Equivalence of chemical reaction networks in a CRN-to-DNA compiler framework.

Stefan Badelt[1], and Erik Winfree[1]

California Institute of Technology, Pasadena, USA
{badelt, winfree}@caltech.edu

The formal language of mass-action chemical reaction networks (CRNs) can be used as an (intermediate) programming language for nucleic acid interaction networks. Recently, we presented the compiler `Nuskell`[1] [1], which automates the translation of formal CRNs into domain-level strand displacement (DSD) networks. `Nuskell` incorporates the fundamental enabling concept for programming languages and compilers: a rigorous abstraction hierarchy with well-defined semantics at each level, and rigorous proofs establishing the correctness of compilation from a higher level to a lower level (see Figure 1).

DSD systems [12] are abstract representations of nucleic acid interaction networks. Intuitively, nucleic acids are represented as sequences of **domains**, as opposed to sequences of nucleotides. If domains are well-designed, then they hybridize as one entity to their respective **complementary domains**, and are unpaired otherwise. A **strand** is a sequence of domains which is connected with a covalent backbone, while the corresponding **structure** describes which domains are hybridized and which are unpaired. A **complex** is an entity composed of one or more strands with a particular structure. **Signal** complexes are at low concentrations and they represent the dynamical information, e.g. input and output. **Fuel** complexes are the energy source at high (ideally constant) concentrations and they mediate the information transfer by consuming and/or releasing signal complexes.

`Nuskell` provides several systematic methods for translating CRNs into DSD systems which have been developed theoretically, and in some cases demonstrated experimentally [3–5, 8, 9, 12, 13]. Similar to compiler flags, these translation schemes can be selected to optimize output for a particular experimental setup. Our contribution reviews different levels of `Nuskell's` verification in order to prove or disprove the equivalence of a formal CRN and the interaction network of the translated system. The case-by-case verification allows translation schemes which can be efficient for certain classes of CRNs, but then introduce unwanted side reactions for a different class of CRNs.

Note, that the presented verification methods are not intertwined with the translation process. Instead, correctness can depend on the predicted biophysics for a specific experimental setup. For example, DSD reaction network enumeration using `Peppercorn`[2] [6], assumes a time-scale separation that is valid only

[1] https://github.com/DNA-and-Natural-Algorithms-Group/nuskell
[2] https://github.com/DNA-and-Natural-Algorithms-Group/peppercornenumerator

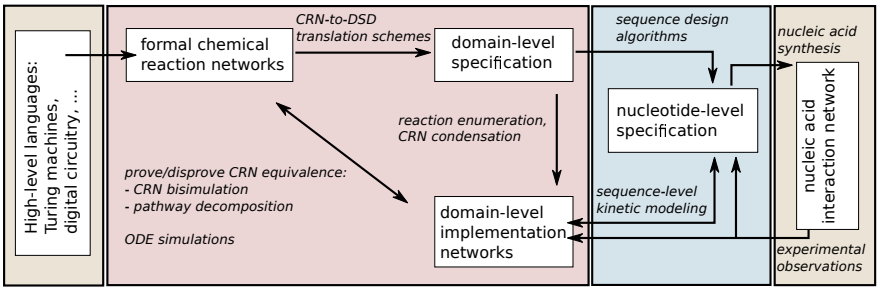

Fig. 1: The `Nuskell` compiler hierarchy. **The current version (red)** translates formal CRNs into a set of domain-level nucleic acid complexes. The algorithm for translation can be chosen from multiple different CRN-to-DSD translation schemes. Domain-level complexes are input for a DSD reaction network enumerator which returns a condensed version of the enumerated CRN. Two CRN equivalence notions can be used to formally verify the equivalence between the domain-level reaction network and the formal CRN. **The next version (blue)** will translate *correct* domain-level specifications into nucleotide-level specifications and use nucleotide-level kinetic models to verify the correct implementation of domain-level reaction networks. **Eventually (yellow)**, the `Nuskell` compiler can incorporate experimental feedback to train domain-level and nucleotide-level biophysics and it can be embedded into workflows that derive CRNs from high-level languages. Figure adapted from [1].

for low concentrations, but one can change the enumeration semantics to include reactions that require higher complex concentrations (e.g. cooperative hybridization [4, 14]). After the enumerated network is *condensed* into a smaller, dynamically equivalent reaction network [6], it can be verified using stochastic-level theories of pathway decomposition equivalence [11] and/or CRN bisimulation [7]. We will discuss different semantics for DSD reaction enumeration and present a refined domain-level (i.e. sequence-independent) reaction rate model, to demonstrate that simulations on enumerated reaction networks are reasonable approximations of experimentally tested DSD systems (unpublished).

The strength of the present `Nuskell` version comes from three features: First, incorporation of new DSD design principles as translation schemes makes the design and optimization of complex networks easily accessible to a broad scientific community. Second, rigorous proofs of correctness guarantee a successful domain-level compilation, and are applied on a flexible case-by-case basis. Third, multiple translation schemes can be compared for a given CRN, exploiting the diversity of DSD circuits implementing the same CRN and allowing for optimization of circuits at the domain level, before proceeding to the computationally more expensive DNA nucleotide-level design and verification.

In the future, `Nuskell` will compile from the domain-level to the more detailed nucleotide-level specification. This requires verification that the dynamics

approximated on the domain-level remain preserved at the nucleotide-level, but also allows for fine-tuning of reaction rates to adjust the time-scale of computation. If time permits, we can present a first step in this direction using the software packages `Multistrand`[3] [10] and `KinDA`[4] [2] to estimate nucleotide-level strand-displacement reaction rates and quantify the formation of unintended side products for a given nucleotide sequence design.

References

1. Badelt, S., Shin, S.W., Johnson, R.F., Dong, Q., Thachuk, C., Winfree, E.: A general-purpose CRN-to-DSD compiler with formal verification, optimization, and simulation capabilities. In: International Conference on DNA-Based Computers. pp. 232–248. Springer (2017)
2. Berleant, J., Berlind, C., Schaeffer, J., Winfree, E.: Automated sequence-level analysis of kinetics and thermodynamics for domain-level dna strand-displacement systems. submitted
3. Cardelli, L.: Strand algebras for DNA computing. Natural Computing 10(1), 407–428 (2011)
4. Cardelli, L.: Two-domain DNA strand displacement. Mathematical Structures in Computer Science 23(02), 247–271 (2013)
5. Chen, Y.J., Dalchau, N., Srinivas, N., Phillips, A., Cardelli, L., Soloveichik, D., Seelig, G.: Programmable chemical controllers made from DNA. Nat. nanotechnol. 8(10), 755–762 (2013)
6. Grun, C., Sarma, K., Wolfe, B., Shin, S.W., Winfree, E.: A domain-level DNA strand displacement reaction enumerator allowing arbitrary non-pseudoknotted secondary structures. arXiv:1505.03738 (2014)
7. Johnson, R.F., Dong, Q., Winfree, E.: Verifying chemical reaction network implementations: a bisimulation approach. Theoretical Computer Science (2018)
8. Lakin, M.R., Youssef, S., Cardelli, L., Phillips, A.: Abstractions for DNA circuit design. Journal of The Royal Society Interface 9(68), 470–486 (2012)
9. Qian, L., Soloveichik, D., Winfree, E.: Efficient Turing-universal computation with DNA polymers. In: DNA Computing and Molecular Programming, pp. 123–140. Springer (2010)
10. Schaeffer, J.M., Thachuk, C., Winfree, E.: Stochastic simulation of the kinetics of multiple interacting nucleic acid strands. In: Phillips, A., Yin, P. (eds.) DNA Computing and Molecular Programming. vol. 9211, pp. 133–153. Springer International Publishing (2015)
11. Shin, S.W., Thachuk, C., Winfree, E.: Verifying chemical reaction network implementations: A pathway decomposition approach. Theoretical Computer Science (2017)
12. Soloveichik, D., Seelig, G., Winfree, E.: DNA as a universal substrate for chemical kinetics. Proceedings of the National Academy of Sciences 107(12), 5393–5398 (2010)
13. Srinivas, N., Parkin, J., Seelig, G., Winfree, E., Soloveichik, D.: Enzyme-free nucleic acid dynamical systems. Science 358(6369) (2017)
14. Zhang, D.Y.: Cooperative hybridization of oligonucleotides. Journal of the American Chemical Society 133(4), 1077–1086 (2010)

[3] https://github.com/DNA-and-Natural-Algorithms-Group/multistrand
[4] https://github.com/DNA-and-Natural-Algorithms-Group/KinDA