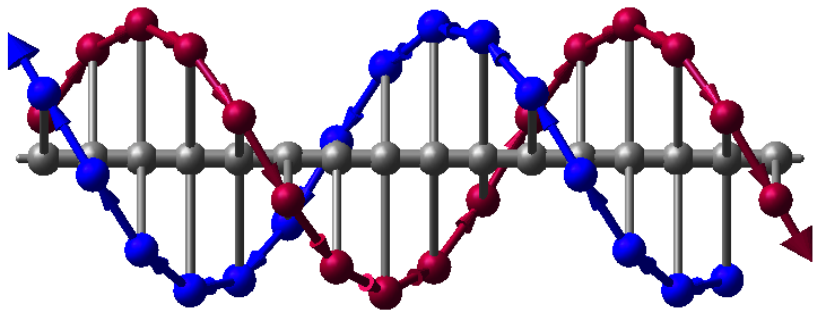
Graph Rewriting System (GRS) for Chemical Reactions on DNA Nanostructures

- PhD Candidate
- Department of Computer Science
- Duke University



(a)

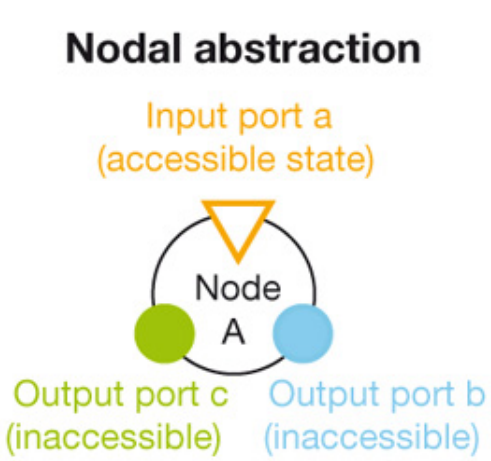
(b)

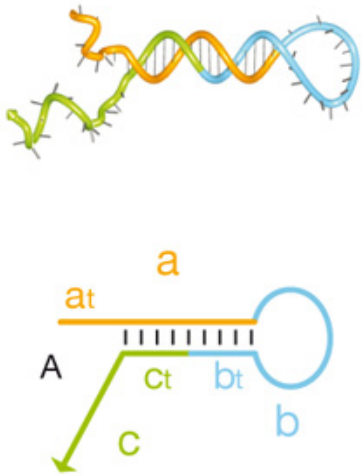
Fig. 1: (a) A rendering of a double-stranded DNA duplex (nanoengineer). (b) A cartoon rendering of the duplex.

Prior work in this area involves the use of graph-based models for the representation of DNA nanodevices, including:

* Sherman et al [1], Jonoska et al [3], Reif et al [10], & Kawamata et al [4]
* Cardelli et al [2,7,8], Kumara et al [6], Yin et al [11] **
* Klavins et al [5] developed graph grammars models for robotic self-assembly, and also adapted this in DNA self-assembly

** Peng Yin's work on programming assembly pathways





Modeling DNA Nanodevices Using Graph Rewriting Systems

Reem Mokhtar, Sudhanshu Garg, Harish Chandran, Hieu Bui, Tianqi Song, and John Reif, Modeling DNA Nanodevices Using Graph Rewriting Systems, invited Chapter, Advances in Unconventional Computing, Springer (2015).

Slides by Reem Mokhtar

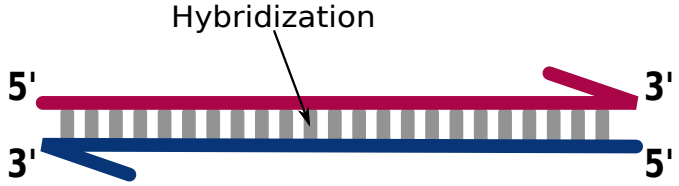
Motivation

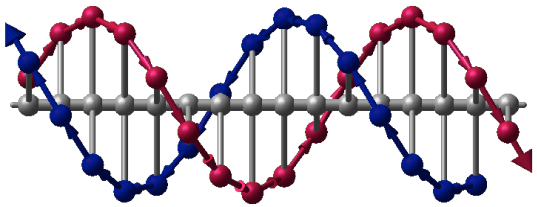
A key need in the field of DNA-based nanosystems is the coarse-grained modeling and simulation of dynamic DNA nanodevices. This means that the model should provide a representation of:

* sequence domains
* secondary structure and
* other properties that affect hybridization and enzymic reactions

but not necessarily specify their tertiary structure or details provided by low-level molecular simulations, not of key importance to these reactions on DNA nanostructures.

DNA cartoons are widely used to provide a visual representation of the secondary structure of DNA nanostructures, including 5' to 3' directionality and hybridizations between single strands of DNA.





They may also be used by visual GUI software, e.g., for graphical specification and rendering of DNA nanostructures. However, explicit graph structures are needed for software to model DNA nanostructures.

Our Approach

We provide a new set of techniques for modeling DNA nanodevices, that includes:

(a) a novel DNA-digraph notation of specifying DNA nanostructures,
(b) a grammar whose productions specify the reactions involving DNA strands and nanostructures,
(d) assignment of kinetic rates to grammar productions,
(e) calculation of kinetics of the resulting reaction system by conventional software.

Our Graph Rewriting System (GRS) for DNA Nanostructures:

A *DNA graph* is formally defined as an 8-tuple $G = (V, E, L_V, L_E, \Sigma, vl, el, \delta$ where:

1. V is a finite set of vertices (domains or strand ends)
2. E is a finite set of edges (bonds or strand-end edge)
3. $L_V = \{\bigcirc, \bullet, 3', 5'\}$ is the finite set of vertex labels, where:
   (a) $\bigcirc$: unhybridized vertex
   (b) $\bullet$: hybridized vertex
   (c) $3'$: 3 prime end.
   (d) $5'$: 5 prime end.
4. $L_E = \{\dashv\!\!\!\!\rightarrow, \cdots\cdots, \longrightarrow, \dashv\}$ is the finite set of edge labels, where
   (a) $\dashv\!\!\!\!\rightarrow$: base-stacking, where the arrow direction indicates $5'$ to 3 directionality.
   (b) $\cdots\cdots$: hybridization.
   (c) $\longrightarrow$: covalent bond between two domains (arrow indicates strand directionality)
   (d) $\dashv$: is an edge label that signifies a strand-end
5. Σ is the set of all possible domain names. For each $d \in \Sigma$, its complement is denoted as $\bar{d}$.
6. $vl : V \mapsto L_V$, is a function that assigns a label to each vertex.
7. $el : E \mapsto L_E$, is a function that assigns a label to each edge.
8. $\delta : V \mapsto \Sigma$ is a mapping between vertices and domain names.

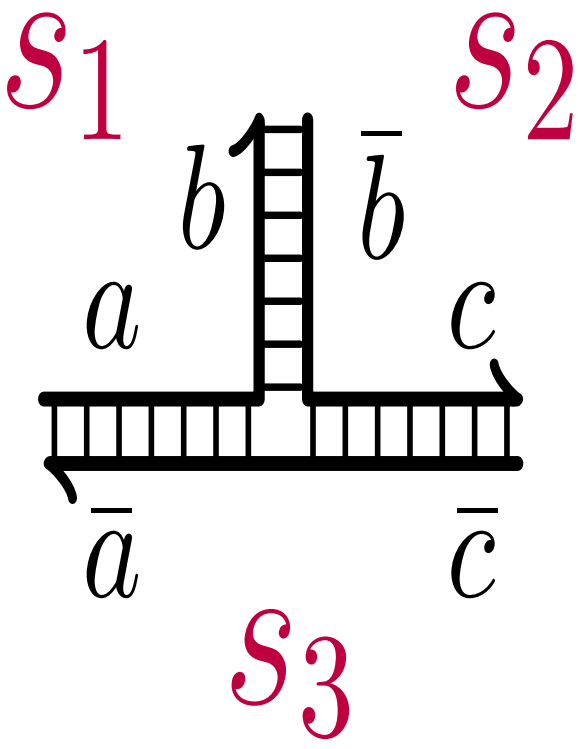
GRS Notation:

DNA Digraph Grammar Notation

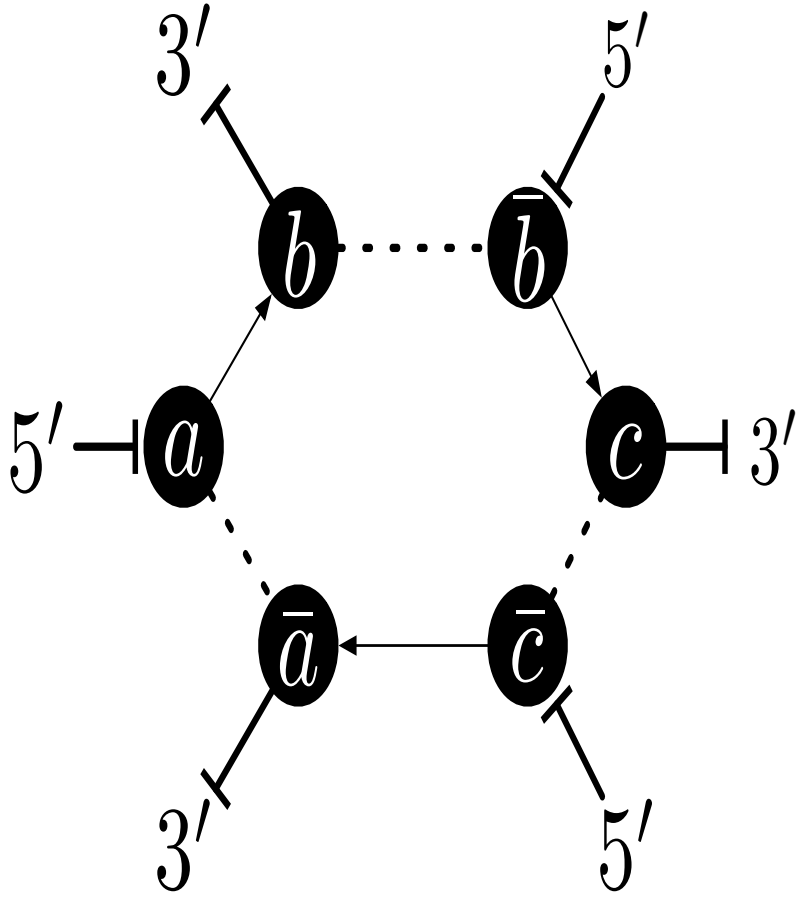
A labeled DNA-digraph $G = (V, E, VL, EL, vl, el)$ where:

- V is the vertex set. Each vertex represents a DNA strand domain or 3′(or 5′) end of a DNA strand.
- E is the edge set. Each edge represents a relationship (chemical bond) between two domains.
- $VL = \{\bigcirc, \bullet, {}^{3'}\!\leftarrow, {}^{5'}\!\rightarrow\}$ is vertex label set, where:
  - $\bigcirc$: unhybridized domain.
  - $\bullet$: hybridized domain.
  - ${}^{3'}\!\leftarrow$: 3′ end.
  - ${}^{5'}\!\rightarrow$: 5′ end.
- $EL = \{\rightarrow\!\!\parallel\!\!\rightarrow, \text{--------}, \longrightarrow\}$ is edge label set, where:
  - $\rightarrow\!\!\parallel\!\!\rightarrow$: base-stacking, where arrows indicate the directionality, going from 5′ to 3′.
  - --------: hybridization.
  - $\longrightarrow$: covalent bond linking base pairs between two domains.
- $vl : V \mapsto VL$, function to assign a label to each vertex.
- $el : E \mapsto EL$, function to assign a label to each edge.

An Example:



A DNA Nanostructure



Graph Model

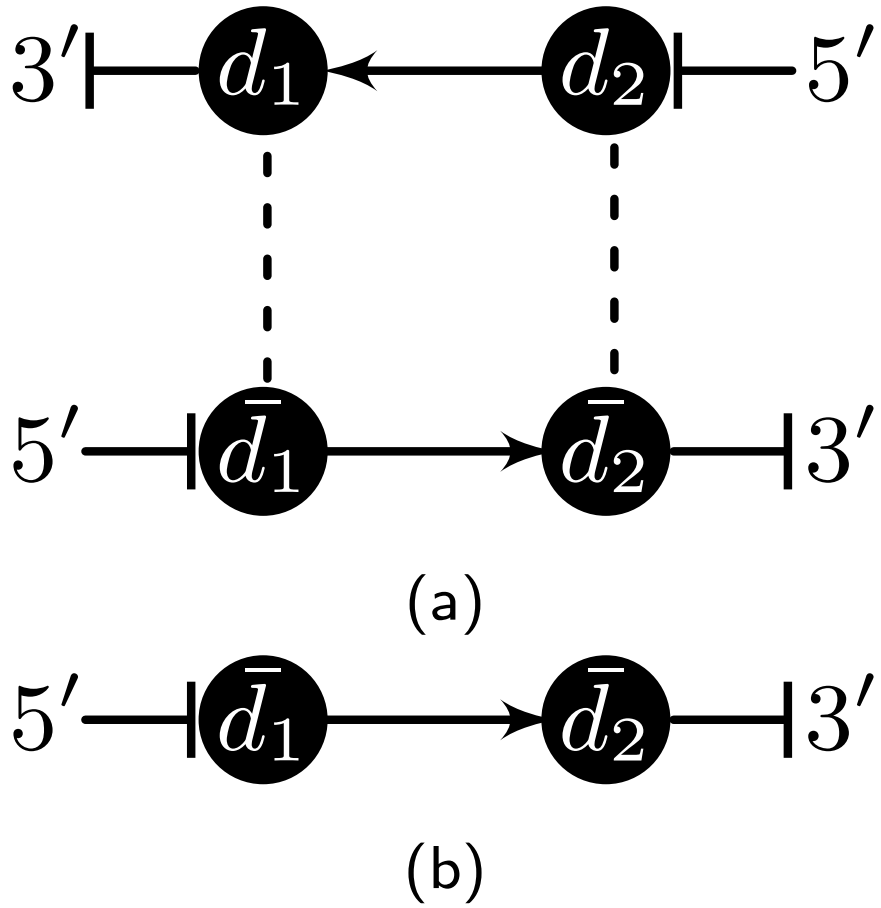
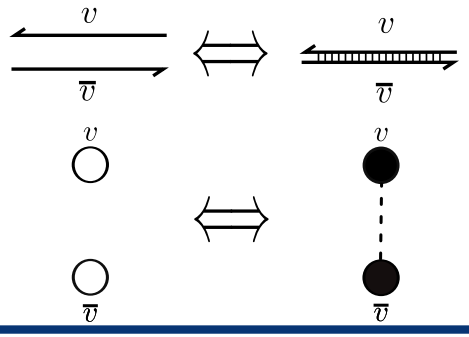


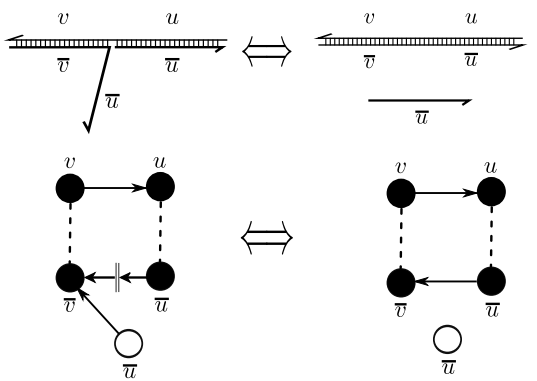
Fig. 3: (a) An example well-formed DNA graph. (b) An example of a DNA strand graph

Production Rules

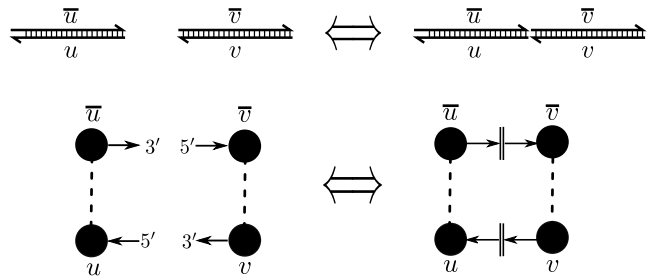
Rule #1:
Toehold binding



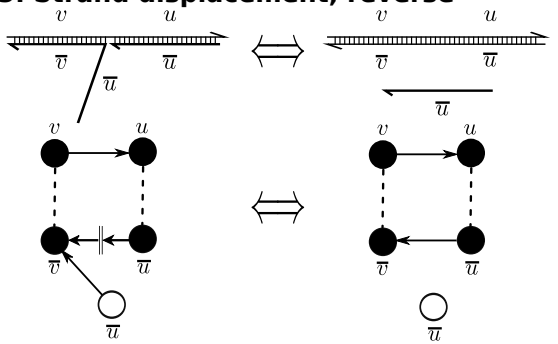
Rule #2:
Strand displacement



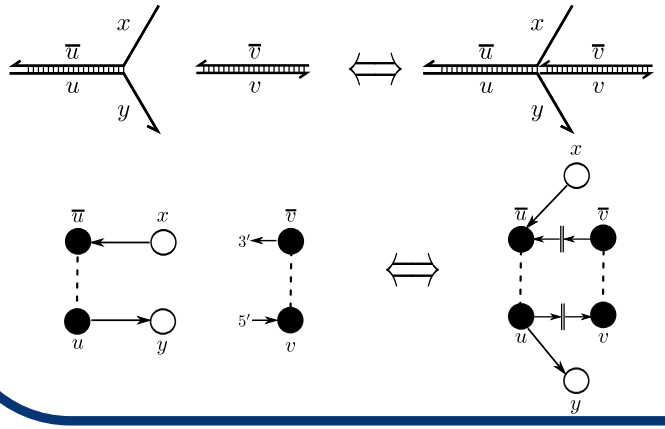
Rule #4: Base stacking



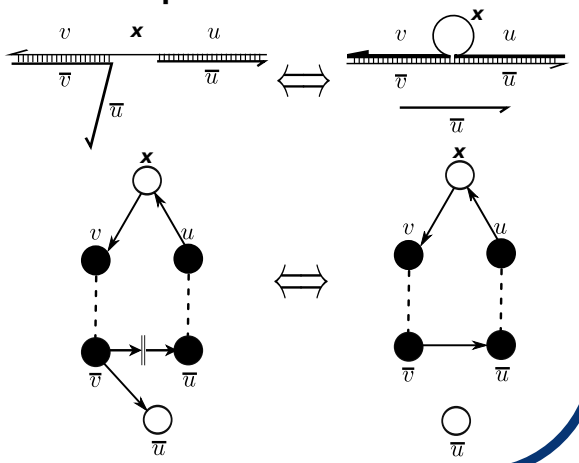
Rule #3: Strand displacement, reverse



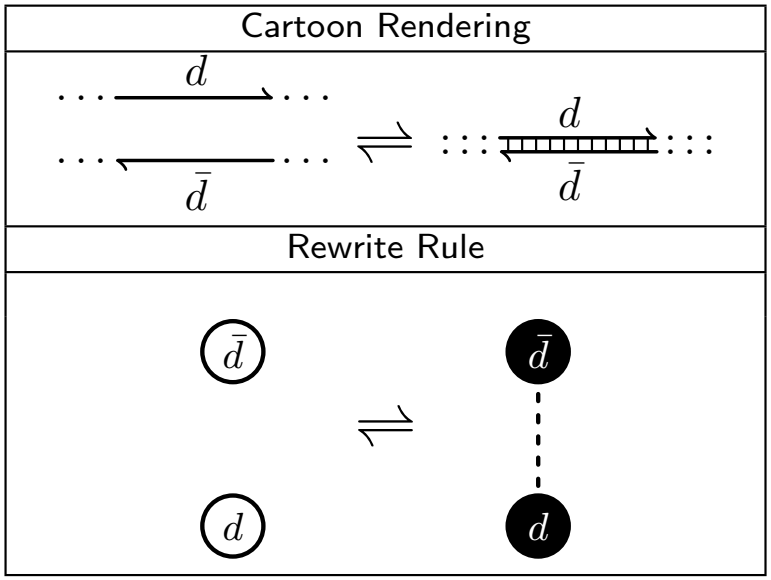
Rule #5:
Base stacking
with overhangs



Rule #6: Remote Toehold mediated Strand Displacement



Rule 1: Toehold Binding

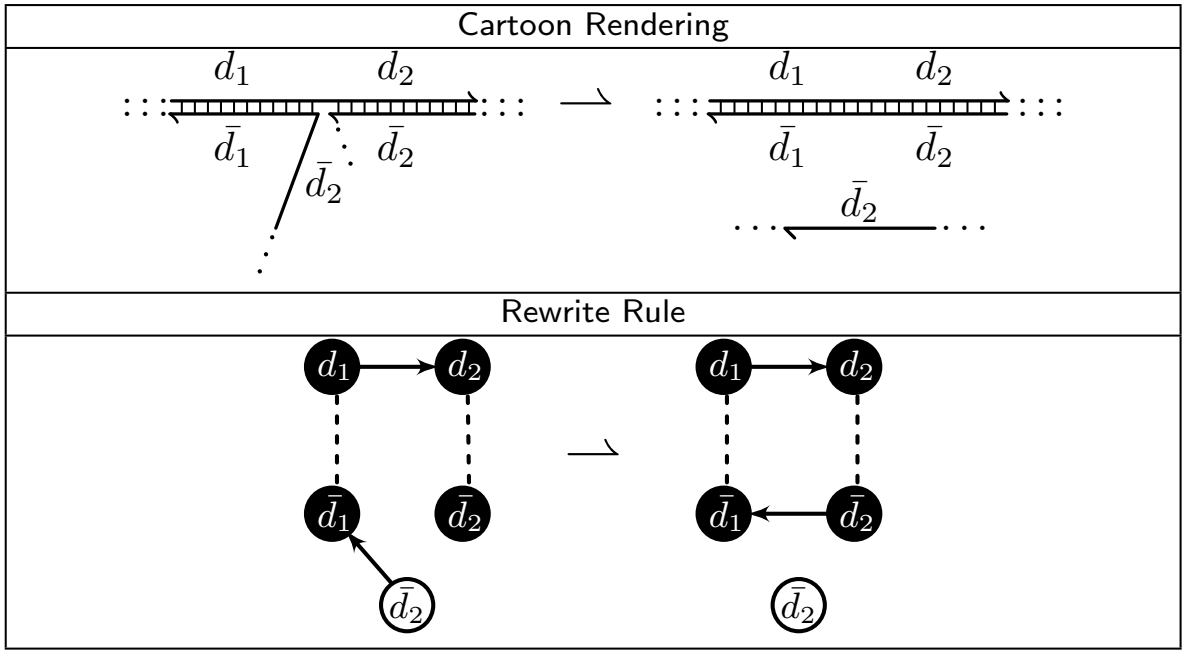


Rewrite Rule: Two complementary DNA domains d and $\bar{d}$ hybridize together to form a duplex, or double-stranded DNA. Note, the cartoon represents two strands each made up of one domain, but the strands may be surrounded by other domains (vertices) in either direction, while maintaining their original directionality.

$\mathbf{L}$: The unhybridized domains d and $\bar{d}$ are each represented by a circle, with no edge between them.

$\mathbf{R}$: They are each represented as solid circles (hybridized). The dashed edge represents the hydrogen bond(s) between d and $\bar{d}$.

Rule 2: Strand Displacement



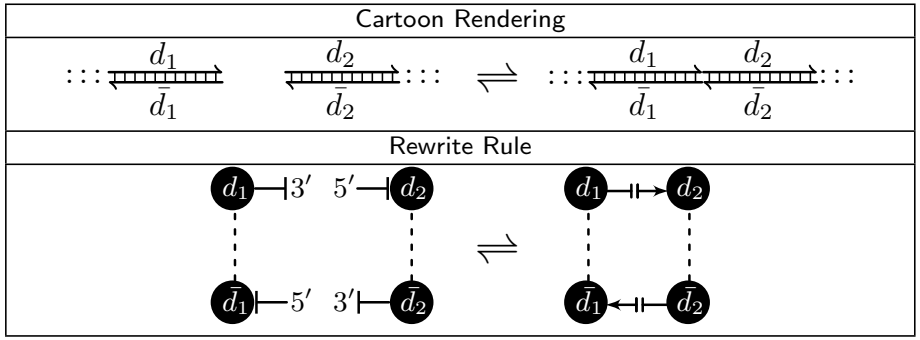
Rewrite Rule: Five domains are involved in this transformation. The vertex with domain label $\bar{d_2}$, which corresponds to the domain covalently bound to the complex on the left-hand side, displaces a vertex labeled with the same domain $\bar{d_2}$, which corresponds to the domain hybridized to d_2.

L: The upper strand region is composed of domains d_1,d_2. d_1 is hybridized to $\bar{d_1}$, and d_2 is hybridized to its complement $\bar{d_2}$.

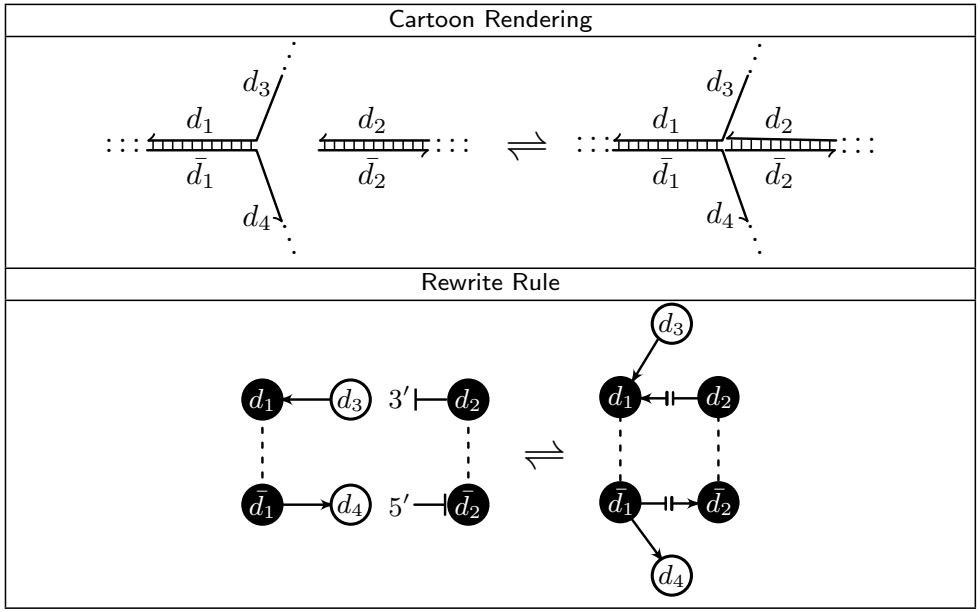
R: The hybridization bond between vertex $\bar{d_2}$ and d_2 has been removed, and $\bar{d_2}$ has now been relabeled with ◯. The vertex with domain $\bar{d_2}$, which is covalently bound to $\bar{d_1}$, is relabeled as hybridized (●), and a hybridization edge has now been added to connect this vertex to vertex $\bar{d_2}$.

Rules for Base Stacking

Rule #3: Base stacking

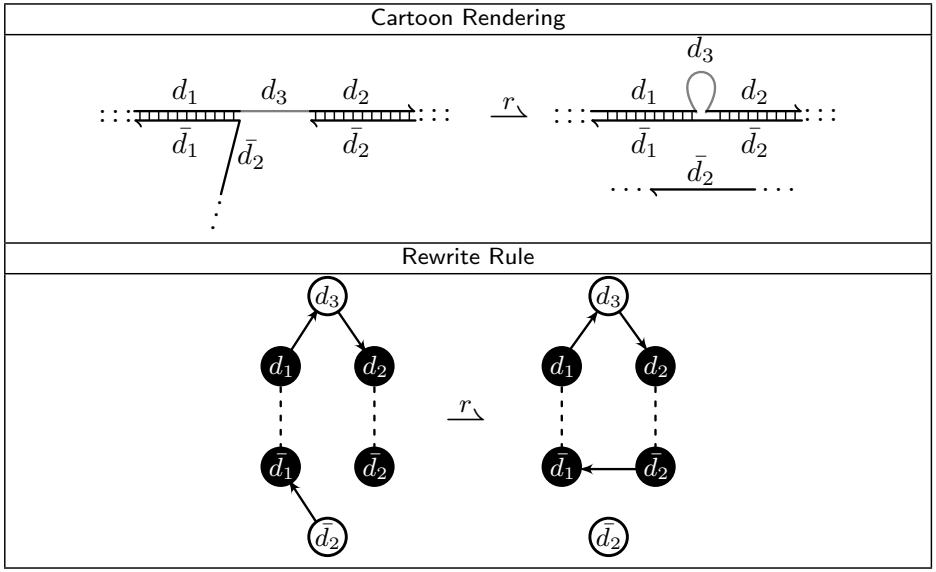


Rule #4: Base stacking with overhangs

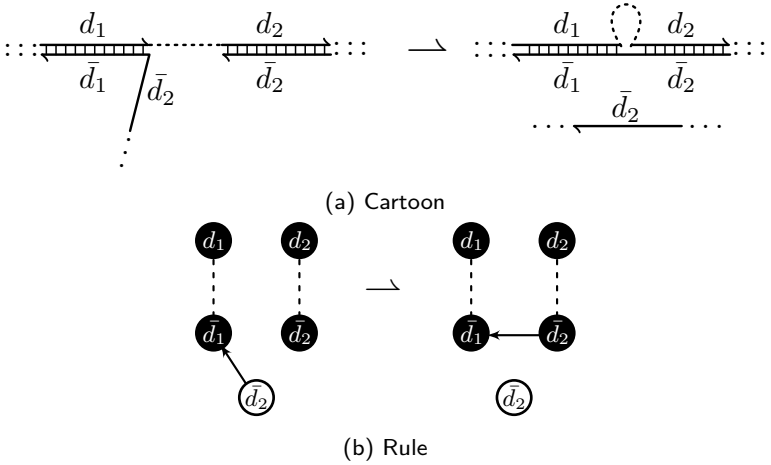


Rules for Remote Toehold mediated Strand Displacement

Rule #5: Remote Toehold Genot et al. (2011) mediated Strand Displacement



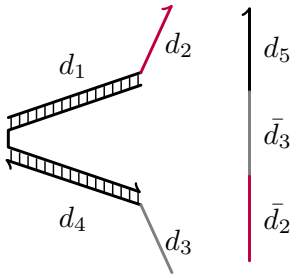
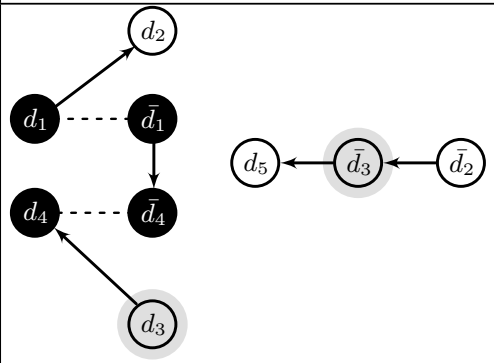
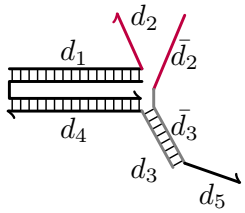
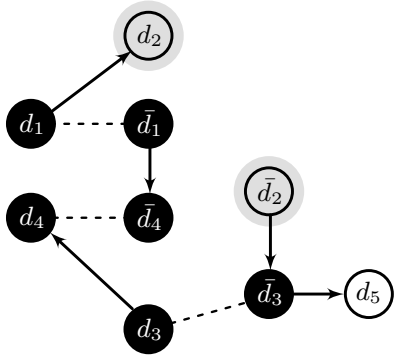
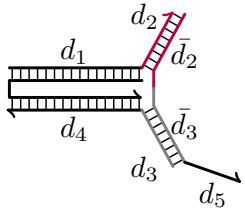
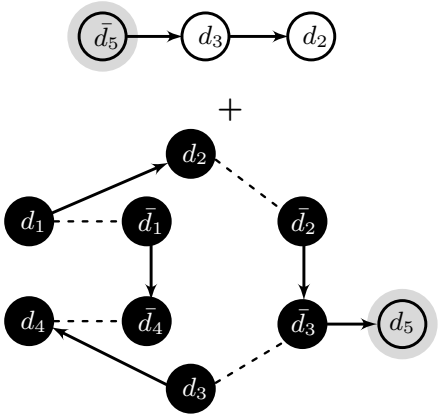
Distal Toehold Mediated Strand Displacement using a connectivity predicate check



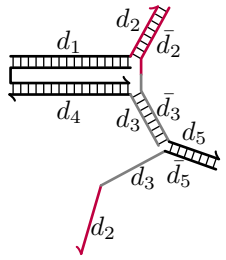
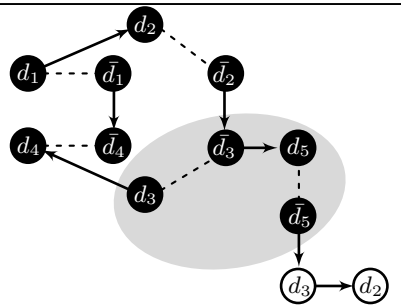
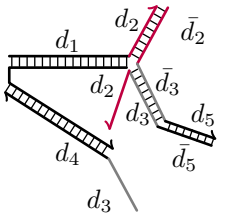
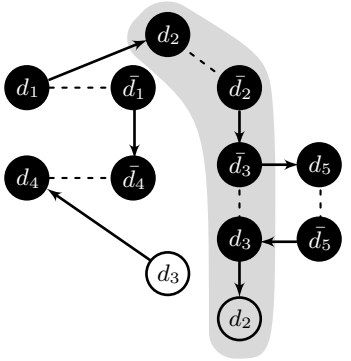
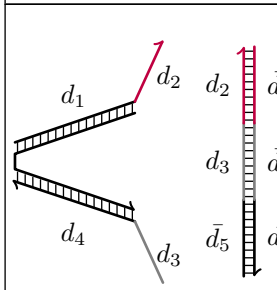
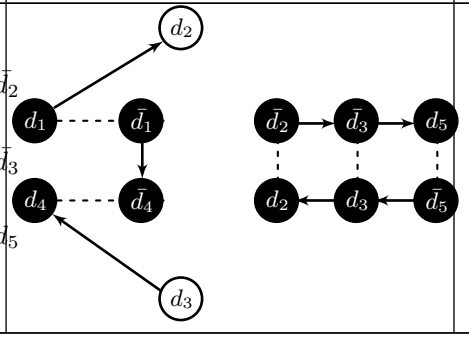
(a) Cartoon

(b) Rule

Table 1: One sequence of rule-applications representing one possible reaction pathway

Cartoon Rendering	Intermediate Graph G_i	Matching Rule
		Initial input species (G_0) has matching subgraph consisting of vertices with domains d_3 and $\bar{d}_3$, which matches the LHS of Rule #1.
		Rule #1
		Rule #1

Continued on next page

Cartoon Rendering	Intermediate Graph G_i	Matching Rule
		Rule #2
		Rule #2
		

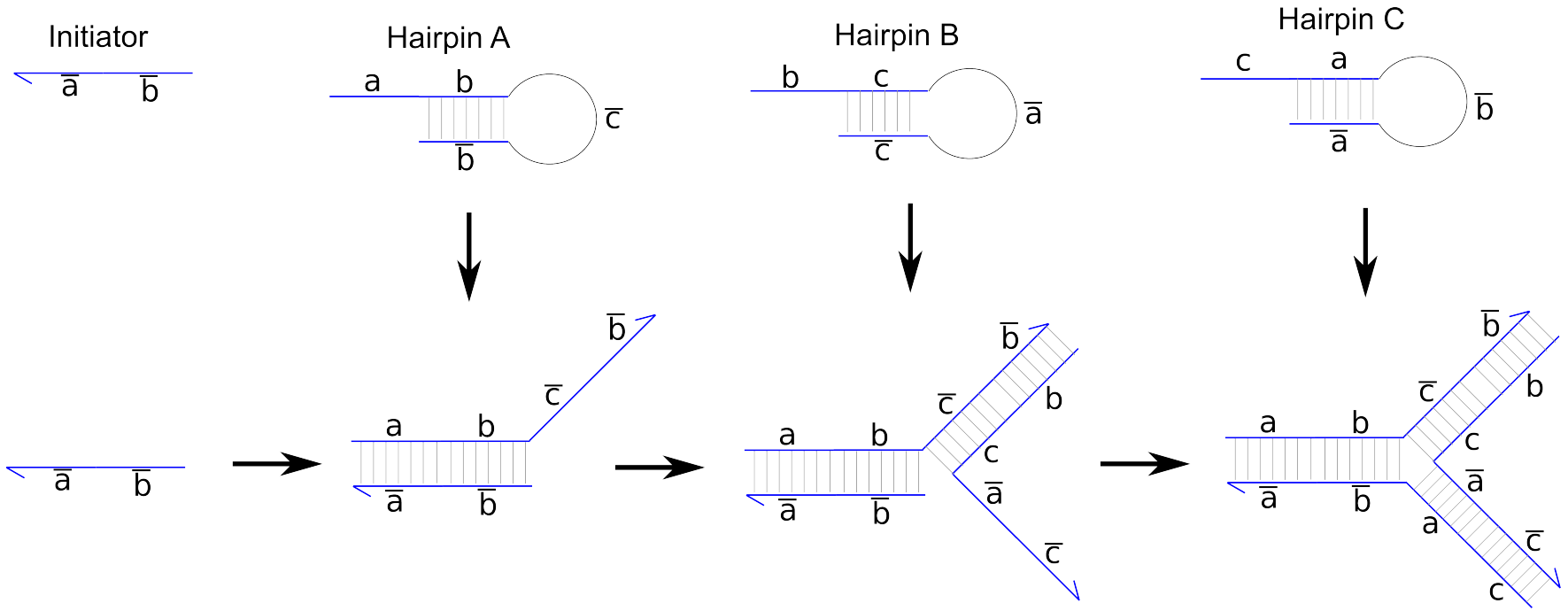


Fig. 6: Catalytic hairpin-based trigger branched junction

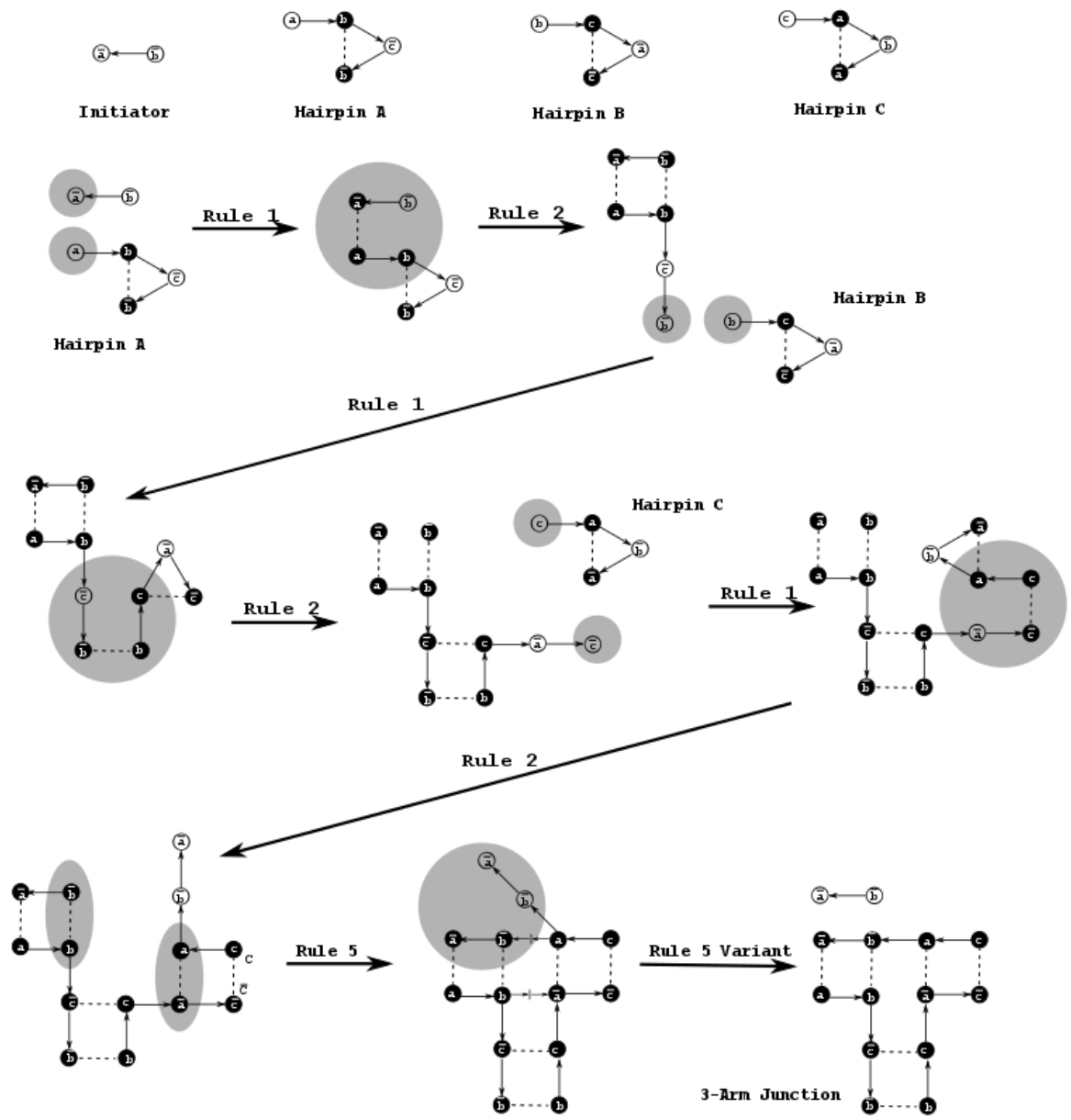


Fig. 7: The figure shows a sequence of graph rewrite rules that can be applied in succession. We obtain the 3-arm junction in the above design by Yin et al. (2008). Note that the application of rule #5 here includes the variation of the original (see Subsection 4.1).

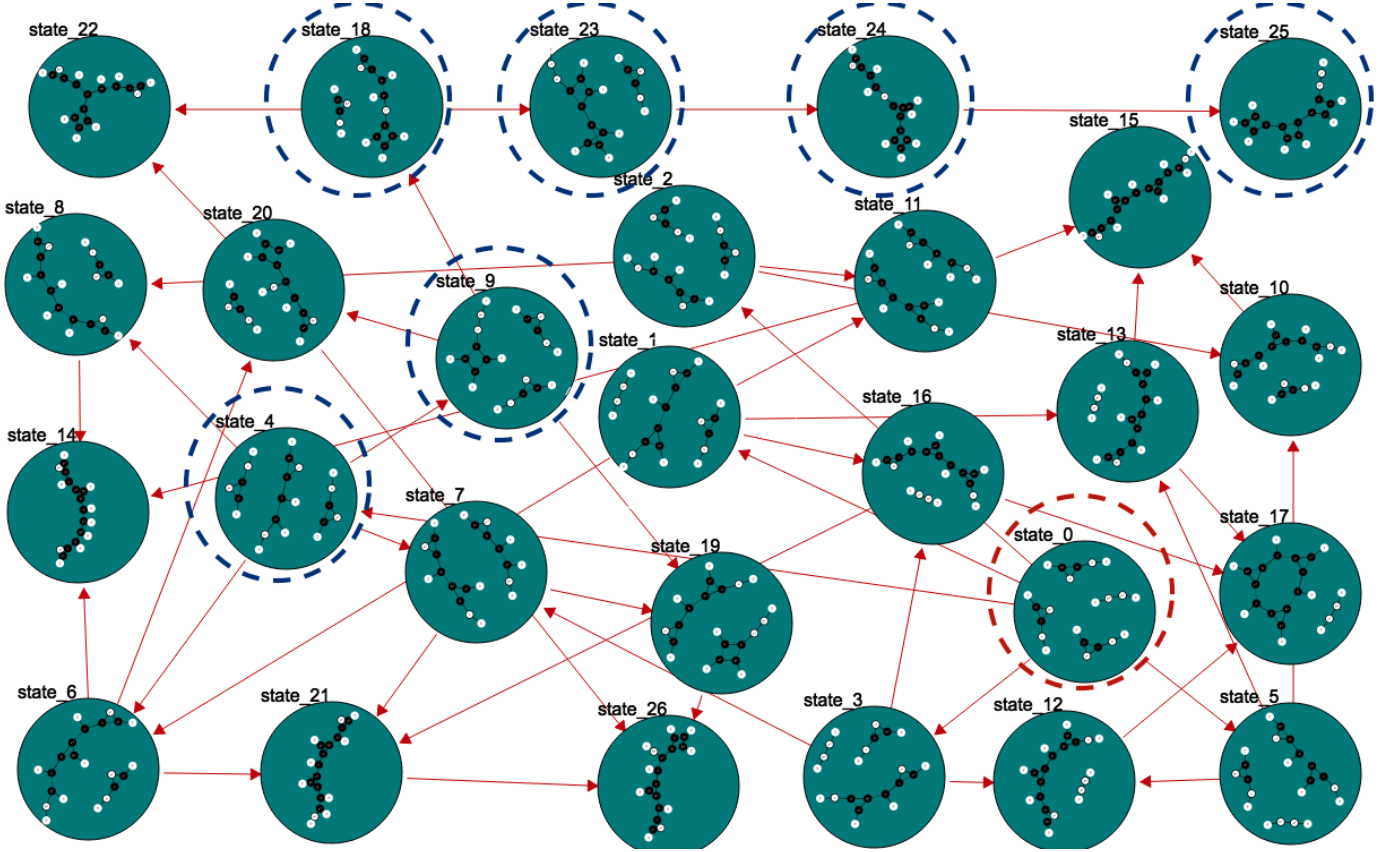


Fig. 9: The set of states generated by the graph rewriting rules given the input species. There are 26 states. Those circled in red (state_0) and blue mark the chosen subset of states in Figure 8

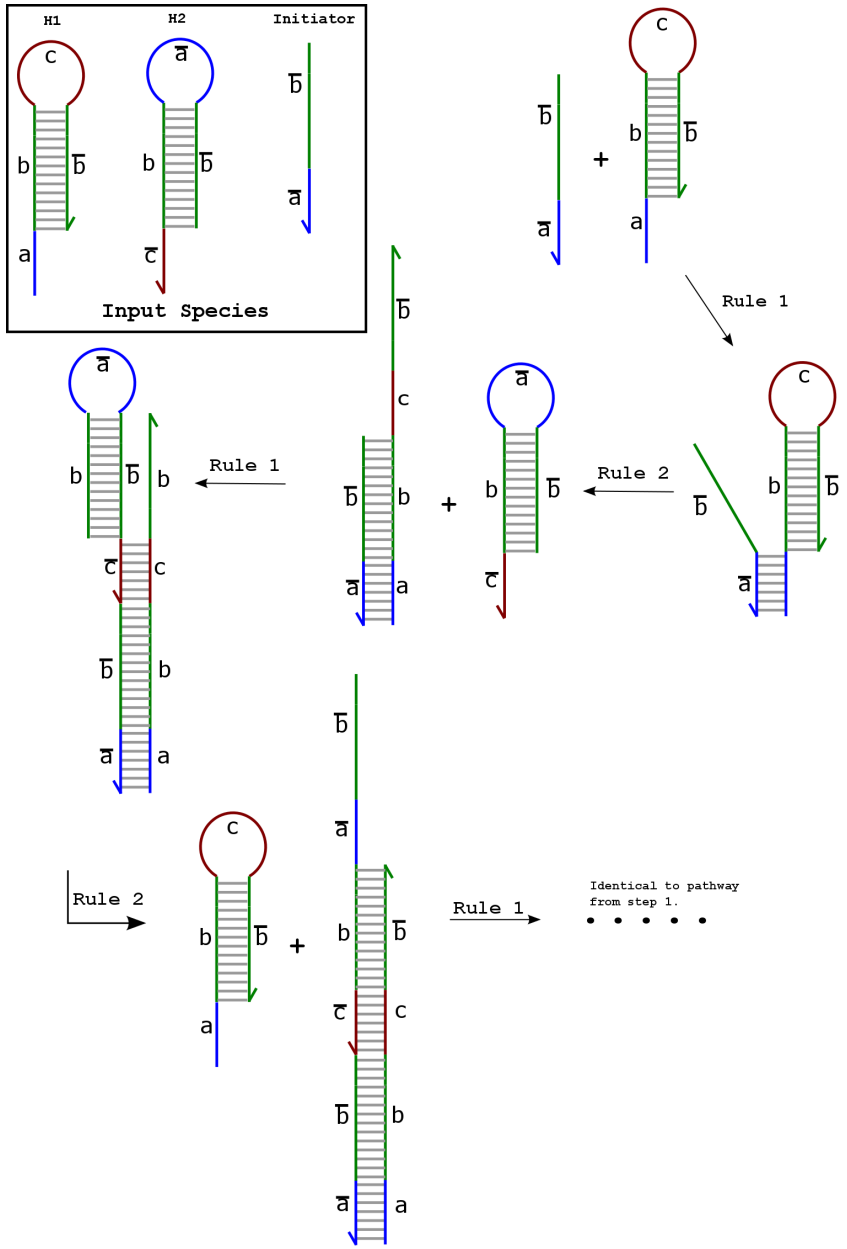


Fig. 12: Depiction of HCR system reaction

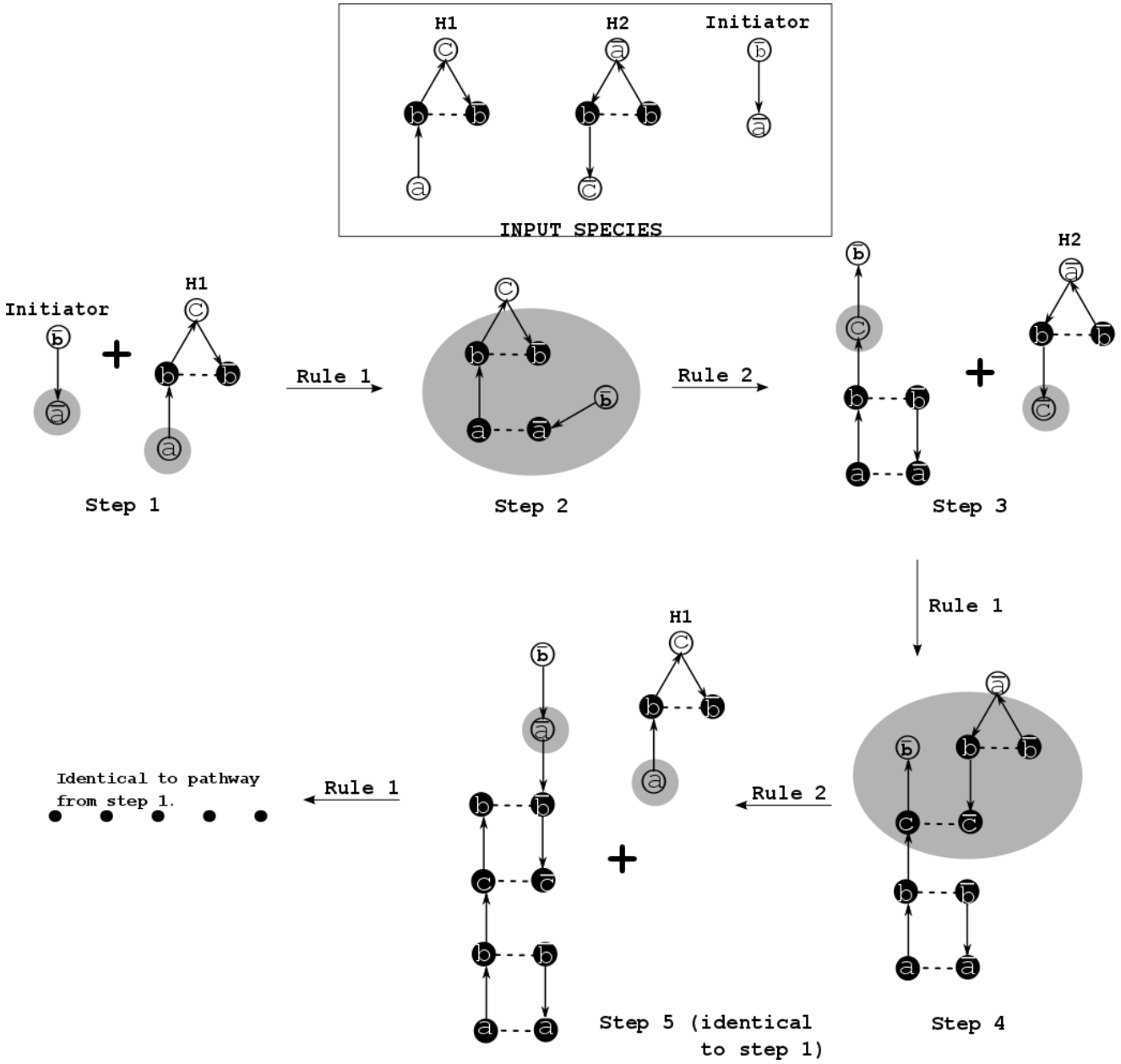


Fig. 13: The figure shows how to apply our graph rewriting systems to the basic HCR system designed by Dirks and Pierce (2004)

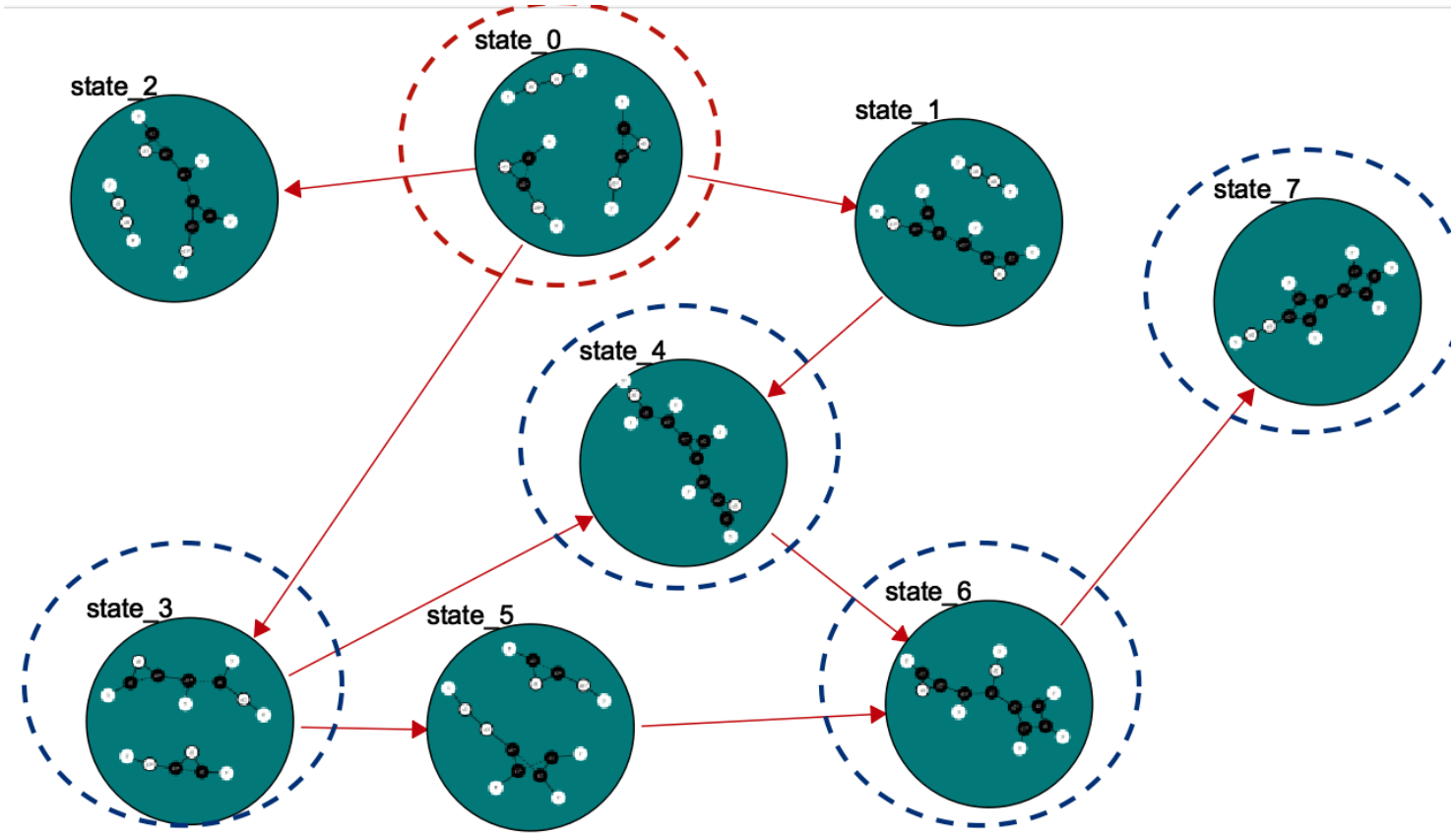
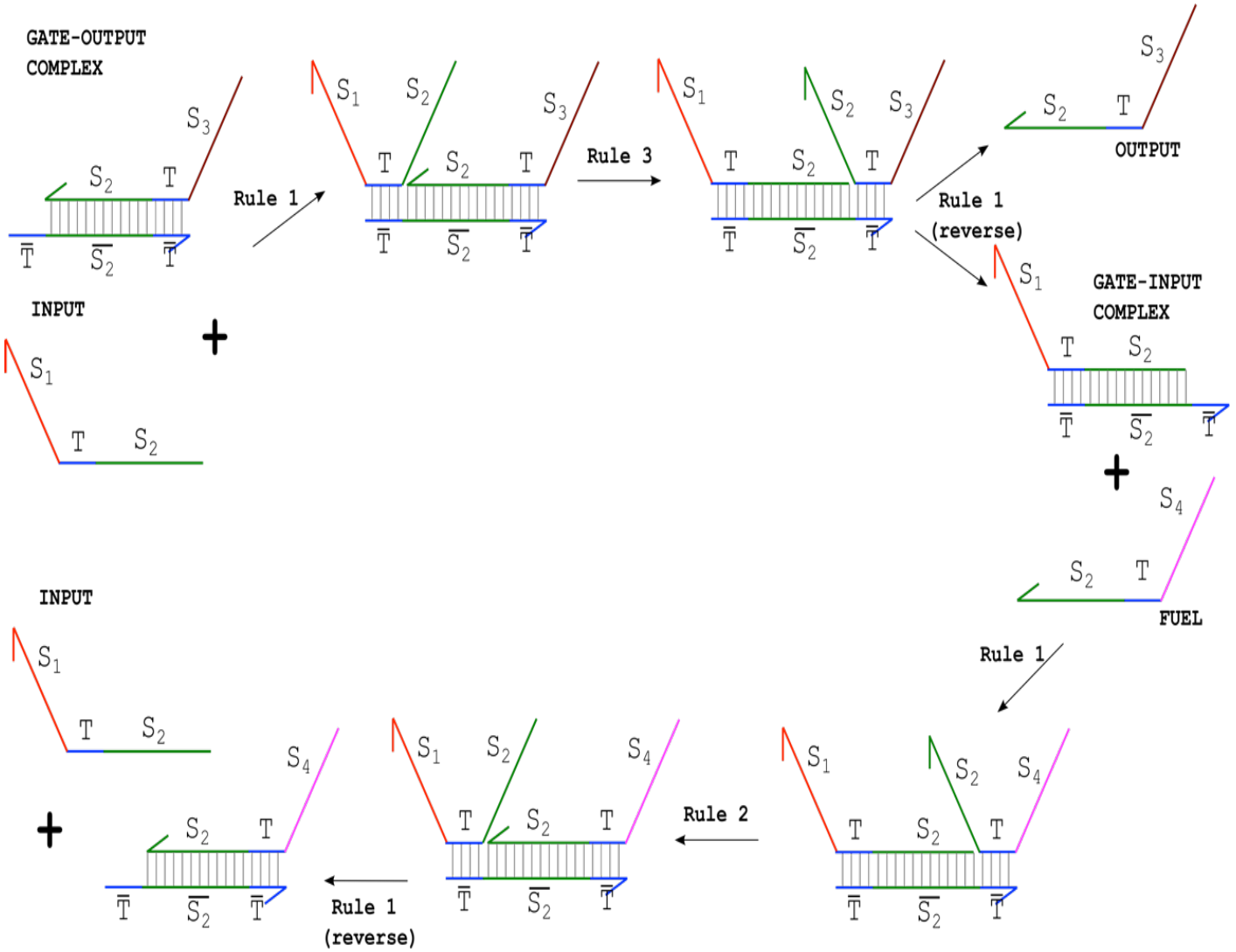


Fig. 15: States generated by DAGRS given the subset of rules supplied. Those circled in red (state_0) and blue mark the chosen subset of states in Figure 14

Example: See-Saw Gates

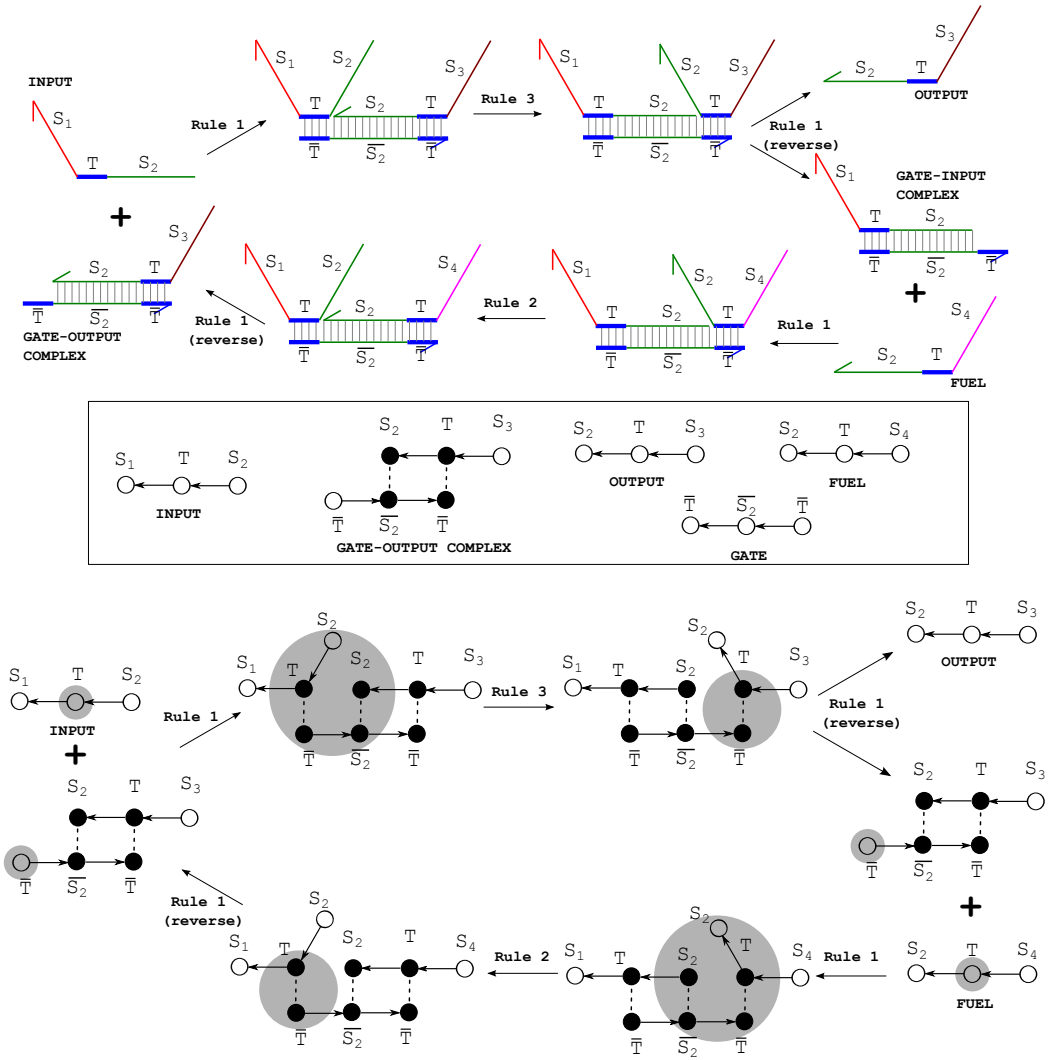


Qian, L., & Winfree, E. (n.d.). A simple DNA gate motif for synthesizing large-scale circuits. *rsif.royalsocietypublishing.org*

Example

Lulu Qian Winfree Seesaw Gate [9]

This example shows a sequence of rules that have been applied for the seesaw system, via which Qian and Winfree developed a circuit that can compute the square root of a fixed integer.

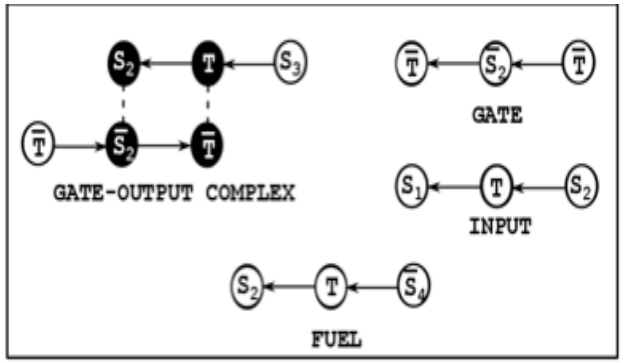


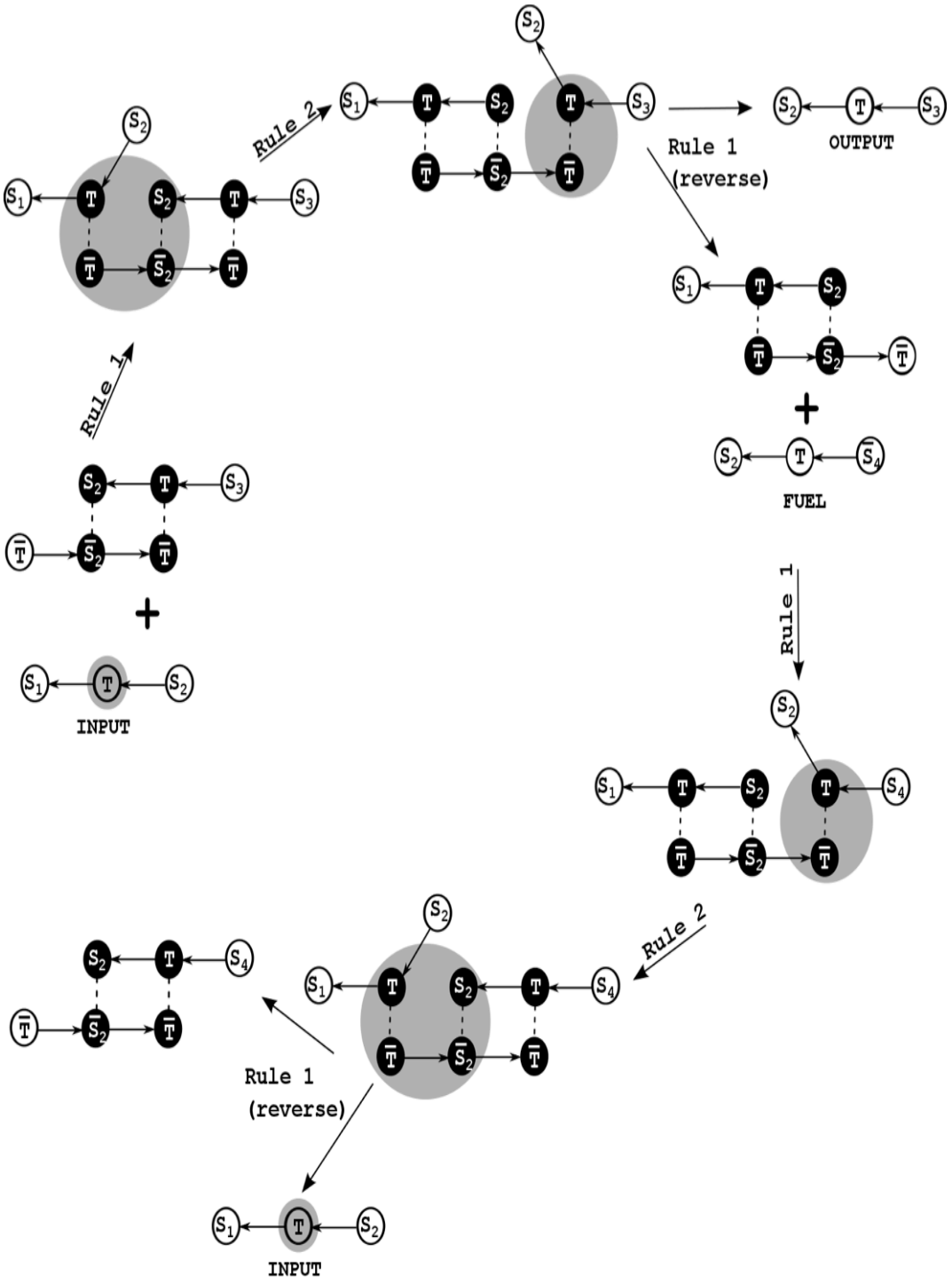
We use three di-graph grammar rules in the simulation of this complex.

Rule#1: Toehold Binding & Unbinding:
This rule is used each time a new strand gets added to the growing complex. Note that toehold unbinding is also used in the figure.

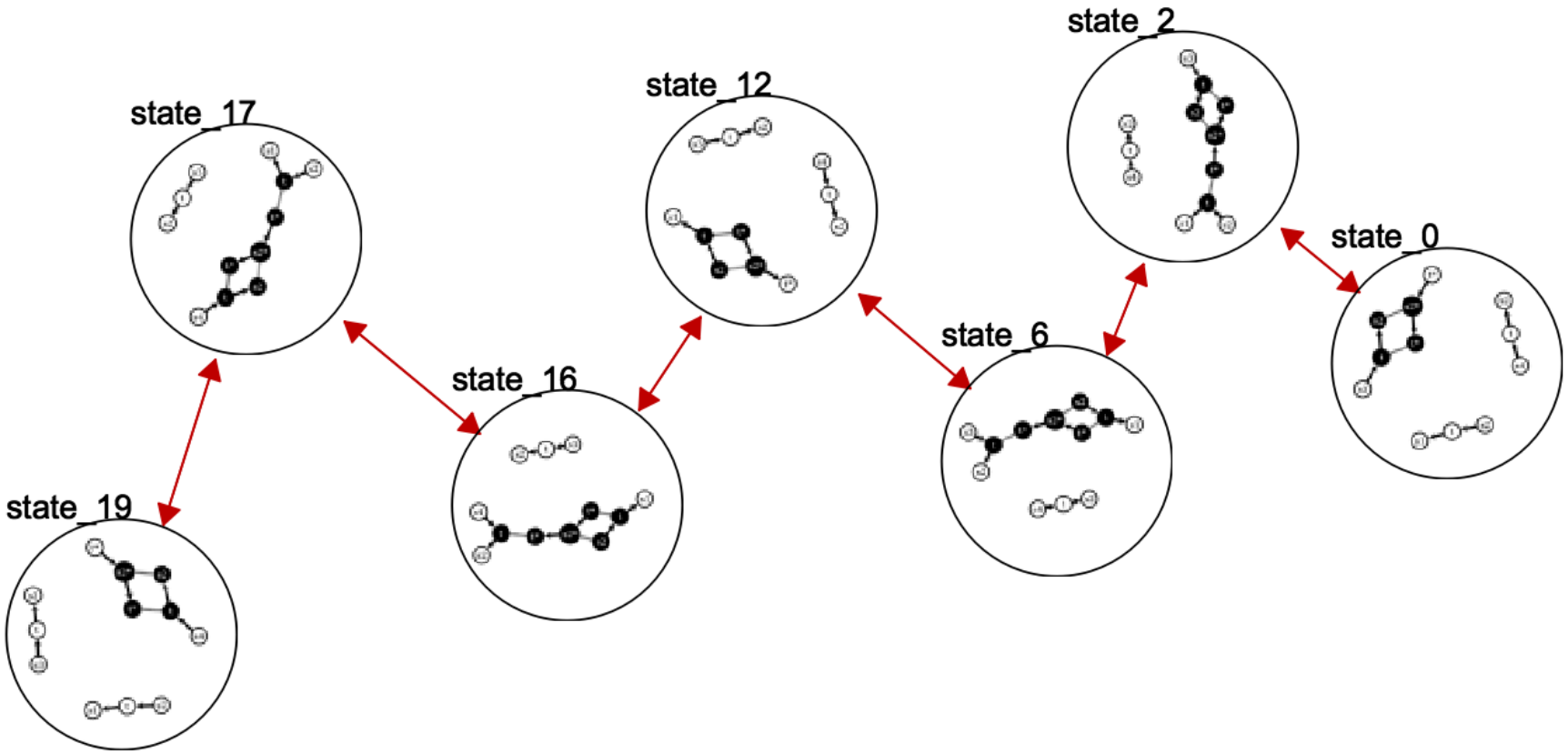
Rule #2 and #3: Strand Displacement takes place in both directions 5'-3' and vice versa.

Example 1: GRS Graph Transformations Modeling See-Saw Gates

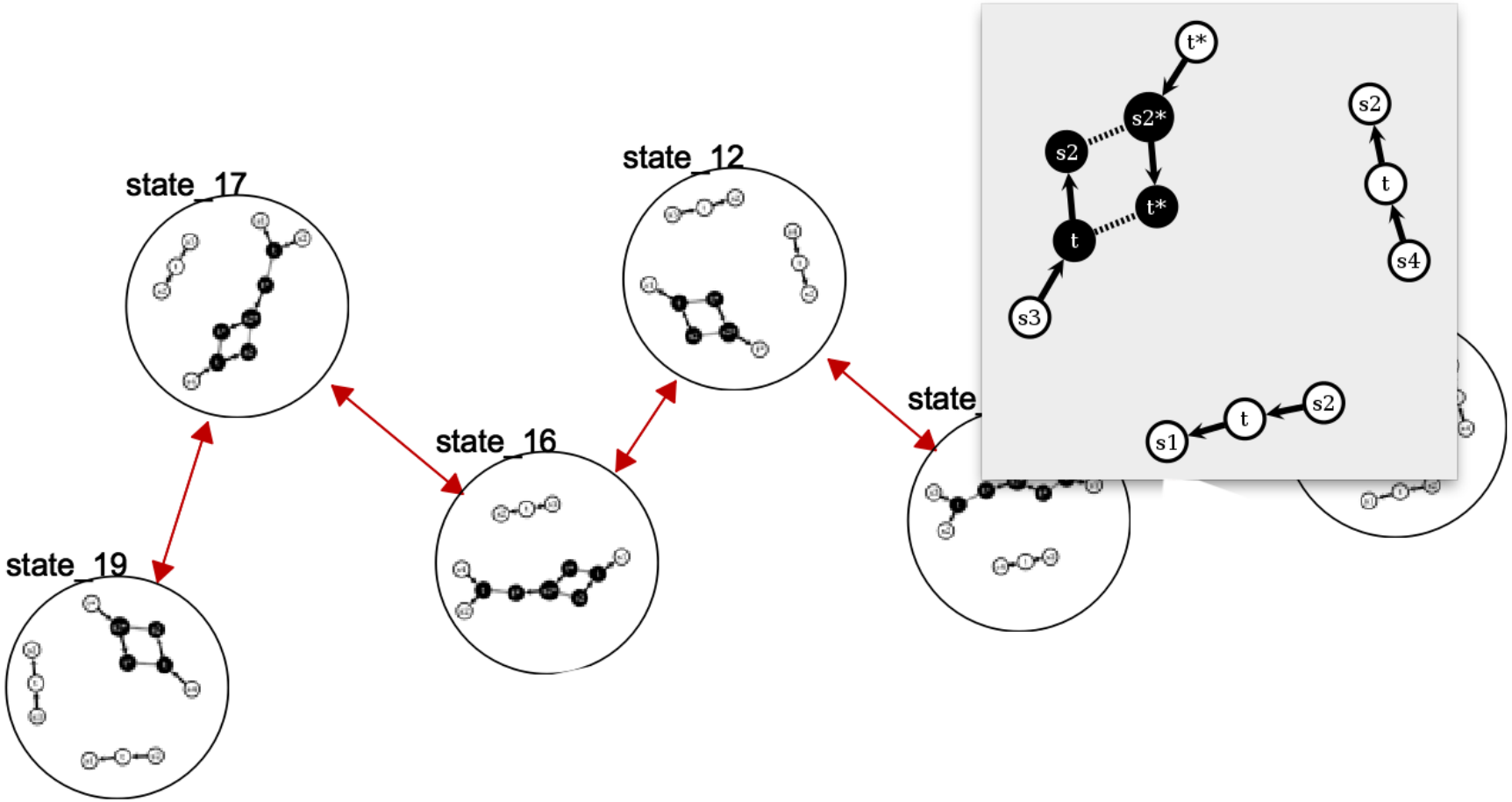




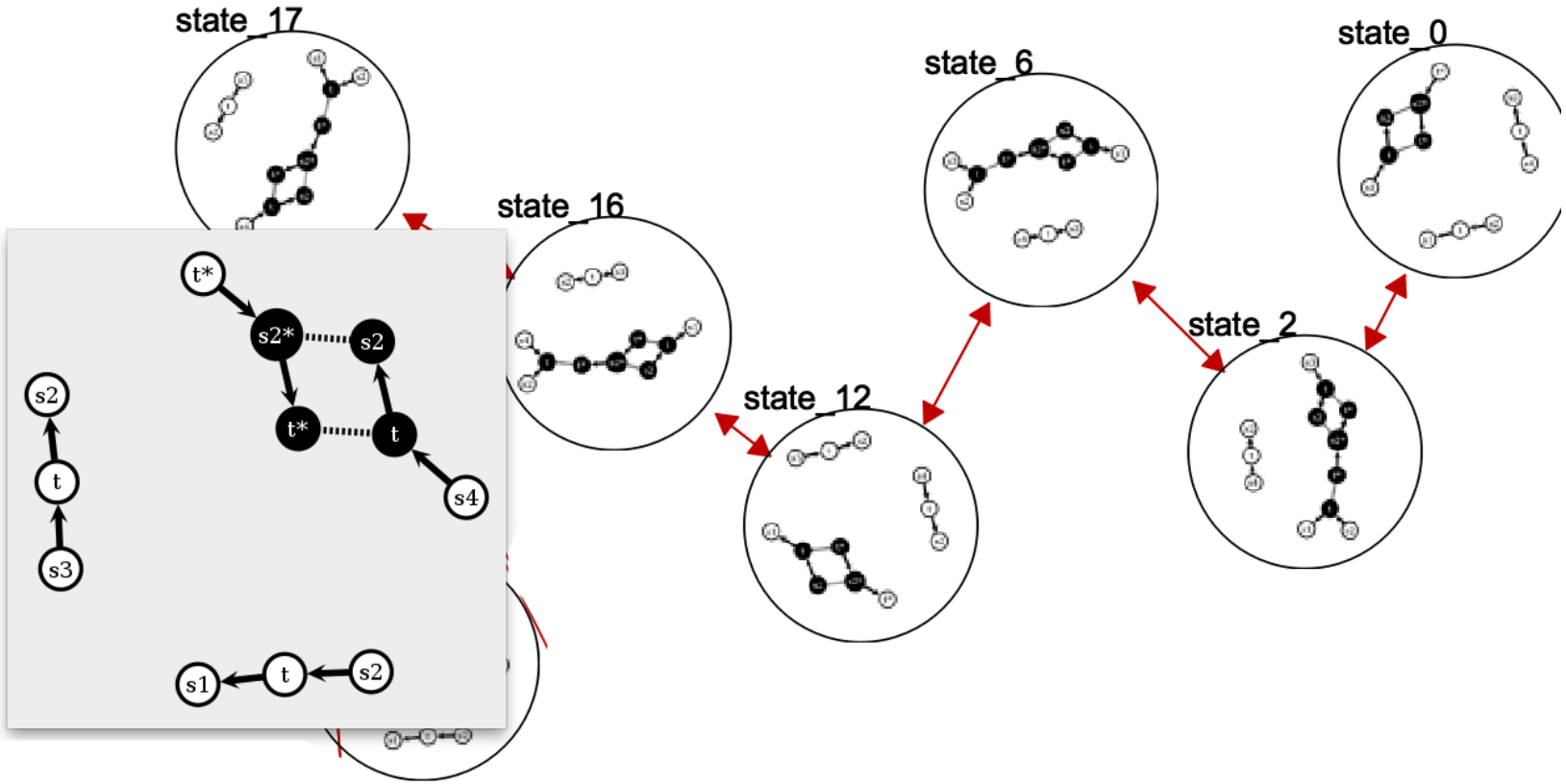
Example Path of GRS Graph Transformations

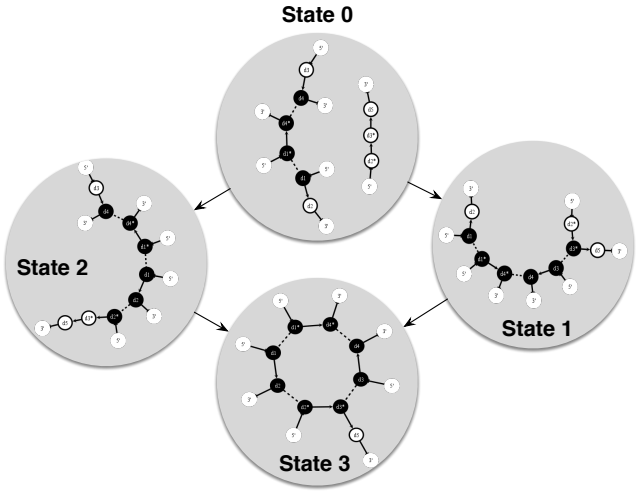


Example Path of GRS Graph Transformations (Cont)

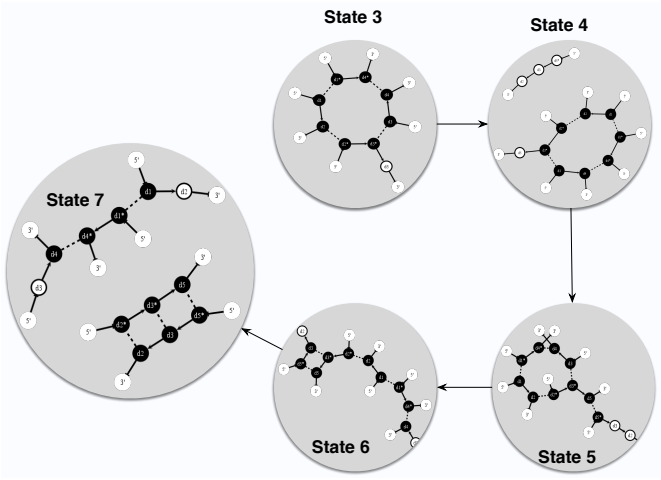


Example Path of GRS Graph Transformations (Cont)

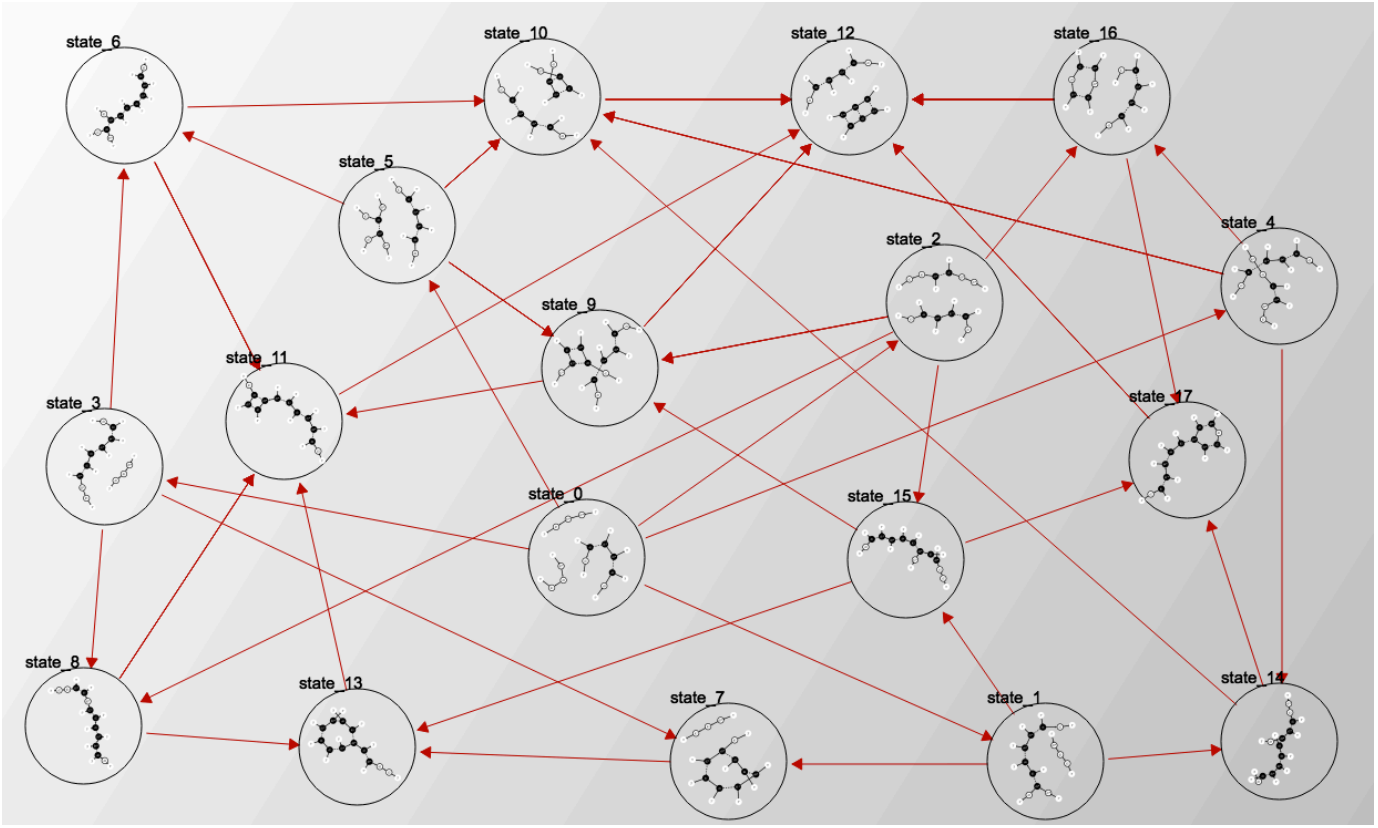



State 0
State 2
State 1
State 3

(a)

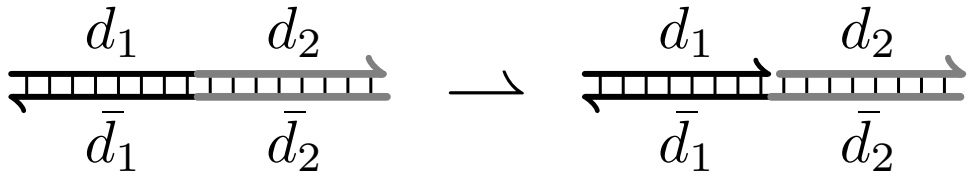

State 3
State 4
State 7
State 6
State 5

(b) .

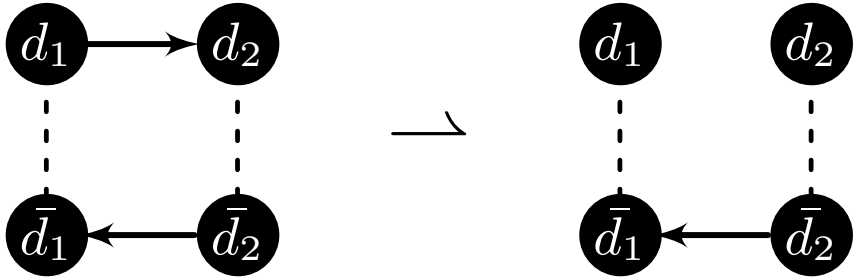

state_6
state_10
state_12
state_16
state_5
state_4
state_2
state_9
state_11
state_3
state_17
state_0
state_15
state_8
state_13
state_7
state_1
state_14

Rule #10: Restriction Enzyme Nicking

Cartoon Rendering

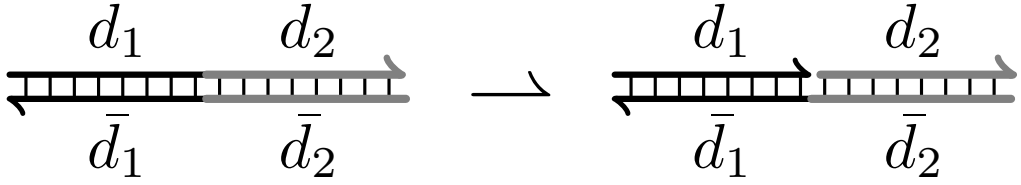


Rewrite Rule

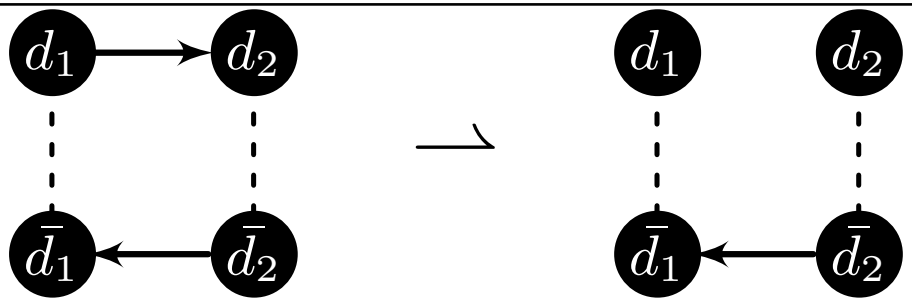


Rule #10: Restriction Enzyme Nicking

Cartoon Rendering

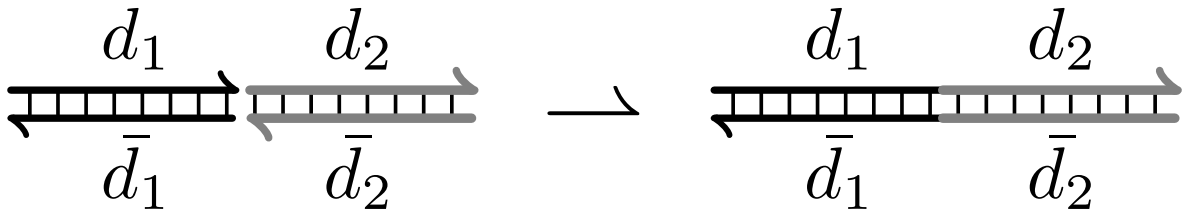


Rewrite Rule

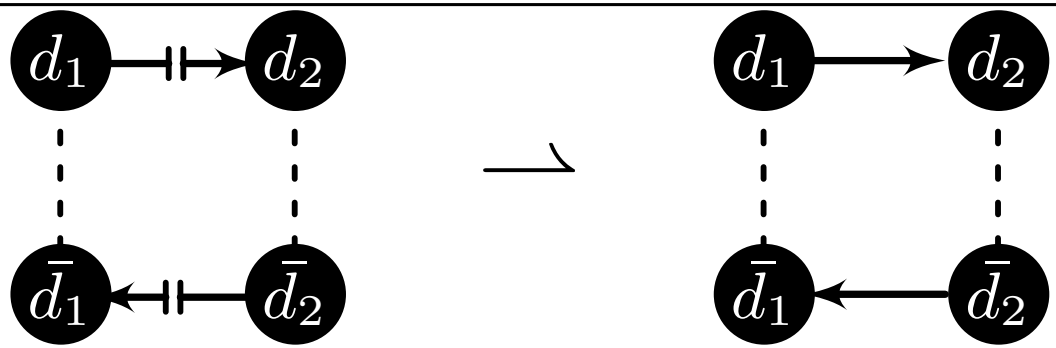


Rule #12: Blunt end Ligation

Cartoon Rendering

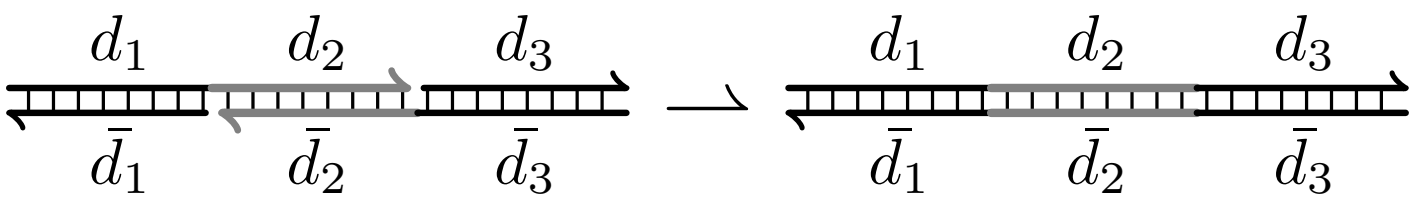


Rewrite Rule

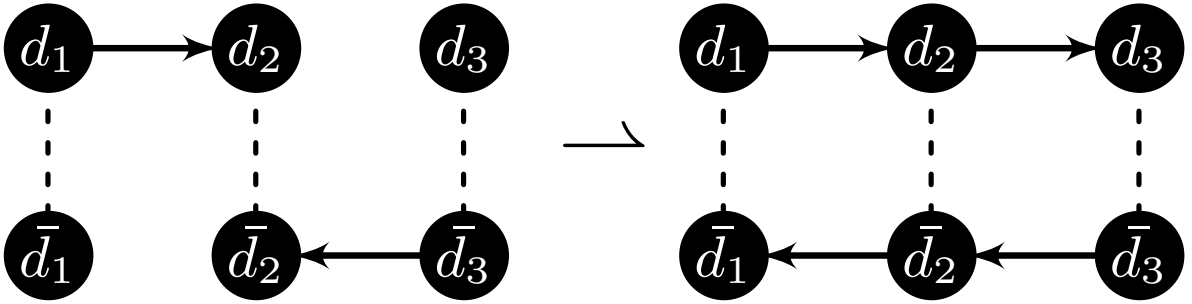


Rule #13: Sticky-end Ligation

Cartoon Rendering

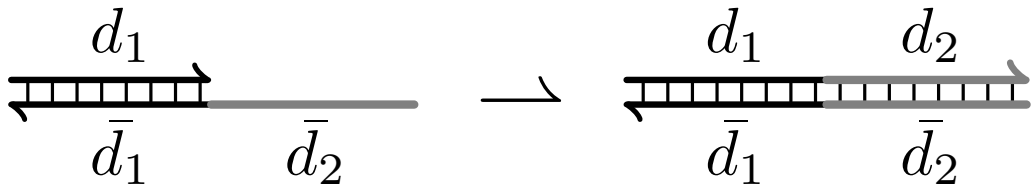


Rewrite Rule

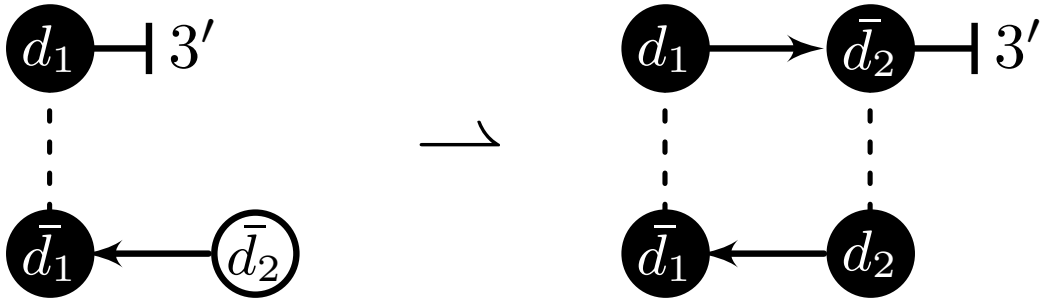


Rule #8: Polymerization

Cartoon Rendering



Rewrite Rule



d_1). The edge between vertex d_1 and the $3'$ vertex is removed, and replaced with an edge from vertex d_1.