Composition and Decomposition in True-Concurrency

Sibylle Fröschle*

Institute of Informatics, University of Warsaw, Poland
sib@mimuw.edu.pl

Abstract. The idea of composition and decomposition to obtain computability results is particularly relevant for true-concurrency. In contrast to the interleaving world, where composition and decomposition must be considered with respect to a process algebra operator, e.g. parallel composition, we can directly recognize whether a truly-concurrent model such as a labelled asynchronous transition system or a 1-safe Petri net can be dissected into independent 'chunks of behaviour'. In this paper we introduce the corresponding concept 'decomposition into independent components', and investigate how it translates into truly-concurrent bisimulation equivalences. We prove that, under a natural restriction, history preserving (hp), hereditary hp (hhp), and coherent hhp (chhp) bisimilarity are decomposable with respect to prime decompositions. Apart from giving a general proof technique our decomposition theory leads to several coincidence results. In particular, we resolve that hp, hhp, and chhp bisimilarity coincide for 'normal form' basic parallel processes.

1 Introduction

In the finite-state world truly-concurrent problems are typically harder than their interleaving counterparts. This is demonstrated by the following examples. Model-checking CTL is well-known to be polynomial-time but model-checking CTL$_P$ is NP-hard [1]. The problem of synthesizing controllers for discrete event systems is decidable in an interleaving setting and can be computed in polynomial-time; in a truly-concurrent setting the problem is undecidable [2]. Classical bisimilarity is polynomial-time decidable while *hereditary history preserving* (*hhp*) *bisimilarity* has been proved undecidable [3]; plain *history preserving* (*hp*) *bisimilarity* is decidable [4] but has been shown DEXPTIME-complete [5, 6].

There is, however, a positive trend for true-concurrency in the *infinite*-state world. The above effect seems reversed for *basic parallel processes* (*BPP*). Under interleaving semantics a small fragment of a logic equivalent to CTL* is undecidable for *very basic* BPP; under partial order interpretation the full logic is decidable for BPP [7]. Trace equivalence on BPP is undecidable but pomset trace and location trace equivalence on BPP are shown decidable in [8]. Classical bisimilarity on BPP is PSPACE-complete [9, 10]; in contrast, for BPP, distributed bisimilarity, and with it hp bisimilarity, are polynomial-time decidable [11]. The positive trend is further confirmed by results of

[*] This work was supported by the EPSRC Grants GR/M84763 and GR/R16891, and the European Community Research Training Network 'GAMES'.

V. Sassone (Ed.): FOSSACS 2005, LNCS 3441, pp. 333–347, 2005.

[12, 13]: hhp bisimilarity on BPP is decidable and coincides with its strengthening to *coherent hhp bisimilarity*.

We can explain this discrepancy as follows. Models such as *labelled asynchronous transition systems* (*lats'*) [14] or *labelled 1-safe Petri nets* (*net systems*) faithfully capture how the transitions of a system are related concerning concurrency and conflict. The way we allow concurrency and conflict to interact will directly impact on the computational power of truly-concurrent equivalences and logics. The negative results of [2] and [3] build on the insight that truly-concurrent models have the power to encode tiling systems. If the interplay between concurrency and conflict is restricted this power can be lost [15], and a truly-concurrent concept may be particularly natural to decide. BPP are infinite-state but, under truly-concurrent semantics, they have a simple tree-like structure, which has turned out to be directly exploitable: e.g. the decidability results of [8] follow by a reduction to the equivalence problem of recognizable tree languages.

In this paper we advocate the following thesis. System classes with a restricted interplay between concurrency and conflict often have characteristic decomposition properties. These might translate into truly-concurrent equivalences or logics in a very concrete way, and thereby allow us to decide the respective concept by a 'divide and conquer' approach.

The idea of decomposition provides one of the crucial techniques to establish decidability and upper complexity bounds in infinite-state verification. For example, the polynomial-time decision procedure for classical bisimilarity on normed BPP [16] is based on the following insight:[1] any normed BPP can be expressed uniquely, up to bisimilarity, as a parallel composition of prime factors [18]. A process is *prime* if it is not the nil process and it is irreducible with respect to parallel composition, up to bisimilarity. Such a decomposition theory translates into cancellation properties of the form "$P \,||\, Q \sim R \,||\, Q$ implies $P \sim R$", which provide the means to reduce pairs of processes to compare into smaller pairs of processes to check. Questions about prime decomposability were first addressed by Milner and Moller in [19].

In the interleaving world, decomposition must be considered with respect to a process algebra operator, e.g. parallel composition, and the behavioural equivalence of choice: can a process term P be expressed as a process term Q of particular form, a parallel composition of prime processes, such that P and Q are bisimilar? In contrast, in true-concurrency, decomposition can be considered at the level of the semantic model: we can directly recognize whether a lats or net system can be dissected into independent 'chunks of behaviour'. Having fixed a specific decomposition view on the level of the model we can then separately investigate whether this view translates into a given equivalence. For example, we might suspect: if two parallel compositions of sequential systems, say S and S', are equivalent under a truly-concurrent bisimilarity then there is a one-to-one correspondence between the components of S and those of S' such that related components are equivalent. For classical bisimilarity this decomposition property will certainly not hold: $a||b$ is bisimilar to $a.b + b.a$.

[1] Very recently this result has been improved to O(n^3) by an algorithm that does not use decomposition in this sense [17].

There are two axioms of independence: (1) If two independent transitions can occur consecutively then they can also occur in the opposite order. (2) If two independent transitions are enabled at the same state then they can also occur one after the other. This indicates that decomposition is inherently connected to the shuffling of transitions: the behaviour of a system corresponds to the shuffle product of the behaviour of its independent components. Therefore, decomposition theorems provide an important tool to establish coincidence between hp, hhp, and chhp bisimilarity: proving that the three equivalences coincide amounts to proving that whenever two systems are hp bisimilar there exists a hp bisimulation that satifies specific shuffle properties, the hereditary and coherent condition.

The contribution of this paper is threefold: (1) We transfer the idea of prime decomposition to the truly-concurrent world. (2) We analyse whether this concept translates into truly-concurrent bisimulation equivalences. We show that, under a natural restriction, hp, and also, hhp and chhp bisimilarity are indeed decomposable with respect to prime decompositions. (3) We apply our decomposition theory to obtain coincidence results. In particular, this gives us several positive results for hhp bisimilarity, a concept which is renowned for being difficult to analyse. In more detail, after presenting the necessary definitions in Section 2, we proceed as follows.

In Section 3 we introduce the notion '*decomposition into independent components*' and a corresponding concept of *prime component* for the model of lats'; components are defined as concrete sub-systems of the respective lats. We show that every non-empty system uniquely decomposes into its set of prime components.

In Section 4 we show that hp, hhp, and chhp bisimilarity are *composable* with respect to decompositions in the following sense: assume two systems S_1, S_2, each decomposed into a set of independent components; whenever we can exhibit a one-to-one correspondence between the components of S_1 and those of S_2 such that related components are hp (hhp, chhp) bisimilar then S_1 and S_2 are hp (hhp, chhp) bisimilar. This is straightforward but guarantees the soundness of our decomposition approach. It is related to congruence in the process algebra world: if $P \sim P'$ and $Q \sim Q'$ then $P \,||\, Q \sim P' \,||\, Q'$.

Section 5 is the core of the paper: we analyse whether hp, hhp, and chhp bisimilarity are *decomposable* in the converse sense. We demonstrate that hp bisimilarity is *not* decomposable with respect to prime decompositions. However, we identify a natural restriction under which this is indeed given for hp, and also, hhp and chhp bisimilarity: for systems whose prime components are, what we shall call, *concurrent step connected* (*csc*). We obtain: whenever two *csc-decomposable systems* S_1, S_2 are hp (hhp, chhp) bisimilar then there is a one-to-one correspondence between the prime components of S_1 and those of S_2 such that related components are hp (hhp, chhp) bisimilar. The proof of this statement is non-trivial. In particular, we require the combinatorial argument of Hall's Marriage Theorem.

In Section 6 we apply our (de)composition theory to prove several coincidence results. As an immediate consequence we obtain coincidence between hp, hhp, and chhp bisimilarity for *parallel compositions of sequential systems*. Most interesting is, perhaps, that this intuitive result has turned out non-trivial to prove, and that the key insight behind it is of general significance. By employing our (de)composition theory in an inductive argument we extend the coincidence result to the class *concurrency-degree bounded communication-free net systems*.

Most importantly, we resolve that hp, hhp, and chhp bisimilarity coincide for the *simple basic parallel processes* (*SBPP*) of [7]. SBPP correspond to BPP in normal form, which in the interleaving world represent the entire BPP class; in true-concurrency they form a strictly smaller class. The coincidence for SBPP complements the positive results already achieved for (h)hp bisimilarity on BPP. Via [11] it follows that hhp bisimilarity on SBPP is polynomial-time decidable. Since hp and hhp bisimilarity do *not* coincide for BPP in general, the coincidence for SBPP underlines that SBPP and BPP do behave differently in the truly-concurrent world.

In Section 7 we conclude the paper and point to future research. Most of the proofs are kept informal in this extended abstract; a detailed account can be found in [20]. Our primary model is lats', but we also informally employ net systems, which can be understood as a class of lats'; for the definition of net systems we also refer to [20].

2 Preliminaries

Systems. A *labelled (coherent) asynchronous transition system* (for this paper simply *system*) is defined as a structure $S = (S_S, s^i_S, T_S, \rightarrow_S, I_S, l_S)$, where S_S is a set of *states* with *initial state* $s^i_S \in S_S$, T_S is the set of *transitions*[2], $\rightarrow_S \subseteq S_S \times T_S \times S_S$ is the *transition relation*, $I_S \subseteq T_S \times T_S$, the *independence relation*, is an irreflexive, symmetric relation, and $l_S : T_S \rightarrow Act$ is the *labelling function*, where $Act = \{a, b, \ldots\}$ is a set of *actions*, such that

1. $t \in T_S \implies \exists s, s' \in S_S.\ s \xrightarrow{t}_S s'$,
2. $s \xrightarrow{t}_S s' \;\&\; s \xrightarrow{t}_S s'' \implies s' = s''$,
3. $t_1 \; I_S \; t_2 \;\&\; s \xrightarrow{t_1}_S s_1 \;\&\; s_1 \xrightarrow{t_2}_S u \implies \exists s_2.\ s \xrightarrow{t_2}_S s_2 \;\&\; s_2 \xrightarrow{t_1}_S u$, and
4. $t_1 \; I_S \; t_2 \;\&\; s \xrightarrow{t_1}_S s_1 \;\&\; s \xrightarrow{t_2}_S s_2 \implies \exists u.\ s_1 \xrightarrow{t_2}_S u \;\&\; s_2 \xrightarrow{t_1}_S u$.

We lift $\rightarrow_S$ to sequences of transitions in the usual way. We also lift I_S to sequences and sets of transitions, e.g. we write $t_1 \ldots t_n \; I_S \; t'_1 \ldots t'_m$ iff $t_i \; I_S \; t'_j$ for all $i \in [1, n]$, $j \in [1, m]$. In this paper we assume a further axiom:

5. $s \in S_S \implies \exists w \in T^*_S.\ s^i_S \xrightarrow{w} s$.

Axiom (1) says that every transition can occur from some state, and axiom (2) that the occurrence of a transition at a state leads to a unique state. Axioms (3) and (4) express the two axioms of independence mentioned in the introduction. Our additional axiom (5) specifies that every state is reachable from the initial state. A system S is *finite* iff S_S and T_S are finite sets. S is *empty* iff $T_S = \emptyset$, and *non-empty* otherwise.

Let S be a system, and $s \in S_S$. The transitions of $T_c \subseteq T_S$ are *concurrently enabled* at s, $T_c \in cenabl_S(s)$, iff $\forall t \in T_c.\ \exists s'.\ s \xrightarrow{t} s'$ and $\forall t, t' \in T_c.\ t \neq t' \Rightarrow t \; I_S \; t'$. We define the *smallest upper bound on the number of transitions that are concurrently*

[2] in the sense of Petri net boxes.

enabled at s by $cbound_S(s) = \min\{\kappa \mid \forall T_c \in cenabl_S(s). |T_c| \leq \kappa\}$. S is *concurrency-degree finite* iff for each $s \in S_S$, $cbound_S(s) \in \mathbb{N}_0$. E.g., finitely branching systems are always concurrency-degree finite. **We only consider systems that are concurrency-degree finite.**

Partial Order Runs. A *pomset* is a labelled partial order; specified via a labelled strict order, it is a tuple $p = (E_p, <_p, l_p)$, where E_p is a set of *events*, $<_p$ a strict order relation on E_p, and l_p a labelling function $l_p : E_p \rightarrow Act$. A function g is an *isomorphism* between pomset p and pomset q iff $g : E_p \rightarrow E_q$ is a bijection such that (1) $l_p = l_q \circ g$, and (2) $e <_p e'$ iff $g(e) <_q g(e')$ for all $e, e' \in E_p$.

Assume a system S. Let $r = t_1 t_2 \ldots t_n \in T_S^*$ be a sequence of transitions. We write $|r|$ for the length of r, that is $|r| = n$; for any $i \in [1, |r|]$ we denote the ith transition of r, t_i, by $r[i]$. r is a *run* of S, $r \in Runs(S)$, iff $s_S^i \xrightarrow{r} s$ for some state $s \in S_S$. The *pomset* of r, $pom(r)$, has as events the integers from 1 to n, where the label of event i is $l_S(t_i)$, and the strict ordering is the transitive closure of the following "proximate cause" relation: event i *proximately causes* event j, written $i <_r^{prox} j$, iff $i < j$ and t_i and t_j are *not* independent in S. We denote this strict ordering on [1,n] by '$<_r$'.

Hp, Hhp, and Chhp Bisimilarity. Hp bisimilarity relates two systems whose behaviour can be bisimulated while preserving the labelling of transitions and the causal dependencies between them. Technically, this can be realized by basing hp bisimulation on pairs of *synchronous runs* [5]: intuitively, two runs are *synchronous* if their induced pomsets are isomorphic, and both runs correspond to the same linearization of the associated pomset isomorphism class. Formally, this amounts to: let S_1, S_2 be two systems; $r_1 \in Runs(S_1)$ and $r_2 \in Runs(S_2)$ are *synchronous*, $(r_1, r_2) \in SRuns(S_1, S_2)$, iff the identity function on $[1, |r_1|]$ is an isomorphism between $pom(r_1)$ and $pom(r_2)$. A set $\mathcal{H} \subseteq SRuns(S_1, S_2)$ is *prefix-closed* iff $(r_1 t_1, r_2 t_2) \in \mathcal{H}$ implies $(r_1, r_2) \in \mathcal{H}$. As noted in [21] it is safe to restrict our attention to prefix-closed hp bisimulations.

Hhp bisimilarity is obtained from hp bisimilarity by the addition of a *backtracking* requirement, and chhp bisimilarity furthermore imposes a *padding* requirement. These conditions reflect the first and, respectively, second axiom of independence.

Definition 1. *Let S_1 and S_2 be two systems. A* history preserving *(*hp*)* bisimulation *relating S_1 and S_2 is a prefix-closed relation $\mathcal{H} \subseteq SRuns(S_1, S_2)$ that satisfies:*

1. $(\varepsilon, \varepsilon) \in \mathcal{H}$.
2. *If $(r_1, r_2) \in \mathcal{H}$ and $r_1 t_1 \in Runs(S_1)$ for some $t_1 \in T_1$, then there is $t_2 \in T_2$ such that $(r_1 t_1, r_2 t_2) \in \mathcal{H}$.*
3. *Vice versa.*

A hp bisimulation $\mathcal{H}$ is hereditary *(*h*) when it further satisfies:*

4. *If $(r_1 t_1 w_1, r_2 t_2 w_2) \in \mathcal{H}$ for some $w_1 \in T_1^*$, $w_2 \in T_2^*$, $t_1 \in T_1$, and $t_2 \in T_2$ such that $|w_1| = |w_2|$, $t_1\ I_1\ w_1$ (or $t_2\ I_2\ w_2$ equivalently), then $(r_1 w_1, r_2 w_2) \in \mathcal{H}$.*

A hhp bisimulation $\mathcal{H}$ is coherent *(*c*) when it further satisfies:*

5. *If $(r_1 w_1, r_2 w_2)$, $(r_1 t_1, r_2 t_2) \in \mathcal{H}$ for some $w_1 \in T_1^*$, $w_2 \in T_2^*$, $t_1 \in T_1$, and $t_2 \in T_2$ such that $|w_1| = |w_2|$, $t_1\ I_1\ w_1$, and $t_2\ I_2\ w_2$, then $(r_1 t_1 w_1, r_2 t_2 w_2) \in \mathcal{H}$.*

S_1 and S_2 are ((c)h)hp bisimilar*, written $S_1 \sim_{((c)h)hp} S_2$, iff there exists a ((c)h)hp bisimulation relating them. Given two systems S_1 and S_2, we also use $\sim_{((c)h)hp}$ to denote the set $\bigcup\{\mathcal{H} : \mathcal{H}$ is a ((c)h)hp bisimulation relating S_1 and $S_2\}$. (Note: chhp bisimulations are not closed under union; so, $\sim_{chhp}$ is not necessarily the largest chhp bisimulation.)*

Further Concepts. Let A, B be alphabets. For $r \in A^*$, if $B \subseteq A$, let $r \upharpoonright B$ denote the sequence obtained by erasing from r all occurrences of letters which are not in B. If $B = T_c$ for some system c we write $r \upharpoonright c$ short for $r \upharpoonright T_c$.

The *shuffle* of n words $u_1, \ldots, u_n \in A^*$ is the set $u_1 \otimes \cdots \otimes u_n$ of all words of the form $u_{1,1}u_{2,1} \cdots u_{n,1}u_{1,2}u_{2,2} \cdots u_{n,2} \cdots u_{1,k}u_{2,k} \cdots u_{n,k}$ with $k \geq 0$, $u_{i,j} \in A^*$, such that $u_{i,1}u_{i,2} \cdots u_{i,k} = u_i$ for $1 \leq i \leq n$ [22]. We carry this notation over to pairs $(u, w) \in A^* \times B^*$ satisfying $|u| = |w|$, considering that such entities can be viewed as words in $(A \times B)^*$.

3 Decomposed Systems

We now introduce our notion of 'decomposition into independent components'. Components are defined as concrete sub-systems of the respective system.

Let S be a system. A system c is a *sub-system* of S iff

1. $S_c \subseteq S_S$,
2. $s^i_c = s^i_S$,
3. $T_c \subseteq T_S$,
4. $\rightarrow_c = \rightarrow_S \cap (S_c \times T_c \times S_c)$,
5. $I_c = I_S \cap (T_c \times T_c)$, and
6. $l_c = l_S \upharpoonright_{T_c}$.

Let c_1 and c_2 be two sub-systems of S. We say c_1 and c_2 are *independent (with respect to S)*, written $c_1 \; I_S \; c_2$, iff $T_{c_1} \; I_S \; T_{c_2}$. The *empty sub-system of S* is defined by $c^S_{empty} = (\{s^i_S\}, s^i_S, \emptyset, \emptyset, \emptyset, \emptyset)$.

Definition 2. *A* decomposition *of a system S is a set $\mathcal{D} = \{c_1, \ldots, c_n\}$, $n \in \mathbb{N}$, of sub-systems of S such that*

1. *$\forall i, j \in [1, n].\ (i \neq j \implies c_i \; I_S \; c_j)$, and*
2. *$Runs(S) = \bigcup\{r_1 \otimes \cdots \otimes r_n \mid r_i \in Runs(c_i)$ for all $i \in [1, n]\}$.*

A decomposed system *is a pair $(S, \mathcal{D})$, where $\mathcal{D}$ is a decomposition of system S.*

Every system S has at least one decomposition: the one consisting of S itself. A system may well have many different decompositions: e.g., $P = a.0 \,||\, b.0 \,||\, c.0$ can be decomposed into $\{(a.0 \,||\, b.0), c.0\}$, into $\{a.0, (b.0 \,||\, c.0)\}$, and into $\{a.0, b.0, c.0\}$. Every non-empty system will, however, uniquely decompose into a set of *prime components*.

Definition 3. *A sub-system c of a system S is a* divisor *of S iff there exists a decomposition $\mathcal{D}$ of S such that $c \in \mathcal{D}$. A system S is* prime *iff S is non-empty, and c^S_{empty} and S are the only divisors of S.*

Theorem 1. *Each non-empty system S has a unique decomposition $\mathcal{D}$ such that for all $c \in \mathcal{D}$ c is prime.*

Proof (Sketch). This can be established following the standard proof of unique prime factorization of natural numbers (see e.g. [23]). Instead of proceeding by induction on $\mathbb{N}$, we proceed by induction on the smallest upper bound on the number of transitions that can occur concurrently at the initial state. This is possible due to our restriction to concurrency-degree finite systems.

Definition 4. *We define the* prime components *of a system S, denoted by $PComps(S)$, as follows: if S is empty we set $PComps(S) = \emptyset$, otherwise we define $PComps(S)$ to be the decomposition associated with S by Theorem 1.*

Theorem 2. *Let S be a finite system. $PComps(S)$ is computable.*

Proof (Sketch). Let S be a non-empty finite system. We partition T_S into non-empty subsets such that each subset is a connected component with respect to the dependence relation (the complement of I_S). The sub-systems naturally induced by these sets of transitions are prime and together they form a decomposition of S.

Convention 1. *In the context of a decomposed system $(S, \mathcal{D})$ we use the following decomposition functions: $K : T_S \rightarrow \mathcal{D}$, defined by $K(t) = c_i \iff t \in T_{c_i}$, and $Ks : T_S^* \rightarrow \mathcal{P}(\mathcal{D})$, defined by $Ks(w) = \{K(t) \mid t \in w\}$. ($K$ is a function by clause (1) of the definition of decomposition, and the irreflexivity of independence.)*
If it is clear from the context that a system S is non-empty and there is no other decomposition specified, we understand S as the decomposed system $S = (S, PComps(S))$.

4 Composition

Hp, hhp, and chhp bisimilarity are composable with respect to decompositions in the following sense: whenever we can exhibit a one-to-one correspondence between the components of two decomposed systems such that related components are hp (hhp, chhp) bisimilar then the two systems are hp (hhp, chhp) bisimilar.

Theorem 3. *Let $x \in \{hp, hhp, chhp\}$; let $(S_1, \mathcal{D}_1)$ and $(S_2, \mathcal{D}_2)$ be two decomposed systems. If there exists a bijection $\beta : \mathcal{D}_1 \rightarrow \mathcal{D}_2$ such that $c_1 \sim_x \beta(c_1)$ for each $c_1 \in \mathcal{D}_1$ then we have $S_1 \sim_x S_2$.*

Proof (Sketch). Let $(S_1, \mathcal{D}_1)$ and $(S_2, \mathcal{D}_2)$ be two decomposed systems. Assume we are given a bijection $\beta : \mathcal{D}_1 \rightarrow \mathcal{D}_2$, say $\beta = \{(c_1^1, c_2^1), \ldots, (c_1^n, c_2^n)\}$, and a family $\{\mathcal{H}^i\}_{i=1}^n$ such that for all $i \in [1, n]$ $\mathcal{H}^i$ is a hp bisimulation relating c_1^i and c_2^i. We define $\mathcal{H} = \bigcup\{r^1 \otimes \cdots \otimes r^n \mid r^i \in \mathcal{H}^i \text{ for all } i \in [1, n]\}$. It is straightforward to check that $\mathcal{H}$ is a hp bisimulation relating S_1 and S_2. Furthermore, it is routine to establish: if for all $i \in [1, n]$ $\mathcal{H}^i$ is hereditary then $\mathcal{H}$ will also be hereditary; if for all $i \in [1, n]$ $\mathcal{H}^i$ is coherent then $\mathcal{H}$ will also be coherent.

5 Decomposition

It is trivial that hp, hhp, and chhp bisimilarity are *not* decomposable in the converse sense: as we saw $P = a.0 \,||\, b.0 \,||\, c.0$ can be decomposed into $\{(a.0 \,||\, b.0), c.0\}$ and also into $\{a.0, (b.0 \,||\, c.0)\}$; but certainly we cannot exhibit a bijection between the two decompositions such that related components are bisimilar. The more natural question to ask is whether a notion of equivalence is decomposable with respect to *prime decompositions*.

The example of Figure 1 demonstrates hp bisimilarity is *not* decomposable in this sense, either. On the one hand, A and B are hp bisimilar. The additional transition b_3' in B can easily be hidden by adopting the following strategy: if b_3' occurs as the first transition we will match it against b_1. Then in both systems 'parallel b' is the only remaining behaviour, and b_1' can safely be matched by b_2. If we start out with b_1' we will match it against b_1. Then the a-transition is disabled in both systems, and this time it will be safe to match b_3' by b_2. On the other hand, a bijection between the prime components of A and those of B can clearly not be found.

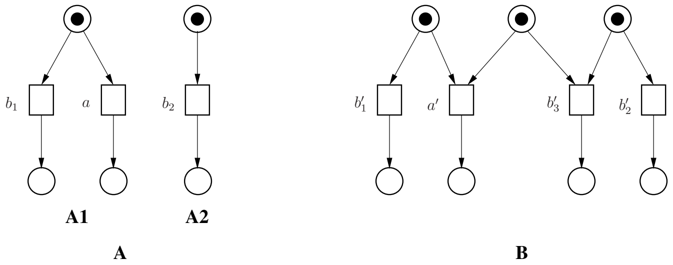

Fig. 1. The transitions of A and B are labelled as their names suggest: e.g. $l(b_1') = b$. A consists of two prime components: A_1 and A_2; B has only one prime component: B itself

A and B are *not* (c)hhp bisimilar: at $(b_1 b_2, b_1' b_3')$ we can backtrack (b_1, b_1'); then the a-transition becomes available in A but not in B. In Section 7 we will briefly discuss whether (c)hhp bisimilarity may be decomposable with respect to prime decompositions. Here we want to analyse whether there are conditions under which we do obtain decomposition for hp bisimilarity; this is important with respect to establishing coincidence results. We will find that, on systems whose prime components are, what we shall call, *concurrent step connected* (*csc*), hp, and also hhp and chhp, bisimilarity are indeed decomposable with respect to prime decompositions: whenever two *csc-decomposable systems* are hp (hhp, chhp) bisimilar then there is a one-to-one correspondence between their prime components such that related components are hp (hhp, chhp) bisimilar.

We start out by explaining two special types of runs, which will play a key role in the proof. A run r is a *concurrent step* iff all the transitions on r occur independently of each other. A run r is *maximal with respect to initial concurrency* iff whenever a further transition t is executed at r, t will occur causally dependent on some transition on r.

Definition 5. *Let S be a system, and $r \in \mathit{Runs}(S)$.*
r is a concurrent step *of S, written $r \in \mathit{csteps}(S)$, iff we have:*
$\forall k, l \in [1, |r|].\ (k \neq l \Rightarrow r[k]\ I_S\ r[l])$*.*
r is maximal with respect to initial concurrency*, written $r \in \mathit{icmax}(S)$, iff we have:*
$\forall t \in T_S.\ (rt \in \mathit{Runs}(S) \Rightarrow \exists i \in [1, |r|].\ i <_{rt} |rt|)$*.*

Clearly, in pairs of synchronous runs, and hence in hp bisimilarity, concurrent steps are always matched against concurrent steps.

Fact 1. *Let S_1 and S_2 be two systems. For all $(r_1, r_2) \in \mathit{SRuns}(S_1, S_2)$ we have: $r_1 \in \mathit{csteps}(S_1) \iff r_2 \in \mathit{csteps}(S_2)$.*

With the concept 'maximal with respect to initial concurrency' it is easy to identify a scenario which, given two decomposed systems $(S_1, \mathcal{D}_1)$, $(S_2, \mathcal{D}_2)$, allows us to infer that two components $c_1 \in \mathcal{D}_1$, $c_2 \in \mathcal{D}_2$ are hp (hhp, chhp) bisimilar:

Lemma 1. *Let $x \in \{hp, hhp, chhp\}$; let $(S_1, \mathcal{D}_1)$, $(S_2, \mathcal{D}_2)$ be two decomposed systems. For any pair $c_1 \in \mathcal{D}_1$, $c_2 \in \mathcal{D}_2$ we have: if there exists $(r_1, r_2) \in \sim_x$ such that for $i = 1$, and 2 $\left\{ \begin{array}{l} c_i \notin Ks(r_i), \text{ and} \\ \forall c'_i \in \mathcal{D}_i \backslash c_i.\ r_i \!\uparrow\! c'_i \in \mathit{icmax}(c'_i) \end{array} \right\}$ then $c_1 \sim_x c_2$.*

Proof (Sketch). Given entities as above, we can extract a hp (hhp, chhp) bisimulation relating c_1 and c_2 from any hp (hhp, chhp) bisimulation containing (r_1, r_2). This is so because: (1) the full behaviour of c_1 and c_2 has still to be matched at (r_1, r_2), and (2) the causal dependencies will force that behaviour of c_1 has to be matched against behaviour of c_2, and vice versa.

From the example of Figure 1 it is clear that, given two hp bisimilar systems, we may never be in a position to apply this lemma. A and B are hp bisimilar but there is no $(r_1, r_2) \in \sim_{hp}$ such that, via Lemma 1, we can deduce $c_1 \sim_{hp} c_2$ for any $c_1 \in \mathit{PComps}(A)$, $c_2 \in \mathit{PComps}(B)$: if B, the only prime component of B, is not contained in $Ks(r_1)$ then $(r_1, r_2) = (\varepsilon, \varepsilon)$; but we neither have $\varepsilon \in \mathit{icmax}(A_1)$ nor $\varepsilon \in \mathit{icmax}(A_2)$.

The scenario of Lemma 1 will, however, certainly be available if for the system class under study we can show: the matching in hp bisimilarity respects prime components in that: let $(r_1, r_2) \in \sim_{hp}$; if, in (r_1, r_2), a transition of prime component c_1 is matched to a transition of prime component c_2, then, in (r_1, r_2), any other transition of c_1 is also matched to a transition of c_2, and vice versa. Then, given $(r_1, r_2) \in \sim_{hp}$, r_1 is 'maximal with respect to initial concurrency' for all but one prime component c_1 such that $c_1 \notin Ks(r_1)$ iff the analogue is true for r_2. On second thought, to guarantee the applicability of Lemma 1 it is sufficient to obtain that the matching of *concurrent steps* (rather than the matching of *all* runs) respects prime components: concurrent steps can be seen as the minimum to consider when we want to achieve maximality with respect to initial concurrency.

We now identify a system class, as large as intuitively possible, which naturally satisfies this criteria: *csc-decomposable systems*. They have the following characteristic: each of their prime components is *cstep connected* (*csc*) in that: whenever we have computed a concurrent step r and we compute one further concurrently enabled transition

t then there is the possibility of computing a sequence of transitions w such that the last transition of w is causally dependent on t and some transition of r. In short we may say: every concurrent step has a causal link with any further concurrently enabled transition.

Definition 6. *Let S be a system.*
Let $r \in Runs(S)$, and $k, l \in [1, |r|]$. $w \in T_S^+$ is a causal link at r between the events k and l*, denoted by $w \in clinks_S(r, k, l)$, iff we have:*
$rw \in Runs(S)$ & $k <_{rw} |rw|$ & $l <_{rw} |rw|$.
S is cstep connected *(*csc*) iff for all $r \in csteps(S)$ with $|r| \geq 1$ we have:*
$\forall t \in T_S.\ (rt \in csteps(S) \Rightarrow \exists k \in [1, |r|].\ \exists w \in T_S^+.\ w \in clinks_S(rt, k, |rt|))$.
S is csc-decomposable *iff every prime component of S is csc. (Note that non-empty csc systems are always prime.)*

Example 1. Consider Figure 1. B is not csc: we can do b_1', and then b_3', but there is no causal link between b_1' and b_3'. *Sequential systems* ($\neg(\exists s, s', t, t'.\ t\ I_S\ t'\ \&\ s \xrightarrow{tt'}_S s')$), such as A_1 and A_2, and *initially sequential systems* ($\forall r \in csteps(S).\ |r| \leq 1$) are trivially csc.

Lemma 2. *Let S_1 and S_2 be two csc-decomposable systems. For all $(r_1, r_2) \in \sim_{hp}$ such that $r_i \in csteps(S_i)$ for $i = 1$, or 2 equivalently (Fact 1), we have:*

$$\forall k, l \in [1, |r_1|].\ (K(r_1[k]) = K(r_1[l]) \iff K(r_2[k]) = K(r_2[l])).$$

Proof (Sketch). We proceed by induction on the length of two related concurrent steps. Let (r_1, r_2) be given as above. Assume, in (r_1, r_2), a transition of prime component c_1 is matched to a transition of prime component c_2, and we want to match a further concurrently enabled c_1-transition, t_1. There will be a causal link at r_1t_1 between event $|r_1t_1|$ and one of the previously matched c_1-events. By induction hypothesis we can assume these are all matched by c_2-events. But then t_1 has to be matched by a c_2-transition: otherwise the causal link could not be matched in a partial order preserving fashion.

It is routine to derive the following corollaries:

Corollary 1. *Let S_1 and S_2 be two csc-decomposable systems.*

1. *For all $(r_1, r_2) \in \sim_{hp}$ such that $r_i \in csteps(S_i)$ for $i = 1$, or 2 equivalently (Fact 1), we have: $|Ks(r_1)| = |Ks(r_2)|$.*
2. *If $S_1 \sim_{hp} S_2$ then $|PComps(S_1)| = |PComps(S_2)|$.*

Corollary 2. *Let S_1 and S_2 be two csc-decomposable systems, and let $(r_1, r_2) \in \sim_{hp}$ such that $r_i \in csteps(S_i)$ for $i = 1$, or 2 equivalently (Fact 1). For any pair of components $c_1 \in PComps(S_1)$, $c_2 \in PComps(S_2)$ such that $K(r_1[k]) = c_1$ and $K(r_2[k]) = c_2$ for some $k \in [1, |r_1|]$ we have: $r_1 \upharpoonright c_1 \in icmax(c_1) \iff r_2 \upharpoonright c_2 \in icmax(c_2)$.*

For hhp and chhp bisimilarity there is now a simple argument that proves, for csc-decomposable systems, the two bisimilarities are indeed decomposable with respect to prime decompositions (c.f. [20]). This argument relies on backtracking; considering hp bisimilarity it is only obvious that, given two csc-decomposable systems S_1, S_2 with $S_1 \sim_{hp} S_2$, a bijection between $PComps(S_1)$ and $PComps(S_2)$ exists, and further, for each $c_1 \in PComps(S_1)$ there is $c_2 \in PComps(S_2)$ such that $c_1 \sim_{hp} c_2$, and vice versa. To prove decomposition for hp bisimilarity we need something more sophisticated: the combinatorial argument of Hall's Marriage Theorem (e.g. see [24]).

Theorem 4. *Let $x \in \{hp, hhp, chhp\}$; let S_1, S_2 be two csc-decomposable systems. If $S_1 \sim_x S_2$ then there exists a bijection $\beta : PComps(S_1) \to PComps(S_2)$ between the prime components of S_1 and those of S_2 such that $c_1 \sim_x \beta(c_1)$ for each $c_1 \in PComps(S_1)$.*

Proof. Let x, S_1, S_2 be given as above, and assume $S_1 \sim_x S_2$. We shall prove that a bijection β exists as required. By Corollary 1(2) we have (A) $|PComps(S_1)| = |PComps(S_2)|$, and it only remains to show that an injective map can be found. For each $c_1 \in PComps(S_1)$ let $C_{2_{c_1}}$ be the set of prime components of S_2 which are x bisimilar to c_1. By Hall's Marriage Theorem the required injection exists if and only if the following condition is fulfilled:

$$(*) \quad \forall C_1 \subseteq PComps(S_1).\, |\bigcup_{c_1 \in C_1} C_{2_{c_1}}| \geq |C_1|.$$

Choose an arbitrary subset C_1 of $PComps(S_1)$. Let $\bar{C}_1 = PComps(S_1) \backslash C_1$, and consider $r_1 \in csteps(S_1)$ such that (B) $Ks(r_1) = \bar{C}_1$, and $\forall c_1 \in \bar{C}_1.\ r_1 \uparrow c_1 \in icmax(c_1)$; this is clearly possible. There must be r_2 such that $(r_1, r_2) \in \sim_x$; set $\bar{C}_2 = Ks(r_2)$, and $C_2 = PComps(S_2) \backslash \bar{C}_2$. By Corollary 2 we obtain $\forall c_2 \in \bar{C}_2.\ r_2 \uparrow c_2 \in icmax(c_2)$. On the other hand, (B) and Corollary 1(1) give us $|\bar{C}_1| = |\bar{C}_2|$, and considering (A) we gain (C) $|C_1| = |C_2|$. Next we show that for each remaining component $c_2 \in C_2$ there is a component $c_1 \in C_1$ such that $c_1 \sim_x c_2$. With (C) this will immediately establish $|\bigcup_{c_1 \in C_1} C_{2_{c_1}}| \geq |C_1|$, and thereby $(*)$.

Assume C_2 is non-empty, and choose any $c_2 \in C_2$. Consider r'_2 such that $r_2 r'_2 \in csteps(S_2)$, $Ks(r'_2) = C_2 \backslash c_2$, and $\forall c'_2 \in C_2 \backslash c_2.\ r'_2 \uparrow c'_2 \in icmax(c'_2)$; this is clearly possible. Note that altogether we have (D) $Ks(r_2 r'_2) = PComps(S_2) \backslash c_2$, and $\forall c'_2 \in PComps(S_2) \backslash c_2.\ r_2 r'_2 \uparrow c'_2 \in icmax(c'_2)$. There must be r'_1 such that $(r_1 r'_1, r_2 r'_2) \in \sim_x$. Corollary 1(1) gives us $|Ks(r_1 r'_1)| = |Ks(r_2 r'_2)|$, and by (D), (A), and (B) this implies $Ks(r_1 r'_1) = PComps(S_1) \backslash c_1$ for some $c_1 \in C_1$. By Corollary 2 we obtain $\forall c'_1 \in PComps(S_1) \backslash c_1.\ r_1 r'_1 \uparrow c'_1 \in icmax(c'_1)$. But altogether this means we can apply Lemma 1 to infer $c_1 \sim_x c_2$. Thus, c_1 provides a component exactly as required.

6 Coincidence Results

We now apply our composition and decomposition theory to prove several coincidence results on hp, hhp, and chhp bisimilarity. First of all, our theory gives us a general proof technique: whenever we consider whether (any two of) the three equivalences coincide for a class of csc-decomposable systems, we can restrict our attention to the respective class of prime components. This is immediate by the following argument:

Argument 1. Assume two csc-decomposable systems S_1 and S_2 that are hp bisimilar. By Theorem 4(hp) we obtain a bijection between the prime components of S_1 and those of S_2 such that related components are hp bisimilar. Then, provided that hp, hhp, and chhp bisimilarity coincide for the class of the prime components, by Theorem 3(chhp) we can conclude that S_1 and S_2 are chhp (and thus also hhp) bisimilar.

It is folklore that for sequential systems hp, hhp, and chhp bisimilarity all coincide with classical bisimilarity (e.g. see [13]). Furthermore, we have already mentioned that sequential systems are csc. Then, with the previous argument we obtain:

Theorem 5. *Hp, hhp, and chhp bisimilarity coincide for parallel compositions of sequential systems.* (Formally, a parallel composition of sequential systems is a system which can be decomposed into sequential components.)

Consider the following generalization of the class 'parallel compositions of sequential systems': each system S is a parallel composition of *initially sequential* components such that each component may, by performing a transition, fork into a parallel composition of initially sequential sub-components, each of which may in turn evolve into a parallel composition of initially sequential sub-components, and so on; this description is complete in that we do not allow any communication between parallel threads. This system class is best known as, and most conveniently captured by, *communication-free net systems*[3]. (Formally, a net system $\mathcal{N}$ is *communication-free* iff $\forall t \in T_N.\ |^\bullet t| = 1$.)

If a communication-free net system S is *concurrency-degree bounded* in that the smallest upper bound on the number of transitions that can be concurrently enabled in S with respect to any state, $cbound(S)$, is given by a natural number, then, for each proper component c of S, $cbound(c)$ will be strictly smaller than $cbound(S)$. With Argument 1 we then obtain coincidence for concurrency-degree bounded communication-free net systems by induction on $cbound(S)$.

Definition 7. *Let S be a system. The* smallest upper bound on the number of transitions that can be concurrently enabled in S with respect to any state*, $cbound(S)$, is defined by* $\max\{cbound_S(s) \mid s \in S_S\}$*. S is* councurrency-degree bounded *iff $cbound(S) \in \mathbb{N}_0$.*

Theorem 6. *Two councurrency-degree bounded communication-free net systems are hp bisimilar iff they are hhp bisimilar iff they are chhp bisimilar.*

By translating Argument 1 into a tableau system, we achieve coincidence for *simple basic parallel processes* (*SBPP*). These can be interpreted as an orthogonal class of communication-free net systems[3]: we lift the restriction to concurrency-degree bounded systems, but require our systems to be finitely representable. Following [7], SBPP are defined by process expressions of the grammar: $E ::= S \mid E \,||\, E$, where '$||$' is parallel composition and S is an *initially sequential process* expression given by: $S ::= \mathbf{0} \mid a.E \mid S + S \mid X$, where $\mathbf{0}$ is the empty process, $a.E$, where $a \in Act$, is action prefix, '$+$' is nondeterministic choice, and X is an 'initially sequential process' variable. Every SBPP can effectively be transformed into a chhp bisimilar *SBPP in normal form*.

[3] As their unfoldings communication-free net systems also capture the class of communication-free weighted Petri nets.

Definition 8. *Let* $\mathit{Vars} = \{X_1, X_2, \ldots\}$ *be a set of process variables, and* $\mathit{Vars}^{\otimes} = \{\alpha, \beta, \ldots\}$ *the set of finite multisets over* Vars*. We identify* $\alpha = \{X, X, Y\}$ *with the parallel composition* $X \,||\, X \,||\, Y$*; the empty multiset is recognized as the process* $\mathbf{0}$.
A SBPP in normal form *is a pair* $\mathcal{E} = (E_0, \Delta_{\mathcal{E}})$*, where* $E_0 \in \mathit{Vars}^{\otimes}$*, and* $\Delta_{\mathcal{E}}$ *is a finite family of recursive equations* $\{X_i := E_i \mid 1 \leq i \leq m\}$*. The* X_i *are distinct, and the* E_i *are of the form:* $a_1.\alpha_1 + a_2.\alpha_2 + \ldots + a_n.\alpha_n$*, where* $n \geq 1$*, and* $\forall i \in [1, n]$. $\alpha_i \in \mathit{Vars}^{\otimes}$*. Further,* $\forall i \in [0, m]$*,* E_i *at most contains the variables* $\{X_1, \ldots, X_m\}$.

Theorem 7. *Two SBPP are hp bisimilar iff they are hhp bisimilar iff they are chhp bisimilar.*

Proof (Sketch). The tableau proof system of Figure 2 gives rise to a decision procedure that decides whether two SBPP in normal form are hp bisimilar, and at the same time, whether they are chhp bisimilar. Rule **Match** provides matching for initially sequential processes; rule **Decomp** reflects our decomposition theory, and provides the means to reduce pairs of processes to check into smaller pairs of processes to compare. Theorem 4(hp) implies forward soundness of **Decomp** for hp bisimilarity, Theorem 3(chhp) gives us backwards soundness of **Decomp** for chhp bisimilarity. Finiteness, completeness for hp bisimilarity, and soundness for chhp bisimilarity of the tableau system can then be proved by using the standard arguments.

Rec $$\frac{X = Y}{E = F} \qquad \text{where } (X := E) \in \Delta_{\mathcal{E}}, (Y := F) \in \Delta_{\mathcal{F}}$$

Match $$\frac{\sum_{i=1}^{n} a_i.\alpha_i = \sum_{j=1}^{m} b_j.\beta_j}{\{\alpha_i = \beta_{f(i)}\}_{i=1}^{n} \qquad \{\alpha_{g(j)} = \beta_j\}_{j=1}^{m}}$$

where $f : [1, n] \to [1, m]$, $g : [1, m] \to [1, n]$ are functions such that $\forall i \in [1, n].\ a_i = b_{f(i)}$, and similarly for g.

Decomp $$\frac{\alpha = \beta}{\{X = Y\}_{(X,Y) \in b}} \qquad \text{where } b : \alpha \to \beta \text{ is a bijection (relating variable instances).}$$

A node n is a *successful terminal* iff

n: $\mathbf{0} = \mathbf{0}$, or

n: $X = Y$, and there is a node n_a: $X = Y$ above n in the tableau.

A node n is an *unsuccessful terminal* iff

n: $\alpha = \beta$, and a bijection b as required by rule **Decomp** does not exist, or

n: $\sum_{i=1}^{n} a_i.\alpha_i = \sum_{j=1}^{m} b_j.\beta_j$, and f and g as required by rule **Match** do not exist.

Fig. 2. A tableau system with respect to two SBPP in normal form $\mathcal{E}$ and $\mathcal{F}$

7 Conclusions

There are further applications of our decomposition theory. In analogy to Argument 1 decidability of hp (hhp, chhp) bisimilarity on a class of finite-state csc-decomposable systems reduces to decidability on the respective class of prime components (recall Theorem 2). Further, if a system is specified in terms of csc components, our decomposition theory is profitable with respect to tackling the state explosion problem: we do not need to check hp (hhp, chhp) bisimilarity on the global state space but we can proceed by checking the respective equivalence on pairs of components.

One might speculate that (c)hhp bisimilarity is decomposable with respect to prime decompositions for systems in general: with the help of backtracking one might be able to prove a general version of Lemma 2; though this may be hard, or at least technically tedious, to carry through. Furthermore, as pointed out to me by Lasota, in the formulation of a general version of Lemma 2 and the decomposition theorem, one will have to address the issue of (c)hhp bisimilar choices: let $P = (P_1 \,||\, P_2) + (P_1 \,||\, P_2)$ and $Q = P_1 \,||\, P_2$; clearly $P \sim_{(c)hhp} Q$ but since P is prime there is no bijection between the prime components of P and those of Q.

It is, of course, also possible to investigate whether a truly-concurrent equivalence satisfies the unique decomposition property usually investigated in the interleaving setting. (Given some class of process terms, is each of them uniquely, up to the equivalence, represented as a parallel composition of primes?) Indeed, unique decomposition with respect to distributed bisimilarity has been proved for BPP [25]. Note, however, that decomposition in this sense is *not* sufficient to establish the results of Section 6.

We hope this paper motivates the particular significance of composition and decomposition for true-concurrency: decomposition characteristics of a system class may translate into truly-concurrent equivalences or logics in a very concrete way, and thereby lead us to decision procedures and/or coincidence results. In this spirit, the ideas of the paper can be taken further: one could investigate whether a similar approach is possible with respect to temporal logics, and, orthogonally, whether our decomposition theory can be generalized by integrating a concept of synchronization. Indeed, the latter idea stands behind the result that (c)hhp bisimilarity is decidable for a class of live free-choice systems [13]. This is so far the only positive result on hhp bisimilarity for a class that admits a flexible form of synchronization. ([26] presents that hhp bisimilarity is decidable for trace-labelled systems but the proof turned out to be incomplete [15].)

Acknowledgements. I would like to thank Walter Vogler: he has provided crucial help by pointing out to me that Hall's Marriage Theorem has to be applied in the proof of an earlier version of Theorem 4. I thank Javier Esparza, Mogens Nielsen, Damian Niwinski, and the anonymous referees for their valuable comments on this work. I thank Monika Maidl, who has helped to clarify a question related to Theorem 2. Finally, I would like to thank Slawomir Lasota for pointing out to me the issue of (c)hhp bisimilar choices.

References

1. Penczek, W., Kuiper, R.: Traces and logic. In: The Book of Traces. World Scientific (1995) 307–381
2. Madhusudan, P., Thiagarajan, P.S.: Controllers for discrete event systems via morphisms. In: CONCUR'98. Volume 1466 of LNCS. (1998) 18–33

3. Jurdziński, M., Nielsen, M., Srba, J.: Domino hereditary history preserving bisimilarity is undecidable. Inform. and Comput. **184** (2003) 343–368
4. Vogler, W.: Deciding history preserving bisimilarity. In: ICALP'91. Volume 510 of LNCS. (1991) 495–505
5. Jategaonkar, L., Meyer, A.R.: Deciding true concurrency equivalences on safe, finite nets. TCS **154** (1996) 107–143
6. Montanari, U., Pistore, M.: Minimal transition systems for history-preserving bisimulation. In: STACS'97. Volume 1200 of LNCS. (1997) 413–425
7. Esparza, J., Kiehn, A.: On the model checking problem for branching time logics and basic parallel processes. In: CAV'95. Volume 939 of LNCS. (1995) 353–366
8. Sunesen, K., Nielsen, M.: Behavioural equivalence for infinite systems—partially decidable! In: ICATPN'96. Volume 1091 of LNCS. (1996) 460–479
9. Jančar, P.: Strong bisimilarity on basic parallel processes is PSPACE-complete. In: LICS'03, IEEE (2003) 216–??.
10. Srba, J.: Strong bisimilarity and regularity of basic parallel processes is PSPACE-hard. In: STACS'02. Volume 2285 of LNCS. (2002) 535–546
11. Lasota, S.: A polynomial-time algorithm for deciding true concurrency equivalences of basic parallel processes. In: MFCS'03. Volume 2747 of LNCS. (2003) 521–530
12. Fröschle, S.: Decidability of plain and hereditary history-preserving bisimulation for BPP. In: EXPRESS'99. Volume 27 of ENTCS. (1999)
13. Fröschle, S.: Decidability and Coincidence of Equivalences for Concurrency. PhD thesis, University of Edinburgh (2004)
14. Winskel, G., Nielsen, M.: Models for concurrency. In: Handbook of logic in computer science, Vol. 4. Oxford Univ. Press (1995) 1–148
15. Fröschle, S.: The decidability border of hereditary history preserving bisimilarity. Information Processing Letters (to appear)
16. Hirshfeld, Y., Jerrum, M., Moller, F.: A polynomial-time algorithm for deciding bisimulation equivalence of normed basic parallel processes. Mathematical Structures in Computer Science **6** (1996) 251–259
17. Jančar, P., Kot, M.: Bisimilarity on normed basic parallel processes can be decided in time o(n^3). In: AVIS'04. ENTCS (2004)
18. Christensen, S., Hirshfeld, Y., Moller, F.: Decomposability, decidability and axiomatisability for bisimulation equivalence on basic parallel processes. In: LICS'93, IEEE (1993) 386–396
19. Milner, R., Moller, F.: Unique decomposition of processes. TCS **107** (1993) 357–363
20. Fröschle, S.: Composition and decomposition in true-concurrency. Technical Report 276, Institute of Informatics, University of Warsaw (2004)
21. Fröschle, S., Hildebrandt, T.: On plain and hereditary history-preserving bisimulation. In: MFCS'99. Volume 1672 of LNCS. (1999) 354–365
22. Pin, J.E.: Syntactic semigroups. In: Handbook of formal languages, Vol. 1. Springer (1997) 680–746
23. Norman, C.W.: Undergraduate Algebra. Oxford Science Publications (1986)
24. Truss, J.K.: Discrete Mathematics for Computer Scientists. Addison-Wesley (1991)
25. Christensen, S.: Decidability and Decomposition in Process Algebras. PhD thesis, University of Edinburgh (1993)
26. Mukund, M.: Hereditary history preserving bisimulation is decidable for trace-labelled systems. In: FST TCS'02. Volume 2556 of LNCS. (2002) 289–300