Space- and energy-efficient computing with DNA

Anne Condon, U. British Columbia

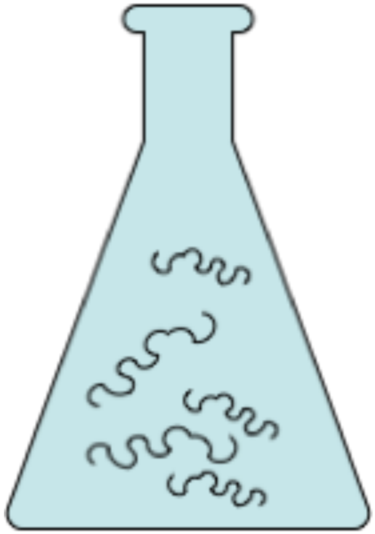
Motivation: Programming molecules

- Computing technologies need not be limited to silicon!
- Nature provides an incredible, nanoscale, molecular toolkit
- DNA molecules are particularly nice to work with

- Potential applications: nanocircuits, DNA storage, facile disease diagnosis, smart drug delivery, and of course, understanding our world

- Molecular programming also raises new theoretical questions, pertaining to models of computation, information encoding, error correction, distributed computing, randomized algorithms, and more

Motivation: Programming molecules

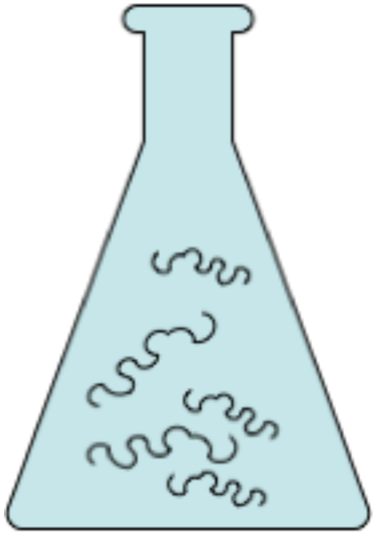


Motivation: Programming molecules

molecular program, as a Chemical Reaction Network (CRN)

$X + X + Y \rightarrow X + X + X$

$X + Y + Y \rightarrow Y + Y + Y$

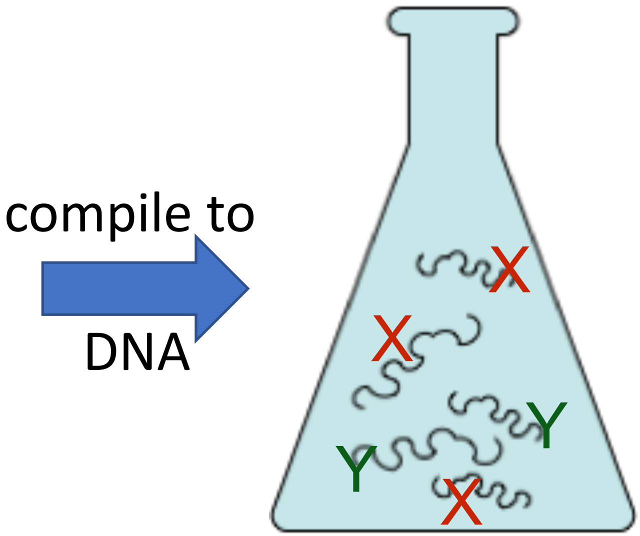


Motivation: Programming molecules

molecular program, as a Chemical Reaction Network (CRN)

$X + X + Y \rightarrow X + X + X$

$X + Y + Y \rightarrow Y + Y + Y$



run the program from
some initial configuration

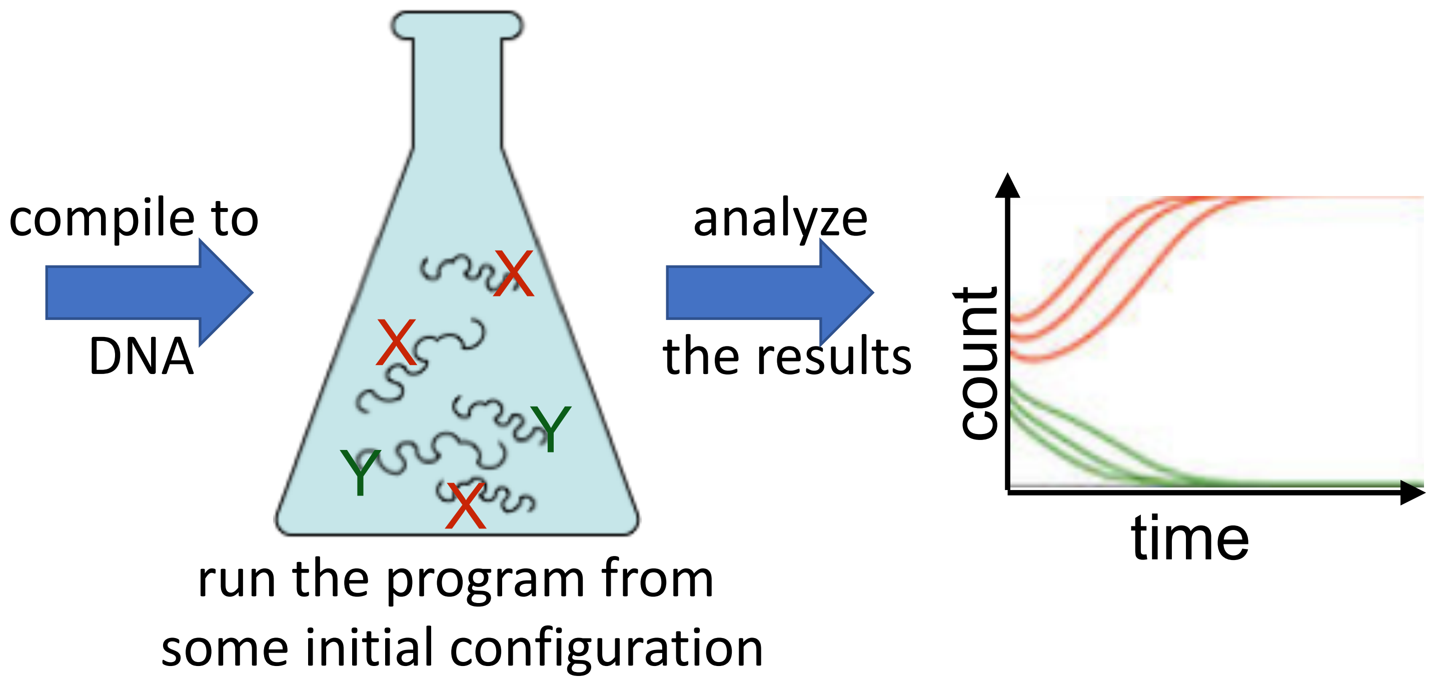
Soloveichik et al., 2013; Chen et al., 2013

Motivation: Programming molecules

molecular program, as a Chemical Reaction Network (CRN)

$$X + X + Y \rightarrow X + X + X$$

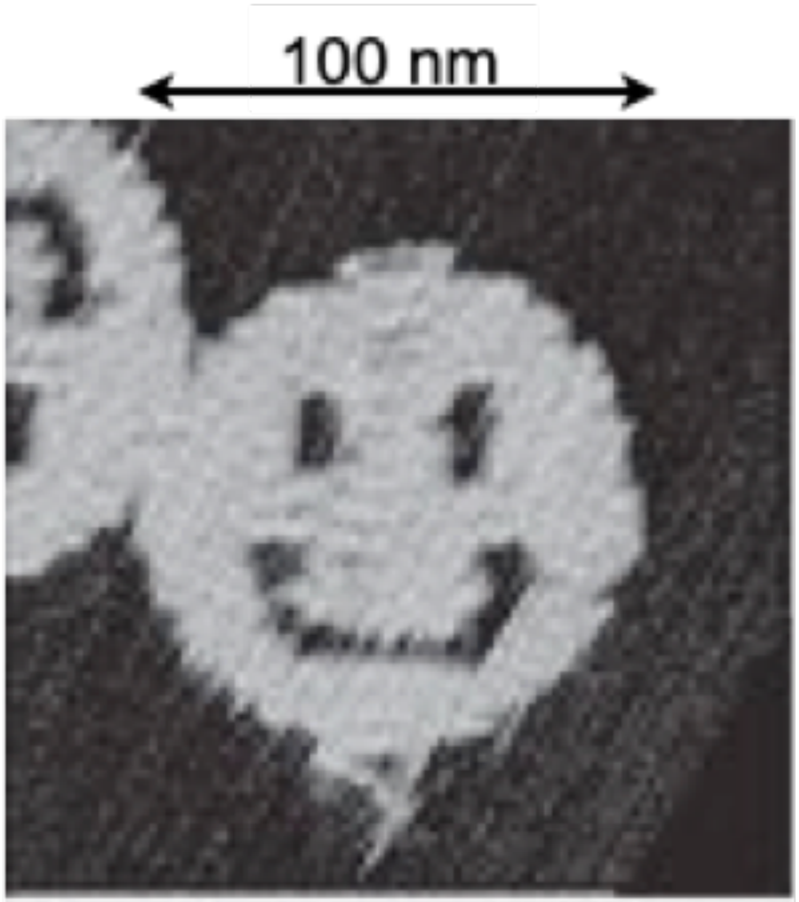
$$X + Y + Y \rightarrow Y + Y + Y$$



Soloveichik et al., 2013; Chen et al., 2013

Motivation: Programming molecules

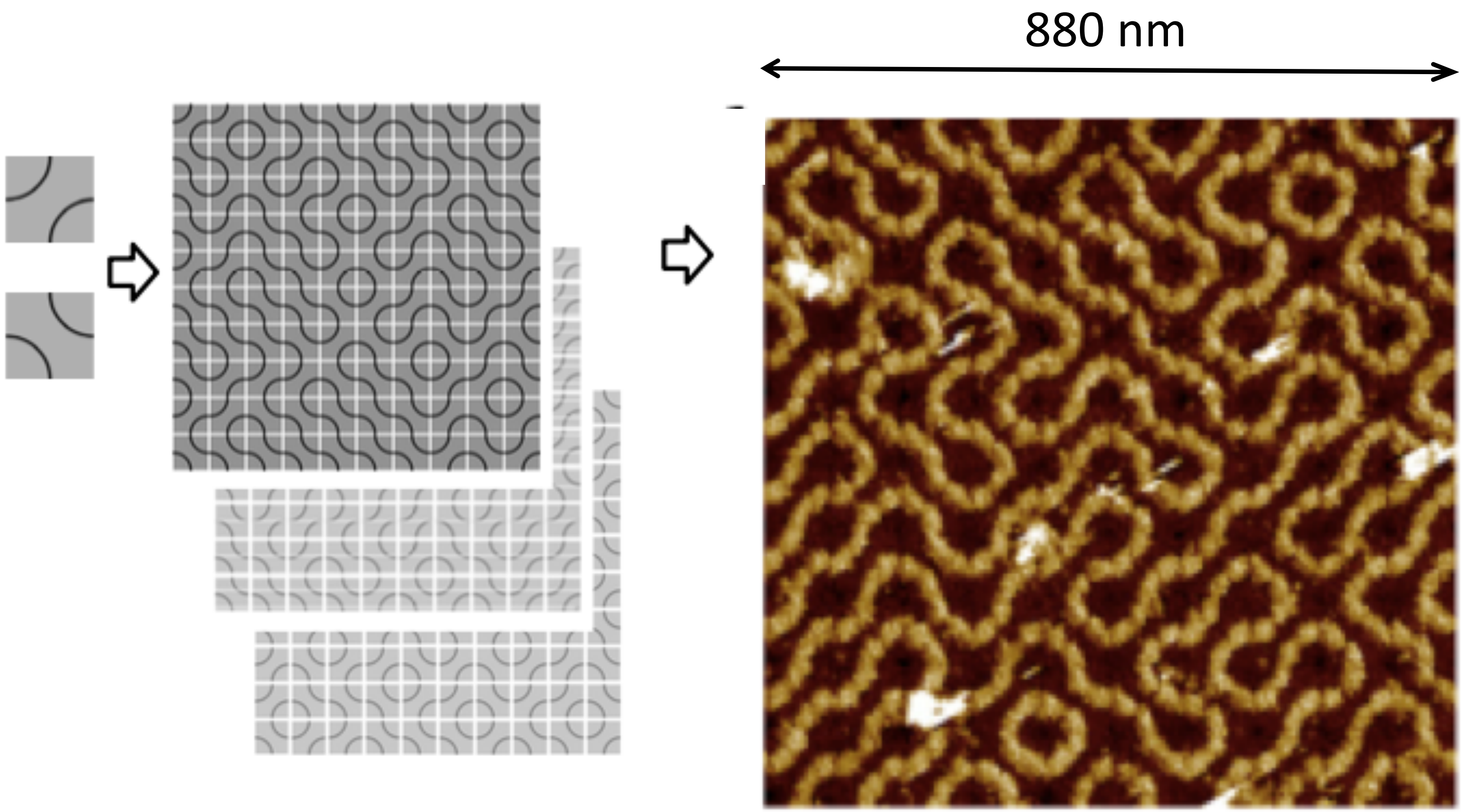
Motivation: Programming molecules



Paul Rothemund, 2006

Motivation: Programming molecules

Motivation: Programming molecules



Tikhomirov et al., 2017

Can computations be energy-efficient?

Can computations be energy-efficient?

"… capable of dissipating an arbitrarily small amount of energy per step if operated sufficiently slowly" (Bennett, 1973).

Can computations be energy-efficient?

Landauer (1961): probably not, because computations are typically logically irreversible (e.g., because they erase or overwrite memory); this implies a lower bound on the entropy generated and energy dissipated at every irreversible step.

Landauer and Bennett, The Fundamental Physical Limits of Computation Scientific American, 1985.

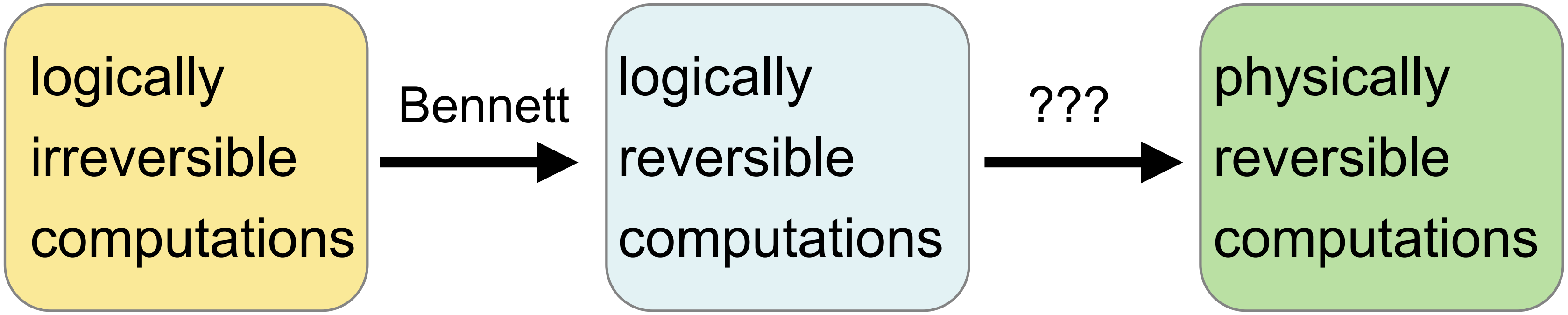
Can computations be energy-efficient?

Landauer (1961): probably not, because computations are typically logically irreversible (e.g., because they erase or overwrite memory); this implies a lower bound on the entropy generated and energy dissipated at every irreversible step.

Bennett (1973): maybe, because logically irreversible computations can be simulated by logically reversible ones, and it might be possible to simulate logically reversible computations by physically reversible ones.

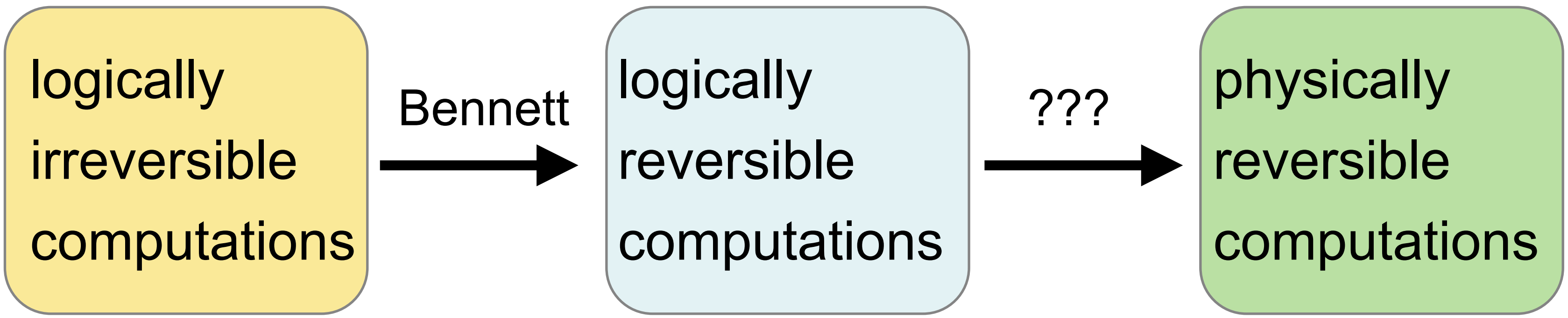
Landauer and Bennett, The Fundamental Physical Limits of Computation Scientific American, 1985.

Can computations be energy-efficient?



Bennett showed how to simulate (irreversible) Turing machines using logically reversible Turing machines.

Can computations be energy-efficient?



Bennett showed how to simulate (irreversible) Turing machines using logically reversible Turing machines.

Turing machines can be simulated by families of Boolean circuits, which in turn can be simulated by chemical reaction networks (CRNs).

Example: a CRN for parity

- *input species:* one copy each of $X_1, X_2, \ldots, X_n$, $X_i \in \{0_i, 1_i\}$
- *output:* N, if parity($X_1, X_2, \ldots, X_n$) = 0

  Y, if parity($X_1, X_2, \ldots, X_n$) = 1

Example: a CRN for parity

- *input species:* one copy each of $X_1, X_2, \ldots, X_n$, $X_i \in \{0_i, 1_i\}$
- *output:* N, if parity$(X_1, X_2, \ldots, X_n) = 0$
  Y, if parity$(X_1, X_2, \ldots, X_n) = 1$

- *auxiliary input species:* one copy of *N*

- *reactions,* $1 \leq i \leq n$:

$$N + 0_i \rightarrow N$$
$$N + 1_i \rightarrow Y$$
$$Y + 0i \rightarrow Y$$
$$Y + 1_i \rightarrow N$$

Example: a CRN for parity

- *input species:* one copy each of $X_1, X_2, \ldots, X_n$, $X_i \in \{0_i, 1_i\}$
- *output:* N, if parity$(X_1, X_2, \ldots, X_n) = 0$
  Y, if parity$(X_1, X_2, \ldots, X_n) = 1$

- *auxiliary input species:* one copy of *N*

- *reactions,* $1 \leq i \leq n$:

$$N + 0_i \rightarrow N$$
$$N + 1_i \rightarrow Y$$
$$Y + 0_i \rightarrow Y$$
$$Y + 1_i \rightarrow N$$

*This CRN is **stochastic**, and **logically irreversible**: it's not possible in general to trace back to the initial input from a given reachable configuration*

Example: a logically reversible CRN for parity

- *input species:* one copy each of $X_1, X_2, \ldots, X_n$, $X_i \in \{0_i, 1_i\}$

  output: N, if parity($X_1, X_2, \ldots, X_n$) = 0

  Y, if parity($X_1, X_2, \ldots, X_n$) = 1

Example: a logically reversible CRN for parity

- *input species:* one copy each of $X_1, X_2, \ldots, X_n$, $X_i \in \{0_i, 1_i\}$
- *output:* N, if parity($X_1, X_2, \ldots, X_n$) = 0

  Y, if parity($X_1, X_2, \ldots, X_n$) = 1

- *auxiliary input species:* one copy of *N* and t_0
- *reactions:*

Example: a logically reversible CRN for parity

- *input species:* one copy each of $X_1, X_2, \ldots, X_n$, $X_i \in \{0_i, 1_i\}$
- *output:* N, if parity($X_1, X_2, \ldots, X_n$) = 0
  Y, if parity($X_1, X_2, \ldots, X_n$) = 1

- *auxiliary input species:* one copy of *N* and t_0
- *reactions:*

$N + 0_1 + t_0 \rightarrow N + 0_1 + t_1$

$N + 1_1 + t_0 \rightarrow Y + 1_1 + t_1$

$N + 0_2 + t_1 \rightarrow N + 0_2 + t_2$

$N + 1_2 + t_1 \rightarrow Y + 1_2 + t_2$

$Y + 0_2 + t_1 \rightarrow Y + 0_2 + t_2$

$Y + 1_2 + t_1 \rightarrow N + 1_2 + t_2$

...

Example: a logically reversible CRN for parity

- *input species:* one copy each of $X_1, X_2, \ldots, X_n$, $X_i \in \{0_i, 1_i\}$
- *output:* N, if parity($X_1, X_2, \ldots, X_n$) = 0

  Y, if parity($X_1, X_2, \ldots, X_n$) = 1

- *auxiliary input species:* one copy of *N* and t_0
- *reactions:*

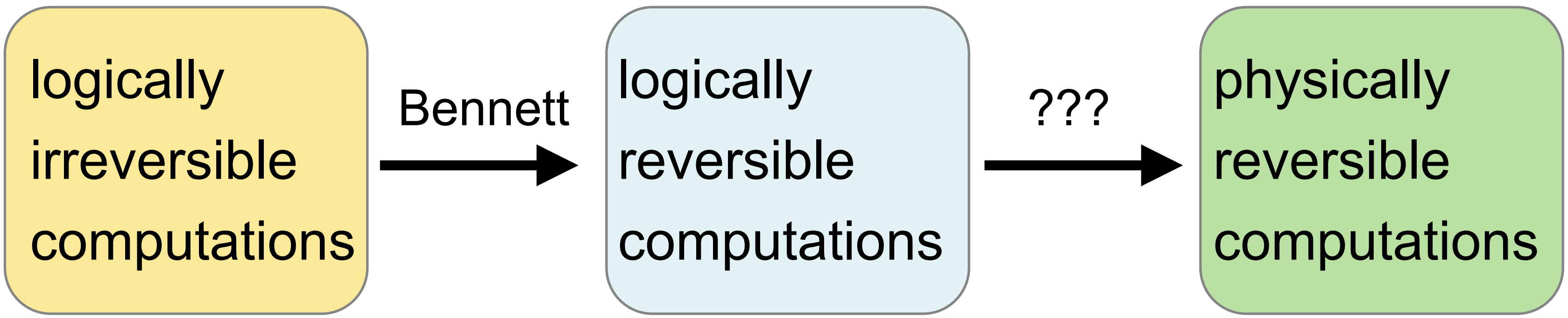
$$N + 0_1 + t_0 \rightarrow N + 0_1 + t_1$$
$$N + 1_1 + t_0 \rightarrow Y + 1_1 + t_1$$

$$N + 0_2 + t_1 \rightarrow N + 0_2 + t_2$$
$$N + 1_2 + t_1 \rightarrow Y + 1_2 + t_2$$
$$Y + 0_2 + t_1 \rightarrow Y + 0_2 + t_2$$
$$Y + 1_2 + t_1 \rightarrow N + 1_2 + t_2$$

...

This CRN is ***deterministic:*** *at most one reaction is applicable at any point, and* ***logically reversible****: there is only one way to step backwards from a reachable configuration.*

Can computations be energy-efficient?



Can computations be energy-efficient?

logically irreversible computations → (Bennett) → logically reversible computations → (???) → physically reversible computations

t-step, deterministic CRN with reactions:

$X_j + X'_j \rightarrow Y_j + Y'_j$

$1 \leq j \leq d$

→

t-step deterministic, logically reversible CRN:

auxiliary input species: $?_i, 1 \leq i \leq t$, t_0

$t_{i-1} + ?_i + X_j + X'_j \rightarrow t_i + j_i + Y_j + Y'_j$

$1 \leq j \leq d,\ 1 \leq i \leq t$

Can computations be energy-efficient?

logically irreversible computations → (Bennett) → logically reversible computations → (???) → physically reversible computations

t-step, deterministic CRN with reactions:

$X_j + X'_j \rightarrow Y_j + Y'_j$

$1 \leq j \leq d$

→

t-step deterministic, logically reversible CRN:

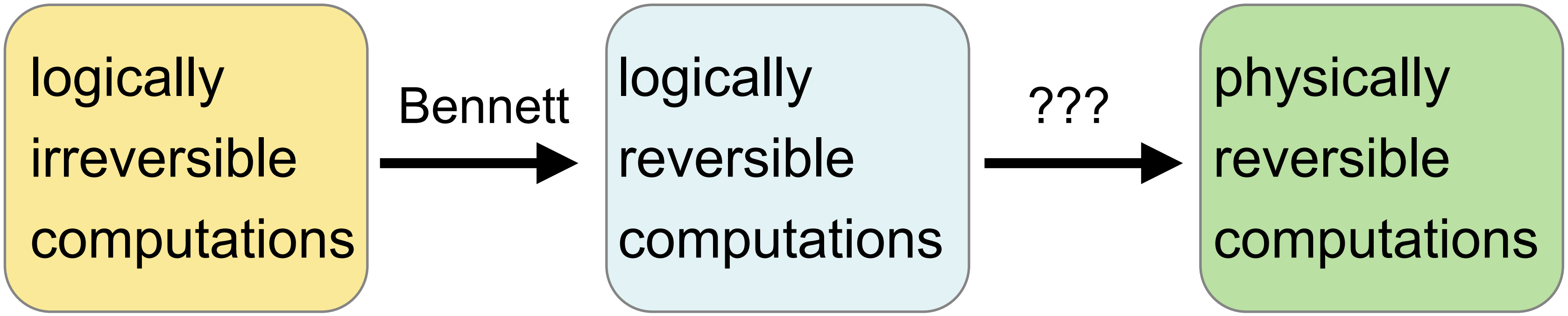
auxiliary input species: $?_i, 1 \leq j \leq t, t_0$

$t_{i-1} + ?_i + X_j + X'_j \rightarrow t_i + j_i + Y_j + Y'_j$

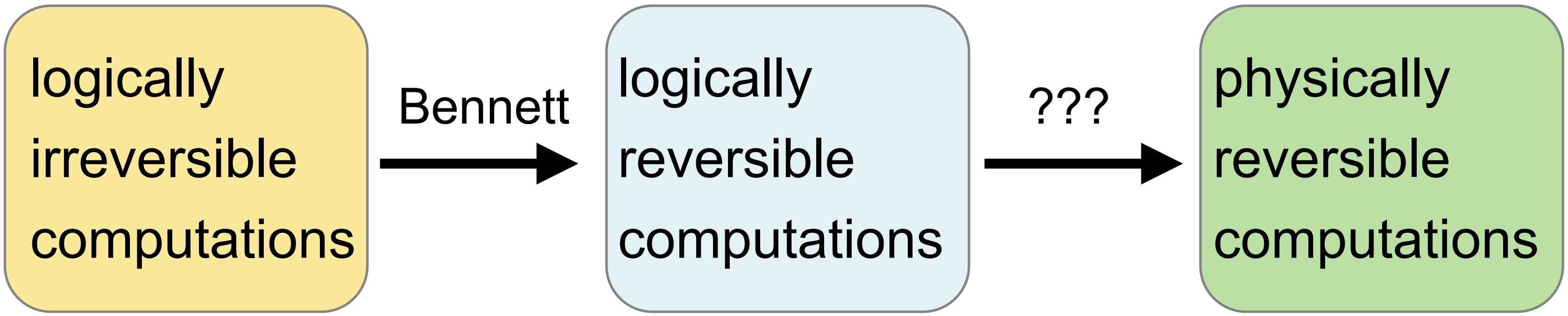
$1 \leq j \leq d, 1 \leq i \leq t$

Time(t) $\subseteq$ log-rev-Time(t)

Can computations be energy-efficient?

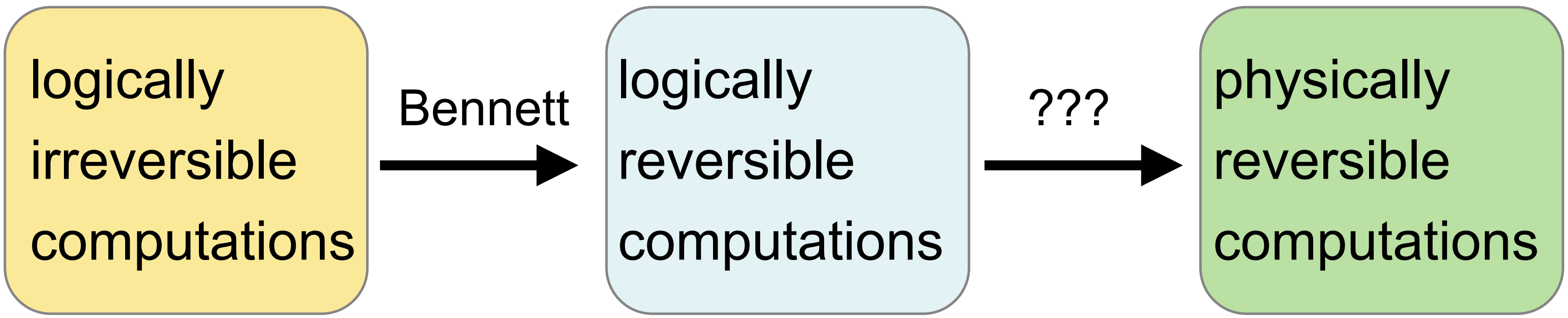


Can computations be energy-efficient?



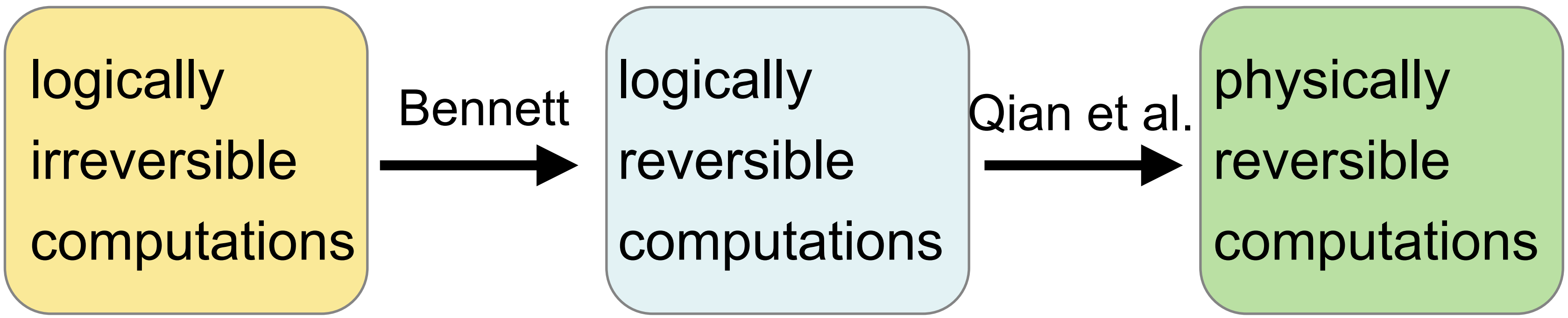
“The existence of logically reversible automata suggests that physical computers might be made thermodynamically reversible, and hence capable of dissipating an arbitrarily small amount of energy per step if operated sufficiently slowly” (Bennett, 1973).

Can computations be energy-efficient?

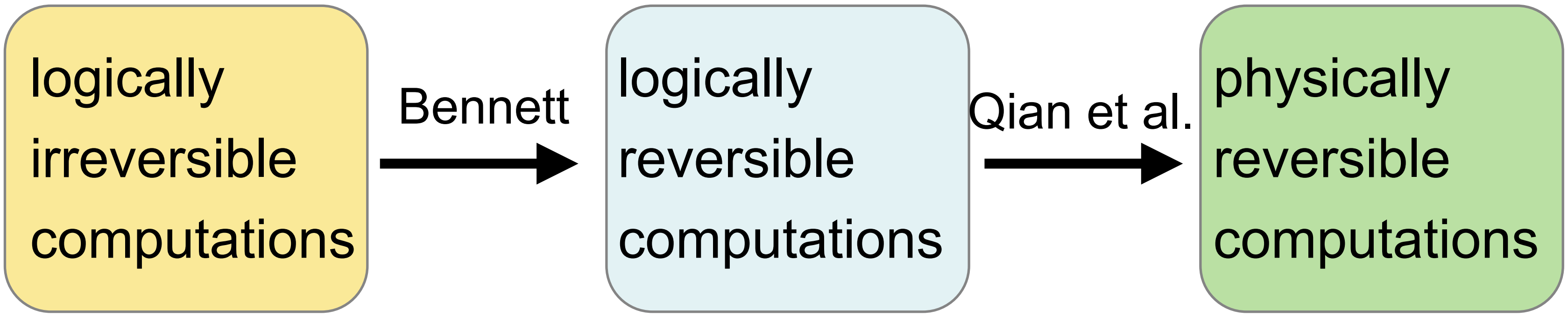


"The chemical realization of a logically reversible computation is a chain of reactions … a major reactant (analogous to DNA) … encodes the logical state, and minor reactants react with the major one to change the logical state … the minor reactants are all present at definite concentrations, which may be manipulated to drive the computation forward or backward." (Bennett, 1973).

Can computations be energy-efficient?

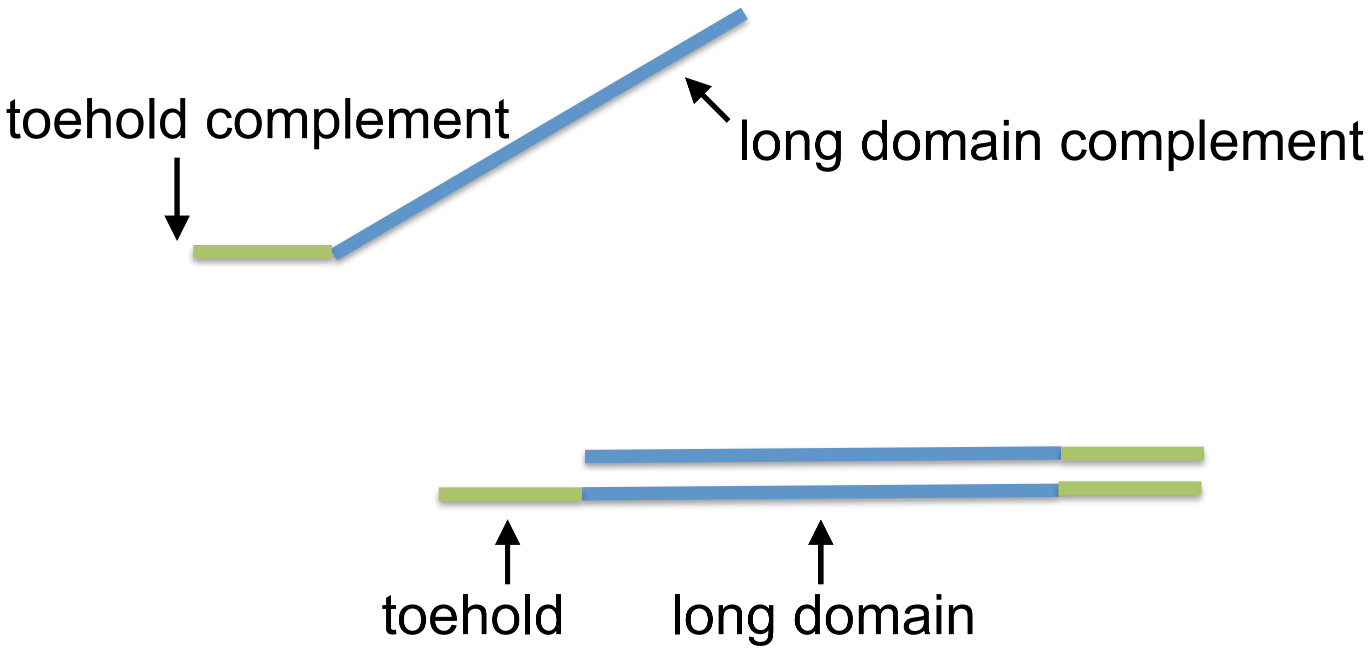


Can computations be energy-efficient?



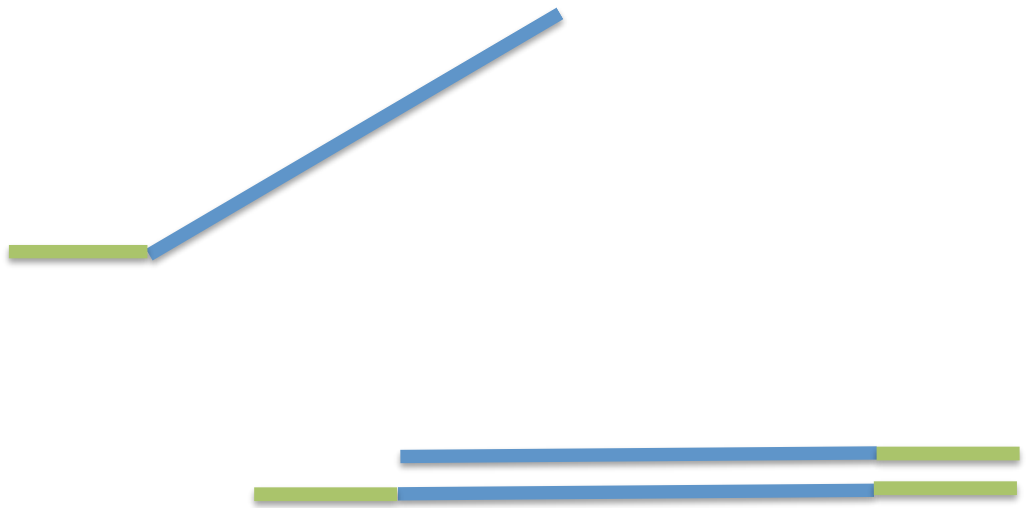
“Here we propose a chemical implementation of [computing machines] using DNA strand displacement cascades as the underlying chemical primitive. We capture the motivating feature of Bennett’s scheme: that physical reversibility corresponds to logically reversible computation, and arbitrarily little energy per computation step is required.” (Qian et al., DNA 2011).

DSDs | DNA strand displacements

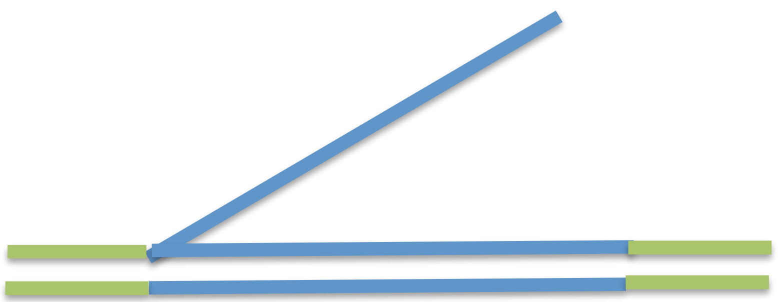


Soloveichik, Seelig, Winfree. "DNA as a universal substrate for chemical kinetics", PNAS 2010

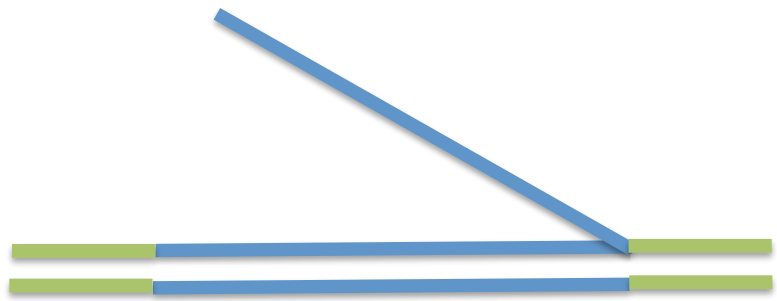
DSDs | DNA strand displacements



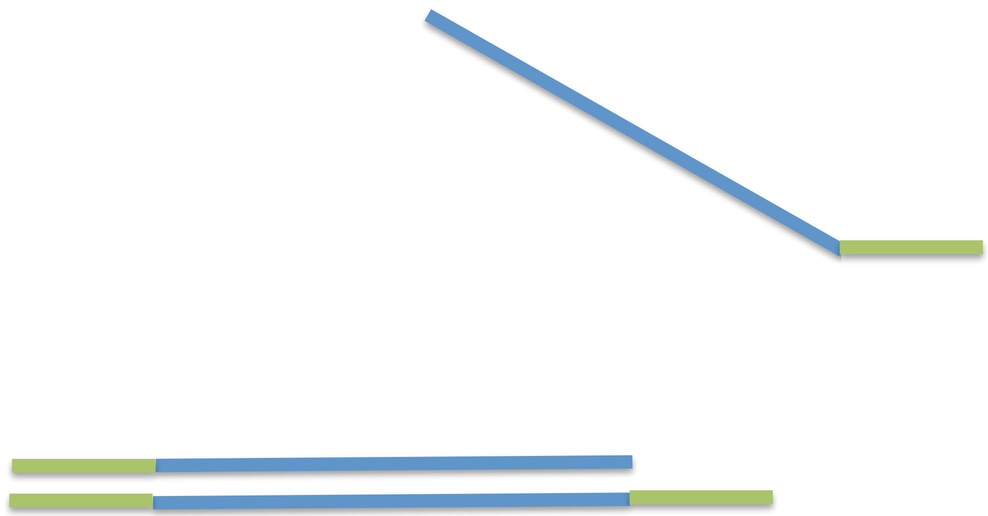
DSDs | DNA strand displacements



DSDs | DNA strand displacements



DSDs | DNA strand displacements

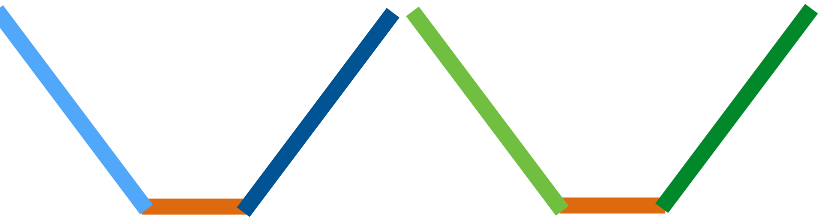


reversible, and thus energy-efficient

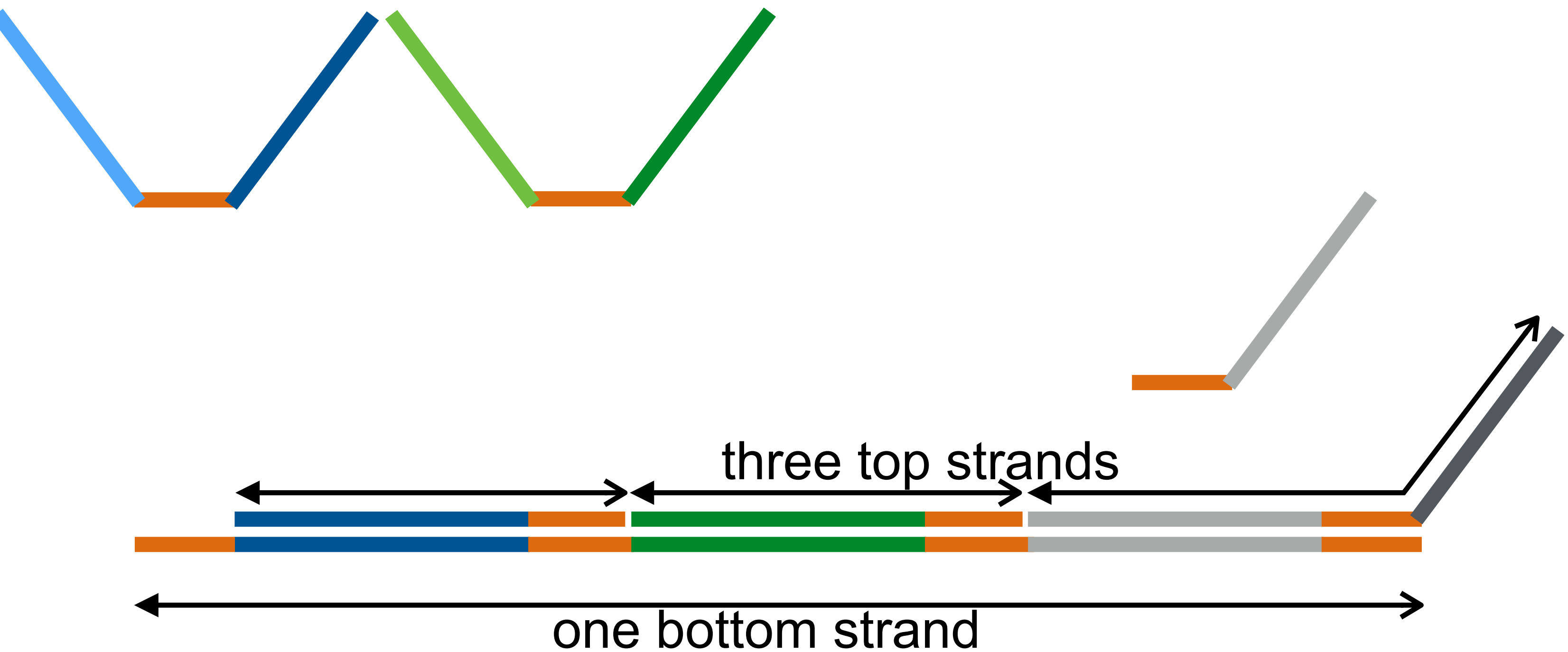
From CRNs to DSDs

$A + B \rightleftharpoons C$

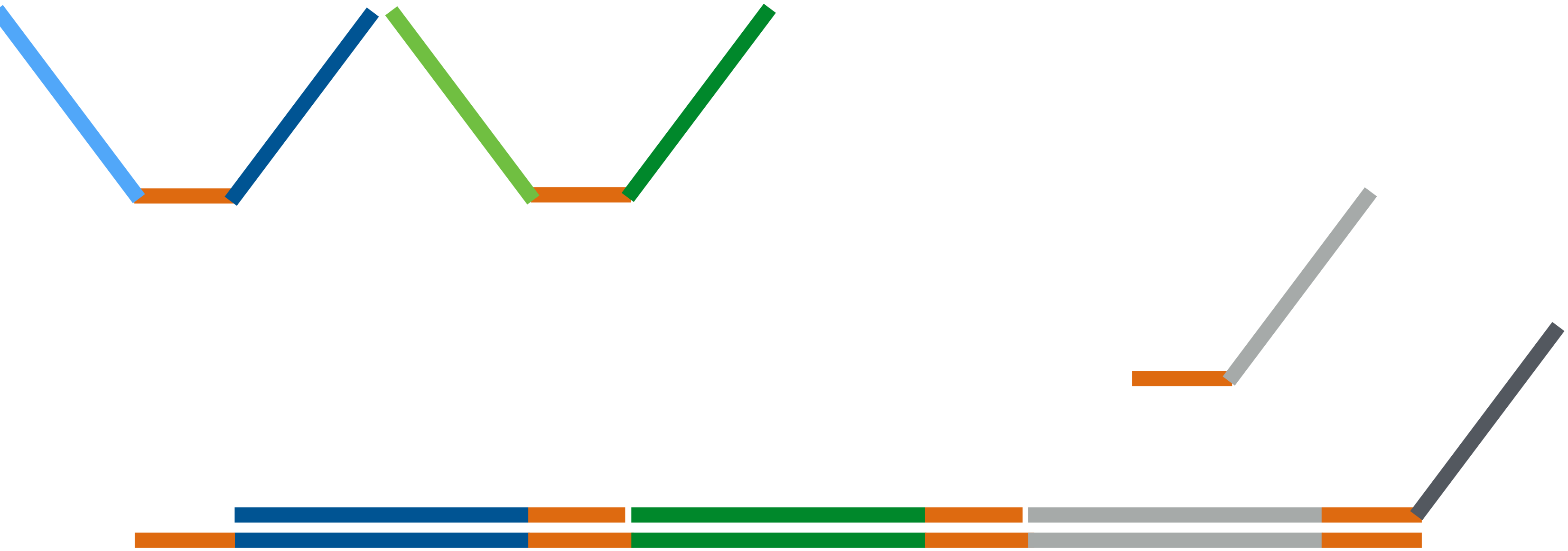
From CRNs to DSDs



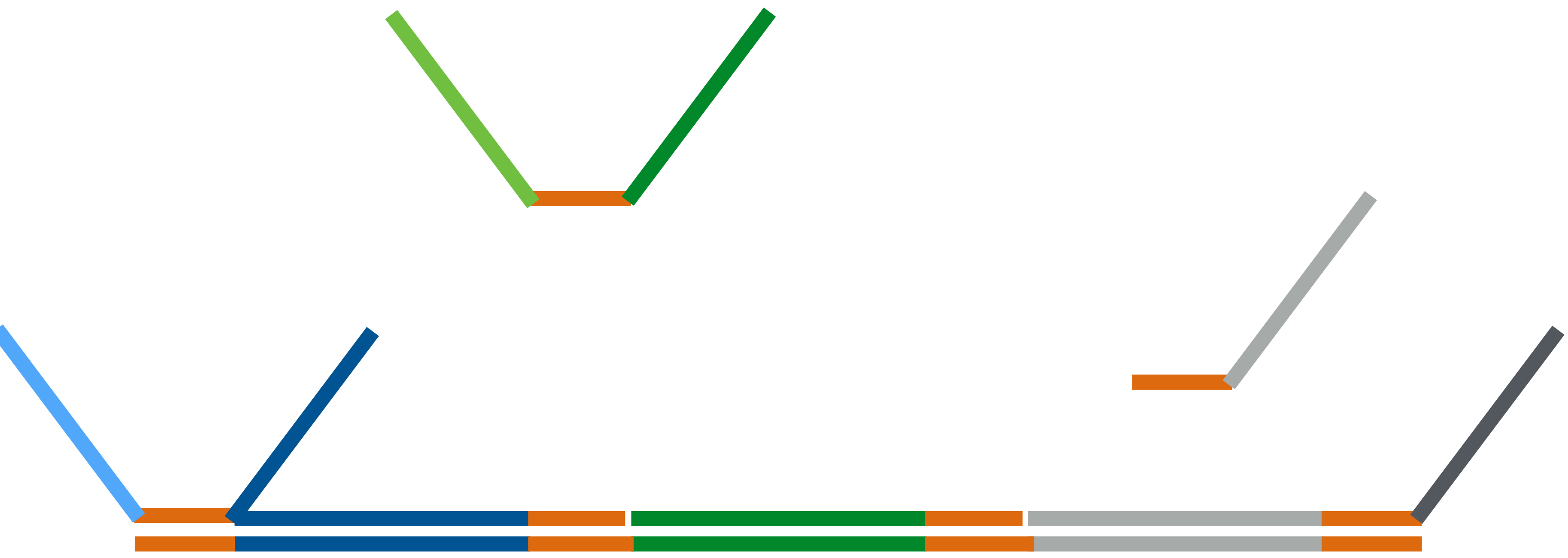
From CRNs to DSDs



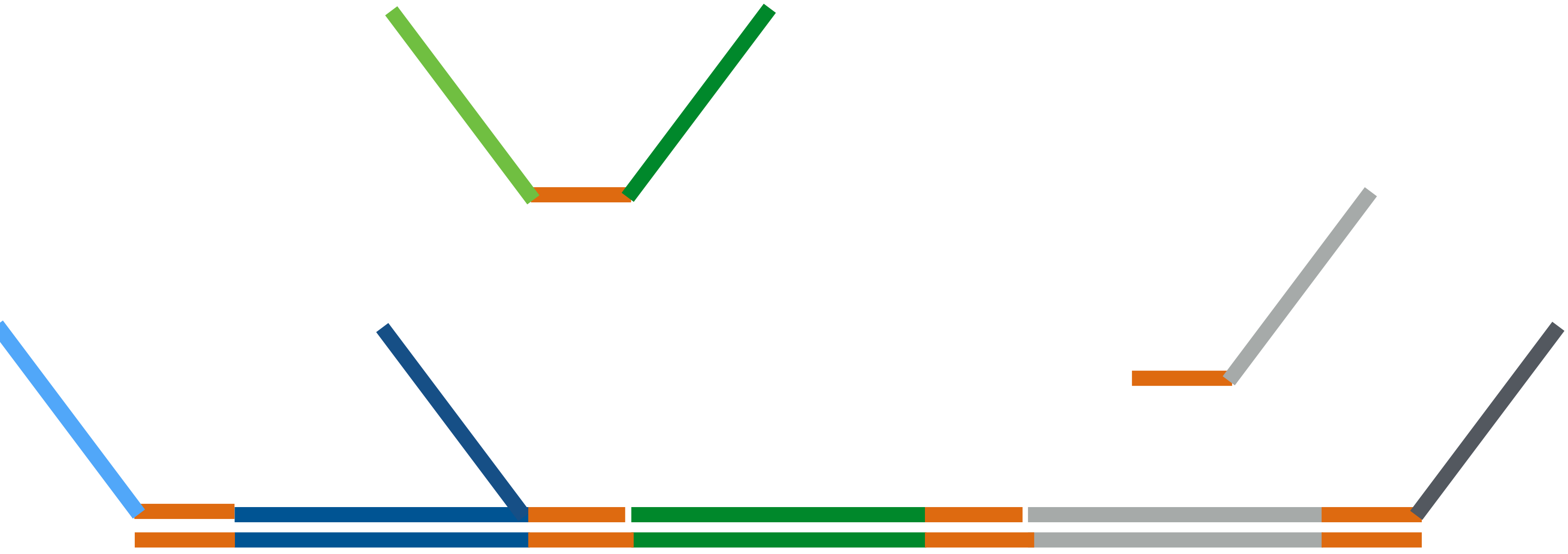
From CRNs to DSDs



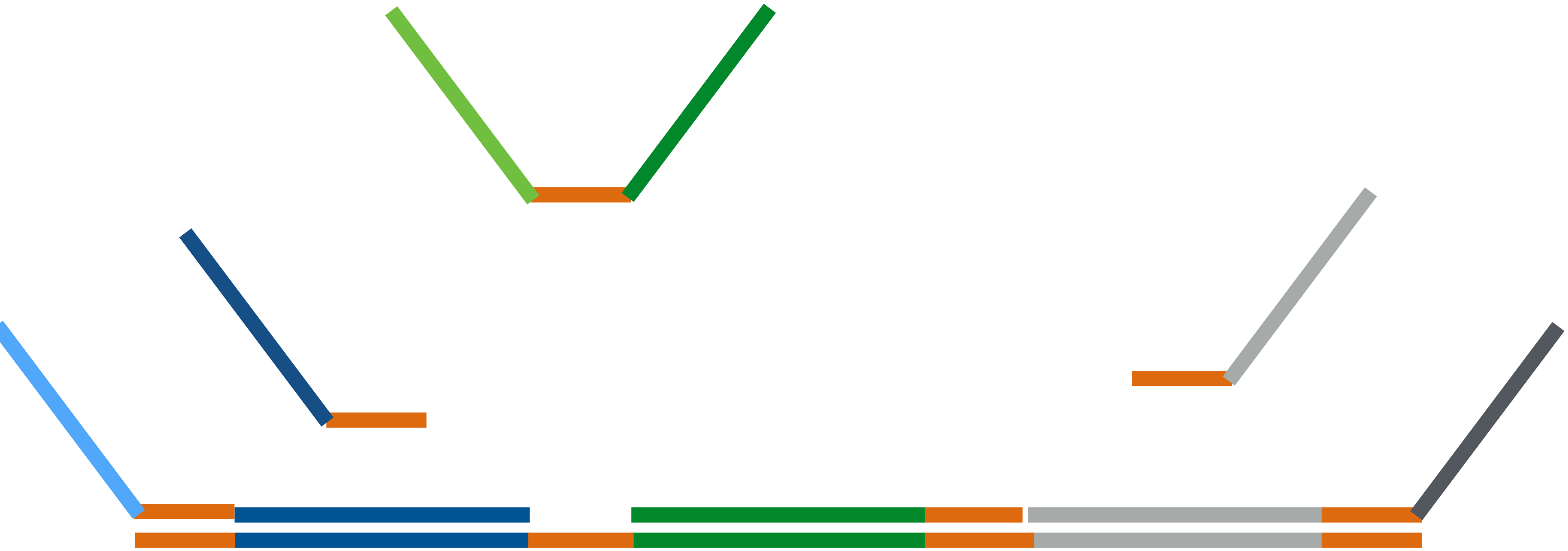
From CRNs to DSDs



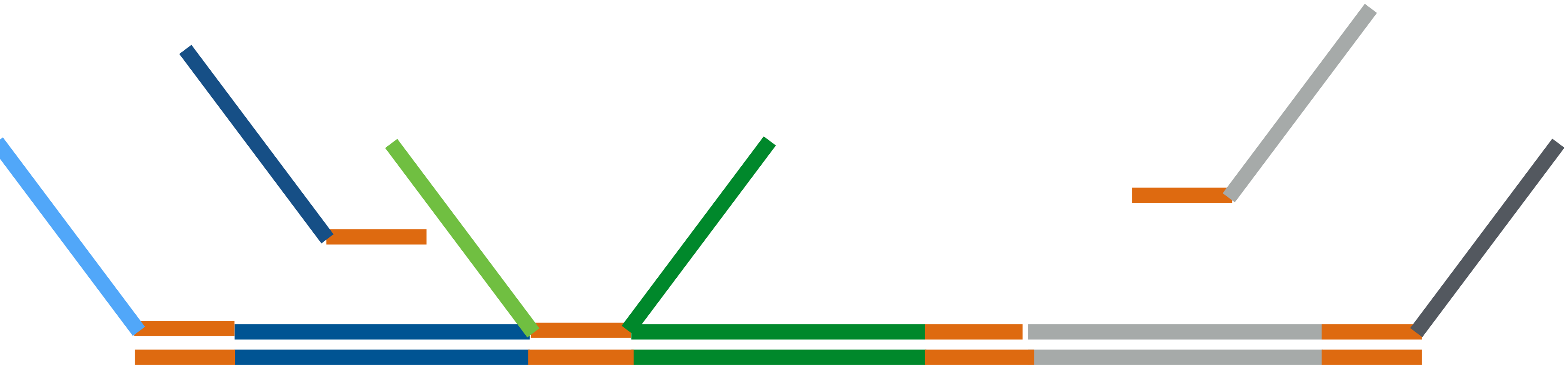
From CRNs to DSDs



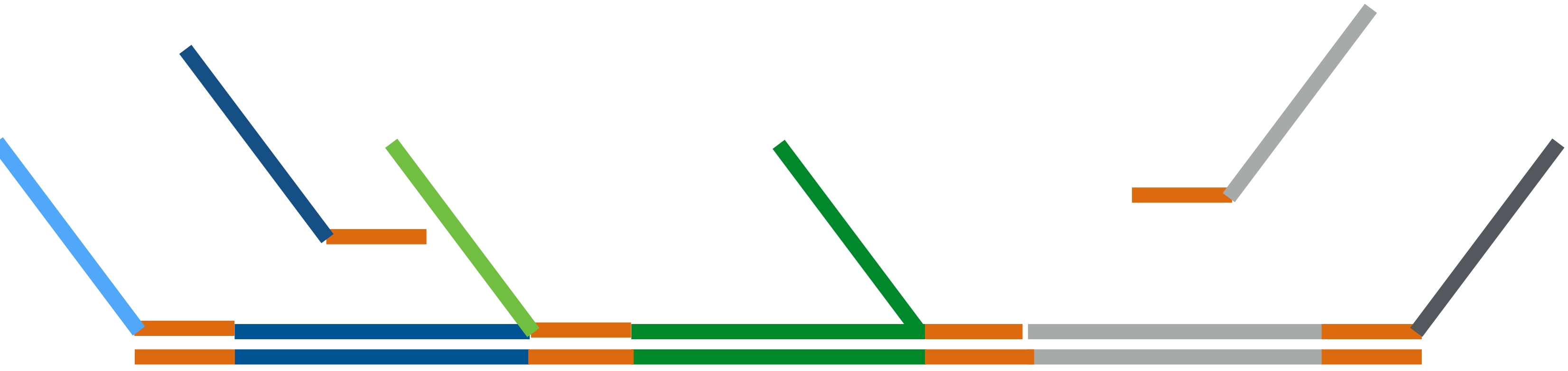
From CRNs to DSDs



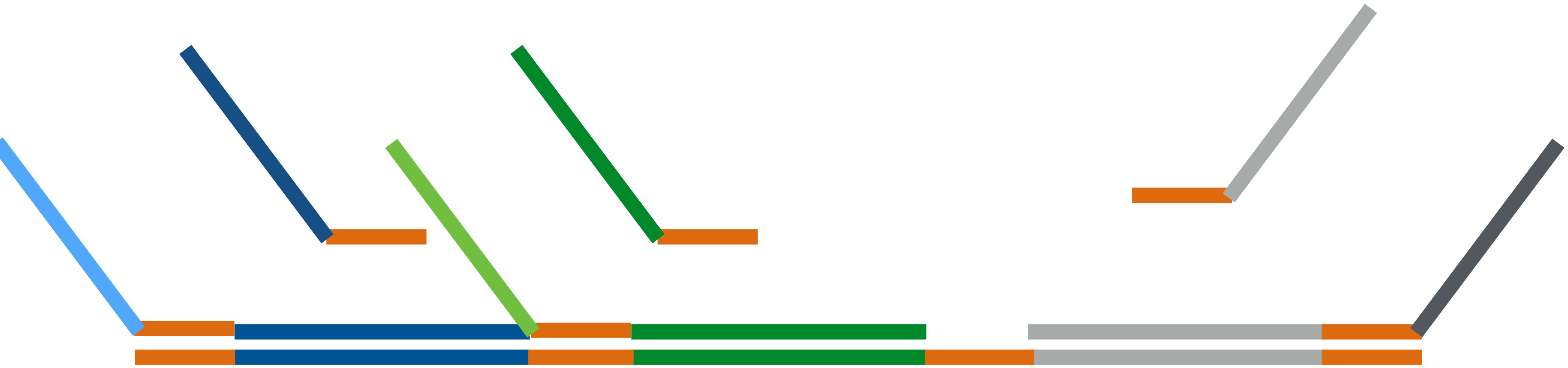
From CRNs to DSDs



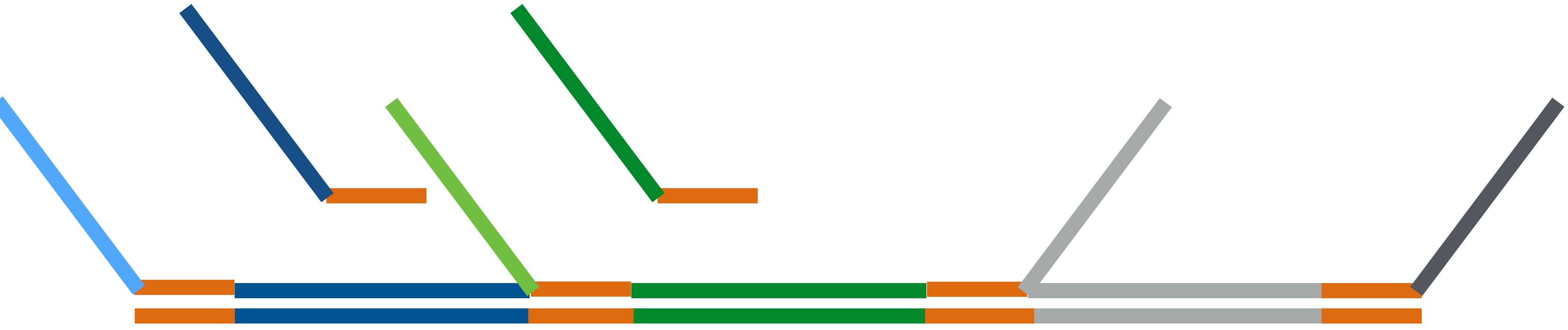
From CRNs to DSDs



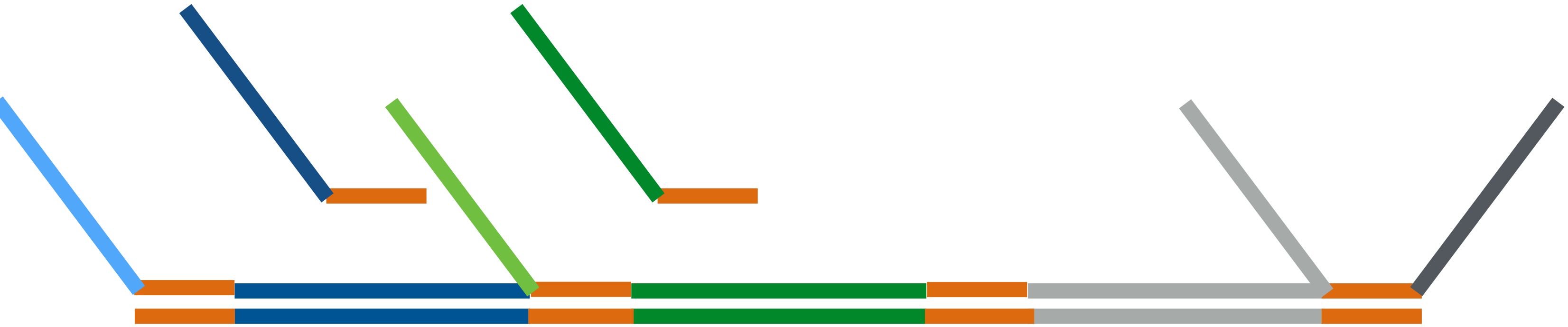
From CRNs to DSDs



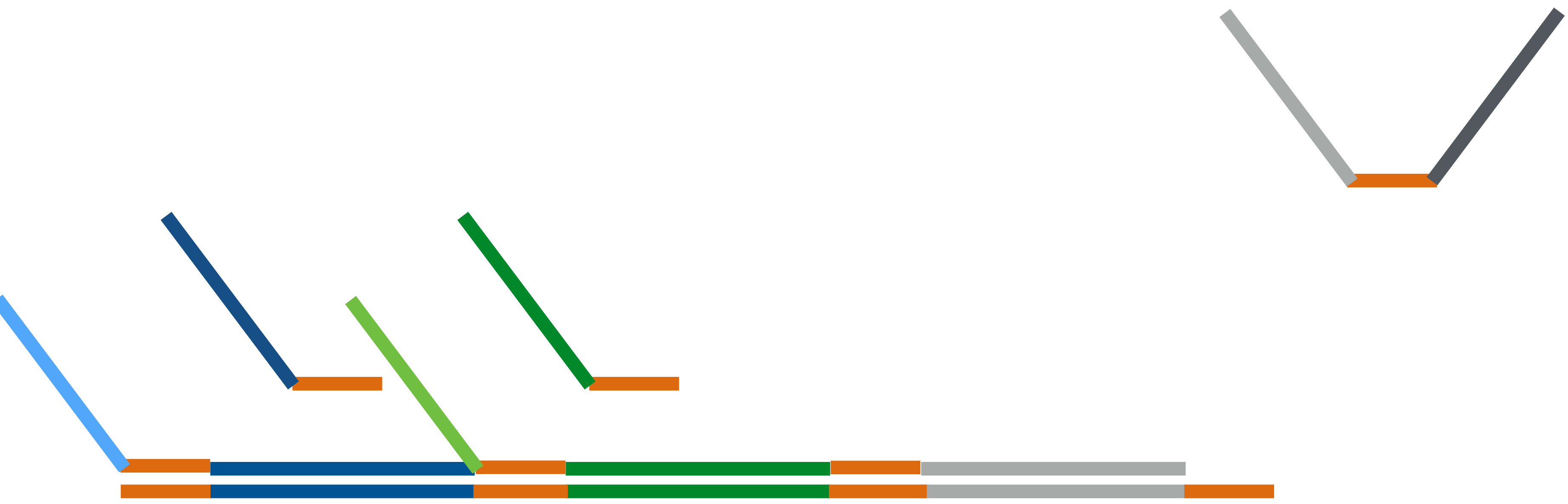
From CRNs to DSDs



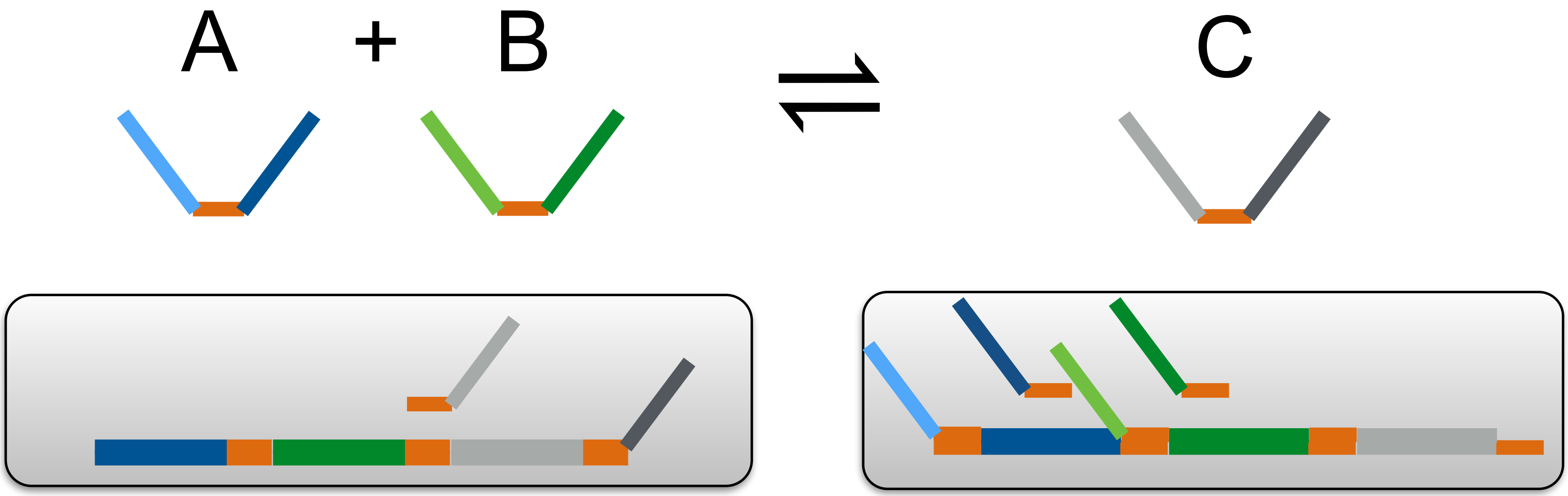
From CRNs to DSDs



From CRNs to DSDs

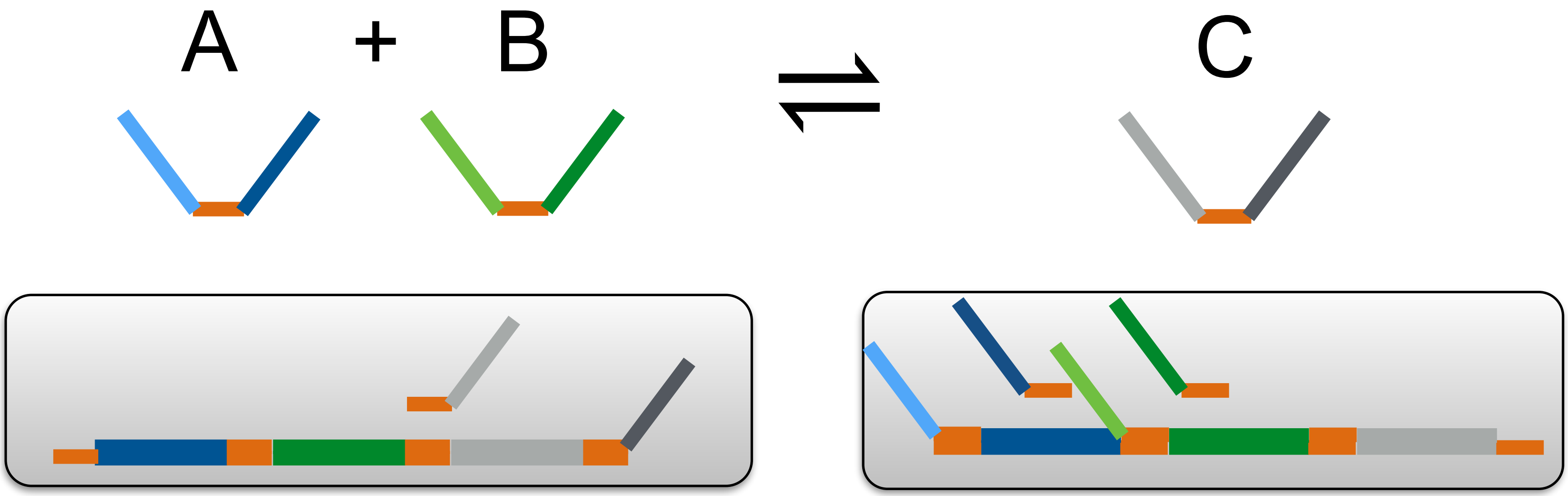


From CRNs to DSDs



transformer molecules

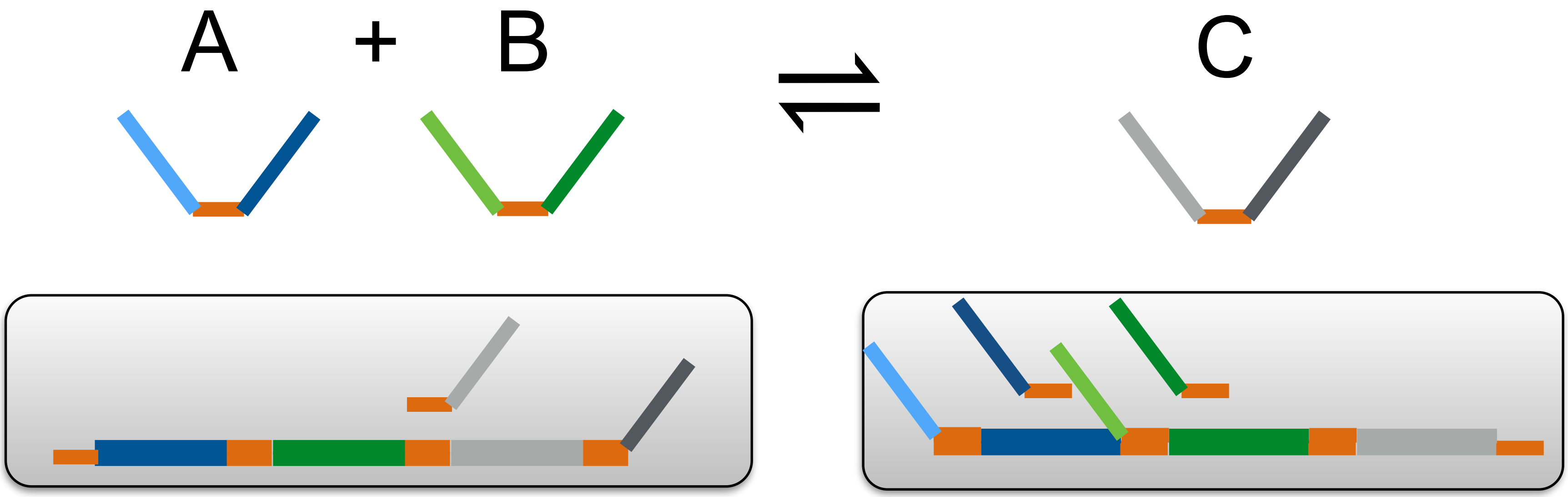
From CRNs to DSDs



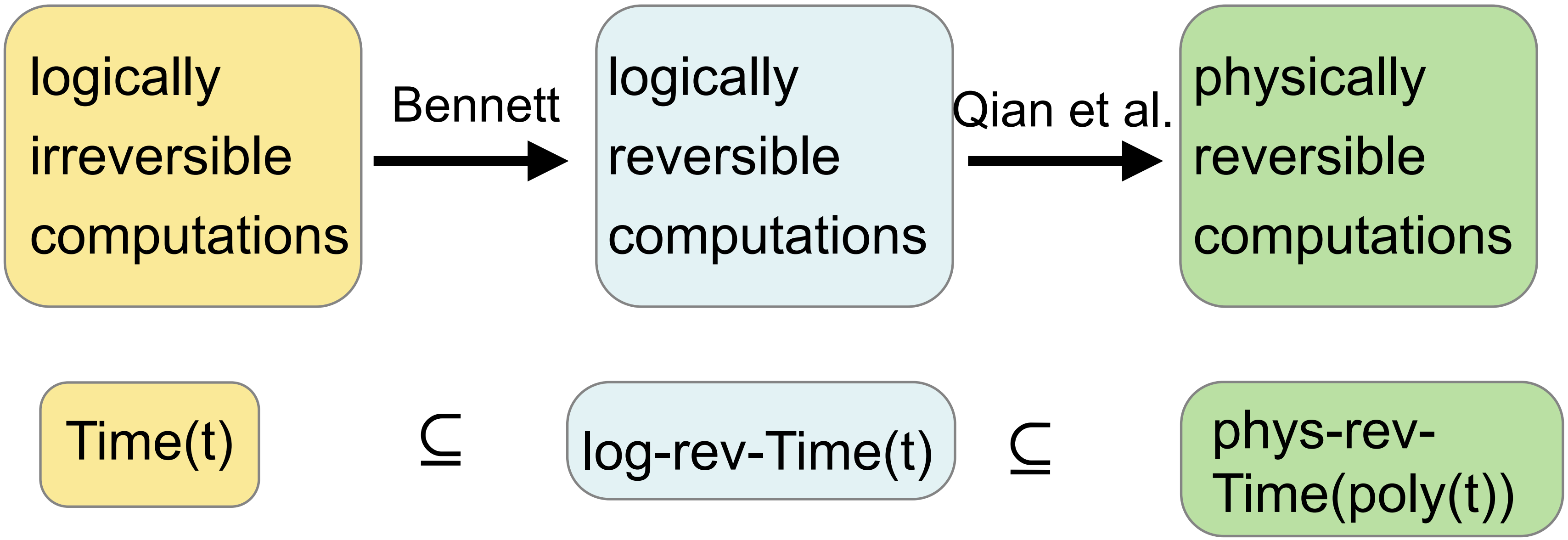
transformer molecules

"The chemical realization of a logically reversible computation is a chain of reactions … a major reactant (analogous to DNA) … encodes the logical state, and minor reactants that react with the major one to change the logical state … the minor reactants are all present at definite concentrations, which may be manipulated to drive the computation forward or backward." (Bennett, 1973).

From CRNs to DSDs



transformer molecules


logically irreversible computations
Bennett
logically reversible computations
Qian et al.
physically reversible computations
Time(t)
⊆
log-rev-Time(t)
⊆
phys-rev-Time(poly(t))

Towards space- and energy-efficient computations

Towards space- and energy-efficient computations

a space-efficient counter

counter execution

$0_3\ 0_2\ 0_1$

$0_3\ 0_2\ 1_1$

$0_3\ 1_2\ 0_1$

$0_3\ 1_2\ 1_1$

$1_3\ 0_2\ 0_1$

$1_3\ 0_2\ 1_1$

$1_3\ 1_2\ 0_1$

$1_3\ 1_2\ 1_1$

Towards space- and energy-efficient computations

a space-efficient counter

initial species: $0_3, 0_2, 0_1$

reactions:

$$0_1 \rightarrow 1_1 \quad (1)$$

$$0_2 + 1_1 \rightarrow 1_2 + 0_1 \quad (2)$$

$$0_3 + 1_2 + 1_1 \rightarrow 1_3 + 0_2 + 0_1 \quad (3)$$

counter execution

$0_3\ 0_2\ 0_1$

$0_3\ 0_2\ 1_1$

$0_3\ 1_2\ 0_1$

$0_3\ 1_2\ 1_1$

$1_3\ 0_2\ 0_1$

$1_3\ 0_2\ 1_1$

$1_3\ 1_2\ 0_1$

$1_3\ 1_2\ 1_1$

Towards space- and energy-efficient computations

a space-efficient counter

initial species: 0_3, 0_2, 0_1

reactions:

$0_1 \rightarrow 1_1$ (1)

$0_2 + 1_1 \rightarrow 1_2 + 0_1$ (2)

$0_3 + 1_2 + 1_1 \rightarrow 1_3 + 0_2 + 0_1$ (3)

counter execution

$0_3\ 0_2\ 0_1$
(1)
$0_3\ 0_2\ 1_1$
(2)
$0_3\ 1_2\ 0_1$
$0_3\ 1_2\ 1_1$
$1_3\ 0_2\ 0_1$
$1_3\ 0_2\ 1_1$
$1_3\ 1_2\ 0_1$
$1_3\ 1_2\ 1_1$

Towards space- and energy-efficient computations

a space-efficient counter

initial species: $0_3,\ 0_2,\ 0_1$

reactions:

$$0_1 \rightarrow 1_1 \qquad (1)$$

$$0_2 + 1_1 \rightarrow 1_2 + 0_1 \qquad (2)$$

$$0_3 + 1_2 + 1_1 \rightarrow 1_3 + 0_2 + 0_1 \qquad (3)$$

counter execution

$0_3\ 0_2\ 0_1$

(1) ↓

$0_3\ 0_2\ 1_1$

(2) ↓

$0_3\ 1_2\ 0_1$

(1) ↓

$0_3\ 1_2\ 1_1$

(3) ↓

$1_3\ 0_2\ 0_1$

(1) ↓

$1_3\ 0_2\ 1_1$

(2) ↓

$1_3\ 1_2\ 0_1$

(1) ↓

$1_3\ 1_2\ 1_1$

Towards space- and energy-efficient computations

a space-efficient counter

initial species: 0_3, 0_2, 0_1

reactions:

$$0_1 \rightarrow 1_1 \qquad (1)$$

$$0_2 + 1_1 \rightarrow 1_2 + 0_1 \qquad (2)$$

$$0_3 + 1_2 + 1_1 \rightarrow 1_3 + 0_2 + 0_1 \qquad (3)$$

this binary counter CRN is

- deterministic
- logically reversible
- space-efficient: *n*-bit counter uses *n* species to count to 2^n

counter execution

$0_3\ 0_2\ 0_1$ (1)

$0_3\ 0_2\ 1_1$ (2)

$0_3\ 1_2\ 0_1$ (1)

$0_3\ 1_2\ 1_1$ (3)

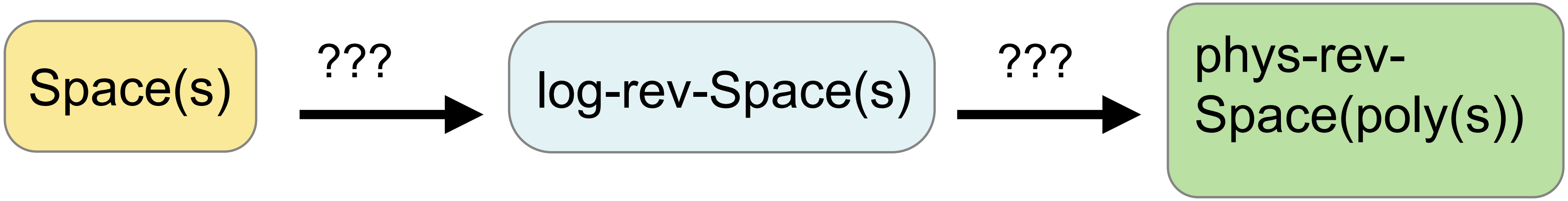
$1_3\ 0_2\ 0_1$ (1)

$1_3\ 0_2\ 1_1$ (2)

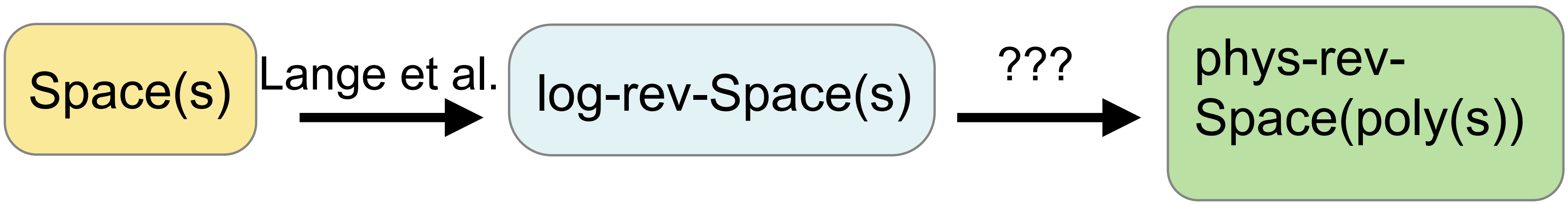
$1_3\ 1_2\ 0_1$ (1)

$1_3\ 1_2\ 1_1$

Towards space- and energy-efficient computations

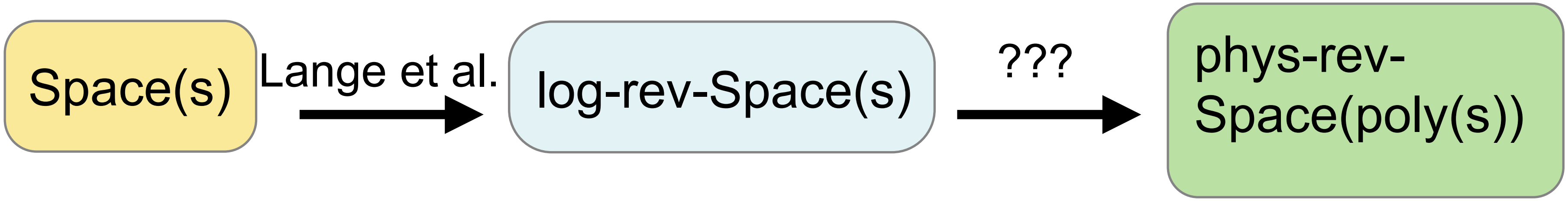


Towards space- and energy-efficient computations

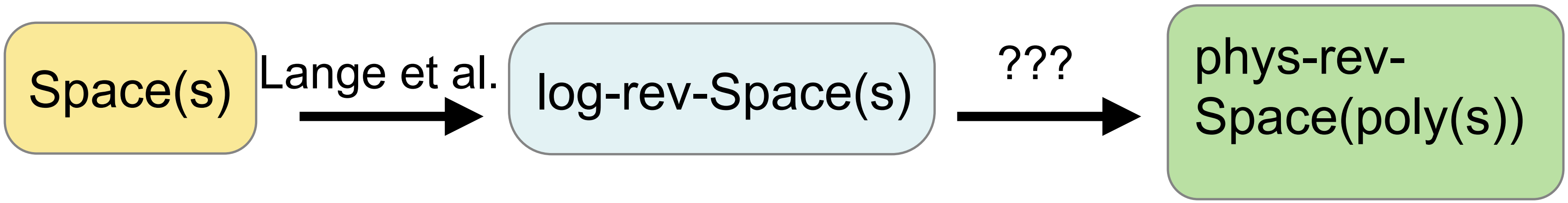


"[We] describe the simulation of an s(n) space-bounded deterministic Turing machine by a reversible Turing machine operating in space s(n). It thus answers a question posed by Bennett in 1989 and refutes the conjecture made by Li and Vityani in 1996” (Lange et al., 1998).

Towards space- and energy-efficient computations



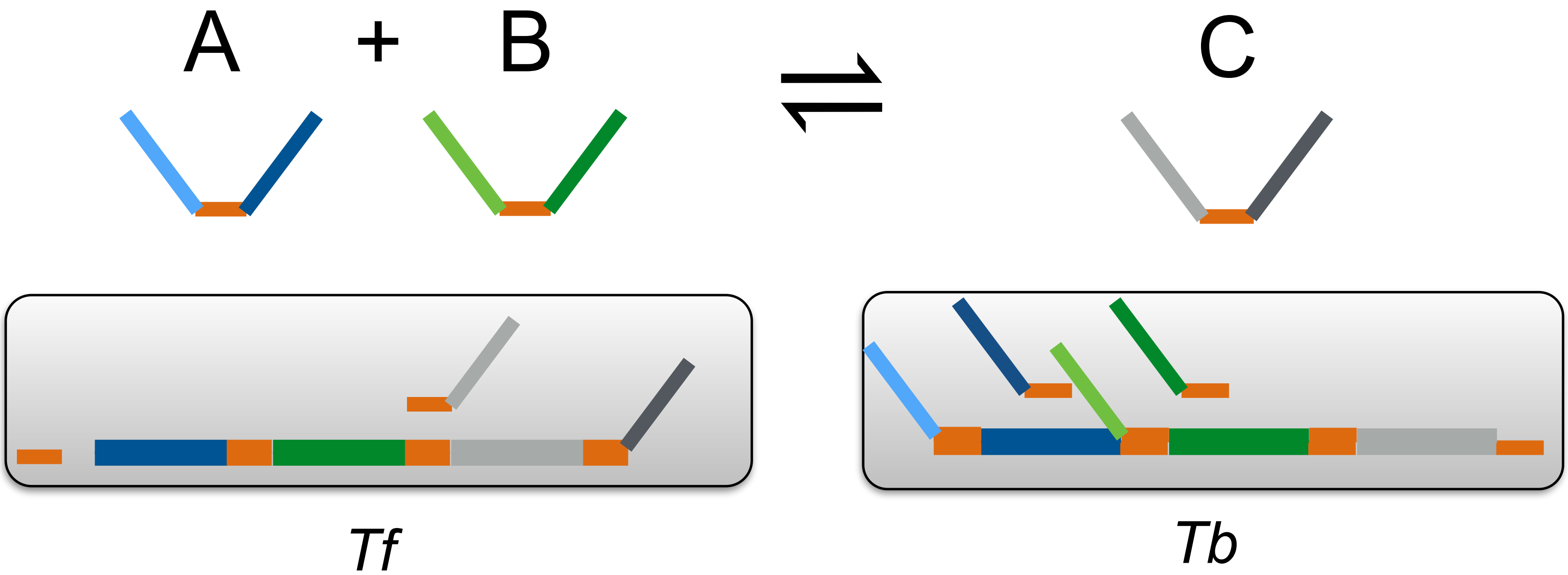
Towards space- and energy-efficient computations



Unfortunately, the compilation of deterministic, logically reversible CRNs with s species into DSDs may result in an exponential blow-up of the number of species, and thus the space (volume).

This is because of the transformer molecules needed by the compilation.

Recall: transformer molecules



$$Tf + A + B \rightleftharpoons C + Tb$$

Accounting for transformers in the counter:

initial species: 0_3, 0_2, 0_1 *(one copy each)*

reactions:

$$0_1 \rightarrow 1_1 \quad (1)$$

$$0_2 + 1_1 \rightarrow 1_2 + 0_1 \quad (2)$$

$$0_3 + 1_2 + 1_1 \rightarrow 1_3 + 0_2 + 0_1 \quad (3)$$

counter execution

$0_3\ 0_2\ 0_1$

(1)

$0_3\ 0_2\ 1_1$

(2)

$0_3\ 1_2\ 0_1$

(1)

$0_3\ 1_2\ 1_1$

(3)

$1_3\ 0_2\ 0_1$

(1)

$1_3\ 0_2\ 1_1$

(2)

$1_3\ 1_2\ 0_1$

(1)

$1_3\ 1_2\ 1_1$

Accounting for transformers in the counter:

initial species: 0_3, 0_2, 0_1 *(one copy each)*

T^f_1 *(min. four copies)*

T^f_2 *(min. two copies)*

T^f_3 *(min. one copy)*

reactions:

$$T^f_1 + 0_1 \rightarrow 1_1 + T^r_1 \quad (1)$$

$$T^f_2 + 0_2 + 1_1 \rightarrow 1_2 + 0_1 + T^r_2 \quad (2)$$

$$T^f_3 + 0_3 + 1_2 + 1_1 \rightarrow 1_3 + 0_2 + 0_1 + T^r_3 \quad (3)$$

counter execution

$0_3\ 0_2\ 0_1$

(1)

$0_3\ 0_2\ 1_1$

(2)

$0_3\ 1_2\ 0_1$

(1)

$0_3\ 1_2\ 1_1$

(3)

$1_3\ 0_2\ 0_1$

(1)

$1_3\ 0_2\ 1_1$

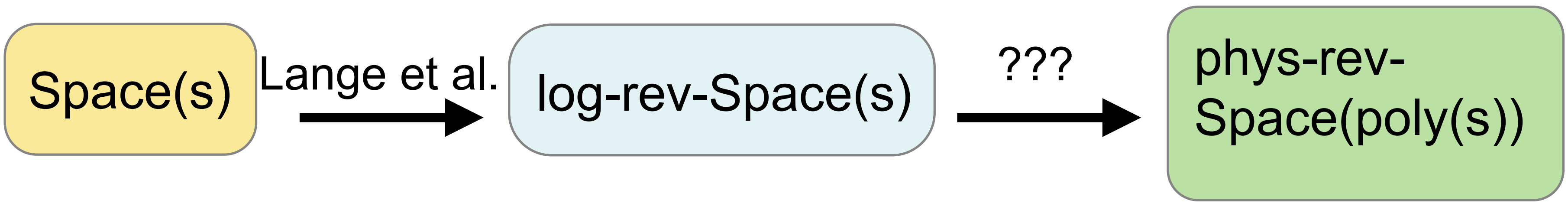
(2)

$1_3\ 1_2\ 0_1$

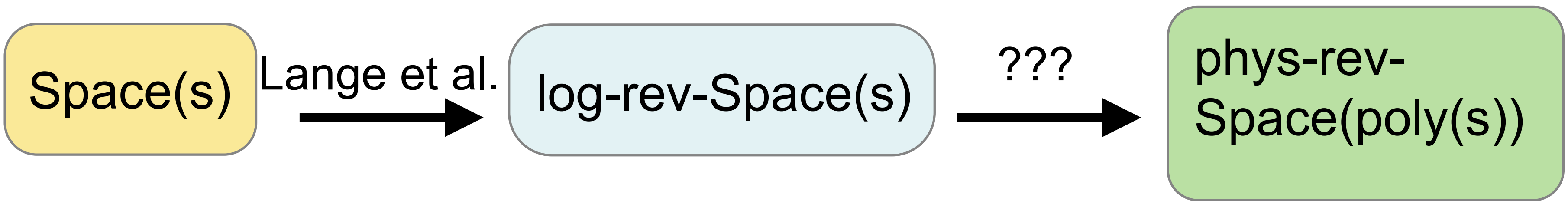
(1)

$1_3\ 1_2\ 1_1$

Towards space- and energy-efficient computations



Towards space- and energy-efficient computations



The Lange et al. construction suffers from the transformer exponential blow-up problem of the traditional binary counter.

Is there a different way to construct a space-efficient, physically reversible counter?

Towards space- and energy-efficient computations

a Grey code counter

initial species: 0_3, 0_2, 0_1 *(one copy each)*

reactions:

$$0_1 \Leftrightarrow 1_1 \qquad (1)$$

$$0_2 + 1_1 \Leftrightarrow 1_2 + 1_1 \qquad (2)$$

$$0_3 + 1_2 + 0_1 \Leftrightarrow 1_3 + 1_2 + 0_1 \qquad (3)$$

counter execution

$0_3\ 0_2\ 0_1$
(1-for)
$0_3\ 0_2\ 1_1$
(2-for)
$0_3\ 1_2\ 1_1$
(1-rev)
$0_3\ 1_2\ 0_1$
(3-for)
$1_3\ 1_2\ 0_1$
(1-for)
$1_3\ 1_2\ 1_1$
(2-rev)
$1_3\ 0_2\ 1_1$
(1-rev)
$1_3\ 0_2\ 0_1$

Condon, Hu, Manuch, Thachuk., J. Royal Soc.

Towards space- and energy-efficient computations

a Grey code counter

accounting for transformer molecules:

initial species: 0_3, 0_2, 0_1 *(one copy each)*

reactions:

$$0_1 \Leftrightarrow 1_1 \qquad (1)$$

$$0_2 + 1_1 \Leftrightarrow 1_2 + 1_1 \qquad (2)$$

$$0_3 + 1_2 + 0_1 \Leftrightarrow 1_3 + 1_2 + 0_1 \qquad (3)$$

counter execution

- $0_3\ 0_2\ 0_1$ (1-for)
- $0_3\ 0_2\ 1_1$ (2-for)
- $0_3\ 1_2\ 1_1$ (1-rev)
- $0_3\ 1_2\ 0_1$ (3-for)
- $1_3\ 1_2\ 0_1$ (1-for)
- $1_3\ 1_2\ 1_1$ (2-rev)
- $1_3\ 0_2\ 1_1$ (1-rev)
- $1_3\ 0_2\ 0_1$

Condon, Hu, Manuch, Thachuk., J. Royal Soc.

Towards space- and energy-efficient computations

a Grey code counter

accounting for transformer molecules:

initial species: 0_3, 0_2, 0_1 *(one copy each)*

reactions:

$$T^f_1 + 0_1 \Leftrightarrow 1_1 + T^r_1 \quad (1)$$

$$T^f_2 + 0_2 + 1_1 \Leftrightarrow 1_2 + 1_1 + T^r_2 \quad (2)$$

$$T^f_3 + 0_3 + 1_2 + 0_1 \Leftrightarrow 1_3 + 1_2 + 0_1 + T^r_3 \quad (3)$$

counter execution

$0_3\ 0_2\ 0_1$
(1-for)
$0_3\ 0_2\ 1_1$
(2-for)
$0_3\ 1_2\ 1_1$
(1-rev)
$0_3\ 1_2\ 0_1$
(3-for)
$1_3\ 1_2\ 0_1$
(1-for)
$1_3\ 1_2\ 1_1$
(2-rev)
$1_3\ 0_2\ 1_1$
(1-rev)
$1_3\ 0_2\ 0_1$

Condon, Hu, Manuch, Thachuk., J. Royal Soc.

Towards space- and energy-efficient computations

a Grey code counter

accounting for transformer molecules:

initial species: 0_3, 0_2, 0_1 *(one copy each)*

T^f_1 *(min. one copy)*
T^f_2 *(min. one copy)*
T^f_3 *(min. one copy)*

reactions:

$$T^f_1 + 0_1 \Leftrightarrow 1_1 + T^r_1 \quad (1)$$

$$T^f_2 + 0_2 + 1_1 \Leftrightarrow 1_2 + 1_1 + T^r_2 \quad (2)$$

$$T^f_3 + 0_3 + 1_2 + 0_1 \Leftrightarrow 1_3 + 1_2 + 0_1 + T^r_3 \quad (3)$$

counter execution

$0_3\ 0_2\ 0_1$
(1-for)
$0_3\ 0_2\ 1_1$
(2-for)
$0_3\ 1_2\ 1_1$
(1-rev)
$0_3\ 1_2\ 0_1$
(3-for)
$1_3\ 1_2\ 0_1$
(1-for)
$1_3\ 1_2\ 1_1$
(2-rev)
$1_3\ 0_2\ 1_1$
(1-rev)
$1_3\ 0_2\ 0_1$

Condon, Hu, Manuch, Thachuk., J. Royal Soc.

Towards space- and energy-efficient computations

a Grey code counter

accounting for transformer molecules:
transformer molecules are recycled!

initial species: 0_3, 0_2, 0_1 *(one copy each)*

T^f_1 *(min. one copy)*
T^f_2 *(min. one copy)*
T^f_3 *(min. one copy)*

reactions:

$$T^f_1 + 0_1 \Leftrightarrow 1_1 + T^r_1 \quad (1)$$
$$T^f_2 + 0_2 + 1_1 \Leftrightarrow 1_2 + 1_1 + T^r_2 \quad (2)$$
$$T^f_3 + 0_3 + 1_2 + 0_1 \Leftrightarrow 1_3 + 1_2 + 0_1 + T^r_3 \quad (3)$$

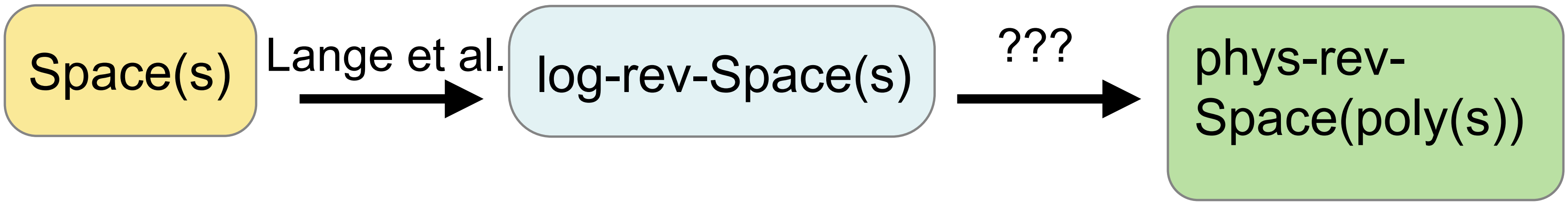
counter execution

$0_3\ 0_2\ 0_1$
(1-for)
$0_3\ 0_2\ 1_1$
(2-for)
$0_3\ 1_2\ 1_1$
(1-rev)
$0_3\ 1_2\ 0_1$
(3-for)
$1_3\ 1_2\ 0_1$
(1-for)
$1_3\ 1_2\ 1_1$
(2-rev)
$1_3\ 0_2\ 1_1$
(1-rev)
$1_3\ 0_2\ 0_1$

Condon, Hu, Manuch, Thachuk., J. Royal Soc.

Towards space- and energy-efficient computations

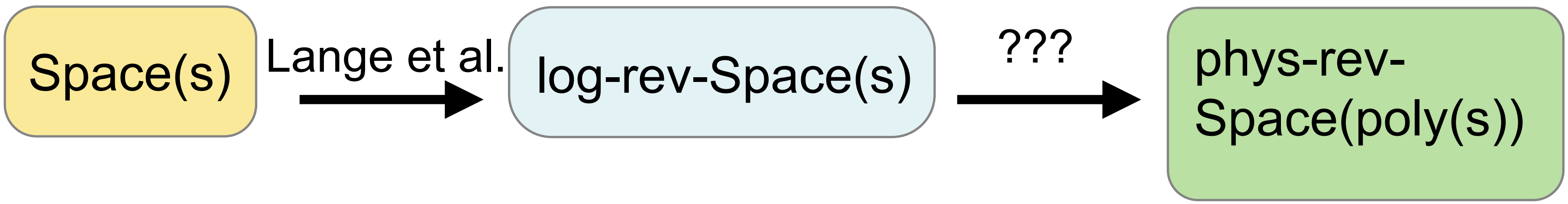
Towards space- and energy-efficient computations



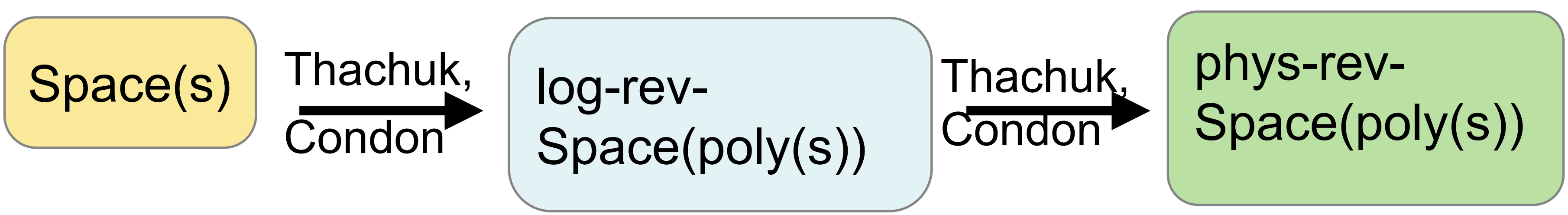
The Lange et al. construction suffers from the transformer exponential blow-up of the traditional binary counter.

Lange, McKenzie, and Tapp, JCSS 2000

Towards space- and energy-efficient computations



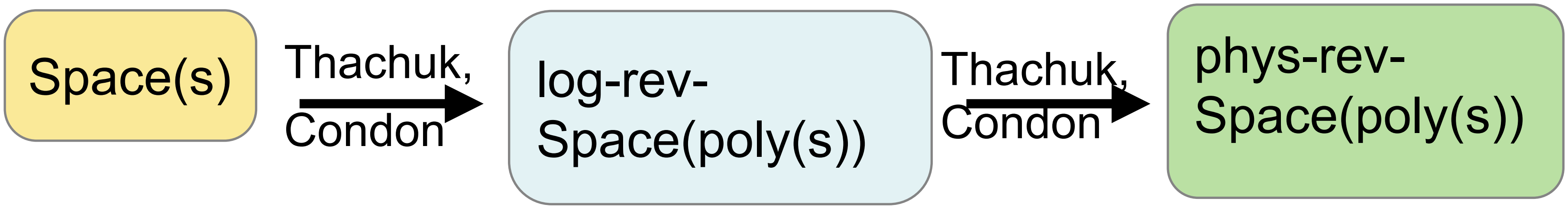
The Lange et al. construction suffers from the transformer exponential blow-up of the traditional binary counter.



Fortunately, building on the grey code counter, a space-efficient compilation of a space-bounded CRN to a physically-reversible DSD is possible.

Thachuk, Condon, DNA18, 2012

Towards space- and energy-efficient computations



Key ideas:

- A CRN with O(n) species can check the truth of a Quantified SAT instance with n variables

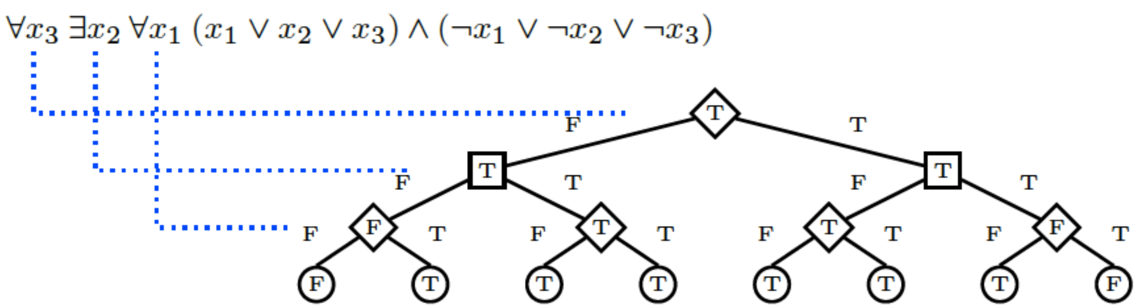
Thachuk, Condon, DNA18, 2012

Towards space- and energy-efficient computations

Space(s) → (Thachuk, Condon) → log-rev-Space(poly(s)) → (Thachuk, Condon) → phys-rev-Space(poly(s))

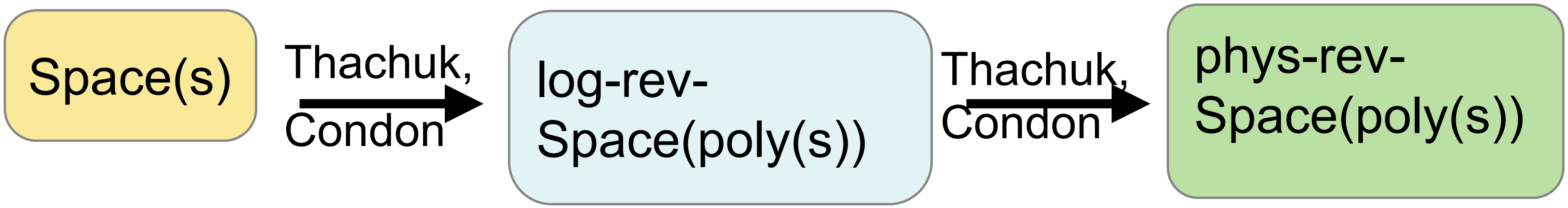
Key ideas:

- A CRN with O(n) species can check the truth of a Quantified SAT instance with n variables



Thachuk, Condon, DNA18, 2012

Towards space- and energy-efficient computations

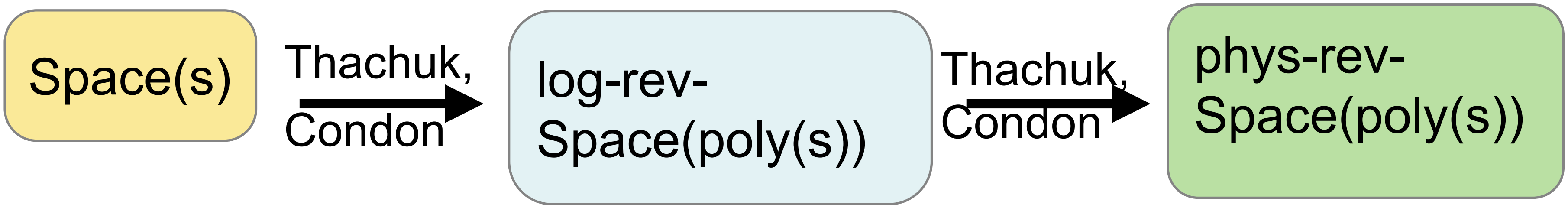


Key ideas:

- A CRN with O(n) species can check the truth of a Quantified SAT instance with n variables

Thachuk, Condon, DNA18, 2012

Towards space- and energy-efficient computations

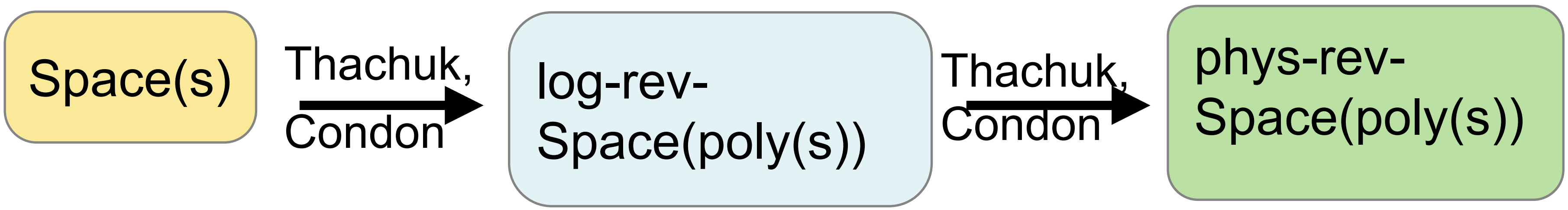


Key ideas:

- A CRN with O(n) species can check the truth of a Quantified SAT instance with n variables
- Concentrations of “minor reactants” (transformers) are the same, so forward and reverse reactions are equally likely

Thachuk, Condon, DNA18, 2012

Towards space- and energy-efficient computations



Key ideas:

- A CRN with O(n) species can check the truth of a Quantified SAT instance with n variables
- Concentrations of "minor reactants" (transformers) are the same, so forward and reverse reactions are equally likely
- It's easy to adapt the construction (by doubling the computation length) so that once the output bit is produced, it's present half of the time

Thachuk, Condon, DNA18, 2012

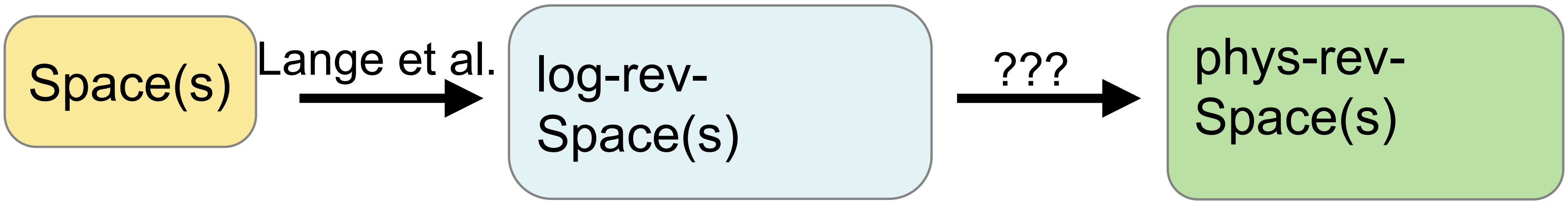
Summary

Logically and physically reversible simulations of irreversible computations are necessary for energy-efficient computations.

Using reactions *in both directions* to advance a computation in a logically reversible way seems useful in facilitating physically reversible, space-efficient computations.

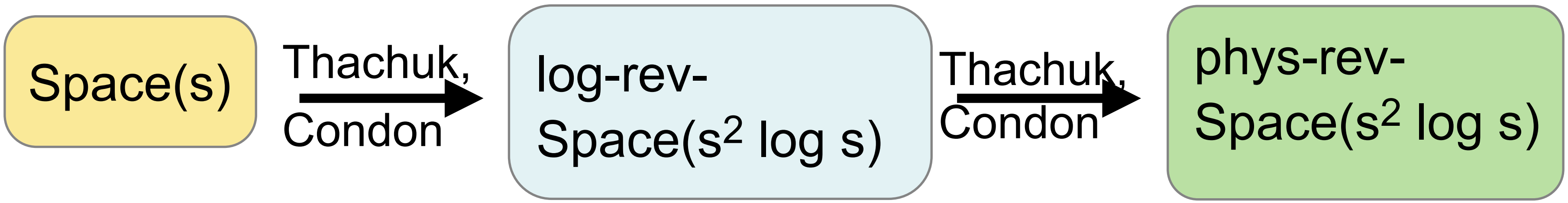
Open questions

Open questions



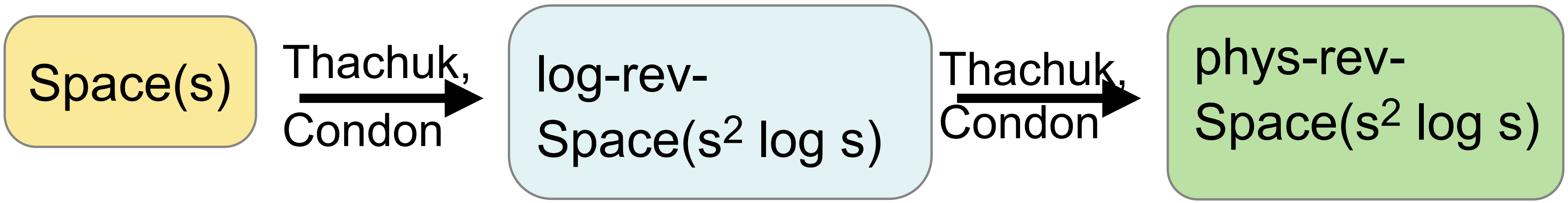
Is there a compiler from CRNs to DSDs, or to an alternative physically reversible DNA computing model, that does not suffer from the exponential blow-up problem?

Open questions

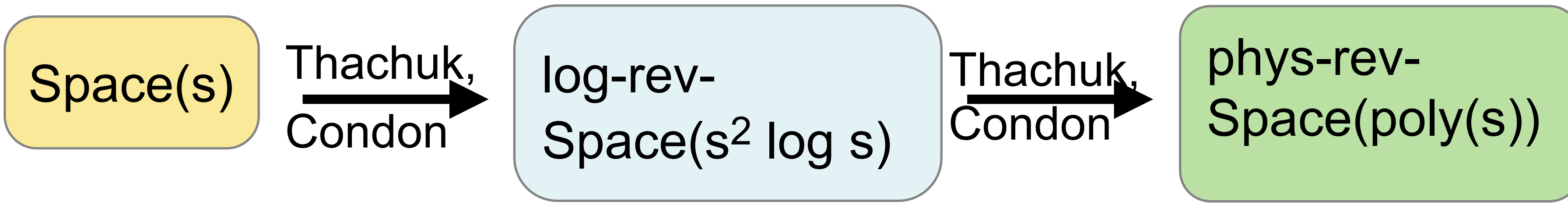


Is it possible to avoid the polynomial increase in space here? Or at least remove the log s factor?

Open questions



Open questions



A logically reversible computation is *k-balanced* if, within every computation prefix, the number of times that the transition is executed in the forwards direction differs from the number of times that the transition is executed in the reverse direction by at most *k*.

If BalancedSPACE(s(n)) is the class of languages recognizable by O(1)-balanced, logically reversible Turing machines, can we show that DSPACE(s(n)) = BalancedSPACE(s(n))?

Open questions

initial species: 0_3, 0_2, 0_1 *(one copy each)*

T^f_1 *(min. four copies)*

T^f_2 *(min. two copies)*

T^f_3 *(min. one copy)*

reactions:

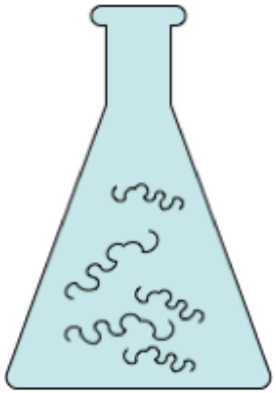
$$T^f_1 + 0_1 \rightarrow 1_1 + T^r_1 \quad (1)$$

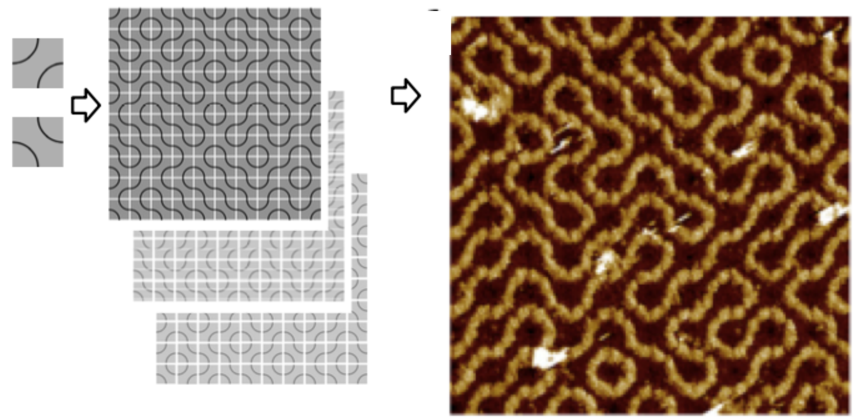
$$T^f_2 + 0_2 + 1_1 \rightarrow 1_2 + 0_1 + T^r_2 \quad (2)$$

$$T^f_3 + 0_3 + 1_2 + 1_1 \rightarrow 1_3 + 0_2 + 0_1 + T^r_3 \quad (3)$$

Can forward and backwards transformers be interconverted?

Thank you!





"Energy permits things to exist and to act, but programming permits things to be purposeful"

- Ware