Linear Temporal Logic for Biologists in BMA

Benjamin A. Hall[1], Nir Piterman[2], and Jasmin Fisher[1,3]

[1] University of Cambridge, Cambridge, UK
[2] University of Leicester, Leicester, UK
[3] Microsoft Research, Cambridge, UK

The BioModelAnalyzer (BMA http://biomodelanalyzer.research.microsoft.com/) is a web based tool for the development of discrete models of biological systems. Through a graphical user interface, it allows rapid development of complex models of gene and protein interaction networks and stability analysis without requiring users to be proficient computer programmers [1, 2]. Here I will present a new set of tools in the BMA that allow users to perform complex queries over models in linear temporal logic, allowing biologists to test specifications based on the dynamics observed in simulations. In keeping with the core objective of tool, queries are constructed graphically and results are presented to the users with examples of simulations. Alongside stability analysis, this new tool allows biologists to verify complex specifications to validate executable biological models.

Linear temporal logic queries are substantially more complex than stability testing due to the fundamental requirement for users to construct the query and select a path length. Biologists specifically face further problems; biological models typically have many variables that may be included in a query, they may not be familiar with complex operator precedence issues, and they must balance parentheses. Whilst this is handled by NuSMV in other tools [3, 4], a design principle of BMA is that computing proficiency is not required so necessarily this must be achieved in the GUI.

We address these issues in the tool through a two-stage workflow (Figure 1). Users define *LTL states*; large conjunctions of variable assignments that may be created by dragging and dropping from the model canvas, or through a drop-down menu inspired by file browsers. These states are transformed into an LTL query in a second temporal and logical layer. Operators in this layer carry a number of "sockets" into which operands (i.e. LTL states) or other operators

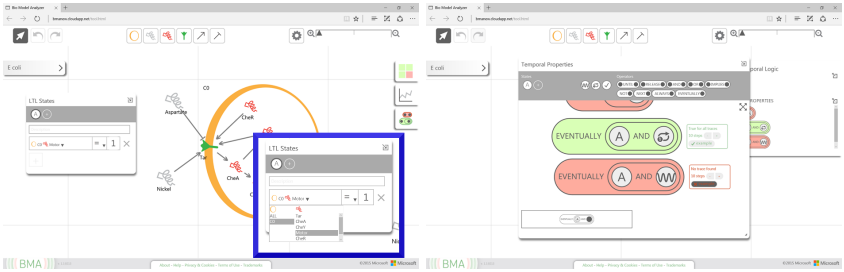

Figure 1: The LTL state editor and the LTL query editor.

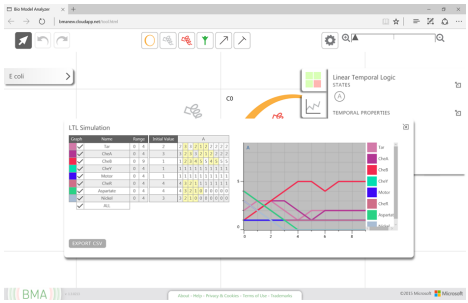

Figure 2: An example trace from an LTL query.

may be dragged and dropped. As such complex queries can be developed through repeatedly nesting operators.

To aid users several default states are included. In addition to "True", users may also search for fixpoints or cycles in the system. These are valuable for studying biological models as they allow users to study developmental end-points specifically. The description of these states through other operators would be difficult. For example, a self loop state is characterised by the formula $\bigwedge_{v \in V} v = Xv$. To the best of our knowledge, most LTL tools do not support such a direct comparison between the value of a variable and its value in the next state.

User testing indicated that the LTL operator "until" confused unfamiliar users, as the first operand need not hold. To address this we supplemented the list of operators with the non-standard operator "upto", which carries a similar meaning in English but ensures that both operands hold (A upto B corresponds to A and next A until B).

On clicking the test button both the query and its negation are checked. This produces three types of result- always true, never true, and true for some. The user can then choose to see examples of simulations that satisfy, or fail to satisfy the query (Figure 2).

References

1. Benque, D., Bourton, S., Cockerton, C., Cook, B., Fisher, J., Ishtiaq, S., Piterman, N., Taylor, A., Vardi, M.: BMA: Visual tool for modeling and analyzing biological networks. In: 24th International Conference on Computer Aided Verification. Volume 7358 of Lecture Notes in Computer Science., Springer (2012) 686–692
2. Chuang, R., Hall, B., Benque, D., Cook, B., Ishtiaq, S., Piterman, N., Taylor, A., Vardi, M., Koschmieder, S., Gottgens, B., Fisher, J.: Drug target optimization in chronic myeloid leukemia using innovative computational platform. Scientific Reports, 5:8190 (2015)
3. Naldi, A., Thieffry, D., Chaouiya, C.: Decision diagrams for the representation and analysis of logical models of genetic networks. In: Computational Methods in Systems Biology. Volume 4695 of Lecture Notes in Computer Science., Springer (2007) 233–247
4. Bean, D., Heimbach, J., Ficorella, L., Micklem, G., Oliver, S., Favrin, G.: esyN: Network building, sharing, and publishing. PLoS ONE **9** (2014) e106035