Verifying Chemical Reaction Network Implementations: A Pathway Decomposition Approach

Seung Woo Shin*, Chris Thachuk†, and Erik Winfree†

*University of California, Berkeley
†California Institute of Technology

Abstract

The emerging fields of genetic engineering, synthetic biology, DNA computing, DNA nanotechnology, and molecular programming herald the birth of a new information technology that acquires information by directly sensing molecules within a chemical environment, stores information in molecules such as DNA, RNA, and proteins, processes that information by means of chemical and biochemical transformations, and uses that information to direct the manipulation of matter at the nanometer scale. To scale up beyond current proof-of-principle demonstrations, new methods for managing the complexity of designed molecular systems will need to be developed. Here we focus on the challenge of verifying the correctness of molecular implementations of abstract chemical reaction networks, where operation in a well-mixed "soup" of molecules is stochastic, asynchronous, concurrent, and often involves multiple intermediate steps in the implementation, parallel pathways, and side reactions. This problem relates to the verification of Petri Nets, but existing approaches are not sufficient for certain situations that commonly arise in molecular implementations, such as what we call "delayed choice." We formulate a new theory of pathway decomposition that provides an elegant formal basis for comparing chemical reaction network implementations, and we present an algorithm that computes this basis. We further show how pathway decomposition can be combined with weak bisimulation to handle a wider class that includes all currently known enzyme-free DNA implementation techniques. We anticipate that our notion of logical equivalence between chemical reaction network implementations will be valuable for other molecular implementations such as biochemical enzyme systems, and perhaps even more broadly in concurrency theory.

1 Introduction

A central problem in molecular computing and bioengineering is that of implementing algorithmic behavior using chemical molecules. The ability to design chemical systems that can sense and react to the environment finds applications in many different fields, such as nanotechnology [4], medicine [6], and robotics [9]. Unfortunately, the complexity of such engineered chemical systems often makes it challenging to ensure that a designed system really behaves according to specification. Since an experiment is generally an expensive process that requires a large amount of resources, it would be useful to have a procedure by which one can theoretically verify the correctness of a design using computer algorithms. In this paper we propose a theory that can serve as a foundation for such automated verification procedures.

Specifically, we focus our attention on the problem of verifying chemical reaction network (CRN) implementations. Informally, a CRN is a set of chemical reactions that specify the behavior of a given chemical system in a well mixed solution. For example, the reaction equation $A + B \rightarrow C$ means that a reactant molecule of type A and another of type B can be *consumed* in order to *produce* a product molecule of type

C. A reaction is applicable if all of its reactants are present in the solution in sufficient quantities. In general, the evolution of the system from some initial set of molecules is a stochastic, asynchronous, and concurrent process. While it is clear that abstract CRNs provide the most widely used formal language for describing chemical systems, it is interesting to note that it can also serve as a programming language in molecular computing or bioengineering. This is because CRNs are often used to specify the target behavior for an engineered chemical system (see Figure 1). How can one realize these "target" CRNs experimentally? Unfortunately, synthesizing chemicals to efficiently interact — and only as prescribed — presents a significant, if not infeasible, engineering challenge. Fortunately, any target CRN can be *emulated* by a (generally more complex) "implementation" CRN. For example, in the field of DNA computing, implementing a given CRN using synthesized DNA strands is a well studied topic that has resulted in a number of translation schemes [18, 2, 16].

In order to evaluate CRN implementations prior to their experimental demonstration, a mathematical model describing the expected molecular interactions is necessary. For this purpose, software simulators that embody the relevant physics and chemistry can be used. Beyond performing simulations – which by themselves can't provide absolute statements about the correctness of an implementation – it is often possible to describe the model of the molecular implementation as a CRN. That is, software called "reaction enumerators" can, given a set of initial molecules, evaluate all possible configuration changes and interactions, possibly generating new molecular species, and repeating until the full set of species and reactions have been enumerated. In the case of DNA systems, there are multiple software packages available for this task [12, 8].

Given a "target" CRN which specifies a desired algorithmic behavior and an "implementation" CRN which purports to implement the target CRN, how can one check that the implementation CRN is indeed correct? As we shall see, this question involves subtle issues that make it difficult to even define a notion of correctness that can be universally agreed upon, despite the fact that in this paper we study a somewhat simpler version of the problem in which chemical kinetics, i.e. rates of chemical reactions, is dropped from consideration. However, we note that this restriction is not without its own advantages. For instance, a theory based on chemical kinetics will necessarily involve approximation and therefore may overlook certain logical flaws in the implementation. In contrast, a theory that ignores chemical kinetics will be exact and therefore emphasize the logical aspect of the correctness question.

The main challenge in this verification problem lies in the fact that the implementation CRN is usually much more complex than the target CRN. This is because each reaction in the target CRN, which is of course a single step in principle, gets implemented as a sequence of steps which may involve "intermediate" species that were not part of the original target CRN. For example, in DNA-based implementations, the implementation CRN can easily involve an order of magnitude more reactions and species than the target CRN. Given that the intermediate species participating in implementations of different target reactions can potentially interact with each other in spurious ways, it becomes very difficult to verify that such an implementation CRN is indeed "correct."

Figure 1 illustrates various different ways that a proposed implementation can be "incorrect." For instance, one can easily see that CRN2 is clearly not a good implementation of CRN1, because it implements the reaction $A + A + B \to C + D$ in place of $A + B \to C + D$. CRN3 is incorrect in a more subtle way. While a cursory look may not reveal any immediate problem with this implementation, one can check that CRN3 can get from the initial state $\{A, A, B\}$ to a final state $\{A, B, C\}$, whereas there is no way to achieve this using reactions from CRN1. CRN4 is incorrect in yet another way. Starting from the initial state $\{A, C\}$, one can see that the system will sometimes get "stuck" in the state $\{i, C\}$, unable to produce $\{C, C\}$, with i becoming an intermediate species that is not really "intermediate." Now, CRN5 seems to be free of any such issue, but with what confidence can we declare that it is a correct implementation of CRN1, having seen the subtle ways that an implementation can go wrong? A goal of this paper is to provide a mathematical definition of "correctness" of CRN implementations which can be used to test them in

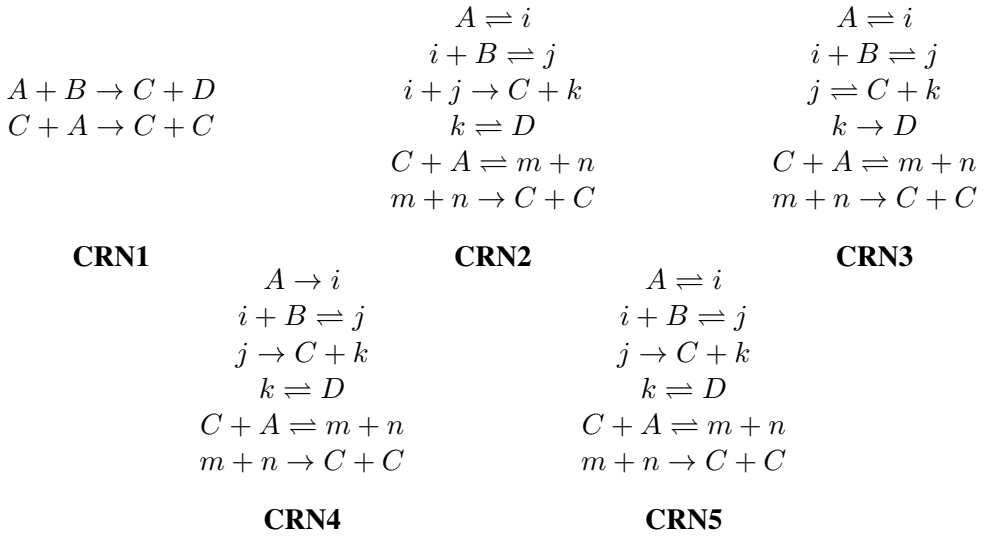

Figure 1: An example of CRN implementation. CRN1 represents the "target" CRN, i.e., the behavior we desire to implement, whereas CRN2-5 are potential "implementations" of this target CRN. The lowercase species are "intermediate" species of the implementations, while the uppercase species are "formal" species.

practice.

In our further discussions, we will restrict our attention to implementation CRNs that satisfy the condition that we call "tidiness." Informally stated, tidy CRNs are implementation CRNs which do not get "stuck" in the way that CRN4 got stuck above, i.e., they always can "clean up" intermediate species. This means that any intermediate species that are produced during the evolution of the system can eventually turn back into species of the target CRN. Of course, the algorithm we present in this paper for testing our definition of correctness will also be able to test whether the given implementation is tidy.

Finally, we briefly mention that many CRN implementations also involve what are called "fuel" and "waste" species, in addition to the already mentioned intermediate species. Fuel species are helper species that are assumed to be always present in the system at fixed concentration, whereas waste species are chemically inert species that sometimes get produced as a byproduct of implemented pathways. While our core theory addresses the version of the problem in which there is no fuel or waste species, as we demonstrate in Section 5, it can easily be extended to handle the general case with fuel and waste species, using existing tools.

2 Motivations for a new theory

To one who is experienced in formal verification, the problem seems to be closely related to various well-studied notions such as reachability, (weak) trace equivalence, (weak) bisimulation, serializability, etc. In this section, we briefly demonstrate why none of these traditional notions seems to give rise to a definition which is entirely satisfactory for the problem at hand.

The first notion we consider is reachability between formal states [15, 14, 7]. We call the species that appear in both the target and the implementation CRNs "formal," to distinguish them from species that appear only in the implementation CRN, which we call "intermediate." Formal states are defined to be states which do not contain any intermediate species. Since we are assuming that our implementation CRN is tidy, it then makes sense to ask whether the target CRN and the implementation CRN have the same reachability

when we restrict our attention to formal states only — this is an important distinction from the traditional Petri net reachability-equivalence problem. That is, given some formal state, what is the set of formal states that can be reached from that state using reactions from one CRN, as opposed to the other CRN? Do the target CRN and the implementation CRN give rise to exactly the same reachability for every formal initial state? While it is obvious that any "correct" implementation must satisfy this condition, it is also easy to see that this notion is not very strong. For example, consider the target CRN $\{A \to B,\ B \to C,\ C \to A\}$ and the implementation CRN $\{A \to i,\ i \to C,\ C \to j,\ j \to B,\ B \to k,\ k \to A\}$. Although the two CRNs are implementing opposite behaviors in some sense, they still give rise to the same reachability between purely formal states.

Trace equivalence [7, 11] is another notion of equivalence that is often found in formal verification literature. *Weak* trace equivalence requires that it should be possible to "label" the reactions of the implementation CRN to be either a reaction of the target CRN or a "null" reaction. This labeling must be such that for any formal initial state, any sequence of reactions that can take place in the target CRN should also be able to take place in the implementation CRN and vice versa, up to the interpretation specified by the given labeling. However, since trace equivalence only concerns reactions and not species, it turns out, in our setting, to be a very weak notion which does not even imply reachability. For example, consider the target CRN $\{A \rightleftharpoons B,\ A \to \emptyset\}$ and the implementation CRN $\{A \to i,\ B \to j,\ i \rightleftharpoons j,\ i \to \emptyset\}$.[1] Then we can label $i \to j$ to be $A \to B$, $j \to i$ to be $B \to A$, $i \to \emptyset$ to be $A \to \emptyset$, and everything else to be null. This shows that this implementation CRN is trace equivalent to the target CRN in the above sense, but it is clear that these two CRNs do not have the same reachability. Whether one can modify the notion of trace equivalence to make it more useful in our setting is an open question.

Bisimulation is perhaps the most influential notion of equivalence in state transition systems such as CRNs or Petri nets. A notion of CRN equivalence based on the idea of weak bisimulation is explored in detail in [5], and indeed it proves to be much more useful than the above two notions. For bisimulation equivalence of CRNs, each intermediate species is "interpreted" as some combination of formal species, such that in any state of the implementation CRN, the set of possible next non-trivial reactions is exactly the same as it would be in the formal CRN. However, one potential problem of this approach is that it demands a way of interpreting every intermediate species in terms of formal species. Therefore, if we implement the target CRN $\{A \to B,\ A \to C,\ A \to D\}$ as $\{A \to i,\ i \to B,\ i \to C,\ A \to j,\ j \to D\}$, we cannot apply this bisimulation approach because the intermediate i cannot be interpreted to be any of A, B, or C. Namely, calling it A would be a bad interpretation because i can never turn into D. Calling it B would be bad because i can turn into C whereas B should not be able to turn into C. For the same reason calling it C is not valid either.

Perhaps this example deserves closer attention. We name this type of phenomenon the "delayed choice" phenomenon, to emphasize the point that i does not choose whether to become a B or a C until the final reaction takes place. There are two reasons that it is interesting; firstly, there may be a sense in which this phenomenon is related to the efficiency of the implementation, because the use of delayed choice may allow for a smaller number of intermediate species in implementing the same CRN. Secondly, this phenomenon actually does arise in actual systems, as presented in [8].

We note an important distinction between the various notions of equivalence discussed here and those found in the Petri net literature. Whereas two Petri nets are compared for (reachability/trace/bisimulation)-equivalence for a particular initial state [11], we are concerned about the various notions of equivalence of two CRNs for *all* initial states. This distinction may limit the applicability of common verification methodologies and software tools [10, 1], since the set of initial states is by necessity always infinite (and the set of reachable states from a particular initial state may also be infinite). Finally, we note that [13] proposes yet another notion of equivalence based on serializability, which works on a limited class of implementations

[1] $A \to \emptyset$ denotes a decaying reaction which consumes A and produces nothing.

in which there is not too much crosstalk between pathways implementing different formal reactions. Interestingly, when restricted to such implementations, the notion of serializability and our notion of pathway decomposition have a close (if not exact) correspondence.

Our approach (originally developed in [17]) differs from any of the above in that we ignore the target CRN and pay attention only to the implementation CRN. Namely, we simply try to infer what CRN the given implementation would look like in a hypothetical world where we cannot observe the intermediate species. We call this notion "formal basis." We show that not only is the formal basis unique for any valid implementation, but it also has the nice property that a CRN that does not have any intermediate species has itself as its formal basis. This leads us to a nice definition of CRN equivalence; we can simply declare two CRNs to be equivalent if and only if they have the same formal basis. Therefore, unlike many other approaches, our definition is actually an equivalence relation and therefore even allows for the comparison of an implementation with another implementation.

3 Theory

3.1 Overview

In previous sections we saw that a reaction which is a single step in the target CRN gets implemented as a pathway of reactions which involves intermediate species whose net effect only changes the number of "formal" species molecules. For instance, the pathway $A \to i, i + B \to j, j \to C + k, k \to D$ involves intermediate molecules i, j, and k but the net effect of this pathway is to consume A and B and produce C and D. In this sense this pathway may be viewed as an implementation of $A + B \to C + D$.

In contrast, we will not want to consider the pathway $A \to i, i \to B, B \to j, j \to C$ to be an implementation of $A \to C$, even though its net effect is to consume A and produce C. Intuitively, the reason is that this pathway, rather than being an indivisible unit, looks like a composition of smaller unit pathways each implementing $A \to B$ and $B \to C$.

The core idea of our definition, which we call *pathway decomposition*, is to identify all the pathways which act as indivisible units in the above sense. The set of these "indivisible units" is called the formal basis of the given CRN. If we can show that all potential pathways in the CRN can be expressed as compositions of these indivisible units, then that will give us ground to claim that this formal basis may be thought of as the target CRN that the given CRN is implementing.

3.2 Basic definitions

We use upper case and lower case letters to denote formal and intermediate chemical species, respectively.

Definition 1. A **state** is a multiset of species. If every species in a state S is a formal species, then S is called a **formal state**.

Definition 2. If S is a state, Formal(S) denotes the multiset we obtain by removing all the intermediate species from S.

Definition 3. A **reaction** is a pair of multisets of species (R, P) and it is **trivial** if $R = P$. Here, R is called the set of **reactants** and P is called the set of **products**. We say that the reaction (R, P) **can occur** in the state S if $R \subseteq S$. If both R and P are formal states, then (R, P) is called a **formal reaction**. If $r = (R, P)$, we will sometimes use the notation $\bar{r}$ to denote the reverse reaction (P, R).

Definition 4. If (R, P) is a reaction that can occur in the state S, we write $S \oplus (R, P)$ to denote the resulting state $S - R + P$. As an operator, $\oplus$ is left-associative.

Definition 5. A **CRN** is a (nonempty) set of nontrivial reactions. A CRN that contains only formal reactions is called a **formal CRN**.

Definition 6. A **pathway** p of a CRN $\mathcal{C}$ is a (finite) sequence of reactions $(r_1, \ldots, r_k)$ with $r_i \in \mathcal{C}$ for all i. We say that a pathway can occur in the state S if all its reactions can occur in succession starting from S. Note that given any pathway, we can find a unique **minimal initial state** associated with it. For convenience, we will simply call it the **initial state** of the pathway. Correspondingly, the **final state** of a pathway will denote the state $S \oplus r_1 \oplus r_2 \oplus \cdots \oplus r_k$ where S is the (minimal) initial state of the pathway. If both the initial and final states of a pathway are formal, it is called a **formal pathway**. Note that the intermediate states within a formal pathway might not be formal!

Definition 7. A pathway $p = (r_1, \ldots, r_k)$ is called **futile** if

1. its initial state is equal to its final state, and
2. Formal$(S_i) \subseteq$ Formal(S_0) for all $1 \leq i \leq k$ where S_0 is the initial state of p and $S_i = S_0 \oplus r_1 \oplus r_2 \oplus \cdots \oplus r_i$.

To absorb these definitions, we can briefly study some examples. Consider the chemical reaction $2A + B \to C$. According to our definitions, this will be written $(\{A, A, B\}, \{C\})$. Here, $\{A, A, B\}$ is called the reactants and $\{C\}$ is called the products, just as one would expect. Note that this reaction can occur in the state $\{A, A, A, B, B\}$ but cannot occur in the state $\{A, B, C, C, C, C\}$ because the latter state does not have all the required reactants. If the reaction takes place in the former state, then the resulting state will be $\{A, B, C\}$ and thus we can write $\{A, A, A, B, B\} \oplus (\{A, A, B\}, \{C\}) = \{A, B, C\}$. In this paper, although we formally define a reaction to be a pair of multisets, we will interchangeably use the chemical notation whenever it is more convenient. For instance, we will often write $2A + B \to C$ instead of $(\{A, A, B\}, \{C\})$.

Note that we say that a pathway $p = (r_1, r_2, \ldots, r_k)$ can occur in the state S if r_1 can occur in S, r_2 can occur in $S \oplus r_1$, r_3 can occur in $S \oplus r_1 \oplus r_2$, and so on. For example, consider the pathway that consists of $2A + B \to C$ and $B + C \to A$. This pathway cannot occur in the state $\{A, A, B\}$ because even though the first reaction can occur in that state, the resulting state after the first reaction, which is $\{C\}$, will not have all the reactants required for the second reaction to occur. In contrast, it is easy to see that this pathway can occur in the state $\{A, A, B, B\}$.

One might wonder why we need the second condition in the definition of a futile pathway. The reason is that even when the initial state and final state are equal, a pathway cannot be considered futile if, at any point, it produces formal species that were not in the initial state. This intuition will be made clear in Section 3.4.

We also point out that we cannot directly express a reversible reaction in this formalism. Thus, a reversible reaction will be expressed using two independent reactions corresponding to each direction, e.g., $A \rightleftharpoons B$ will be expressed as two reactions $A \to B$ and $B \to A$.

Before we proceed, we formally define the notion of tidiness which we informally introduced in Section 1.

Definition 8. Let p be a pathway with a formal initial state and T its final state. Then, a (possibly empty) pathway $p' = (r_1, \ldots, r_k)$ is said to be the **closing pathway** of p if p' can occur in T and $T \oplus r_1 \oplus \cdots \oplus r_k$ is a formal state. A CRN is **weakly tidy** if every pathway with a formal initial state has a closing pathway.

As was informally explained before, this means that the given CRN is always capable of cleaning up all the intermediate species. For example, the CRN $\{A \to i,\ i + B \to C\}$ will not be weakly tidy because if the system starts from the state $\{A\}$, it can transition to the state $\{i\}$ and become "stuck" in a non-formal state: there does not exist a reaction to convert the intermediate species i back into some formal species.

For a more subtle example, let us consider the CRN $\{A \to i + B,\ i + B \to B\}$, which is weakly tidy according to the definition as stated above. In fact, it is easy to see that this implementation CRN will

never get stuck when it is operating by itself, starting with any formal initial state. However, this becomes problematic when we begin to think about composing different CRNs. Namely, when intermediate species require other formal species in order to get removed, the implementation CRN may not work correctly if some other formal reactions are also operating in the system. For instance, if the above implementation runs in an environment that also contains the reaction $B \to C$, then it is no longer true that the system is always able to get back to a formal state.

This is not ideal because the ability to compose different CRNs, at least in the case where they do not share any intermediate species, is essential for CRN implementations to be useful. To allow for this type of composition, we define a stronger notion of tidiness which is preserved under such composition.

Definition 9. A closing pathway is **strong** if its reactions do not consume any formal species. A CRN is **strongly tidy** if every pathway with a formal initial state has a strong closing pathway.

In the rest of the paper, unless indicated otherwise, we will simply say tidiness to mean strong tidiness. Similarly, we will simply say closing pathway to mean strong closing pathway.

		$A \to i + D$
$A \to i + j$	$A \to i$	$D \to E$
$i + j \to B$	$i + B \to C$	$E + i \to C$
strongly tidy	**not tidy**	**weakly tidy**

Figure 2: Some examples of tidy and non-tidy CRNs

3.3 Pathway decomposition

Now we formally define the notion of pathway decomposition. Following our intuition from Section 3.1, we first define what it means to implement a formal reaction.

Definition 10. We say that a pathway $p = (r_1, \ldots, r_k)$ **implements** a formal reaction (R, P) if it satisfies the following conditions.

1. $R = S$ and $P = T$, where S and T are the initial and final states of p.
2. Let $S_i = S \oplus r_1 \oplus \cdots \oplus r_i$ (so that $S_0, S_1, \ldots, S_k$ are all the states that p goes through). Then, there exists a **turning point** reaction $r_j = (R', P')$ such that Formal$(S_i) \subseteq S$ for all $i < j$, Formal$(S_i) \subseteq T$ for all $i \geq j$, and Formal$(S_{j-1} - R') = \emptyset$. When a formal pathway satisfies this condition, we call it **regular**.

While the first condition is self-evident, the second condition needs a careful explanation. It asserts that there should be a point in the pathway prior to which we only see the formal species from the initial state and after which we only see the formal species from the final state. The existence of such a "turning point" allows us to interpret the pathway as an implementation of the formal reaction (R, P) where in a sense the real transition is occurring at that turning point. Importantly, this condition rules out such counterintuitive implementations as $(A \to i,\ i \to C{+}j,\ C{+}j \to k,\ k \to B)$ or $(A \to i{+}B,\ i{+}B \to j{+}A,\ j{+}A \to B)$ as implementations of $A \to B$. Note that a formal pathway that consumes but does not produce formal species prior to its turning point, and thereafter produces but does not consume formal species, is by this definition regular, and this is the "typical case." However our definition also allows additional flexibility; for example, the reactants can fleetingly bind, as in $\{A \to i, i + B \to j, j \to B + k, k + B \to C\}$. One may also wonder why we need the condition Formal$(S_{j-1} - R') = \emptyset$. This is to prevent ambiguity that may arise in the

case of catalytic reactions. Consider the pathway $(A \to i + A, i \to B)$. Without the above condition, both reactions in this pathway qualify as a turning point, but the second reaction being interpreted as the turning point is counterintuitive because the product A gets produced before the turning point.

One problem of the above definition is that it interprets the pathway $(A \to i,\ A \to i,\ i \to B,\ i \to B)$ as implementing $A + A \to B + B$. As explained in Section 3.1, we would like to be able to identify such a pathway as a composition of smaller units.

Definition 11. A formal pathway p is **decomposable** if it satisfies any of the following two conditions:

1. (Decomposition into smaller units) p can be partitioned into two nonempty subsequences (which need not be contiguous) that are each formal pathways.
2. (Futile loop removal) p can be partitioned into two nonempty subsequences q and r such that q is a formal pathway and r is a futile pathway.

A pathway that is not decomposable is called **prime**. For clarity, note that when a sequence is partitioned into two subsequences, no reordering is allowed. Also note that the decomposition of a formal pathway into prime formal pathways need not be unique.

Definition 12. The set of prime formal pathways in a given CRN is called the **elementary basis** of the CRN. The **formal basis** is the set of $(\text{initial state}, \text{final state})$ pairs of the nonfutile pathways in the elementary basis. The elementary basis and/or the formal basis can be either finite or infinite.

Definition 13. A CRN is **regular** if every prime formal pathway implements some formal reaction. Equivalently, a CRN is regular if every prime formal pathway is regular.

Definition 14. Two tidy and regular CRNs are said to be **pathway decomposition equivalent** if their formal bases are identical.

3.4 Theorems

3.4.1 Properties

It is almost immediate that pathway decomposition equivalence satisfies many nice properties, some of which are expressed in the following theorems. Proofs, except the ones that are trivial, can be found in Appendix A.

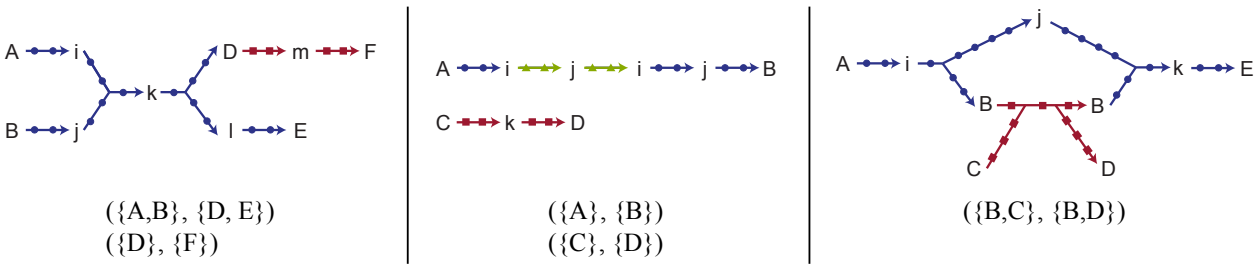

Figure 3: Three examples of decomposable formal pathways and their corresponding formal basis. The partition of reactions is marked by lines of different types and colors. Lines marked as light green with triangles denote a futile pathway. In the right most example, the decomposed pathway denoted by blue lines with circles is not regular and is therefore not part of the formal basis.

Theorem 3.1. *Pathway decomposition equivalence is an equivalence relation, i.e., it satisfies the reflexive, symmetric, and transitive properties.*

Theorem 3.2. *If $\mathcal{C}$ is a formal CRN with no trivial reactions, its formal basis is itself.*

Corollary 3.3. *If $\mathcal{C}_1$ and $\mathcal{C}_2$ are formal CRNs, they are pathway decomposition equivalent if and only if $\mathcal{C}_1 = \mathcal{C}_2$, up to removal or addition of trivial reactions.*

Theorem 3.4. *Any formal pathway of $\mathcal{C}$ can be generated by interleaving one or more prime formal pathways and futile pathways of $\mathcal{C}$.*

It is perhaps worth noting here that the decomposition of a formal pathway may not always be unique. For example, the pathway $(A \to i,\ B \to i,\ i \to C,\ i \to D)$ can be decomposed in two different ways: $(A \to i,\ i \to C)$ and $(B \to i,\ i \to D)$, and $(A \to i,\ i \to D)$ and $(B \to i,\ i \to C)$. Pathway decomposition differs from other notions such as (weak) bisimulation or (weak) trace equivalence in that it allows such degeneracy of interpretations. We note that such degeneracy, which is closely related to the previously mentioned delayed choice phenomenon, may permit a more efficient implementation of a target CRN in terms of the number of species or reactions used in the implementation CRN. For example, if we wish to implement the CRN consisting of reactions $A \rightleftharpoons B$, $A \rightleftharpoons C$, and $B \rightleftharpoons C$, it may be more efficient to implement it as reactions $A \rightleftharpoons i$, $B \rightleftharpoons i$, and $C \rightleftharpoons i$ than to implement all 6 reactions separately.

The following theorems illuminate the relationship between a tidy and regular CRN $\mathcal{C}$ and its formal basis $\mathcal{F}$ and how to better understand this degeneracy of interpretations.

Definition 15. We say a pathway $p = (r_1, \ldots, r_k)$ in $\mathcal{C}$ can be **interpreted** as a pathway $q = (s_1, \ldots, s_l)$ in $\mathcal{F}$ if

1. q can occur in the initial state S of p,
2. $S \oplus r_1 \oplus \cdots \oplus r_k = S \oplus s_1 \oplus \cdots \oplus s_l$, and
3. there is a decomposition of p into prime formal pathways and futile pathways such that if we replace the turning point reaction of each prime formal pathway (as implicitly defined in Section 3.3) with the corresponding element of $\mathcal{F}$ and remove all other reactions, the result is q up to removal of trivial reactions.

It is clear that the interpretation may not be unique, because there can be many different decompositions of p as well as many different choices of the turning point reactions. One might also wonder why we do not simply require that p and q must have the same initial states. This is because p may contain futile pathways that may unmeaningfully "blow up" its initial state, e.g., $A \to i, i + B \to j, j \to i + B, i \to C$ decomposes into $A \to i, i \to C$ and a futile pathway $i + B \to j, j \to i + B$. The original pathway has $\{A, B\}$ as its initial state, but the interpretation will only have $\{A\}$.

Theorem 3.5. *Suppose $\mathcal{C}$ is a tidy and regular CRN and $\mathcal{F}$ is its formal basis.*

1. *For any pathway q in $\mathcal{F}$, there exists a pathway p in $\mathcal{C}$ whose initial and final states are equal to those of q, such that p can be interpreted as q.*
2. *Any formal pathway p in $\mathcal{C}$ can be interpreted as some pathway q in $\mathcal{F}$.*

In our final theorem we prove that pathway decomposition equivalence implies formal state reachability equivalence. Note that the converse is not true because $\{A \to B, B \to C, C \to A\}$ is not pathway decomposition equivalent to $\{A \to C, C \to B, B \to A\}$.

Theorem 3.6. *If two tidy and regular CRNs $\mathcal{C}_1$ and $\mathcal{C}_2$ are pathway decomposition equivalent, they give rise to the same reachability between formal states.*

3.4.2 Modular composition of CRNs

In this section, we prove some theorems that pathway decomposition equivalence is preserved under composition of CRNs, as long as those CRNs do not share any intermediate species.

Theorem 3.7. *If $\mathcal{C}$ and $\mathcal{C}'$ are two tidy CRNs that do not share any intermediate species, then $\mathcal{C} \cup \mathcal{C}'$ is also tidy.*

Theorem 3.8. *If $\mathcal{C}$ and $\mathcal{C}'$ are two regular CRNs that do not share any intermediate species, then $\mathcal{C} \cup \mathcal{C}'$ is also regular.*

Theorem 3.9. *Let $\mathcal{C}$ and $\mathcal{C}'$ be two tidy and regular CRNs that do not share any intermediate species, and $\mathcal{F}$ and $\mathcal{F}'$ their formal bases respectively. Then the formal basis of $\mathcal{C} \cup \mathcal{C}'$ is exactly $\mathcal{F} \cup \mathcal{F}'$.*

4 Algorithm

In this section, we present a simple algorithm for finding the formal basis of a given CRN. The algorithm can also test tidiness and regularity.

Our algorithm works by enumerating pathways that have formal initial states. The running time of our algorithm depends on a quantity called maximum width, which can be thought of as the size of the largest state that a prime formal pathway can ever generate. Unfortunately it is easy to see that this quantity is generally unbounded; e.g., $\{A \to i, i \to i + i, i \to \emptyset\}$ has a finite formal basis $\{A \to \emptyset\}$ but it can generate arbitrarily large states.[2] However, since such implementations are highly unlikely to arise in practice, in this paper we focus on the bounded width case. We note that even in the bounded width case it is still nontrivial to come up with an algorithm that finishes in finite time, because it is unclear at what width we can safely stop the enumeration.

4.1 Exploiting bounded width

We begin by introducing a few more definitions and theorems.

Definition 16. A pathway that has a formal initial state is called **semiformal**.

Definition 17. A semiformal pathway p is **decomposable** if it satisfies any of the following two conditions:

1. (Decomposition into smaller units) p can be partitioned into two nonempty subsequences (which need not be contiguous) that are each semiformal pathways.
2. (Futile loop removal) p can be partitioned into two nonempty subsequences q and r such that q is a semiformal pathway and r is a futile pathway.

It is obvious that this reduces to our previous definition of decomposability if p is a formal pathway.

Definition 18. Let $p = (r_1, \ldots, r_k)$ be a pathway and let $S_i = S \oplus r_1 \oplus \cdots \oplus r_i$ where S is the initial state of p. The **width** of p is defined to be $\max_i |S_i|$.

Definition 19. The **branching factor** of a CRN $\mathcal{C}$ is defined to be the following value.

$$\max_{(R,P) \in \mathcal{C}} \max\{|R|, |P|\}$$

We note that many implementations that arise in practice have small branching factors.

[2] Clearly, there may also be cases where the formal basis itself is infinite, e.g. $\{A \to i, i \to i + i, i \to B\}$.

Theorem 4.1. *Suppose a pathway p is obtained by interleaving pathways $p_1, \ldots, p_l$. Let S be the initial state of p and $S_1, \ldots, S_l$ the initial states of $p_1, \ldots, p_l$ respectively. Then, $S \subseteq S_1 + S_2 + \cdots + S_l$.*

Theorem 4.2. *If p is an undecomposable semiformal pathway of width $w > 0$, there exists an undecomposable semiformal pathway of width smaller than w but at least $(w - b)/b$, where b is the branching factor of the CRN. (Note that if $w = 1$, the lower bound $(w - b)/b$ might be negative. In this case, it would simply mean that there exists an undecomposable semiformal pathway of width 0, which would be the empty pathway.)*

Corollary 4.3. *Suppose w and w_{max} are integers such that $(w + 1)b \leq w_{max}$. Then, if there is no undecomposable semiformal pathway of width greater than w and less than or equal to w_{max}, then there exists no undecomposable semiformal pathway of width greater than w.*

4.2 Sketch of the algorithm

While Corollary 4.3 gives us a way to exploit the bounded width assumption, it is still unclear whether the enumeration can be made finite, because the number of undecomposable semiformal pathways of bounded width may still be infinite. For example, if the CRN consists of $\{A \to i,\ i \to A,\ i + B \to i + C,\ i + C \to i + B\}$, we have infinitely many undecomposable semiformal pathways of width 2, because after the initial reaction $A \to i$, the segment $i + B \to i + C,\ i + C \to i + B$ can be repeated arbitrarily many times without ever making the pathway decomposable. In this section, we provide a high-level overview of how we resolve this difficulty to obtain a finite-time algorithm. A detailed exposition of the algorithm can be found in Appendix B.

The principal technique that lets us avoid enumerating infinitely many pathways is memoization. To use memoization, we first define what is called the **signature** of a pathway, which is a collection of information about many important properties of the pathway, such as its initial and final states, decomposability, etc. It turns out that the number of possible signatures of bounded width pathways is always finite, even if the number of pathways themselves may be infinite. This means that in the enumeration algorithm there is no need to duplicate pathways that have the same signatures, as long as signatures give us all the information that is necessary to determine the formal basis and to test tidiness and regularity.

Therefore, the algorithm consists in simply enumerating all semiformal pathways of width up to $(w + 1)b$, where w is the maximum width of the undecomposable semiformal pathways discovered so far, while excluding pathways that have the same signatures as previously discovered pathways. It is important to emphasize that no a priori knowledge of the width bound is assumed, and the algorithm is guaranteed to halt as long as there exists some finite bound. While the existence of this algorithm shows that the problem of finding the formal basis is decidable with the bounded width assumption, the worst-case time complexity seems to be adverse as is usual for enumeration-based algorithms. It is an open question to understand the computational complexity of this problem as well as to find an algorithm that has a better practical performance. Another important open question is whether the problem without the bounded width assumption is decidable.

5 Handling the general case

In this section, we discuss some important issues that pertain to practical applications and hint at the possibility of further theoretical investigations.

As we briefly mentioned earlier, many CRN implementations that arise in practice involve not only formal and intermediate species but also what are called fuel and waste species. Fuel species are chemical species that are assumed to be always present in the system at fixed concentration, as in a buffer. For

instance, DNA implementations [18, 2, 16] often employ fuel species that are present in the system in large concentration and have the ability to transform formal species into various other intermediates. This type of "implementation" is also prevalent in biological systems, where the concentrations of energy-carrying species such as ATP, synthetic precursors such as NTPs, and general-purpose enzymes such as ribosomes and polymerases, are all maintained in roughly constant levels by the cellular metabolism.

In CRN verification, the standard approach to fuel species is to preprocess implementation CRNs such that all occurrences of fuel species are simply removed. For instance, if the CRN contained reaction $A+g \to i+t$ where g and t are fuel species, the preprocessed CRN will only have $A \to i$. The justification for this type of preprocessing is that since fuel species are always present in the system in large concentration by definition, consuming or producing a finite number of fuel species molecules do not have any effect on the system. While no formal justification has been established for this technique, it is nevertheless widely accepted in the field at the present time.

On the other hand, implementations sometimes produce "waste" species as byproducts. Waste species are supposed to be chemically inert and thus cannot have interaction with other formal or intermediate species. However, in practice it is often difficult to implement a chemical species which is completely inert and therefore they may interact with other species in trivial or nontrivial ways. Therefore the main challenge is to ensure that these unwanted interactions do not give rise to a logically erroneous behavior. One way to deal with this problem is to define some formal notion of waste and preprocess those species classified as wastes in a similar manner to fuel species, but it has been difficult to arrive at a definition of waste that is completely satisfactory to all.

Another related problem which must be solved before we can use pathway decomposition is that some implementations may have multiple chemical species that are interpreted as the same formal species. (For example, see DNA implementations [18, 2] with "history domains.") Since our mathematical framework implicitly assumes one-to-one correspondence between formal species of the target CRN and formal species of the implementation CRN, it is not immediately clear how we can apply our theory in such cases.

Interestingly, the weak bisimulation-based approach to CRN equivalence proposed in [5] does not seem to suffer from any of these problems. This is mainly because this approach in fact does not have any particular distinction between these different types of species except fuel species. Rather, it requires that there must be a way to interpret each species that appears in the implementation CRN as one or more formal species. For instance, if $\{A \rightleftharpoons i, B+i \rightleftharpoons j, j \to C\}$ is proposed as an implementation of $A+B \to C$, the weak bisimulation approach will interpret A and i as $\{A\}$, B as $\{B\}$, j as $\{A,B\}$, and C as $\{C\}$. Therefore the state of the system at any moment will have an instantaneous interpretation as some formal state, which is not provided by pathway decomposition. On the other hand, the weak bisimulation approach cannot handle interesting phenomena that are allowed in the pathway decomposition approach, most notably the delayed choice phenomenon explained in Section 2.

Our proposed solution to the problem of wastes and multiple formal labeling is a hybrid approach between weak bisimulation and pathway decomposition. Namely, we take the implementation CRN from which only the fuel species have been preprocessed, and tag as "formal" species all the species that have been labeled by the user as either an implementation of a target CRN species or a waste. All other species are tagged as "intermediates". Then we can apply the theory of pathway decomposition to find its formal basis (with respect to the tagging, as opposed to the smaller set of species in the target CRN). Note that waste species must be tagged as "formal" rather than "intermediate" because they will typically accumulate, and thus tagging them as "intermediate" would result in a non-tidy CRN to which pathway decomposition theory does not apply. Finally, we verify that the resulting formal basis of tagged species is weak bisimulation equivalent to the target CRN under the natural interpretation, which interprets implementations of each target CRN species as the target CRN species itself and wastes as "null." If the implementation is incorrect, or if some species was incorrectly tagged as "waste", the weak bisimulation test will fail. See Figure 4 for example.

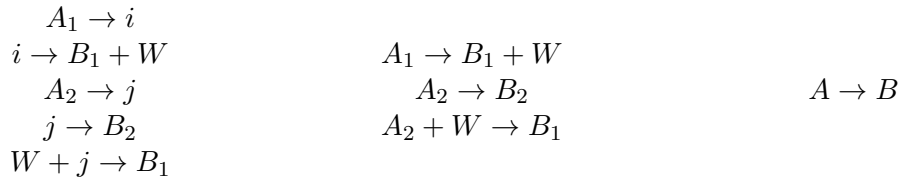

Figure 4: The hybrid approach. We first apply pathway decomposition, treating the upper case species as formal species and lower case species as intermediate species. Then, we apply weak bisimulation using the natural interpretation which interprets A_1 and A_2 as $\{A\}$, B_1 and B_2 as $\{B\}$, and W as $\emptyset$. Thus, in two steps, the implementation CRN has been shown to be a correct implementation of $\{A \to B\}$.

On the other hand, we note that the weak bisimulation approach can sometimes handle interesting cases which pathway decomposition cannot. For instance, the design proposed in [16] for reversible reactions implements $A + B \rightleftharpoons C + D$ as $\{A \rightleftharpoons i, i + B \rightleftharpoons j, j \rightleftharpoons k + C, k \rightleftharpoons D\}$. Note that this implementation CRN is not regular according to our theory because of the prime formal pathway $A \to i, i + B \to j, j \to k + C, k + C \to j, j \to i + B, i \to A$. Interestingly, this type of design seems to directly oppose the foundational principles of the pathway decomposition approach. One of the key ideas that inspired pathway decomposition is that of "base touching," namely the idea that even though the evolution of the system involves many intermediate species, a pathway implementing a formal reaction must eventually produce all its formal products and thus "touch the base." This principle is conspicuously violated in the above pathway, because while the only intuitive way to interpret it is as $A + B \to C + D$ and then $C + D \to A + B$, the first part does not touch the base by producing a D molecule. In contrast, the weak bisimulation approach naturally has no problem handling this implementation: i is interpreted as $\{A\}$, j is interpreted as $\{A, B\}$, and k is interpreted as $\{D\}$.

The fact that the two approaches are good for different types of instances motivates us to further generalize the hybrid approach explained above. Formally, we can define the generalized hybrid approach as follows.

Definition 20. Suppose we are given a target CRN $\mathcal{C}_1$ and an implementation CRN $\mathcal{C}_2$. Let $\mathcal{F}$ and $\mathcal{S}$ denote the species of $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively. Let $\mathcal{X} \subseteq S$ be the set of species that have been labeled by the user as implementations of target CRN species or wastes.

In the hybrid approach, we say $\mathcal{C}_2$ is a correct implementation of $\mathcal{C}_1$ if there exists some $\mathcal{X} \subseteq \mathcal{V} \subseteq S$ such that the formal basis of $\mathcal{C}_2$ with respect to $\mathcal{V}$ as formal species is weak bisimulation equivalent to $\mathcal{C}_1$ under some interpretation that respects the labels on $\mathcal{X}$ provided by the user.

Note that the weak bisimulation approach provides an "interpretation map" m from $\mathcal{V}$ to states of $\mathcal{C}_1$ [5]. Although the domain m is formally $\mathcal{V}$, there is an obvious sense in which we can apply it to formal states, formal reactions, or pathways consisting of formal reactions, e.g., $m(S) = \sum_{x \in S} m(x)$ for a state S, $m((R, P)) = (m(R), m(P))$ for a reaction (R, P), etc. To prove a theorem that provides a sense in which the hybrid approach is correct, we first extend the notion of interpretation of pathways that we introduced in Section 3.4 to include this concept of interpretation map.

Definition 21. Suppose m is a map from $\mathcal{V}$ to states of $\mathcal{C}_1$, where $\mathcal{V}$ is the set of formal species of $\mathcal{C}_2$. We say a pathway $p = (r_1, \ldots, r_k)$ in $\mathcal{C}_2$ can be **interpreted** as a pathway $q = (s_1, \ldots, s_l)$ in $\mathcal{C}_1$ under m if

1. q can occur in $m(S)$, where S is the initial state of p,

2. $m(S \oplus r_1 \oplus \cdots \oplus r_k) = m(S) \oplus s_1 \oplus \cdots \oplus s_l$, and
3. there is a decomposition of p into prime formal pathways and futile pathways such that if we replace the turning point reaction of each prime formal pathway with the corresponding element of $\mathcal{C}_1$ and remove all other reactions and call the resulting pathway p', then $m(p')$ is equal to q up to removal of trivial reactions.

Theorem 5.1. *Suppose an implementation CRN $\mathcal{C}_2$ is a correct implementation of the target CRN $\mathcal{C}_1$ according to the hybrid approach. Then, there exists a mapping m from $\mathcal{V}$ to states of $\mathcal{C}_1$ such that the following two conditions hold.*

1. *Let q and S be a pathway and a state in $\mathcal{C}_1$ such that q can occur in S. Then, for any state S' that uses species from $\mathcal{V}$ such that $m(S') = S$, there exists a pathway p in $\mathcal{C}_2$ that can occur in S' and can be interpreted as q under m.*
2. *Any formal pathway p in $\mathcal{C}_2$ can be interpreted as some pathway q in $\mathcal{C}_1$ under m.*

The above theorem provides a sense in which the hybrid definition is a good notion, similar to one provided by Theorem 3.5. While this approach seems to be the most general approach proposed thus far in terms of the range of implementations that it can address, its exact capabilities are not completely understood at the present. Probing the potential of this approach is an important open question.

6 Conclusions

In this paper, we studied the problem of formal verification of CRN implementations, and proposed a solution which is unique and distinctive in its concept from the traditional formal verification methods such as reachability or (weak) bisimulation. We further showed how our approach can be combined with weak bisimulation to handle a wider class that, to our best knowledge, includes all currently known enzyme-free DNA implementation techniques.

While we focused almost exclusively on the theoretical aspects of the problem in this paper, we mention that an earlier incarnation of this theory has been implemented and tested in practice for DNA-based implementations [17]. To do this, the formal basis enumeration algorithm was implemented and interfaced with other software pieces that together form the verification pipeline. First, the given target CRN was converted into a set of DNA molecules using the BioCRN compiler presented in [17]. Second, all the reactions that can occur between these DNA molecules were enumerated using an earlier incarnation of the domain-level DNA reaction enumerator described in [8]. Note that these reactions form the implementation CRN. Finally, the formal basis enumerator was run on the implementation CRN to see if it yields a formal basis that is equal to the target CRN. Despite the apparent brute-force nature, the algorithm successfully tested implementation CRNs of size up to one hundred reactions (the implementation CRN is typically an order of magnitude larger than the corresponding target CRN). Interestingly, the algorithm finished in less than a second for most test cases, whereas there were certain implementations that had a tendency to blow up the search space (i.e. number of possible signatures) and make the algorithm fail to halt in three hours even though the size of the implementation CRN was comparable.

Concluding the paper, we quickly sketch major open questions remaining in the area.

1. In reaction enumeration which was briefly mentioned in Section 1, a pathway consisting of multiple reaction steps are often truncated to yield a one-step reaction. It is tempting to try to use our theory to formally prove that this truncation step is "correct," because it seems that the weak bisimulation approach cannot handle this problem in its full generality. (As shown in [8], there are DNA constructions which yield CRNs of the form $\{A + B \to i, i \to j, i \to k, j \to C + D, k \to X + Y\}$, which clearly exhibits the delayed choice phenomenon.)

2. Can we better understand the delayed choice phenomenon? As we mentioned before, it seems that there should be a sense in which this phenomenon is related to the efficiency of the implementation, especially in terms of the number of intermediate species used in an implementation. Can we identify cases where this phenomenon naturally arises and find a practical use for it? Could there be some sense of an "optimizing compiler" which makes use of this phenomenon?
3. What is the computational complexity of the problem of finding the formal basis of a given CRN (for both bounded and unbounded width cases)? Can we improve upon the algorithm presented herein?
4. Can we better understand the "hybrid approach" proposed in Section 5? What is the class of implementations it can handle? Is there a notion of equivalence which is even more general? Is there an efficient algorithm for testing hybrid equivalence? Can we show that preprocessing to remove fuels from the implementation CRN is equivalent to tagging them as "formal" and then removing them in a post processing step?
5. Can we incorporate chemical kinetics into existing theories of CRN verification? (See [3] for recent developments.)

References

[1] R. Brayton and A. Mishchenko. ABC: An academic industrial-strength verification tool. In *Computer Aided Verification*, pages 24–40. Springer, 2010.

[2] L. Cardelli. Strand algebras for DNA computing. *Natural Computing: an international journal*, 10(1):407–428, 2011.

[3] L. Cardelli. Morphisms of reaction networks that couple structure to function. Manuscript, 2014.

[4] Y.-J. Chen, N. Dalchau, N. Srinivas, A. Phillips, L. Cardelli, D. Soloveichik, and G. Seelig. Programmable chemical controllers made from DNA. *Nature nanotechnology*, 8(10):755–762, 2013.

[5] Q. Dong. A bisimulation approach to verification of molecular implementations of formal chemical reaction networks. Master's thesis, Stony Brook University, 2012.

[6] S. M. Douglas, I. Bachelet, and G. M. Church. A logic-gated nanorobot for targeted transport of molecular payloads. *Science*, 335(6070):831–834, 2012.

[7] J. Esparza and M. Nielsen. Decidability issues for Petri nets. *Petri nets newsletter*, 94:5–23, 1994.

[8] C. Grun, K. Sarma, B. Wolfe, S. W. Shin, and E. Winfree. A domain-level DNA strand displacement reaction enumerator allowing arbitrary non-pseudoknotted secondary structures. In *VEMDP*, 2014.

[9] H. Gu, J. Chao, S.-J. Xiao, and N. C. Seeman. A proximity-based programmable DNA nanoscale assembly line. *Nature*, 465(7295):202–205, 2010.

[10] G. J. Holzmann. The model checker SPIN. *IEEE Transactions on software engineering*, 23(5):279–295, 1997.

[11] P. Jančar, J. Esparza, and F. Moller. Petri nets and regular processes. *Journal of Computer and System Sciences*, 59(3):476–503, 1999.

[12] M. R. Lakin and A. Phillips. *Visual DSD*. Microsoft Research.

[13] M. R. Lakin, A. Phillips, and D. Stefanovic. Modular verification of DNA strand displacement networks via serializability analysis. In *Proceedings of the 19th International Conference on DNA Computing and Molecular Programming*, 2013.

[14] R. Lipton. The reachability problem requires exponential space. *Research Report 62, Department of Computer Science, Yale University, New Haven, Connecticut*, 1976.

[15] E. Mayr. Persistence of vector replacement systems is decidable. *Acta Informatica*, 15(3):309–318, 1981.

[16] L. Qian, D. Soloveichik, and E. Winfree. Efficient Turing-universal computation with DNA polymers. *Lecture Notes in Computer Science*, 6518:123–140, 2011.

[17] S. W. Shin. Compiling and verifying DNA-based chemical reaction network implementations. Master's thesis, California Institute of Technology, 2011.

[18] D. Soloveichik, G. Seelig, and E. Winfree. DNA as a universal substrate for chemical kinetics. *Proceedings of the National Academy of Sciences*, 107:5393–5398, 2010.

A Proofs

Proof of Theorem 3.5. 1. Replace each reaction in q with the corresponding prime formal pathway of $\mathcal{C}$.

2. Fix a decomposition of p and pick a turning point for each prime formal pathway. Replace the turning points with the corresponding formal basis element and remove all other reactions. We call the resulting pathway q.

   To show that q can occur in the initial state of p, we start by proving the following simple lemma.

   Lemma A.1. *Let p be a formal pathway that can be decomposed into a formal pathway p' and a futile pathway p''. Then, the initial state of p' is contained in the initial state of p.*

   Now, we remove from p all the reactions that have been labeled as part of a futile pathway and call the resulting pathway p'. By the above lemma, p' can occur in the initial state of p.

   We finish the proof by showing that $q = (s_1, \ldots, s_l)$ can occur in the initial state S of p' using a hybrid argument. Define p'_j to be the pathway obtained by replacing the first j turning points in p' by the corresponding formal basis elements and removing all other reactions that belong to the same prime formal pathways. In particular, note that $p'_0 = p'$ and $p'_l = q$.

   It suffices to show that p'_j can occur in the initial state of p'_{j-1} for all $j > 0$. First, write $p'_{j-1} = (r_1, \ldots, r_m)$ and $p'_j = (r_{i_1}, \ldots, r_{i_k}, s_j, r_{i_{k+1}}, \ldots, r_{i_n})$. Then it follows from the definition of a turning point that $\text{Formal}(S \oplus r_{i_1} \oplus \cdots \oplus r_{i_{x-1}}) \supseteq \text{Formal}(S \oplus r_1 \oplus \cdots \oplus r_{i_x - 1})$ for every $1 \leq x \leq k$. Therefore $(r_{i_1}, \ldots, r_{i_k})$ can occur in S. Moreover, since the definition of a turning point asserts that all the reactants must be consumed at the turning point, it also implies that $\text{Formal}(S \oplus r_{i_1} \oplus \cdots \oplus r_{i_k}) \supseteq \text{Formal}(S \oplus r_1 \oplus \cdots \oplus r_{t-1} - X + R)$ where $r_t = (X, Y)$ denotes the turning point that is being replaced by s_j in this round and R denotes the reactants of s_j. Therefore, s_j can occur in $S \oplus r_{i_1} \oplus \cdots \oplus r_{i_k}$. Finally, it again follows from the definition of a turning point that $\text{Formal}(S \oplus r_{i_1} \oplus \cdots \oplus r_{i_k} \oplus s_j \oplus r_{i_{k+1}} \oplus \cdots \oplus r_{i_{x-1}}) \supseteq \text{Formal}(S \oplus r_1 \oplus \cdots \oplus r_{i_x - 1})$ for every $k + 1 \leq x \leq n$. We conclude that $(r_{i_1}, \ldots, r_{i_k}, s_j, r_{i_{k+1}}, \ldots, r_{i_n}) = p'_j$ can occur in S.

□

Proof of Theorem 3.6. Suppose there is a formal pathway p in $\mathcal{C}_1$ whose initial state is S and final state is T. By Theorem 3.5, it can be interpreted as some pathway q consisting of the reactions in the formal basis of $\mathcal{C}_1$. Since $\mathcal{C}_1$ and $\mathcal{C}_2$ have the same formal basis, by another application of Theorem 3.5, there exists some formal pathway p' in $\mathcal{C}_2$ that can be interpreted as q. Moreover, p' can occur in S and will yield a final state T. By symmetry between $\mathcal{C}_1$ and $\mathcal{C}_2$, the theorem follows. □

Proof of Theorem 3.7. Suppose p is a pathway of $\mathcal{C} \cup \mathcal{C}'$ that has a formal initial state. Since $\mathcal{C}$ and $\mathcal{C}'$ do not share intermediate species, we can partition the intermediate species found in the final state of p into two multisets A and A', corresponding to the intermediate species used by $\mathcal{C}$ and $\mathcal{C}'$ respectively. Now, if we remove from p all the reactions that belong to $\mathcal{C}'$ and call the resulting pathway q, then the multiset of all the intermediate species found in the final state of q will be exactly A. This is because the removed reactions, which belonged to C', cannot consume or produce any intermediate species used by $\mathcal{C}$. Since $\mathcal{C}$ is tidy, q has a closing pathway r. This time, remove from p all the reactions that belong to $\mathcal{C}$ and call the resulting pathway q'. By a symmetric argument, q' must have a closing pathway r'. Now observe that $r + r'$ is a closing pathway for p. □

Proof of Theorem 3.8. Let p be a prime formal pathway in $\mathcal{C} \cup \mathcal{C}'$. Partition p into two subsequences q and q', which contains all reactions of p which came from $\mathcal{C}$ and $\mathcal{C}'$ respectively. Since the two CRNs do not share any intermediate species, it is clear that q and q' must be both formal. Since p was prime, it implies that one of q and q' must be empty. Therefore, p is indeed a prime formal pathway in either $\mathcal{C}$ or $\mathcal{C}'$, and since each was a regular CRN, p must be regular. □

Proof of Theorem 3.9. Let p be a prime formal pathway in $\mathcal{C} \cup \mathcal{C}'$. By the same argument as in the proof of Theorem 3.8, p is a prime formal pathway of either $\mathcal{C}$ or $\mathcal{C}'$. Therefore, the formal basis of $\mathcal{C} \cup \mathcal{C}'$ is contained in $\mathcal{F} \cup \mathcal{F}'$. The other direction is trivial. □

Proof of Theorem 4.2. Since $w > 0$, p is nonempty. Let p_{-1} denote the pathway obtained by removing the last reaction (R, P) from p. Also, let $S_0, \ldots, S_k$ be the states that p goes through, and $S'_0, \ldots, S'_{k-1}$ the states that p_{-1} goes through. S_i is potentially unequal to S'_i because if the last reaction in p consumes some new formal species, then the minimal initial state of p_{-1} might be smaller than that of p. We note that no decomposition of p_{-1} can include a futile loop, because this would directly imply that p also contains a futile loop, which is a contradiction.

It is obvious that the minimal initial state of p_{-1} is smaller than the minimal initial state of p by at most $|R|$, i.e., $|S_0| - |S'_0| \leq |R|$. This means that for all $0 \leq i \leq k-1$, we have that $|S_i| - |S'_i| \leq |R|$. Clearly, if there exists some $0 \leq i \leq k-1$ such that $|S_i| = w$, then $|S'_i| \geq |S_i| - |R| = w - |R| \geq w - b$, so p_{-1} has width at least $w - b$. If there exists no such i, then we have that $|S_k| = w$. Clearly, $|S_{k-1}| = |S_k| - |P| + |R|$ and it follows that

$$|S_k| - |P| + |R| - |S'_{k-1}| = |S_{k-1}| - |S'_{k-1}| \leq |R|.$$

This is equivalent to $|S_k| - |S'_{k-1}| \leq |P|$. Since $|S_k| = w$, we have that $|S'_{k-1}| \geq w - |P| \geq w - b$. Thus, p_{-1} achieves width at least $w - b$.

Then, we decompose p_{-1} until it is no longer decomposable. As a result, we will end up with $l \geq 1$ undecomposable pathways $p_1, p_2, \ldots, p_l$ which by interleaving can generate p_{-1}. Also, they are all semiformal. First, we show that l is at most b. Assume towards a contradiction that $l > b$. Then, by the pigeonhole principle, there exists i such that (R, P) can occur in the sum of the final states of $p_1, \ldots, p_{i-1}, p_{i+1}, \ldots, p_l$ (since $|R| \leq b$ and (R, P) can occur in the sum of the final states of $p_1, \ldots, p_l$, the b reactants of (R, P) are distributed among $l > b$ pathways and there exists at least one p_i that does not provide a reactant and can be omitted). Then, consider the decomposition (p_i, p'_i) of p_{-1} where p'_i denotes the pathway we obtain by interleaving $p_1, \ldots, p_{i-1}, p_{i+1}, \ldots, p_l$ in the same order that those reactions occur in p_{-1}. By Theorem 4.1, p'_i is semiformal. Since p_i's are all semiformal, the theorem also tells us that the intermediate species in the final state of p'_i will be exactly the same as those in the sum of the final states of $p_1, \ldots, p_{i-1}, p_{i+1}, \ldots, p_l$. That is, the final state of p'_i contains all the intermediate species that (R, P) needs to occur, i.e., p'_i with (R, P) appended at the end should have a formal initial state. However, this means that p is decomposable which is a contradiction. Hence, $l \leq b$.

Now, note that if we have l pathways each with widths $w_1, \ldots, w_l$, any pathway obtained by interleaving them can have width at most $\sum_{i=1}^{l} w_i$ (for the same reason as in the previous paragraph). Since p_{-1} had width at least $w - b$, we have that $w - b \leq \sum_{i=1}^{l} w_i$. Then, if $w_i < (w-b)/b$ for all i, then $\sum_{i=1}^{l} w_i < w - b$, which is contradiction. Thus, we conclude that at least one of $p_1, \ldots, p_l$ has width greater than or equal to $(w - b)/b$. It is also clear that its width cannot exceed w. Thus, we have found a pathway p' which

1. has a smaller length than p, and
2. has width at least $(w - b)/b$ and at most w.

If p' has width exactly w, then we have failed to meet the requirements of the claim. However, since we have decreased the length of the pathway by at least one and the width of a zero-length pathway is always 0, we can eventually get a smaller width than w by repeating this argument. □

Proof of Corollary 4.3. Assume towards a contradiction that there exists an undecomposable semiformal pathway p of width $w' > w$. If $w' \leq w_{\max}$, then it is an immediate contradiction. Thus, assume that $w' > w_{\max}$. By Theorem 4.2, we can find an undecomposable semiformal pathway q of width v where $(w'-b)/b \leq v < w'$. Since $w' > w_{\max}$, we have that $v \geq (w'-b)/b > (w_{\max}-b)/b \geq ((w+1)b-b)/b = w$.

If $v \leq w_{\max}$, we have a contradiction. If $v > w_{\max}$, then take q as our new p and repeat the above argument. Since v is smaller than w' by at least one, we will eventually reach a contradiction.

Thus, there exists no undecomposable semiformal pathway of width greater than w. □

Proof of Theorem 5.1. Let m be the interpretation map provided by the weak bisimulation equivalence [5], and $\mathcal{I}$ the formal basis of $\mathcal{C}_2$ with respect to $\mathcal{V}$ as formal basis.

1. By the equivalence theorem in Section 2.3 of [5], we have a pathway p' in $\mathcal{I}$ that can occur in S' and $m(p') = q$. Now replace each reaction in p' by the prime formal pathway that implements that reaction and call the resultilng pathway p. Clearly, p can occur in S'. To show that p can be interpreted as q under m, observe that the first condition follows from the fact that q can clearly occur in $m(S')$ and S' is a superset of the initial state of p (because p can occur in S'). Since p and p' have the same initial and final states and $m(p') = q$, we also satisfy the second condition. The final condition trivially follows from the way p was constructed and the fact that $m(p') = q$.
2. By Theorem 3.5, p can be interpreted as some pathway p' in $\mathcal{I}$. Let q be the pathway we obtain by removing all the trivial reactions from $m(p')$. Now we show that p can be interpreted as q under m. For the first condition, use the equivalence theorem in Section 2.3 of [5] to see that q can occur in $m(S)$, where S is the initial state of p' and therfore also of p. The second condition follows immediately from the way q was constructed and the fact that p and p' have the same net effect. The final condition follows from the fact that p can be interpreted as p' in $\mathcal{I}$ and that $m(p') = q$ up to removal of trivial reactions.

□

Proof of Lemma A.1. Let S and S' denote the initial state of $p = (r_1, \ldots, r_k)$ and $p' = (r_{i_1}, \ldots, r_{i_l})$ respectively. Assume towards a contradiction that p' cannot occur in S. Let j be the smallest integer such that r_{i_j} cannot occur in $S \oplus r_{i_1} \oplus \cdots \oplus r_{i_{j-1}}$. Since p' is a formal pathway, the state $S \oplus r_{i_1} \oplus \cdots \oplus r_{i_{j-1}}$ must contain all the intermediate reactants of r_{i_j}. However, because of the second condition in the definition of a futile pathway, Formal$(S \oplus r_{i_1} \oplus \cdots \oplus r_{i_{j-1}}) \supseteq$ Formal$(S \oplus r_1 \oplus r_2 \oplus \cdots \oplus r_{i_j-1})$. This implies that r_{i_j} cannot occur in $S \oplus r_1 \oplus r_2 \oplus \cdots \oplus r_{i_j-1}$, which is a contradiction. □

B Algorithm

B.1 Signature of a pathway

While Corollary 4.3 gives us a way to make use of the bounded width assumption, it is still unclear whether the enumeration can be made finite, because the number of undecomposable semiformal pathways of bounded width may still be infinite. To resolve this problem, we need to define a few more concepts.

Definition 22. Let p be a semiformal pathway. The **decomposed final states (DFS)** of p is defined as the set of all unordered pairs (T_1, T_2) that can be obtained by decomposing p into two semiformal pathways and taking their final states. Note that for an undecomposable pathway, the DFS is the empty set.

Definition 23. Let $p = (r_1, \ldots, r_k)$ be a semiformal pathway. Also, let $S_i = S \oplus r_1 \oplus \cdots \oplus r_i$ where S is the initial state of p. The **formal closure** of p is defined as the unique minimal state S' such that $S_i \subseteq S'$ for all i.

The **regular final states (RFS)** of p is defined as the set of all minimal states T such that there exists a turning point reaction $r_j = (R, P)$ which satisfies Formal$(S_i) \subseteq S$ for all $i < j$, Formal$(S_i) \subseteq T$ for all $i \geq j$, and Formal$(S_{j-1} - R') = \emptyset$.

Definition 24. Let p be a semiformal pathway. Then, the **futile loop creators (FLC)** of p is defined to be the set of all reactions $r = (R, P)$ (which may or may not be contained in the given CRN) such that $R \cap P = \emptyset$ and $p + (r)$ is a semiformal pathway that can be decomposed into a nonempty semiformal pathway and a futile pathway which contains r.

We also introduce an Boolean indicator variable which we simply call **futility**, which is 1 if and only if p can be decomposed into a nonempty semiformal pathway and a futile pathway.

Definition 25. The **signature** of the pathway is defined to be the 8-tuple of the initial state, final state, width, formal closure, DFS, RFS, FLC, and futility.

Theorem B.1. *If m is any number, the set of signatures of all semiformal pathways of width at most m is finite.*

Proof. Clearly, there is only a finite number of possible initial states, final states, widths, formal closures, and RFS. Also, since there is only a finite number of possible final states, there is only a finite number of possibilities for DFS. Finally, it follows immediately from the definition of a futile pathway that any reaction (R, P) in FLC of p must satisfy that $|R|$ and $|P|$ are at most the width of p. Therefore, there is also a finite number of possibilities for FLC. □

Theorem B.2. *Suppose p_1 and p_2 are two pathways with the same signature. Then, for any reaction r, $p_1 + (r)$ and $p_2 + (r)$ also have the same signature.*

Proof. Let $p_1' = p_1 + (r)$ and $p_2' = p_2 + (r)$. It is trivial that p_1' and p_2' have the same initial and final states, formal closure, and width.

Now we consider futility. If p_1 has futility of 1, then automatically all of p_1', p_2, and p_2' have futility of 1, so we are done. If p_1 has futility of 0, then p_1' has futility of 1 if and only if r is contained in the FLC of p_1. Since p_2 and p_1 have the same FLC, p_2' must have the same futility value as p_1'.

Next, we show that p_1' and p_2' have the same DFS. Suppose (T_1, T_2) is in the DFS of p_1'. That is, there exists a decomposition (q_1', q_2') of p_1' where q_1' and q_2' have final states T_1 and T_2. The last reaction r is either contained in q_1' or q_2'. Without loss of generality, suppose the latter is the case. Then, if $q_1 = q_1'$ and $q_2 + (r) = q_2$, then (q_1, q_2) should decompose p_1, which is a prefix of p_1'. Since p_1 and p_2 have the same DFS, there should be a decomposition (s_1, s_2) of p_2 that has the same final states as q_1 and q_2. Clearly, $(s_1, s_2 + (r))$ should be a decomposition of p_2' and thus (T_1, T_2) is also in the DFS of p_2'. By symmetry, it follows that p_1' and p_2' have the same DFS.

Now we argue that p_1' and p_2' should have the same RFS. Suppose T is contained in the RFS of p_1'. It suffices to show that some subset of it is contained in the RFS of p_2'.

1. If the turning point for T in p_1' is the last reaction r, it must be the case that the formal closure of p_1 is a subset of the initial state of p_1'. Since p_2 has the same formal closure and final state as p_1, it immediately follows that some subset of T must be contained in the RFS of p_2'.
2. Otherwise, it must be the case that some subset T' of T is contained in the RFS of p_1 and that the final state of p_1' is a subset of T. Since T' is also contained in the RFS of p_2 and p_2' has the same final state as p_1', it follows that RFS of p_2' must contain some subset of T.

Finally, we prove that p_1' and p_2' have the same FLC. Suppose a reaction s is contained in the FLC of p_1'. We show that it must also be contained in the FLC of p_2'. By definition, there exists some decomposition of p_1' into q and q' such that q is a semiformal pathway and q' is a futile pathway which ends with s.

1. If q' also contains r, we see that q is indeed a subsequence of p_1. This means that the FLC of p_1 contains the reaction (X, Y) where X and Y are the initial and final states of the pathway (r, s). Since p_2 has the same FLC as p_1, it follows that $p_2 + ((X, Y))$ can be decomposed into a semiformal pathway

and a futile pathway x containing (X, Y). Now we observe that s cannot consume any formal species. This is because s is the last reaction of at least one futile pathway and the definition of a futile pathway precludes it. Therefore replacing (X, Y) with two reactions r and s will not make the futile pathway x non-futile. We conclude that s must be contained in the FLC of p_2'.

2. If q' does not contain r, it immediately follows that s must be contained in the FLC of p_1. That is, there is a decomposition of $p_1 + (s)$ into a semiformal pathway a and a futile pathway b. Therefore it is also contained in the FLC of p_2, which means that there is a decomposition of $p_2 + (s)$ into a semiformal pathway y and a futile pathway $x = x' + (s)$. Suppose $y + (r)$ is not semiformal. This implies that the final state of y does not contain all the intermediate reactants of r. However, since x is futile, the net effect of x' must exactly cancel that of s. Therefore the final state of y must contain exactly the same set of intermediate species as the final state of a. Since $a + (r)$ is semiformal, this is a contradiction. Therefore, $y + (r)$ is semiformal, which implies that $p_2' = p_2 + (r, s)$ contains a futile loop that ends with s. We conclude that s is contained in the FLC of p_2'.

□

Theorem B.3. *A pathway p is a prime formal pathway if and only if its signature satisfies all the following conditions.*

1. *The initial and final states are formal.*
2. *The DFS is the empty set.*
3. *The futility is* 0.

B.2 Algorithm for enumerating signatures

It is now clear that we can find the formal basis by enumerating the signatures of all undecomposable semiformal pathways. In this section we present a simple algorithm for achieving this.

```
function enumerate(p, w, ret)
    if p is not semiformal or has width greater than w then return ret
    sig = signature of p
    if sig is in ret then return ret
    add sig to ret
    for every reaction rxn
        ret = enumerate(p + [rxn], w, ret)
    end for
    return ret
end function

function main()
    w_max = 0
    b = branching factor of the given CRN
    while true
        signatures = enumerate([], w_max, {})
        w = maximum width of an undecomposable pathway in signatures
        if (w+1)*b <= w_max then break
        w_max = (w+1)*b
    end while
    return signatures
end function
```

The subroutine `enumerate` is a function that enumerates the signatures of all semiformal pathways of width at most `w`. Note that it uses memoization to avoid duplicating pathways that have identical signatures, as justified by Theorem B.2. Because of this memoization, Theorem B.1 ensures that this subroutine will terminate in finite time.

The subroutine `main` repeatedly calls `enumerate`, increasing the width bound according to Corollary 4.3. It is obvious that `main` will terminate in finite time if and only if there exists a bound to the width of an undecomposable semiformal pathway.

It is out of scope of this paper to attempt theoretical performance analysis of this algorithm or to study the computational complexity of finding the formal basis. While it is obvious that there are further optimizations one can make to improve the performance of the algorithm in practice, we meet our goal in this paper in demonstrating the existence of a finite time algorithm and leave further explorations as a future task.

B.3 Testing tidiness and regularity

Finally, we discuss how to use the enumerated signatures to test tidiness and regularity of the given CRN.

Definition 26. A semiformal pathway p is said to have property A if there does not exist a pathway q such that

1. q consists of reactions that do not consume a formal species, and
2. $p + q$ contains a futile loop that uses at least one reaction from p.

Theorem B.4. *If semiformal pathways p and p' have the same signature, either they both satisfy property A or they both do not satisfy property A.*

Proof. If the futility of p and p' is 1, it is trivial that they both do not satisfy property A.

Suppose the futility of the two pathways is 0. Without loss of generality, assume towards a contradiction that p satisfies property A while p' does not. Then there exists q' such that q' does not consume a formal species and $p' + q'$ contains a futile loop that uses at least one reaction from p'. Without loss of generality, we can assume that the whole of q' is used in this futile loop (otherwise simply remove the reactions that are not used). Let s denote the reactions in p' that are used in the futile loop. Since the futility of p' was 0, s alone cannot be a futile loop. That is, it must violate at least one of the two conditions of the definition of a futile loop. If the first condition is violated, it immediately implies that q' cannot be a futile loop, because we know that the initial and final states of $s + q'$ are equal. If the first condition is satisfied but the second condition is violated, since q' does not consume any formal species, this implies that q' must not produce any formal species either. But then $s + q'$ must also violate the second condition, which is a contradiction. Therefore q' is not a futile loop.

Sicne $p' + q'$ contains a futile loop, we know that (S, T) is contained in the FLC of p' where S and T are the initial and final states of q' respectively. However, since p has the same FLC as p', $p + q'$ must also contain a futile loop. Moreover, since we established that q' is not a futile loop, the futile loop in $p + q'$ uses at least one reaction from p. □

Theorem B.5. *A CRN is tidy if and only if every undecomposable semiformal pathway with property A has a closing pathway.*

Proof. The forward direction is trivial. For the reverse direction, we show that if a CRN is not tidy, there exists an undecomposable semiformal pathway with property A that does not have a closing pathway.

First, we show that there must exist a semiformal pathway p with property A that does not have a closing pathway. By definition, we have a semiformal pathway p' that does not have a closing pathway. If p' has property A, we are done. If p' does not have property A, then there exists q' such that q' does not consume

a formal species and $p' + q'$ contains a futile loop that uses at least one reaction from p'. Without loss of generality, we can assume that the futile loop uses all reactions of q' (otherwise simply remove the reactions that are not used in the futile loop). Remove from p' all the reactions that are used in this futile loop and call the resulting pathway p''. Now observe the following facts.

1. If s'' is a closing pathway for p'', then $s' = q' + s''$ is a closing pathway for p'. Therefore, p'' does not have a closing pathway.
2. The length of p'' is strictly smaller than that of p'.
3. Any semiformal pathway of length 1 has property A.

These three facts imply that if we repeat this procedure, we must eventually obtain a semiformal pathway p with property A that does not have a closing pathway.

Now, if p is undecomposable, then we are done. Suppose p is decomposable. Since property A ensures that p cannot have a futile loop, p decomposes into two semiformal pathways p_1 and p_2. Suppose both of these pathways have closing pathways. Then, since the final state of p has the same intermediate species as the sum of the final states of p_1 and p_2 (by Theorem 4.1 and the fact that p_1 and p_2 are semiformal), the two closing pathways concatenated will be a closing pathway of p (because a closing pathway does not consume any formal species). This is a contradiction. Thus, we conclude that at least one of p_1 or p_2 does not have a closing pathway. Moreover, we show that p_1 and p_2 must satisfy property A. Without loss of generality we show this for p_1. Assume towards a contradiction that p_1 does not have property A. Then, there exists q such that q does not consume a formal species and $p_1 + q$ contains a futile loop that uses at least one reaction from p_1. Now note that this same futile loop also exists in $p + q$. Moreover, the pathway that results from the removal this futile loop must be semiformal, because it is obtained by interleaving two semiformal pathways (that is, $p_1 + q$ minus the futile loop and p_2). This is a contradiction to the assumption that p had property A.

Thus, we conclude that either p is an undecomposable semiformal pathway with property A or we can find a shorter pathway with property A that does not have a closing pathway. Since a pathway of length 0 has a closing pathway, repeating this argument will eventually yield an undecomposable semiformal pathway with property A that does not have a closing pathway. □

Theorem B.6. *If p is an undecomposable semiformal pathway with property A and q is any pathway that never consumes a formal species, $p + q$ is undecomposable.*

Proof. Suppose $p + q$ is decomposable. Since p has property A, $p + q$ cannot contain a futile loop. this means that $p + q$ decomposes into two nonempty semiformal pathways p_1 and p_2. Then it must be the case that one of p_1 and p_2 contains all the reactions of p, because otherwise p must be decomposable also. Without loss of generality, suppose p_1 contains all the reactions of p. Then p_2 consists only of reactions from q. However, q does not consume any formal species, and therefore p_2 must be empty in order to be semiformal. Contradiction. □

Theorem B.7. *If p is a semiformal pathway that has a closing pathway, it also has a closing pathway that does not contain a futile loop.*

To test tidiness, we attempt to find a closing pathway for every undecomposable semiformal pathway p enumerated by the main algorithm. Theorems B.4 and B.5 ensure that it is safe to consider only these pathways despite the memoization employed by the main algorithm. We achieve this by enumerating the signatures of all undecomposable semiformal pathways of the form $p + q$ where q is a pathway that does not consume a formal species. If we do this using the same techniques as the main algorithm, Theorems B.6 and B.7 guarantee that we will end up in one of the following cases:

1. Discover a closing pathway for p.

2. Discover that p does not satisfy property A.
3. Certify that p is an undecomposable semiformal pathway with property A that does not have a closing pathway.

The testing of regularity is trivial, using the following theorem.

Theorem B.8. *A prime formal pathway is regular if and only if its RFS contains its final state.*

We emphasize that these methods work only because of the bounded width assumption we made on undecomposable semiformal pathways. Without this assumption, it is unclear whether these problems are still decidable.