

# Repairing Boolean regulatory networks using Answer Set Programming

Alexandre Lemos, Pedro T. Monteiro and Inês Lynce

INESC-ID/Instituto Superior Técnico, Universidade de Lisboa, Portugal

Models of biological regulatory networks are increasingly used to formally describe and understand complex biological processes. Such models are often repaired whenever new observations become available, because the model cannot generate behaviours consistent with the new observations. However, the model repair procedure is often a manual process and therefore prone to errors.

In this work, we describe biological regulatory networks using the Boolean logical formalism [1], where nodes denote biological components and edges denote regulatory interactions between components. Each component is associated with a Boolean variable representing its activity level. The evolution of each variable is defined by a Boolean logical function depending on the values of its regulators.

Here, we propose to repair the model by changing functions, responsible for the generation of inconsistent behaviours, with four types of atomic repairs which can be further combined. The goal is to find the cardinality minimal set of repairs which make the model satisfy all available observations [2].

The proposed method is implemented using Answer Set Programming (ASP)<sup>1</sup> [3] and is tested using data from *Escherichia coli* and *Candida albicans* model organisms.

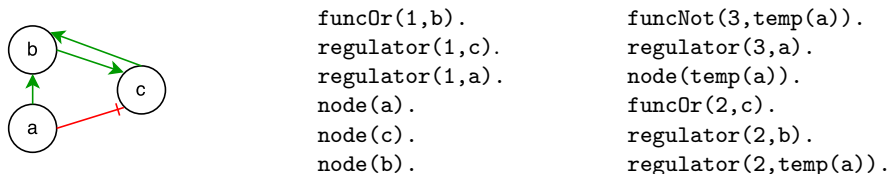


Fig. 1: The representation of small network (left) and the partial encoding in ASP of the network (right).

Given a model, such as the one described in Figure 1, new experimental or literature data may represent new observations that cannot be generated by the model. For example, considering the experimental profile ( $a=\text{true}$ ,  $b=\text{false}$ ,  $c=\text{true}$ ), one can see that the model in Figure 1 is inconsistent, since the functions that explains the value of the node *b* and *c*, differs from the experimental profile. Potential model revisions should aim at finding the set of minimal repairs that make the model consistent with all the available data.

<sup>1</sup> The complete encoding is available at <http://web.ist.utl.pt/~alexandre.lemos/rbnasp/>

Here, we propose the use of four basic types of repairs to the logical functions, which can then be further combined to produce more detailed revisions:

- e* - removes a regulator (never removes the last regulator, *i.e.*, component cannot become an input);
- i* - negates a regulator;
- n* - changes an AND/OR function into NAND/NOR function, respectively;
- g* - changes a AND to OR and a NOT to the **identity** function.

Considering the model described in Figure 1 and the example experimental profile (**a=true**, **b=false**, **c=true**), the model can be repaired by applying two repairs of type *n*, *i.e.* first negating the function `funcOr(1,b)` and then the function `funcOr(2,c)`. Here, one can see that these repairs are cardinality minimal, corresponding to the minimal number of repairs required to correct the model. However, they are not unique, since it is possible to perform other types of repairs to correct this model, such as removing the NOT and negating regulators *b* and *c*.

Interestingly, the combination of repair *i*, which allows the negation of a function arguments (only allows the negation of arguments that have not been previously negated), with repair *g*, yields an output more general than when applying only the repair *n*. The functions NAND and NOR are a subgroup of the functions produced when combining those repairs.

Generically, the number of possible Boolean functions that can be used to repair a function will increase with the number of regulators influencing a given component, following the expression  $2^{2^n}$ , where *n* is the number of arguments of the function. For example, the number of possible functions with one argument is four, with two arguments is sixteen, etc.

Nevertheless, when considering a function with two arguments, by combining repairs *e*, *i* and *g*, one can only achieve a total of twelve functions (all basic the Boolean functions, plus one of the derived Boolean functions, the implication). The functions XOR (exclusive or), XNOR (equivalence), **true** and **false** are not achievable by any combination of these repairs.

The proposed method for model repair was tested using two regulatory networks of known model organisms, *Escherichia coli* and *Candida albicans*, with 1915 and 6410 nodes respectively. We shown that it is possible to find a set of repairs that turns both regulatory networks consistent with the experimental data available.

- [1] Thomas, R. Boolean formalization of genetic control circuits. *J. Theor. Biol.* (1973) 42, 563-585
- [2] Gebser, M. *et al.* Repair and prediction (under inconsistency) in large biological networks with Answer Set Programming. *Principles of Knowledge Representation and Reasoning* (2010)
- [3] Gebser, M. *et al.* Answer set solving in practice. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, Morgan and Claypool Publishers (2012)

**Acknowledgements:** This work was partially supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013. Alexandre Lemos was supported by FCT project grant EXCL/EEI-ESS/0257/2012. Pedro T. Monteiro was supported by FCT grant IF/01333/2013.