On the Computational Power of Timed Differentiable Petri Nets

Serge Haddad[1], Laura Recalde[2], and Manuel Silva[2]

[1] LAMSADE-CNRS UMR 7024, University Paris-Dauphine, France
haddad@lamsade.dauphine.fr
[2] GISED, University Zaragossa, Spain
{lrecalde, silva}@unizar.es

Abstract. Well-known hierarchies discriminate between the computational power of discrete time and space dynamical systems. *A contrario* the situation is more confused for dynamical systems when time and space are continuous. A possible way to discriminate between these models is to state whether they can simulate Turing machine. For instance, it is known that continuous systems described by an ordinary differential equation (ODE) have this power. However, since the involved ODE is defined by overlapping local ODEs inside an infinite number of regions, this result has no significant application for differentiable models whose ODE is defined by an explicit representation. In this work, we considerably strengthen this result by showing that Time Differentiable Petri Nets (TDPN) can simulate Turing machines. Indeed the ODE ruling this model is expressed by an explicit linear expression enlarged with the "minimum" operator. More precisely, we present two simulations of a two counter machine by a TDPN in order to fulfill opposite requirements: robustness and boundedness. These simulations are performed by nets whose dimension of associated ODEs is constant. At last, we prove that marking coverability, submarking reachability and the existence of a steady-state are undecidable for TDPNs.

1 Introduction

Hybrid systems. Dynamic systems can be classified depending on the way time is represented. Generally, trajectories of discrete-time systems are obtained by iterating a transition function whereas the ones of continuous-time systems are often solutions of a differential equation. When a system includes both continuous and discrete transitions it is called an *hybrid system*. On the one hand, the expressive power of hybrid systems can be strictly greater than the one of Turing machines (see for instance [12]). On the other hand, in restricted models like timed automata [1], several problems including reachability can be checked in a relatively efficient way (i.e. they are $PSPACE$-complete). The frontier between decidability and undecidability in hybrid systems is still an active research topic [8,10,4,11].

Continuous systems. A special kind of hybrid systems where the trajectories are continuous (w.r.t. standard topology) and right-differentiable functions of

E. Asarin and P. Bouyer (Eds.): FORMATS 2006, LNCS 4202, pp. 230–244, 2006.

time have been intensively studied. They are defined by a finite number regions and associated ordinary differential equations ODEs such that inside a region r, a trajectory fulfills the equation $\dot{x}_d = f_r(x)$ where x is the trajectory and $\dot{x}_d$ its right derivative. These additional requirements are not enough to limit their expressiveness. For instance, the model of [2] has piecewise constant derivatives inside regions which are polyhedra and it is Turing equivalent if its space dimension is at least 3 (see also [3,5] for additional expressiveness results).

Differentiable systems. A more stringent requirement consists in describing the dynamics of the system by a single ODE $\dot{x} = f(x)$ where f is continuous, thus yielding continuously differentiable trajectories. We call such models, *differentiable systems*. In [6], the author shows that differentiable systems in $\mathbb{R}^3$ can simulate Turing machine. The corresponding ODE is obtained by extrapolation of the transition function of the Turing machine over every possible configuration. Indeed such a configuration is represented as a point in the first dimension of the ODE (and also in the second one for technical reasons) and the third dimension corresponds to the time evolution. The explicit local ODE around every representation of a configuration is computed from this configuration and its successor by the Turing machine. Thus the explicit equations of the ODE are piecewise defined inside *an infinite number regions* which is far beyond the expressiveness of standard ODE formalisms used for the design and analysis of dynamical systems. So the question to determine which (minimal) set of operators in an explicit expression of f is required to obtain Turing machine equivalence, is still open.

Our contribution. In this work, we (partially) answer this question by showing that Time Differentiable Petri Nets (a model close to Time Continuous Petri Nets [7,13]) can simulate Turing machines. Indeed the ODE ruling this model is particularly simple. First its expression is a linear expression enlarged with the “minimum” operator. Second, it can be decomposed into a finite number of linear ODEs $\dot{x} = \mathbf{M} \cdot x$ (with $\mathbf{M}$ a matrix) inside polyhedra.

More precisely, we present two simulations of two counter machines in order to fulfill opposite requirements: robustness (allowing some perturbation of the simulation) and boundedness of the simulating net system. Our simulation is performed by a net with a constant number of places, i.e. whose dimension of its associated ODE is constant (in $(\mathbb{R}_{\geq 0})^6$ for robust simulation and in $[0, K]^{14}$ for bounded simulation). Afterwards, by modifying the simulation, we prove that marking coverability, submarking reachability and the existence of a steady-state are undecidable for (bounded) TDPNs.

Outline of the paper. In section 2, we recall notions of dynamical systems and simulations. In section 3, we introduce TDPNs. Then we design a robust simulation of counter machines in section 4 and a bounded one in section 5. Afterwards, we establish undecidability results in section 6. At last, we conclude and give some perspectives to this work.