Information and Computation ••• (••••) •••–•••

Contents lists available at ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

Petri games: Synthesis of distributed systems with causal memory ☆

Bernd Finkbeiner [a,*], Ernst-Rüdiger Olderog [b,*]

[a] Department of Computer Science, Universität des Saarlandes, Saarbrücken, Germany
[b] Department of Computing, Universität Oldenburg, Oldenburg, Germany

ARTICLE INFO

Article history:
Received 15 April 2015
Available online xxxx

Keywords:
Petri nets
Causality
Unfolding
Cuts
Strategies
Graph games
Synthesis

ABSTRACT

We present a new multiplayer game model for the interaction and the flow of information in a distributed system. The players are tokens on a Petri net. As long as the players move in independent parts of the net, they do not know of each other; when they synchronize at a joint transition, each player gets informed of the causal history of the other player. We show that for Petri games with a single environment player and an arbitrary bounded number of system players, deciding the existence of a safety strategy for the system players is EXPTIME-complete.

1. Introduction

Games are a natural model of the interaction between a computer system and its environment. Specifications are interpreted as winning conditions, implementations as strategies. An implementation is correct if the strategy is *winning*, i.e., it ensures that the specification is met for all possible behaviors of the environment. Algorithms that determine the winner in the game between the system and its environment can be used to determine whether it is possible to implement a specification (the *realizability* question) and, if the answer is yes, to automatically construct a correct implementation (the *synthesis* problem).

We present a new game model for the interaction and the flow of information in a distributed system. The players are tokens on a Petri net. In Petri nets, causality is represented by the flow of tokens through the net. It is therefore natural to designate tokens also as the carriers of information. As long as different players move in concurrent places of the net, they do not know of each other. Only when they synchronize at a joint transition, each player gets informed of the history of the other player, represented by all places and transitions on which the joint transition causally depends. The idea is that after such a joint transition, a strategy for a player can take the history of all other players participating in the joint transition into account. Think of a workflow where a document circulates in a large organization with many clerks and has to be signed by everyone, endorsing it or not. Suppose a clerk wants to make the decision whether or not to endorse it depending on who has endorsed it already. As long as the clerk does not see the document, he is undecided. Only when he receives the document, he sees all previous signatures and then makes his decision.

☆ This research was partially supported by the German Research Council (DFG) in the Transregional Collaborative Research Center SFB/TR 14 AVACS. The paper is a revised and extended version of [1].

* Corresponding authors.
E-mail addresses: finkbeiner@cs.uni-saarland.de (B. Finkbeiner), olderog@informatik.uni-oldenburg.de (E.-R. Olderog).

http://dx.doi.org/10.1016/j.ic.2016.07.006
0890-5401/

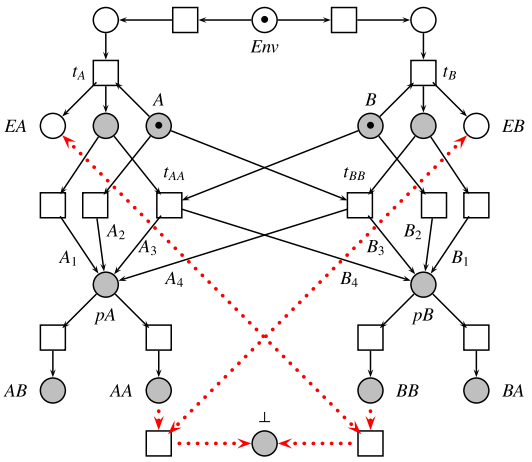

Fig. 1. Introductory example of a Petri game modeling a distributed security alarm. Places belonging to the system players A and B are shown in gray. In the Petri game, the transitions to the bad place $\perp$ are shown with dotted lines.

We call our extension of Petri nets *Petri games*. The players are organized into two teams, the system players and the environment players, where the system players wish to avoid a certain "bad" place (i.e., they follow a safety objective), while the environment players wish to reach just such a place. To partition the tokens into the teams, we label each place as belonging to either the system or the environment. A token belongs to a team whenever it is on a place that belongs to the team.

In the tradition of Zielonka's asynchronous automata [2], Petri games model distributed systems with *causal memory*, i.e., distributed systems where the processes memorize their causal history and communicate it to each other during each synchronization [3–5]. Petri games thus abstract from the concrete content of a communication in that we assume that the processes always exchange the *maximal* possible information, i.e., their entire causal history. This is useful at a design stage before the details of the interface have been decided and one is more interested in restricting *when* a communication can occur (e.g., when a device is connected to its base station, while a network connection is active, etc.) than *what* may be communicated. The final interface is then determined by the information actually used by the winning strategies, which is typically only a small fraction of the causal history. Note that even though we assume the players to communicate everything they know, the flow of information in a Petri game is far from trivial. At any point, the players of the Petri game may have a different level of knowledge about the global state of the game, and the level of informedness changes dynamically as a result of the synchronizations chosen by the players.

Consider the development of a distributed security alarm system. If a burglar triggers the alarm at one location, the alarm should go off everywhere, and all locations should report the location where the original alarm occurred. This situation is depicted as a Petri net in Fig. 1. The token that initially resides on place *Env* represents the environment, which is, in our example, the burglar, who can decide to break into our building either at location A or B. The tokens that initially reside on places A and B represent the distributed controller consisting of two processes, the one on the left for location A and the one on the right for location B. In the following, we will refer to the Petri net of Fig. 1 as a *Petri game*, to emphasize that the tokens in fact represent players and that the nondeterminism present in the net is to be restricted by the (yet to be determined) strategy of the system players.

The system players and the environment players move on separate places in the net, the places belonging to the system players are shown in gray. In the example, our goal is to find a strategy for the system players that avoids a *false alarm*, i.e., a marking where the environment token is still on *Env* and at least one system token is on one of the places at the bottom, i.e., *AA*, *AB*, etc., and a *false report*, i.e., a marking where the environment token is on place *EA* and some system token is on *AB* or *BB* or a marking where the environment token is on *EB* and some system token is on *AA* or *BA*. To identify such undesirable markings we introduce a distinguished place $\perp$. Fig. 1 shows (dashed) transitions towards $\perp$ firing at two instances of false reports, when tokens are on both *EA* and *BB* or on both *EB* and *AA*. Similar transitions for other erroneous situations are omitted here to aid visibility.

Suppose that, in our Petri game, the burglar breaks into location A by taking the left transition. Once the system token in A has recorded this via transition t_A, it has two possibilities: either synchronize with the system token in B by taking transition t_{AA}, or skip the communication and go straight to pA via transition A_1. Intuitively, only the choice to synchronize is a good move, because the system token in B has no other way of hearing about the alarm. The only remaining move for the system token in B would be to move "spontaneously" via transition B_2 to pB, at which point it would need to move to *BA*, because the combination of *BB* and *EA* would constitute a false report. However, the token in pB has no way of distinguishing this situation from one where the environment token is still on *Env*; in this situation, the move to *EA* would reach a false alarm.

Please cite this article in press as: B. Finkbeiner, E.-R. Olderog, Petri games: Synthesis of distributed systems with causal memory, Inf. Comput. (2016), http://dx.doi.org/10.1016/j.ic.2016.07.006