A theorem prover for a logical calculus of molecular biology

Filippo Sestini Silvia Crafa
Università di Padova, Italy

Drawing on the idea that biochemical pathways can be viewed as deductive processes, Boniolo et al. ([BDDF10, BDPP13, BDPP15]) proposed a suitable fragment of linear logic as a toolbox for molecular biology. More precisely, in their logical calculus, called Zsyntax, a proof of $\Gamma \vdash \Delta$ can be seen as describing the biochemical path of reactions going from the molecular aggregate represented by Γ to the one represented by Δ. The underlying biological interpretation determines two main features that distinguish Zsyntax from the standard linear logic: *controlled monotonicity* and *non commutativity* of derivations.

Traditional linear consequence is monotonic with respect to multiplicative conjunction, in the sense that $A \vdash B$ always implies $A \otimes C \vdash B \otimes C$ for any instantiation of the propositional variables. This is at odds with the fact that many state transitions of interest in molecular biology are context-sensitive: a reaction may be inhibited by the presence of a suitable molecular aggregate in the context in which it occurs. Therefore, the deductive system of Zsyntax ([BDPP13, BDPP15]) considers context-sensitive state transitions: a transition may take place in every molecular context that satisfies its *control condition*, which describes all the contexts that inhibit such transition. In this sense, it is reminiscent of other formalisms like default logic ([Rei87]). A direct consequence of context-sensitivity is *non commutativity* of derivations, that is Zsyntax proofs crucially depend on the order in which conditional formulas are eliminated. Moreover, control conditions —expressed in Zsyntax with so-called *control sets*—are intended to be empirically determined, i.e., they result from empirical knowledge obtained in the laboratory. Since their content may change over time as new experimental data is collected, the resulting logical system is “open”: theorems may lose their status depending on modifications of the empirical knowledge.

In this work we present the development of an automated theorem prover for a fragment of Zsyntax with Z-conjunction and Z-conditional operators (the Zsyntax counterparts of multiplicative conjunction and linear implication), which covers both controlled monotonicity, non commutativity and openness of the system, and is expressive enough to encode all the biochemical pathways considered in [BDDF10, BDPP15, Bon16], such as the glycolytic pathway from D-Glucose to Fructose-1,6-bisphosphate or the melanoma network. Both the logical foundation of the theorem prover and the efficient implementation of the automated deduction procedure draw on existing literature on automated deduction and intuitionistic linear logic but they require new solutions to deal with nontrivial aspects such as control conditions and the temporal dimension of transition processes.

A proof search-oriented logical foundation. As a first step, the original natural deduction-style formulation of [BDPP15] is reformulated as a sequent calculus for intuitionistic linear logic enriched with annotations to represent biochemical features and perform monotonicity control. More precisely, inference rules rely on annotated sequents of the form $\Gamma; \Delta \Longrightarrow_l C$, where Γ and Δ are, respectively, an unrestricted context and a linear context ([Cha06, Gir93]), C is a single formula and l is a *reaction list*. The unrestricted context is one where the usual structural rules of contraction and weakening are allowed, and provides in our setting a natural and elegant way to represent the set of biological axioms that are assumed to hold, and that can be freely used during proof search. Reaction lists are used to deal with the distinctive concept of control sets: they essentially keep track of the intermediate transitions that were used to transform the initial context Δ into the final C, and the intermediate forms assumed by Δ in the process. In biological terms, a valid sequent $\Gamma; \Delta \Longrightarrow_l C$ states that, under the empirical knowledge encoded by Γ, there exists a biochemical reaction starting from the aggregate Δ and ending in the aggregate C, with a sequence of intermediate reactions described by l. Moreover, this reaction is allowed to happen in any surrounding biochemical context that is compatible with l.

An analysis of Zsyntax shows that the Z-conditional operator contains an implicit existential quantification which is not fine-tuned to a calculus with control sets. Indeed, introducing an element of type $A \rightarrow B$ requires to show that *there exists* an aggregate that, paired with A, transitions to B, while to prove something by elimination on an element of type $A \rightarrow B$ is to prove it without knowledge on which aggregate was used to establish $A \rightarrow B$, thus the contexts of valid deductions are required to inhibit *no (currently known) instance* of $A \rightarrow B$. We therefore rely on an alternative, *annotated* conditional operator, allowing a deduction to refer to a specific instance of a conditional formula, i.e., a specific biochemical reaction. Notice that the annotated conditional introduces a further source of non commutativity in the logical proofs, since the specific order in which conditionals are introduced is reflected into the annotations. We claim that a conditional operator annotated with biologically relevant information is well suited to a formal language for molecular biology, and we observe that the original Z-conditional can be obtained from the newer one by an explicit existential quantification over the annotations; more details about a preliminary investigation on the precise relationships between the original and the annotated conditionals can be found in the extended abstract.

The automated deduction procedure. To implement an efficient automated deduction procedure, we resort to Chauduri's idea ([Cha06]) of combining two well-known reasoning techniques for linear logic: focusing and the inverse method. The main issue here is the non commutativity of Zsyntax derivations: since the focusing mechanism imposes a precise order in the application of inference rules, many derivations that would be equivalent in a focused calculus for linear logic are very different under the biochemical, non-monotonic interpretation of Zsyntax. Therefore we consider a weaker form of focusing that only acts on multiplicative conjunction and preserves completeness. The resulting backward calculus of derived rules is then converted to the forward direction, that does not suffer from multiplicative non-determinism but lacks the goal-orientation typical of backward search. Therefore, still following Chaudhuri's work, we use the inverse method to recover goal-orientation in the forward direction, and combine it with focusing to get a *focused inverse method* for Zsyntax.

A final important issue is the decidability of the logical calculus, which is nontrivial since the presence of the unrestricted context—the possibility of freely introducing formulas representing biochemical axioms into the linear context— suggests a correspondence with the intuitionistic fragment of multiplicative linear logic with "of course" exponential (MELL) [Lin94]. A precise characterization of the complexity of the logic of Zsyntax is out of scope, even if it obviously affects the design of an automated theorem prover. Our implementation uses a simple solution by limiting the number of sequents that can be generated during the search for a single goal sequent, and signalling a failure when such a limit is reached. This is a common approach, for instance the `llprover`[llp] theorem prover for a fragment of linear logic with "of course" exponential imposes a bound on the number of contractions per derivation path. The search procedure we employ is a standard Otter loop [Kal01]. As in [Cha06] we represent inference rules as sort of curried functions, so to treat partially applied rules in a uniform way.

Front-end. The theorem prover is implemented under the design principle that the intended users are molecular biologists. Therefore its front-end shows almost nothing of the underlying logical complexity such as reaction lists or any other monotonicity control technicalities. The user simply enters the set of empirically valid axioms and queries for the validity of a goal sequent, no biochemical annotation is required and any sequent that is found to logically subsume the goal is returned, regardless of its reaction list. The only conditionals that we allow to be specified in the goal sequent are those contained in the unrestricted context of empirical axioms; we instead prevent their addition to the linear context since their treatment requires the above mentioned existential quantification over the annotated information.

To conclude, we observe that our theorem prover, although relatively simple, is still expressive enough to encode many biologically interesting cases. All the examples using Zsyntax to encode biochemical pathways as deductions given in [BDDF10, BDPP15, Bon16] have been easily formalized and checked in the tool. Finally, the theorem prover is currently used as en experimental tool for biological reasoning by a group of molecular biologists of Università di Ferrara, Italy.

References

[BDDF10] Giovanni Boniolo, Marcello D'Agostino, and Pier Paolo Di Fiore. Zsyntax: a formal language for molecular biology with projected applications in text mining and biological prediction. *PloS one*, 5(3):e9511, 2010.

[BDPP13] Giovanni Boniolo, Marcello D'Agostino, Mario Piazza, and Gabriele Pulcini. A logic of non-monotonic interactions. *J. Applied Logic*, 11(1):52–62, 2013.

[BDPP15] Giovanni Boniolo, Marcello DAgostino, Mario Piazza, and Gabriele Pulcini. Adding logic to the toolbox of molecular biology. *European Journal for Philosophy of Science*, 5(3):399–417, 10 2015.

[Bon16] Giovanni Boniolo. Decomposing biological complexity into a conjunction of theorems. the case of the melanoma network. *HUMANA.MENTE*, 2016.

[Cha06] Kaustuv Chaudhuri. *The Focused Inverse Method for Linear Logic*. PhD thesis, Carnegie Mellon University, 2006.

[Gir93] Jean-Yves Girard. On the unity of logic. *Annals of Pure and Applied Logic*, 59(3):201 – 217, 1993.

[Kal01] John Arnold Kalman. *Automated Reasoning with Otter*. Rinton Press, Incorporated, Paramus, NJ, 2001.

[Lin94] Patrick Lincoln. Deciding provability of linear logic formulas. In *Advances in Linear Logic*, pages 109–122. Cambridge University Press, 1994.

[llp] Web page of `llprover`, a linear logic prover. http://bach.istc.kobe-u.ac.jp/llprover/.

[Rei87] R. Reiter. Readings in nonmonotonic reasoning. chapter A Logic for Default Reasoning, pages 68–93. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.