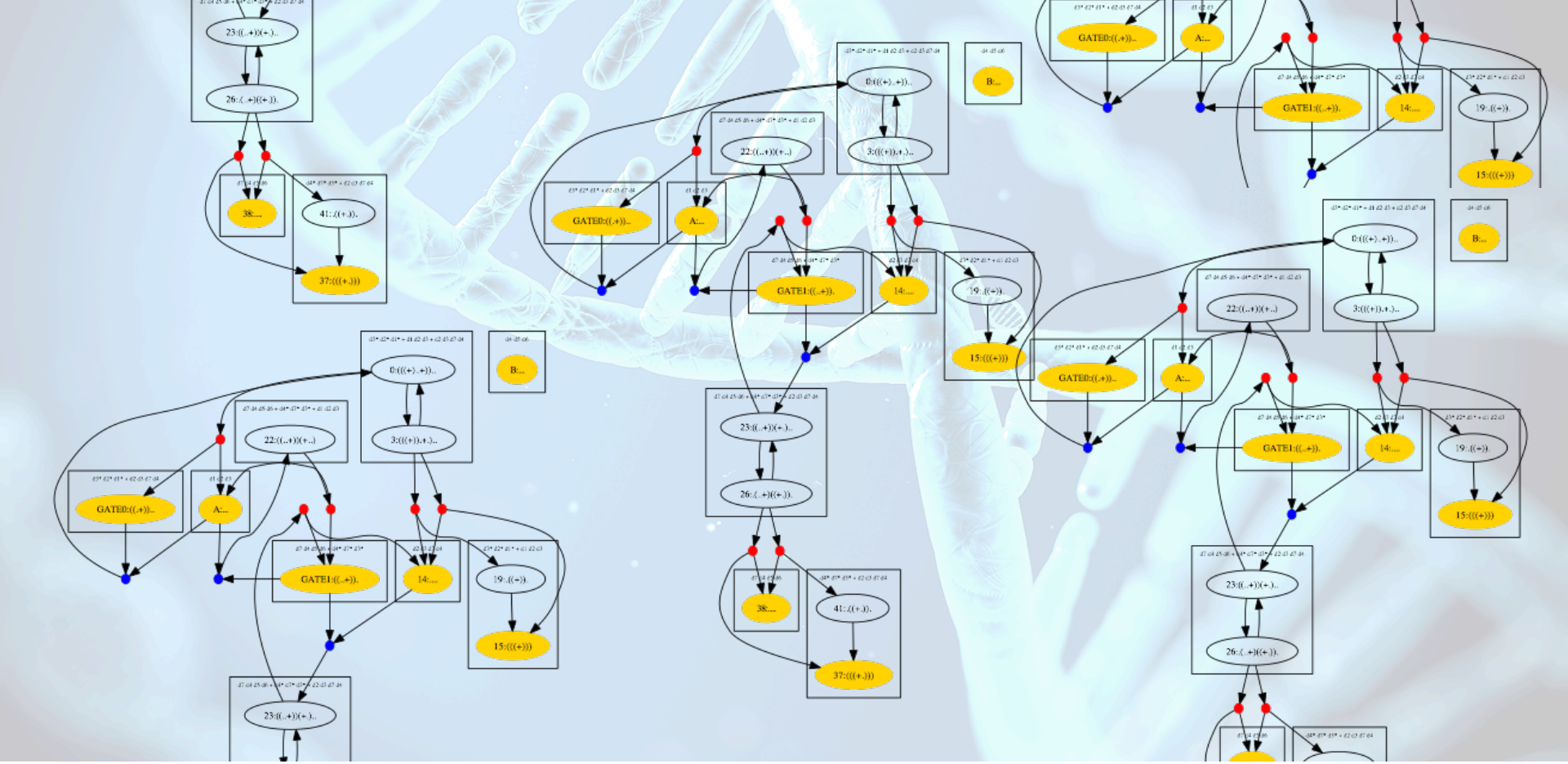


Verifying CRN implementations

Seung Woo Shin

UC Berkeley

BIRS Workshop 6/12/2014

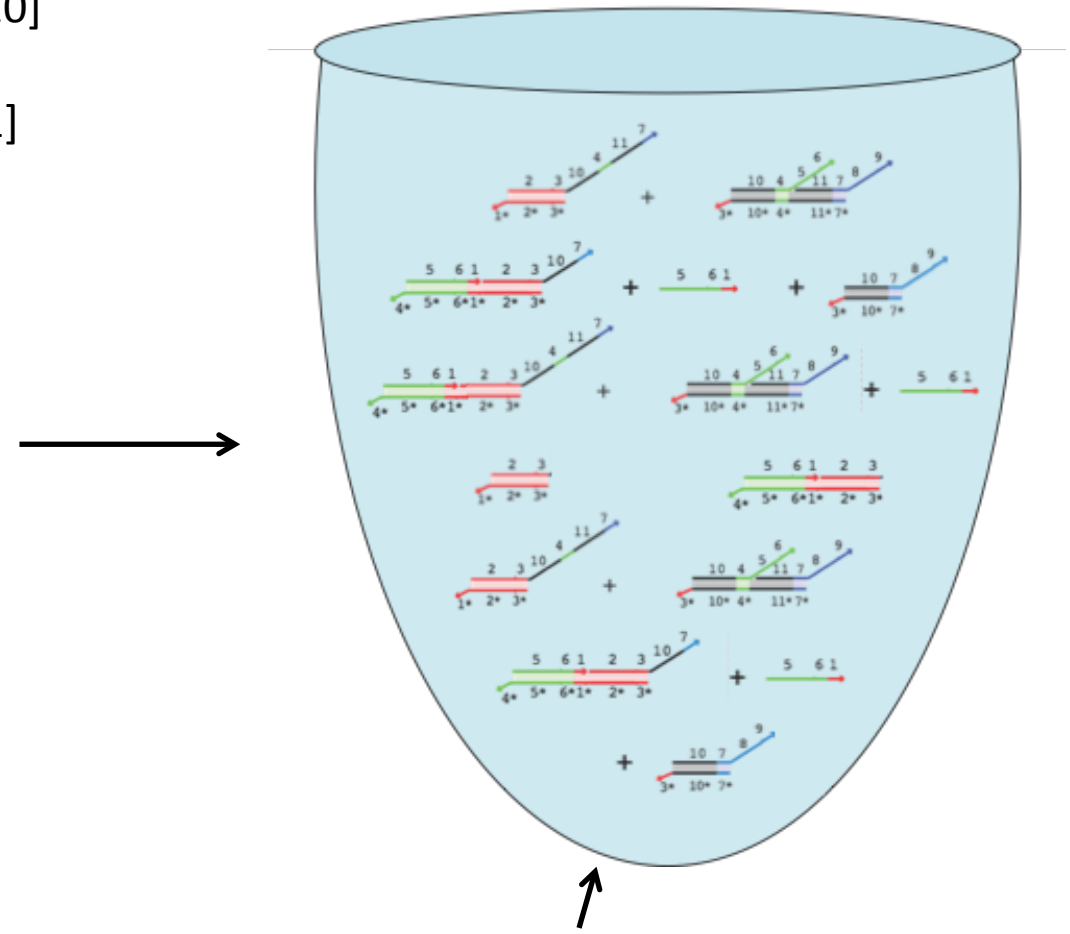
CRN implementation verification

[Soloveichik, Seelig, Winfree 2010]
[Cardelli 2011]
[Qian, Soloveichik, Winfree 2011]

$$X3 + X4 \xrightarrow{k_1} X5$$

$$X5 \xrightarrow{k_2} X1$$

$$X1 + X2 \xrightarrow{k_3} X3$$



There is a CRN that describes this system.

CRN implementation verification

$$X3 + X4 \xrightarrow{k_1} X5$$
$$X5 \xrightarrow{k_2} X1$$
$$X1 + X2 \xrightarrow{k_3} X3$$

$$X3 + g1 \xrightarrow{k_4} i + g2$$
$$i + g2 \xrightarrow{k_5} X3 + g1$$
$$i + x4 \xrightarrow{k_6} j + w1$$
$$j + g3 \xrightarrow{k_7} X5 + w2$$
$$X5 + g4 \xrightarrow{k_8} k + w3$$
$$k + g5 \xrightarrow{k_9} X1 + w4$$
$$X1 + g6 \xrightarrow{k_{10}} l + g7$$
$$l + g7 \xrightarrow{k_{11}} X1 + g6$$
$$l + X2 \xrightarrow{k_{12}} m + w5$$
$$m + g8 \xrightarrow{k_{13}} X3 + w6$$

Can be done using "**reaction enumerators**"

[Grun, Sarma, Wolfe, S, Winfree '14]
Visual DSD [Lakin, Youssef, Polo, Emmott, Phillips '11]

CRN implementation verification

Q: Can we develop a mathematical notion of "equivalence" between the original CRN and the implementation CRN?

$$X3 + X4 \xrightarrow{k_1} X5$$
$$X5 \xrightarrow{k_2} X1$$
$$X1 + X2 \xrightarrow{k_3} X3$$

$\stackrel{?}{=}$

$$X3 + g1 \xrightarrow{k_4} i + g2$$
$$i + g2 \xrightarrow{k_5} X3 + g1$$
$$i + x4 \xrightarrow{k_6} j + w1$$
$$j + g3 \xrightarrow{k_7} X5 + w2$$
$$X5 + g4 \xrightarrow{k_8} k + w3$$
$$k + g5 \xrightarrow{k_9} X1 + w4$$
$$X1 + g6 \xrightarrow{k_{10}} l + g7$$
$$l + g7 \xrightarrow{k_{11}} X1 + g6$$
$$l + X2 \xrightarrow{k_{12}} m + w5$$
$$m + g8 \xrightarrow{k_{13}} X3 + w6$$

Can be done using "**reaction enumerators**"

[Grun, Sarma, Wolfe, S, Winfree '14]
Visual DSD [Lakin, Youssef, Polo, Emmott, Phillips '11]

Why do we care?

Obviously, immediate motivation is verification of CRN implementations.

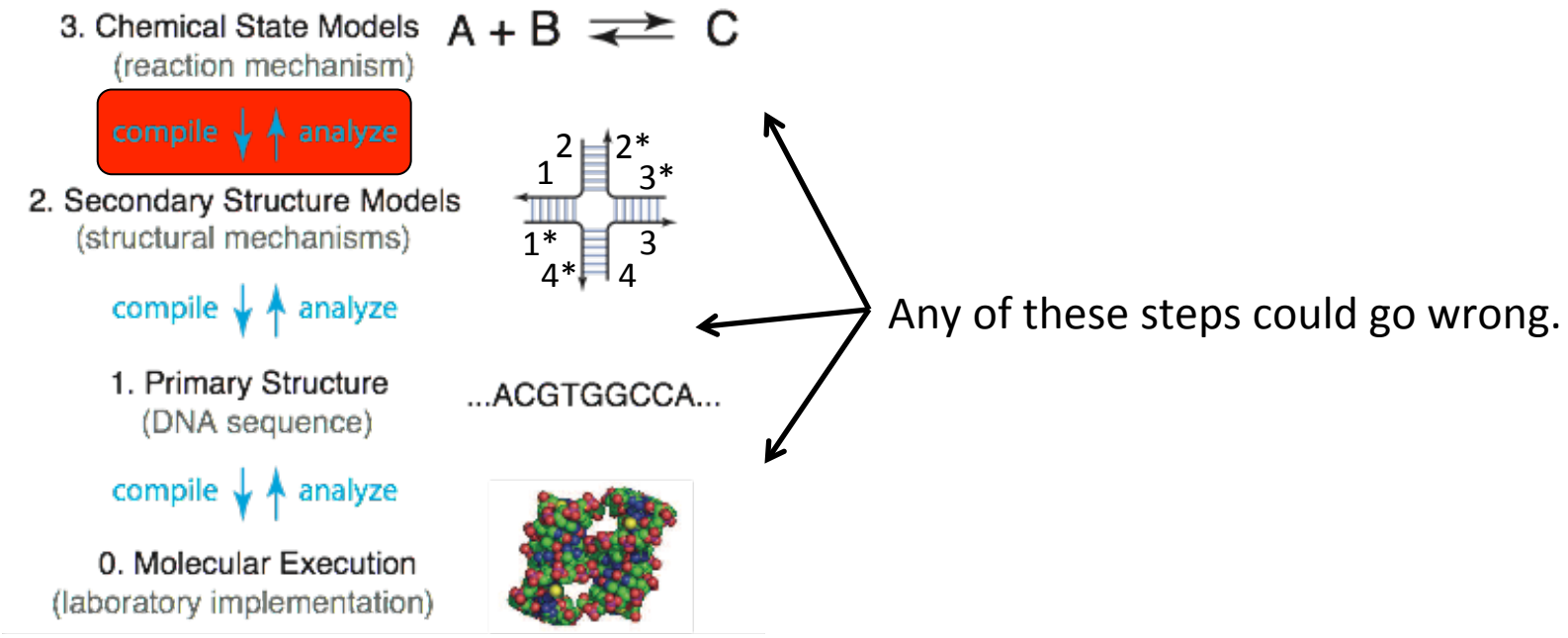


Figure: Niles Pierce

Why do we care?

But can I not just do this by hand?

$$X3 + X4 \xrightarrow{k_1} X5$$
$$X5 \xrightarrow{k_2} X1$$
$$X1 + X2 \xrightarrow{k_3} X3$$

?

$$X3 + g1 \xrightarrow{k_4} i + g2$$
$$i + g2 \xrightarrow{k_5} X3 + g1$$
$$i + x4 \xrightarrow{k_6} j + w1$$
$$j + g3 \xrightarrow{k_7} X5 + w2$$
$$X5 + g4 \xrightarrow{k_8} k + w3$$
$$k + g5 \xrightarrow{k_9} X1 + w4$$
$$X1 + g6 \xrightarrow{k_{10}} l + g7$$
$$l + g7 \xrightarrow{k_{11}} X1 + g6$$
$$l + X2 \xrightarrow{k_{12}} m + w5$$
$$m + g8 \xrightarrow{k_{13}} X3 + w6$$

No. Formalization and automation seems *necessary*.

Life ain't easy…

Reactions
GATE0 + X2 -> 0
GATE0 + X3 -> 1
GATE0 + X1 -> 2
GATE0 + X4 -> 3
GATE0 + X5 -> 4
4 -> GATE0 + X5
3 -> GATE0 + X4
2 -> GATE0 + X1
1 -> 19
1 -> GATE0 + X3
19 -> 1
19 -> GATE9 + 30
0 -> GATE0 + X2
30 + X2 -> 35
30 + X3 -> 36
30 + X1 -> 37
30 + X4 -> 38
30 + X5 -> 39
39 -> X5 + 30
38 -> 46
38 -> X4 + 30
46 -> 38
46 -> 56 + GATE2
37 -> X1 + 30
36 -> X3 + 30
35 -> X2 + 30
56 + X2 -> 70
56 + X3 -> 71
56 + X1 -> 72
56 + X4 -> 73
56 + X5 -> 74
74 -> 56 + X5
73 -> 56 + X4
72 -> 56 + X1
71 -> 56 + X3
70 -> 56 + X2
GATE1 + GATE0 -> 95
GATE1 + 30 -> 96
GATE1 + 56 -> 97
97 -> 100
97 -> 56 + GATE1
100 -> 97
100 -> 110 + X5
96 -> GATE1 + 30
95 -> GATE0 + GATE1
110 + X2 -> 120
110 + X3 -> 121
110 + X1 -> 122
110 + X4 -> 123
110 + X5 -> 100
110 + GATE1 -> 125
125 -> 110 + GATE1
123 -> 110 + X4
122 -> 110 + X1
121 -> 110 + X3
120 -> 110 + X2
GATE2 + GATE0 -> 146
GATE2 + 30 -> 147
GATE2 + 56 -> 46
GATE2 + 110 -> 149
149 -> 110 + GATE2
147 -> 30 + GATE2
146 -> GATE0 + GATE2
GATE3 + X2 -> 162
GATE3 + X3 -> 163
GATE3 + X1 -> 164
GATE3 + X4 -> 165
GATE3 + X5 -> 166
GATE3 + GATE1 -> 167
GATE3 + GATE2 -> 168
168 -> GATE3 + GATE2
167 -> GATE1 + GATE3
166 -> X5 + GATE3
165 -> GATE3 + X4
164 -> 187
164 -> X1 + GATE3
187 -> 164
187 -> 197 + GATE5
163 -> X3 + GATE3
162 -> X2 + GATE3
197 + X2 -> 207
197 + X3 -> 208
197 + X1 -> 209
197 + X4 -> 210
197 + X5 -> 211
197 + GATE1 -> 212
197 + GATE2 -> 213
213 -> 197 + GATE2
212 -> 197 + GATE1
211 -> 197 + X5
210 -> 197 + X4
209 -> 197 + X1
208 -> 197 + X3
207 -> 240
207 -> 197 + X2
240 -> 207
240 -> 250 + GATE13
250 + X2 -> 252
250 + X3 -> 253
250 + X1 -> 254
250 + X4 -> 255
250 + X5 -> 256
250 + GATE1 -> 257
250 + GATE2 -> 258
258 -> 250 + GATE2
257 -> 250 + GATE1
256 -> 250 + X5
255 -> 250 + X4
254 -> 250 + X1
253 -> 250 + X3
252 -> 250 + X2
GATE4 + GATE0 -> 287
GATE4 + 30 -> 288
GATE4 + 56 -> 289
GATE4 + 110 -> 290
GATE4 + GATE3 -> 291
GATE4 + 197 -> 292
GATE4 + 250 -> 293
293 -> 250 + GATE4
292 -> 197 + GATE4
291 -> GATE4 + GATE3
290 -> 110 + GATE4
289 -> 56 + GATE4
288 -> GATE4 + 30
287 -> GATE0 + GATE4
GATE5 + GATE0 -> 322
GATE5 + 30 -> 323
GATE5 + 56 -> 324
GATE5 + 110 -> 325
GATE5 + GATE3 -> 326
GATE5 + 197 -> 187
GATE5 + 250 -> 328
328 -> 250 + GATE5
326 -> GATE3 + GATE5
325 -> 110 + GATE5
324 -> 56 + GATE5
323 -> GATE5 + 30
322 -> GATE0 + GATE5
GATE6 + GATE0 -> 353
GATE6 + 30 -> 354
GATE6 + 56 -> 355
GATE6 + 110 -> 356
GATE6 + GATE3 -> 357
GATE6 + 197 -> 358
GATE6 + 250 -> 359
359 -> 362
359 -> 250 + GATE6
362 -> 359
362 -> 372 + X3
358 -> 197 + GATE6
357 -> GATE6 + GATE3
356 -> 110 + GATE6
355 -> 56 + GATE6
354 -> GATE6 + 30
353 -> GATE0 + GATE6
372 + X2 -> 398
372 + X3 -> 362
372 + X1 -> 400
372 + X4 -> 401
372 + X5 -> 402
372 + GATE1 -> 403
372 + GATE2 -> 404
372 + GATE4 -> 405
372 + GATE5 -> 406
372 + GATE6 -> 407
407 -> 372 + GATE6
406 -> 372 + GATE5
405 -> 372 + GATE4
404 -> 372 + GATE2
403 -> 372 + GATE1
402 -> 372 + X5
401 -> 372 + X4
400 -> 372 + X1
398 -> 372 + X2
GATE7 + GATE0 -> 444
GATE7 + 30 -> 445
GATE7 + 56 -> 446
GATE7 + 110 -> 447
GATE7 + GATE3 -> 448
GATE7 + 197 -> 449
GATE7 + 250 -> 450
GATE7 + 372 -> 451
451 -> 372 + GATE7
450 -> 250 + GATE7
449 -> 197 + GATE7
448 -> GATE7 + GATE3
447 -> 110 + GATE7
446 -> 56 + GATE7
445 -> GATE7 + 30
444 -> GATE0 + GATE7
GATE8 + GATE0 -> 484
GATE8 + 30 -> 485
GATE8 + 56 -> 486
GATE8 + 110 -> 487
GATE8 + GATE3 -> 488
GATE8 + 197 -> 489
GATE8 + 250 -> 490
GATE8 + 372 -> 491
491 -> 372 + GATE8
490 -> 250 + GATE8
489 -> 197 + GATE8
488 -> GATE8 + GATE3
487 -> 511 + 512
487 -> 110 + GATE8
486 -> 56 + GATE8
485 -> GATE8 + 30
484 -> GATE0 + GATE8
GATE9 + GATE0 -> 529
GATE9 + 30 -> 19
GATE9 + 56 -> 531
GATE9 + 110 -> 532
GATE9 + GATE3 -> 533
GATE9 + 197 -> 534
GATE9 + 250 -> 535
GATE9 + 372 -> 536
536 -> 372 + GATE9
535 -> 250 + GATE9
534 -> 197 + GATE9
533 -> GATE9 + GATE3
532 -> 110 + GATE9
531 -> 56 + GATE9
529 -> GATE0 + GATE9
GATE10 + GATE0 -> 565
GATE10 + 30 -> 566
GATE10 + 56 -> 567
GATE10 + 110 -> 568
GATE10 + GATE3 -> 569
GATE10 + 197 -> 570
GATE10 + 250 -> 571
GATE10 + 372 -> 572
572 -> 576 + 577
572 -> 372 + GATE10
571 -> 250 + GATE10
570 -> 197 + GATE10
569 -> GATE10 + GATE3
568 -> 110 + GATE10
567 -> 56 + GATE10
566 -> GATE10 + 30
565 -> GATE0 + GATE10
GATE11 + X2 -> 610
GATE11 + X3 -> 611
GATE11 + X1 -> 612
GATE11 + X4 -> 613
GATE11 + X5 -> 614
GATE11 + GATE1 -> 615
GATE11 + GATE2 -> 616
GATE11 + GATE4 -> 617
GATE11 + GATE5 -> 618
GATE11 + GATE6 -> 619
GATE11 + GATE7 -> 620
GATE11 + GATE8 -> 621
GATE11 + GATE9 -> 622
GATE11 + GATE10 -> 623
623 -> GATE11 + GATE10
622 -> GATE11 + GATE9
621 -> GATE11 + GATE8
620 -> GATE11 + GATE7
619 -> GATE11 + GATE6
618 -> GATE11 + GATE5
617 -> GATE11 + GATE4
616 -> GATE11 + GATE2
615 -> GATE11 + GATE1
614 -> 662
614 -> GATE11 + X5
662 -> 614
662 -> GATE12 + 673
613 -> GATE11 + X4
612 -> GATE11 + X1
611 -> GATE11 + X3
610 -> GATE11 + X2
673 + X2 -> 690
673 + X3 -> 691
673 + X1 -> 692
673 + X4 -> 693
673 + X5 -> 694
673 + GATE1 -> 695
673 + GATE2 -> 696
673 + GATE4 -> 697
673 + GATE5 -> 698
673 + GATE6 -> 699
673 + GATE7 -> 700
673 + GATE8 -> 701
673 + GATE9 -> 702
673 + GATE10 -> 703
703 -> GATE10 + 673
702 -> GATE9 + 673
701 -> GATE8 + 673
700 -> GATE7 + 673
699 -> GATE6 + 673
698 -> 673 + GATE5
697 -> 730
697 -> GATE4 + 673
730 -> 697
730 -> 740 + X1
696 -> 673 + GATE2
695 -> GATE1 + 673
694 -> X5 + 673
693 -> 673 + X4
692 -> 673 + X1
691 -> X3 + 673
690 -> X2 + 673
740 + X2 -> 770
740 + X3 -> 771
740 + X1 -> 730
740 + X4 -> 773
740 + X5 -> 774
740 + GATE1 -> 775
740 + GATE2 -> 776
740 + GATE4 -> 777
740 + GATE5 -> 778
740 + GATE6 -> 779
740 + GATE7 -> 780
740 + GATE8 -> 781
740 + GATE9 -> 782
740 + GATE10 -> 783
783 -> 740 + GATE10
782 -> 740 + GATE9
781 -> 740 + GATE8
780 -> 799 + 800
780 -> 740 + GATE7
779 -> 740 + GATE6
778 -> 740 + GATE5
777 -> 740 + GATE4
776 -> 740 + GATE2
775 -> 740 + GATE1
774 -> 740 + X5
773 -> 740 + X4
771 -> 740 + X3
770 -> 740 + X2
GATE12 + GATE0 -> 841
GATE12 + 30 -> 842
GATE12 + 56 -> 843
GATE12 + 110 -> 844
GATE12 + GATE3 -> 845
GATE12 + 197 -> 846
GATE12 + 250 -> 847
GATE12 + 372 -> 848
GATE12 + GATE11 -> 849
GATE12 + 673 -> 662
GATE12 + 740 -> 851
851 -> 740 + GATE12
849 -> GATE11 + GATE12
848 -> 372 + GATE12
847 -> 250 + GATE12
846 -> 197 + GATE12
845 -> GATE12 + GATE3
844 -> 110 + GATE12
843 -> 56 + GATE12
842 -> GATE12 + 30
841 -> GATE0 + GATE12
GATE13 + GATE0 -> 892
GATE13 + 30 -> 893
GATE13 + 56 -> 894
GATE13 + 110 -> 895
GATE13 + GATE3 -> 896
GATE13 + 197 -> 897
GATE13 + 250 -> 240
GATE13 + 372 -> 899
GATE13 + GATE11 -> 900
GATE13 + 673 -> 901
GATE13 + 740 -> 902
902 -> 740 + GATE13
901 -> GATE13 + 673
900 -> GATE11 + GATE13
899 -> 372 + GATE13
897 -> 197 + GATE13
896 -> GATE13 + GATE3
895 -> 110 + GATE13
894 -> 56 + GATE13
893 -> GATE13 + 30
892 -> GATE0 + GATE13

Why do we care?

But you are implementing each reaction separately. So why can't I verify them separately?

$$X3 + X4 \xrightarrow{k_1} X5$$

$$X5 \xrightarrow{k_2} X1$$

$$X1 + X2 \xrightarrow{k_3} X3$$

?

$$X3 + g1 \xrightarrow{k_4} i + g2$$

$$i + g2 \xrightarrow{k_5} X3 + g1$$

$$i + x4 \xrightarrow{k_6} j + w1$$

$$j + g3 \xrightarrow{k_7} X5 + w2$$

$$X5 + g4 \xrightarrow{k_8} k + w3$$

$$k + g5 \xrightarrow{k_9} X1 + w4$$

$$X1 + g6 \xrightarrow{k_{10}} l + g7$$

$$l + g7 \xrightarrow{k_{11}} X1 + g6$$

$$l + X2 \xrightarrow{k_{12}} m + w5$$

$$m + g8 \xrightarrow{k_{13}} X3 + w6$$

Yes you can, if there is no crosstalk between modules implementing different reactions.

[Lakin, Phillips, Stefanovic 2013]

Why do we care?

Because of scaling issues that arise in practice, one may want to use an implementation that ***minimizes*** the number of species/reactions used.

E.g.

Target CRN

$$A \to Z$$
$$B \to Z$$
$$C \to Z$$
$$D \to Z$$

Proposed implementation #1

$$A \to i$$
$$i \to Z$$
$$B \to j$$
$$j \to Z$$
$$C \to k$$
$$k \to Z$$
$$D \to l$$
$$l \to Z$$

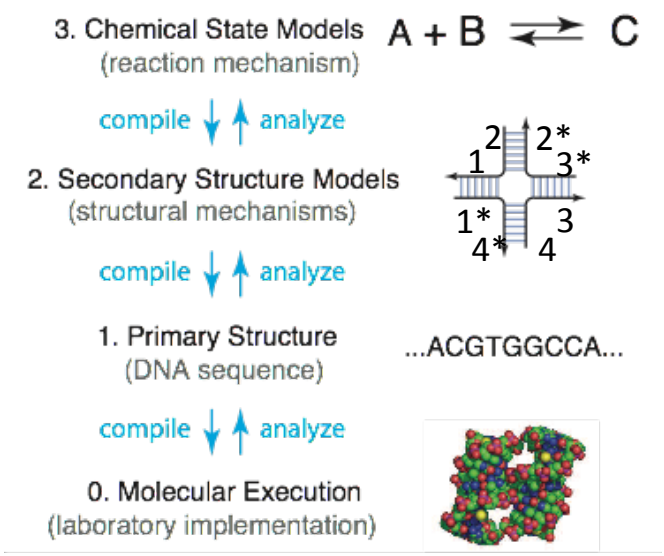
Proposed implementation #2

$$A \to i$$
$$B \to i$$
$$C \to i$$
$$D \to i$$
$$i \to Z$$

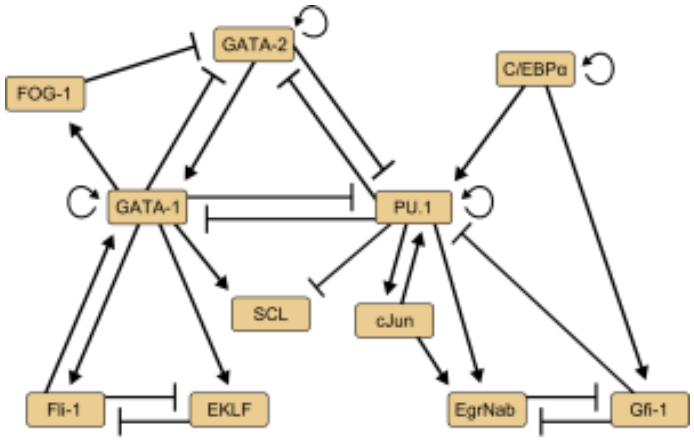
Open question: Optimization techniques for CRN implementations?

Why do we care? (In my wildest dreams . . .)

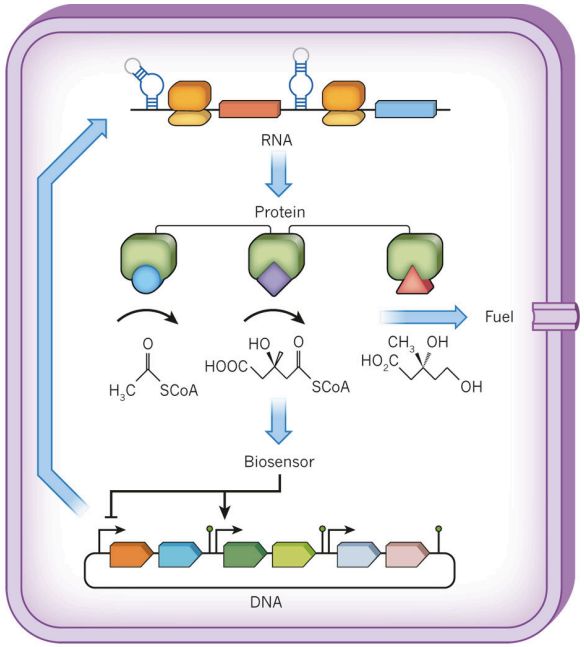
The paradigm of CRN equivalence testing seems to be very general.

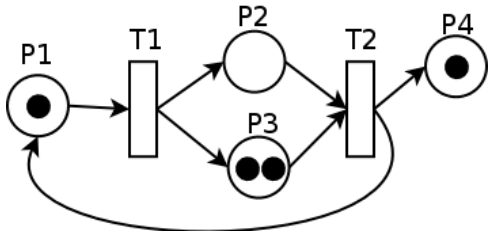


Verify other steps?



Verify other engineered chemical systems?





Any implication in other models of concurrent computation?

Study naturally arising systems?

We ignore chemical kinetics

We want to know whether the two CRNs have the same "logical" behaviors.

$$X3 + X4 \to X5$$
$$X5 \to X1$$
$$X1 + X2 \to X3$$

$\stackrel{?}{=}$

$$X3 + g1 \to i + g2$$
$$i + g2 \to X3 + g1$$
$$i + x4 \to j + w1$$
$$j + g3 \to X5 + w2$$
$$X5 + g4 \to k + w3$$
$$k + g5 \to X1 + w4$$
$$X1 + g6 \to l + g7$$
$$l + g7 \to X1 + g6$$
$$l + X2 \to m + w5$$
$$m + g8 \to X3 + w6$$

Open question: CRN equivalence testing with chemical kinetics? [Lakin et al. '12] [Cardelli '14]

We ignore gate species and waste species

$$X3 + X4 \rightarrow X5$$
$$X5 \rightarrow X1$$
$$X1 + X2 \rightarrow X3$$

$\stackrel{?}{=}$

$$X3 + g1 \rightarrow i + g2$$
$$i + g2 \rightarrow X3 + g1$$
$$i + x4 \rightarrow j + w1$$
$$j + g3 \rightarrow X5 + w2$$
$$X5 + g4 \rightarrow k + w3$$
$$k + g5 \rightarrow X1 + w4$$
$$X1 + g6 \rightarrow l + g7$$
$$l + g7 \rightarrow X1 + g6$$
$$l + X2 \rightarrow m + w5$$
$$m + g8 \rightarrow X3 + w6$$

We ignore gate species and waste species

$$X3 + X4 \rightarrow X5$$
$$X5 \rightarrow X1$$
$$X1 + X2 \rightarrow X3$$

$\stackrel{?}{=}$

$$X3 + g1 \rightarrow i + g2$$
$$i + g2 \rightarrow X3 + g1$$
$$i + x4 \rightarrow j + w1$$
$$j + g3 \rightarrow X5 + w2$$
$$X5 + g4 \rightarrow k + w3$$
$$k + g5 \rightarrow X1 + w4$$
$$X1 + g6 \rightarrow l + g7$$
$$l + g7 \rightarrow X1 + g6$$
$$l + X2 \rightarrow m + w5$$
$$m + g8 \rightarrow X3 + w6$$

We ignore gate species and waste species

Formal CRN

$$X3 + X4 \rightarrow X5$$
$$X5 \rightarrow X1$$
$$X1 + X2 \rightarrow X3$$

?

Implementation CRN

$$X3 \rightarrow i$$
$$i \rightarrow X3$$
$$i + x4 \rightarrow j$$
$$j \rightarrow X5$$
$$X5 \rightarrow k$$
$$k \rightarrow X1$$
$$X1 \rightarrow l$$
$$l \rightarrow X1$$
$$l + X2 \rightarrow m$$
$$m \rightarrow X3$$

Formal species: X1, X2, X3, X4, X5
Intermediate species: I, j, k, l, m

We consider “tidy” CRNs.

Some implementations may get “stuck,” producing an intermediate species that is not really intermediate.

$$A \longrightarrow i$$

$$i + B \longrightarrow C$$

We consider "tidy" CRNs.

Some implementations may get "stuck," producing an intermediate species that is not really intermediate.

$$A \rightleftharpoons i$$

$$i + B \rightarrow C$$

Tidy CRNs clean up all its intermediate species and get back to a formal state.

Formal CRN

$$X3 + X4 \rightarrow X5$$
$$X5 \rightarrow X1$$
$$X1 + X2 \rightarrow X3$$

$\stackrel{?}{=}$

***Tidy* implementation CRN**

$$X3 \rightarrow i$$
$$i \rightarrow X3$$
$$i + x4 \rightarrow j$$
$$j \rightarrow X5$$
$$X5 \rightarrow k$$
$$k \rightarrow X1$$
$$X1 \rightarrow l$$
$$l \rightarrow X1$$
$$l + X2 \rightarrow m$$
$$m \rightarrow X3$$

First attempt: reachability between *formal* states

For every formal initial state, the set of reachable formal states must be exactly the same for the two CRNs.

$$A \to B$$
$$B \to C$$
$$C \to A$$

$$A \to C$$
$$C \to B$$
$$B \to A$$

Besides, reachability between two specific states is an extremely hard problem. (EXPSPACE-hard)

Now we have to test for reachability between every pair of formal states.

Bisimulation approach [Dong 2012]

Inspired by the standard notion of *weak bisimulation* from state transition system literature.

$$A + B \rightarrow C + D$$
$$A \rightarrow E$$

$$A \overset{A}{\rightleftharpoons} i^{A}$$
$$i^{A} + B \overset{B}{\longrightarrow} j^{A+B}$$
$$j^{A+B} \overset{C}{\longrightarrow} C + k^{D}$$
$$k^{D} \overset{D}{\longrightarrow} D$$
$$A \overset{A}{\longrightarrow} E^{E}$$

Two CRNs are bisimulation equivalent if we can find a good "**interpretation**" of the species of the implementation CRN.

Bisimulation approach [Dong 2012]

Inspired by the standard notion of *weak bisimulation* from state transition system literature.

$$A + B \rightarrow C + D$$
$$A \rightarrow E$$

$$A \overset{A}{\rightleftharpoons} i$$
$$i + B \overset{B}{\rightarrow} j \overset{C+D}{}$$
$$j \overset{C+D}{\rightarrow} C + k \overset{C}{\overset{D}{}}$$
$$k \overset{D}{\rightarrow} D \overset{D}{}$$
$$A \overset{A}{\rightarrow} E \overset{E}{}$$

Two CRNs are bisimulation equivalent if we can find a good "**interpretation**" of the species of the implementation CRN.

Bisimulation approach [Dong 2012]

Inspired by the standard notion of *weak bisimulation* from state transition system literature.

Two CRNs are bisimulation equivalent if there exists an "**interpretation**" of the species of the implementation CRN as formal species such that:

1) **Atomic**: For every species X in the formal CRN, there is at least one species in the implementation CRN that is interpreted as X.

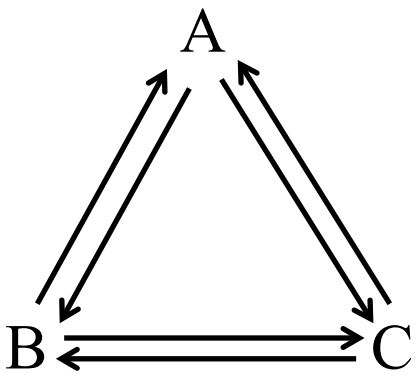
2) **Delimiting**: Under the interpretation, the set of reactions in the two CRNs are exactly the same, up to trivial reactions.

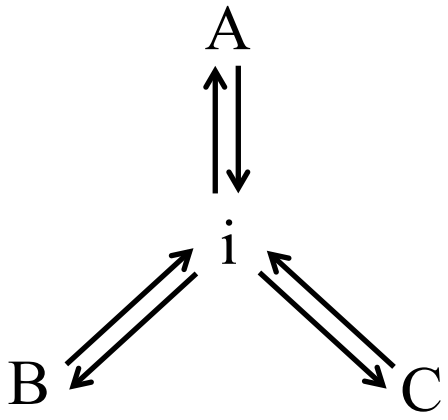
3) **Permissive**: For any state S' of implementation CRN, if S is its interpretation and a formal reaction R can occur in S, then there exists at least one reaction R' in the implementation CRN which can "occur" in S' and whose interpretation is R.

Bisimulation approach [Dong 2012]

Bisimulation approach can handle all existing DNA implementations.

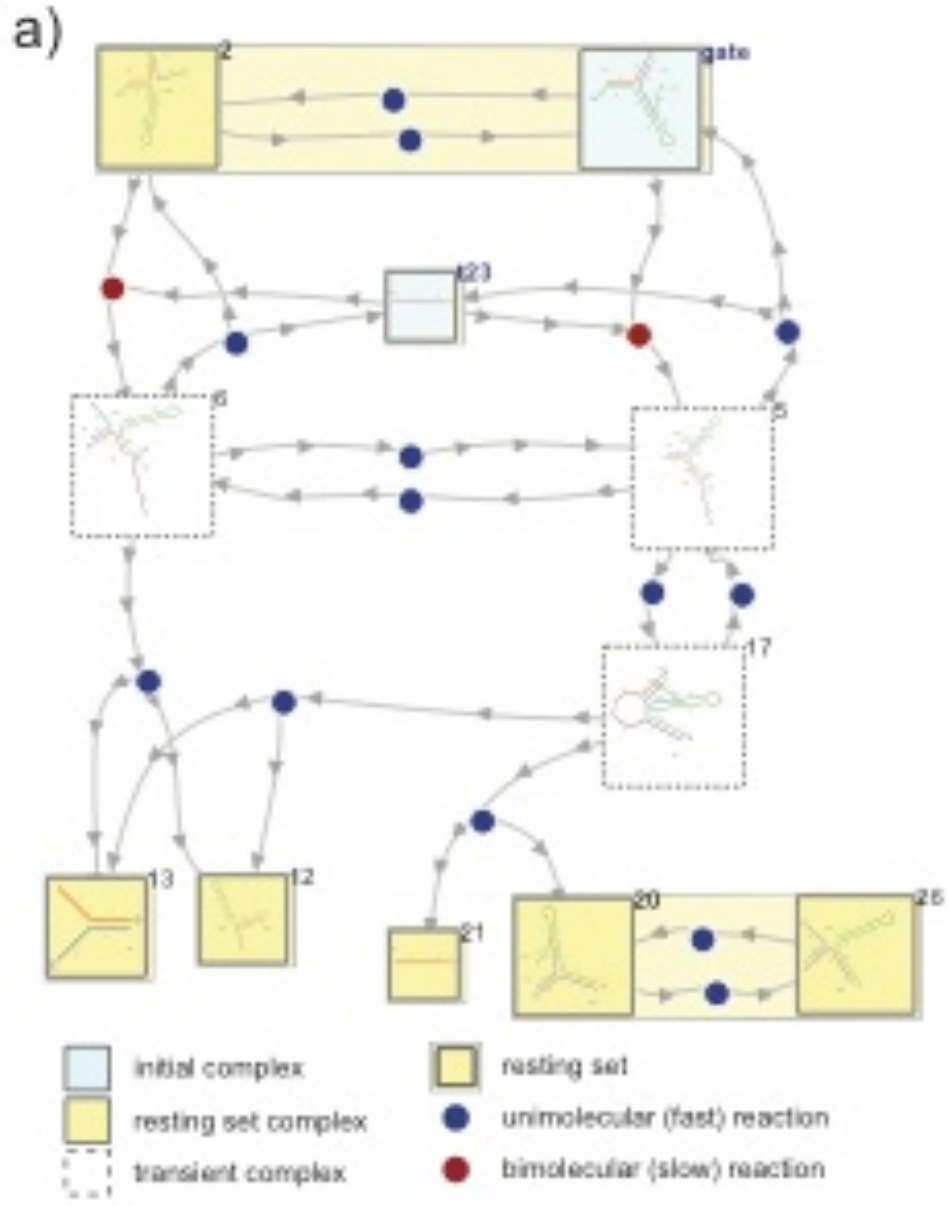
However, in the long run, it may have certain limitations:



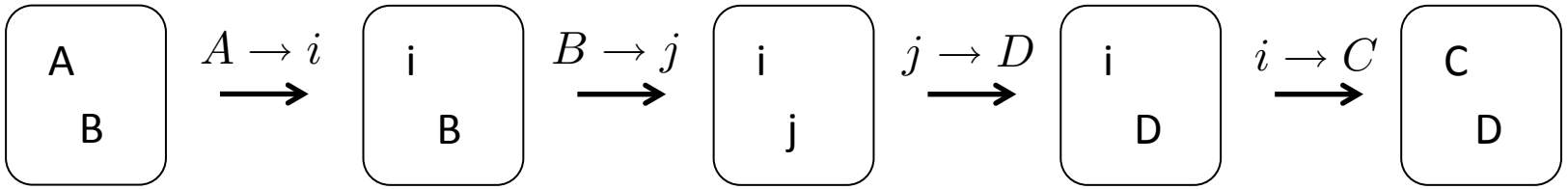


"Delayed choice"

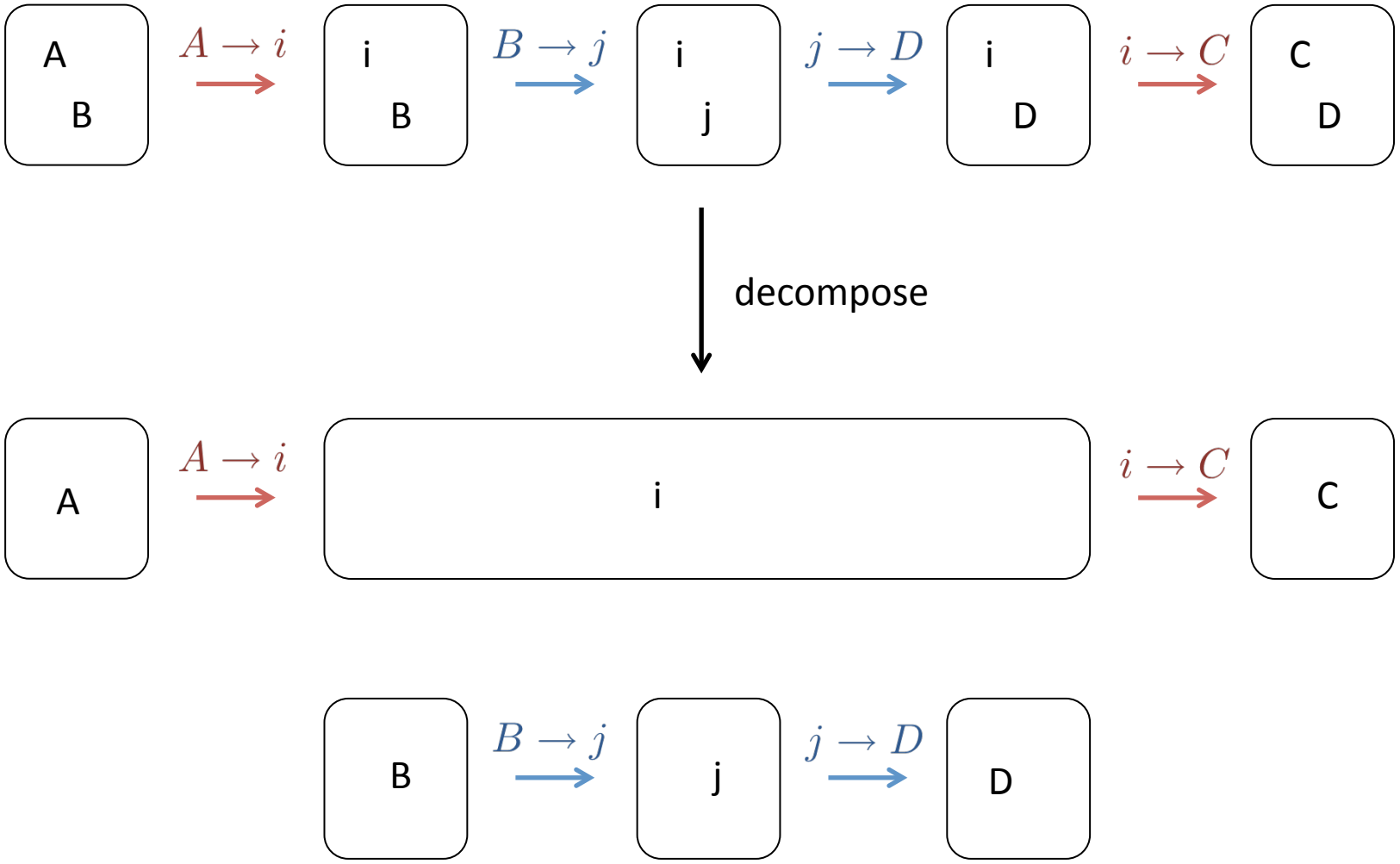
- Seems to be related to optimization of # species and # reactions.
- Can naturally occur in DNA strand displacement systems.


a)
2
gate
23
6
5
17
13
12
21
20
25
initial complex
resting set complex
transient complex
resting set
unimolecular (fast) reaction
bimolecular (slow) reaction

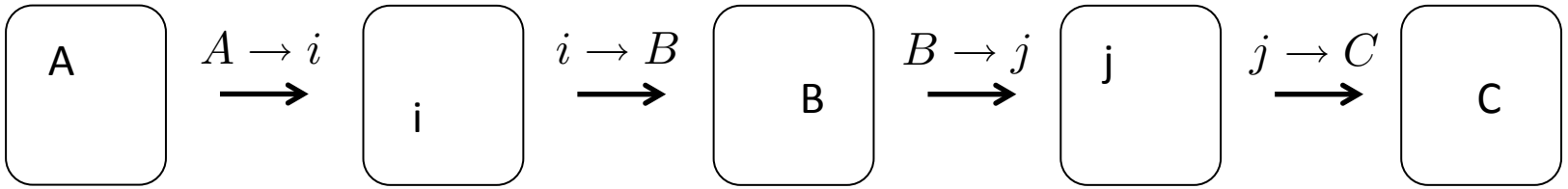
Pathway decomposition approach [S 2011] [S, Thachuk, Winfree 2014]



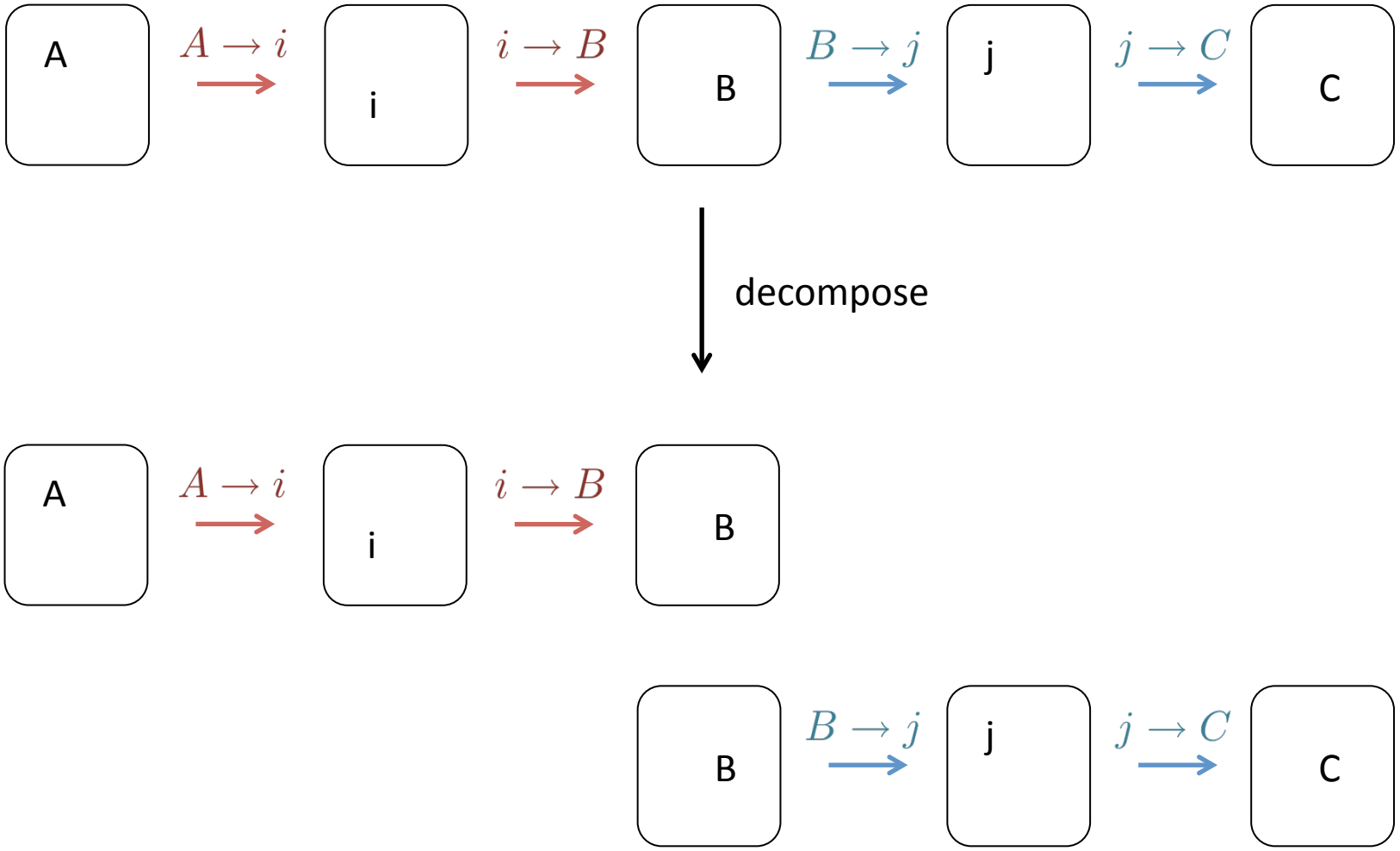
Pathway decomposition approach [S 2011] [S, Thachuk, Winfree 2014]



Pathway decomposition approach [S 2011] [S, Thachuk, Winfree 2014]



Pathway decomposition approach [S 2011] [S, Thachuk, Winfree 2014]



Pathway decomposition approach [S 2011] [S, Thachuk, Winfree 2014]

A formal pathway is **decomposable** if

1. it can be partitioned into two or more formal pathways, or
2. it can be partitioned into a formal pathway and a futile loop.

The **formal basis** of a CRN is defined to be the set of (initial state, final state) pairs of all *un*decomposable formal pathways.

$$A + B \rightarrow C + D$$
$$A \rightarrow E$$

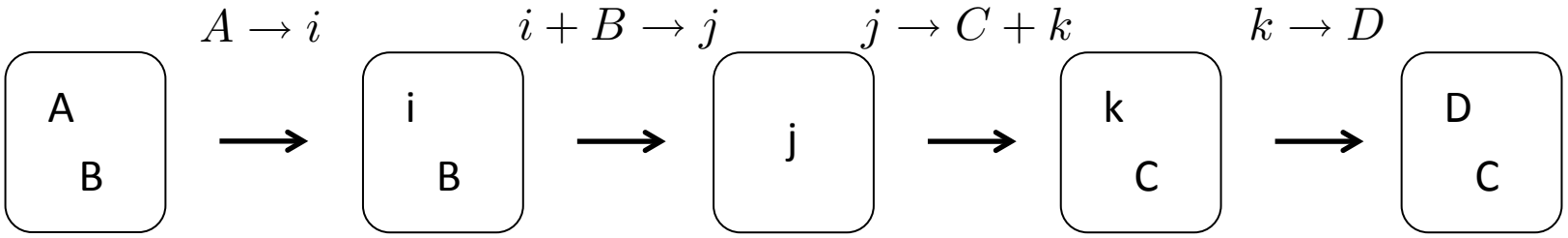
$$A \rightleftharpoons i$$
$$i + B \rightarrow j$$
$$j \rightarrow C + k$$
$$k \rightarrow D$$
$$A \rightarrow E$$

Formal basis

$$A \rightarrow A$$
$$A + B \rightarrow C + D$$
$$A \rightarrow E$$

Pathway decomposition approach [S 2011] [S, Thachuk, Winfree 2014]

A CRN is called **regular** if every undecomposable formal pathway has a clear "**turning point**".



The **formal basis** of a CRN is defined to be the set of (initial state, final state) pairs of all *un*decomposable formal pathways.

$$A + B \to C + D$$
$$A \to E$$

$$A \rightleftharpoons i$$
$$i + B \to j$$
$$j \to C + k$$
$$k \to D$$
$$A \to E$$

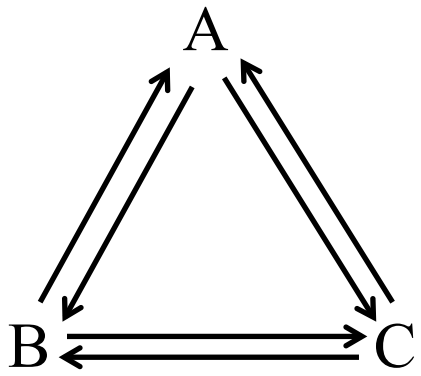
Formal basis

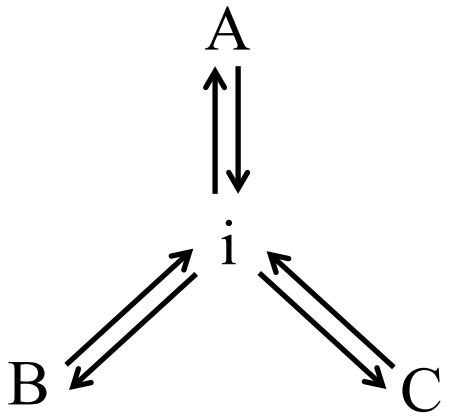
$$A \to A$$
$$A + B \to C + D$$
$$A \to E$$

Pathway decomposition approach [S 2011] [S, Thachuk, Winfree 2014]

A tidy and regular implementation CRN is **pathway decomposition equivalent** to the formal CRN, if its formal basis is equal to the formal CRN up to trivial reactions.

Can handle delayed choice!





Pathway decomposition approach [S 2011] [S, Thachuk, Winfree 2014]

A tidy and regular implementation CRN is **pathway decomposition equivalent** to the formal CRN, if its formal basis is equal to the formal CRN up to trivial reactions.

But there are some implementations that pathway decomposition can't handle while bisimulation can.

Implementations have to "**touch the base.**"

$$A + B \rightleftharpoons C + D$$

$$\begin{aligned} A &\rightleftharpoons i \\ i + B &\rightleftharpoons j \\ j &\rightleftharpoons k + C \\ k &\rightleftharpoons D \end{aligned}$$

[Qian, Soloveichik, Winfree 2011]

Open problems

1. Understand existing notions better.
2. “Hybrid approach” [S, Thachuk, Winfree 2014]
3. Can we incorporate chemical kinetics?
4. Justify simplification of reaction enumeration semantics?
5. Verification in CRNs that involve polymers?
6. Computational complexity of testing equivalence notions? Faster algorithms? [Johnson 2014] [Hesse, work in progress]
7. Optimization of CRN implementations? Delayed choice phenomenon?
8. Are these theories applicable in other related areas?
9. Category theory?