

Modular verification of DNA strand displacement networks via serializability analysis (APPENDICES)

Matthew R. Lakin¹, Andrew Phillips², and Darko Stefanovic^{1,3}

¹ Department of Computer Science, University of New Mexico

² Biological Computation Group, Microsoft Research, Cambridge

³ Center for Biomedical Engineering, University of New Mexico

{mlakin,darko}@cs.unm.edu, andrew.phillips@microsoft.com

A Proof sketches for key results

A.1 Proof sketch for properties of trace rewriting

Proof (of Lemma 2). Assume that $\mathbf{S} \vdash_{\mathcal{C}} \tau \rightsquigarrow \tau'$. By definition we get that $\mathbf{S} \vdash_{\mathcal{C}} \tau$. If (CANCEL) was used to derive $\mathbf{S} \vdash_{\mathcal{C}} \tau \rightsquigarrow \tau'$ then τ has the form $\tau_1:\tau_2:\tau_3$ and τ' has the form $\tau_1:\tau_3$, and furthermore we know that $\mathbf{S} \xrightarrow{\tau_1} \mathbf{S}' \xrightarrow{\tau_2} \mathbf{S}'$. It follows immediately that $\mathbf{S} \vdash_{\mathcal{C}} \tau_1:\tau_3$, and hence that $\mathbf{S} \vdash_{\mathcal{C}} \tau'$ and $\text{final}_{\mathcal{C}}(\mathbf{S}, \tau) = \text{final}_{\mathcal{C}}(\mathbf{S}, \tau')$, as required. On the other hand, if (SWAP) was used to derive $\mathbf{S} \vdash_{\mathcal{C}} \tau \rightsquigarrow \tau'$ then τ has the form $\tau_1:\tau_2:\tau_3:\tau_4$ and τ' has the form $\tau_1:\tau_3:\tau_2:\tau_4$. Furthermore, we know that $\mathbf{S} \vdash_{\mathcal{C}} \tau_1:\tau_3:\tau_2$, and $\text{final}_{\mathcal{C}}(\mathbf{S}, \tau_1:\tau_3:\tau_2) = \text{final}_{\mathcal{C}}(\mathbf{S}, \tau_1:\tau_2:\tau_3)$. Thus it follows that $\mathbf{S} \vdash_{\mathcal{C}} \tau_1:\tau_3:\tau_2:\tau_4$, and hence that $\mathbf{S} \vdash_{\mathcal{C}} \tau'$ and $\text{final}_{\mathcal{C}}(\mathbf{S}, \tau) = \text{final}_{\mathcal{C}}(\mathbf{S}, \tau')$, as required. The cases for the remaining rules, (REFL) and (TRANS), are straightforward. \square

A.2 Proof sketch for serializability

Proof (of Lemma 4). We assume that $\mathbf{S} \vdash_{\mathcal{E}} \tau$, i.e., that

$$\mathbf{S} \vdash_{\mathcal{E}} \tau_1:[r_1^\alpha]:\dots:\tau_{k-1}:[r_{k-1}^\alpha]:\tau_k:[r_{com}^\alpha]:\tau_{k+1}:[r_{k+1}^\alpha]:\dots:\tau_n:[r_n^\alpha]:\tau_{rest}.$$

Since r_{com}^α is the first commit reaction in τ , all reactions prior to it must be reversible. Also, we note that the only species which can be shared between encodings are those of which the encodings are copy tolerant, and, because of the stratified CRN property, a single reaction encoding cannot consume more fuel or formal species than is necessary to complete an execution.

By assumption, we know that $\mathbf{S} \vdash_{\mathcal{E}} \tau_1:[r_1^\alpha]$ and $\mathbf{S} \vdash_{\mathcal{E}} [r_1^\alpha]$. We must show that there exists τ_1' such that $\mathbf{S} \vdash_{\mathcal{E}} \tau_1:[r_1^\alpha] \rightsquigarrow [r_1^\alpha]:\tau_1'$. We observe that τ_1 and $[r_1^\alpha]$ may conflict if τ_1 consumes n copies of some species x that is a reactant of r_1^α , and subsequently returns at least one copy of x to the environment. In this case, the minimum amount of x that could be present in the initial state is n , as the copies of x returned to the environment by τ_1 can function as a reactant

of r_1^α . Since τ_1 is a trace of pre-commit reactions, in which formal species are consumed by forward steps and re-emitted by backward steps, it follows from Definition 14 that the encoding must revisit one of its previous states in order to return a species to the environment. Therefore we can derive $\mathbf{S} \vdash_{\mathcal{E}} \tau_1 \rightsquigarrow \tau'_1$, for some τ'_1 which contains no backward steps. It follows that \mathbf{S} contains sufficient species to execute both τ'_1 and r_1^α , and hence it follows that

$$\mathbf{S} \vdash_{\mathcal{E}} \tau \rightsquigarrow [r_1^\alpha]:\tau'_1:\tau_2:[r_2^\alpha]:\dots:\tau_{k-1}:[r_{k-1}^\alpha]:\tau_k:[r_{com}^\alpha]:\tau_{k+1}:[r_{k+1}^\alpha]:\dots:\tau_n:[r_n^\alpha]:\tau_{rest}.$$

By repeated application of this argument we get that

$$\mathbf{S} \vdash_{\mathcal{E}} \tau \rightsquigarrow [r_1^\alpha, \dots, r_{k-1}^\alpha, r_{com}^\alpha]:\tau':\tau_{k+1}:[r_{k+1}^\alpha]:\dots:\tau_n:[r_n^\alpha]:\tau_{rest},$$

where $\tau' = \tau'_1:\dots:\tau'_k$.

Now, r_n^α is the last reaction in a complete execution of $[[\alpha]]$, which means that it must be a forward step. By Definition 14, forward steps after the commit reaction only have fuels and intermediates as reactants, and by Definition 15 we know that fuel and intermediate strands could potentially be shared with other reaction encodings. However, by Definition 13 we know that if any reaction in τ_n consumes a reactant x of r_n^α , then that reaction must be from a copy of reaction encoding which is in a state where it could consume x anyway, and hence that copy must have its own copy of x present anyway. Hence, since we know that r_n^α can be executed directly after r_{n-1}^α , we can move r_n^α forward in the reaction trace, to get

$$\mathbf{S} \vdash_{\mathcal{E}} \tau \rightsquigarrow [r_1^\alpha, \dots, r_{k-1}^\alpha, r_{com}^\alpha]:\tau':\tau_{k+1}:[r_{k+1}^\alpha]:\dots:\tau_{n-1}:[r_{n-1}^\alpha, r_n^\alpha]:\tau_n:\tau_{rest}.$$

We now consider the subtrace $[r_{n-1}^\alpha, r_n^\alpha]$. In general, this trace could contain both forward and backward steps in the $[[\alpha]]$ encoding, which means that these reactions could also consume reactants which correspond to formal species. This has the potential to cause a conflict if some reaction in τ_{n-1} also consumed that species. However, by Definition 14 we know that a formal species can only be consumed by a post-commit reaction if that reaction is a backward step, and this means that by the end of the subtrace $[r_{n-1}^\alpha, r_n^\alpha]$, any formal species consumed by a backward step will eventually be re-generated by a later reaction in the subtrace. Hence, τ_{n-1} and $[r_{n-1}^\alpha, r_n^\alpha]$ can be swapped using the (SWAP) from Definition 11, and we get that

$$\mathbf{S} \vdash_{\mathcal{E}} \tau \rightsquigarrow [r_1^\alpha, \dots, r_{k-1}^\alpha, r_{com}^\alpha]:\tau':\tau_{k+1}:[r_{k+1}^\alpha]:\dots:\tau_{n-2}:[r_{n-2}^\alpha, r_{n-1}^\alpha, r_n^\alpha]:\tau_{n-1}:\tau_n:\tau_{rest}.$$

By repeated application of this argument we get that

$$\mathbf{S} \vdash_{\mathcal{E}} \tau \rightsquigarrow [r_1^\alpha, \dots, r_{k-1}^\alpha, r_{com}^\alpha, r_{k+1}^\alpha, \dots, r_n^\alpha]:\tau'_{rest},$$

as required, where $\tau'_{rest} = \tau':\tau_{k+1}:\dots:\tau_n:\tau_{rest}$, i.e., $\tau'_{rest} = \tau'_1:\dots:\tau'_k:\tau_{k+1}:\dots:\tau_n:\tau_{rest}$. \square

A.3 Proof sketch for soundness

Proof (of Theorem 2). We begin by recalling that, by Definition 15 there can be no direct interaction between species which do not appear in the same reaction encoding, and that the only species which are shared are those of which the encodings are copy tolerant. Furthermore, a single copy of a given reaction encoding must be copy tolerant to any formal or fuel species that it uses. Thus, the trace τ must consist of an interleaving of reactions which all correspond to a valid state transition from precisely one of the constituent reaction encodings. Furthermore, since there is only a finite supply of fuel, there can only be finitely many commit reactions in τ . However, τ could include infinite cycles of reversible reactions—we assume that τ has already been rewritten to eliminate these. We proceed by induction on $\chi(\tau)$, the number of commit reactions in τ .

Base case: $\chi(\tau) = 0$. Since there are no commit reactions, all reactions in τ must be reversible. Therefore, by Lemma 3 there exists a trace τ'' such that $(\mathcal{M}(\mathbf{X}) + \mathbf{F}) \vdash_{\mathcal{E}} \tau:\tau'' \rightsquigarrow \epsilon$. Then, if $pt(\mathbf{X}, \mathbf{F}) = \emptyset$ we can set $\tau' = \tau''$ to get that $(\mathcal{M}(\mathbf{X}) + \mathbf{F}) \vdash_{\mathcal{E}} \tau:\tau' \rightsquigarrow \epsilon$, as required. On the other hand, if $pt(\mathbf{X}, \mathbf{F}) \neq \emptyset$ then there must be a formal reaction α such that $\mathbf{X} \vdash_{\mathcal{E}} \alpha$ and for which $reqfuel([\alpha]) \leq \mathbf{F}$. Then, for a complete execution τ_{α} of $[[\alpha]]$, we can set $\tau' = \tau'':\tau_{\alpha}$ to get $(\mathcal{M}(\mathbf{X}) + \mathbf{F}) \vdash_{\mathcal{E}} \tau:\tau' \rightsquigarrow \tau_{\alpha}$, which is a serial execution of $[\alpha] \in pt(\mathbf{X}, \mathbf{F})$, as required.

Inductive case: $\chi(\tau) = n + 1$. Suppose that r_{com}^{α} is the first commit reaction in τ , belonging to the encoding of the formal reaction $\alpha = (\mathbf{R}_{\alpha} \rightarrow \mathbf{P}_{\alpha})$. By Definition 14, in order to pass the commit reaction we must consume all of the species from $\mathcal{M}(\mathbf{R}_{\alpha})$, and hence it follows that $\mathbf{R}_{\alpha} \leq \mathbf{X}$. Furthermore, since we are only considering initial states with fuel to run complete encodings, we get that $[\alpha] \in pt(\mathbf{X}, \mathbf{F})$ and hence $pt(\mathbf{X}, \mathbf{F}) \neq \emptyset$.

We proceed by identifying those reactions from τ which precede and follow r_{com}^{α} to make up a trace from $[[\alpha]]$. If τ does not contain a full execution of $[[\alpha]]$, let τ'_a denote a trace which completes an execution of $[[\alpha]]$. Since $[\alpha] \in pt(\mathbf{X}, \mathbf{F})$ we know that $(\mathcal{M}(\mathbf{X}) + \mathbf{F}) \vdash_{\mathcal{E}} r_1^{\alpha}$ holds, where r_1^{α} is the first reaction of the execution of $[[\alpha]]$. Then, by Lemma 4 we get that $(\mathcal{M}(\mathbf{X}) + \mathbf{F}) \vdash_{\mathcal{E}} \tau:\tau'_a \rightsquigarrow \tau_{\alpha}:\tau_{rest}$, where τ_{α} is an execution of $[[\alpha]]$.

Let \mathbf{X}' be a formal state such that $\mathbf{X} \xrightarrow{\alpha} \mathbf{X}'$. Then, we get that $(\mathcal{M}(\mathbf{X}) + \mathbf{F}) \xrightarrow{\tau_{\alpha}} (\mathcal{M}(\mathbf{X}') + (\mathbf{F} - \mathbf{F}_{\alpha}) + \mathbf{L}_{\alpha})$, where \mathbf{L}_{α} is the non-formal species which remain after the execution of $[[\alpha]]$ has completed. By Lemma 2 we can deduce that $(\mathcal{M}(\mathbf{X}') + (\mathbf{F} - \mathbf{F}_{\alpha}) + \mathbf{L}_{\alpha}) \vdash_{\mathcal{E}} \tau_{rest}$. All reaction encodings must be copy tolerant of the species from \mathbf{L}_{α} , which means that the reactions in τ_{rest} may occur with or without the additional leftover species from \mathbf{L}_{α} . It follows that $(\mathcal{M}(\mathbf{X}') + (\mathbf{F} - \mathbf{F}_{\alpha})) \vdash_{\mathcal{E}} \tau_{rest}$, and we write $\mathbf{S}_{\alpha} = \mathcal{M}(\mathbf{X}') + (\mathbf{F} - \mathbf{F}_{\alpha})$.

Since $\chi(\tau_{rest}) = n = \chi(\tau) - 1$, we can invoke our induction hypothesis on τ_{rest} . Now, we perform a case split on whether $pt(\mathbf{X}', \mathbf{F} - \mathbf{F}_{\alpha})$ is empty. If $pt(\mathbf{X}', \mathbf{F} - \mathbf{F}_{\alpha}) = \emptyset$, by induction there exists τ'_b such that $\mathbf{S}_{\alpha} \vdash_{\mathcal{E}} \tau_{rest}:\tau'_b \rightsquigarrow \epsilon$, and

hence that $\mathbf{S}_\alpha + \mathbf{L}_\alpha \vdash_{\mathcal{E}} \tau_{rest}:\tau'_b \rightsquigarrow \epsilon$. It follows that $(\mathcal{M}(\mathbf{X}) + \mathbf{F}) \vdash_{\mathcal{E}} \tau:(\tau'_a:\tau'_b) \rightsquigarrow \tau_\alpha$, and since τ_α is a serial execution of $[\alpha] \in pt(\mathbf{X}, \mathbf{F})$, we get the result.

If $pt(\mathbf{X}', \mathbf{F} - \mathbf{F}_\alpha) \neq \emptyset$, by induction there exists τ'_b such that $\mathbf{S}_\alpha \vdash_{\mathcal{E}} \tau_{rest}:\tau'_b \rightsquigarrow \tau_{ser}$, where τ_{ser} is a serial execution of some formal trace $\tau_{formal} \in pt(\mathbf{X}', \mathbf{F} - \mathbf{F}_\alpha)$. From this we get $\mathbf{S}_\alpha + \mathbf{L}_\alpha \vdash_{\mathcal{E}} \tau_{rest}:\tau'_b \rightsquigarrow \tau_{ser}$, and hence that $(\mathcal{M}(\mathbf{X}) + \mathbf{F}) \vdash_{\mathcal{E}} \tau:(\tau'_a:\tau'_b) \rightsquigarrow \tau_\alpha:\tau_{ser}$. Since $\tau_{formal} \in pt(\mathbf{X}', \mathbf{F} - \mathbf{F}_\alpha)$ it follows that $[\alpha]:\tau_{formal} \in pt(\mathbf{X}, \mathbf{F})$, and since $\tau_\alpha:\tau_{ser}$ is a serial execution of $[\alpha]:\tau_{formal}$, we get the result. \square

B Further details of verification example

B.1 Details of catalyst gate

The following is the DSD code for the catalyst gate module, together with an initial state which instantiates it to implement the reaction $x + y \rightarrow y + z$:

```
(* DSD code for two-domain catalyst gate. *)
(* Use with Infinite DSD semantics. *)

(* Define a global toehold *)
new t

(* Catalyst gate module, x + y -> x + z *)
def C(N,x,y,z) = new a new c
  ( N * {t^*}[x t^]:[y t^]:[c]:[a t^]:[a]
  | N * [x]:[t^ z]:[c]:[t^ y]:[t^ a]{t^*}
  | N * <t^ c a>
  | N * <z c t^> )

(* Example initial state *)
( C(1,x,y,z) | <t^ x> | <t^ y> )
```

We used the *Infinite* semantics option in the DSD compiler to generate the full sets of reactions and species generated from this initial state. These are presented in Figure S1 and Figure S2 respectively.

In order to verify that this encoding satisfies the criteria from Definition 14, we must investigate the state space starting from the initial state defined above. We can use the DSD compiler to show that this system does indeed have a universally reachable terminal state—the initial and terminal states are illustrated in Figure S3. A summary of the full state space is presented in Figure S4. The initial and terminal states are drawn in full, and for the other states only the formal species are drawn, to save space. From this we see that every terminal trace from the initial state has a commit reaction (in fact, this is the same reaction in all cases) which satisfies the criteria from Definition 14, therefore the encoding does indeed satisfy our correctness criteria.

B.2 Details of approximate majority example

As discussed above, the approximate majority example circuit consists of the following four chemical reactions.



We implement this system by instantiating the catalyst gate from the previous section to encode each of the four chemical reactions. To verify this system, we must first compile each of these reaction gates separately and analyze their state spaces so we can check that they all satisfy the requirements from Definition 14. For this we use the DSD compiler to instantiate the catalyst gate from Section B.1 for each reaction in turn: the corresponding sets of species are presented in Figure S5 and Figure S6. The a and c domains from the module definition are instantiated with a different domain in each case, which we denote by $a.1$, $a.2$, $c.1$, $c.2$, etc., as is the convention in DSD syntax. This ensures that the gate strands are unique between the different encodings. In each case, the CRN graph and the state space follows the template of those presented above, from which we conclude that the encodings all satisfy Definition 14.

It remains to see that the four reaction encodings are pairwise compatible in the sense of Definition 15, and for this we again we use the DSD compiler to check that no interactions are possible between species that never appear in the same reaction encoding. It is not hard to check that the only shared species between these gates are waste strands, strands which correspond to formal species and certain intermediate strands. We must show that the encodings are copy tolerant of these shared strands, and we use the DSD compiler to achieve this by adding an copy of these species to the starting state of the relevant reaction encodings and verifying that the state space is identical in each case (modulo the additional copies of those species in every state). Adding a single copy suffices because none of the two-domain reactions from Figure 1 require more than one of a particular reactant, so just one additional copy will be enough to show up any additional reactions enabled by extra copies of this species. Hence, by Theorem 2 it follows that any trace generated by these four reaction encodings can be rewritten to produce a serial trace which corresponds to a valid execution of the underlying formal reactions.

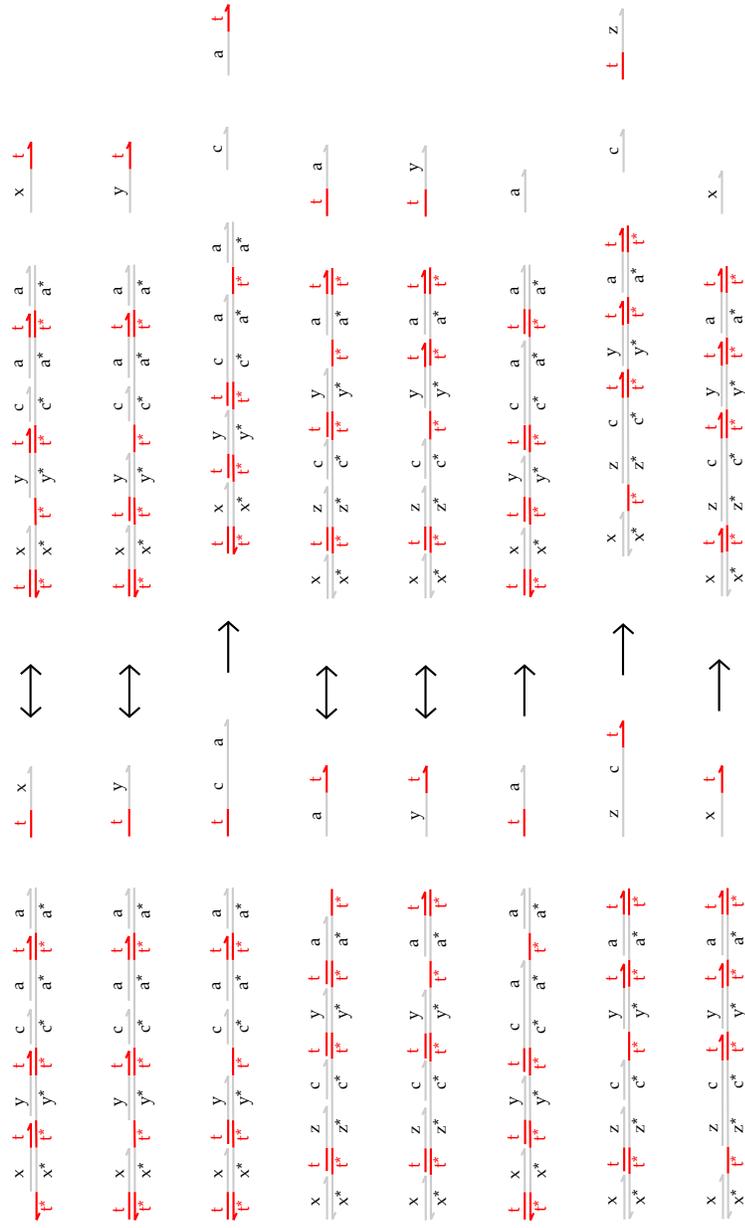


Fig. S1. Catalyst gate reactions, from an encoding of the reaction $x + y \rightarrow y + z$.

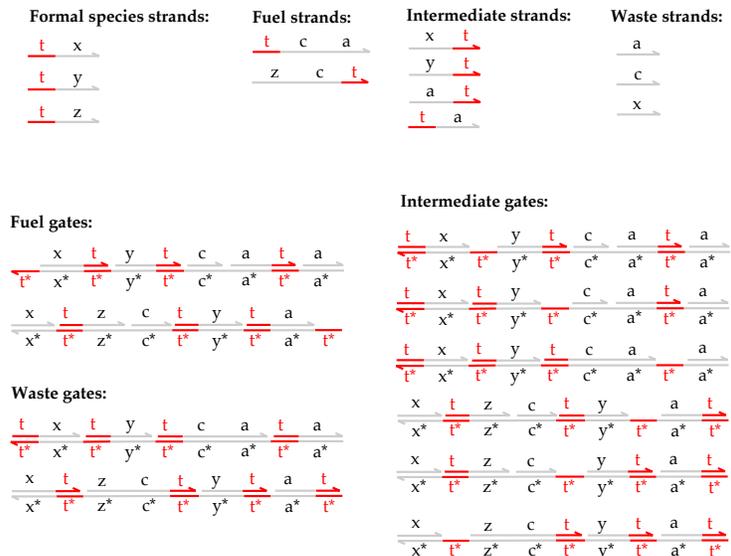


Fig. S2. Catalyst gate species, from an encoding of the reaction $x + y \rightarrow y + z$.

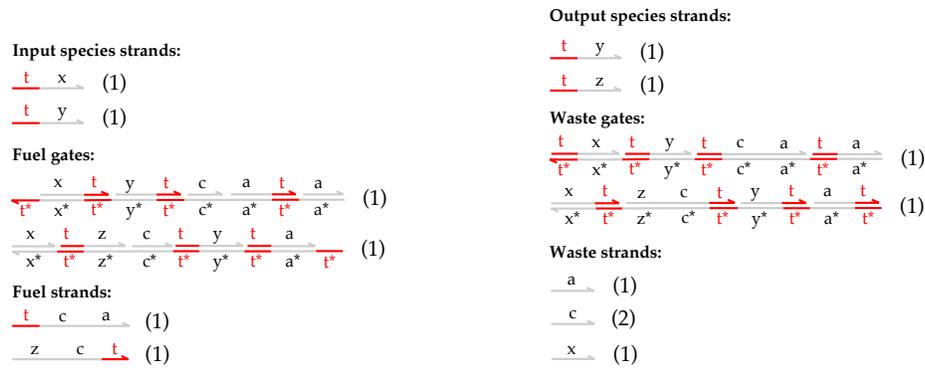


Fig. S3. Initial (left) and terminal (right) states for the catalyst gate which encodes the reaction $x + y \rightarrow y + z$.

1. $x + y \rightarrow y + b$		2. $x + y \rightarrow x + b$	
$\frac{x}{x^*} \frac{t}{t^*} \frac{y}{y^*} \frac{c.1}{c.1^*} \frac{a.1}{a.1^*} \frac{a.1}{a.1^*}$	$\frac{t}{t^*} \frac{c.1}{c.1^*} \frac{a.1}{a.1^*}$	$\frac{y}{y^*} \frac{t}{t^*} \frac{x}{x^*} \frac{c.2}{c.2^*} \frac{a.2}{a.2^*} \frac{a.2}{a.2^*}$	$\frac{t}{t^*} \frac{c.2}{c.2^*} \frac{a.2}{a.2^*}$
$\frac{x}{x^*} \frac{t}{t^*} \frac{b}{b^*} \frac{c.1}{c.1^*} \frac{t}{t^*} \frac{y}{y^*} \frac{a.1}{a.1^*} \frac{a.1}{a.1^*}$	$\frac{b}{b^*} \frac{c.1}{c.1^*} \frac{t}{t^*}$	$\frac{y}{y^*} \frac{t}{t^*} \frac{b}{b^*} \frac{c.2}{c.2^*} \frac{t}{t^*} \frac{x}{x^*} \frac{a.2}{a.2^*} \frac{a.2}{a.2^*}$	$\frac{b}{b^*} \frac{c.2}{c.2^*} \frac{t}{t^*}$
$\frac{t}{t^*} \frac{x}{x^*} \frac{y}{y^*} \frac{c.1}{c.1^*} \frac{a.1}{a.1^*} \frac{t}{t^*} \frac{a.1}{a.1^*}$	$\frac{t}{t^*} \frac{x}{x^*}$	$\frac{t}{t^*} \frac{y}{y^*} \frac{x}{x^*} \frac{c.2}{c.2^*} \frac{a.2}{a.2^*} \frac{t}{t^*} \frac{a.2}{a.2^*}$	$\frac{t}{t^*} \frac{y}{y^*}$
$\frac{t}{t^*} \frac{x}{x^*} \frac{t}{t^*} \frac{y}{y^*} \frac{c.1}{c.1^*} \frac{a.1}{a.1^*} \frac{t}{t^*} \frac{a.1}{a.1^*}$	$\frac{x}{x^*} \frac{t}{t^*}$	$\frac{t}{t^*} \frac{y}{y^*} \frac{t}{t^*} \frac{x}{x^*} \frac{c.2}{c.2^*} \frac{a.2}{a.2^*} \frac{t}{t^*} \frac{a.2}{a.2^*}$	$\frac{t}{t^*} \frac{x}{x^*}$
$\frac{t}{t^*} \frac{x}{x^*} \frac{t}{t^*} \frac{y}{y^*} \frac{c.1}{c.1^*} \frac{a.1}{a.1^*} \frac{a.1}{a.1^*}$	$\frac{t}{t^*} \frac{y}{y^*}$	$\frac{t}{t^*} \frac{y}{y^*} \frac{t}{t^*} \frac{x}{x^*} \frac{c.2}{c.2^*} \frac{a.2}{a.2^*} \frac{a.2}{a.2^*}$	$\frac{y}{y^*} \frac{t}{t^*}$
$\frac{x}{x^*} \frac{t}{t^*} \frac{b}{b^*} \frac{c.1}{c.1^*} \frac{t}{t^*} \frac{y}{y^*} \frac{a.1}{a.1^*} \frac{t}{t^*}$	$\frac{c.1}{c.1^*}$	$\frac{y}{y^*} \frac{t}{t^*} \frac{b}{b^*} \frac{c.2}{c.2^*} \frac{t}{t^*} \frac{x}{x^*} \frac{a.2}{a.2^*} \frac{t}{t^*}$	$\frac{x}{x^*} \frac{t}{t^*}$
$\frac{t}{t^*} \frac{x}{x^*} \frac{t}{t^*} \frac{y}{y^*} \frac{c.1}{c.1^*} \frac{a.1}{a.1^*} \frac{t}{t^*} \frac{a.1}{a.1^*}$	$\frac{a.1}{a.1^*} \frac{t}{t^*}$	$\frac{y}{y^*} \frac{t}{t^*} \frac{b}{b^*} \frac{c.2}{c.2^*} \frac{x}{x^*} \frac{t}{t^*} \frac{a.2}{a.2^*} \frac{t}{t^*}$	$\frac{c.2}{c.2^*}$
$\frac{x}{x^*} \frac{t}{t^*} \frac{b}{b^*} \frac{c.1}{c.1^*} \frac{y}{y^*} \frac{t}{t^*} \frac{a.1}{a.1^*} \frac{t}{t^*}$	$\frac{t}{t^*} \frac{a.1}{a.1^*}$	$\frac{t}{t^*} \frac{y}{y^*} \frac{t}{t^*} \frac{x}{x^*} \frac{c.2}{c.2^*} \frac{a.2}{a.2^*} \frac{t}{t^*} \frac{a.2}{a.2^*}$	$\frac{a.2}{a.2^*} \frac{t}{t^*}$
$\frac{x}{x^*} \frac{b}{b^*} \frac{c.1}{c.1^*} \frac{t}{t^*} \frac{y}{y^*} \frac{a.1}{a.1^*} \frac{t}{t^*}$	$\frac{a.1}{a.1^*}$	$\frac{y}{y^*} \frac{t}{t^*} \frac{b}{b^*} \frac{c.2}{c.2^*} \frac{x}{x^*} \frac{t}{t^*} \frac{a.2}{a.2^*} \frac{t}{t^*}$	$\frac{t}{t^*} \frac{a.2}{a.2^*}$
$\frac{x}{x^*} \frac{t}{t^*} \frac{b}{b^*} \frac{c.1}{c.1^*} \frac{t}{t^*} \frac{y}{y^*} \frac{a.1}{a.1^*} \frac{t}{t^*}$	$\frac{y}{y^*} \frac{t}{t^*}$	$\frac{y}{y^*} \frac{t}{t^*} \frac{b}{b^*} \frac{c.2}{c.2^*} \frac{x}{x^*} \frac{t}{t^*} \frac{a.2}{a.2^*} \frac{t}{t^*}$	$\frac{a.2}{a.2^*}$
	$\frac{x}{x^*}$		$\frac{t}{t^*} \frac{b}{b^*}$
	$\frac{t}{t^*} \frac{b}{b^*}$		$\frac{y}{y^*}$

Fig. S5. Species for catalyst gates which implement reactions 1 and 2 from the approximate majority circuit.

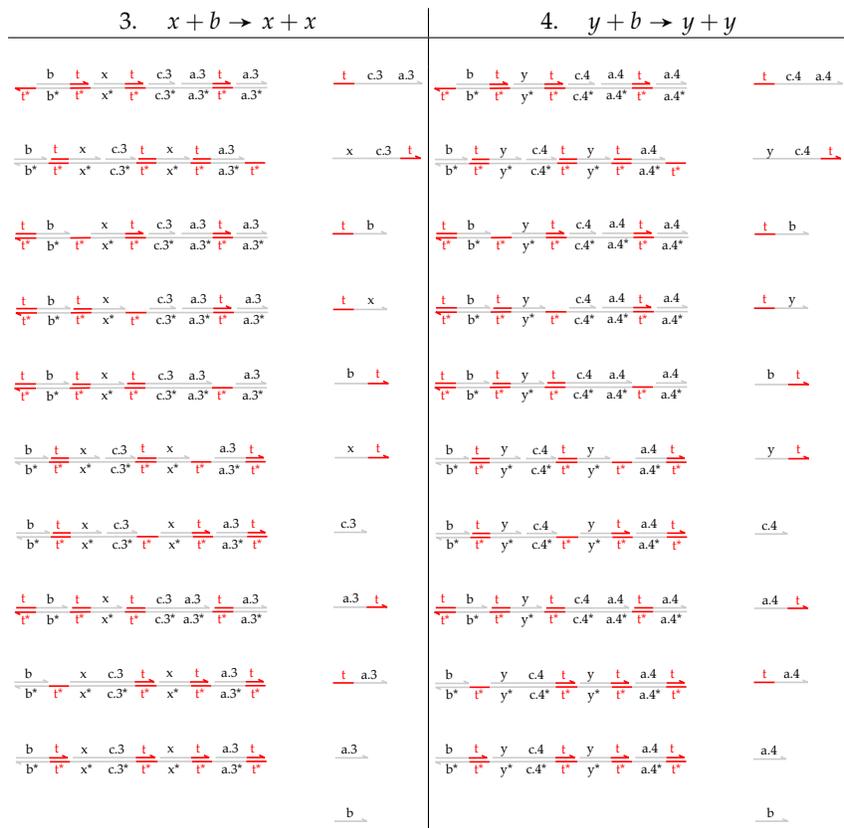


Fig. S6. Species for catalyst gates which implement reactions 3 and 4 from the approximate majority circuit.