12th IFAC/IEEE Workshop on Discrete Event Systems
Cachan, France. May 14-16, 2014

Model Checking Logical Regulatory Networks ⋆

P.T. Monteiro * **W. Abou-Jaoudé** ** **D. Thieffry** ***
C. Chaouiya ****

* *Instituto de Engenharia de Sistemas e Computadores - Investigação e Desenvolvimento (INESC-ID), Lisbon, Portugal (e-mail: ptgm@kdbio.inesc-id.pt)*
** *Institut de Biologie de l'Ecole Normale Supérieure (IBENS) - UMR CNRS 8197 - INSERM 1024, Paris, France (e-mail: wassim@biologie.ens.fr)*
*** *Institut de Biologie de l'Ecole Normale Supérieure (IBENS) - UMR CNRS 8197 - INSERM 1024, Paris, France (e-mail: thieffry@ens.fr)*
**** *Instituto Gulbenkian de Ciência (IGC), Oeiras, Portugal (e-mail: chaouiya@igc.gulbenkian.pt)*

Abstract: Regulatory and signalling networks control cell behaviours in response to environmental cues. The logical formalism has been widely employed to study these interaction networks, which are modelled as discrete dynamical systems. While biologists identify networks encompassing more and more components, properties of biological relevance become hard to verify. Here, we report on the use of model-checking techniques to address this challenge. This approach is illustrated by an application dealing with the modelling of T-helper lymphocyte differentiation.

Keywords: Systems Biology, Formal Verification, Discrete Event Systems, Biological networks, Regulatory Networks, Signalling Networks, Logical Modelling, Cell Differentiation, T-helper lymphocyte

1. INTRODUCTION

Cellular processes such as cell proliferation and differentiation are governed by molecular regulatory networks. As technological advances facilitate the identification of these complex networks, mathematical models are needed to study their functioning (de Jong (2002); Schlitt and Brazma (2007)). In this context, we focus on the formalism initially introduced by R. Thomas and collaborators (Thomas and D'Ari (1990)). In short, models are defined as Logical Regulatory Graphs that encompass nodes (the regulatory components) and directed edges (regulatory interactions). Moreover, nodes are associated with discrete variables, which embody regulatory component levels of activity. For each node, a logical function defines the target value of the corresponding variable, depending on its regulators (for further details see Thomas and D'Ari (1990); Chaouiya et al. (2003)). These models lead to discrete (a)synchronous dynamics, conveniently represented by State Transition Graphs (STG), where nodes and arcs denote states and transitions, respectively.

Properties of interest mainly relate to the attractors and their reachability. An attractor is defined as a terminal strongly connected component of the STG, i.e., a strongly connected component with no transition leaving it. Attractors capture potential long term behaviours of the regulatory network and refer to stable states (e.g. stable pattern of gene expression) or sustained oscillations. Not surprisingly, combinatorial explosion hinders efficient analysis of large networks dynamical properties, in particular concerning attractor reachability.

Here, we focus on the use of model-checking and related software tools for the analysis of dynamical properties of biological networks. As illustrated with a model for lymphocyte differentiation, logical models are particularly amenable to formal verification approaches (Naldi et al. (2010)). In addition to classical reachability properties, we show how novel questions concerning the structure of a regulatory network can be efficiently addressed.

2. MODEL-CHECKING

As models of regulatory networks grow in size, they tend to generate more complex behaviours. Determining whether a biological model satisfies a set of observed biological properties is thus often manually untractable. This motivates the use of automated formal verification techniques capable of verifying qualitative systems generating very large state transition graphs up to 10^{20} states (Burch et al. (1990)). One of these techniques is model-checking (Clarke et al. (2000)).

Model-checking is a computer science technique, which has been proposed 30 years ago to solve the problem of

⋆ W. Abou-Jaoudé was supported by the LabEx MemoLife (`http://www.memolife.biologie.ens.fr`). P.T. Monteiro was supported by FCT grants PEst-OE/EEI/LA0021/2013 and IF/01333/2013.

verification of very large systems in hardware and software industry. Since then, many improvements and variations have been developed as application fields diversified.

2.1 Application to Biological Systems

During the last decade, model-checking techniques have been successfully applied to the broad field of systems biology (Chabrier-Rivier et al. (2004); Bernot et al. (2004); Batt et al. (2005)), with variations in the mathematical modelling formalisms and/or the type of properties to be verified. Most of the formalisms considered to model biological networks are already discrete or can be discretised under a suitable abstraction criterion, therefore permitting the generation of a STG representing the transitions between the qualitative states of the system. A STG can be directly mapped into a Kripke structure, as used in model-checking, through a direct correspondance of their states and a labelling denoting the values of the components in each state. Temporal logic formulas can then be verified on such structures.

Signalling networks differ from gene regulatory networks by the consideration of input signals imposing restrictions on the dynamics. From the modelling perspective, these restrictions apply to paths in the STG that only occur for specific input valuations. As a consequence, both the characterisation of the attractors and their reachability become highly dependent on the input signals. Additionally, input variables which do not have an associated regulatory function, remain constant throughout the simulation.

From the model-checking perspective, the verification of properties is performed over Kripke structures where all the variables define the current state of the system. For models with (fixed) input variables, the STG is organised into disconnected sub-graphs, one for each complete instantiation of the inputs. Indeed, because these variables are maintained constant, no transitions allow the system to go from one state to another that differs on the values of input variables. This leads to a state-space size proportional to the number of input variables and their possible values. However, one can compress this state-space by using transition labels to specify the input values for which transitions are enabled. Only non-input variables are then explicitly denoting the states of the system (Naldi et al. (2012)).

In the model-checking community, a structure combining labels on both states and transitions is called a Kripke Transition System. Pecheur and Raimondi augmented the classical CTL semantics to account for the existence of actions, called Action Restricted CTL (ARCTL) (Lomuscio et al. (2007)). ARCTL includes the same temporal operators as CTL, except that the paths can be restricted with a given action formula α. The syntax and semantics of ARCTL is described in Table 1. This was implemented as an extension to the known NuSMV model-checking tool [1], being particularly useful for the verification of the behaviour of models influenced by fixed/varying input variables.

[1] NuSMV-ARCTL is freely available for download at `http://lvl.info.ucl.ac.be/Tools/NuSMV-ARCTL-TLACE`.

Table 1. Syntax and semantics of canonical ARCTL temporal operators used in this work (for a complete description see Lomuscio et al. (2007)); α is a path restriction defined by input valuations; ϕ and ψ denote internal component valuations at a given state.

Syntax	Semantics
$\mathsf{EAF}(\alpha)(\phi)$	ϕ has to hold eventually at some future state for some α-restricted path
$\mathsf{AAF}(\alpha)(\phi)$	ϕ has to hold eventually at some future state for all α-restricted paths
$\mathsf{EAG}(\alpha)(\phi)$	ϕ has to hold along the subsequent path for some α-restricted path
$\mathsf{AAG}(\alpha)(\phi)$	ϕ has to hold along the subsequent path for all α-restricted paths
$\mathsf{EA}(\alpha)[\phi\ \mathsf{U}\ \psi]$	ϕ has to hold along the subsequent path until ψ holds, for some α-restricted path
$\mathsf{AA}(\alpha)[\phi\ \mathsf{U}\ \psi]$	ϕ has to hold along the subsequent path until ψ holds, for all α-restricted paths

Due to the range of existing model-checkers and temporal logics, the adoption of this technique for the analysis of biological systems remains a difficult task for non-expert users. As a consequence, in the field of formal verification, recurrent properties have been grouped into high-level patterns to help non-expert users formulate temporal-logic queries (Dwyer et al. (1999)). In systems biology, Chabrier-Rivier et al. (2004) enumerated properties on reachability, pathways and stability of a discrete system. Bernot et al. (2004) tested necessary conditions leading to a particular state of a bistable system. Batt et al. (2005) also tested conditions leading to a given state, imposing restrictions on sequences of events along the path. Although different, these properties all share the same conceptual form: they correspond to reachability properties verifying the existence of a path between a (set of) initial state(s) and a (set of) reachable state(s). This has motivated the definition of patterns for recurrent biological properties to ease their formal verification (Monteiro et al. (2008)).

2.2 Modelling and verification tools

Here, we survey the main model-checking tools that have been applied to biological networks.

NuSMV is a symbolic model-checker based on Binary Decision Diagrams that provides a description language to specify generic finite state machines, supporting modules and processes, and verification through a set of temporal logic CTL or LTL formulas (Cimatti et al. (2002)).

Construction and Analysis of Distributed Processes (CADP) is a toolbox for the design of asynchronous concurrent systems (Garavel et al. (2007)), containing a powerful model-checker for temporal logics with highly expressive power, like CTRL (Mateescu et al. (2011)) and others. It allows on-the-fly model-checking and diagnostic generation over labeled transition systems, i.e., system information is represented on the transitions rather than on the states.

Analysis of Networks through TEmporal-LOgic sPEcifications (ANTELOPE) is a model-checker for the analy-