

Functions on Solutions - Language Extensions for ML-Rules

Sebastian Nähring, Tobias Helms, Stefan Rybacki, and Adelinde M. Uhrmacher

University of Rostock, Institute of Computer Science
Modeling and simulation group
{sebastian.naehring,tobias.helms,stefan.rybacki,
adelinde.uhrmacher}@uni-rostock.de

<http://wwwmosi.informatik.uni-rostock.de/Plone/en>

Keywords: rule-based modeling, multi-level modeling, functions on solutions

1 Motivation and Background

ML-Rules is a rule-based language for multi-level modeling of cell biological systems [4]. Each species can be characterized by attributes and the set of species, i.e., the solution, that it contains. Thus, species can build hierarchies in ML-Rules. Rule-schemata can be applied on different levels and access and manipulate the hierarchical structure. New species can be added to solutions and species will be removed once a corresponding reaction occurred. But no general approach is available to access and manipulate solutions. Hence some biological phenomena are currently hard or impossible to express with ML-Rules, e.g. dividing a solution equally into multiple new solutions to model cell division.

2 Results

The goal of this work has been to extend ML-Rules in such a way that its expressiveness is increased while still maintaining a high accessibility. Therefore, we added the lambda-calculus to the ML-Rules language, as has been done in React(C) [3]. To simplify the development of those lambda-functions, an own syntax was developed resembling the existing ML-Rules syntax. New functions can be described through multiple patterns that are applied to the solutions similar to rule-schemata. Thereby, solutions (set of species) can be accessed and manipulated recursively. Whenever one of the patterns can be matched to the functions input values, the corresponding result will be returned. Functions can also be used in other functions, even as higher-order functions. Despite the similarity to the ML-Rules syntax, we expect that most often power users will define and provide these functions via libraries. Those functions then enable modelers to express complex dynamics in a compact manner, e.g. to split a solution into two equal new solutions to model cell division (Fig. 1), to describe the fission of mitochondria into heterogeneous mitochondria, or to change attributes of all

species that reside in one area of the cell. The extension appears rather powerful and able to capture a wide range of functionalities. E.g., it also allows to count species that have certain characteristics, for which in BNGL or SBML specific language constructs were added [1, 2]. However, only future applications can show the benefit of the approach, user studies are required to test its accessibility, and its effect on run-time has still to be analyzed in performance studies.

```
// new function definition
split [] _ = ([], [])
split [s + sol?] v = ([left s + fst.rec], [right s + snd.rec])
                    where left = round(#s * v)
                          right = #s - left

rec = split.sol? v

// rule schema with function application
Cell[sol?] -> Cell[sol1?] + Cell[sol2?] @ k
  where (sol1?, sol2?) = split.sol? 0.5
```

Fig. 1. The splitting function is defined with multiple patterns that decompose a given solution. Through recursive application any given solution can be split into two new solutions given a splitting value. Such a function may be defined by a power user, whereas the rule schema at the end demonstrates how it can be used by a modeler.

Acknowledgments

This work was funded by the German Research Foundation (DFG) as part of the Research Training Group dIEM oSiRiS (<http://www.diemosiris.de>).

References

1. James R. Faeder, Michael L. Blinov, and William S. Hlavacek. Rule-based modeling of biochemical systems with bionetgen. In *Systems Biology*, volume 500 of *Methods in Molecular Biology*, pages 113–167. Humana Press, 2009.
2. Michael Hucka, Lucian Smith, Darren Wilkinson, Frank Bergmann, Stefan Hoops, Sarah Keating, Sven Sahle, and James Schaff. The systems biology markup language (SBML): language specification for level 3 version 1 core. *Nature Precedings*, October 2010.
3. Mathias John, Cédric Lhoussaine, Joachim Niehren, and Cristian Versari. Biochemical reaction rules with constraints. In *Programming Languages and Systems*, volume 6602, pages 338–357. 2011.
4. Carsten Maus, Stefan Rybacki, and Adelinde M. Uhrmacher. Rule-based multi-level modeling of cell biological systems. *BMC Systems Biology*, 5(1):166, 2011.