17th European Symposium on Computer Aided Process Engineering – ESCAPE17
V. Plesu and P.S. Agachi (Editors)

Automatic Synthesis of Alternative Paths of Biochemical Networks using Model Checking

Jinkyung Kim, Il Moon

Department of Chemical Engineering, Yonsei University, 134 Shinchon-dong Seodaemun-ku, Seoul 120-749, KOREA, E-mail : bayaba@yonsei.ac.kr

Abstract

This study focuses on automatic search and verification of feasible alternatives in all possible paths of biochemical networks using model checking method. To understand biochemical networks, the representation of hierarchical structure knowledge is a major challenge for bioinformatics. The automatic search and synthesis for alternative paths within complex and large networks in biological process can provide a huge amount of solutions, which is difficult to handle manually. Model checking is an automatic method for verifying if a circuit or a condition, expressed as a concurrent transition system, satisfies a set of properties expressed in a temporal logic such as CTL. This paper represents that model checking is feasible in biochemical network verification and it shows some advantages over simulation for querying and searching of special behavioral properties in biochemical processes.

Keywords

Automatic synthesis, Model checking, CTL, Biochemical networks, Pathway

1. Introduction

Biological behaviors in a biochemical system are controlled by large and complex networks of genes, proteins, small molecules, and their interconnections. The search for alternative paths and their analysis are a crucial point in the understanding of biochemical networks like metabolic networks,

signal transduction networks, or gene regulatory networks. As a biochemical system becomes more complex, however, it is difficult to analyze and validate through experimental data. In order to search and understand the behavior of the system, it is necessary to analyze and verify the system using computer aided method.

The quantitative information about the dynamics of biological systems is not sufficient, even in the current best technology, thus it is difficult for traditional numerical models and analysis techniques to be applied. In addition, numerical predictions on the biological system are difficult to verify, because available data are mostly qualitative in nature. Models of biochemical networks may become quite large, as they include many genes, proteins, and complex interconnections, manual verification of these properties and behaviors is error-prone or even practically infeasible. The use of formal tools for modeling biochemical processes and for reasoning about the behaviors seems to be a required research to contribute a lot in computer science and bioinformatics. Automatic verification technique using model checking is to ensure that the comparison of model predictions with experimental data is efficient and reliable.

Recently, researchers have explored the possibility of applying formal methods to biological and biochemical system by using both differential equation based framework and logic based frameworks.

In this paper, we propose an approach towards model verification and alternative path search to focus on the issue of providing automated model checking method for querying and verifying models in biochemical processes. More specially, we propose,

- the use of temporal logic CTL as a query language for alternative path search and model verification of biochemical processes,
- the use of automata theory for modeling of biochemical networks,
- the use of symbolic model checking techniques for automatically tracing and validating CTL queries in quantitative models.

2. Model checking method

The basic idea of what is known as model checking is to use algorithms, executed by computer tools, to verify the correctness of systems. Model checking is an automatic technique that, given a finite-state model of a system and a logical property, systematically checks whether this property holds for that model. Model checking is performed as an exhaustive state space search that is guaranteed to terminate since the model is finite. The user inputs as description of a model of the system (the possible behavior) and a description of the requirements specification (the desirable or undesirable behavior) and leaves the verification up to the machine.

If an error is recognized the model checker provides a counter-example consists of a scenario in which the model behaves in an undesired way. Thus the counter-example provides evidence that the model is faulty and needs to be revised. In biochemical system, the counter-example scenario can be an alternative path with desirable and undesirable behaviors, conditions, and constraints. This allows the biochemist to search alternative paths or to verify the error, transition behaviors and conditions for the experiment.

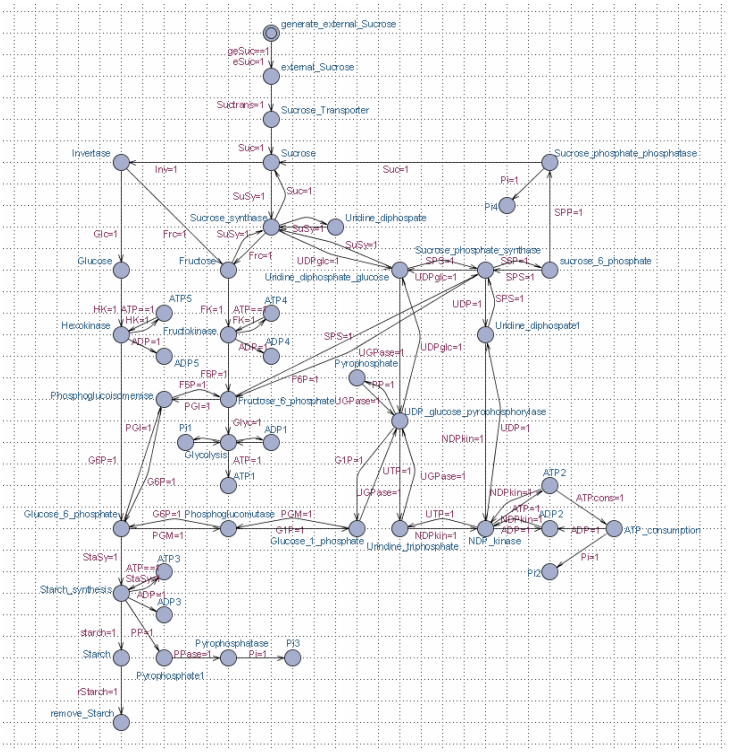

Figure 1. The model description of the sucrose to starch breakdown in the potato tuber.

3. Case Study

The accumulation of starch in the potato tuber is a crucial point in biotechnology. The major flux in the potato tuber carbon metabolism is the conversion of sucrose through hexose phosphates. Nearly all genes, believed to be directly involved in the sucrose breakdown transformation, have been cloned by transgenic approaches. A deeper understanding of the network behaviors, under laying the whole metabolism, might be obviously of help.

The model description for the networks of the system is showed in Figure 1. We

instantiate the biological queries for reachability and pathways with different kinds of constrains about a Boolean model of the bionetworks.

Specification 1: There is no path from eSuc to starch excluding ATP and including UTP.

A[] !((eSuc==1 && starch==1) && (ATP==0 && UTP==1))

Specification 2: There is no path from eSuc to starch including Glc and excluding HK.

A[] !((eSuc==1 && starch==1) && (Glc==1 && HK==0))

Specification 3: There is no path from eSuc to starch excluding ATP, Susy, and PP.

A[] !((eSuc==1 && starch==1) && SuSy==0 && (ATP==0 && PP==0))

Specification 4: There is no path from UDPglc to Suc excluding ADP,ATP, UDP, and UTP.

A[] !((UDPglc==1 && Suc==1) && (ADP==0 && ATP==0) && (UDP==0 && UTP==0))

Specification 5: There is no path from G6P to Suc excluding ADP, ATP, UDP, and UTP.

A[] !((G6P==1 && Suc==1) && (ADP==0 && ATP==0) && (UDP==0 && UTP==0))

These specifications are used for path search queries with constraints. The search for all possible paths from eSuc to starch without any constraints yields a lot of paths. This large number of paths is made because of the complex network structure containing many reversible reactions. Among these all possible paths, we are able to search alternative paths with constraints. The path trace for specification 1 yields one possible path through repeat querying and tracing. The answer for specification 2 is satisfied because it is obvious Glc can only be converted into G6P through HK. The specification 3 is to get all alternative paths, when sucrose synthase and pyrophosphate are not available, also excluding ATP. ADP must not be excluded explicitly, because ADP and ATP occur together on each path. As results from the specification 3, the path search through the counter-example produces two paths (including a tracing path in Figure 2) with a sensible biological interpretation. The other pathway is following:

generate_external_Sucrose→external_Sucrose→Sucrose_Transpoter→Sucrose →Invertase→Fructose→Fructokinase→Fructose_6_phosphate→Phosphoglucoisomerase→Starch_synthesis→Starch

In case of specification 4 and specification 5 about re-synthesis sucrose, after initial state is changed form *generated_external Sucrose* to *Uridine_diphosphate_glucose* and *Glucose_6_phosphate* respectively, queries are verified and the path traces are produced. Three paths for specification 4 (including a tracing path in Figure 2) are produced, the others are flowing:
1.Uridine_diphosphate_glucose→ Sucrose_synthase→Sucrose,
2.Uridine_diphosphate_glucose→UDP_glucose_pyrophosphorylase→Glucose_1_phosphate→Phosphoglucomutase→Glucose_6_phosphate→Phosphoglucoisomerse→Fructose_6_phosphate→Sucrose_phosphate_synthase→sucrose_6_phosphate→ Sucorse_phosphate_phosphatase→Sucrose.

Three correct paths for specification 5 are synthesized through the tracing counter-examples, the both of the others except a path in Figure 2 are following:
1.Glucose_6_phosphate→Phosphoglucomutase→Glucose_1_phosphate→UDP glucose_pyrophosphorylase→Uridine_diphosphate_glucose→Sucrose_synthase →Sucrose,
2.Glucose_6_phosphate→Phosphoglucomutase→Glucose_1_phosphate→UDPglucose_pyrophosphorylase→Uridine_diphosphate_glucose→Sucrose_phosphate_synthase→sucrose_6_phosphate→Sucorse_phosphate_phosphatase→Sucrose.
There is no path through invertase, which is correct, because the invertase driven reaction is irreversible. Three paths go through *Phosphoglucomutase* and *Phosphoglucoisomerase*, which reflects the biological behavior correctly.

4. Conclusions

This research introduces a new modeling and searching technique to design and to find alternative paths with constraints in biochemical networks. Discrete models for all possible path networks are developed using automata theory and specifications to find desirable and undesirable paths are formulated using CTL. We have shown how symbolic model checking techniques could be applied to the querying and verification of Boolean abstractions of networks of the generic property patterns and the protein interactions. First, the temporal logic CTL is used enough to formalize various different kinds of biological queries of interest about the networks: the reasonable reachability, the analysis of steady states and cyclic behaviors, the existence of checkpoints, and the synthesis under pathway constraints. Model checker tool produces the alternative paths of the sensible biological interpretation. The main advantage of this approach is that automatic verification and paths search with interest constraint through all possible paths and behaviors in complex biochemical networks, and model description is developed by user-friendly graphical modeling technique.

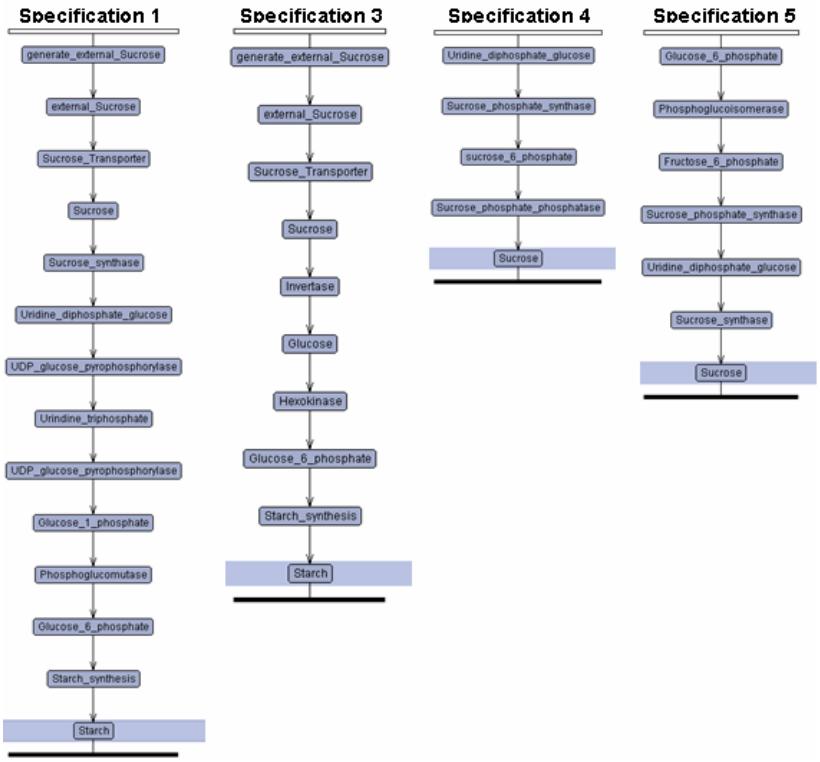

Figure 2. The alternative paths for specifications 1, 3, 4 and 5.

References

1. R. Alur and D. Dill, Theoretical Computer Science B 126 (1994) 183
2. G. Behrmann, A. David and K. G. Larsen, A Tutorial on Uppaal. Available at http://www.uppaal.com
3. N. Chabrier and F. Fages, Lecture Notes in Computer Science 2602 (2003) 149
4. M. Heiner, I. Koch and J. Will, Biosystems 75 (2004) 15
5. J. Kim and I. Moon, Computers and Chemical Engineering 24 (2000) 385
6. J. Kim, Y. Lee and I. Moon, ESCAPE 16 meeting Jul. 9-13 (2006)
7. I. Koch, M. Schö ler and M. Heiner, In Silico Biology, 5 (2005) 129
8. I. Moon, G. J. Power, J. R. Burch and E. M. Clarke, AIChE Journal 38 (1992) 67
9. S. Eker, M. Knapp, K. Laderoute, P. Lincoln and C. Talcott, Electronic Notes in Theoretical Computer Science 71 (2002)
10. M. Heiner, I. Koch and J. Will, Biosystems 75 (2004) 15