Journal of Automata, Languages and Combinatorics **21** (2016) 1–2, 7 – 26

SMALL UNIVERSAL DETERMINISTIC PETRI NETS WITH INHIBITOR ARCS

Artiom Alhazov[(A)] Sergiu Ivanov[(B)] Elisabeth Pelz[(B)]
Sergey Verlan[(B)]

[(A)] *Institute of Mathematics and Computer Science, Academy of Sciences of Moldova*
Academiei 5, MD-2028, Chişinău, Moldova
artiom@math.md

[(B)] *Laboratoire d'Algorithmique, Complexité et Logique, Université Paris Est*
61, av. du gén. de Gaulle, F-94010 Créteil, France
sergiu.ivanov@u-pec.fr pelz@u-pec.fr verlan@u-pec.fr

ABSTRACT

This paper describes a series of small universal Petri nets with inhibitor arcs. Four parameters of descriptional complexity are considered: the number of places, transitions, inhibitor arcs, and the maximal degree of a transition. A number of techniques for reducing the values of these parameters, with special attention on places, are presented: we describe strongly universal Petri nets with 30, 21, 14, 11, and 5 places, and weakly universal Petri nets with similar parameters. We also show a universal Petri net with 2 inhibitor arcs only. Our investigation highlights several trade-offs. Due to equivalence of the corresponding models, our results can be immediately translated to multiset rewriting with forbidding conditions, to P systems with cooperative rules and inhibitors, or to vector addition systems.

Keywords: Petri nets, universality, descriptional complexity

1. Introduction

Universality is an important concept in the theory of computation. The question of finding a universal computing device in the class of Turing machines was originally proposed by A. Turing himself in [22]. A universal Turing machine would be capable of simulating any other Turing machine $\mathcal{T}$: given a description of $\mathcal{T}$ and the encoding of the input tape contents, the universal machine would halt with tape contents which would correspond to the encoding of the output of $\mathcal{T}$ for the supplied input.

In a more general setting of an class arbitrary class $\mathfrak{C}$ of computing devices, the universality problem consists in finding such a *fixed* element $\mathcal{M}_0 \in \mathfrak{C}$ which would be able to simulate any other element $\mathcal{M} \in \mathfrak{C}$. More formally, if the result of running $\mathcal{M}$ with the input x is y (usually written as $M(x) = y$), then $y = f(\mathcal{M}_0(\langle g(\mathcal{M}'), h(x)\rangle))$, where g is the function enumerating $\mathfrak{C}$, $\langle\rangle$ is some couple encoding (e. g. the Cantor pairing function), while f and h are the decoding and encoding functions respectively. Although technically this definition does not impose any restriction on the encoding