Undecidability of Domino Games and Hhp-Bisimilarity[1]

Marcin Jurdziński[†2] and Mogens Nielsen[‡] and Jiří Srba[‡]

[†] *Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, USA*

[‡] **BRICS**[3], *Department of Computer Science, University of Aarhus, Denmark*

E-mail: mju@eecs.berkeley.edu; mn@brics.dk; srba@brics.dk

History preserving bisimilarity (hp-bisimilarity) and hereditary history preserving bisimilarity (hhp-bisimilarity) are behavioural equivalences taking into account causal relationships between events of concurrent systems. Their prominent feature is that they are preserved under action refinement, an operation important for the top-down design of concurrent systems. It is shown that, in contrast to hp-bisimilarity, checking hhp-bisimilarity for finite labelled asynchronous transition systems is undecidable, by a reduction from the halting problem of 2-counter machines. To make the proof more transparent a novel intermediate problem of checking domino bisimilarity for origin constrained tiling systems is introduced and shown undecidable. It is also shown that the unlabelled domino bisimilarity problem is undecidable, which implies undecidability of hhp-bisimilarity for unlabelled finite asynchronous systems. Moreover, it is argued that the undecidability of hhp-bisimilarity holds for finite elementary net systems.

1. INTRODUCTION

The notion of behavioural equivalence which has attracted most attention in concurrency theory is bisimilarity, originally introduced by Park [25] and Milner [19]; concurrent programs are considered to have the same meaning if they are bisimilar. The prominent role of bisimilarity is due to many pleasant properties it enjoys; we mention a few of them here.

A process of checking whether two transition systems are bisimilar can be seen as a two player game which is in fact an Ehrenfeucht-Fraïssé type of game for modal logic. More precisely, there is a winning strategy for a player who wants to show

[1] A revised and extended version of the STACS 2000 extended abstract [16].

[2] The work reported here has been done while the author was a graduate student at BRICS, Department of Computer Science, University of Aarhus, Denmark.

[3] **B**asic **R**esearch in **C**omputer **S**cience, Centre of the Danish National Research Foundation.

that the systems are bisimilar if and only if the systems cannot be distinguished by the formulas of the logic; the result due to Hennessy and Milner [12].

Another notable property of bisimilarity is its computational feasibility; see for example the overview note [21]. Let us illustrate this on the examples of finite transition systems and a class of infinite-state transition systems generated by context free grammars. For finite transition systems there are very efficient polynomial time algorithms for checking bisimilarity [17, 24], in sharp contrast to PSPACE-completeness of the classical language equivalence. For transition systems generated by context free grammars, while language equivalence is undecidable, bisimilarity is decidable [4], and if the grammar has no redundant nonterminals, even in polynomial time [13]. Furthermore, as the results of Groote and Hüttel [11] indicate, bisimilarity has a very rare status of being a decidable equivalence for context free grammars: all the other equivalences in the linear-branching time hierarchy [8] are undecidable. The algorithmic tractability makes bisimilarity especially attractive for automatic verification of concurrent systems.

The essence of bisimilarity, quoting Hennessy and Milner [12], "is that the behaviour of a program is determined by how it communicates with an observer." Therefore, the notion of what can be observed of a behaviour of a system affects the notion of bisimilarity. An abstract definition of bisimilarity for arbitrary categories of models due to Joyal et al. [15] formalizes this idea. Given a category of models where objects are behaviours and morphisms correspond to extension of behaviours, and given a subcategory of observable behaviours, the abstract definition yields a notion of bisimilarity for all behaviours with respect to observable behaviours. For example, for rooted labelled transition systems, taking synchronization trees [19] into which they unfold as their behaviours, and sequences of actions as the observable behaviours, we recover the standard strong bisimilarity of Park and Milner [15].

In order to model concurrency more faithfully several models have been introduced (see [30] for a survey) that make explicit the distinction between events that can occur concurrently, and those that are causally related. Then a natural choice is to replace sequences, i.e., linear orders as the observable behaviours, by partial orders of occurrences of events with causality as the ordering relation. For example, taking unfoldings of labelled asynchronous transition systems into event structures as the behaviours, and labelled partial orders as the observations, Joyal et al. [15] obtained from their abstract definition the hereditary history preserving bisimilarity (hhp-bisimilarity), independently introduced and studied by Bednarczyk [1].

A similar notion of bisimilarity has been studied before, namely history preserving bisimilarity (hp-bisimilarity), introduced by Rabinovich and Trakhtenbrot [26] and van Glabbeek and Goltz [9]. For the relationship between hp- and hhp-bisimilarity see for example [1, 15, 7].

One of the important motivations to study partial order based equivalences was the discovery that hp-bisimilarity has a rare status of being preserved under action refinement [9], an operation important for the top-down design of concurrent systems. Bednarczyk [1] has extended this result to hhp-bisimilarity.

There is a natural logical characterization of hhp-bisimilarity checking games as shown by Nielsen and Clausen [22]: they are characteristic games for an extension of modal logic with backwards modalities, interpreted over event structures.

Hp-bisimilarity has been shown to be decidable for 1-safe Petri nets by Vogler [29], and to be DEXP-complete by Jategaonkar, and Meyer [14]; let us just mention here that 1-safe Petri nets can be regarded as a proper subclass of finite asynchronous transition systems (see [30] for details), and that decidability of hp-bisimilarity can be easily extended to all finite asynchronous transition systems using the methods of [14].

Hhp-bisimilarity appears to be only a slight strengthening of hp-bisimilarity [15] and hence many attempts have been made to extend the above mentioned algorithms to the case of hhp-bisimilarity. However, decidability of hhp-bisimilarity has remained open, despite several attempts over the years [22, 23, 3, 7]. Fröschle and Hildebrandt [7] have discovered an infinite hierarchy of bisimilarity notions refining hp-bisimilarity, and coarser than hhp-bisimilarity, such that hhp-bisimilarity is the intersection of all the bisimilarities in the hierarchy. They have shown all these bisimilarities to be decidable for 1-safe Petri nets. Fröschle [6] has proved hhp-bisimilarity to be decidable for BPP-processes, a class of infinite state systems.

In this paper we resolve the question of decidability of hhp-bisimilarity for all finite state systems by showing it to be undecidable for finite, both labelled and unlabelled, asynchronous transition systems and finite elementary net systems. In order to make the proof more transparent we first introduce an intermediate problem of domino bisimilarity and show its undecidability by a direct reduction from the undecidable halting problem for 2-counter machines [20]. The undecidability of the novel problem of checking domino bisimilarity seems to be interesting in its own right and does not follow from somewhat related results for domino snakes [5] and domino games [10], nor from the undecidability of the classical tiling problems [2].

2. HEREDITARY HISTORY PRESERVING BISIMILARITY

In this section we define hereditary history preserving bisimulation for asynchronous transition systems and we introduce the algorithmic problem of checking hereditary history preserving bisimilarity. We also mention the equivalent, and sometimes technically more convenient, problem of solving hereditary history preserving bisimilarity checking games.

Definition 2.1. (Labelled/unlabelled asynchronous transition system).
A *labelled asynchronous transition system* is a tuple $A = (S, s^{\mathrm{ini}}, E, \rightarrow, L, \lambda, I)$, where S is its set of *states*, $s^{\mathrm{ini}} \in S$ is the *initial state*, E is the set of *events*, $\rightarrow \subseteq S \times E \times S$ is the set of *transitions*, L is the set of labels, and $\lambda : E \rightarrow L$ is the *labelling function*, and $I \subseteq E^2$ is the *independence relation* which is irreflexive and symmetric. We often write $s \stackrel{e}{\rightarrow} s'$, instead of $(s, e, s') \in \rightarrow$. Moreover, the following conditions have to be satisfied:

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

An asynchronous transition system is *coherent* if it satisfies the following condition:

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

An asynchronous transition system is *prime* if it is acyclic and satisfies the following condition:

4. if $s \xrightarrow{e} t$ and $s' \xrightarrow{e'} t$ then $(e, e') \in I$.

We say that an asynchronous transition system is *unlabelled* if the set of labels L is a singleton set.

Winskel and Nielsen [30, 23] 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 independence relation is meant to model concurrency: independent events can occur concurrently, while those that are not independent are causally related or in conflict.

Let $A = (S, s^{\text{ini}}, E, \rightarrow, L, \lambda, I)$ be a labelled asynchronous transition system. A sequence of events $\overline{e} = \langle e_1, e_2, \ldots, e_n \rangle \in E^*$ is a *run* of A if there are states $s_1, s_2, \ldots, s_{n+1} \in S$, such that $s_1 = s^{\text{ini}}$, and for all $i \in \{1, 2, \ldots, n\}$, we have $s_i \xrightarrow{e_i} s_{i+1}$. We write $\texttt{Runs}(A)$ to denote the set of runs of A. We extend the labelling function λ to runs in the standard way.

Let $\overline{e} = \langle e_1, e_2, \ldots, e_n \rangle \in \texttt{Runs}(A)$. We say that the k-th event, $1 \leq k < n$, is *swappable* in $\overline{e}$ if $(e_k, e_{k+1}) \in I$. We define $\text{Swap}(\overline{e})$ to be the set of numbers of swappable events in $\overline{e}$. We write $\overline{e} \otimes k$ to denote the result of *swapping* the k-th event of $\overline{e}$ with the $(k+1)$-st, i.e., the sequence $\langle e_1, \ldots, e_{k-1}, e_{k+1}, e_k, \ldots, e_n \rangle$. Note that if $k \in \text{Swap}(\overline{e})$ then $\overline{e} \otimes k \in \texttt{Runs}(A)$; it follows from condition 2. of the definition of an asynchronous transition system.

A run of a transition system models a finite *sequential* behaviour of a system: a sequence of occurrences of events. In order to model *concurrent* behaviours of a system we define an equivalence relation on the set of runs of an asynchronous transition system. We define the equivalence relation $\cong_A$ on $\texttt{Runs}(A)$ to be the reflexive, symmetric, and transitive closure of

$$\left\{ \; (\overline{e}, \overline{e} \otimes k) \; : \; \overline{e} \in \texttt{Runs}(A) \text{ and } k \in \text{Swap}(\overline{e}) \; \right\}.$$

In other words, we have that $\overline{e_1} \cong_A \overline{e_2}$, for $\overline{e_1}, \overline{e_2} \in \texttt{Runs}(A)$, if and only if $\overline{e_2}$ can be obtained from $\overline{e_1}$ by a finite number of swaps of swappable events.

We define an unfolding operation on asynchronous transition systems into prime asynchronous transition systems. The states of the unfolding of an asynchronous transition system A are meant to represent all concurrent behaviours of a system, just like the states of a synchronization tree represent all sequential behaviours of a system.

Definition 2.2. (Unfolding).
Let $A = (S, s^{\text{ini}}, E, \rightarrow, L, \lambda, I)$ be an asynchronous transition system. The unfolding $\texttt{Unf}(A)$ of A is an asynchronous transition system with the same set of events, the labelling function, and the independence relation as A. The set of states, the initial state, and the transition relation of $\texttt{Unf}(A)$ are defined as follows:

- the set of states $S_{\texttt{Unf}(A)}$ of $\texttt{Unf}(A)$ is defined to be $\texttt{Runs}(A)/_{\cong_A}$, i.e., the set of concurrent behaviours of A,
- the initial state $s^{\text{ini}}_{\texttt{Unf}(A)}$ of $\texttt{Unf}(A)$ is $[\varepsilon]_{\cong_A}$, i.e., the $\cong_A$-equivalence class of the empty run,
- the set of transitions $\rightarrow_{\texttt{Unf}(A)}$ of $\texttt{Unf}(A)$ consists of transitions of the form $\left([\overline{e}]_{\cong_A}, e, [\overline{e} \cdot e]_{\cong_A}\right)$, for all $\overline{e} \in E^*$, and $e \in E$, such that $\overline{e} \cdot e \in \texttt{Runs}(A)$.

The following proposition follows easily from the definition of $\mathtt{Unf}(A)$.

PROPOSITION 2.1. *If A is an asynchronous transition system then its unfolding $\mathtt{Unf}(A)$ is a prime asynchronous transition system.*

Let $\overline{e} = \langle e_1, e_2, \ldots, e_n \rangle \in \mathtt{Runs}(A)$. We say that the k-th event, $1 \leq k \leq n$, is *most recent* in $\overline{e}$ if and only if $(e_k, e_\ell) \in I$, for all ℓ, such that $k < \ell \leq n$. We define $\mathrm{MR}(\overline{e})$ to be the set of numbers of most recent events in $\overline{e}$. We write $\overline{e} \ominus k$ to denote the result of removing the k-th event from $\overline{e}$, i.e., the sequence $\langle e_1, \ldots, e_{k-1}, e_{k+1}, \ldots, e_n \rangle$. Note that if $k \in \mathrm{MR}(\overline{e})$ then $\overline{e} \ominus k \in \mathtt{Runs}(A)$; it follows from condition 2. of the definition of an asynchronous transition system.

DEFINITION 2.3. (Hereditary history preserving bisimulation).
Let $A_i = (S_i, s_i^{\mathrm{ini}}, E_i, \rightarrow_i, L, \lambda_i, I_i)$, for $i \in \{1, 2\}$, be labelled asynchronous transition systems. A relation $B \subseteq \mathtt{Runs}(A_1) \times \mathtt{Runs}(A_2)$ 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 $\lambda_1(\overline{e_1}) = \lambda_2(\overline{e_2})$, and:

2. for all $i \in \{1, 2\}$, and $e_i \in E_i$, if $\overline{e_i} \cdot e_i \in \mathtt{Runs}(A_i)$ then there exists $e_{3-i} \in E_{3-i}$, such that $\overline{e_{3-i}} \cdot e_{3-i} \in \mathtt{Runs}(A_{3-i})$, $\lambda_1(e_1) = \lambda_2(e_2)$, and $(\overline{e_1} \cdot e_1, \overline{e_2} \cdot e_2) \in B$,
3. $\mathrm{MR}(\overline{e_1}) = \mathrm{MR}(\overline{e_2})$,
4. if $k \in \mathrm{MR}(\overline{e_1}) = \mathrm{MR}(\overline{e_2})$ then $(\overline{e_1} \ominus k, \overline{e_2} \ominus k) \in B$.

Two asynchronous transition systems A_1, and A_2 are hereditary history preserving (hhp-) *bisimilar*, if there is an hhp-bisimulation relating them. For alternative and slightly varying definitions of hhp-bisimulation that all give rise to the same notion of hhp-bisimilarity see, e.g., the papers by Bednarczyk [1], Joyal et al. [15], Nielsen and Clausen [22], Nielsen and Winskel [23], or Fröschle and Hildebrandt [7].

The following proposition is straightforward since every asynchronous transition system A and its unfolding $\mathtt{Unf}(A)$ have the same set of runs and the same independence relation.

PROPOSITION 2.2. *Asynchronous transition systems A_1 and A_2 are hhp-bisimilar if and only if their unfoldings $\mathtt{Unf}(A_1)$ and $\mathtt{Unf}(A_2)$ are hhp-bisimilar.*

The main results of this paper are the following theorems proved in Section 4.

THEOREM 2.1 (Undecidability of hhp-bisimilarity).
Hhp-bisimilarity is undecidable for finite labelled asynchronous transition systems.

THEOREM 2.2 (Undecidability of unlabelled hhp-bisimilarity).
Hhp-bisimilarity is undecidable for finite unlabelled asynchronous transition systems.

The process of checking hhp-bisimilarity of asynchronous transition systems is conveniently viewed as a game played on runs of the systems by two players: *Spoiler* and *Duplicator*. Duplicator aims to prove the systems to be bisimilar while Spoiler tries to show the opposite [27, 28, 22].

DEFINITION 2.4. (Hhp-bisimilarity checking game).
Let $A_i = (S_i, s_i^{\mathrm{ini}}, E_i, \rightarrow_i, L, \lambda_i, I_i)$ for $i \in \{1,2\}$ be labelled asynchronous transition systems. *Configurations* of the *hhp-bisimilarity checking game* $\mathcal{B}_{\mathrm{hhp}}(A_1, A_2)$ are elements of the set $\mathtt{Runs}(A_1) \times \mathtt{Runs}(A_2)$. Game $\mathcal{B}_{\mathrm{hhp}}(A_1, A_2)$ is played by two players: *Spoiler* and *Duplicator*. The *initial configuration* is the pair of empty runs $(\varepsilon, \varepsilon)$. In each *move* the players change the current configuration $(\overline{e_1}, \overline{e_2})$ of $\mathcal{B}_{\mathrm{hhp}}(A_1, A_2)$ in one of the following ways chosen by Spoiler.

- Forward move:

  1. Spoiler chooses an $i \in \{1,2\}$ and an event $e_i \in E_i$, such that $\overline{e_i} \cdot e_i \in \mathtt{Runs}(A_i)$;

  2. Duplicator responds by choosing an event $e_{3-i} \in E_{3-i}$, such that $\overline{e_{3-i}} \cdot e_{3-i} \in \mathtt{Runs}(A_{3-i})$, and $\lambda_1(e_1) = \lambda_2(e_2)$;

  the pair $(\overline{e_1} \cdot e_1, \overline{e_2} \cdot e_2)$ becomes the current configuration.

- Backward move:

  1. Spoiler chooses an $i \in \{1,2\}$ and a $k \in \mathrm{MR}(\overline{e_i})$;

  2. Duplicator can only respond if $k \in \mathrm{MR}(\overline{e_{3-i}})$; otherwise Duplicator gets stuck;

  if $k \in \mathrm{MR}(\overline{e_1})$, and $k \in \mathrm{MR}(\overline{e_2})$ then $(\overline{e_1} \oslash k, \overline{e_2} \oslash k)$ becomes the current configuration.

A *play* of $\mathcal{B}_{\mathrm{hhp}}(A_1, A_2)$ if a *maximal* sequence of configurations formed by players making moves in the fashion described above. Duplicator is the winner in every infinite play; a finite play is lost by the player who is stuck. Note that Spoiler gets stuck only if both transition systems have no transitions going out from their initial states.

We skip the tedious details of formalizing notions of strategies and winning strategies for either of the players. The following standard fact is proved by arguing that an hhp-bisimulation is a good formalization of the notion of a winning strategy for Duplicator in an hhp-bisimilarity checking game [22].

PROPOSITION 2.3. *Asynchronous transition systems A_1 and A_2 are hhp-bisimilar if and only if there is a winning strategy for Duplicator in hhp-bisimilarity checking game $\mathcal{B}_{\mathrm{hhp}}(A_1, A_2)$.*

It is easy to see how an hhp-bisimulation $B \subseteq \mathtt{Runs}(A_1) \times \mathtt{Runs}(A_2)$ can serve as a winning strategy for Duplicator in $\mathcal{B}_{\mathrm{hhp}}(A_1, A_2)$. Intuitively, the hhp-bisimulation B contains all configurations of $\mathcal{B}_{\mathrm{hhp}}(A_1, A_2)$ which can become current configurations when Duplicator is following the strategy determined by B. The strategy is defined as follows. Let $(\overline{e_1}, \overline{e_2})$ be the current configuration of

$\mathcal{B}_{\mathrm{hhp}}(A_1, A_2)$. If Spoiler chooses event e_i of A_i in a forward move, then Duplicator responds by choosing an event e_{3-i} of A_{3-i}, such that $(\overline{e_1} \cdot e_1, \overline{e_2} \cdot e_2) \in B$. If Spoiler makes a backward move then response of Duplicator is unique. This strategy contains the initial configuration $(\varepsilon, \varepsilon)$ by condition 1. of the definition of an hhp-bisimulation, and it is well defined by conditions 2..–4..

It is not very hard to argue that bisimilarity checking games are determined.

PROPOSITION 2.4 (Determinacy). *Bisimilarity checking games are determined, i.e., in every game exactly one of the players has a winning strategy.*

It follows that if one of the players does not have a winning strategy in a bisimilarity checking game then the other player does.

3. DOMINO BISIMILARITY IS UNDECIDABLE

In this section we introduce a novel algorithmic problem of checking domino bisimilarity for labelled origin constrained tiling systems and an equivalent problem of solving domino bisimilarity checking games. The main results are the undecidability of the problem and its extension to the unlabelled case. These results serve crucially in Section 4 to establish the main result of the paper, i.e., the undecidability of hhp-bisimilarity for finite, both labelled and unlabelled, asynchronous transition systems and elementary net systems.

In Subsection 3.1 we define the algorithmic problem of checking domino bisimilarity. In Subsection 3.2 we recall 2-counter machines [20] and in Subsection 3.3 we prove the undecidability of labelled domino bisimilarity by a reduction from the halting problem for 2-counter machines. Finally, in Subsection 3.4 we extend the undecidability result to the unlabelled case.

It is worthwhile to compare our domino bisimilarity, or equivalently domino bisimilarity checking games, to domino snakes studied by Etzion-Petruschka et al. [5] and domino games of Grädel [10]. Despite apparent similarities of our domino bisimilarity checking games to the latter, our games seem to be of quite a different flavour and indeed the proofs of undecidability are very different.

3.1. Domino bisimilarity

DEFINITION 3.1. (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* $\lambda : D \to L$.

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 3.2. (Domino bisimulation).
Let $T_i = \big(D_i, D_i^{\text{ori}}, (H_i, H_i^0), (V_i, V_i^0), L_i, \lambda_i\big)$ 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 $(d_1, d_2, x, y) \in B$ implies that $\lambda_1(d_1) = \lambda_2(d_2)$, and the following conditions are satisfied for all $i \in \{1, 2\}$:

1. for all $d_i \in D_i^{\text{ori}}$, there is $d_{3-i} \in D_{3-i}^{\text{ori}}$, 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 is $d_{3-i} \in D_{3-i}$, 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 neighbours $(x', y') \in \mathbb{N} \times \mathbb{N}$ of (x, y), 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 $(d_1', d_2', x', y') \in B$.

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

The main result of this section is the following theorem proved in Subsection 3.3.

THEOREM 3.1 (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 Subsection 3.3 two origin constrained tiling systems T_1, and T_2, such that the following holds.

LEMMA 3.1. *Machine M does not halt if and only if there is a domino bisimulation relating T_1 and T_2.*

The process of checking domino bisimilarity of origin constrained tiling systems is conveniently viewed as a game played on an infinite grid by two players: Spoiler and Duplicator. As in the case of hhp-bisimilarity checking games Duplicator aims to prove the tiling systems to be bisimilar while Spoiler tries to show the opposite.

DEFINITION 3.3. (Origin constrained domino bisimilarity checking game).
Let T_1 and T_2 be origin constrained tiling systems. *Configurations* of the *origin constrained domino bisimilarity checking game* $\mathcal{B}_{\text{d}}(T_1, T_2)$ are elements of the set $D_1 \times D_2 \times \mathbb{N} \times \mathbb{N}$. Game $\mathcal{B}_{\text{d}}(T_1, T_2)$ is played by two players *Spoiler* and *Duplicator*.

- First the players fix an *initial configuration*:

  1. Spoiler chooses an $i \in \{1, 2\}$, and a configuration (d_i, x, y) of T_i,

  2. Duplicator responds by choosing a domino $d_{3-i} \in D_{3-i}$, such that (d_{3-i}, x, y) is a configuration of T_{3-i}, and $\lambda_1(d_1) = \lambda_2(d_2)$;

if both players were able to make their choices then the tuple (d_1, d_2, x, y) becomes the *current configuration* of $\mathcal{B}_{\mathrm{d}}(T_1, T_2)$.

• In each *move* of the domino bisimilarity checking game the players change the current configuration (d_1, d_2, x, y):

1. Spoiler chooses an $i \in \{1, 2\}$, and a configuration (d_i', x', y') of T_i compatible with configuration (d_i, x, y),

2. Duplicator responds by choosing a domino $d_{3-i}' \in D_{3-i}$ so that (d_{3-i}', x', y') is a configuration of T_{3-i}, and $\lambda_1(d_1) = \lambda_2(d_2)$, and configurations (d_{3-i}, x, y) and (d_{3-i}', x', y') of T_{3-i} are compatible;

if both players were able to make their choices then the tuple (d_1', d_2', x', y') becomes the *current configuration* of $\mathcal{B}_{\mathrm{d}}(T_1, T_2)$.

A *play* of $\mathcal{B}_{\mathrm{d}}(T_1, T_2)$ is a *maximal* sequence of configurations formed by players making moves in the fashion described above. Duplicator is the winner in every infinite play; a finite play is lost by the player who is stuck.

We avoid tedious details of formalizing notions of strategies and winning strategies for either of the players. The following simple fact is proved by arguing that a domino bisimulation is a good formalization of a winning strategy for Duplicator in a domino bisimilarity checking game.

PROPOSITION 3.1. *Origin constrained tiling systems T_1 and T_2 are domino bisimilar if and only if Duplicator has a winning strategy in the domino bisimilarity game $\mathcal{B}_{\mathrm{d}}(T_1, T_2)$.*

As in the case of hhp-bisimilarity checking games it is easy to argue that domino bisimilarity checking games are determined; in other words Proposition 2.4 holds also for domino bisimilarity checking games.

3.2. Counter machines

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

- ℓ: `c`$_i$:= `c`$_i$ + 1; goto m
- ℓ: if `c`$_i$ = 0 then `c`$_i$:= `c`$_i$ + 1; goto m
  else `c`$_i$:= `c`$_i$ - 1; goto n
- `halt`:

where $i = 1, 2$; $\ell, m, n \in L$, and $\{\texttt{start}, \texttt{halt}\} \subseteq L$. 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 $\texttt{c}_1$ and $\texttt{c}_2$, respectively; we denote the set of configurations of M by $\texttt{Confs}(M)$. The semantics of 2-counter machines is standard: let $\vdash_M \subseteq \texttt{Confs}(M) \times \texttt{Confs}(M)$ 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 $(\mathtt{Confs}(M), \vdash_M)$, 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 $(\mathtt{Confs}(M), \vdash_M)$ 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 is formalized in the following proposition.

PROPOSITION 3.2. *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 the relation $\sim_M \subseteq \mathtt{Confs}(M) \times \mathtt{Confs}(M)$ is the symmetric and transitive closure of $\vdash_M$.*

3.3. The reduction

In this subsection we give a proof of Theorem 3.1 by proving Lemma 3.1. The idea is to design a tiling system which "simulates" the 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 $(\ell, x, y) \in \mathtt{Confs}(M)$ 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: `c1 := c1 + 1; goto` m
  - m: `if c1 = 0 then c1 := c1 + 1; goto` n `else c1 := c1 - 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 c1 = 0 then c1 := c1 + 1; goto` m `else c1 := c1 - 1; goto` n

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 $c, c' \in \mathtt{Confs}(M)$ 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 configurations of T_M. The following proposition is then straightforward.

PROPOSITION 3.3. *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 Lemma 3.1. 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 $D_1^{\mathrm{ori}} = D_2^{\mathrm{ori}} \cup (L \times \{3\})$, 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 Lemma 3.1, and hence conclude the proof of Theorem 3.1, it suffices to establish the following two lemmas.

LEMMA 3.2. *If machine M halts on input $(0,0)$ then origin constrained tiling systems T_1 and T_2 are not domino bisimilar.*

Proof. By Proposition 3.1 it suffices to show that if machine M halts on input $(0,0)$ then Spoiler has a winning strategy in the game $\mathcal{B}_{\mathrm{d}}(T_1, T_2)$. Spoiler starts by choosing the configuration $\big((\mathtt{start}, 3), 0, 0\big)$ of T_1. Duplicator has to respond with domino $(\mathtt{start}, 2)$ of T_2 since $(\mathtt{start}, 1) \notin D_2^{\mathrm{ori}}$. Then Spoiler "simulates" the finite computation of M on input $(0,0)$ in the following way. If $\big((\ell, 3), (\ell, 2), x, y\big)$ is the current configuration of the game then Spoiler chooses the configuration $\big((\ell', 3), x', y'\big)$ of T_1, such that $\big(\ell, x, y\big) \vdash_M (\ell', x', y')$. This move is allowed thanks to Proposition 3.3. Then Duplicator can only respond with domino $(\ell', 2)$ of T_2, and $\big((\ell', 3), (\ell', 2), x', y'\big)$ becomes the current configuration of the game. In the last step of the simulation Spoiler chooses a configuration $\big((\mathtt{halt}, 3), x', y'\big)$ for some $x', y' \in \mathbb{N}$ which makes Duplicator stuck because $(\mathtt{halt}, 2) \notin D_2$. ■

LEMMA 3.3. *If machine M does not halt on input $(0,0)$ then origin constrained tiling systems T_1 and T_2 are domino bisimilar.*

Proof. By Proposition 3.1 it suffices to show that if machine M does not halt on input $(0,0)$ then Duplicator has a winning strategy in the game $\mathcal{B}_{\mathrm{d}}(T_1,T_2)$. We claim that the following is a winning strategy for Duplicator.

If in the first step Spoiler chooses a configuration $\big((\ell,j),x,y\big)$ of T_1 or T_2 for $j\in\{1,2\}$, then Duplicator responds with the domino (ℓ,j) of the other tiling system. It is obvious that then Duplicator can respond to all moves of Spoiler because both players play on identical pruned copies of T_M.

If instead Duplicator chooses a configuration $\big((\ell,3),x,y\big)$ of T_1 in the first step then Duplicator responds with:

- domino $(\ell,1)$ of T_2 if $(\ell,x,y)\sim_M(\mathtt{halt},x',y')$ or $(\ell,x,y)=(\mathtt{halt},x',y')$ for some $x',y'\in\mathbb{N}$, and
- domino $(\ell,2)$ of T_2 if $(\ell,x,y)\not\sim_M(\mathtt{halt},x',y')$ for all $x',y'\in\mathbb{N}$.

In the first case the only way Spoiler can make Duplicator stuck is to be able to choose configuration $\big((\mathtt{start},3),0,0\big)$ of T_1 since the only difference between copy 3 of T_M in T_1 and copy 1 of T_M in T_2 is that the latter does not have the triple $(\mathtt{start},0,0)$ as a configuration. Hence in order to prove that Duplicator has a winning strategy from the initial configuration $\big((\ell,3),(\ell,1),x,y\big)$, it suffices to show that $(\ell,x,y)\not\approx_M(\mathtt{start},0,0)$. Assume for the sake of contradiction that $(\ell,x,y)\approx_M(\mathtt{start},0,0)$. By Proposition 3.3 we then have $(\ell,x,y)\sim_M(\mathtt{start},0,0)$. This, by our assumption that $(\ell,x,y)\sim_M(\mathtt{halt},x',y')$ for some $x',y'\in\mathbb{N}$, implies that $(\mathtt{start},0,0)\sim_M(\mathtt{halt},x',y')$ for some $x',y'\in\mathbb{N}$. Then Proposition 3.2 implies that $(\mathtt{start},0,0)\vdash_M^+(\mathtt{halt},x',y')$, which contradicts the assumption of the lemma that machine M does not halt on input $(0,0)$.

The argument in the other case is similar. It suffices to show that $(\ell,x,y)\not\approx_M(\mathtt{halt},x',y')$ for all $x',y'\in\mathbb{N}$, because the only difference between copy 3 of T_M in T_1 and copy 2 of T_M in T_2 is that the latter has no triple $(\mathtt{halt},x',y')$ as a configuration. By applying Proposition 3.3 to our assumption that $(\ell,x,y)\not\sim_M(\mathtt{halt},x',y')$ for all $x',y'\in\mathbb{N}$, we immediately get that $(\ell,x,y)\not\approx_M(\mathtt{halt},x',y')$ for all $x',y'\in\mathbb{N}$. ■

3.4. Undecidability of unlabelled domino bisimilarity

In this section we argue that the problem of deciding domino bisimilarity is undecidable even for unlabelled origin constrained tiling systems or, equivalently, for origin constrained tiling systems with a singleton set of labels.

THEOREM 3.2 (Undecidability of unlabelled domino bisimilarity).
Unlabelled domino bisimilarity is undecidable for origin constraint tiling systems.

Proof. We show that the unlabelled origin constrained domino bisimilarity checking game is undecidable and then we use Proposition 3.1.

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. Without loss of generality we may assume that $L_1=L_2=\{1,\ldots,n\}$, for some $n\geq 1$. We give an effective construction of unlabelled origin constrained tiling systems $\overline{T}_i=\big(\overline{D}_i,D_i^{\mathrm{ori}},(\overline{H}_i,\overline{H}_i^0),(\overline{V}_i,\overline{V}_i^0)\big)$, for $i\in\{1,2\}$, such that the following holds.

LEMMA 3.4 (The reduction).
Duplicator has a winning strategy in the labelled domino bisimilarity checking game $\mathcal{B}_{\mathrm{d}}(T_1, T_2)$ *if and only if he has a winning strategy in the unlabelled game* $\mathcal{B}_{\mathrm{d}}(\overline{T}_1, \overline{T}_2)$.

Establishing this lemma will complete the proof of Theorem 3.2. Tiling systems $\overline{T}_i$, for $i \in \{1, 2\}$, are defined as follows:

- $\overline{D}_i = D_i \cup \{\mathtt{c}_d^1, \ldots, \mathtt{c}_d^{\lambda_i(d)} : d \in D_i\} \cup \{\mathtt{b}_d : d \in D_i\}$,

the dominoes in D_i are called *plain dominoes* and the ones in $\overline{D}_i \setminus D_i$ are called *auxiliary dominoes*,

- $\overline{H}_i = H_i \cup \{ (d, \mathtt{c}_d^1) : d \in D_i \} \cup \{ (\mathtt{c}_d^i, \mathtt{c}_d^{i+1}) : d \in D_i \text{ and } 1 \leq i < \lambda_i(d) \}$,
- $\overline{H}_i^0 = H_i^0 \cup \{ (d, \mathtt{c}_d^1) : d \in D_i \} \cup \{ (\mathtt{c}_d^i, \mathtt{c}_d^{i+1}) : d \in D_i \text{ and } 1 \leq i < \lambda_i(d) \}$,
- $\overline{V}_i = V_i \cup \{ (d, \mathtt{b}_d) : d \in D_i \}$,
- $\overline{V}_i^0 = V_i^0 \cup \{ (d, \mathtt{b}_d) : d \in D_i \}$.

For the proof of the "if" part of Lemma 3.4 the following two facts will be instrumental.

LEMMA 3.5. *Let* (d_1, d_2, x, y) *be a configuration of the unlabelled domino bisimulation checking game* $\mathcal{B}_{\mathrm{d}}(\overline{T}_1, \overline{T}_2)$, *such that* d_i *is a plain domino and* d_{3-i} *is an auxiliary domino, for some* $i \in \{1, 2\}$. *Then Spoiler has a winning strategy from this configuration in the unlabelled game* $\mathcal{B}_{\mathrm{d}}(\overline{T}_1, \overline{T}_2)$.

Proof. Consider the case $i = 1$; the other case is symmetric. Spoiler wins immediately by making the move $(\mathtt{b}_{d_1}, x, y + 1)$: Duplicator has no response to this move. ■

LEMMA 3.6. *Let* (d_1, d_2, x, y) *be a configuration of the unlabelled domino bisimilarity checking game* $\mathcal{B}_{\mathrm{d}}(T_1, T_2)$, *such that* $\lambda_1(d_1) \neq \lambda_2(d_2)$. *Then Spoiler has a winning strategy from this configuration in the unlabelled game* $\mathcal{B}_{\mathrm{d}}(\overline{T}_1, \overline{T}_2)$.

Proof. Without loss of generality assume that $\lambda_1(d_1) > \lambda_2(d_2)$. Let Spoiler play the following sequence of moves from the configuration (d_1, d_2, x, y) in the unlabelled game $\mathcal{B}_{\mathrm{d}}(\overline{T}_1, \overline{T}_2)$:

$$(\mathtt{c}_{d_1}^1, x + 1, y), (\mathtt{c}_{d_1}^2, x + 2, y), \ldots, \big(\mathtt{c}_{d_1}^{\lambda_2(d_2)}, x + \lambda_2(d_2), y\big).$$

The only way for Duplicator to avoid losing immediately as in Lemma 3.5 is to respond with the following sequence of moves:

$$(\mathtt{c}_{d_2}^1, x + 1, y), (\mathtt{c}_{d_2}^2, x + 2, y), \ldots, \big(\mathtt{c}_{d_2}^{\lambda_2(d_2)}, x + \lambda_2(d_2), y\big).$$

From the configuration $\big(\mathtt{c}_{d_1}^{\lambda_2(d_2)}, \mathtt{c}_{d_2}^{\lambda_2(d_2)}, x + \lambda_2(d_2), y\big)$ Spoiler wins immediately by playing the move $(\mathtt{c}_{d_1}^{\lambda_2(d_2)+1}, x + \lambda_2(d_2) + 1, y)$: Duplicator has no response to this move. ■

We are now ready to establish the "if" part of Lemma 3.4.

LEMMA 3.7 ("if").
If Duplicator has a winning strategy in the unlabelled domino bisimilarity checking game $\mathcal{B}_{\mathrm{d}}(\overline{T}_1, \overline{T}_2)$ *then he has a winning strategy in the labelled game* $\mathcal{B}_{\mathrm{d}}(T_1, T_2)$.

Proof. Note that every configuration of the labelled domino bisimilarity checking game $\mathcal{B}_{\mathrm{d}}(T_1, T_2)$ is also a configuration of the unlabelled domino bisimilarity checking game $\mathcal{B}_{\mathrm{d}}(\overline{T}_1, \overline{T}_2)$. Given a winning strategy for Duplicator in the unlabelled game $\mathcal{B}_{\mathrm{d}}(\overline{T}_1, \overline{T}_2)$ we define a strategy for Duplicator in the labelled game $\mathcal{B}_{\mathrm{d}}(T_1, T_2)$ to be equal to the restriction of the former strategy to configurations and moves of the labelled game $\mathcal{B}_{\mathrm{d}}(T_1, T_2)$. This strategy is well defined because of Lemmas 3.5 and 3.6; it is clearly a winning strategy. ■

Finally, we conclude the proof of Theorem 3.2 by sketching a proof of the "only if" part of Lemma 3.4.

LEMMA 3.8 ("only if").
If Duplicator has a winning strategy in the labelled domino bisimilarity checking game $\mathcal{B}_{\mathrm{d}}(T_1, T_2)$ *then he has a winning strategy in the unlabelled game* $\mathcal{B}_{\mathrm{d}}(\overline{T}_1, \overline{T}_2)$.

Proof. Suppose that Duplicator has a winning strategy in the labelled domino bisimilarity checking game $\mathcal{B}_{\mathrm{d}}(T_1, T_2)$. A configuration of the unlabelled game $\mathcal{B}_{\mathrm{d}}(\overline{T_1}, \overline{T_2})$ is called *admissible* if it is also a configuration of the labelled game $\mathcal{B}_{\mathrm{d}}(T_1, T_2)$ and it belongs to the winning strategy for Duplicator there, i.e., if it is reachable by Duplicator playing the strategy. We define the strategy for Duplicator in the unlabelled game $\mathcal{B}_{\mathrm{d}}(\overline{T_1}, \overline{T_2})$ to be equal to the strategy in the labelled game for all admissible configurations and moves in which Spoiler chooses a plain domino. We need to define the strategy for Duplicator for the configurations that are not admissible or moves in which Spoiler chooses an auxiliary domino.

The strategy we define for Duplicator in the unlabelled game $\mathcal{B}_{\mathrm{d}}(\overline{T_1}, \overline{T_2})$ has the property that every configuration $C = (d_1, d_2, x, y)$ that belongs to the strategy, i.e., that can be reached when Duplicator follows the strategy, is one of the following forms:

1. the configuration C belongs to the winning strategy for Duplicator in the labelled game $\mathcal{B}_{\mathrm{d}}(T_1, T_2)$;

2. $d_1 = \mathtt{c}^k_{d_3}$ and $d_2 = \mathtt{c}^k_{d_4}$, for some plain dominoes $d_3 \in D_1$ and $d_4 \in D_2$, and $\lambda_1(d_3) = \lambda_2(d_4)$, and if $x \geq k$ then the configuration $(d_3, d_4, x-k, y)$ belongs to the winning strategy for Duplicator in the labelled game $\mathcal{B}_{\mathrm{d}}(T_1, T_2)$;

3. $d_1 = \mathtt{b}_{d_3}$ and $d_2 = \mathtt{b}_{d_4}$, for some plain dominoes $d_3 \in D_1$ and $d_4 \in D_2$, and $\lambda_1(d_3) = \lambda_2(d_4)$, and if $y > 0$ then the configuration $(d_3, d_4, x, y-1)$ belongs to the winning strategy for Duplicator in the labelled game $\mathcal{B}_{\mathrm{d}}(T_1, T_2)$.

It is a tedious but routine exercise to inspect that if Spoiler plays from a configuration of one of the forms 1.–3. listed above then Duplicator can always respond so as to make the next configuration fall into one of the categories 1.–3. ■

■

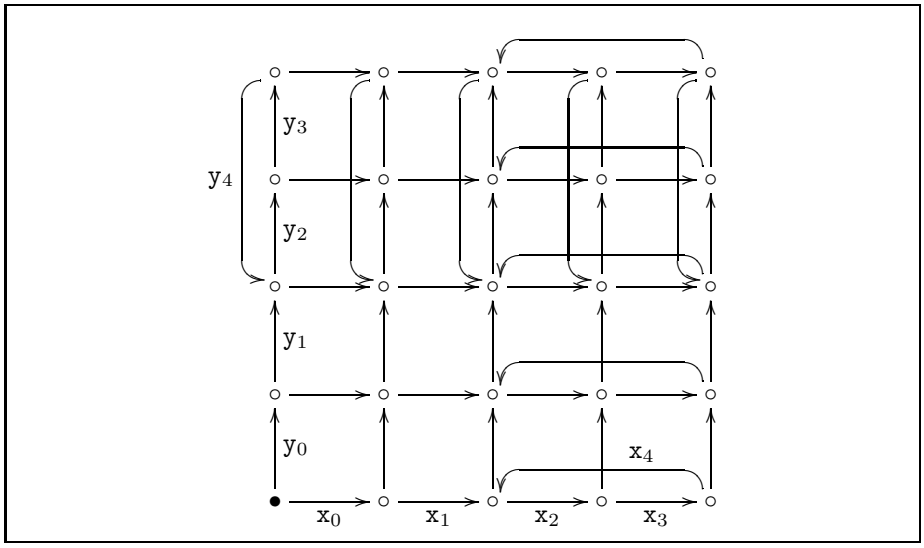


Figure 1. Modelling the infinite grid.

4. HHP-BISIMILARITY IS UNDECIDABLE

The main result of this section is a proof of Theorem 2.1, i.e., the undecidability of hhp-bisimilarity for finite state asynchronous transition systems. We also give a few extensions of this result: undecidability of hhp-bisimilarity for finite unlabelled asynchronous transition systems and for finite (unlabelled) elementary net systems.

The proof of our main result is a reduction from the problem of deciding domino bisimilarity for origin constrained tiling systems shown to be undecidable in the previous section. The method of encoding a tiling system on an infinite grid in the unfolding of a finite asynchronous transition system is due to Madhusudan and Thiagarajan [18]. For each origin constrained tiling system T, we give an effective definition a finite asynchronous transition system $A(T)$, such that the following holds.

LEMMA 4.1 (The reduction).
Origin constrained tiling systems T_1 and T_2 are domino bisimilar if and only if asynchronous transition systems $A(T_1)$ and $A(T_2)$ are hhp-bisimilar.

In Subsections 4.1–4.3 we define the finite asynchronous transition system $A(T)$ and prove Lemma 4.1 thus completing the proof of Theorem 2.1. In Subsection 4.4 we prove Theorem 2.2, i.e., we show that hhp-bisimilarity is undecidable even for unlabelled finite asynchronous transition systems. Finally, in Subsection 4.5 we argue that the hhp-bisimilarity undecidability results hold for the class of asynchronous transition systems induced by elementary net systems.

4.1. Asynchronous transition system $A(T)$

Let $T = \big(D, D^{\mathrm{ori}}, (H, H^0), (V, V^0), L, \lambda\big)$ be an origin constrained tiling system. The infinite grid structure is modelled by the unfolding of the asynchronous transition system shown in Figure 1. The set of events of this asynchronous transition

system is $E = \{\mathtt{x}_0, \mathtt{x}_1, \mathtt{x}_2, \mathtt{x}_3, \mathtt{x}_4, \mathtt{y}_0, \mathtt{y}_1, \mathtt{y}_2, \mathtt{y}_3, \mathtt{y}_4\}$. The independence relation I is the symmetric closure of $\{\ (\mathtt{x}_i, \mathtt{y}_j)\ :\ i,j \in \{0,1,2,3,4\}\ \}$.

We identify the states of the asynchronous transition system in Figure 1 with pairs of numbers $(i,j) \in \{0,1,2,3,4\}^2$, where i is the horizontal coordinate and j is the vertical coordinate. The state in the bottom-left corner in Figure 1 is $(0,0)$; it is the initial state. For all $n \in \mathbb{N}$, define:

$$\widehat{n} = \begin{cases} n & \text{if } n \leq 4, \\ 2 + \big((n-2) \bmod 3\big) & \text{if } n > 4. \end{cases}$$

A position $(n,m) \in \mathbb{N}^2$ of the infinite grid is represented by state $(\widehat{n}, \widehat{m})$ in the asynchronous transition system $A(T)$.

Configurations of the tiling system T are modelled by extra transitions going out of states of the grid structure in Figure 1, and labelled by events of the form d_{ij}, for $d \in D$, and $i,j \in \{0,1,2,3\}$. We define a set of events E_D as follows:

$$E_D = \{\ d_{ij}\ :\ d \in D;\ \text{and } i,j \in \{0,1,2,3\};\ \text{and } i=j=0 \text{ implies } d \in D^{\mathrm{ori}}\ \}.$$

The idea is, for every $d \in D$, to have a transition going out of each state $(i,j) \in \{0,1,2,3,4\}^2$ labelled with d_{ij}, provided that (d,i,j) is a configuration of T. In fact, for a technical reason we need to use events d_{i1} and d_{1j} at states $(i,4)$ and $(4,j)$, for $i,j \in \{0,1,2,3\}$, respectively, instead of d_{i4} and d_{4j}. In order to avoid special treatment of this case throughout the rest of the paper we adopt the following notation, for all $n \in \mathbb{N}$:

$$\widetilde{n} = \begin{cases} n & \text{if } n \leq 3, \\ 1 + \big((n-1) \bmod 3\big) & \text{if } n > 3. \end{cases}$$

Horizontal and vertical compatibility relations for configurations of the tiling system T are modelled by an independence relation I_D on E_D, according to which events d_{ij} and $e_{k\ell}$ corresponding to "neighbouring" configurations are independent if and only if the configurations are compatible. More precisely, we define I_D to be the symmetric closure of the following set:

$$\begin{aligned} &\{\ (d_{0j}, e_{1j})\ :\ j \in \{0,1,2,3\} \text{ and } (d,e) \in H_0\ \} \cup \\ &\{\ (d_{ij}, e_{\widetilde{(i+1)}j})\ :\ i \in \{1,2,3\}, j \in \{0,1,2,3\},\ \text{and } (d,e) \in H\ \} \cup \\ &\{\ (d_{i0}, e_{i1})\ :\ i \in \{0,1,2,3\} \text{ and } (d,e) \in V_0\ \} \cup \\ &\{\ (d_{ij}, e_{i\widetilde{(j+1)}})\ :\ i \in \{0,1,2,3\}, j \in \{1,2,3\},\ \text{and } (d,e) \in V\ \}. \end{aligned}$$

For all $i,j \in \{0,1,2,3,4\}$, and $d \in D$, we have up to four transitions going out of state (i,j) and labelled by the following events in E_D: $d_{\widetilde{i}\widetilde{j}}, d_{\widetilde{(i-1)}\widetilde{j}}$, if $i > 0$, $d_{\widetilde{i}\widetilde{(j-1)}}$ if $j > 0$, and $d_{\widetilde{(i-1)}\widetilde{(j-1)}}$ if $i,j > 0$. We write $(i,j,\{d_{i'j'}\})$ to denote the state reached by the transition labelled by the event $d_{i'j'}$ going out of state (i,j). In other words, for all $i,j \in \{0,1,2,3,4\}$ we have the following transitions:

- $(i,j) \xrightarrow{d_{\widetilde{i}\widetilde{j}}} \big(i,j,\{d_{\widetilde{i}\widetilde{j}}\}\big)$,

- $(i,j) \xrightarrow{d_{(i-1)\widetilde{j}}} \big(i,j,\{d_{(i-1)\widetilde{j}}\}\big)$ if $i>0$,
- $(i,j) \xrightarrow{d_{\widetilde{i}(j-1)}} \big(i,j,\{d_{\widetilde{i}(j-1)}\}\big)$ if $j>0$,
- $(i,j) \xrightarrow{d_{(i-1)(j-1)}} \big(i,j,\{d_{(i-1)(j-1)}\}\big)$ if $i,j>0$.

Moreover, if there are transitions:

- $(i,j) \xrightarrow{d_{k\ell}} \big(i,j,\{d_{k\ell}\}\big)$, and
- $(i,j) \xrightarrow{e_{k'\ell'}} \big(i,j,\{e_{k'\ell'}\}\big)$,

and $(d_{k\ell}, e_{k'\ell'}) \in I_D$, then there is also a state $\big(i,j,\{d_{k\ell}, e_{k'\ell'}\}\big)$ and transitions:

- $\big(i,j,\{d_{k\ell}\}\big) \xrightarrow{e_{k'\ell'}} \big(i,j,\{d_{k\ell}, e_{k'\ell'}\}\big)$, and
- $\big(i,j,\{e_{k'\ell'}\}\big) \xrightarrow{d_{k\ell}} \big(i,j,\{d_{k\ell}, e_{k'\ell'}\}\big)$.

Finally, there are transitions:

- $\big(i,j,\{d_{\widetilde{i}\widetilde{j}}\}\big) \xrightarrow{\mathtt{x}_i} \big(\widehat{i+1},j,\{d_{\widetilde{i}\widetilde{j}}\}\big)$, if $\big(i,j,\{d_{\widetilde{i}\widetilde{j}}\}\big)$ is a state, and
- $\big(i,j,\{d_{\widetilde{i}(j-1)}\}\big) \xrightarrow{\mathtt{x}_i} \big(\widehat{i+1},j,\{d_{\widetilde{i}(j-1)}\}\big)$, if $\big(i,j,\{d_{\widetilde{i}(j-1)}\}\big)$ is a state,

and transitions:

- $\big(i,j,\{d_{\widetilde{i}\widetilde{j}}\}\big) \xrightarrow{\mathtt{y}_j} \big(i,\widehat{j+1},\{d_{\widetilde{i}\widetilde{j}}\}\big)$, if $\big(i,j,\{d_{\widetilde{i}\widetilde{j}}\}\big)$ is a state, and
- $\big(i,j,\{d_{(i-1)\widetilde{j}}\}\big) \xrightarrow{\mathtt{y}_j} \big(i,\widehat{j+1},\{d_{(i-1)\widetilde{j}}\}\big)$, if $\big(i,j,\{d_{(i-1)\widetilde{j}}\}\big)$ is a state.

The sets of states $S_{A(T)}$ and transitions $\to_{A(T)}$ of the asynchronous transition system $A(T) = (S_{A(T)}, s^{\mathrm{ini}}_{A(T)}, E_{A(T)}, \to_{A(T)}, \lambda_{A(T)}, I_{A(T)})$ are as described above. The set of events is defined by $E_{A(T)} = E \cup E_D$. The initial state is $s^{\mathrm{ini}}_{A(T)} = (0,0)$. The independence relation $I_{A(T)}$ is defined as the symmetric closure of the set:

$$I \cup I_D \cup \big\{\ (\mathtt{x}_i, d_{\widetilde{i}\widetilde{j}}), (\mathtt{y}_j, d_{\widetilde{i}\widetilde{j}})\ :\ i,j \in \{0,1,2,3,4\} \text{ and } d_{\widetilde{i}\widetilde{j}} \in E_D\ \big\}.$$

Finally, the labelling function $\lambda_{A(T)}$ is an identity on E, and for elements of E_D it replaces the dominoes with their labels in the tiling system T, i.e.,

$$\lambda_{A(T)}(e) = \begin{cases} e & \text{if } e \in E, \\ \big(\lambda(d)\big)_{ij} & \text{if } e \in E_D \text{ and } e = d_{ij}. \end{cases}$$

Proposition 4.1. *The labelled transition systems $A(T)$ is a labelled asynchronous transition system.*

4.2. The unfolding of $A(T)$

In this subsection we sketch the structure of the unfolding $\mathtt{Unf}\big(A(T)\big)$ of asynchronous transition system $A(T)$ defined in the previous subsection.

For notational convenience we will write $(i,j,\emptyset)$ for a state (i,j) of $A(T)$. In order to avoid heavy use of notations $\widehat{n}$ and $\widetilde{m}$ we adopt the following conventions:

- we write $\mathtt{x}_n$ and $\mathtt{y}_m$, for all $n,m \in \mathbb{N}$, to denote events $\mathtt{x}_{\widehat{n}}, \mathtt{y}_{\widehat{m}} \in E$, respectively;

- we write d_{nm} to denote an event $d_{\widetilde{n}\widetilde{m}} \in E_D$, for all $n, m \in \mathbb{N}$.

PROPOSITION 4.2. *The set of states of* $\mathtt{Unf}\big(A(T)\big)$ *reachable from the initial state* $(0,0,\emptyset)$ *consists of triples* $(n,m,C) \in \mathbb{N}\times\mathbb{N}\times\wp(E_D)$*, such that either:*

- $C = \emptyset$*; or*
- $C = \{d_{n'm'}\}$ *such that* $d_{n'm'} \in E_D$*, and* $n' \in \{n-1, n\}$*, and* $m' \in \{m-1, m\}$*; or*
- $C = \{d_{(n-1)m'}, e_{nm'}\}$ *such that* $d_{(n-1)m'}, e_{nm'} \in E_D$*, and* $m' \in \{m-1, m\}$*, and configurations* $(d, n-1, m')$ *and* (e, n, m') *of* T *are compatible; or*
- $C = \{d_{n'(m-1)}, e_{n'm}\}$ *such that* $d_{n'(m-1)}, e_{n'm} \in E_D$*, and* $n' \in \{n-1, n\}$*, and configurations* $(d, n', m-1)$ *and* (e, n', m) *of* T *are compatible.*

States of the first category above represent positions on the infinite grid; in particular the state $(n,m,\emptyset)$ represents the position $(n,m) \in \mathbb{N}\times\mathbb{N}$. States of the second category above represent configurations of the tiling system T; in particular configuration $(d,n,m) \in D\times\mathbb{N}\times\mathbb{N}$ is represented by states $\big(n', m', \{d_{nm}\}\big)$ for $n' \in \{n, n+1\}$ and $m' \in \{m, m+1\}$. States of the third and forth categories above are used to "check compatibility" of neighbouring configurations of the tiling system T.

PROPOSITION 4.3. *The set of transitions of* $\mathtt{Unf}\big(A(T)\big)$ *consists of the following:*

- $(n,m,C) \xrightarrow{\mathtt{x}_n}_{\mathtt{Unf}(A(T))} (n+1,m,C)$

for $C = \emptyset$*, or for* $C = \{d_{nm'}\}$*, where* $m' \in \{m-1, m\}$*,*

- $(n,m,C) \xrightarrow{\mathtt{y}_m}_{\mathtt{Unf}(A(T))} (n,m+1,C)$

for $C = \emptyset$*, or for* $C = \{d_{n'm}\}$*, where* $n' \in \{n-1, n\}$*,*

- $(n,m,\emptyset) \xrightarrow{d_{n'm'}}_{\mathtt{Unf}(A(T))} \big(n,m,\{d_{n'm'}\}\big)$

for $n' \in \{n-1, n\}$*, and* $m' \in \{m-1, m\}$*, and* $d_{n'm'} \in E_D$*,*

- $\big(n,m,\{d_{(n-1)m'}\}\big) \xrightarrow{e_{nm'}}_{\mathtt{Unf}(A(T))} \big(n,m,\{d_{(n-1)m'}, e_{nm'}\}\big)$ *and* $\big(n,m,\{e_{nm'}\}\big) \xrightarrow{d_{(n-1)m'}}_{\mathtt{Unf}(A(T))} \big(n,m,\{d_{(n-1)m'}, e_{nm'}\}\big)$,

for $m' \in \{m-1, m\}$ *if configurations* $(d, n-1, m')$ *and* (e, n, m') *of* T *are compatible,*

- $\big(n,m,\{d_{n'(m-1)}\}\big) \xrightarrow{e_{n'm}}_{\mathtt{Unf}(A(T))} \big(n,m,\{d_{n'(m-1)}, e_{n'm}\}\big)$ *and* $\big(n,m,\{e_{n'm}\}\big) \xrightarrow{d_{n'(m-1)}}_{\mathtt{Unf}(A(T))} \big(n,m,\{d_{n'(m-1)}, e_{n'm}\}\big)$

for $n' \in \{n-1, n\}$*, if configurations* $(d, n', m-1)$ *and* (e, n', m) *of* T *are compatible.*

4.3. Translations between hhp- and domino bisimulations

By Proposition 2.2 it follows that in order to prove Lemma 4.1 it suffices to demonstrate that a domino bisimulation relating T_1 and T_2 gives rise to an hhp-bisimulation relating $\mathtt{Unf}\big(A(T_1)\big)$ and $\mathtt{Unf}\big(A(T_2)\big)$, and vice versa.

In other words, by Propositions 2.3 and 3.1, it suffices to argue that a winning strategy for Duplicator in $\mathcal{B}_{\mathrm{d}}(T_1, T_2)$ can be translated to a winning strategy for him in $\mathcal{B}_{\mathrm{hhp}}\big(\mathtt{Unf}(A(T_1)), \mathtt{Unf}(A(T_2))\big)$, and vice versa. In what follows, in order to keep the arguments from becoming too dull or cumbersome, we are mixing freely

at our convenience the two ways of talking about bisimulations: as relations, and as winning strategies in bisimilarity checking games.

For notational convenience we introduce the following convention for writing elements of an hhp-bisimulation relating $\mathtt{Unf}\big(A(T_1)\big)$ and $\mathtt{Unf}\big(A(T_2)\big)$, or equivalently, for configurations of game $\mathcal{B}_{\mathrm{hhp}}\big(\mathtt{Unf}(A(T_1)), \mathtt{Unf}(A(T_2))\big)$. Note that if a pair of runs $(\overline{e_1}, \overline{e_2}) \in \mathtt{Runs}\big(\mathtt{Unf}(A(T_1))\big) \times \mathtt{Runs}\big(\mathtt{Unf}(A(T_2))\big)$ belongs to an hhp-bisimulation then the states reached by these runs are of the forms (n, m, C_1) and (n, m, C_2) for some $n, m \in \mathbb{N}$, respectively. In what follows we write (n, m, C_1, C_2) to denote such a pair $(\overline{e_1}, \overline{e_2})$. This notation is a bit sloppy because it is not 1-1. For example, $(1, 1, \emptyset)$ is used to denote both $(\mathtt{x}_0\mathtt{y}_0, \mathtt{x}_0\mathtt{y}_0)$ and $(\mathtt{y}_0\mathtt{x}_0, \mathtt{y}_0\mathtt{x}_0)$. It is not hard to see that this sloppiness is not a problem here.

From domino to hhp-bisimulation. Let $B \subseteq D_1 \times D_2 \times \mathbb{N} \times \mathbb{N}$ be a domino bisimulation relating T_1 and T_2. We define a winning strategy for Duplicator in game $\mathcal{B}_{\mathrm{hhp}}\big(\mathtt{Unf}(A(T_1)), \mathtt{Unf}(A(T_2))\big)$ in the following way.

If Spoiler makes a backward move then the response of Duplicator is determined uniquely. Moreover, this response can always be performed because asynchronous transition systems $\mathtt{Unf}\big(A(T_1)\big)$ and $\mathtt{Unf}\big(A(T_2)\big)$ have the property that every pair of runs with equal labelling sequences has equal sets of most recent events. If Spoiler makes a forward move by choosing an event $\mathtt{x}_n$ or $\mathtt{y}_m$, for $n, m \in \mathbb{N}$, then Duplicator responds with the same event in the other transition system.

The only non-trivial responses of Duplicator are the ones to be made when Spoiler makes a forward move by choosing an event of the form d_{nm}, where d is a domino, and $n, m \in \mathbb{N}$. We define these responses by referring to the domino bisimulation B. The strategy for Duplicator we define below has the following property.

PROPERTY 4.1. *Suppose that a configuration (n, m, C_1, C_2) of an hhp-bisimilarity checking game $\mathcal{B}_{\mathrm{hhp}}\big(\mathtt{Unf}\big(A(T_1)\big), \mathtt{Unf}\big(A(T_2)\big)\big)$ can be reached from the initial configuration while Duplicator is playing according to the strategy. Then $d_{n'm'} \in C_1$ and $e_{n'm'} \in C_2$, for $n' \in \{n-1, n\}$ and $m' \in \{m-1, m\}$, imply that $(d, e, n', m') \in B$.*

Suppose without loss of generality that Spoiler makes a move in $\mathtt{Unf}\big(A(T_1)\big)$; the other case is symmetric. We consider several cases depending on the current configuration of $\mathcal{B}_{\mathrm{hhp}}\big(\mathtt{Unf}\big(A(T_1)\big), \mathtt{Unf}\big(A(T_2)\big)\big)$.

- The current configuration of the game $\mathcal{B}_{\mathrm{hhp}}\big(\mathtt{Unf}(A(T_1)), \mathtt{Unf}(A(T_2))\big)$ is

$$(n, m, \emptyset, \emptyset)$$

for some $n, m \in \mathbb{N}$. Spoiler can choose an event $d_{n'm'}$, such that $n' \in \{n-1, n\}$ and $m' \in \{m-1, m\}$. Then Duplicator responds with an event $e_{n'm'}$ in $\mathtt{Unf}\big(A(T_2)\big)$, such that $(d, e, n', m') \in B$.

- The current configuration of the game $\mathcal{B}_{\mathrm{hhp}}\big(\mathtt{Unf}(A(T_1)), \mathtt{Unf}(A(T_2))\big)$ is

$$\big(n, m, \{d_{n'm'}\}, \{e_{n'm'}\}\big)$$

such that $n' \in \{n-1, n\}$ and $m' \in \{m-1, m\}$. Spoiler can choose an event $d'_{k\ell}$, such that either $k = n'$ and $\{m', \ell\} = \{m-1, m\}$, or $\ell = m'$ and $\{n', k\} = \{n-1, n\}$. In both cases Duplicator responds with an event $e'_{k\ell}$, such that configurations (e, n', m') and (e', k, ℓ) of T_2 are compatible, and $(d', e', k, \ell) \in B$.

Note that all the responses we have defined above are indeed possible due to Property 4.1 and the definition of a domino bisimulation, and moreover, they maintain Property 4.1.

From hhp- to domino bisimulation. Let B be an hhp-bisimulation relating $\mathtt{Unf}\big(A(T_1)\big)$ and $\mathtt{Unf}\big(A(T_2)\big)$. We define a winning strategy for Duplicator in game $\mathcal{B}_{\mathrm{d}}(T_1, T_2)$. The strategy for Duplicator we define below has the following property.

PROPERTY 4.2. *If configuration (d, e, n, m) of $\mathcal{B}_{\mathrm{d}}(T_1, T_2)$ can be reached while Duplicator is playing according to the strategy then $\big(n, m, \{d_{nm}\}, \{e_{nm}\}\big) \in B$.*

Suppose without loss of generality that Spoiler makes a move in T_1; the other case is symmetric. We consider the two kinds of moves possible in a domino bisimilarity game.

• In order to fix an initial configuration of $\mathcal{B}_{\mathrm{d}}(T_1, T_2)$ Spoiler chooses a configuration (d, n, m) of T_1. Note that for all $n, m \in \mathbb{N}$, we have that $(n, m, \emptyset, \emptyset) \in B$. Let e_{nm} be Duplicator's response if Spoiler makes a forward move in the game $\mathcal{B}_{\mathrm{hhp}}\big(\mathtt{Unf}\big(A(T_1)\big), \mathtt{Unf}\big(A(T_2)\big)\big)$ by choosing event d_{nm} in configuration $(n, m, \emptyset, \emptyset)$. Then we take e to be Duplicator's response to Spoiler's choice of configuration (d, n, m).

• Let (d, e, n, m) be the current configuration of $\mathcal{B}_{\mathrm{d}}(T_1, T_2)$. In a next move Spoiler can choose a configuration (d', n', m') of T_1 compatible with (d, n, m). We consider cases when $(n', m') = (n - 1, m)$ and $(n', m') = n, m + 1)$; the other two cases are analogous. Note that by Property 4.2 we have that

$$\big(n, m, \{d_{nm}\}, \{e_{nm}\}\big) \in B. \tag{1}$$

– Let $(n', m') = (n-1, m)$. Since configurations (d, n, m) and $(d', n-1, m)$ of T_1 are compatible, by applying condition 2. of the definition of an hhp-bisimulation to (1) we get that there is a domino e' of T_2, such that configurations (e, n, m) and $(e', n-1, m)$ of T_2 are compatible, and

$$\big(n, m, \{d'_{(n-1)m}, d_{nm}\}, \{e'_{(n-1)m}, e_{nm}\}\big) \in B. \tag{2}$$

We define event e' to be Duplicator's response in $\mathcal{B}_{\mathrm{d}}(T_1, T_2)$ for Spoiler's move consisting of choosing configuration $(d', n-1, m)$ of T_1. By applying condition 4. of the definition of an hhp-bisimulation to (2) twice we get that

$$\big(n-1, m, \{d'_{(n-1)m}\}, \{e'_{(n-1)m}\}\big) \in B.$$

– Let $(n', m') = (n, m+1)$. By applying condition 2. of the definition of an hhp-bisimulation to (1) we get that

$$\big(n, m+1, \{d_{nm}\}, \{e_{nm}\}\big) \in B. \tag{3}$$

Since configurations (d, n, m) and $(d', n, m+1)$ of T_1 are compatible, by applying condition 2. of the definition of an hhp-bisimulation to (3) we get that there is a domino e' of T_2, such that configurations (e, n, m) and $(e', n, m+1)$ of T_2 are compatible, and

$$\big(n, m+1, \{d_{nm}, d'_{n(m+1)}\}, \{e_{nm}, e'_{n(m+1)}\}\big) \in B. \tag{4}$$

We define event e' to be Duplicator's response in $\mathcal{B}_{\mathrm{d}}(T_1, T_2)$ for Spoiler's move consisting of choosing configuration $(d', n, m+1)$ of T_1. By applying condition 4. of the definition of an hhp-bisimulation to (4) we get

$$\big(n, m+1, \{d'_{n(m+1)}\}, \{e'_{n(m+1)}\}\big) \in B.$$

Note that all the responses we have defined above are indeed possible due to Property 4.2 and the definition of a domino bisimulation, and moreover, they maintain Property 4.2.

4.4. Unlabelled hhp-bisimilarity

In this subsection we prove Theorem 2.2, i.e., we show that undecidability of hhp-bisimilarity holds also for unlabelled asynchronous transition systems. For an unlabelled origin constrained tiling system T we give an effective definition of an asynchronous transition system $A'(T)$, which is obtained by only a slight modification of the asynchronous transition system $A(T)$: we add new states s_1, s_2, s_3, s_4, s_5, and new transitions $(0,1) \xrightarrow{\mathtt{c}} s_1 \xrightarrow{\mathtt{c}} s_2 \xrightarrow{\mathtt{c}} s_3 \xrightarrow{\mathtt{c}} s_4 \xrightarrow{\mathtt{c}} s_5$, where $\mathtt{c}$ is a new event. The independence relation of $A'(T)$ is the same as in $A(T)$. There is no labelling function in $A'(T)$.

Obviously, if Duplicator has a winning strategy in the labelled hhp-bisimilarity checking game $\mathcal{B}_{\mathrm{hhp}}\big(\mathtt{Unf}(A(T_1)), \mathtt{Unf}(A(T_2))\big)$ then he has also a winning strategy in the unlabelled game $\mathcal{B}_{\mathrm{hhp}}\big(\mathtt{Unf}(A'(T_1)), \mathtt{Unf}(A'(T_2))\big)$. By determinacy of bisimilarity checking games the following lemma is the converse of the previous statement.

LEMMA 4.2 (The reduction).
If Spoiler has a winning strategy in the labelled hhp-bisimilarity checking game $\mathcal{B}_{\mathrm{hhp}}\big(\mathtt{Unf}(A(T_1)), \mathtt{Unf}(A(T_2))\big)$ *then he has a winning strategy in the unlabelled game* $\mathcal{B}_{\mathrm{hhp}}\big(\mathtt{Unf}(A'(T_1)), \mathtt{Unf}(A'(T_2))\big)$.

The rest of this section is devoted to proving the above lemma from which Theorem 2.2 follows by Theorem 3.2 and the proof of Theorem 2.1.

For notational convenience, below we sometimes identify configurations of the games $\mathcal{B}_{\mathrm{hhp}}\big(\mathtt{Unf}(A(T_1)), \mathtt{Unf}(A(T_2))\big)$ or $\mathcal{B}_{\mathrm{hhp}}\big(\mathtt{Unf}(A'(T_1)), \mathtt{Unf}(A'(T_2))\big)$, respectively, with pairs of states of the transition systems $\mathtt{Unf}\big(A(T_1)\big)$ and $\mathtt{Unf}\big(A(T_2)\big)$, or from the transition systems $\mathtt{Unf}\big(A'(T_1)\big)$ and $\mathtt{Unf}\big(A'(T_2)\big)$, respectively. The following observation will be useful.

LEMMA 4.3. *Let* (n_1, m_1, C_1) *and* (n_2, m_2, C_2) *be states of* $\mathtt{Unf}\big(A'(T_1)\big)$ *and* $\mathtt{Unf}\big(A'(T_2)\big)$, *respectively, such that* $|C_1| \neq |C_2|$. *Then Spoiler has a winning strategy in* $\mathcal{B}_{\mathrm{hhp}}\big(\mathtt{Unf}(A'(T_1)), \mathtt{Unf}(A'(T_2))\big)$ *from these states.*

Proof. Note that $|C_1|, |C_2| \in \{0, 1, 2\}$. If $|C_i| = 0$ and $|C_{3-i}| \neq 0$, for $i \in \{1, 2\}$, then an infinite number of forward moves is possible from (n_i, m_i, C_i), and only finitely many from the state $(n_{3-i}, m_{3-i}, C_{3-i})$. Hence Spoiler has a winning strategy in this case.

If $|C_1| > 0$, $|C_2| > 0$, and $|C_1| \neq |C_2|$ then a winning strategy for Spoiler is as follows. Assume without loss of generality that $|C_1| = 1$; then of course $|C_2| = 2$. Spoiler makes a backward move in the first component, such that the next configu-

ration consists of states $(n_1, m_1, \emptyset)$ and (n_2', m_2', C_2'), where $|C_2'| \neq 0$. By the preceding argument it follows that Spoiler has a winning strategy from this configuration. ■

The argument to prove Lemma 4.2 is as follows. Consider a winning strategy for Spoiler in the game $\mathcal{B}_{\mathrm{hhp}}\big(\mathtt{Unf}(A(T_1)), \mathtt{Unf}(A(T_2))\big)$. A configuration of the unlabelled hhp-bisimilarity checking game $\mathcal{B}_{\mathrm{hhp}}\big(\mathtt{Unf}(A'(T_1)), \mathtt{Unf}(A'(T_2))\big)$ is called *admissible* if the corresponding configuration in the labelled hhp-bisimilarity checking game $\mathcal{B}_{\mathrm{hhp}}\big(\mathtt{Unf}(A(T_1)), \mathtt{Unf}(A(T_2))\big)$ belongs to the winning strategy for Spoiler, i.e., if this configuration is reachable by playing the strategy from the initial configuration $(0, 0, \emptyset, \emptyset)$. Obviously, the initial configuration is admissible. We define a strategy for Spoiler in the unlabelled game to be equal to the strategy in the labelled game in all admissible configurations. In a sequence of lemmas (Lemmas 4.4–4.7 below) we argue that in all admissible configurations, every response of Duplicator which leads to a non-admissible configuration enables Spoiler to win in a finite number of steps.

LEMMA 4.4. *Let* $(0, 0, \emptyset)$ *and* $(0, 0, \emptyset)$ *be states of* $\mathtt{Unf}\big(A'(T_1)\big)$ *and* $\mathtt{Unf}\big(A'(T_2)\big)$, *respectively. If Spoiler chooses the move* $(0, 0, \emptyset) \xrightarrow{\mathtt{x}_0} (1, 0, \emptyset)$ *then Duplicator must answer with the same move in the other system; otherwise Spoiler can win in a finite number of steps. The similar holds if Spoiler chooses the move* $(0, 0, \emptyset) \xrightarrow{\mathtt{y}_0} (0, 1, \emptyset)$.

Proof. By Lemma 4.3 we know that Duplicator cannot choose any transition $(0, 0, \emptyset) \xrightarrow{d_{00}} \big(0, 0, \{d_{00}\}\big)$, for $d \in D$. Suppose then that his choice was $(0, 0, \emptyset) \xrightarrow{\mathtt{y}_0} (0, 1, \emptyset)$. We argue that Spoiler has a winning strategy now. He plays $(0, 1, \emptyset) \xrightarrow{c} s_1$. If Duplicator answers with some $(n, m, \emptyset)$ then he loses because of similar arguments as in the proof of Lemma 4.3. In any other case (after performing this move) there are at most three possible forward moves for Duplicator, whereas Spoiler can perform another four forward moves. Therefore, Spoiler has a winning strategy. ■

LEMMA 4.5. *Note that* $(n, m, \emptyset, \emptyset)$ *is an admissible configuration in the unlabelled hhp-bisimilarity checking game* $\mathcal{B}_{\mathrm{hhp}}\big(\mathtt{Unf}(A'(T_1)), \mathtt{Unf}(A'(T_2))\big)$, *for all* $n, m \in \mathbb{N}$. *If Spoiler chooses the move* $(n, m, \emptyset) \xrightarrow{\mathtt{x}_n} (n+1, m, \emptyset)$ *then Duplicator must answer with the same move in the other system; otherwise Spoiler can win in a finite number of steps. The similar holds if Spoiler chooses the move* $(n, m, \emptyset) \xrightarrow{\mathtt{y}_m} (n, m+1, \emptyset)$.

Proof. The special case when $n = m = 0$ is proved in Lemma 4.4. By Lemma 4.3 it follows that Duplicator cannot choose any transition $(n, m, \emptyset) \xrightarrow{d_{n'm'}} \big(n, m, \{d_{n'm'}\}\big)$. Suppose that the response of Duplicator is $(n, m, \emptyset) \xrightarrow{\mathtt{y}_m} (n, m+1, \emptyset)$. Assume without loss of generality that the last event in the run $\overline{e}$ leading to the state $(n, m, \emptyset)$ was $\mathtt{y}_{m-1}$. Now, Spoiler can win immediately by taking a backwards move since $n + m \in \mathrm{MR}(\overline{e} \cdot \mathtt{x}_n)$ and $n + m \notin \mathrm{MR}(\overline{e} \cdot \mathtt{y}_m)$. ■

LEMMA 4.6. *Note that* $(n, m, \emptyset, \emptyset)$ *is an admissible configuration in the unlabelled hhp-bisimilarity checking game* $\mathcal{B}_{\mathrm{hhp}}\big(\mathtt{Unf}(A'(T_1)), \mathtt{Unf}(A'(T_2))\big)$, *for all* $n, m \in \mathbb{N}$.

If Spoiler chooses a move

$$(n, m, \emptyset) \xrightarrow{d_{n'm'}} \big(n, m, \{d_{n'm'}\}\big),$$

for $n' \in \{n-1, n\}$, $m' \in \{m-1, m\}$, *and* $d_{n'm'} \in E_{D_1}$, *then Duplicator must answer with*

$$(n, m, \emptyset) \xrightarrow{e_{n'm'}} \big(n, m, \{e_{n'm'}\}\big)$$

in the other system, for some $e_{n'm'} \in E_{D_2}$*; otherwise Spoiler can win in a finite number of steps.*

Proof. By Lemma 4.3 Duplicator must respond with a move of the form

$$(n, m, \emptyset) \xrightarrow{e_{n''m''}} \big(n, m, \{e_{n''m''}\}\big).$$

We show that Spoiler has a winning strategy in all cases where $n' \neq n''$ or $m' \neq m''$. We use Lemma 4.3 without explicitly mentioning it.

- If $n' = n-1$ and $m' = m-1$ then Duplicator looses for any other choice of indices n'' and m''. The reason is that from the state $\big(n, m, \{d_{n'm'}\}\big)$ no forward move that preserves the cardinality of the set $\{d_{n'm'}\}$ is available, and instead from the state $\big(n, m, \{e_{n''m''}\}\big)$, a forward move can be made by choosing either the event $\mathtt{x}_n$ or $\mathtt{y}_m$.
- If $n' = n$ and $m' = m$ then two forward moves are possible for Spoiler from the state $\big(n, m, \{d_{n'm'}\}\big)$ that preserve the cardinality of the set $\{d_{n'm'}\}$. If Duplicator chooses a different index n'' or m'' then he can perform at most one forward move preserving the cardinality of the set $\{e_{n''m''}\}$.
- If $n' = n-1$ and $m' = m$ then the only choice for Duplicator, that does not respect indices n' and m', is $n'' = n$ and $m'' = m-1$ since for the other choices where $n'' = n$ and $m'' = m$, or $n'' = n-1$ and $m'' = m-1$, Duplicator looses because of the previous arguments. Consider the states $\big(n, m, \{d_{(n-1)m}\}\big)$ and $\big(n, m, \{e_{n(m-1)}\}\big)$. We show that Spoiler has a winning strategy starting from these states. Spoiler chooses

$$\big(n, m, \{d_{(n-1)m}\}\big) \xrightarrow{\mathtt{y}_m} \big(n, m+1, \{d_{(n-1)m}\}\big)$$

and Duplicator's only response is

$$\big(n, m, \{e_{n(m-1)}\}\big) \xrightarrow{\mathtt{x}_n} \big(n+1, m, \{e_{n(m-1)}\}\big).$$

However, Spoiler can now perform a backwards move reaching the state $\big(n, m+1, \emptyset\big)$ and Duplicator's only response is by the same backwards move, reaching the state $\big(n+1, m, \emptyset\big)$. Spoiler has a winning strategy now, by the arguments from Lemma 4.5.

- The case where $n' = n$ and $m' = m-1$ is similar to the previous one.

■

LEMMA 4.7. *Let* $\big(n, m, \{d_{n'm'}\}, \{e_{n'm'}\}\big)$ *be an admissible configuration in the unlabelled hhp-bisimilarity checking game* $\mathcal{B}_{\mathrm{hhp}}\big(\mathtt{Unf}(A'(T_1)), \mathtt{Unf}(A'(T_2))\big)$. *If Spoiler chooses a move*

$$(n, m, \{d_{n'm'}\}) \stackrel{d'_{n''m''}}{\longrightarrow} \big(n, m, \{d_{n'm'}, d'_{n''m''}\}\big),$$

for some $n' \in \{n-1, n\}$, $m' \in \{m-1, m\}$, *and* $d'_{n''m''} \in E_{D_1}$, *then Duplicator must answer with*

$$(n, m, \{e_{n'm'}\}) \stackrel{e'_{n''m''}}{\longrightarrow} \big(n, m, \{e_{n'm'}, e'_{n''m''}\}\big)$$

in the other system, for some $e'_{n''m''} \in E_{D_2}$; *otherwise Spoiler can win in a finite number of steps.*

We omit the proof of this lemma; similar arguments can be used as in the proof of Lemma 4.6.

Observe that all the E_D-events in the labelled asynchronous transition system $A(T)$ have the same label since T is by our assumption an unlabelled origin constrained tiling system. Therefore, Lemmas 4.4–4.7 cover all the relevant cases in the argument for Lemma 4.2 sketched before Lemma 4.4. This concludes the reduction of unlabelled domino bisimilarity to unlabelled hhp-bisimilarity, and hence Theorem 2.2 follows from Theorem 3.2.

4.5. Finite elementary net system $N(T)$

In this subsection we argue that undecidability of hhp-bisimilarity for finite elementary net systems follows as a corollary of our proof for finite asynchronous transition systems.

Given a tiling system T we define an elementary net system $N(T)$ and we argue that $A(T)$ is isomorphic to the asynchronous transition system $na\big(N(T)\big)$ corresponding to the net $N(T)$; see the articles by Nielsen and Winskel [30, 23] for the definition of the asynchronous transition system $na\big(N(T)\big)$. This immediately implies the following facts.

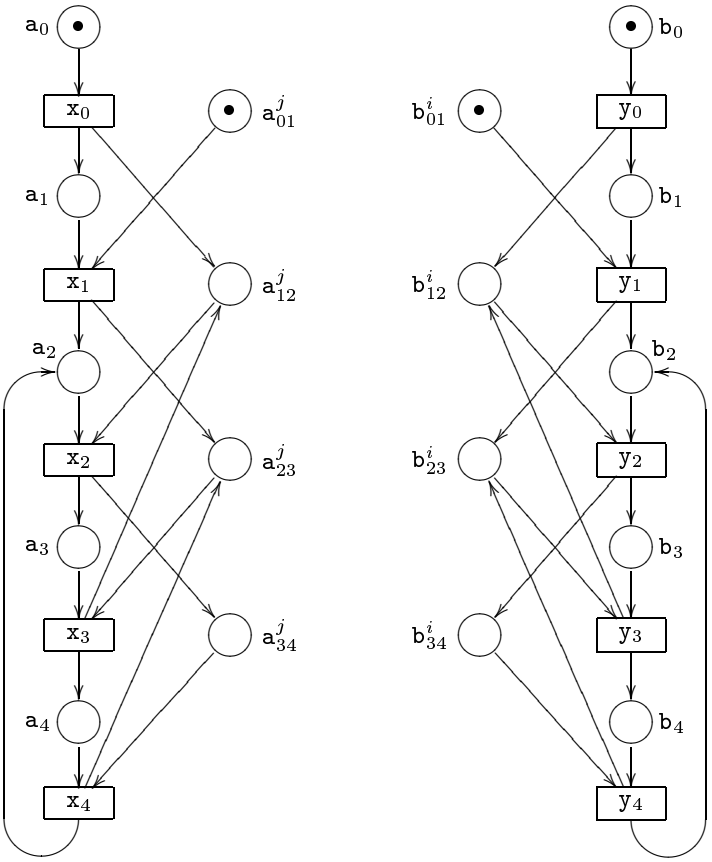
THEOREM 4.1. *Hhp-bisimilarity is undecidable for finite labelled elementary net systems.*

THEOREM 4.2. *Hhp-bisimilarity is undecidable for finite unlabelled elementary net systems.*

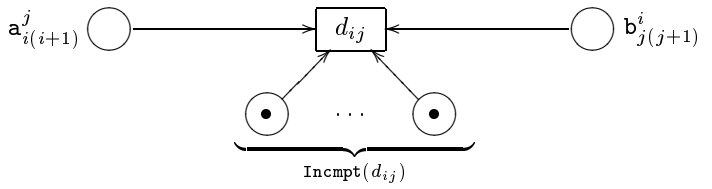
The elementary net system $N(T) = (P_{N(T)}, E_{N(T)}, \mathtt{pre}_{N(T)}, \mathtt{post}_{N(T)}, M_{N(T)})$ is shown in Figure 2 and it consists of the following:

- the set of conditions

$$\begin{aligned} P_{N(T)} \;&=\; \big\{\, \mathtt{a}_i, \mathtt{b}_i \;:\; i \in \{0,1,2,3,4\} \,\big\} \\ &\cup\; \big\{\, \mathtt{a}^j_{i(i+1)}, \mathtt{b}^i_{j(j+1)} \;:\; i,j \in \{0,1,2,3\} \,\big\} \cup E_D; \end{aligned}$$



Variables i and j range over $\{0,1,2,3\}$. Therefore we have four copies of each place in the second column, for $j \in \{0,1,2,3\}$, and four copies of each place in the third column, for $i \in \{0,1,2,3\}$.



The preconditions of each $d_{ij} \in E_D$, are the two places $\mathtt{a}_{i(i+1)}^j$ and $\mathtt{b}_{j(j+1)}^i$, and all the places in $\mathtt{Incmpt}(d_{ij}) = \{\, e_{k\ell} \in E_D \;:\; (d_{ij}, e_{k\ell}) \notin I_D \,\}$.

Figure 2. The elementary net system $N(T)$.

- the set of events $E_{N(T)} = E_{A(T)}$;
- the function $\mathtt{pre}_{N(T)} : E_{N(T)} \to \wp(P_{N(T)})$ specifying the set of places in the pre-condition of an event:

$$\mathtt{pre}_{N(T)}(e) = \begin{cases} \{\mathtt{a}_0\} & \text{if } e = \mathtt{x}_0, \\ \{\mathtt{b}_0\} & \text{if } e = \mathtt{y}_0, \\ \{\mathtt{a}_i\} \cup \mathtt{A}_{(i-1)i} & \text{if } e = \mathtt{x}_i \text{ for } i \in \{1,2,3,4\}, \\ \{\mathtt{b}_j\} \cup \mathtt{B}_{(j-1)j} & \text{if } e = \mathtt{y}_j \text{ for } j \in \{1,2,3,4\}, \\ \{\mathtt{a}^j_{i(i+1)}, \mathtt{b}^i_{j(j+1)}\} \cup \mathtt{Incmpt}(d_{ij}) & \text{if } e = d_{ij} \in E_D, \end{cases}$$

where for $i, j \in \{0,1,2,3\}$, we define $\mathtt{A}_{i(i+1)} = \left\{ \mathtt{a}^k_{i(i+1)} \ : \ k \in \{0,1,2,3\} \right\}$ and $\mathtt{B}_{j(j+1)} = \left\{ \mathtt{b}^k_{j(j+1)} \ : \ k \in \{0,1,2,3\} \right\}$, and for $d_{ij} \in E_D$, we define the set of dominoes *incompatible* with d_{ij} by:

$$\mathtt{Incmpt}(d_{ij}) = \{\, e_{k\ell} \in E_D \ : \ (d_{ij}, e_{k\ell}) \notin I_D \,\};$$

- the function $\mathtt{post}_{N(T)} : E_{N(T)} \to \wp(P_{N(T)})$ specifying the set of places in the post-condition of an event:

$$\mathtt{post}_{N(T)}(e) = \begin{cases} \{\mathtt{a}_{i+1}\} \cup \mathtt{A}_{(i+1)(i+2)} & \text{if } e = \mathtt{x}_i \text{ for } i \in \{0,1,2\}, \\ \{\mathtt{a}_4\} \cup \mathtt{A}_{12} & \text{if } e = \mathtt{x}_3, \\ \{\mathtt{a}_2\} \cup \mathtt{A}_{23} & \text{if } e = \mathtt{x}_4, \\ \{\mathtt{b}_{j+1}\} \cup \mathtt{B}_{(j+1)(j+2)} & \text{if } e = \mathtt{y}_j \text{ for } j \in \{0,1,2\}, \\ \{\mathtt{b}_4\} \cup \mathtt{B}_{12} & \text{if } e = \mathtt{y}_3, \\ \{\mathtt{b}_2\} \cup \mathtt{B}_{23} & \text{if } e = \mathtt{y}_4, \\ \emptyset & \text{if } e \in E_D; \end{cases}$$

- the initial marking $M_{N(T)} = \{\mathtt{a}_0, \mathtt{b}_0\} \cup \mathtt{A}_{01} \cup \mathtt{B}_{01} \cup E_D$.

PROPOSITION 4.4. *The asynchronous transition system $A(T)$ is isomorphic to $na\big(N(T)\big)$.*

Proof. We define a function $\Xi : S_{A(T)} \to \wp(P_{N(T)})$ as follows:

$$\Xi\big((i,j,C)\big) = \{\mathtt{a}_i, \mathtt{b}_j\} \cup \mathtt{X}_i \cup \mathtt{Y}_j \cup E_D \setminus \bigcup_{c \in C} \mathtt{pre}_{N(T)}(c),$$

where

$$\mathtt{X}_i = \begin{cases} \mathtt{A}_{01} & \text{if } i = 0, \\ \mathtt{A}_{(i-1)i} \cup \mathtt{A}_{i(i+1)} & \text{if } i \in \{1,2,3\}, \\ \mathtt{A}_{34} \cup \mathtt{A}_{12} & \text{if } i = 4, \end{cases}$$

and similarly

$$\mathtt{Y}_j = \begin{cases} \mathtt{B}_{01} & \text{if } j = 0, \\ \mathtt{B}_{(j-1)j} \cup \mathtt{B}_{j(j+1)} & \text{if } j \in \{1,2,3\}, \\ \mathtt{B}_{34} \cup \mathtt{B}_{12} & \text{if } j = 4. \end{cases}$$

In order to argue that Ξ is an isomorphism of asynchronous transition systems $A(T)$ and $na\big(N(T)\big)$ it suffices to establish the following:

1. $\Xi(s^{\mathrm{ini}}_{A(T)}) = M_{N(T)}$, i.e., the initial state of $A(T)$ is mapped by Ξ to the initial marking of $N(T)$,

2. for all $s \in S_{A(T)}$,

   (i) if $s \stackrel{e}{\rightarrow}_{A(T)} t$ then $\Xi(s) \stackrel{e}{\rightarrow}_{na(N(T))} \Xi(t)$,

   (ii) if $\Xi(s) \stackrel{e}{\rightarrow}_{na(N(T))} M$ then there is $t \in S_{A(T)}$, such that $s \stackrel{e}{\rightarrow}_{A(T)} t$ and $M = \Xi(t)$,

3. $(e, f) \in I_{A(T)}$ if and only if ${}^\bullet e^\bullet \cap {}^\bullet f^\bullet = \emptyset$.

It is a tedious but routine exercise to verify that clauses 1.–3. hold. ■

ACKNOWLEDGMENT

We are grateful to Thomas Hildebrandt, P. S. Thiagarajan, and Sibille Fröschle for many stimulating discussions on the subject, and to Julian Bradfield and Walter Vogler for comments that helped to improve the presentation.

REFERENCES

1. 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 at `http://www.ipipan.gda.pl/~marek`.
2. Egon Börger, Erich Grädel, and Yuri Gurevich. *The Classical Decision Problem.* Springer-Verlag, 1997.
3. Gian Luca Cattani and Vladimiro Sassone. Higher dimensional transition systems. In *Proceedings, 11th Annual IEEE Symposium on Logic in Computer Science*, pages 55–62, New Brunswick, New Jersey, 27–30 July 1996. IEEE Computer Society Press.
4. Søren Christensen, Hans Hüttel, and Colin Stirling. Bisimulation equivalence is decidable for all context-free processes. *Information and Computation*, 121(2):143–148, 1995.
5. Yael Etzion-Petruschka, David Harel, and Dale Myers. On the solvability of domino snake problems. *Theoretical Computer Science*, 131:243–269, 1994.
6. Sibylle Fröschle. Decidability of plain and hereditary history-preserving bisimilarity for BPP. In Ilaria Castellani and Björn Victor, editors, *Proceedings of EXPRESS'99 the 6th International Workshop on Expressiveness in Concurrency*, volume 27 of *Electronic Notes in Theoretical Computer Science*, 2000.
7. Sibylle Fröschle and Thomas Hildebrandt. On plain and hereditary history-preserving bisimulation. In Mirosław Kutyłowski, Leszek Pacholski, and Tomasz Wierzbicki, editors, *Mathematical Foundations of Computer Science 1999, 24th International Symposium, MFCS'99*, volume 1672 of *LNCS*, pages 354–365, Szklarska Poręba, Poland, 6–10 September 1999. Springer-Verlag.
8. R. J. van Glabbeek. The linear time-branching time spectrum (Extended abstract). In J. C. M. Baeten and J. W. Klop, editors, *CONCUR '90, Theories of Concurrency: Unification and Extension*, volume 458 of *LNCS*, pages 278–297, Amsterdam, The Netherlands, 27–30 August 1990. Springer-Verlag.
9. 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.
10. Erich Grädel. Domino games and complexity. *SIAM Journal on Computing*, 19(5):787–804, 1990.
11. Jan Friso Groote and Hans Hüttel. Undecidable equivalences for basic process algebra. *Information and Computation*, 115(2):354–371, 1994.

12. Matthew Hennessy and Robin Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM*, 32(1):137–161, 1985.

13. Yoram Hirshfeld, Mark Jerrum, and Faron Moller. A polynomial algorithm for deciding bisimilarity of normed context-free processes. *Theoretical Computer Science*, 158(1–2):143–159, 1996.

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

15. 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.

16. Marcin Jurdziński and Mogens Nielsen. Hereditary history preserving bisimilarity is undecidable. In Horst Reichel and Sophie Tison, editors, *STACS 2000, 17th Annual Symposium on Theoretical Aspects of Computer Science, Proceedings*, volume 1770 of *Lecture Notes in Computer Science*, pages 358–369, Lille, France, February 2000. Springer-Verlag.

17. Paris C. Kanellakis and Scott A. Smolka. CCS expressions, finite state processes, and three problems of equivalence. *Information and Computation*, 86(1):43–68, 1990.

18. 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-Verlag.

19. R. Milner. *A Calculus of Communicating Systems*, volume 92 of *LNCS*. Springer-Verlag, 1980.

20. M. Minsky. *Computation: Finite and Infinite Machines*. Prentice Hall, 1967.

21. Faron Moller and Scott A. Smolka. On the computational complexity of bisimulation. *ACM Computing Surveys*, 27(2):287–289, 1995.

22. Mogens Nielsen and Christian Clausen. Games and logics for a noninterleaving bisimulation. *Nordic Journal of Computing*, 2(2):221–249, 1995.

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

24. Robert Paige and Robert E. Tarjan. Three partition refinement algorithms. *SIAM Journal on Computing*, 16(6):973–989, 1987.

25. D. M. R. Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, *Theoretical Computer Science: 5th GI-Conference*, volume 104 of *LNCS*, pages 167–183. Springer-Verlag, 1981.

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

27. Colin Stirling. Bisimulation, model checking and other games. Notes for Mathfit Instructional Meeting on Games and Computation, available at `http://www.dcs.ed.ac.uk/home/cps`, 1997.

28. Wolfgang Thomas. On the Ehrenfeucht-Fraïssé game in theoretical computer science. In M.-C. Gaudel and J.-P. Jouannaud, editors, *TAPSOFT'93: Theory and Practice of Software Development, 4th International Joint Conference CAAP/FASE*, volume 668 of *LNCS*, pages 559–568, Orsay, France, April 1993. Springer-Verlag.

29. 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.

30. 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.