Everything Is PSPACE-Complete in Interaction Systems

Mila Majster-Cederbaum and Christoph Minnameier*

Institut für Informatik
Universität Mannheim, Germany
cmm@informatik.uni-mannheim.de

Abstract. We study complexity issues for interaction systems, a general model for component-based systems that allows for a very flexible interaction mechanism. We present complexity results for important properties of interaction systems such as local/global deadlock-freedom, progress and availability of components.

1 Introduction

First introduced by Sifakis et al. [GS03], *interaction systems* are a general model for component-based systems. Its main features can be summarized as follows. The description of a component is hidden to any other component, in particular a component does not refer to methods or operations of other components. Components offer ports for cooperation with other components. Components are put together by some kind of a (separate) gluing mechanism in a such way that the identity of each component is maintained. Components and the glue can thus be modified freely. The gluing is realized via connectors, that consist of ports of various components. Connectors can be of different size and each port can participate in more than one connector.

The model has been discussed in [GS03, Sif05, GS05, GGM$^+$07]. In [GQ07] the model has been enriched by hierarchical connectors. A version including variables and value passing was implemented in the BIP-project [BBS06] and in the Prometheus-project [Goe06] and was used to implement and study a variety of component-based systems. The relevance of the model is also stressed by the fact that it is used as a common semantic framework in the European SPEEDS-project [GO07].

Interaction systems can be viewed as a generalization of interface automata [dAH01] as well as of input/output automata [LT89].

Given that interaction systems are a suitable and comfortable framework to model component-based systems it is interesting to investigate their properties.

Here, we study algorithmic properties of interaction systems, in particular reachability, local and global deadlock-freedom, progress and availability of components. These properties are defined on the global state space which is exponentially large in the number of components. We show that deciding either of

* Corresponding author.

J.S. Fitzgerald, A.E. Haxthausen, and H. Yenigun (Eds.): ICTAC 2008, LNCS 5160, pp. 216–227, 2008.

the mentioned properties is PSPACE-complete. To do so we build on a connection between interaction systems and 1-safe Petri nets that was first presented in [MM08b] and yields PSPACE-hardness for reachability in interaction systems.

This paper is organized as follows. In Section 2 we give the basic definitions for interaction systems and the various properties we are going to discuss and point out the advantages of interaction systems over the closely-related models of 1-safe Petri nets. Section 3 presents the polynomial reductions that show that all discussed behavioral questions are PSPACE-complete. In Section 4 we give a short conclusion and discuss related work.

2 Interaction Systems

2.1 Syntax and Semantics

An ***interaction system*** is a tuple $Sys = (K, \{A_i\}_{i\in K}, C, Comp, \{T_i\}_{i\in K})$, where K is the set of ***components***. Often, we assume $K = \{1, \ldots, n\}$.

Each component $i \in K$ offers a finite, nonempty set of ***ports*** (resp. ***actions***) A_i for cooperation with other components. The ***port sets*** A_i are pairwise disjoint.

Cooperation is described by connectors and complete interactions. A ***connector*** is a finite, nonempty set of actions $c \subseteq \bigcup_{i\in K} A_i$, subject to the constraint that for each component i at most one action $a_i \in A_i$ is in c. A connector $c = \{a_{i_1}, \ldots, a_{i_k}\}$ with $a_{i_j} \in A_{i_j}$ describes the fact that the components $i_1, \ldots, i_j$ may cooperate via these ports. A ***connector set*** C is a finite set of connectors, s.t. every action of every component occurs in at least one connector and no connector contains any other connector. We define the set of **interactions** $Int := \{\alpha \mid \exists c \in C, \text{ s.t. } \alpha \subseteq c\}$. In some cases we want to allow that a connector is performed only partially, e.g. if not all components involved in a connector are ready to perform their respective action. For this we may designate certain interactions as ***complete interactions***. Let $Comp \subseteq Int$ be a designated set of complete interactions. $Comp$ has to be upwards-closed w.r.t. C, i.e. $\forall \alpha \in Comp\, \forall \alpha' \in Int : ((\alpha \subset \alpha') \Rightarrow \alpha' \in Comp)$.

The local behavior of each component i is described by $T_i = (Q_i, A_i, \rightarrow_i, q_i^0)$, where Q_i is a finite set of local states, $\rightarrow_i \subseteq Q_i \times A_i \times Q_i$ the local transition relation and $q_i^0 \in Q_i$ is the local starting state. We assume that the T_i's are non-terminating, i.e. each $q_i \in Q_i$ has at least one outgoing edge.

We call the class of all interaction systems ***IS***.

The ***global behavior*** $T_{Sys} = (Q, C \cup Comp, \rightarrow_{Sys}, q^0)$ of Sys (henceforth also referred to as the global transition system) is obtained from the behaviors of the individual components, given by the transition systems T_i, and the interactions in $C \cup Comp$ in a straightforward manner:

- $Q := \prod_{i\in K} Q_i$, the Cartesian product of the Q_i, which we consider to be order independent. We denote states by tuples $(q_1, \ldots, q_n)$ and call them global states.

– the relation $\rightarrow_{Sys} \subseteq Q \times (C \cup Comp) \times Q$, defined by $\forall \alpha \in (C \cup Comp)\ \forall q, q' \in Q \quad q = (q_1, \ldots, q_n) \xrightarrow{\alpha}_{Sys} q' = (q'_1, \ldots, q'_n)$ iff $\forall i \in K \quad (q_i \xrightarrow{a_i}_i q'_i$ if $\alpha \cap A_i = \{a_i\}$ and $q'_i = q_i$ otherwise).
– $q^0 := (q^0_1, \ldots, q^0_n)$ is the starting state for *Sys*.

Less formally, a transition labeled by α may take place in the global transition system when all ports occuring in α are offered by the respective components.

Example 1. The following interaction system $Count_{(3,4)}$ demonstrates the capability of interaction systems to synchronize with different numbers of participants. $Count_{(3,4)} = (\{1, 2, 3, 4\}, \{A_i\}_{1 \leq i \leq 4}, C, Comp, \{T_i\}_{1 \leq i \leq 4})$, where
$A_i = \{inc_i, dec_i\}$ $(1 \leq i \leq 3)$, $A_4 = \{inc_4, dummy_4\}$,
$C = \{\{inc_1, dummy_4\}, \{inc_2, dec_1\}, \{inc_3, dec_2, dec_1\}, \{inc_4, dec_3, dec_2, dec_1\}\}$,
$Comp = \{\{inc_1\}, \{inc_1, dummy_4\}\}$, and the T_i's are given in Figure 1.

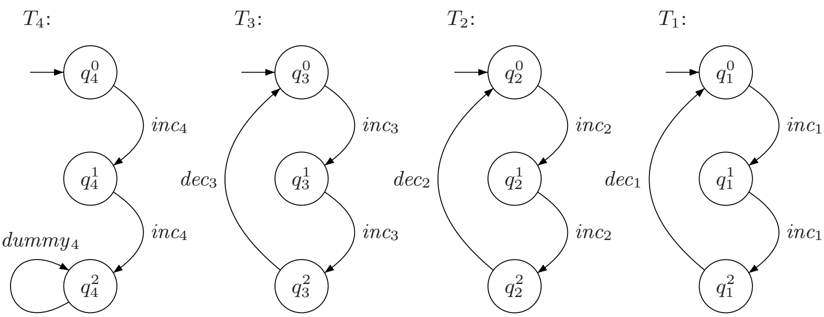

Fig. 1. The local transition systems for $Count_{(3,4)}$

The behavior[1] of our example system $Count_{(3,4)}$ is as follows: It performs a deterministic computation starting in $(q^0_1, q^0_2, q^0_3, q^0_4)$. The system describes a 3^4-counter that counts from 0 to 3^4-1=80 and then cannot perform any further interaction.

We refer to the local transition system T_i of a component i of some previously defined system *Sys* by $T_i[Sys]$. The same notation is used for the other elements of the interaction system tuple. E.g. $Comp[Count_{(3,4)}] = \{\{inc_1\}, \{inc_1, dummy_4\}\}$. Whenever it is obvious by the context to which system we refer (as e.g. in the next subsection), we may simply write Q instead of $Q[Sys]$, etc. for ease of notation.

2.2 Properties of Interaction Systems

For the following definitions let $Sys \in IS$:

– For local states $q_i \in Q_i$ we define the set of ***enabled actions*** $ea(q_i) := \{a_i \in A_i \mid \exists q'_i \in Q_i, \text{ s.t. } q_i \xrightarrow{a_i}_i q'_i\}$.

[1] Note that $dummy_4$ is introduced only to ensure that T_4 is non-terminating.

- For $q, q' \in Q$ we say that **there is a path from q to q'** if $q = q'$ or $\exists k \in \mathbb{N}_0\ \exists q^1, \ldots, q^k \in Q\ \exists \alpha_0, \ldots, \alpha_k \in (C \cup Comp)$ s.t. $q \xrightarrow{\alpha_0}_{Sys} q^1 \xrightarrow{\alpha_1}_{Sys} \ldots \xrightarrow{\alpha_{k-1}}_{Sys} q^k \xrightarrow{\alpha_k}_{Sys} q'$. Such a transition sequence is called a path ϕ from q to q'.
- We call an infinite transition sequence $q \xrightarrow{\alpha_0}_{Sys} \ldots$ **a run ρ from q**.
- For a system $Sys \in IS$ and a global state $q \in Q$ we define $\boldsymbol{reach(q)} := \{q' \in Q \mid \exists$ a path ϕ from q to $q'\}$. Note that the existence of a run from q implies (together with the finiteness of the global transition system) the existence of a cycle that is reachable from q.
- We define the set of reachable states of Sys (with global starting state q^0) by $\boldsymbol{reach(Sys)} := reach(q^0)$.
- For $\alpha \in Int, k \in K$ we say that **k participates in α** if $k(\alpha) := \alpha \cap A_k \neq \emptyset$. If we have $k(\alpha) = \{a_k\}$ we say that k participates in α with a_k. Otherwise, we say that k does not participate in α.
- A global state **q enables an interaction α** $\in (C \cup Comp)$ if $\exists q' \in Q$: $q \xrightarrow{\alpha} q'$. We write $q \not\rightarrow$ if q does not enable any $\alpha \in (C \cup Comp)$.
- A global state **q enables a component k** $\in K$ if q enables some interaction α in which k participates. **q enables an action a_k** of some component $k \in K$ (resp. a_k is enabled in q) if q enables an interaction α in which k participates with a_k.
- Let $q = (q_1, \ldots, q_n) \in Q$ be a global state. We say that some non-empty set $\tilde{K} = \{j_1, j_2, \ldots, j_{|\tilde{K}|}\} \subseteq K$ of components **is in local deadlock in q** if $\forall i \in \tilde{K}\ \forall \alpha \in (C \cup Comp)$: $(\alpha \cap ea(q_i) \neq \emptyset) \Rightarrow (\exists j \in \tilde{K}\ j(\alpha) \not\subseteq ea(q_j))$.
- A **global deadlock** is a special case of a local deadlock, when $\tilde{K} = K$.
- For a system Sys that has no global deadlock, we define that **$k \in K$ does progress in Sys** if k occurs infinitely often in every run from q^0.
- For a system Sys that has no global deadlock, we define that **$k \in K$ is available in Sys** if k is enabled infinitely often in (states occuring in) every run from q^0.

We define in the following a list of decidability problems:
Reachability $:= \{(Sys, q) \mid Sys \in IS$ and $q \in reach(Sys)\}$.
LDIS $:= \{Sys \in IS \mid \exists q \in reach(Sys)\ \exists\ \tilde{K} \subseteq K$ s.t. $\tilde{K}$ is in local deadlock in $q\}$.
GDIS $:= \{Sys \in IS \mid \exists q \in reach(Sys)$, s.t. $q \not\rightarrow\}$.
Progress $:= \{(Sys, k) \mid Sys \in (IS \setminus GDIS)$ and $k \in K[Sys]$ does progress in $Sys\}$.
Availability $:= \{(Sys, k) \mid Sys \in (IS \setminus GDIS)$ and $k \in K[Sys]$ is available in $Sys\}$.

2.3 Interaction Systems and 1-Safe Petri Nets

In this subsection we give a short discussion of interaction systems versus 1-safe Petri-nets. As we showed in [MM08b] we can translate a 1-safe Petri net into an interaction system in time polynomial in the size of the input such that the property of reachability is preserved. This will be the basis for our PSPACE-hardness results. On the other hand one can show that there is no general translation from interaction systems to 1-safe Petri nets that yields bisimilarity for the global transition systems and that there is no polynomial translation that yields isomorphy even for the unlabeled versions of the global transition systems.

Still one might want to ask for further motivation why one should deal with interaction systems instead of using Petri nets.

Our first argument concerns the fact that we want to model and investigate component-based systems. In a component-based system it should be possible to freely combine components in a very flexible way, substitute a component by another one or change the glue code by which components are put together. As we argue interaction systems are a model that satisfies these needs.

There have been attempts to use Petri nets for the analysis of component-based systems. In [BB04, BB06] the model CompoNets based on colored Petri nets is proposed. In this model every component offers a set of ports. Its behavior is described by a Petri net. There is a set of syntactic rules that regulate how components are glued together via their port sets. However there is no formalism that allows to determine the behavior of the global system. Moreover when components are put together the identity of a component is lost and hence substituting one component for another one in the composed systems or asking for the liveness of a component is not feasible. Other approaches using Petri nets to model component-based systems can be found in [AS99, AS02, PK07, SVvdW]. General problems with Petri nets approaches are that Petri nets lack full compositionality and the loss of the identity of components in the composed system which is needed for reconfiguration of systems.

Given this situation one could think of modeling components systems by interaction systems and then transforming the interaction system by our translation into a 1-safe Petri net which could then be analyzed by a Petri net tools or submitted to a model checker. When however trying to translate an interaction system into a 1-safe Petri net one can show that there are simple interaction systems for which no bisimilar 1-safe Petri exists. In a context of model checking e.g. with respect to modal μ-calculus bisimilarity is however very important as two processes satisfy the same set of formulae if and only if they are bisimilar. Hence if we are interested in general properties as expressed by modal μ-calculus this approach does not work.

3 The Polynomial Time Reductions

In [MM08b], we gave a polynomial translation from 1-safe Petri nets to interaction systems, which yielded PSPACE-hardness for reachability in interaction systems. In this section, we give four polynomial reductions $f_1, \ldots, f_4$ that build a reduction chain as depicted in Figure 2. The chain allows us to derive the PSPACE-hardness result from reachability for all considered properties as well as PSPACE-solvability from availability for all properties in the chain. Hence we prove all problems in the chain to be PSPACE-complete. Although the reductions vary strongly in their degree of difficulty they also have some basic idea in common. In each of the reductions, we add a component *main* to the system. However, the local transition system of *main* will be a different one for each reduction.

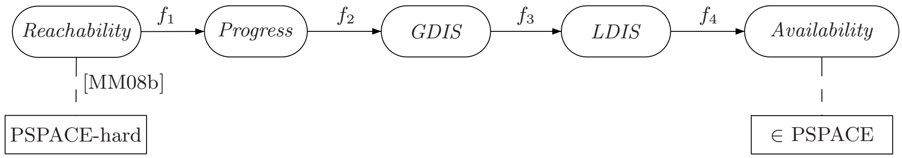

Fig. 2. The Polynomial Time Reductions $f_i \quad (1 \leq i \leq 4)$

For each reduction, we present its formal definition followed by a short explanation. Explicit formal proofs have been omitted for better readability. The proofs for the reductions are sketched in the various following subsections and the verification of their logspace computability is left to the reader.

We will now give a short reasoning why *Availability* is in *PSPACE*:

Given an interaction system and one of its components k we want to decide whether from every reachable global state we will, - no matter in which way we continue our transition sequence - eventually reach a state that enables k.

Note that $\overline{Availability}$ is the question whether there exists a reachable global state q, from which q itself is reachable by a non-empty transition sequence $q \to q' \to \ldots \to q$ such that none of the global states $q, q', \ldots$ enables k.

To solve $\overline{Availability}$ we first guess our way from the global starting state q^0 to some q as described above. It is easy to verifiy in each step in polynomial space that we follow indeed an allowed edge in the global transition system. Next, once we reach q, we store it and guess the cycle described above back to q. It is possbile in polynomial space to verifiy that the cycle is non-empty, that none of the visited states enables k and that we do indeed reach q after all.

So $\overline{Availability}$ is in *NPSPACE* and thus *Availability* is in *co-NPSPACE* which equals *PSPACE* due to Savitch [Sav70].

3.1 *Reachability* Is Polynomially Reducible to *Progress*

Theorem 1. *Reachability is polynomially reducible to Progress*

Proof. Let $Sys \in IS$ and $q = (q_1, \ldots, q_n) \in Q[Sys]$. We associate with (Sys,q) an interaction system $f_1(Sys, q)$ (which is free of global deadlocks) s.t.

$$((Sys,q) \in Reachability) \Leftrightarrow ((f_1(Sys,q),main) \notin Progress).$$

Formal definition of f_1.
Let $Sys = \{K, \{A_i\}_{i\in K}, C, Comp, \{T_i\}_{i\in K}\}$, then
$f_1(Sys, q) = \{K', \{A_i'\}_{i\in K'}, C', Comp', \{T_i'\}_{i\in K'}\}$, where

$\boldsymbol{K'} := K \cup \{main\}$,
For $i \in K$: $\boldsymbol{A_i'} := A_i \cup \{run_i\}$,
$\boldsymbol{A_{main}'} := \{dummy_{main}, check_{main}\}$,

For $i \in K$: $\boldsymbol{T_i'} := (Q_i, A_i', \rightarrow_i', q_i^0)$, where

$$\rightarrow_i' := \rightarrow_i \cup \{(q_i, run_i, q_i)\},$$

$$\boldsymbol{T'_{main}} := (\{q^0_{main}\}, A'_{main}, \rightarrow'_{main}, q^0_{main}), \text{ where}$$

$$\rightarrow'_{main} := \{(q^0_{main}, check_{main}, q^0_{main}), (q^0_{main}, dummy_{main}, q^0_{main})\}.$$

$$\boldsymbol{C'} := \{c \cup \{check_{main}\} \mid c \in C\} \cup \{\{run_i \mid 1 \leq i \leq n\}\} \cup \{\{dummy_{main}\}\},$$

$$\boldsymbol{Comp'} := \{\alpha \cup \{check_{main}\} \mid \alpha \in Comp\}.$$

Explanation. We add a component *main* whose local transition system consists of a single state with two loops. Also, for each local transition system T_i we add a loop in the state q_i labeled by run_i. Clearly $f_1(Sys, q) \in IS$ holds. The loop of *main* labeled by $dummy_{main}$ can be performed independently (i.e. $\{dummy_{main}\}$ is a connector) and assures that $f_1(Sys,q) \notin GDIS$ (which is a precondition for asking for progress). The second loop is labeled by the action $check_{main}$, which is added to every interaction $\alpha \in C \cup Comp$. Hence, the only interaction in $C \cup Comp$ in which *main* does not participate is $\{run_1, \ldots, run_n\}$.

This fact, together with the obvious observation that q is reachable in *Sys* iff q extended by q^0_{main} is reachable in $f_1(Sys, q)$ allows us to conclude that in $f_1(Sys, q)$ there is a run from q in which *main* does not participate iff q is reachable in *Sys*.

3.2 *Progress* Is Polynomially Reducible to *GDIS*

Preliminaries. The construction applied in Example 1 in Section 2.1 can easily be parameterized in order to build an interaction system for an m^n-counter, $m, n \in \mathbb{N}$:

$Count_{(m,n)} = (\{n+1, \ldots, 2n\}, \{A_i\}_{n+1 \leq i \leq 2n}, C, Comp, \{T_i\}_{n+1 \leq i \leq 2n})$,

where $A_i = \{inc_i, dec_i\}$ for $n+1 \leq i \leq 2n-1$ and $A_{2n} = \{inc_{2n}, dummy_{2n}\}$

$$C = \{\{inc_{n+1}, dummy_{2n}\}\} \cup \bigcup_{i=n+2}^{2n} \{c(inc_i)\}$$

$$\text{where } c(inc_i) = \{inc_i\} \cup \bigcup_{j=n+1}^{i-1} \{dec_j\},$$

$$Comp = \{\{inc_{n+1}\}, \{inc_{n+1}, dummy_{2n}\}\},$$

$$T_i = (Q_i, A_i, \rightarrow_i, q_i^0), \text{ where } Q_i = \{q_i^0, \ldots, q_i^{m-1}\} \text{ and}$$

$$\rightarrow_i = \begin{cases} \{(q_i^j, inc_i, q_i^{j+1}) \mid 0 \leq j \leq m-2\} \cup \{(q_i^{m-1}, dec_i, q_i^0)\} \,; n+1 \leq i \leq 2n-1 \\ \{(q_i^j, inc_i, q_i^{j+1}) \mid 0 \leq j \leq m-2\} \cup \{(q_i^{m-1}, dummy_{2n}, q_i^{m-1})\} \,; i = 2n \end{cases}$$

As already pointed out in Section 2.1, such a system behaves deterministically and simply performs $m^n - 1$ ("counting") interactions before stopping.

Theorem 2. *Progress is polynomially reducible to GDIS*[2].

Proof. Let $Sys \in (IS \setminus GDIS)$ and $k \in K[Sys]$. In case k participates in every $\alpha \in C \cup Comp$, k does progress[3]. Otherwise, we associate with (Sys,k) an interaction system $f_2(Sys, k)$ s.t.

$$((Sys,k) \in Progress) \Leftrightarrow (f_2(Sys,k) \notin GDIS).$$

In the following, let $m := max\{|Q_i| \mid i \in K[Sys]\}$.

[2] Please note that an alternative proof of PSPACE-hardness of GDIS is given in [MM08b]. Thus f_2 mainly serves to establish PSPACE-completeness.

[3] We have to consider this case explicitly because $f_2(Sys, k) \notin IS$ for such an input.

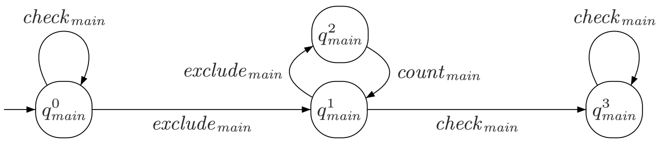

Fig. 3. The local transition system T'_{main}

Formal definition of f_2.
Let $Sys = \{K, \{A_i\}_{i\in K}, C, Comp, \{T_i\}_{i\in K}\}$, then
$f_2(Sys, k) = \{K', \{A'_i\}_{i\in K'}, C', Comp', \{T'_i\}_{i\in K'}\}$, where

$$\begin{aligned}
\boldsymbol{K'} &:= K \cup \{n+1, \ldots, 2n, main\},\\
\text{For } i \in K\text{: } \boldsymbol{A'_i} &:= A_i,\\
\text{For } i \in \{n+1, \ldots, 2n\}\text{: } \boldsymbol{A'_i} &:= A_i[Count_{(m,n)}],\\
\boldsymbol{A'_{main}} &:= \{check_{main}, exclude_{main}, count_{main}\},\\
\text{For } i \in K\text{: } \boldsymbol{T'_i} &:= T_i,\\
\text{For } i \in \{n+1, \ldots, 2n\}\text{: } \boldsymbol{T'_i} &:= T_i[Count_{(m,n)}],\\
\text{and } \boldsymbol{T'_{main}} &\ \text{ is depicted in Figure 3.}\\
\boldsymbol{C_{check}} &:= \{c \cup \{check_{main}\} \mid c \in C\}\\
\boldsymbol{Comp_{check}} &:= \{\alpha \cup \{check_{main}\} \mid \alpha \in Comp\}\\
\boldsymbol{C_{exclude}} &:= \{c \cup \{exclude_{main}\} \mid c \in C \wedge k(c) = \emptyset\}\\
\boldsymbol{Comp_{exclude}} &:= \{\alpha \cup \{exclude_{main}\} \mid \alpha \in Comp \wedge k(\alpha) = \emptyset\}\\
\boldsymbol{C_{counter}} &:= \{c \cup \{count_{main}\} \mid c \in C[Count_{(m,n)}]\}\\
\boldsymbol{Comp_{counter}} &:= \{\alpha \cup \{count_{main}\} \mid \alpha \in Comp[Count_{(m,n)}]\}\\
&(= \{\{inc_{n+1}, count_{main}\}, \{inc_{n+1}, dummy_{2n}, count_{main}\}\})\\
\boldsymbol{C'} &:= C_{check} \cup C_{exclude} \cup C_{counter}\\
\boldsymbol{Comp'} &:= Comp_{check} \cup Comp_{exclude} \cup Comp_{counter}
\end{aligned}$$

Explanation. First, we observe that $f_2(Sys, k) \in IS$ holds. *Sys* is globally deadlock-free and we want to know whether it contains a run from q^0, in which k does not participate infinitely often. This amounts to the question, whether there is a reachable global state, that lies on a cycle that does not involve k. As m^n is an upper bound for the size of the global state space of *Sys*, this is equivalent to asking whether it is possible to perform m^n consecutive interactions in which k does not participate.

3.3 *GDIS* Is Polynomially Reducible to *LDIS*

Theorem 3. *GDIS is polynomially reducible to LDIS*

Proof. Let $Sys \in IS$. We associate with *Sys* an interaction system $f_3(Sys)$ s.t.

$$(Sys \in GDIS) \Leftrightarrow (f_3(Sys) \in LDIS).$$

Formal definition of f_3.
Let $Sys = \{K, \{A_i\}_{i\in K}, C, Comp, \{T_i\}_{i\in K}\}$, then
$f_3(Sys) = \{K', \{A'_i\}_{i\in K'}, C', Comp', \{T'_i\}_{i\in K'}\}$, where

$$\begin{aligned}
\boldsymbol{K'} &:= K \cup \{main\},\\
\text{For } i\in K\text{: } \boldsymbol{A'_i} &:= A_i \cup \{dummy_i\},\\
\boldsymbol{A'_{main}} &:= \{dummy_{main}, check_{main}\},\\
\text{For } i\in K\text{: } \boldsymbol{T'_i} &:= (Q_i, A'_i, \to'_i, q_i^0), \text{ where}\\
\to'_i &:= \to_i \cup \{(q_i, dummy_i, q_i) \mid q_i \in Q_i\}.\\
\boldsymbol{T'_{main}} &:= (\{q^0_{main}, q^1_{main}\}, A'_{main}, \to'_{main}, q^0_{main}), \text{ where}\\
\to'_{main} &:= \{(q^0_{main}, check_{main}, q^1_{main}), (q^1_{main}, dummy_{main}, q^0_{main})\},\\
\boldsymbol{C'} &:= \{c \cup \{check_{main}\} \mid c \in C\} \cup \{\{dummy_1, \ldots, dummy_n, dummy_{main}\}\},\\
\boldsymbol{Comp'} &:= \{\alpha \cup \{check_{main}\} \mid \alpha \in Comp\}.
\end{aligned}$$

Explanation. Clearly, $f_3(Sys) \in IS$. We add an additional component *main* which alternatingly accompanies orignal interactions of *Sys* in one step and then allows the system to perform a connector including all components in a second step. This preserves global deadlocks but resolves local ones.

3.4 *LDIS* Is Polynomially Reducible to *Availability*

Theorem 4. *LDIS is polynomially reducible to Availability*

Proof. Let $Sys \in IS$. We associate with *Sys* an interaction system $f_4(Sys)$ (which is free of global deadlocks) s.t.

$$(Sys \in LDIS) \Leftrightarrow ((f_4(Sys), main) \notin Availability).$$

Formal definition of f_4.
Let $Sys = \{K, \{A_i\}_{i\in K}, C, Comp, \{T_i\}_{i\in K}\}$,
then $f_4(Sys) = \{K', \{A'_i\}_{i\in K'}, C', Comp', \{T'_i\}_{i\in K'}\}$, where[4]

$$\begin{aligned}
\boldsymbol{K'} &:= K \cup \{n{+}1\} \cup \{main\}\\
\text{For } i\in K\text{: } \boldsymbol{A'_i} &:= A_i \cup \{\hat{a}_i \mid a_i \in A_i\} \cup \{lock_i, unlock_i, d_i, \overline{d}_i, clear_i\}\\
\boldsymbol{A'_{n+1}} &:= \{dummy_{n+1}, lock_{n+1}, unlock_{n+1}\}\\
\boldsymbol{A'_{main}} &:= \{lock_{main}, unlock_{main}, clear_{main}, progress_{main}\}\\
\text{For } i\in K\text{: } \boldsymbol{T'_i} &:= (Q'_i, A'_i, \to'_i, q_i^0), \text{ where}\\
Q'_i &:= \textstyle\bigcup_{q_i\in Q_i}\{q_i, \hat{q}_i, q_i^D, \hat{q}_i^D, q_i^{\overline{D}}, \hat{q}_i^{\overline{D}}, q_i^{clr}\}\\
\to'_i &:= \textstyle\bigcup_{q_i\in Q_i}\{\, (q_i, lock_i, \hat{q}_i), (\hat{q}_i, d_i, q_i^D), (q_i^D, unlock_i, \hat{q}_i^D), (\hat{q}_i, \overline{d}_i, q_i^{\overline{D}}),\\
&\qquad (q_i^{\overline{D}}, unlock_i, \hat{q}_i^{\overline{D}}), (\hat{q}_i^D, \textstyle\bigcup_{a_i\in ea(q_i)}\{a_i, \hat{a}_i\}, q_i^{clr}),\\
&\qquad (\hat{q}_i^{\overline{D}}, A_i \cup \{all_i\}, \hat{q}_i^{\overline{D}}), (q_i^{clr}, clear_i, q_i^{clr})\}\\
&\quad \cup \to_i
\end{aligned}$$

$\boldsymbol{T'_{n+1}}$ and $\boldsymbol{T'_{main}}$ are given in Figure 5.

[4] For ease of notation we use sets of actions as edge labels in the definition of $\to'_i$ as well as in Figure 4. When we write $(q, A, q') \in\to'_i$ we mean $(q, a, q') \in\to'_i \ \forall a \in A$. Note that by $ea(q_i)$ we refer to the enabled actions of the local state q_i in *Sys* (not in $f_4(Sys)$).

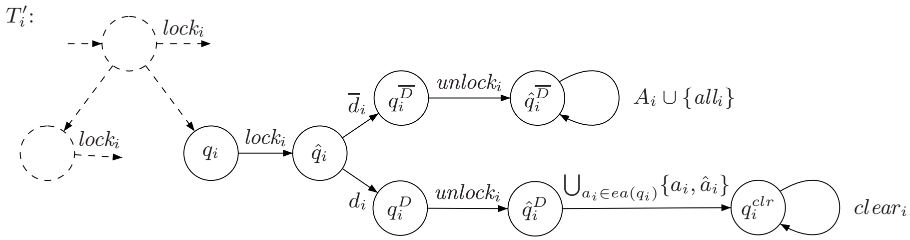

Fig. 4. The modification for a local state q_i in the local transition system T_i'

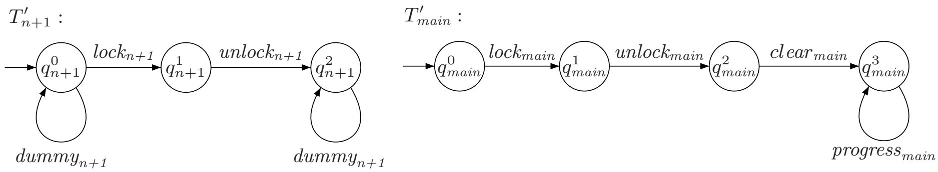

Fig. 5. The local transition system T_{main}'

The result of our modifications is sketched for a single state $q_i \in Q_i$ in Figure 4.

$$\begin{aligned}
\boldsymbol{C'} := \; & \{\{dummy_{n+1}\}, \{lock_1, \ldots, lock_n, lock_{n+1}, lock_{main}\}\} \\
& \cup \{\{unlock_1, \ldots, unlock_n, unlock_{n+1}, unlock_{main}\}\} \\
& \cup \{\{d_1\}, \ldots, \{d_n\}, \{\overline{d}_1\}, \ldots, \{\overline{d}_n\}\} \\
& \cup \{\{all_1, \ldots, all_n, clear_{main}\}\} \\
& \cup \{\{clear_1, clear_{main}\}, \ldots, \{clear_n, clear_{main}\}\} \\
& \cup \{\{progress_{main}\}\} \\
& \cup C \cup C^{clear}, \text{ where} \\
C^{clear} := \; & \{\{clear_{main}, \hat{a}\} \cup (c \setminus a) \mid a \in c \in C\} \\
\boldsymbol{Comp'} := \; & Comp \cup Comp^{clear}, \text{ where} \\
Comp^{clear} := \; & \{\{clear_{main}, \hat{a}\} \cup (\alpha \setminus a) \mid a \in \alpha \in Comp\}
\end{aligned}$$

Explanation. Clearly, $f_4(Sys) \in IS$ holds. Component $n+1$ guarantees $f_4(Sys) \notin GDIS$. The idea of our reduction is as follows: In the beginning *main* offers in any reachable state an action $lock_{main}$, which can participate in the *lock*-interaction which includes all components. As a result, *main* will always be enabled as long as this action is not performed. Now in any reachable state q of *Sys* we want to be able to check whether there is a local deadlock in q. For this purpose in any reachable state $(q_1, \ldots, q_n, q_{n+1}^0, q_{main}^0)$, the interaction $\{lock_1, \ldots, lock_n, lock_{n+1}, lock_{main}\}$ can be performed leading to a state where for every $i \in K$ a choice between d_i and $\overline{d}_i$ takes place. Those components j that select d_j form a subset $\tilde{K} \subseteq K$. If and only if $\tilde{K}$ is in local deadlock in $(q_1, \ldots, q_n)$ in *Sys*, the component *main* will not be able to participate in any further interaction.

4 Conclusion and Related Work

We give a complete complexity-theoretic characterization of the most relevant behavioral questions in interaction systems. Similar results have been proved for 1-safe Petri nets in [CEP93]. The PSPACE-hardness results motivate other approaches to guarantee the discussed properties. One approach is to establish conditions that can be tested in polynomial time and imply the desired properties [MMM07, MMM, IU01, BHH$^+$06]. Another approach exploits compositionality [AB03, GGM$^+$07]. Further, one may put restrictions on the communication structure of the interaction system [MM08a, BCD02]. The aim of all these approaches is to derive global properties from local information as much as possible.

References

[AB03] Aldini, A., Bernardo, M.: A General Approach to Deadlock Freedom Verification for Software Architectures. In: Araki, K., Gnesi, S., Mandrioli, D. (eds.) FME 2003. LNCS, vol. 2805, pp. 658–677. Springer, Heidelberg (2003)

[AS99] Aoumeur, N., Saake, G.: Towards an Object Petri Nets Model for Specifying and Validating Distributed Information Systems. In: Jarke, M., Oberweis, A. (eds.) CAiSE 1999. LNCS, vol. 1626, pp. 381–395. Springer, Heidelberg (1999)

[AS02] Aoumeur, N., Saake, G.: A Component-based Petri Net Model for Specifying and Validating Cooperative Information Systems. Data Knowl. Eng. 42(2), 143–187 (2002)

[BB04] Bastide, R., Barboni, E.: Component-based Behavioural Modelling with High-Level Petri Nets. In: Proceedings of MOCA 2004, pp. 37–46 (2004)

[BB06] Bastide, R., Barboni, E.: Software Components: a Formal Semantics Based on Coloured Petri Nets. In: Proceedings of FACS 2005. ENTCS, vol. 160, pp. 57–73 (2006)

[BBS06] Basu, A., Bozga, M., Sifakis, J.: Modeling Heterogeneous Real-time Components in BIP. In: Proceedings of SEFM 2006, pp. 3–12. IEEE Computer Society Press, Los Alamitos (2006)

[BCD02] Bernardo, M., Ciancarini, P., Donatiello, L.: Architecting Families of Software Systems with Process Algebras. ACM Trans. on Software Engineering and Methodology 11, 386–426 (2002)

[BHH$^+$06] Baumeister, H., Hacklinger, F., Hennicker, R., Knapp, A., Wirsing, M.: A Component Model for Architectural Programming. In: Proceedings of FACS 2005. ENTCS, vol. 160, pp. 75–96 (2006)

[CEP93] Cheng, A., Esparza, J., Palsberg, J.: Complexity Results for 1-safe Nets. In: Shyamasundar, R.K. (ed.) FSTTCS 1993. LNCS, vol. 761, pp. 326–337. Springer, Heidelberg (1993)

[dAH01] de Alfaro, L., Henzinger, T.: Interface Automata. In: Matsui, M. (ed.) FSE 2001. LNCS, vol. 2355, pp. 109–120. Springer, Heidelberg (2002)

[GGM$^+$07] Goessler, G., Graf, S., Majster-Cederbaum, M., Martens, M., Sifakis, J.: Ensuring Properties of Interaction Systems. In: Reps, T., Sagiv, M., Bauer, J. (eds.) Wilhelm Festschrift. LNCS, vol. 4444, pp. 201–224. Springer, Heidelberg (2007)

[GO07] Graf, S., Ober, I.: Modelling and Validation of Software and Architecture with Omega-UML and the IF Validation Toolbox. In: Genie Logiciel (to appear, 2007), http://www.speeds.eu.com

[Goe06] Goessler, G.: Component-based Design of Heterogeneous Reactive Systems in Prometheus. Technical report 6057, INRIA (December 2006)

[GQ07] Graf, S., Quinton, S.: Contracts for BIP: Hierarchical Interaction Models for Compositional Verification. In: Derrick, J., Vain, J. (eds.) FORTE 2007. LNCS, vol. 4574, pp. 1–18. Springer, Heidelberg (2007)

[GS03] Goessler, G., Sifakis, J.: Component-based Construction of Deadlock-free Systems. In: Pandya, P.K., Radhakrishnan, J. (eds.) FSTTCS 2003. LNCS, vol. 2914, pp. 420–433. Springer, Heidelberg (2003)

[GS05] Goessler, G., Sifakis, J.: Composition for Component-based Modeling. Sci. Comput. Program. 55(1-3), 161–183 (2005)

[IU01] Inverardi, P., Uchitel, S.: Proving Deadlock Freedom in Component-Based Programming. In: Hussmann, H. (ed.) FASE 2001. LNCS, vol. 2029, pp. 60–75. Springer, Heidelberg (2001)

[LT89] Lynch, N.A., Tuttle, M.R.: An Introduction to Input/Output Automata. In: CWI-Quarterly, pp. 219–246 (1989)

[MM08a] Majster-Cederbaum, M., Martens, M.: Compositional Analysis of Tree-Like Component Architectures (submitted for publication, 2008)

[MM08b] Majster-Cederbaum, M., Minnameier, C.: Deriving Complexity Results for Interaction Systems from 1-safe Petri Nets. In: Geffert, V., Karhumäki, J., Bertoni, A., Preneel, B., Návrat, P., Bieliková, M. (eds.) SOFSEM 2008. LNCS, vol. 4910, pp. 352–363. Springer, Heidelberg (2008)

[MMM] Majster-Cederbaum, M., Martens, M., Minnameier, C.: Liveness in Interaction Systems. In: Proceedings of FACS 2007. ENTCS (2007)

[MMM07] Majster-Cederbaum, M., Martens, M., Minnameier, C.: A Polynomial-time Checkable Sufficient Condition for Deadlock-Freedom of Component-based Systems. In: van Leeuwen, J., Italiano, G.F., van der Hoek, W., Meinel, C., Sack, H., Plášil, F. (eds.) SOFSEM 2007. LNCS, vol. 4362, pp. 888–899. Springer, Heidelberg (2007)

[PK07] Padberg, J., Kuessel, U.: A Component-based Verification Approach based on Petri Net Components. In: Proceedings of FORMS/FORMAT 2007, pp. 40–50 (2007)

[Sav70] Savitch, W.: Relationships between Nondeterministic and Deterministic Tape Complexities. Journal of Computer and System Sciences 4, 177–192 (1970)

[Sif05] Sifakis, J.: A Framework for Component-based Construction. In: Proceedings of the Third IEEE International Conference on Software Engineering and Formal Methods, pp. 293–300. IEEE Computer Society Press, Los Alamitos (2005)

[SVvdW] Sidorova, N., Voorhoeve, M., van der Woude, J.C.S.P.: A Calculus of Petri Net Components. In: Proceedings of MOCA 2001, pp. 121–132 (2001)