Acceptance modes for asynchronous cellular automata for pomsets and first order logic[1]

Dietrich Kuske[2]
Institut für Algebra
Technische Universität Dresden
D-01062 Dresden
email: kuske@math.tu-dresden.de

In a distributed system, some events may occur concurrently, meaning that they may occur in any order or simultaneously or even that their executions may overlap. This is the case in particular when two events use independent resources. On the other hand, some events may causally depend on each other. Therefore, a distributed behaviour may be abstracted as a pomset, that is a set together with a partial order which describes causal dependencies of events and with a labeling function. In this paper, we consider pomsets without autoconcurrency, i.e. events with the same label are linearly ordered. These pomsets are called semi-words in [St81, Di94]. For studies how general pomsets can be used to represent parallel processes and how they can be composed, we refer the reader e.g. to [Pr87, Gi88].

There are several ways to describe the behaviour of a system. For instance, logic formulas are suited for specification issues. Depending on the properties one has to express, we can use various logics such as temporal logics, the first order logic or the (monadic) second order logic.

When dealing with distributed systems, it is natural to look for transition systems that faithfully reflect the concurrency. For instance, Petri nets are a widely studied class of such transition systems. Asynchronous cellular automata (ACA) form another fundamental class of transition systems with built-in concurrency. They were introduced for Mazurkiewicz traces by Zielonka [Zi87, Zi89]. A trace ([CF69, Ma77, Ma87]) is a pomset where the partial order is dictated by a static dependence relation over the actions of the system.

In [DG96], Droste & Gastin generalize the notion of ACA's so that they can work on pomsets without autoconcurrency. They consider the relation between monadic second order logic on pomsets, ACA's and deterministic ACA's. Their first result states that any set of pomsets recognized by an ACA can be defined by an existential sentence of the monadic second order (MSO) logic. The question, whether the converse holds in general is left open. It is proved for a large class of pomsets, called CROW-pomsets. Actually, Droste & Gastin were able to construct from an arbitrary MSO-sentence a *deterministic* ACA which accepts precisely the CROW-pomsets satisfying the formula. Thus, they proved the equivalence between MSO logic, existential MSO logic, deterministic ACA's and nondeterministic ACA's.

In [DG96, Remark 3.3] the authors shortly discuss several alternative definitions of acceptance by ACA's and state: "In fact, these differences are not crucial and we will see ... that under some assumptions they are equivalent." Therefore, they mainly restrict their attention to only one of the four suggested definitions.

While, as remarked above, the question whether any MSO-sentence can be translated into an equivalent ACA is left open in [DG96], we answer it negatively for all possible acceptance modes of ACA's in this paper. More precisely, we show that there exists a first order sentence

[1]For the complete paper see [Ku97]

[2]Supported by the German Research Foundation (DFG).

such that the set of all pomsets satisfying this sentence is not recognizable by an ACA. Since, as we also show, the complement of this set is recognizable in any mode, we obtain as a corollary that for any acceptance mode the class of recognizable pomset languages is not closed under complement. Hence not every nondeterministic ACA can be transformed into an equivalent deterministic one. A similar result has been obtained by Gastin for one of the acceptance modes [Ga].

Furthermore, we show that the differences between the acceptance modes become crucial if we consider the set of all pomsets. More precisely, we show that they yield mutually different classes of recognizable pomset languages and we completely determine the inclusion structure between these classes.

References

[CF69] P. Cartier and D. Foata: *Problemes combinatoires de commutation et rearrangements.* Lecture Notes in Mathematics 85, 1969.

[Di94] V. Diekert: *A partial trace semantics for petri nets.* Theoretical Computer Science **113** (1994), pp. 87-105.

[DG96] D. Droste, P. Gastin: *Asynchronous cellular automata for pomsets without auto-concurrency.* In: CONCUR 96. Lecture Notes in Computer Science 1119, pp. 627-638, 1996.

[Gi88] J.L. Gischer: *The equational theory of pomsets.* Theoretical Computer Science **61** (1988), pp. 199-224.

[Ga] P. Gastin: private communication.

[Ku97] D. Kuske: *Comparing asynchronous cellular automata for pomsets and first order logic.* Technical Report, TU Dresden, `http://www.math.tu-dresden.de/~kuske/abstracts/aca.html`, 1997.

[Ma77] A. Mazurkiewicz: *Concurrent program schemes and their interpretation.* Tech. rep. DAIMI PB 78, Aarhus University, 1977.

[Ma87] A. Mazurkiewicz: *Trace Theory.* In: Advances in Petri Nets 86 (W. Brauer et al., eds.), Lecture Notes in Computer Science 255, pp. 279-324, 1987.

[Pr87] V.R. Pratt: *Modelling concurrency with partial orders.* J. of Parallel Programming **15** (1987), pp. 33-71.

[St81] P.H. Starke: *Processes in Petri Nets.* EIK **17** (1981), pp. 389-416.

[Zi87] W. Zielonka: *Notes on finite asynchronous automata.* R.A.I.R.O. – Informatique Théorique et Applications **21** (1987), pp. 99-135.

[Zi89] W. Zielonka: *Safe executions of recognizable trace languages by asynchronous automata.* In: Logical Foundations of Computer Science (A.R. Meyer et al., eds.), Lecture Notes in Computer Science 363, pp. 278-289, 1989.