Reachability in Cooperating Systems with Architectural Constraints is PSPACE-Complete

Mila Majster-Cederbaum   Nils Semmelrock*

University Mannheim
Mannheim, Germany

mcb@informatik.uni-mannheim.de   nsemmelr@informatik.uni-mannheim.de

The reachability problem in cooperating systems is known to be PSPACE-complete. We show here that this problem remains PSPACE-complete when we restrict the communication structure between the subsystems in various ways. For this purpose we introduce two basic and incomparable subclasses of cooperating systems that occur often in practice and provide respective reductions. The subclasses we consider consist of cooperating systems the communication structure of which forms a line respectively a star.

1   Introduction

Cooperating systems are systems that consist of subsystems which cooperate by a *glue-code*. The reachable state space of such systems can be exponentially large in the number of subsystems what is referred to as the *state space explosion problem*. Moreover, there are PSPACE-completeness results for the reachability problem in various formalisms that model cooperating systems, e.g., see [18] for interaction systems, [13] for results in 1-conservative Petri nets and [7] for results in 1-save Petri nets. Clearly, all methods that rely on the exploration of the reachable state space of cooperating systems suffer from these results. Particularly formal verification techniques as LTL or CTL model checking have a runtime that is exponential in the number of subsystems. There are various ways to cope with this problem. One approach is to identify subclasses for which an analysis can be achieved in polynomial time. Hence, the question arises whether there are relevant subclasses of cooperating systems where the reachability problem is decidable in polynomial time. Popular decision problems that are complete in NP or even in PSPACE are decidable in polynomial or linear time in certain subclasses of instances. Maybe the most popular example is the Boolean satisfiability problem where 3SAT is NP-complete, 2SAT is decidable in polynomial time and HORNSAT (the problem of deciding whether a given set of propositional Horn clauses is satisfiable) is even decidable in linear time. Similarly, the quantified 3SAT problem is PSPACE-complete, whereas the quantified 2SAT problem is also decidable in polynomial time (see [8] for descriptions and more examples).

There are various starting points to specify subclasses of cooperating systems.

1. Restrictions regarding the behavior of the subsystems.

2. The degree of synchronization among the subsystems as systems with a very high degree of synchronization tend to display a smaller reachable state space.

3. The glue-code, i.e., the structure of the interaction among the subsystems.

---

*Corresponding author - Phone: (49) 621-181 2564 - Fax: (49) 621-181 2560

Here, our concern lies on the latter.

As a formal model we consider here interaction systems [10], a very general formalism for modeling cooperating systems that allows for multiway interactions between subsystems. The results in this paper can be easily applied to other formalisms that model cooperating systems. This can be achieved by either adapting the results, e.g., the formalism of interface automata [1] comes close to interaction systems, or by using a mapping among formalisms, e.g., a mapping between interaction systems and 1-save Petri nets can be found in [17].

Deciding the reachability problem in general interaction systems is PSPACE-complete [18]. Here we strengthen this result by showing that the reachability problem remains PSPACE-complete in subclasses consisting of interaction systems the communication structure of which forms a star or a linear sequence of components. As star structures appear in practice in, e.g., client/server systems as banking or booking systems and linear structures appear in, e.g., pipeline systems as instruction pipelines or general queue based algorithms, it is important to know that even for such "simply" structured systems there is no general efficient analysis method. Our results justify investigations that search for sufficient conditions that can check and guarantee reachability in polynomial time. Also approaches that guarantee correctness by construction, i.e., modeling rules that ensure certain system properties, and are based on structural restrictions become justified by our results. See for example [16, 11, 14, 15, 12, 6, 4] for approaches that treat these topics.

The paper is organized as follows. Section 2 contains the definitions. In Section 3 we introduce a reduction from the acceptance problem in linear bounded Turing machines to the reachability problem in linear interaction systems. A reduction from the reachability problem in general interaction systems to star-like interaction systems is introduced in Section 4. Section 5 concludes this paper.

2   Interaction Systems

Interaction systems have been proposed by Sifakis and Gössler in [10] to model cooperating systems. The model was studied, e.g., in [16, 3, 5, 20, 9]. An interaction system consists of components which cooperate through so called interactions. An interaction specifies a multiway cooperation among components by connecting different interfaces (called ports) of different components. The model is defined in two layers. The first layer, the interaction model, specifies the components, their interfaces and the communication between them. The second layer, the interaction system, describes the behavior of the components by labeled transition systems. In contrast to [10] we allow an interaction to be contained in another interaction and do not consider complete interactions.

Definition 1 *Let K be a set of **components** and $\{A_i\}_{i \in K}$ a family of pairwise disjunct sets of **ports** of the components in K. In the following we assume that $K = \{1, 2, \ldots, n\}$. An **interaction** α is a nonempty set of ports from different components, i.e., $\alpha \subseteq \bigcup_{i \in K} A_i$ and for all $i \in K$ $|\alpha \cap A_i| \leq 1$ holds.*

*An interaction $\alpha_i = \{a_{i_1}, a_{i_2}, \ldots, a_{i_k}\}$ with $a_{i_j} \in A_{i_j}$ ($j \in \{1, 2, \ldots, k\}$) denotes a possible cooperation among the components $i_1, \ldots, i_k$ via their respective ports. A set Int of interactions is called **interaction set (for** K), if each port appears in at least one interaction in Int, i.e., $\bigcup_{i \in K} A_i = \bigcup_{\alpha \in Int} \alpha$. The tuple $IM = (K, \{A_i\}_{i \in K}, Int)$ is called **interaction model** if Int is an interaction set for K.*

Example 1 *Let $r > 0$ be a natural number and $K = \{S, c_1, c_2, \ldots, c_r\}$ a set of components. S models a server with a set of ports $A_S = \{connect, disconnect\}$ where connect models the connection of a client to this server and disconnect models the disconnection. For $1 \leq i \leq r$ component c_i models a client with a set of ports $A_{c_i} = \{connect_i, disconnect_i\}$. $connect_i$ models the connection of client i to the server and $disconnect_i$ the disconnection.*

For $1 \leq i \leq r$ the interaction $connect\_S\_c_i = \{connect, connect_i\}$ models a connection from client i to the server and the interaction $disconnect\_S\_c_i = \{disconnect, disconnect_i\}$ models the disconnection. Let

$$Int = \{connect\_S\_c_i, disconnect\_S\_c_i | 1 \leq i \leq r\}$$

be a set of interactions. Note that Int is an interaction set for K, i.e., $IM = (K, \{A_i\}_{i \in K}, Int)$ is a well defined interaction model.

Definition 2 *Let $IM = (K, \{A_i\}_{i \in K}, Int)$ be an interaction model. $Sys = (IM, \{T_i\}_{i \in K})$ is called **interaction system** where $T_i = (Q_i, A_i, \rightarrow_i, q_i^0)$ for $i \in K$ is a labeled transition systems that models the behavior of component $i \in K$. Q_i is a finite state space, $\rightarrow_i \subseteq Q_i \times A_i \times Q_i$ a transition relation and $q_i^0 \in Q_i$ an initial state. We refer to T_i for $i \in K$ as the **local behavior** of component i and we denote $q_i \xrightarrow{a_i}_i q_i'$ instead of $(q_i, a_i, q_i') \in \rightarrow_i$.*

*We say a state $q_i \in Q_i$ **enables** a_i if there is $q_i' \in Q_i$ with $q_i \xrightarrow{a_i}_i q_i'$. We denote the set of enabled ports of a state $q_i \in Q_i$ by $en(q_i)$. This is, $en(q_i) = \{a_i \in A_i | \exists_{q_i' \in Q_i} q_i \xrightarrow{a_i}_i q_i'\}$.*

Example 2 *Let $IM = (K, \{A_i\}_{i \in K}, Int)$ be the interaction model from example 1. Figure 1 depicts possible local behavior T_i for $i \in K$. This is, the tuple $Sys = (IM, \{T_i\}_{i \in K})$ is a well defined interaction system. We mark initial states by an incoming arrow.*

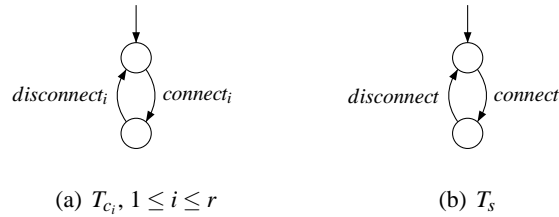(a) T_{c_i}, $1 \leq i \leq r$ (b) T_s

Figure 1: Local behavior of the components in a simple client/server model.

The behavior of an interaction system is defined as follows.

Definition 3 *Let $Sys = (IM, \{T_i\}_{i \in K})$ be an interaction system where the interaction model is given by $IM = (K, \{A_i\}_{i \in K}, Int)$. The **global behavior** of Sys is the transition system $T = (Q, Int, \rightarrow, q^0)$ where*

- *the Cartesian product $Q = \prod_{i \in K} Q_i$ is the **global state space** which we assume to be order independent,*

- *$q^0 = (q_1^0, \ldots, q_n^0)$ is the **global initial state** and*

- *$\rightarrow \subseteq Q \times Int \times Q$ is the **global transition relation** with $q \xrightarrow{\alpha} q'$ if for all $i \in K$:*
  - *$q_i \xrightarrow{a_i}_i q_i'$ if $\alpha \cap A_i = \{a_i\}$ and*
  - *$q_i = q_i'$ if $\alpha \cap A_i = \emptyset$.*

*A state $q \in Q$ is called a **global state**. Globally, a transition $q \xrightarrow{\alpha} q'$ can be performed if each port in α can be performed in the state of the local behavior of its respective component.*

Definition 4 *Let Sys be an interaction system and $T = (Q, Int, \rightarrow, q^0)$ the associated global transition system. A global state $q \in Q$ is called **reachable** iff there is a path in T that leads from the initial state q^0 to q. Given an interaction system Sys and a global state q, the **reachability problem** consists of deciding whether or not q is reachable in the global behavior of Sys.*

In order to define subclasses of interaction systems we study architectural constraints with respect to the communication structure between components, i.e., our constraints are defined on the interaction model and are independent from the behavior of the components. The communication structure is defined by an undirected graph the nodes of which are components that are connected by an edge if these components are able to interact.

Definition 5 *Let $IM = (K, \{A_i\}_{i \in K}, Int)$ be an interaction model with $|K| = n$. The **interaction graph** $G = (K, E)$ of IM is an undirected graph with $\{i, j\} \in E$ $(i \neq j)$ iff there is an interaction $\alpha \in Int$ with $\alpha \cap A_i \neq \emptyset$ and $\alpha \cap A_j \neq \emptyset$, i.e., if there is an interaction in which both components participate.*

*An interaction model IM is called **star-like** iff G is a star, i.e., exactly one node is of degree $n-1$ and all other nodes are of degree 1. IM is called **linear** iff G is connected, two nodes are of degree 1 and any other node is of degree 2. An interaction system Sys is called star-like respectively linear if the interaction model of Sys is star-like respectively linear.*

Remark 1 *Note that star-like and linear interaction systems with a set Int of interactions imply that for all $\alpha \in Int$ $|\alpha| \leq 2$.*

A star-like or linear interaction system can be seen as a system with a simple hierarchical communication structure, e.g., the simple client/server system in Example 1. Of course, such systems can be far more complex and thus exhibit a highly branched communication structure. This is, a PSPACE-completeness result for deciding the reachability problem in the subclass of star-like or linear interaction systems implies the PSPACE-completeness of deciding the reachability problem in systems with a hierarchical communication structure.

As deciding the reachability problem in general interaction systems is in PSPACE it follows that the same holds for the classes of linear and star-like systems.

Example 3 *The interaction graph G of the interaction model $IM = (K, \{A_i\}_{i \in K}, Int)$ from Example 1 is depicted in Figure 2. The interaction graph is a star, i.e., IM is star-like and thus, every interaction system, particularly Sys in Example 2, that contains IM is star-like.*
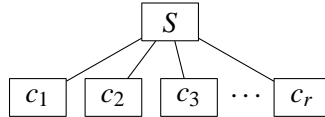

Figure 2: Interaction graph *G* for the interaction model *IM* in Example 1.

Example 4 *This example illustrates a linear interaction system. We consider a simple communication pipeline consisting of n stations. Station one initiates passing a message to station two, station two passes the message to station three and so on. If the message arrives at station n then station n passes an acknowledge message, on the same way, back to station one.*

Let $IM = (K, \{A_i\}_{i \in K}, Int)$ be the interaction model with components $K = \{s_1, s_2, \ldots, s_n\}$ for $n \geq 2$ where s_i models station i for $1 \leq i \leq n$. A station s_i with $1 < i < n$ can receive a message (rec_m_i), pass the message forward ($send\_m_i$), receive an acknowledge (rec_a_i) and pass the acknowledge forward ($send\_a_i$). Station s_1 can only send the initial message and receive the acknowledge and station s_n can only receive a message and send an acknowledge. This is, the port sets of the components are defined as follows.

$$
\begin{aligned}
A_{s_1} &= \{send\_m_1, rec\_a_1\} \\
A_{s_i} &= \{rec\_m_i, send\_m_i, rec\_a_i, send\_a_i\}, \ 1 < i < n \\
A_{s_n} &= \{rec\_m_n, send\_a_n\}
\end{aligned}
$$

The interaction set Int is given by the following interactions.

$$
\begin{aligned}
send\_message_i &= \{send\_m_i, rec\_m_{i+1}\},\ 1 \le i < n \\
send\_acknowledge_i &= \{send\_a_i, rec\_a_{i-1}\},\ 1 < i \le n
\end{aligned}
$$

Let $Sys = (IM, \{T_i\}_{i \in K})$ be the interaction system with local behavior depicted in Figure 3.
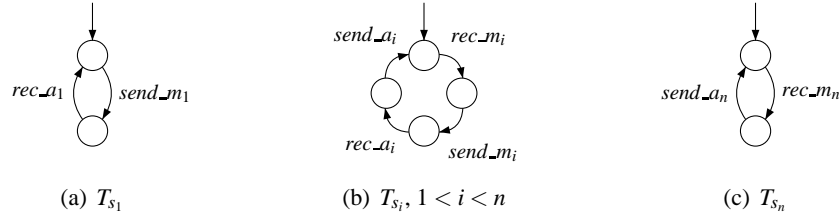

(a) T_{s_1} (b) $T_{s_i},\ 1 < i < n$ (c) T_{s_n}

Figure 3: Local behavior of the components in a simple communication pipeline.

The interaction graph G of IM is depicted in Figure 4. G forms a line of components. Thus, IM is a linear interaction model and Sys is a linear interaction system.
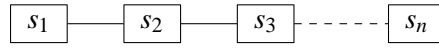

$$ s_1 \;\text{—}\; s_2 \;\text{—}\; s_3 \;\text{------}\; s_n $$

Figure 4: Interaction graph *G* for the interaction model *IM* in Example 4.

3 PSPACE-completeness of Reachability in Linear Systems

In the following we give a reduction from the accepting problem in linear bounded Turing machines to the reachability problem in linear interaction systems. We use the following syntax for a Turing machine but we refrain from repeating the well known semantics (see [8] for details).

Definition 6 *A 4-tuple $M = (\Gamma, \Sigma, P, \delta)$ is called **deterministic Turing machine (DTM)** where*

- Γ *is a finite set of **tape symbols**,*

- $\Sigma \subseteq \Gamma$ *is a set of **input symbols** with a distinguished **blank symbol** $b \in \Gamma \setminus \Sigma$,*

- P *is a finite set of **states**, including an **initial state** p^0 and two **halt states** p^Y and p^N and*

- δ *is the **transition function** with $\delta : (P \setminus \{p^Y, p^N\}) \times \Gamma \to P \times \Gamma \times \{-1, +1\}$.*

We consider a both-sided infinite tape with cells labeled by integers. Given an input $x \in \Sigma^*$ written on the cells labeled 1 through $|x|$ we assume M to be initially in the initial state p^0 and the tape head pointing at cell 1. For a string $x \in \Sigma^*$ with $|x| = n$ we denote the ith letter in x by x^i for $1 \le i \le n$.

*A DTM M is called **linear bounded** if no computation on M uses more than $n + 1$ tape cells, where n is the length of the input string. A **configuration** of a bounded DTM M is denoted by $(p; \gamma_0, \ldots, \underline{\gamma_i}, \ldots, \gamma_{n+1})$ where M is in state p, γ_j is the tape symbol in cell $0 \le j \le n + 1$ and the tape head is on cell $\underline{i}$.*

Definition 7 *The problem **linear space acceptance (LSA)** has as input a linear bounded DTM M and a finite string x over the input alphabet of M. The question is whether M accepts x, i.e., does M halt in the state p^Y. It is well known that LSA is PSPACE-complete [8].*

The idea for our reduction is to model the cells of a DTM M by components of an interaction system Sys_M and the transition function of M by interactions such that a path in the global behavior of Sys_M corresponds to an execution of M. In order to calculate the next configuration of M we need the current tape head position, the current tape symbol in the respective cell and the current state of M. We model all these informations in each cell, i.e., in order to model the calculation of the next configuration we need interactions between the component that models the cell with the tape head and the respective components that model the neighboring cells.

Let $M = (\Gamma, \Sigma, P, \delta)$ be a linear bounded DTM and $x \in \Sigma^*$ an input with $|x| = n$. Let $Sys_M = (IM_M, \{T_i\}_{i \in K})$ be an interaction system with interaction model $IM = (K, \{A_i\}_{i \in K}, Int)$ such that $K = \{0, \dots, n+1\}$.

The set of ports A_i for a component $i \in K$ with $1 \leq i \leq n$ is given by

$$A_i = \{(p, \gamma)_i^1, (p, \gamma)_i^2 \mid p \in P \setminus \{p^Y, p^N\}, \gamma \in \Gamma\}.$$

$(p, \gamma)_i^1$ models that the tape head moves away from cell i where γ is the current tape symbol in this cell and M is in state p. Analogously, $(p, \gamma)_i^2$ models that the tape head moves onto cell i where γ is written and M is in state p.

Because of M being linear bounded, we know that δ does not move the tape head from cell 0 to the left respectively from cell $n+1$ to the right. Thus, we can omit ports in A_0 and A_{n+1} that model a head movement from or onto cell -1 and $n+2$. A_0 is given by

$$\begin{aligned} A_0 \quad = \quad & \{(p, \gamma)_0^1 \mid p \in P \setminus \{p^Y, p^N\}, \gamma \in \Gamma, \neg \exists_{p', \gamma'} \delta(p, \gamma) = (p', \gamma', -1)\} \cup \\ & \{(p, \gamma)_0^2 \mid p \in P \setminus \{p^Y, p^N\}, \gamma \in \Gamma, \neg \exists_{p', \gamma'} \delta(p, \gamma) = (p', \gamma', 1)\}. \end{aligned}$$

A_{n+1} is defined analogously. The set of interactions is given by

$$Int = \{\{(p, \gamma)_i^1, (p, \gamma)_{i+T}^2\} \mid \exists_{p', \gamma'} \delta(p, \gamma) = (p', \gamma', T), 0 \leq i + T \leq n+1\}.$$

For $i \in K$ let $T_i = (Q_i, A_i, \rightarrow_i, q_i^0)$ be the local behavior of component i with $Q_i = \{(p, \gamma) \mid p \in P \cup \{s\}, \gamma \in \Gamma\}$ where s is an auxiliary symbol that is not included in P. $(p, \gamma) \in Q_i$ with $p \neq s$ models that the tape head is currently on cell i and the current tape symbol in this cell is γ. (s, γ) models that γ is the content of cell i and the tape head is not on this cell. The local initial states are derived from the initial word on the tape, i.e., $q_0^0 = (s, b)$, $q_1^0 = (p^0, x^1)$, $q_i^0 = (s, x^i)$ for $2 \leq i \leq n$ and $q_{n+1}^0 = (s, b)$. For $i \in K$ let $\rightarrow_i$ be the union of the following transitions.

a) For all $\gamma, \gamma' \in \Gamma$ and $p \in P \setminus \{p^Y, p^N\}$ let $(p, \gamma) \xrightarrow{(p, \gamma)_i^1}_i (s, \gamma')$ if there are $p' \in P$ and $T \in \{-1, 1\}$ such that $\delta(p, \gamma) = (p', \gamma', T)$.

b) For all $\gamma, \tilde{\gamma} \in \Gamma$, $p \in P \setminus \{p^Y, p^N\}$ and $p' \in P$ let $(s, \tilde{\gamma}) \xrightarrow{(p, \gamma)_i^2}_i (p', \tilde{\gamma})$ if there are $\gamma' \in \Gamma$ and $T \in \{-1, 1\}$ such that $\delta(p, \gamma) = (p', \gamma', T)$.

The transitions described in a) model the impact of δ on cell i if the tape head is on this cell. Let M be in state p and the tape head on cell i reading γ, i.e., T_i is in the state (p, γ). If $\delta(p, \gamma) = (p', \gamma', T)$ then γ' is written and the tape head moves to a neighboring cell, i.e., T_i moves to the state (s, γ'). On the other hand, the transitions described in b) model a head movement onto cell i. Let $\tilde{\gamma}$ be the current tape symbol on cell i, i.e., T_i is in state $(s, \tilde{\gamma})$ before the head moves. After the movement let M change its state to p', i.e., T_i moves to the state $(p', \tilde{\gamma})$.

Remark 2 Sys_M *satisfies the conditions of an interaction system: every port of a component occurs in at least one interaction. Let* $i \in K$, $(p, \gamma)_i^1 \in A_i$ *and* $\delta(p, \gamma) = (p', \gamma', T)$ *then* $0 \leq i + T \leq n + 1$ *and* $\{(p, \gamma)_i^1, (p, \gamma)_{i+T}^2\} \in Int$. *For* $(p, \gamma)_i^2 \in A_i$ *is* $0 \leq i - T \leq n + 1$ *and* $\{(p, \gamma)_{i-T}^1, (p, \gamma)_i^2\} \in Int$.

   Sys_M *has a linear communication structure because every component* $1 \leq i \leq n$ *only interacts with its neighboring components* $i - 1$ *and* $i + 1$.

Remark 3 *The reduction is polynomial in the size of an underlying DTM* $M = (\Gamma, \Sigma, P, \delta)$, *since* $|Int| \leq |P| \cdot |\Gamma|$ *and for all* $i \in K$ $|A_i| \leq 2 \cdot |P| \cdot |\Gamma|$ *and* $|Q_i| \leq (|P| + 1) \cdot |\Gamma|$.

Theorem 1 *Let* $M = (\Gamma, \Sigma, P, \delta)$ *be a linear bounded DTM,* $x \in \Sigma^*$ *with* $|x| = n$ *an input for* M *and* Sys_M *the associated linear interaction system.* M *accepts* x *iff a global state* $q = (q_0, \ldots, q_{n+1})$ *is reachable in Sys such that there is* $i \in \{0, \ldots, n+1\}$ *with* $q_i = (p^Y, \gamma)$ *for a tape symbol* $\gamma \in \Gamma$.

Proof 1 *We prove this theorem by giving an isomorphism, with respect to transitions in* Sys_M *and transitions among configurations in* M, *between global states of* Sys_M *and configurations of* M. *The statement of the theorem then follows by induction as the isomorphism maps the initial configuration of* M *to the initial state of* Sys_M.

   Let R *be the set of configurations of* M. *We map* $(p; \gamma_0, \ldots, \underline{\gamma_i}, \ldots, \gamma_{n+1}) \in R$ *to a global state* $q = (q_0, \ldots, q_{n+1})$ *such that* $q_i = (p, \gamma_i)$ *and* $q_j = (s, \gamma_j)$ *for* $j \neq i$. *Let* Q' *be the set of global states that correspond to the configurations in* R. *It is clear that this mapping is a bijection between* R *and* Q'.

   Let $(p; \gamma_0, \ldots, \underline{\gamma_i}, \ldots, \gamma_{n+1}) \in R$ *and* $q = (q_0, \ldots, q_{n+1}) \in Q'$ *be the associated state in* Sys_M. *Let* $\delta(p, \gamma_i) = (p', \gamma_i', T)$, *i.e., the next configuration in* M *is* $(p'; \gamma_0, \ldots, \gamma_i', \underline{\gamma_{i+1}}, \ldots, \gamma_{n+1}) \in R$ *if* $T = 1$ *(the case* $T = -1$ *is treated analogously). The only enabled port in component* i *is* $(p, \gamma_i)_i^1$, *then the only enabled interaction in* q *is* $\{(p, \gamma_i)_i^1, (p, \gamma_i)_{i+T}^2\}$. *Thus, component* i *reaches the state* (s, γ_i') *and component* $i + T$ *the state* (p', γ_{i+T}). *The resulting global state* q' *corresponds to the respective configuration in* M. *The fact that the inverse of the mapping is also a homomorphism can be shown analogously.*

Remark 4 *An instance of the reachability problem is an interaction system Sys and a global state q. The interaction system* Sys_M *for a linear bounded DTM* M *and an input x can be extended such that a distinguished global state is reached if* M *halts on x. This can be achieved by a technique that is used in [19] for tree-like interaction systems. The idea is to invoke, starting from the component that reached* (p^Y, γ), *that each component shall reach a distinguished state. This invocation can be propagated through neighboring components.*

4   PSPACE-completeness of Reachability in Star-Like Systems

Here we show that deciding the reachability problem in the class of star-like interaction systems is PSPACE-complete by providing a reduction from a general interaction systems *Sys* to a star-like systems *Sys'*. The idea of the reduction is to construct a "control component" *cc* that forms the center of the star structure in *Sys'* and is surrounded by the components of *Sys*. An interaction in *Sys* is modeled by multiple interactions in *Sys'*. The execution of an interaction in *Sys* then corresponds to the execution of a sequence of interactions in *Sys'* that is coordinated by *cc* and achieved in two steps. Let α be an interaction in *Sys*.

   a) In a first step *cc* interacts with each component that participates in α and checks whether the respective port in α is enabled without changing the local states of the components. If this check fails then *cc* returns to its initial state.

   b) If the check succeeds then *cc* interacts with each respective component on the ports in α, i.e., a global transition in *Sys* that is labeled by α is simulated.

Let $Q = \prod_{i \in K} Q_i$ be the global state space of *Sys* then we have a global state space $\prod_{i \in K \cup \{cc\}} Q_i$ for *Sys'* with the property that $q \in Q$ is reachable in *Sys* iff a state q' is reachable in *Sys'* such that q' equals q up to the local state of the component *cc*. Since reachability in general interaction systems is PSPACE-complete, the consequence of this transformation is the PSPACE-completeness of reachability in star-like interaction systems.

Let $Sys = (IM, \{T_i\}_{i \in K})$ be an interaction system with interaction model $IM = (K, \{A_i\}_{i \in K}, Int)$ and $Sys' = (IM', \{T_i'\}_{i \in K'})$ be an interaction system with interaction model $IM = (K', \{A_i'\}_{i \in K'}, Int')$.

Let $K' = K \cup \{cc\}$, where *cc* is a control component that coordinates sequences of interactions in *Int'* that correspond to interactions in *Int*. For $i \in K$ let $A_i' = A_i \cup \{a_i^{ok}, a_i^{\neg ok} | a_i \in A_i\}$. a_i^{ok} respectively $a_i^{\neg ok}$ models that component *i* is in a local state that enables respectively does not enable the port $a_i \in A_i$. The set of ports A_{cc} of component *cc* is given by

$$A_{cc} = \{a\_i_{cc}^{ok}, a\_i_{cc}^{\neg ok}, a\_i_{fire}^{cc} | i = 1, \ldots, n, a_i \in A_i\} \cup \{\alpha_{cc} | \alpha \in Int\}.$$

Let $i \in K$ and $a_i \in A_i$ a port in *i* then $a\_i_{cc}^{ok}$ models that component *i* currently enables a_i and $a\_i_{cc}^{\neg ok}$ models that a_i is currently not enabled by *i*. $a\_i_{fire}^{cc}$ models that component *i* performs a transition labeled by a_i. For an interaction $\alpha \in Int$ the port α_{cc} models the initiation of a process that checks whether α is enabled by the respective components and, if applicable, coordinates that all ports in α interact one after another.

   The set of interactions *Int'* is given by

$$Int' \quad = \quad \{\{a_i^{ok}, a\_i_{cc}^{ok}\}, \{a_i^{\neg ok}, a\_i_{cc}^{\neg ok}\}, \{a_i, a\_i_{cc}^{fire}\} | a_i \in A_i, i = 1, \ldots, n\} \cup \{\{\alpha_{cc}\} | \alpha \in Int\}.$$

   The local behavior of $i \in K$ is given by $T_i' = (Q_i, A_i', \rightarrow_i', q_i^0)$ with

$$\rightarrow_i' = \quad \rightarrow_i \cup \{(q_i, a_i^{ok}, q_i) | q_i \in Q_i \wedge a_i \in en(q_i)\} \cup \{(q_i, a_i^{\neg ok}, q_i) | q_i \in Q_i \wedge a_i \notin en(q_i)\}.$$

T_i' extends T_i such that for each port $a_i \in A_i$ there is a loop on each state $q_i \in Q_i$ that is labeled by a_i^{ok} if q_i enables a_i and by $a_i^{\neg ok}$ otherwise. These transitions are used to check whether or not each port of an interaction $\alpha \in Int$ is enabled in a global state of *Sys'* without changing the local state of the respective components.

   Let $\alpha^j = \{a_{j_1}, \ldots, a_{j_{|\alpha^j|}}\} \in Int$. Figure 5 depicts the part of the local behavior $T_{cc} = (Q_{cc}, A_{cc}, \rightarrow_{cc}, q_{cc}^0)$ of component *cc* that coordinates a test that checks whether each port in α^j is enabled in *Sys'* and, if applicable, enables ports that can interact with each port in α^j. q_{cc}^0 is marked by an incoming arrow.
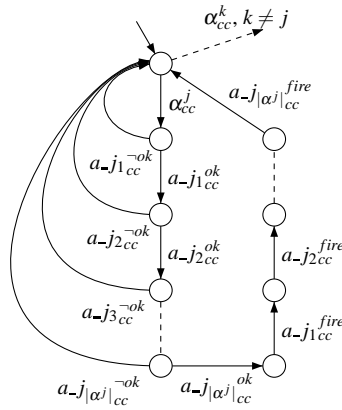

Figure 5: Parts of the behavior of component *cc*.

Remark 5 *Each port of Sys' occurs in at least one interaction, i.e., Sys' satisfies the conditions of an interaction system. It is clear that Sys' is star-like because each component that originated from Sys interacts only with the control component cc.*

Furthermore, the size of Sys' is polynomial in the size of Sys. $|K'| = |K| + 1$, $|Int'| = |Int| + \sum_{i \in K} 3 \cdot |A_i|$ and for $i \in K$ holds $|A'_i| = 3 \cdot |A_i|$ and $| \to'_i | = | \to_i | + |Q_i| \cdot |A_i|$. For $cc \in K'$ holds $|A_{cc}| = |Int| + \sum_{i \in K} 3 \cdot |A_i|$, $|Q_{cc}| = 1 + \sum_{\alpha \in Int} 2 \cdot |\alpha|$ and $| \to_{cc} | = \sum_{\alpha \in Int} (3 \cdot |\alpha| + 1)$.

Theorem 2 *Let Sys be an interaction system with components K and Sys' the associated star-like interaction system. A global state q is reachable in Sys iff a global state q' is reachable in Sys' such that $q_i = q'_i$ for $i \in K$ and $q'_{cc} = q^0_{cc}$.*

Proof 2 *Let q be a state in the global behavior T of Sys and q' be the state in the global behavior T' of Sys' where $q_i = q'_i$ for $i \in K$ and $q'_{cc} = q^0_{cc}$, i.e., component cc is in its initial state. Consider $\alpha^j = \{a_{j_1}, \ldots, a_{j_{|\alpha^j|}}\} \in Int$ such that each port in α^j is enabled in q, i.e., all local states q'_l, $l = j_1, \ldots, j_{|\alpha^j|}$ in q' enable the ports a_l^{ok} and a_l and do not enable $a_l^{\neg ok}$. q' enables the interaction $\{\alpha^j_{cc}\}$. If this interaction is performed then the only possible sequence of interactions results in a state $\tilde{q}'$ with $\tilde{q}_i = \tilde{q}'_i$ for $i \in K$ and $\tilde{q}'_{cc} = q^0_{cc}$. Let there be a port in α^j that is not enabled in q, e.g., q_l with $l \in \{j_1, \ldots, j_{|\alpha^j|}\}$ does not enable a_l then q'_l does enable $a_l^{\neg ok}$ and not a_l^{ok}. If $\{\alpha^j_{cc}\}$ performed in q' then the only possible sequence of interactions in Sys' leads back to state q'. For the global initial states q^0 of Sys and $q^{0'}$ of Sys' holds that $q^0_i = q^{0'}_i$ for $i \in K$ and $q^{0'}_{cc}$ is the initial state of the local behavior of component cc. The "if" part follows by induction over paths in the global behavior of Sys. The "and only if" part follows analogously.*

5   Conclusion and Related Work

We investigated complexity issues for classes of interaction systems that are relevant in various applications. One with a linear the other with a star-like communication pattern. We showed that even for these simply structured systems deciding the reachability problem is PSPACE-complete. These results strengthen PSPACE-completeness results of the reachability problem in general interaction systems [18]. The formalism of interaction systems is very basic, and thus our results are easily applicable to other formalisms that model cooperating systems. Our results justify techniques that are based on a sufficient condition and establish reachability or reachability dependent system properties in subclasses of cooperating systems that are defined by a restricted communication structure that forms a star or a line or in respective superclasses, which are sketched in the following.

[4] examined a process algebra based on an architectural description language called *PADL* and considers deadlock-freedom in systems with a tree-like communication pattern (a proper superclass of systems with a star-like or linear pattern). The technique is based on a compatibility condition that is tested among pairs of cooperating subsystems, i.e., the composite behavior of two subsystems is weak bisimilar to the behavior of one of the components. An efficient technique based on a sufficient conditions for establishing deadlock-freedom in interaction systems with a star-like communication pattern is introduced in [14] where, similar to [4], a compatibility condition based on branching bisimilarity is tested. A sufficient condition for establishing deadlock-freedom for the subclass of tree-like interaction systems is described in [16] where a condition is tested on the reachable state spaces of pairs of interacting components. In [15] the condition in [16] is extended such that deadlock-freedom can be established in a proper superclass of tree-like interaction systems. Hennicker et al. proposed in [2, 11] a technique to construct so called *observable* behavior of a cooperating system with an acyclic communication pattern

which can be used to establish certain system properties. [6] describes a general communication graph for CSP models and shows how tree structures can be constructed by merging several processes. *Communicating Sequential Processes* are introduced in [12] where a directed communication structure based on input/output communication is considered. It is argued that communicating processes, if a directed input/output communication structure forms a rooted tree, can not deadlock.

References

[1] Luca de Alfaro & Thomas A. Henzinger (2001): *Interface Automata*. SIGSOFT Softw. Eng. Notes 26(5), pp. 109–120, doi:10.1145/503271.503226.

[2] Hubert Baumeister, Florian Hacklinger, Rolf Hennicker, Alexander Knapp & Martin Wirsing (2006): *A Component Model for Architectural Programming*. Electr. Notes Theor. Comput. Sci. 160, pp. 75–96, doi:10.1016/j.entcs.2006.05.015.

[3] Saddek Bensalem, Marius Bozga, Joseph Sifakis & Thanh-Hung Nguyen (2008): *Compositional Verification for Component-Based Systems and Application*. In: *Proceedings of the 6th International Symposium on Automated Technology for Verification and Analysis*, ATVA '08, Springer-Verlag, Berlin, Heidelberg, pp. 64–79, doi:10.1007/978-3-540-88387-6_7.

[4] Marco Bernardo, Paolo Ciancarini & Lorenzo Donatiello (2002): *Architecting Families of Software Systems with Process Algebras*. ACM Trans. Softw. Eng. Methodol. 11(4), pp. 386–426, doi:10.1145/606612.606614.

[5] Borzoo Bonakdarpour, Marius Bozga & Gregor Goessler (2011): *A Theory of Fault Recovery for Component-Based Models*. In: *Proceedings of the 2011 IEEE 30th International Symposium on Reliable Distributed Systems*, SRDS '11, IEEE Computer Society, Washington, DC, USA, pp. 265–270, doi:10.1109/SRDS.2011.39.

[6] Stephen D. Brookes & A. W. Roscoe (1991): *Deadlock Analysis in Networks of Communicating Processes*. Distributed Computing 4, pp. 209–230, doi:10.1007/BF01784721.

[7] Allan Cheng, Javier Esparza & Jens Palsberg (1993): *Complexity results for 1-safe nets*. In: *Proceedings of FSTTCS'93*, LNCS 761, pp. 326–337, doi:10.1007/3-540-57529-4_66.

[8] Michael R. Garey & David S. Johnson (1979): *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.

[9] Gregor Gössler, Sussane Graf, Mila Majster-Cederbaum, M. Martens & Joseph Sifakis (2007): *An Approach to Modelling and Verification of Component Based Systems*. In: *Proceedings of the 33rd conference on Current Trends in Theory and Practice of Computer Science*, SOFSEM '07, Springer-Verlag, Berlin, Heidelberg, pp. 295–308, doi:10.1007/978-3-540-69507-3_24.

[10] Gregor Gössler & Joseph Sifakis (2005): *Composition for Component-Based Modeling*. Sci. Comput. Program. 55(1-3), pp. 161–183, doi:10.1016/j.scico.2004.05.014.

[11] Rolf Hennicker, Stephan Janisch & Alexander Knapp (2010): *On the Observable Behaviour of Composite Components*. Electron. Notes Theor. Comput. Sci. 260, pp. 125–153, doi:10.1016/j.entcs.2009.12.035.

[12] Charles A. R. Hoare (1985): *Communicating Sequential Processes*. Prentice-Hall International series in computer science, Englewood Cliffs, NJ [u.a.].

[13] Neil D. Jones, Lawrence H. Landweber & Y. Edmund Lien (1977): *Complexity of Some Problems in Petri Nets*. Theor. Comput. Sci. 4(3), pp. 277–299, doi:10.1016/0304-3975(77)90014-7.

[14] Christian Lambertz (2009): *Exploiting Architectural Constraints and Branching Bisimulation Equivalences in Component-Based Systems*. In: *Proceedings of the Doctoral Symposium on FM2009*, University of Eindhoven Technical Report 0915, pp. 1–7. Available at `http://www.win.tue.nl/~mousavi/fm09ds.pdf`.

[15] Christian Lambertz & Mila Majster-Cederbaum (2012): *Analyzing Component-Based Systems on the Basis of Architectural Constraints*. In: *Proceedings of the 4th IPM international conference on Fundamentals of Software Engineering*, FSEN'11, Springer-Verlag, Berlin, Heidelberg, pp. 64–79, doi:`10.1007/978-3-642-29320-7_5`.

[16] Mila Majster-Cederbaum & Moritz Martens (2008): *Compositional Analysis of Deadlock-Freedom for Tree-Like Component Architectures*. In: *Proceedings of the 8th ACM international conference on Embedded software*, EMSOFT '08, ACM, New York, NY, USA, pp. 199–206, doi:`10.1145/1450058.1450085`.

[17] Mila Majster-Cederbaum & Christoph Minnameier (2008): *Deriving Complexity Results for Interaction Systems from 1-Safe Petri Nets*. In: *Proceedings of the 34th conference on Current trends in theory and practice of computer science*, SOFSEM'08, Springer-Verlag, Berlin, Heidelberg, pp. 352–363, doi:`10.1007/978-3-540-77566-9_30`.

[18] Mila Majster-Cederbaum & Christoph Minnameier (2008): *Everything Is PSPACE-Complete in Interaction Systems*. In: *Proceedings of the 5th international colloquium on Theoretical Aspects of Computing*, Springer-Verlag, Berlin, Heidelberg, pp. 216–227, doi:`10.1007/978-3-540-85762-4_15`.

[19] Mila Majster-Cederbaum & Nils Semmelrock (2010): *Reachability in Tree-Like Component Systems is PSPACE-Complete*. Electron. Notes Theor. Comput. Sci. 263, pp. 197–210, doi:`10.1016/j.entcs.2010.05.012`.

[20] Moritz Martens & Mila Majster-Cederbaum (2009): *Using Architectural Constraints for Deadlock-Freedom of Component Systems with Multiway Cooperation*. In: *Proceedings of the 2009 Third IEEE International Symposium on Theoretical Aspects of Software Engineering*, TASE '09, IEEE Computer Society, Washington, DC, USA, pp. 225–232, doi:`10.1109/TASE.2009.18`.