Abstract modelling of tethered DNA circuits

Matthew R. Lakin[1], Rasmus Petersen[2], Kathryn E. Gray[2,3], and Andrew Phillips[2]

[1] Department of Computer Science, University of New Mexico, Albuquerque, NM, USA
[2] Microsoft Research, Cambridge, UK
[3] Computer Laboratory, University of Cambridge, Cambridge, UK

mlakin@cs.unm.edu aphillip@microsoft.com

Abstract. Sequence-specific DNA interactions are a powerful means of programming nanoscale locomotion. These systems typically use a DNA track that is tethered to a surface, and molecular interactions enable a signal or cargo to traverse this track. Such low copy number systems are highly amenable to mechanized analyses such as probabilistic model checking, which requires a formal encoding. In this paper we present the first general encoding of tethered DNA species into a formal language, which allows the interactions between tethered species to be derived automatically using standard reaction rules. We apply this encoding to a previously published tethered DNA circuit architecture based on hairpin assembly reactions. This work enables automated analysis of large-scale tethered DNA circuits and, potentially, synthesis of optimized track layouts to implement specific logic functions.

1 Introduction

Nanoscale locomotion, driven by motor proteins such as kinesin and myosin, is a key component of many cellular processes [1]. Recent attempts to implement synthetic analogs of such systems typically rely on DNA components that are physically tethered to a surface, e.g., a DNA origami tile, to form a track. The intuition here is that tethered components can only interact if they are tethered in close proximity to one another, so that when a component is attached to a particular tethered track location it can only move to nearby available track locations. This approach has been used to implement a range of DNA walkers [2, 3], molecular-scale assembly lines [4], and localized DNA logic circuits [5, 6].

Since these systems typically involve small numbers of molecules, they are highly suited to formal analysis using methods such as probabilistic model checking [7]. Probabilistic model checking has previously been applied to solution-phase DNA strand displacement circuits [8] but its practical utility is limited by the state space explosion caused by large species populations. Previous work on model checking of DNA walkers [9] used a manually constructed representation of the state space, which is not scalable to larger track sizes with more reachable states. Therefore, it is important to define a general mechanism for deriving the interactions of tethered DNA species in large-scale systems. In this paper we present such a mechanism, by extending the DSD language [10, 11] with new syntactic constucts and reaction rules for encoding of tethered DNA strand displacement systems on tiles. This encoding could be used to formalize, simulate and analyze large-scale tethered DNA circuits.

Domain lists without tethers	$\mathtt{S} ::= D_1 \cdots D_n$
Left domain lists	$\mathtt{L} ::= \mathtt{tether}(\mathtt{a}_1,\ldots,\mathtt{a}_n)\ \mathtt{S} \mid \mathtt{S}$
Right domain lists	$\mathtt{R} ::= \mathtt{S}\ \mathtt{tether}(\mathtt{a}_1,\ldots,\mathtt{a}_n) \mid \mathtt{S}$
Strands	$\mathtt{A} ::= \langle \mathtt{L} \rangle \mid \langle \mathtt{R} \rangle \mid \{\mathtt{L}\} \mid \{\mathtt{R}\}$
Segments (no hairpins)	$\mathtt{M}_{NH} ::= \{\mathtt{L}'\}\langle \mathtt{L} \rangle [\mathtt{S}] \langle \mathtt{R} \rangle \{\mathtt{R}'\}$
Segments (left hairpin)	$\mathtt{M}_{LH} ::= \langle \mathtt{S}'\}[\mathtt{S}]\langle \mathtt{R} \rangle \{\mathtt{R}'\}$
Segments (right hairpin)	$\mathtt{M}_{RH} ::= \{\mathtt{L}'\}\langle \mathtt{L} \rangle [\mathtt{S}] \{\mathtt{S}'\rangle$
Segment join operators	$\sim ::= \mathtt{:} \mid \mathtt{::}$
Gates (no hairpins)	$\mathtt{G}_{NH} ::= \mathtt{M}_{NH} \mid \mathtt{M}_{NH} \sim \mathtt{G}_{NH}$
Gates (left hairpin)	$\mathtt{G}_{LH} ::= \mathtt{M}_{LH} \mid \mathtt{M}_{LH} \sim \mathtt{G}_{NH}$
Gates (right hairpin)	$\mathtt{G}_{RH} ::= \mathtt{M}_{RH} \mid \mathtt{G}_{NH} \sim \mathtt{M}_{RH}$
Gates (two hairpins)	$\mathtt{G}_{TH} ::= \mathtt{M}_{LH} \sim \mathtt{G}_{RH}$
Gates	$\mathtt{G} ::= \mathtt{G}_{NH} \mid \mathtt{G}_{LH} \mid \mathtt{G}_{RH} \mid \mathtt{G}_{TH}$
Species	$\mathtt{X} ::= \mathtt{A} \mid \mathtt{G}$
Tethered species	$\mathtt{XT} ::= \mathtt{X} \quad (\text{if } \mathit{tethered}(\mathtt{X}))$
Untethered species	$\mathtt{XU} ::= \mathtt{X} \quad (\text{if } \mathit{untethered}(\mathtt{X}))$
Tethered systems	$\mathtt{T} ::= \mathtt{XT} \mid (\mathtt{T}_1 \parallel \mathtt{T}_2)$
Mixed systems	$\mathtt{I} ::= \mathtt{X} \mid (\mathtt{I}_1 \parallel \mathtt{I}_2)$
Systems	$\mathtt{U} ::= \mathtt{XU} \mid [[\mathtt{T}]] \mid (\mathtt{U}_1 \parallel \mathtt{U}_2)$

Fig. 1: Extended syntax for DSD with tethered species, hairpins, and DNA tiles. The predicate $\mathit{tethered}(\mathtt{X})$ is satisfied if $\mathtt{X}$ contains at least one tether, and the predicate $\mathit{untethered}(\mathtt{X})$ is satisfied if $\mathtt{X}$ contains no tethers.

2 Abstract specifications of tether locations

In an initial design of a large-scale tethered DNA circuit, the designer will not necessarily have precise locations in mind for each of the individual tethered species. Hence, in this early design phase it may be simpler to represent the relationship between tethered species abstractly, by simply specifying which tethered species are located sufficiently close to react with which other tethered species.

Here we adopt this abstract approach to the specification of the locations of tethered species. We represent physical proximity using *location tags*, which are chosen from an alphabet $\mathbb{A} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots\}$. Each tether in a species will be annotated with a finite number of tags, and we assume that all of the species which share a particular tag are tethered such that they are close enough together to react with each other (but not with any species that do not share that tag). Hence, two tethered species can interact if they have at least one tag in common. Below, we will associate each tag with a local concentration, so that certain tethered species may interact at a faster rate than others.

3 DSD syntax for tethered species

To model tethered species in the DSD language, we introduce the reserved keyword $\mathtt{tether}(\mathtt{a}_1,\ldots,\mathtt{a}_n)$ to represent a tether point that attaches the species to a surface, where $\mathtt{a}_1,\ldots,\mathtt{a}_n$ is a finite, non-empty list of location tags. This keyword is somewhat

like a domain, except that it cannot be complemented and its occurrences in structures are syntactically restricted. A species with no tethers is free to diffuse in solution.

Figure 1 defines a grammar for tethered DSD systems. (For brevity, we omit module definitions and local domain declarations, which are present in the standard DSD syntax.) Here and henceforth, *D* ranges over domains *excluding* tethers. This enables us to syntactically limit the occurrences of tethers in the segment syntax. Strands are divided into "upper" strands ($\langle \mathtt{L} \rangle$ or $\langle \mathtt{R} \rangle$, which are rendered 5' to 3' from left to right), and "lower" strands ($\{\mathtt{L}\}$ or $\{\mathtt{R}\}$, which are rendered 3' to 5' from left to right). Note that this grammar limits single strands to at most one tether point.

Multi-strand complexes are known as "gates" in the DSD language, and are composed of one or more concatenated "segments", which have a double-stranded portion, possibly with single-stranded overhangs. From our previous work [10, 11], the general form of a segment is $\{\mathtt{L}'\}\langle \mathtt{L} \rangle [\mathtt{S}] \langle \mathtt{R} \rangle \{\mathtt{R}'\}$. Here, S is the double-stranded portion and the other domain sequences denote the upper and lower single-stranded overhangs, which are distinguished based on the brackets as in the case of single strands. Multiple segments may be composed using the segment join operator (:) for concatenation of the lower strand, or (::) for concatenation of the upper strand.

Here, we extend the DSD syntax with hairpin loops, which can occur at either end of a gate. This is an important extension because metastable hairpins are widely used as a fuel supply in the design of DNA nanomachines [2, 12–14], but they are not representable in the previously published DSD syntax [11]. We write $\langle \mathtt{S} \}$ for a hairpin loop at the left-hand end of a gate and $\{\mathtt{S} \rangle$ for a hairpin loop at the right-hand end of a gate. Within a hairpin loop, we list domains from 5' to 3', that is, clockwise (since, by convention, the upper strand runs 5' to 3' from left to right). The grammar includes multiple syntactic categories for gates, to ensure that hairpins can only appear at the *ends* of gate structures. Note that we omit empty overhangs when writing down gate structures and, as standard in DSD, we assume that the only single-stranded complementary domains are toeholds.

We let the metavariables XT and XU range over species (i.e., strands or gates) with and without tethers, respectively. We then define *tethered systems* T that consist entirely of tethered species, and *systems* U that consist of untethered species and *tiles* $[[\mathtt{T}]]$, which represent a DNA tile with the tethered species T attached to it. Hence, a system corresponds to a DSD program, in which tethered species can only occur within a tile construct. This provides a syntactic means of delimiting the occurrences of tethers in a program, and allows us to encode and simulate solutions containing many tethered circuits on many tiles. (We also define *mixed systems* I that may contain both tethered and untethered species—these are not considered well-formed and only appear during intermediate computations of reactions between tiles and untethered species.)

For simplicity, the grammar in Figure 1 admits gate structures with tethers in unrealistic locations at the joins between gate segments. Instead, we assume that such gates are disallowed by a subsequent well-formedness check on grammatical structures which requires that, if two neighbouring gate segments are joined along a particular strand, the domains adajcent to the join operator cannot be tethers. To simplify the semantics, we assume that hairpins are sufficiently short that nothing will bind to a domain in one of these structures. These extensions to the DSD syntax will enable us to model teth-

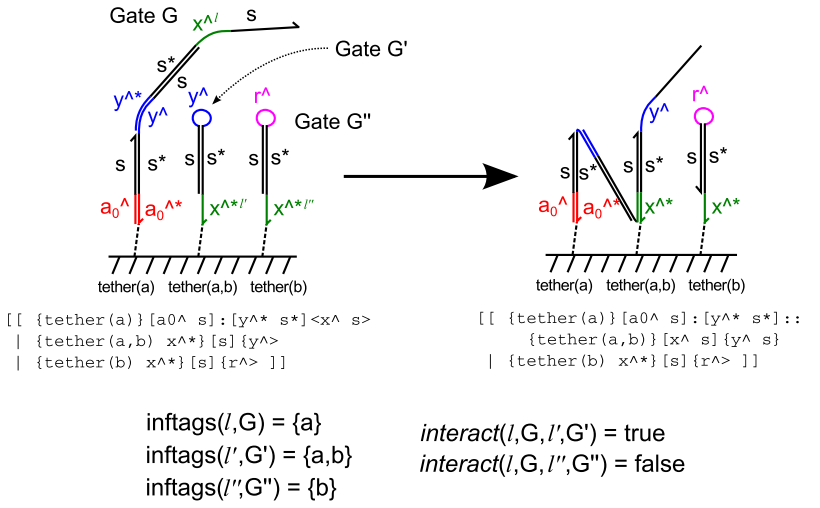


Fig. 2: Example of computing the sets of tags that influence several exposed toeholds in tethered gates $\mathtt{G}$, $\mathtt{G}'$ and $\mathtt{G}''$, together with evaluations of the *interact* predicate to determine whether pairs of gates may interact. The gate structures are derived from the transmission line design from [6], and corresponding DSD code is presented. In this example, the domains labelled ℓ and ℓ'' have no tags in common and are therefore deemed too far apart to interact, whereas the domains labelled ℓ' and ℓ'' are both influenced by the location tag $\mathtt{a}$ and can therefore interact. Thus the design enforces that only neighbouring structures in the transmission line can interact.

ered DNA circuits using hairpin fuels, as shown below. Appendix A presents additional extensions to the DSD syntax and semantics to encode internal loops and bulges (the Appendices are available for download from the corresponding authors' websites).

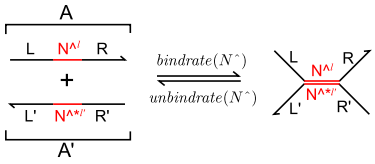
4 Computing interactions between tethered species

In order to derive bimolecular interactions involving tethered species, we must calculate whether the domains involved are close enough to interact. Thus we must determine which tether points are exerting influence over which domains, in order to determine whether those domains are tethered close enough to interact.

4.1 Labels

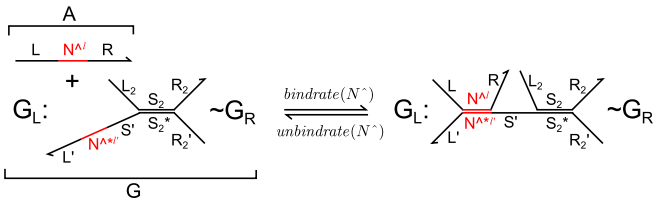
To identify the particular domains involved in an interaction, we fix a countably infinite set Λ of *labels* $\ell_1, \ell_2, \ldots$ and assume that every occurrence of every domain is associated with a *globally unique* label. For example, the gate $\{\mathtt{T\hat{}^*}\}[\mathtt{X}]\langle\mathtt{X}\rangle$ might be labelled as $\{\mathtt{T\hat{}^*}^{\ell_1}\}[\mathtt{X}^{\ell_2}]\langle\mathtt{X}^{\ell_3}\rangle$. Domain labels are not part of the user-visible language syntax, rather,

Two strands binding / unbinding:



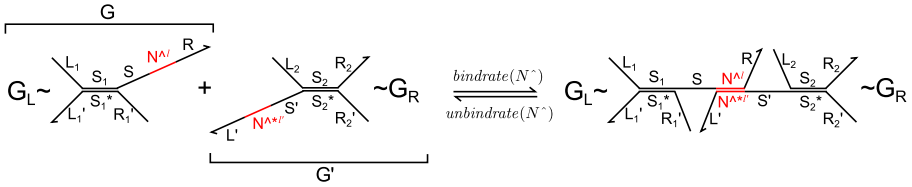
... where forward reaction is only derivable if $\mathit{interact}(\ell, \mathtt{A}, \ell', \mathtt{A}')$.

A strand binding to / unbinding from a gate:



... where forward reaction is only derivable if $\mathit{interact}(\ell, \mathtt{A}, \ell', \mathtt{G})$.

Two gates binding / unbinding:



... where forward reaction is only derivable if $\mathit{interact}(\ell, \mathtt{G}, \ell', \mathtt{G}')$.

Fig. 3: Bimolecular DSD reaction rules for tethered species.

they are an internal mechanism to distinguish between multiple instances of the same domain when calculating which tether is exerting influence over that domain. Hence, the particular assignment of labels to domain occurrences is not important, provided that they are globablly unique. We do not state labels explicitly unless they are required in a reaction rule.

4.2 Computing bimolecular interactions with tethered species

The key operation in the tethered semantics is to compute the set of tethers that are currently exerting influence on a particular domain labelled with ℓ. We write $\mathit{inftags}(\ell, \mathtt{G})$ for the set of location tags that influence the domain labelled by ℓ in the gate $\mathtt{G}$. To do this we traverse the structure of the species, starting from the labelled domain in question and moving outwards, and assume that the first tethers that we find in either direction (written as $\mathit{LHtags}(\ell, \mathtt{G})$ and $\mathit{RHtags}(\ell, \mathtt{G})$) are those exerting influence on the

position of the labelled domain:

$$\mathit{inftags}(\ell, \mathtt{G}) \triangleq \bigcup(\mathit{LHtags}(\ell, \mathtt{G}) \cup \mathit{RHtags}(\ell, \mathtt{G}))$$

In this definition, the inner union is over sets of tag lists, while the outer union combines the resulting set of tag lists into a single set of location tags. The $\mathit{inftags}(\ell, \mathtt{G})$ function, and the case for tethered strands, can be fully defined as follows.

Given a segment $\mathtt{M}$, we write $\mathit{LHtags}(\mathtt{M})$ and $\mathit{RHtags}(\mathtt{M})$ for the sets of tag lists $\mathtt{a}_1, \ldots, \mathtt{a}_n$ such that $\mathtt{tether}(\mathtt{a}_1, \ldots, \mathtt{a}_n)$ appears on the left or right overhang of the segment $\mathtt{M}$, respectively. Furthermore, we write $\mathit{tags}(\mathtt{M})$ for the set of *all* tag lists $\mathtt{a}_1, \ldots, \mathtt{a}_n$ such that $\mathtt{tether}(\mathtt{a}_1, \ldots, \mathtt{a}_n)$ appears *anywhere* in $\mathtt{M}$.

We now define functions $\mathit{LHtags}(\mathtt{G})$ and $\mathit{RHtags}(\mathtt{G})$, which return the leftmost and rightmost tag sets found by searching a gate $\mathtt{G}$ segment-wise, respectively. These functions can be defined by recursion on the structure of gates, as follows.

$$\mathit{LHtags}(\mathtt{M}) \triangleq \mathit{tags}(\mathtt{M}) \qquad \mathit{LHtags}(\mathtt{M} \sim \mathtt{G}) \triangleq \begin{cases} \mathit{tags}(\mathtt{M}) & \text{if } \mathit{tags}(\mathtt{M}) \neq \varnothing \\ \mathit{LHtags}(\mathtt{G}) & \text{otherwise.} \end{cases}$$

$$\mathit{RHtags}(\mathtt{M}) \triangleq \mathit{tags}(\mathtt{M}) \qquad \mathit{RHtags}(\mathtt{G} \sim \mathtt{M}) \triangleq \begin{cases} \mathit{tags}(\mathtt{M}) & \text{if } \mathit{tags}(\mathtt{M}) \neq \varnothing \\ \mathit{RHtags}(\mathtt{G}) & \text{otherwise.} \end{cases}$$

We now define the first tag sets found by searching outwards from a particular labelled domain in a gate structure. Suppose that the domain in question has label ℓ, and that the gate $\mathtt{G}$ has the form $\mathtt{G}_L \sim \mathtt{M} \sim \mathtt{G}_R$, where $\mathtt{M}$ is the segment containing the domain with label ℓ. Then, we define functions *LHtags* and *RHtags* that compute the first tag sets found in a segment-wise search outward from the segment in $\mathtt{G}$ containing ℓ, as follows.

$$\mathit{LHtags}(\ell, \mathtt{G}_L \sim \mathtt{M} \sim \mathtt{G}_R) \triangleq \begin{cases} \mathit{LHtags}(\mathtt{M}) & \text{if } \mathit{LHtags}(\mathtt{M}) \neq \varnothing \\ \mathit{RHtags}(\mathtt{G}_L) & \text{otherwise.} \end{cases}$$

$$\mathit{RHtags}(\ell, \mathtt{G}_L \sim \mathtt{M} \sim \mathtt{G}_R) \triangleq \begin{cases} \mathit{RHtags}(\mathtt{M}) & \text{if } \mathit{RHtags}(\mathtt{M}) \neq \varnothing \\ \mathit{LHtags}(\mathtt{G}_R) & \text{otherwise.} \end{cases}$$

In the case where $\mathtt{G}$ has the form $\mathtt{M} \sim \mathtt{G}_R$, where $\mathtt{M}$ is the segment containing the domain with label ℓ, the definitions are as follows.

$$\mathit{LHtags}(\ell, \mathtt{M} \sim \mathtt{G}_R) \triangleq \mathit{LHtags}(\mathtt{M})$$

$$\mathit{RHtags}(\ell, \mathtt{M} \sim \mathtt{G}_R) \triangleq \begin{cases} \mathit{RHtags}(\mathtt{M}) & \text{if } \mathit{RHtags}(\mathtt{M}) \neq \varnothing \\ \mathit{LHtags}(\mathtt{G}_R) & \text{otherwise.} \end{cases}$$

Finally, in the case where $\mathtt{G}$ has the form $\mathtt{G}_L \sim \mathtt{M}$, where $\mathtt{M}$ is the segment containing the domain with label ℓ, the definitions are as follows.

$$\mathit{LHtags}(\ell, \mathtt{G}_L \sim \mathtt{M}) \triangleq \begin{cases} \mathit{LHtags}(\mathtt{M}) & \text{if } \mathit{LHtags}(\mathtt{M}) \neq \varnothing \\ \mathit{RHtags}(\mathtt{G}_L) & \text{otherwise.} \end{cases}$$

$$\mathit{RHtags}(\ell, \mathtt{G}_L \sim \mathtt{M}) \triangleq \mathit{RHtags}(\mathtt{M}).$$

These functions are used to define $\mathit{inftags}(\ell, \mathtt{G})$, as shown above. Furthermore, since single strands may also be tethered, we must also define a similar function for strands: assuming that the label ℓ appears in the strand $\mathtt{A}$, we simply let $\mathit{inftags}(\ell, \mathtt{A})$ return the union of all tag lists $\mathtt{a}_1, \ldots, \mathtt{a}_n$ such that $\mathtt{tether}(\mathtt{a}_1, \ldots, \mathtt{a}_n)$ appears in $\mathtt{A}$. Our well-formedness conditions on the occurrences of tethers mean that $\mathit{inftags}(\ell, \mathtt{A})$ must contain at tags from at most one tag set, as the syntax only allows a tether at one end of a single strand.

The $\mathit{inftags}(\ell, \mathtt{X})$ function, where $\mathtt{X}$ could be a gate $\mathtt{G}$ or a strand $\mathtt{A}$, will be used below to define interaction rules for tethered species. Figure 2 presents the result of computing the sets of tags that influence exposed toeholds in an example interaction between a tethered gate and a tethered strand.

We can now define the additional tests, expressed in terms of the *inftags* function, to govern bimolecular interactions involving species that may be tethered. If species $\mathtt{X}_1$ and $\mathtt{X}_2$ may interact via toeholds with labels ℓ_1 and ℓ_2, the interaction is possible if the predicate $\mathit{interact}(\mathtt{X}_1, \ell_1, \mathtt{X}_2, \ell_2)$ is satisfied, which is defined as follows.

$$\begin{aligned}\mathit{interact}(\mathtt{X}_1, \ell_1, \mathtt{X}_2, \ell_2) \triangleq\ & (\mathit{inftags}(\mathtt{X}_1, \ell_1) \cap \mathit{inftags}(\mathtt{X}_2, \ell_2)) \neq \varnothing \\ & \vee\, \mathit{inftags}(\mathtt{X}_1, \ell_1) = \varnothing \vee \mathit{inftags}(\mathtt{X}_2, \ell_2) = \varnothing\end{aligned}$$

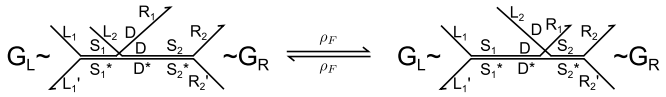
The first clause of the definition covers the case when the reactants $\mathtt{X}_1$ and $\mathtt{X}_2$ are both tethered, and when the interacting toehold domains have one or more location tags in common. This means that the species are tethered close enough together to interact. The two remaining clauses cover the cases when one or both reactants contain no tethers, and are therefore freely diffusing. In these cases, the reaction is always possible because a freely diffusing species can always find any other species to interact with. This definition will be used below to formalize the reaction rules for tethered species.

5 Reaction rules for tethered species

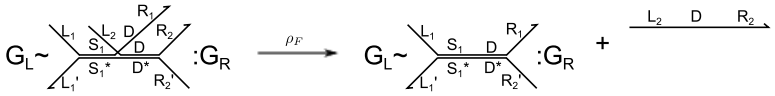
We write $\mathtt{G}_{\mathfrak{L}}$ for any gate capable of serving as a left-hand context, that is, either $\mathtt{G}_{NH}$ (no hairpins) or $\mathtt{G}_{LH}$ (hairpin present only on the left-hand side), or an empty context. Similarly, we write $\mathtt{G}_{\mathfrak{R}}$ for any gate capable of serving as a right-hand context, that is, either $\mathtt{G}_{NH}$ (no hairpins) or $\mathtt{G}_{RH}$ (hairpin present only on the right-hand side), or an empty context. In this section we present rules that define the possible reactions between species, including permissible structural contexts. Each reaction rule is labelled with the reaction rate constant: we assume the existence of functions *bindrate* and *unbindrate* that map each toehold domain $N^{\wedge}$ (and its complement $N^{\wedge *}$) to the associated binding rate constant $\mathit{bindrate}(N^{\wedge})$ and unbinding rate constant $\mathit{unbindrate}(N^{\wedge})$ respectively, and rate constants ρ_F for "fast" unimolecular reactions (e.g., branch migration) and ρ_S for "slow" unimolecular reactions (e.g., formation of internal loops, which involves internal diffusion).

Figure 3 presents bimolecular binding rules for strands and gates, and the corresponding unimolecular unbinding rules. Since these species may be tethered, the bimolecular rules use the *interact* predicate defined in Section 4.2 as a crucial additional test, so that two tethered species may only bind if they are tethered close enough together. Figure 4 recaps the basic unimolecular reaction rules from the DSD semantics

Branch migration:

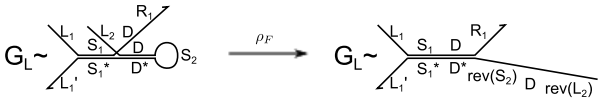


Strand displacement:



Gate displacement:

Hairpin displacement:



Hairpin binding / unbinding:

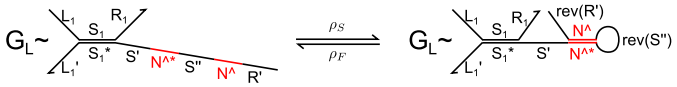


Fig. 4: Unimolecular DSD reaction rules, including additional rules to model hairpins. Note that the DSD convention is to list domains from left to right on the page, which corresponds to 5' to 3' for "upper" strands but 3' to 5' for "lower" strands. The inclusion of hairpins in the syntax muddies this distinction somewhat, and we must use the "rev" keyword to reverse the appropriate domain sequences in hairpin reactions.

and presents additional rules to define intramolecular hairpin opening (displacement) and (un)binding reactions. (ASCII representations of all rules, using the DSD syntax, are presented in Appendix B.) Note that the formation rules for hairpins is an instance of the *remote toehold* design concept [15]. To formalize the interactions of DNA tiles as tethering surfaces in the DSD language, we require the following rules, to turn the reactions of tethered species into reactions involving the corresponding tile species.

$$(\textsc{tInt})\ \frac{\mathtt{T} \xrightarrow{\rho} \mathtt{T}'}{[[\mathtt{T}]] \xrightarrow{\rho} [[\mathtt{T}']]} \qquad (\textsc{tBnd})\ \frac{\mathtt{XU} \parallel \mathtt{T} \xrightarrow{\rho} \mathtt{T}'}{\mathtt{XU} \parallel [[\mathtt{T}]] \xrightarrow{\rho} [[\mathtt{T}']]} \qquad (\textsc{tUBnd})\ \frac{\mathtt{T} \xrightarrow{\rho} \mathtt{T}' \parallel \mathtt{XU}}{[[\mathtt{T}]] \xrightarrow{\rho} [[\mathtt{T}']] \parallel \mathtt{XU}}$$

Rule (tINT) handles direct interactions between tethered species on the tile, since all reactants and products are tethered to the tile (see syntax definitions above). Rule (tBND) handles the case where an incoming diffusing species $\mathtt{XU}$ (which could be a strand or a gate) binds to a tile. Similarly, rule (tUBND) covers the case where a reaction on a tile produces an untethered species that is now free to diffuse. Note that the premisses of rules (tBND) and (tUBND) are instances of *mixed systems* of tethered and untethered species, but the final derived reactions in all cases involve well-formed *systems* in which all and only tethered species are encapsulated within tile constructs. These rules do not allow any crosstalk between two tiles—a species that is tethered to a tile can only interact with another species tethered to the same tile, or with a freely diffusing species. This is a reasonable assumption because of the slow diffusion rate of large DNA tiles compared to non-tile species, and means that two tile-based circuits can only communicate via a freely diffusing signal. Hence, all interactions taking place inside a tile are modeled as unimolecular reactions. Furthermore, the entire tile in a particular configuration must be used as the reactant species, to enable accurate modelling and simulation of populations of tiles. Finally, some additional contextual rules are required to complete the definition of the semantics: these are presented in Appendix C.

6 Calculating the propensities of tethered interactions

For simulations or probabilistic model checking of tethered circuits, we must compute the propensity of every possible interaction in the system, including tethered interactions. In mass action kinetics, the propensity, p, of a bimolecular reaction with reactants $\mathtt{X}_1$ and $\mathtt{X}_2$ and rate constant, k, is given by $p \triangleq k \times [\mathtt{X}_1] \times [\mathtt{X}_2]$, where $[\mathtt{X}_i]$ is the concentration of species $\mathtt{X}_i$. In tethered DSD systems, we use this expression for the propensity of bimolecular reactions in which both reactants are freely diffusing or precisely one reactant is tethered. In the latter case, we justify the use of this expression because the tiles to which the tethered species are attached are assumed to be well-mixed in the solution.

For bimolecular reactions involving two tethered reactants, however, this expression is not valid because tethered species do not satisfy the well-mixed assumption of mass action kinetics. To compute the propensities of bimolecular interactions between two tethered species, we use the concept of "local concentration" developed in previous work on the kinetics of biomolecular interactions between tethered species [5, 15]. This approach approximates the corresponding rates by computing the volume swept out by flexible tethered strands, to estimate the probability that the two species will be close enough to interact at a given point in time. For example, Genot *et al.* [15] computed local concentrations of $\sim 1 \times 10^5$ nM for localized strand displacement reactions.

To incorporate this theory into our tethered DSD framework, we assume the existence of a function lc that maps every location tag $\mathtt{a}$ to the local concentration for interactions occurring between species influenced by that tag. A higher value for the local concentration means that species sharing that tag are tethered relatively close to each other and will therefore interact at a faster rate. Then, for a bimolecular reaction between two tethered species $\mathtt{X}_1$ and $\mathtt{X}_2$ that interact via domains labelled ℓ_1 and ℓ_2 with rate constant k, we compute the reaction propensity, p, as $p \triangleq k \times \max(lc(\mathtt{a}_1), \ldots, lc(\mathtt{a}_n))$, where

$\mathtt{a}_1, \ldots, \mathtt{a}_n = \mathit{inftags}(\ell_1, \mathtt{X}_1) \cap \mathit{inftags}(\ell_2, \mathtt{X}_2)$. According to the rules from Figure 3, the bimolecular reaction can only occur if $\mathit{inftags}(\ell_1, \mathtt{X}_1) \cap \mathit{inftags}(\ell_2, \mathtt{X}_2)$ is non-empty. If there are multiple shared tags in this intersection, we use the largest of the corresponding local concentrations. We take this design decision because multiple shared tags do not enable additional mechanisms for a given reaction to occur—instead, they simply impose further constraints on how tethered species could be placed on a tile so that they will interact with the specified local concentrations. Hence the largest local concentration is the dominant one when computing the rate of a given interaction. Thus we are able to model the rates of bimolecular interactions between tethered species, enabling simulation and probabilistic model checking of solutions of tile-based circuits.

7 Examples

As an example application of our abstract modelling framework for tethered DNA circuits, we encoded the hairpin-based tethered circuit architecture from [6] into our extended DSD language. Figure 5 presents the DSD code and reduction sequence for the three-stator transmission line system from [6]. Note that the stators are all contained within a syntactic tile construct, and that all of the tags are assigned the same local concentration, i.e., the signal is passed between each pair of stators at the same rate. Furthermore, the distribution of location tags prevents the fuel bound to the first stator from binding directly to the third stator—hence, the signal must be passed sequentially along the stators with none being missed. Importantly, this causal dependence between the binding reactions can be deduced automatically by the DSD compiler, thanks to the use of location tags. Finally, a freely-diffusing strand displacement probe produces an increase in bulk fluorescence to indicate that the signal has reached the last stator. Figure 6 encodes a threshold-based spatial AND gate design from [6] by using different local concentration values for different location tags. The resulting reaction propensities mean that there is a high probability that the first input will bind to the threshold rather than the output stator. If this happens, the second input is required to trigger the output, achieving the desired AND logic. However, there is a non-zero probability that the first input will erroneously activate the output without the second input. We have implemented our syntax and semantics for tethered systems in the Visual DSD software tool [16], and Appendix D presents simulation and state space analysis results from encoding the examples from this section in the latest version of Visual DSD.

8 Discussion

To summarize, we have defined an encoding of tethered DNA circuits on tiles in the DSD language, which uses *location tags* to abstractly specify the pattern of tethering, and therefore the pattern of possible interactions between tethered species. We have extended the DSD syntax to include hairpins, which are often used as fuel for DNA nanomachines, and also to include DNA tiles, which colocalize tethered species in solution. We have demonstrated a formalization of the hairpin-based tethered circuit design from [6]. Our abstract representation strategy removes the need to explicitly formalize the layout of the track and the structure of the supporting surface, which could

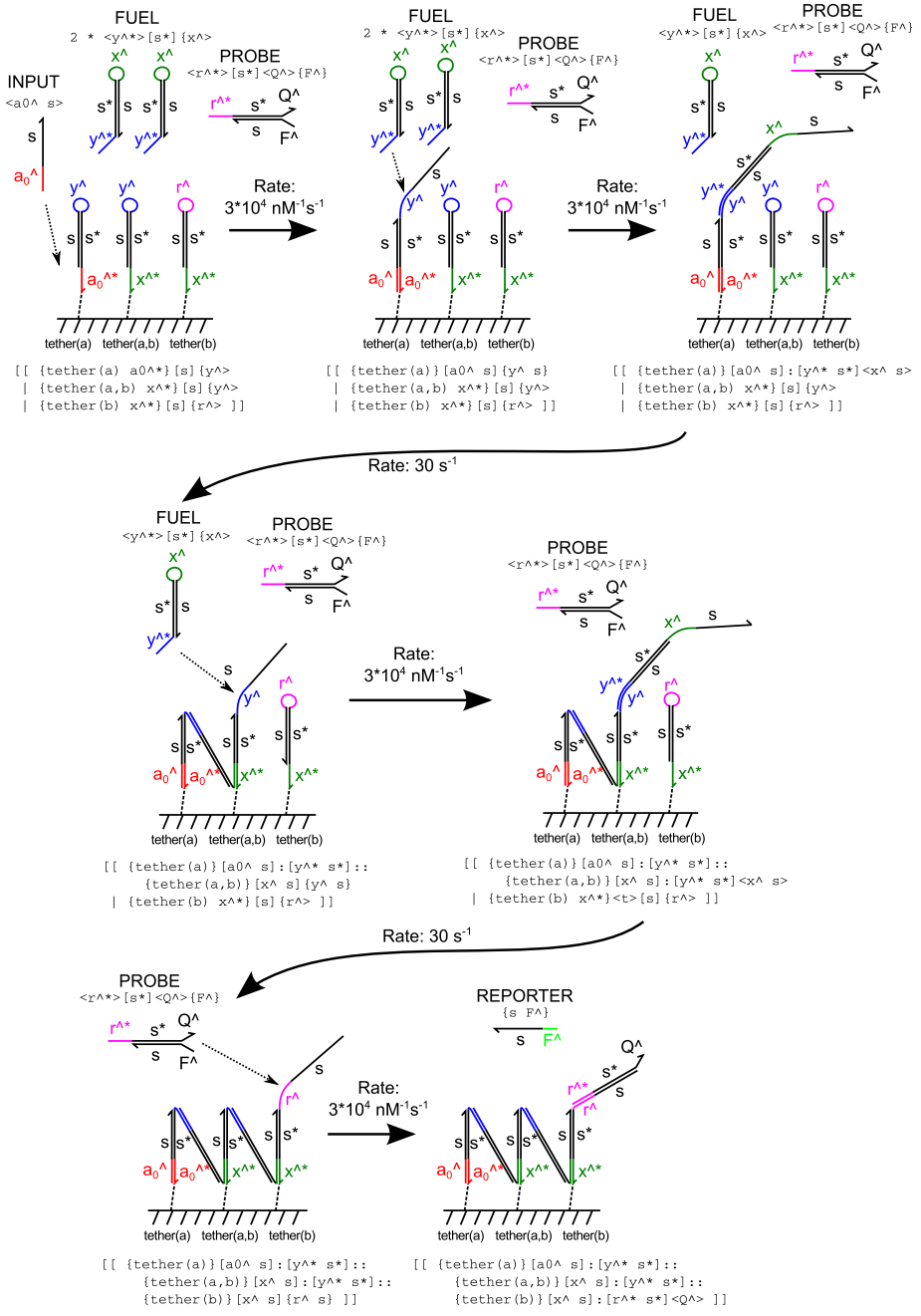


Fig. 5: DSD encoding of a variant on the full three-stator transmission line system from Figure 7 of [6]. To derive the reaction rate constants, we assume that all toeholds bind at the DSD default rate ($3 \times 10^{-4}\,\mathrm{nM}^{-1}\,\mathrm{s}^{-1}$) and that $lc(a) = lc(b) = 1 \times 10^5\,\mathrm{nM}$, giving a tethered interaction rate of $30\,\mathrm{s}^{-1}$.

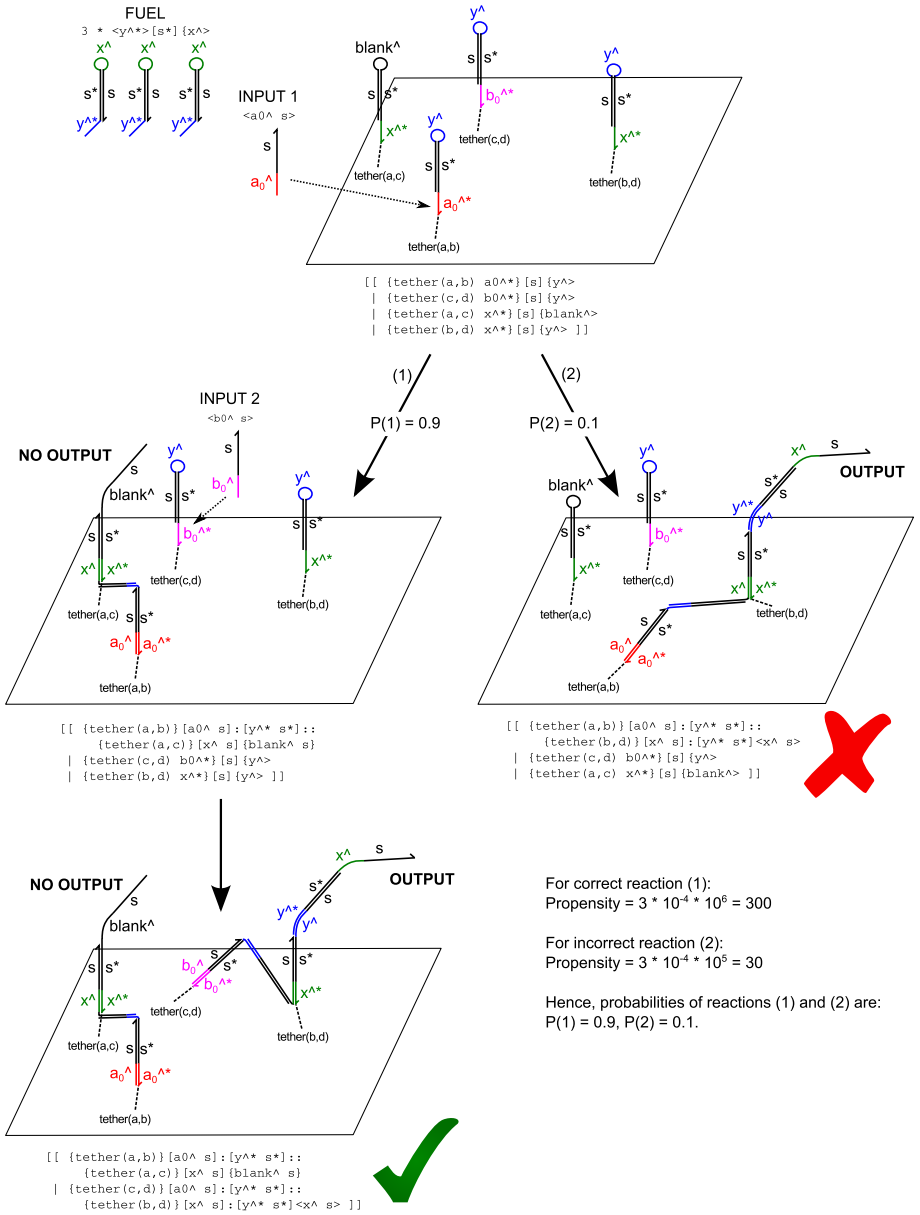


Fig. 6: DSD encoding of threshold-based spatial AND gate system from Figure 9 of [6]. We assume that input 1 arrives first, followed by input 2. The two possible trajectories for the system are outlined: one where the first input correctly binds to the threshold, and one where the first input erroneously triggers the output. To derive the reaction rate constants, we assume that all toeholds bind at the DSD default rate ($3 \times 10^{-4}\,\mathrm{nM}^{-1}\,\mathrm{s}^{-1}$), and that $lc(a) = lc(c) = 1 \times 10^6\,\mathrm{nM}$ and $lc(b) = lc(d) = 1 \times 10^5\,\mathrm{nM}$. The differing local concentrations produce a thresholding effect that gives AND logic.

be a complex DNA nanostructure that is non-trivial to represent in a formal language. The inclusion of DNA tiles in the language provides a means of encapsulating tethered species such that multiple tethered circuits can be simulated in a single solution. The result is a powerful tool for modelling and verifying more sophisticated tethered systems, e.g., to analyze the possible routes taken by walkers in a multi-track system [9].

8.1 Abstractions for tethered circuit design

For detailed design of tethered DNA circuits, a coordinate-based system for specifying the absolute positions of tethered components on a surface, e.g., a DNA origami tile, would be required. Ideally, this would be paired with a graphical design tool, so that the user could draw out the desired tether geometry directly, and could be integrated with existing DNA origami design tools such as caDNAno [17].

However, the coordinate-based approach requires a highly general biophysical model to predict the interaction rates of arbitrary DSD-representable structures with arbitrary toehold and tether locations. Previous calculations [5, 15] have derived expressions for tethered interaction rates for particular structures and tethering geometries, whereas to cope with the full generality of the DSD metalanguage a far more comprehensive physical model would be required.

In the absence of such a model, we chose a level of abstraction that is similar to the "channel-based" approach to specifying inter-process interactions in modelling languages such as the stochastic π-calculus [18, 19]. In the channel-based approach, all possible interactions between processes must be provided explicitly by the design via the mechanism of channel sharing. In this paper we have moved away from that idea to some extent by using the DSD reduction semantics to derive certain interactions between species, however, we still rely on a channel-like approach to specify which tethered species are close enough to react according to the reduction rules, via the mechanism of shared location tags.

By requiring the modeller to directly associate location concentrations with the various location tags, we shift the burden of computing the local concentrations to the modeller, who can perform structure-specific analyses to determine a reasonable value for the local concentration value, or alternatively fit these rate constants directly to experimental data, if available. Hence, a fully general biophysical model of the dynamics of tethered toehold interactions is not required. This approach gives a high degree of modelling flexibility, allowing measured rates to be included directly where available, or to be estimated using a biophysical model [5, 15]. However, the need to specify all possible interactions between tethered species means that the user must have some idea of the desired track geometry before encoding the system in DSD. Furthermore, potentially undesired interactions, such as track-jumping behaviour in molecular walker systems [9], cannot be inferred automatically by the compiler.

Hence, we envision that this method for the specification of tethered circuit behaviour will form but one layer of an abstraction hierarchy for the design and simulation of tethered DNA circuits, similar to our approach to the semantics of strand displacement reactions [11]. We see this model as sitting atop a more detailed coordinate-level specification, as described above. A geometric interpretation of location tags is that

each tag corresponds to a point, whose physical coordinate is the average of the physical coordinates of the tether locations that share that tag. If we assume the existence of a detailed, realistic model of tethered reaction kinetics, this geometric interpretation could form the basis of a compilation phase that takes a tethered system specified abstractly using location tags and computes possible physical coordinates for each tether location, producing a concrete design suitable for experimental implementation. This may involve an iterative optimization routine to find tether coordinates that satisfy the constraints specified in the abstract model.

Alternatively, coordinate-level specifications could be translated back into the abstract domain for ease of analysis—this process could automatically generate the location tag-based encoding without further input from the user. Furthermore, these translations could be combined so that a tethered circuit design specified in the abstract domain can be compiled into a detailed, coordinate-based representation and subsequently lifted back into the abstract domain. This process would exploit the detailed model of tethered species to detect any spurious interactions between components in an abstractly specified circuit, and could be iterated to refine the tether geometry to minimize or eliminate the spurious interactions. This abstraction hierarchy could be extended further by implementing automated layout algorithms that directly compile logical specifications into geometrically arranged tracks that execute the corresponding computation with minimal spurious interactions.

8.2 Molecular spiders

Another potential approach to implementing nanoscale locomotion is via multivalent catalytic walkers known as *molecular spiders*. These comprise multiple "legs" each of which is a catalytically active DNAzyme [20], all attached to a rigid body such as a streptavidin molecule. Molecular spiders move in a biased random walk due to cleavage of substrates displayed on a surface, and have been realized experimentally [21,22]. Previous work on computational analysis of molecular spider behaviour has used models ranging from the physically detailed [23, 24] to the more abstract [25, 26]. These simpler models have enabled computational studies of various effects, including cooperative nanoscale search due to self-avoidance [27] and maze navigation [28]. The latter is of particular interest with regard to our emphasis on the verification of track designs for molecular walkers. However, to model these systems in our framework would require us to extend the DSD framework further to model DNAzyme-catalyzed substrate cleavage reactions. Furthermore, the mechanism of spider motion implies that, for a given body position, any unattached leg has the choice of a number of potential attachment points. Encoding this mechanism using the approach proposed in this paper would require a very large number of interaction tags, because a separate tag would be needed for each pair of displayed substrates S_1 and S_2 that are sufficiently close together that a spider with a leg attached to S_1 can attach another leg to S_2. Hence, the resulting system would be rather cumbersome to simulate. Therefore, we believe that existing methods of simulating molecular spider dynamics using custom Monte Carlo simulation routines [23–26] are more practical than encoding them in our framework. Our work is more suited to encoding of tethered circuits whose interactions are constrained by the geometry of the track layout [5, 6].

8.3 Representable structures

As the set of DSD-representable structures grows, we will gain increased power and flexibility for the design of tethered reaction systems. In particular, by enabling automatic, integrated compilation of enzymatic reactions, such as restriction enzyme and nickase reactions, we hope to model an important class of DNA walkers powered by enzymatic reactions, e.g., as in [3]. We also hope to combine this work with a formalization of dendritic DSD structures to enable simulation of tethered logic circuits with fan-in where both inputs must bind simultaneously [5], although several examples, such as the threshold-based AND circuit described in Section 7, do not require this extension. Finally, including four-way branch migration would enable us to encode additional published DNA walker designs [2].

Acknowledgments

The authors thank Filippo Polo for his work on the DSD implementation of the tethered semantics. This material is based upon work supported by the National Science Foundation under grants 1028238 and 1318833. M.R.L. gratefully acknowledges support from the New Mexico Cancer Nanoscience and Microsystems Training Center (NIH/NCI grant 5R25CA153825).

References

1. R. D. Vale. The molecular motor toolbox for intracellular transport. *Cell*, 112(4):467–480, 2003.
2. R. A. Muscat, J. Bath, and A. J. Turberfield. A programmable molecular robot. *Nano Lett*, 11(3):982–987, 2011.
3. S. F. J. Wickham, J. Bath, Y. Katsuda, M. Endo, K. Hidaka, H. Sugiyama, and A. J. Turberfield. A DNA-based molecular motor that can navigate a network of tracks. *Nature Nanotech*, 7:169–173, 2012.
4. H. Gu, J. Chao, S.-J. Xiao, and N. C. Seeman. A proximity-based programmable DNA nanoscale assembly line. *Nature*, 465:202–205, 2010.
5. H. Chandran, N. Gopalkrishnan, A. Phillips, and J. Reif. Localized hybridization circuits. In L. Cardelli and W. Shih, editors, *Proceedings of DNA17*, volume 6937 of *LNCS*, pages 64–83. Springer-Verlag, 2011.
6. R. A. Muscat, K. Strauss, L. Ceze, and G. Seelig. DNA-based molecular architecture with spatially localized components. In *Proceedings of ISCA '13*, 2013.
7. J. Heath, M. Kwiatkowska, G. Norman, D. Parker, and O. Tymchyshyn. Probabilistic model checking of complex biological pathways. *Theor Comput Sci*, 319(3):239–257, 2008.
8. M. R. Lakin, D. Parker, L. Cardelli, M. Kwiatkowska, and A. Phillips. Design and analysis of DNA strand displacement devices using probabilistic model checking. *J R Soc Interface*, 9(72):1470–1485, 2012.
9. F. Dannenberg, M. Kwiatkowska, C. Thachuk, and A. J. Turberfield. DNA walker circuits: Computational potential, design, and verification. In D. Soloveichik and B. Yurke, editors, *Proceedings of DNA19*, volume 8141 of *LNCS*, pages 31–45. Springer-Verlag, 2013.
10. A. Phillips and L. Cardelli. A programming language for composable DNA circuits. *J R Soc Interface*, 6(Suppl. 4):S419–S436, 2009.

11. M. R. Lakin, S. Youssef, L. Cardelli, and A. Phillips. Abstractions for DNA circuit design. *J R Soc Interface*, 9(68):470–486, 2012.
12. A. J. Turberfield, J. C. Mitchell, B. Yurke, A. P. Mills, Jr., M. I. Blakey, and F. C. Simmel. DNA fuel for free-running nanomachines. *Phys Rev Lett*, 90(11):118102, 2003.
13. G. Seelig, B. Yurke, and E. Winfree. Catalyzed relaxation of a metastable DNA fuel. *J Am Chem Soc*, 128:12211–12220, 2006.
14. S. J. Green, J. Bath, and A. J. Turberfield. Coordinated chemomechanical cycles: A mechanism for autonomous molecular motion. *Phys Rev Lett*, 101:238101, 2008.
15. A. J. Genot, D. Y. Zhang, J. Bath, and A. J. Turberfield. Remote toehold: A mechanism for flexible control of DNA hybridization kinetics. *J Am Chem Soc*, 133(7):2177–2182, 2011.
16. M. R. Lakin, S. Youssef, F. Polo, S. Emmott, and A. Phillips. Visual DSD: a design and analysis tool for DNA strand displacement systems. *Bioinformatics*, 27(22):3211–3213, 2011.
17. S. M. Douglas, A. H. Marblestone, S. Teerapittayanon, A. Vazquez, G. M. Church, and W. M. Shih. Rapid prototyping of three-dimensional DNA-origami shapes with caDNAno. *Nucleic Acids Res*, 37:5001–5006, 2009.
18. C. Priami, A. Regev, E. Shapiro, and W. Silverman. Application of a stochastic name-passing calculus to representation and simulation of molecular processes. *Inform Process Lett*, 80:25–31, 2001.
19. A. Phillips and L. Cardelli. Efficient, correct simulation of biological processes in the stochastic pi-calculus. In M. Cakder and S. Gilmore, editors, *Proceedings of CMSB 07*, volume 4695 of *LNCS*, pages 184–199. Springer-Verlag, 2007.
20. Y. Li and R. R. Breaker. Deoxyribozymes: new players in the ancient game of biocatalysis. *Curr Opin Struct Biol*, 9:315–323, 1999.
21. K. Lund, A. J. Manzo, N. Dabby, N. Michelotti, A. Johnson-Buck, J. Nangreave, S. Taylor, R. Pei, M. N. Stojanovic, N. G. Walter, E. Winfree, and H. Yan. Molecular robots guided by prescriptive landscapes. *Nature*, 465:206–210, 2010.
22. R. Pei, S. K. Taylor, D. Stefanovic, S. Rudchenko, T. E. Mitchell, and M. N. Stojanovic. Behavior of polycatalytic assemblies in a substrate-displaying matrix. *J Am Chem Soc*, 128(39):12693–12699, 2006.
23. M. J. Olah. *Multivalent Random Walkers: A computational model of superdiffusion at the nanoscale*. PhD thesis, University of New Mexico, 2012.
24. M. J. Olah and D. Stefanovic. Superdiffusive transport by multivalent molecular walkers moving under load. *Phys Rev E*, 87:062713, 2013.
25. O. Semenov. *Abstract Models of Molecular Walkers*. PhD thesis, University of New Mexico, 2013.
26. O. Semenov, D. Mohr, and D. Stefanovic. First passage properties of molecular spiders. *Phys Rev E*, 88:012724, 2013.
27. O. Semenov, M. J. Olah, and D. Stefanovic. Cooperative linear cargo transport with molecular spiders. *Natural Computing*, 12(2):259–276, 2013.
28. D. Stefanovic. Maze exploration with molecular-scale walkers. In A.-H. Dediu, C. Martín-Vide, and B. Truthe, editors, *Proceedings of TPNC 2012*, volume 7505 of *LNCS*, pages 216–226. Springer-Verlag, 2012.