
  if $x = 0$, then $(d, d') \in H^0$, and if $x > 0$, then $(d, d') \in H$.

Definition 5 (Domino bisimulation)
Let $T_i = \big(D_i, D_i^{\mathrm{ori}}, (H_i, H_i^0), (V_i, V_i^0), L_i, \lambda_i\big)$ for $i \in \{1, 2\}$ be origin constrained tiling systems. A relation $B \subseteq D_1 \times D_2 \times \mathbb{N} \times \mathbb{N}$ is a *domino bisimulation* relating T_1 and T_2, if the following conditions are satisfied for all $i \in \{1, 2\}$:

1. for all $d_i \in D_i^{\mathrm{ori}}$, there exists $d_{3-i} \in D_{3-i}^{\mathrm{ori}}$, such that $\lambda_1(d_1) = \lambda_2(d_2)$, and $(d_1, d_2, 0, 0) \in B$,

2. for all $x, y \in \mathbb{N}$, such that $(x, y) \neq (0, 0)$, and $d_i \in D_i$, there exists $d_{3-i} \in D_{3-i}$, such that $\lambda_1(d_1) = \lambda_2(d_2)$, and $(d_1, d_2, x, y) \in B$,

3. if $(d_1, d_2, x, y) \in B$, then for all $x', y' \in \mathbb{N}$, and $d_i' \in D_i$, if configurations (d_i, x, y), and (d_i', x', y') of T_i are compatible, then there exists $d_{3-i}' \in D_{3-i}$, such that $\lambda_1(d_1) = \lambda_2(d_2)$, and configurations (d_{3-i}, x, y), and (d_{3-i}', x', y') of T_{3-i} are compatible, and $(d_1', d_2', x', y') \in B$.

[Definition 5] $\square$

We say that two tiling systems are *domino bisimilar* if and only if there is a domino bisimulation relating them.

Theorem 6 (Undecidability of domino bisimilarity)
Domino bisimilarity is undecidable for origin constrained tiling systems.

The proof is a reduction from the halting problem for deterministic 2-counter machines. For a deterministic 2-counter machine M we define in section 2.3 two origin constrained tiling systems T_1, and T_2, enjoying the following property.

Proposition 7 Machine M does not halt, if and only if there is a domino bisimulation relating T_1 and T_2.

2.2 Counter machines

A 2-counter machine M consists of a finite program with the set L of instruction labels, and instructions of the form:

- `start:` `goto` ℓ
- ℓ: $\mathtt{c}_i$ `:=` $\mathtt{c}_i$ `+ 1; goto` m
- ℓ: `if` $\mathtt{c}_i$ `= 0 then` $\mathtt{c}_i$ `:=` $\mathtt{c}_i$ `+ 1; goto` m
  `else` $\mathtt{c}_i$ `:=` $\mathtt{c}_i$ `- 1; goto` n
- `halt:`

where $i = 1, 2$; $\ell, m, n \in L$, and $\{\mathtt{start}, \mathtt{halt}\} \subset L$. A configuration of M is a triple $(\ell, x, y) \in L \times \mathbb{N} \times \mathbb{N}$, where ℓ is the label of the current instruction, and x, and y are the values stored in counters $\mathtt{c}_1$, and $\mathtt{c}_2$, respectively; we denote the set of configurations of M by $\mathrm{Conf}(M)$. The semantics of 2-counter machines is standard: let $\vdash_M \subseteq \mathrm{Conf}(M) \times \mathrm{Conf}(M)$ be the usual one-step derivation relation on configurations of M; by $\vdash_M^+$ we denote the reachability (in at least one step) relation for configurations, *i.e.*, the transitive closure of $\vdash_M$.

Before we give a reduction from the halting problem of 2-counter machines to origin constrained domino bisimilarity let us take a look at the directed graph $(\mathrm{Conf}(M), \vdash_M)$, with configurations of M as vertices, and edges denoting derivation in one step. Since machine M is deterministic, for each configuration there is at most one outgoing edge; moreover only halting configurations have no outgoing edges. It follows that connected components of the graph $(\mathrm{Conf}(M), \vdash_M)$ are either trees with edges going to the root which is the unique halting configuration in the component, or have no halting configuration at all. This observation implies the following proposition.

Proposition 8 Let M be a 2-counter machine. The following conditions are equivalent:

1. machine M halts on input $(0, 0)$, *i.e.*, $(\mathtt{start}, 0, 0) \vdash_M^+ (\mathtt{halt}, x, y)$ for some $x, y \in \mathbb{N}$,

2. $(\mathtt{start}, 0, 0) \sim_M (\mathtt{halt}, x, y)$ for some $x, y \in \mathbb{N}$, where $\sim_M \subseteq \mathrm{Conf}(M) \times \mathrm{Conf}(M)$ is the symmetric, and transitive closure of $\vdash_M$.

2.3 The reduction

Now we go for a proof of Proposition 7. The idea is to design a tiling system which "simulates" behaviour of a 2-counter machine.

Let M be a 2-counter machine. We construct a tiling system T_M with the set L of instruction labels of M as the set of dominoes, and the identity function on L as the labelling function. Note that this implies that all tuples belonging to a domino bisimulation relating copies of T_M are of the form

(ℓ, ℓ, x, y), so we can identify them with configurations of M, *i.e.*, sometimes we will make no distinction between (ℓ, ℓ, x, y) and $(\ell, x, y) \in \mathrm{Conf}(M)$ for $\ell \in L$.

We define the horizontal compatibility relations $H_M, H_M^0 \subseteq L \times L$ of the tiling system T_M as follows:

- $(\ell, m) \in H_M$ if and only if either of the instructions below is an instruction of machine M:
  - ℓ: $\mathtt{c}_1$ `:=` $\mathtt{c}_1$ `+ 1; goto` m
  - m: `if` $\mathtt{c}_1$ `= 0 then` $\mathtt{c}_1$ `:=` $\mathtt{c}_1$ `+ 1; goto` n
 `else` $\mathtt{c}_1$ `:=` $\mathtt{c}_1$ `- 1; goto` ℓ
- $(\ell, m) \in H_M^0$ if and only if $(\ell, m) \in H_M$, or the instruction below is an instruction of machine M:
  - ℓ: `if` $\mathtt{c}_1$ `= 0 then` $\mathtt{c}_1$ `:=` $\mathtt{c}_1$ `+ 1; goto` m
 `else` $\mathtt{c}_1$ `:=` $\mathtt{c}_1$ `- 1; goto` n

The vertical compatibility relations V_M, and V_M^0 are defined in the same way, with $\mathtt{c}_1$ instructions replaced with $\mathtt{c}_2$ instructions. We also take $D_M^{\mathrm{ori}} = L$, *i.e.*, all dominoes are allowed in position $(0, 0)$. Note that the identity function is a 1-1 correspondence between configurations of M, and configurations of the tiling system T_M; from now on we will hence identify configurations of M and T_M. It follows immediately from the construction of T_M, that two configurations $c, c' \in \mathrm{Conf}(M)$ are compatible as configurations of T_M, if and only if $c \vdash_M c'$, or $c' \vdash_M c$, *i.e.*, compatibility relation of T_M coincides with the symmetric closure of $\vdash_M$. By $\approx_M$ we denote the symmetric and transitive closure of the compatibility relation of T_M. The following proposition is then straightforward.

Proposition 9 The two relations $\sim_M$, and $\approx_M$ coincide.

Now we are ready to define the two origin constrained tiling systems T_1, and T_2, postulated in Proposition 7. The idea is to have two independent and slightly pruned copies of T_M in T_2: one without the initial configuration $(\mathtt{start}, 0, 0)$, and the other without any halting configurations $(\mathtt{halt}, x, y)$. The other tiling system T_1 is going to have three independent copies of T_M: the two of T_2, and moreover, another full copy of T_M.

More formally we define $D_2 = \big(L \times \{1, 2\}\big) \setminus \big\{(\mathtt{halt}, 2)\big\}$, and $D_2^{\mathrm{ori}} = D_2 \setminus \big\{(\mathtt{start}, 1)\big\}$, and $V_2 = \big((V_M \otimes 1) \cup (V_M \otimes 2)\big) \cap (D_2 \times D_2)$, where for a binary relation R we define $R \otimes i$ to be the relation $\big\{\big((a, i), (b, i)\big) : (a, b) \in R\big\}$. The other compatibility relations V_2^0, H_2, and H_2^0 are defined analogously from the respective compatibility relations of T_M.

The tiling system T_1 is obtained from T_2 by adding yet another independent copy of T_M, this time a complete one: $D_1 = D_2 \cup (L \times \{3\})$, and $D_1^{\text{ori}} = D_2^{\text{ori}} \cup (V_M \otimes 3)$, and $V_1 = V_2 \cup (V_M \otimes 3)$, *etc.* The labelling functions of T_1, and T_2 are defined as $\lambda_i\big((\ell, i)\big) = \ell$.

In order to show Proposition 7 we establish the following two claims.

Claim 10
If machine M halts on input $(0, 0)$, then there is no domino bisimulation relating T_1 and T_2.

Proof: If B is a domino bisimulation relating T_1 and T_2, then it must be the case that $\big((\texttt{start}, 3), (\texttt{start}, 2), 0, 0\big) \in B$, since $(\texttt{start}, 1) \notin D_2^{\text{ori}}$. But then using condition 3 of the definition of a domino bisimulation while "simulating" the halting computation of M in copy 3 of T_1, we conclude that we must have $\big((\texttt{halt}, 3), (\texttt{halt}, 2), x, y\big) \in B$ for some $x, y \in \mathbb{N}$, but this is impossible, since $(\texttt{halt}, 2) \notin D_2$. [Claim 10] ■

Claim 11
If machine M does not halt on input $(0, 0)$, then there is a domino bisimulation relating T_1 and T_2.

Proof: Suppose that M does not halt on input $(0, 0)$. We claim that the following relation B is a domino bisimulation for T_1 and T_2:

$$\begin{array}{l}\Big\{ \big((\ell, i), (\ell, i), x, y\big) \;:\; i \in \{1, 2\} \text{ and } \big((\ell, i), x, y\big) \in \mathrm{Conf}(T_1) \Big\} \;\cup \\ \Big\{ \big((\ell, 1), (\ell, 3), x, y\big) \;:\; (\ell, x, y) \sim_M (\texttt{halt}, x', y') \text{ for some } x', y' \in \mathbb{N} \Big\} \;\cup \\ \Big\{ \big((\ell, 2), (\ell, 3), x, y\big) \;:\; (\ell, x, y) \not\sim_M (\texttt{halt}, x', y') \text{ for all } x', y' \in \mathbb{N} \Big\}.\end{array}$$

Conditions 1. and 2. of the definition of a domino bisimulation, and condition 3. for the first component of the above union follow immediately. We check condition 3. for elements of the second and third components of the above union.

Suppose that $\big((\ell, 1), (\ell, 3), x, y\big) \in B$; note that it suffices to check that (ℓ, x, y) is not compatible with $(\texttt{start}, 0, 0)$, since copy 1 of T_1 differs from copy 3 of T_2 only in that it does not have $(\texttt{start}, 0, 0)$ as a configuration. By definition of B we have that $(\ell, x, y) \sim_M (\texttt{halt}, x', y')$ for some $x', y' \in \mathbb{N}$, so the assumption that M does not halt on input $(0, 0)$, and Proposition 8 imply that $(\texttt{start}, 0, 0) \not\sim_M (\ell, x, y)$. Then it follows by Proposition 9 that $(\texttt{start}, 0, 0) \not\approx_M (\ell, x, y)$, so in particular (ℓ, x, y) is not compatible with $(\texttt{start}, 0, 0)$ and hence we are done.

The case for $\big((\ell, 2), (\ell, 3), x, y\big) \in B$ is similar. It suffices then to check that (ℓ, x, y) is not compatible with $(\texttt{halt}, x', y')$ for any $x', y' \in \mathbb{N}$. This follows from Proposition 9 applied to the assumption that $(\ell, x, y) \not\leadsto_M (\texttt{halt}, x', y')$ for all $x', y' \in \mathbb{N}$. [Claim 11] ■

This concludes the proof of Theorem 6.

3 Hhp-bisimilarity is undecidable

The proof of Theorem 3 is a reduction from the problem of deciding bisimilarity for origin constrained tiling systems. A method of encoding a tiling system on an infinite grid in the graph of behaviours of a finite asynchronous transition system is due to Madhusudan and Thiagarajan [MT98]; we use a modified version of a gadget invented by them. For each origin constrained tiling system T we define an asynchronous transition system $A(T)$, such that the following proposition holds.

Proposition 12 There is a domino bisimulation relating origin constrained tiling systems T_1 and T_2, if and only if there is a hhp-bisimulation relating the asynchronous transition systems $A(T_1)$ and $A(T_2)$.

Let $T = \big(D, D^{\mathrm{ori}}, (H, H^0), (V, V^0), L, \lambda\big)$ be an origin constrained tiling system. We define the asynchronous transitions system $A(T)$. The schematic structure of $A(T)$ can be seen in Figure 1. The set of events is defined as:

$$
\begin{aligned}
E_{A(T)} &= \big\{ \, x_i, y_i \; : \; i \in \{0,1,2,3\} \, \big\} \\
&\cup \; \big\{ \, d_{ij}, \overline{d}_{ij} \; : \; i, j \in \{0,1,2\}, d \in D, \text{ and } d \in D^{\mathrm{ori}} \text{ if } (i,j) = (0,0) \, \big\}.
\end{aligned}
$$

By abuse of notation we sometimes write d_{xy} or $\overline{d}_{xy}$ for $x, y \in \mathbb{N}$; we always mean by that the events $d_{\hat{x}\hat{y}}$ or $\overline{d}_{\hat{x}\hat{y}}$, respectively, where for $z \in \mathbb{N}$ we define

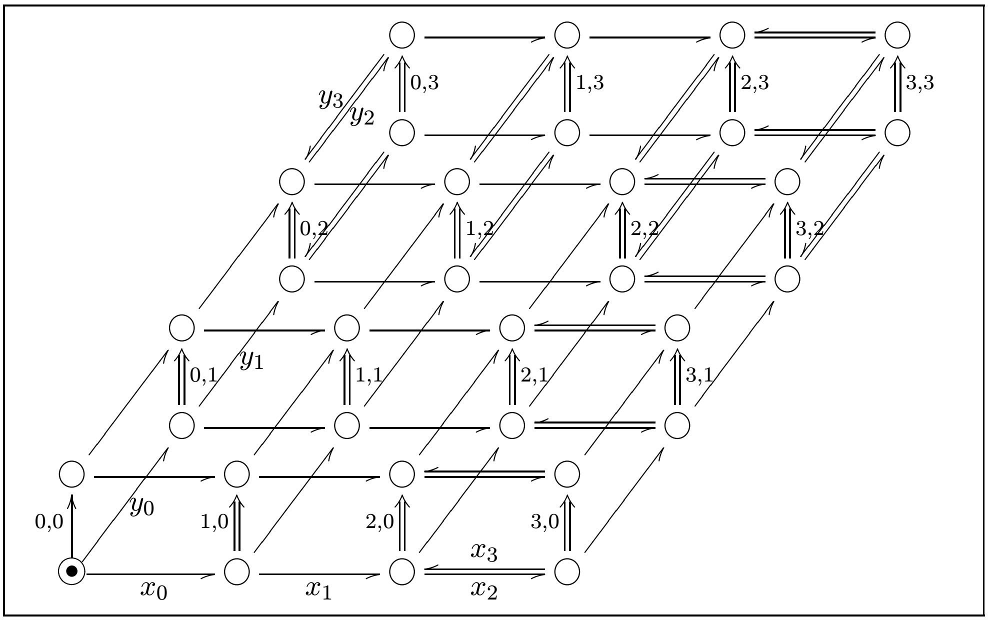
$$
\hat{z} = \begin{cases} z & \text{if } z \leq 2, \\ 2 - (z \bmod 2) & \text{if } z > 2. \end{cases}
$$

The labelling function replaces dominoes in "d"-, and "$\overline{d}$"-events, with their labels in the tiling system:

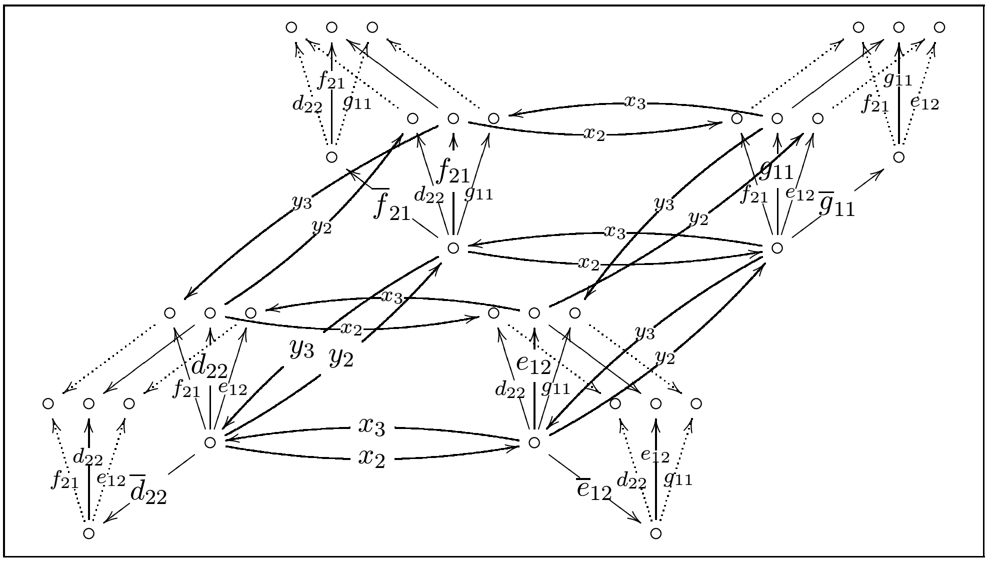
$$
\lambda_{A(T)}(e) = \begin{cases} e & \text{if } e \in \big\{ \, x_i, y_i \; : \; i \in \{0, \ldots, 3\} \, \big\}, \\ \lambda(d)_{ij} & \text{if } e = d_{ij}, \text{ for some } d \in D, \\ \overline{\lambda(d)}_{ij} & \text{if } e = \overline{d}_{ij}, \text{ for some } d \in D. \end{cases}
$$

The states, events, and transitions of $A(T)$ can be read from Figure 1; we briefly explain below how to do it.

There are sixteen states in the bottom layer of the structure in Figure 1(a). Let us identify these sixteen states with pairs of numbers shown on the vertical



(a) The structure of $A(T)$ in the large.



(b) The fine structure of the upper-right cube of $A(T)$.

Figure 1: The structure of the asynchronous transition system $A(T)$.

macro-arrows originating in these states shown in Figure 1(a). Each of these macro-arrows denotes a bundle of d_{ij}-, and $\overline{d}_{ij}$-event transitions sticking out of the state below, arranged in the fashion shown in Figure 1(b). For each state (i, j), and domino $d \in D$, there are d_{ij}-, and $\overline{d}_{ij}$-event transitions sticking out, and moreover for each state (i', j') from which there is an arrow in Figure 1(a) to state (i, j), there is a $d_{i'j'}$-event transition sticking out of (i, j). The state $(0, 0)$ is exceptional: only dominoes from the origin constraint D^{ori} are allowed as events of transitions sticking out of it. It is also the initial state of $A(T)$.

As can be seen in Figure 1(b), from both ends of the d_{ij}-event transition rooted in state (i, j), there is an x_i-event transition to the corresponding (bottom, or top) $(i \oplus 1, j)$ state, and an y_i-event transition to the corresponding $(i, j \oplus 1)$ state, where

$$i \oplus 1 = \begin{cases} i + 1 & \text{for } i < 3, \\ 2 & \text{for } i = 3. \end{cases}$$

For each $d_{i'j'}$-event transition t sticking out of state (i, j), and each $e \in D$, there can be a pair of transitions which together with t and the $\overline{e}_{ij}$-event transition form a "diamond" of transitions; the events of the transitions lying on the opposite sides of the diamond coincide then. This type of transitions is shown in Figure 1(b) as dotted arrows. The condition for the two transitions closing the diamond to exist is that configurations (d, i', j') and $\big(e, i' + |i' - i|, j' + |j' - j|\big)$ of T are compatible, or $(i', j') = (i, j)$ and $e = d$. We define the independence relation $I_{A(T)} \subseteq E_{A(T)} \times E_{A(T)}$, to be the symmetric closure of the set:

$$\begin{aligned}
&\big\{\ (x_i, y_j), (x_i, d_{ij}), (y_j, d_{ij})\ :\ i, j \in \{0, \ldots, 3\},\ \text{and } d \in D\ \big\}\ \cup \\
&\big\{\ (d_{ij}, \overline{d}_{ij})\ :\ i, j \in \{0, \ldots, 3\},\ \text{and } d \in D\ \big\}\ \cup \\
&\big\{\ (d_{0j}, \overline{e}_{1j})\ :\ j \in \{0, \ldots, 3\},\ \text{and } (d, e) \in H^0\ \big\}\ \cup \\
&\big\{\ (d_{ij}, \overline{e}_{(i+1)j})\ :\ i \in \{1, 2, 3\}, j \in \{0, \ldots, 3\},\ \text{and } (d, e) \in H\ \big\} \\
&\big\{\ (d_{i0}, \overline{e}_{i1})\ :\ i \in \{0, \ldots, 3\},\ \text{and } (d, e) \in V^0\ \big\}\ \cup \\
&\big\{\ (d_{ij}, \overline{e}_{i(j+1)})\ :\ i \in \{0, \ldots, 3\}, j \in \{1, 2, 3\},\ \text{and } (d, e) \in V\ \big\}.
\end{aligned}$$

Note that it follows from the above that all diamonds of transitions in $A(T)$ are in fact independence diamonds.

Proof (of Proposition 12)
The idea is to show that every domino bisimulation for T_1 and T_2 gives rise to a hhp-bisimulation for $A(T_1)$ and $A(T_2)$, and *vice versa*. First observe, that a run of $A(T_i)$ for $i \in \{1, 2\}$ is uniquely determined by the numbers x and y of the occurrences of the x_j-events, and the y_k-events, respectively, and the set of its "d"- and "$\overline{d}$"-events, which is of size at most two. In other

words, we can identify runs of $A(T_i)$ with triples (F_i, x, y), where $F_i \subseteq E_{A(T_i)}$ contains at most two "d"- and "$\overline{d}$"-events, and $x, y \in \mathbb{N}$, and elements of a hhp-bisimulation relating $A(T_1)$ and $A(T_2)$ with quadruples (F_1, F_2, x, y). The following claim immediately implies Proposition 12.

Claim 13

1. Let $B \subseteq \mathrm{Conf}\big(A(T_1), A(T_2)\big)$ be an hhp-bisimulation relating $A(T_1)$ and $A(T_2)$. Then the set $\left\{\, (d, e, x, y) \;:\; \big(\{\overline{d}_{xy}\}, \{\overline{e}_{xy}\}, x, y\big) \in B \,\right\}$ is a domino bisimulation for T_1 and T_2.

2. Let $B \subseteq \mathrm{Conf}(T_1, T_2)$ be a domino bisimulation relating T_1 and T_2. Then the set $\left\{\, \big(\{\overline{d}_{xy}\}, \{\overline{e}_{xy}\}, x, y\big) \;:\; (d, e, x, y) \in B \,\right\}$ can be extended to an hhp-bisimulation for $A(T_1)$ and $A(T_2)$.

Proof: Let $B \subseteq \mathrm{Conf}\big(A(T_1), A(T_2)\big)$ be an hhp-bisimulation. We need to check that the set $\left\{\, (d, d', x, y) \;:\; \big(\{\overline{d}_{xy}\}, \{\overline{d'}_{xy}\}, x, y\big) \in B \,\right\}$ satisfies conditions 1.–3. of Definition 5. Conditions 1. and 2. follow easily by first applying condition 1. and then repeatedly applying condition 2., or condition 3. of Definition 2.

We argue that condition 3. of Definition 5 is satisfied as well. Without loss of generality assume that $i = 1$; the other case is symmetric. Let $\big(\{\overline{d}_{xy}\}, \{\overline{e}_{xy}\}, x, y\big) \in B$, and configurations (d, x, y) and (d', x', y') of T_1 be compatible. Let $x' = x + 1$, and $y' = y$; the other three cases are analogous.

From the construction of $A(T_1)$ and $A(T_2)$, and by condition 2. of Definition 2 it follows that $\big(\{\overline{d}_{xy}, d_{xy}\}, \{\overline{e}_{xy}, e_{xy}\}, x, y\big) \in B$. Hence by condition 5. of Definition 2 we have that $\big(\{d_{xy}\}, \{e_{xy}\}, x, y\big) \in B$, and by condition 2., that $\big(\{d_{xy}\}, \{e_{xy}\}, x+1, y\big) = \big(\{d_{xy}\}, \{e_{xy}\}, x', y'\big) \in B$.

If $x > 0$ then $(d_{xy}, d'_{x'y'}) \in H_{T_1}$, and if $x = 0$ then $(d_{xy}, d'_{x'y'}) \in H^0_{T_1}$. Hence by the construction of $A(T_1)$ it follows that $\big(\{d_{xy}, \overline{d'}_{x'y'}\}, x', y'\big)$ is a run of $A(T_1)$. Then condition 2. of Definition 2 implies that there is an $e' \in D_{T_2}$, such that $\big(\{e_{xy}, \overline{e'}_{x'y'}\}, x', y'\big)$ is a run of $A(T_2)$, and $\lambda_1(d') = \lambda_2(e')$, and $\big(\{d_{xy}, \overline{d'}_{x'y'}\}, \{e_{xy}, \overline{e'}_{x'y'}\}, x', y'\big) \in B$. Therefore, by condition 5. of Definition 2 we have that $\big(\{\overline{d'}_{x'y'}\}, \{\overline{e'}_{x'y'}\}, x', y'\big) \in B$, and by construction of $A(T_2)$ it follows that $(d', e') \in H_{T_2}$ if $x > 0$, and $(d', e') \in H^0_{T_2}$ if $x = 0$, *i.e.*, configurations (e, x, y) and (e', x', y') of T_2 are compatible. This concludes the proof of clause 1. of the claim.

Now we sketch a proof of clause 2. of the claim. Note that the maximal runs of $A(T_i)$ for $i \in \{1, 2\}$ are of one of the following forms:

1. $\big(\{d_{xy}, \overline{d}_{xy}\}, x, y\big)$,

2. $\big(\{d_{(x-1)y}, \overline{d'}_{xy}\}, x, y\big)$, such that $(d, d') \in H^0_{T_i}$ if $x = 1$, and $(d, d') \in H_{T_i}$ if $x > 1$.

3. $(\{d_{x(y-1)}, \overline{d'}_{xy}\}, x, y)$, such that $(d, d') \in V^0_{T_i}$ if $y = 1$, and $(d, d') \in V_{T_i}$ if $y > 1$.

Suppose $B \subseteq D_1 \times D_2 \times \mathbb{N} \times \mathbb{N}$ is a domino bisimulation relating T_1 and T_2. To get an hhp-bisimulation $\overline{B} \subseteq \mathrm{Run}\,(A(T_1)) \times \mathrm{Run}\,(A(T_2))$ we do the following:

1. add to $\overline{B}$ all the tuples $(\{d_{xy}, \overline{d}_{xy}\}, \{e_{xy}, \overline{e}_{xy}\}, x, y)$, such that $(d, e, x, y) \in B$,
2. add to $\overline{B}$ all the tuples $(\{d_{(x-1)y}, \overline{d'}_{xy}\}, \{e_{(x-1)y}, \overline{e'}_{xy}\}, x, y)$, such that $(d, e, x-1, y) \in B$ and $(d', e', x, y) \in B$, and moreover, $(d, x-1, y)$ and (d', x, y) are compatible as configurations of T_1, and $(e, x-1, y)$ and (e', x, y) are compatible as configurations of T_2,
3. add to $\overline{B}$ all the tuples $(\{d_{x(y-1)}, \overline{d'}_{xy}\}, \{e_{x(y-1)}, \overline{e'}_{xy}\}, x, y)$, such that $(d, e, x, y-1) \in B$ and $(d', e', x, y) \in B$, and moreover, $(d, x, y-1)$ and (d', x, y) are compatible as configurations of T_1, and $(e, x-1, y)$ and (e', x, y) are compatible as configurations of T_2,
4. close $\overline{B}$ "downwards", *i.e.*, complete it so that condition 5. of the Definition 2 is satisfied.

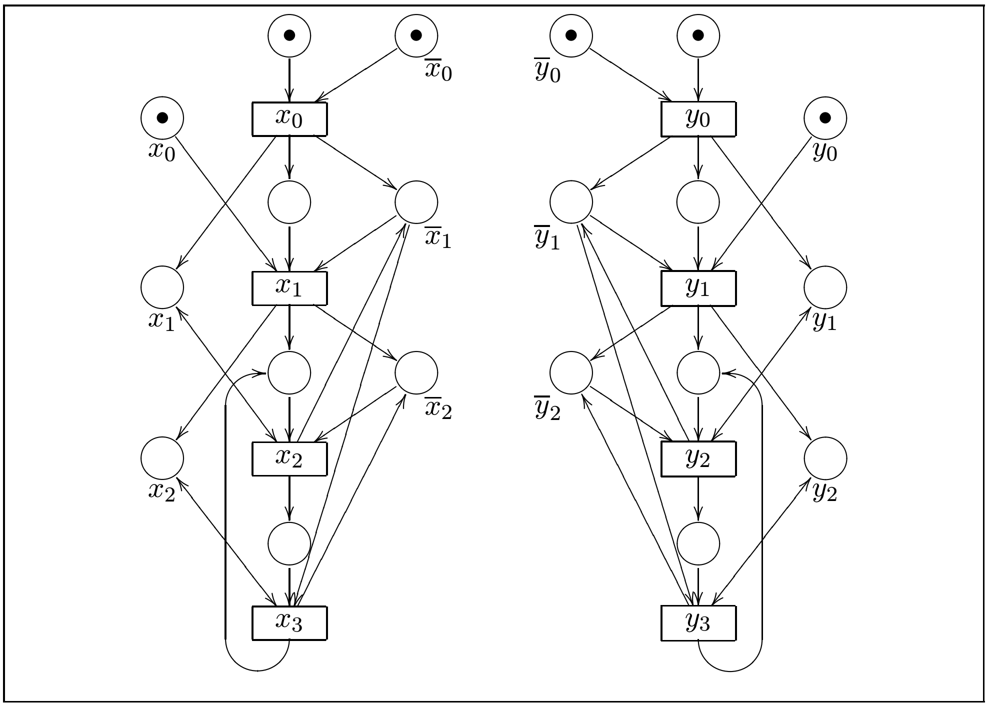
We leave it as an exercise to the Reader to check that $\overline{B}$ is indeed an hhp-bisimulation. [Claim 13] ■ [Proposition 12] ■

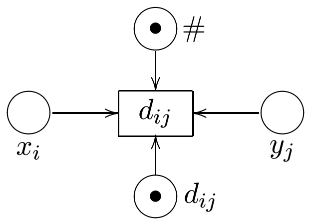
This concludes the proof of Theorem 3.

4 A 1-safe Petri net

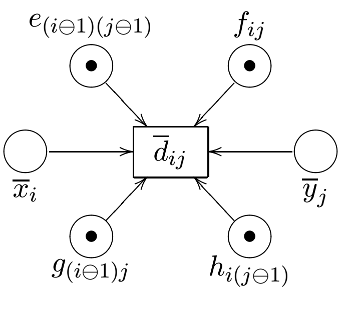
An attentive Reader might have noticed, that the asynchronous transitions system $A(T)$ as described in the previous section, and sketched in Figure 1, is not coherent, while all asynchronous transition systems derived from (1-safe) Petri nets are [WN95, NW96]. It turns out, however, that $A(T)$ is not far from being coherent: it suffices to close all the diamonds with events d_{ij}, and x_i in positions $(i, j \oplus 1)$, and with events d_{ij}, and y_j in positions $(i \oplus 1, j)$, for $i, j \in \{0, \ldots, 3\}$; note that runs ending at the top of these diamonds are maximal runs. This completion of the transition structure of $A(T)$ does not affect the arguments used to establish Claim 13, and hence Theorem 3, but since it would obscure the picture in Figure 1(b), we have decided not to draw it there.

Our aim now is to present a finite 1-safe Petri net $N(T)$ whose derived asynchronous transition system is isomorphic to the completion of $A(T)$ mentioned above. The set of transitions of the net $N(T)$ is just the set of events of the asynchronous transitions system $A(T)$. The structure of the net $N(T)$ is shown in Figure 2. The operation $k \ominus 1$ for $k \in \{0, 1, 2\}$ is defined as follows:





For all $d \in D$, and $i, j \in \{0, 1, 2\}$, such that $(i, j) = (0, 0)$ implies $d \in D^{\mathrm{ori}}$.



For all $d, \ldots, h \in D$, and $i, j \in \{0, 1, 2\}$, such that $(i, j) = (0, 0)$ implies $d \in D^{\mathrm{ori}}$, and:

- $f \neq d$,
- $(g, d) \notin H^0$, if $i \ominus 1 = 0$, and $(g, d) \notin H$, if $i \ominus 1 > 0$,
- $(h, d) \notin V^0$, if $j \ominus 1 = 0$, and $(h, d) \notin V$, if $j \ominus 1 > 0$.

Figure 2: The structure of the 1-safe Petri net $N(T)$.

$$k \ominus 1 = \begin{cases} \text{undefined} & \text{for } k = 0, \\ 0 \text{ or } 2 & \text{for } k = 1, \\ 1 & \text{for } k = 2. \end{cases}$$

For example a transition $\overline{d}_{12}$ has arrows from places e_{01} and e_{21} for all $e \in D$, because possible values for $(1 \ominus 1, 2 \ominus 1)$ are $(0, 1)$ and $(2, 1)$.

We leave it as an exercise for the Reader to check that there is indeed a 1-1 correspondence between firing sequences of the Petri net $N(T)$, and runs of (a completion—mentioned above—of) the asynchronous transition system $A(T)$, and this correspondence also respects the independence structure of the Petri net and the asynchronous transition system. Let us only give a few remarks on the design of $N(T)$:

- The place # serves as a resource shared by all "d_{ij}"-events for all $i, j \in \{0, 1, 2\}$, and hence guarantees "mutual exclusion" of these events. Note that for example if the place # was not connected to events d_{11} and e_{21} for some $d \neq e$, then they could both occur in a configuration of $N(T)$.

- For every configuration C of $N(T)$ we have that $n \in \mathbb{N}$ is the number of occurrences of x_i-events, and $m \in \mathbb{N}$ is the number of occurrences of y_j-events in C for $i, j \in \{0, 1, 2, 3\}$, if and only if there is a token in place $\overline{x}_{\hat{n}}$, and there is a token in place $\overline{y}_{\hat{m}}$ in the marking of $N(T)$ corresponding to C, or an event $\overline{d}_{nm}$ occurs in C.

- Arrows from places x_i for $i \in \{0, 1, 2\}$ to events x_{i+1}, and arrows from places y_j for $j \in \{0, 1, 2\}$ to events y_{i+1}, together with the arrows from places x_i and y_j to events d_{ij}, guarantee that the $N(T)$ counterparts of configurations $\big(\{d_{xy}\}, x+1, y+1\big)$ of $A(T)$ are maximal.

References

[Bed88] Marek A. Bednarczyk. *Categories of Asynchronous Systems.* PhD thesis, University of Sussex, 1988.

[Bed91] Marek A. Bednarczyk. Hereditary history preserving bisimulations or what is the power of the future perfect in program logics. Technical report, Polish Academy of Sciences, Gdańsk, April 1991. Available electronically at `http://www.ipipan.gda.pl/~marek`.

[FH99] Sibylle Fröschle and Thomas Hildebrandt. On plain and hereditary history preserving bisimulation. To apper in *Proceedings of Mathematical Foundations of Computer Science 1999, 24th International Symposium, MFCS'99*, Szklarska Poręba, Poland, September 1999.

[Frö99] Sibylle Fröschle. Decidability of plain and hereditary history-preserving bisimilarity for BPP. To be presented at 6th International Workshop on Expressiveness in Concurrency, EXPRESS'99, Eindhoven, The Netherlands, August 1999.

[JM96] Lalita Jategaonkar and Albert R. Meyer. Deciding true concurrency equivalences on safe, finite nets. *Theoretical Computer Science*, 154:107–143, 1996.

[JNW96] André Joyal, Mogens Nielsen, and Glynn Winskel. Bisimulation from open maps. *Information and Computation*, 127(2):164–185, 1996. A preliminary version appeared in *Proceedings of Eighth Annual IEEE Symposium on Logic in Computer Science*, pages 418–427, Montreal, Canada, June 1993. IEEE Computer Society Press.

[MT98] P. Madhusudan and P. S. Thiagarajan. Controllers for discrete event systems via morphisms. In Davide Sangiorgi and Robert de Simone, editors, *CONCUR'98, Concurrency Theory, 9th International Conference, Proceedings*, volume 1466 of *LNCS*, pages 18–33, Nice, France, September 1998. Springer.

[NC95] M. Nielsen and C. Clausen. Games and logics for a noninterleaving bisimulation. *Nordic Journal of Computing*, 2(2):221–249, 1995.

[NW96] Mogens Nielsen and Glynn Winskel. Petri nets and bisimulation. *Theoretical Computer Science*, 153(1–2):211–244, 1996.

[RT88] A. Rabinovich and B. Trakhtenbrot. Behaviour structures and nets of processes. *Fundamenta Informaticae*, 11:357–404, 1988.

[Shi85] M. W. Shields. Concurrent machines. *Computer Journal*, 28:449–465, 1985.

[vGG89] Rob van Glabbeek and Ursula Goltz. Equivalence notions for concurrent systems and refinement of actions (Extended abstract). In A. Kreczmar and G. Mirkowska, editors, *Mathematical Foundations of Computer Science 1989*, volume 379 of *LNCS*, pages 237–248, Porąbka-Kozubnik, Poland, August/September 1989. Springer-Verlag.

[Vog91] Walter Vogler. Deciding history preserving bisimilarity. In Javier Leach Albert, Burkhard Monien, and Mario Rodríguez-Artalejo, editors, *Automata, Languages and Programming, 18th International Colloquium, ICALP'91*, volume 510 of *LNCS*, pages 493–505, Madrid, Spain, 8–12 July 1991. Springer-Verlag.

[WN95] Glynn Winskel and Mogens Nielsen. Models for concurrency. In S. Abramsky, Dov M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 4, Semantic Modelling, pages 1–148. Oxford University Press, 1995.

Recent BRICS Report Series Publications

RS-99-19 **Marcin Jurdziński and Mogens Nielsen.** ***Hereditary History Preserving Bisimilarity is Undecidable*****. June 1999. 18 pp.**

RS-99-18 **M. Oliver Möller and Harald Rueß.** ***Solving Bit-Vector Equations of Fixed and Non-Fixed Size*****. June 1999. 18 pp. Revised version of an article appearing under the title** ***Solving Bit-Vector Equations*** **in Gopalakrishnan and Windley, editors,** ***Formal Methods in Computer-Aided Design: Second International Conference*****, FMCAD '98 Proceedings, LNCS 1522, 1998, pages 36–48.**

RS-99-17 **Andrzej Filinski.** ***A Semantic Account of Type-Directed Partial Evaluation*****. June 1999. To appear in Nadathur, editor,** ***International Conference on Principles and Practice of Declarative Programming*****, PPDP99 '99 Proceedings, LNCS, 1999.**

RS-99-16 **Rune B. Lyngsø and Christian N. S. Pedersen.** ***Protein Folding in the 2D HP Model*****. June 1999. 15 pp.**

RS-99-15 **Rune B. Lyngsø, Michael Zuker, and Christian N. S. Pedersen.** ***An Improved Algorithm for RNA Secondary Structure Prediction*****. May 1999. 24 pp. An alloy of two articles appearing in Istrail, Pevzner and Waterman, editors,** ***Third Annual International Conference on Computational Molecular Biology*****, RECOMB 99 Proceedings, 1999, pages 260–267, and** ***Bioinformatics*****, 15, 1999.**

RS-99-14 **Marcelo P. Fiore, Gian Luca Cattani, and Glynn Winskel.** ***Weak Bisimulation and Open Maps*****. May 1999. To appear in Longo, editor,** ***Fourteenth Annual IEEE Symposium on Logic in Computer Science*****, LICS '99 Proceedings, 1999.**

RS-99-13 **Rasmus Pagh.** ***Hash and Displace: Efficient Evaluation of Minimal Perfect Hash Functions*****. May 1999. 11 pp. A short version to appear in** ***Algorithms and Data Structures: 6th International Workshop*****, WADS '99 Proceedings, LNCS, 1999.**

RS-99-12 **Gerth Stølting Brodal, Rune B. Lyngsø, Christian N. S. Pedersen, and Jens Stoye.** ***Finding Maximal Pairs with Bounded Gap*****. April 1999. 31 pp. To appear in** ***Combinatorial Pattern Matching: 10th Annual Symposium*****, CPM '99 Proceedings, LNCS, 1999.**