

A GENERAL-PURPOSE CRN-TO-DSD COMPILER WITH FORMAL VERIFICATION, OPTIMIZATION, AND SIMULATION CAPABILITIES

Stefan Badelt, Seung Woo Shin, Robert F. Johnson,
Qing Dong, Chris Thachuk, and Erik Winfree

DNA and Natural Algorithms (DNA) Group, Caltech

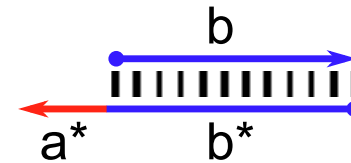
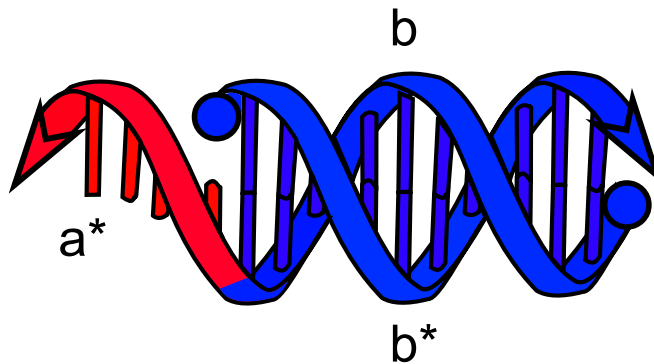
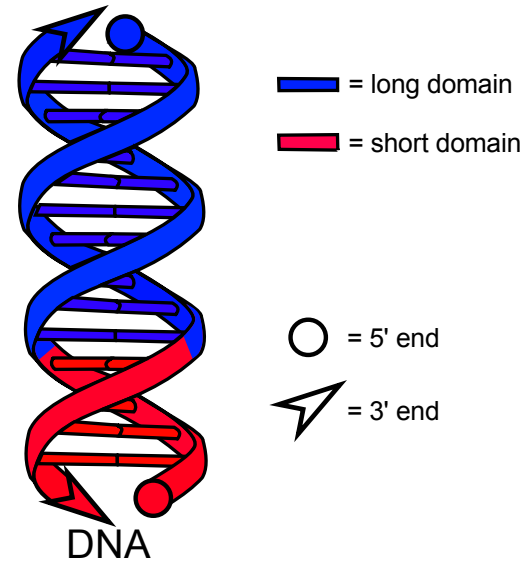
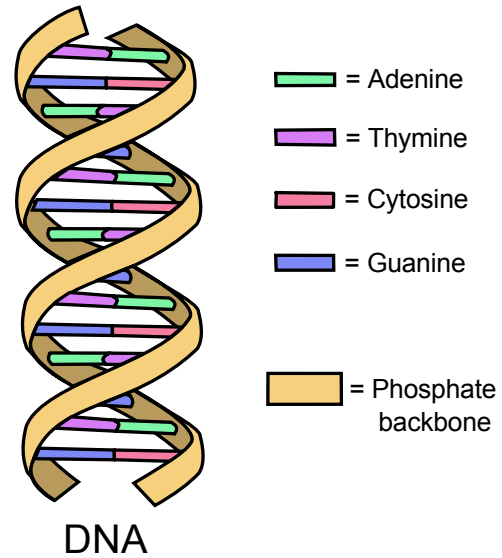
Austin, Sept, 26th, 2017

23rd DNA Computing & Molecular Programming Conference

OUTLINE

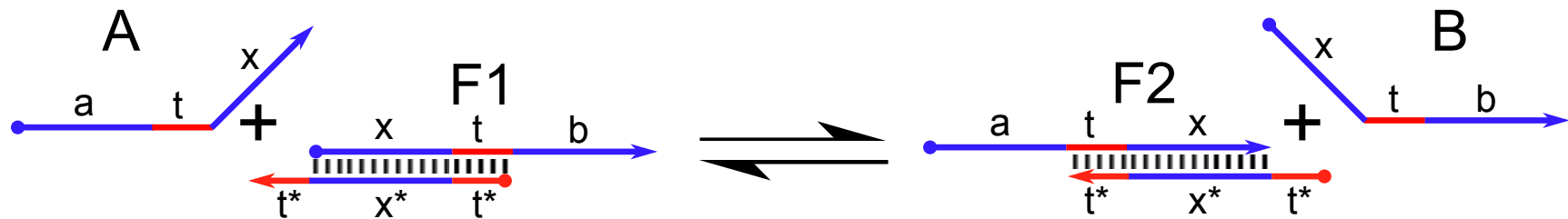
domain-level strand displacement (DSD) systems
formal chemical reaction networks (CRNs)
importance of a CRN-to-DSD compiler
translate formal CRNs into DSD systems
verify the correctness of a DSD system
examples of DSD system analysis

DNA STRAND DISPLACEMENT



DOMAIN-LEVEL STRAND DISPLACEMENT

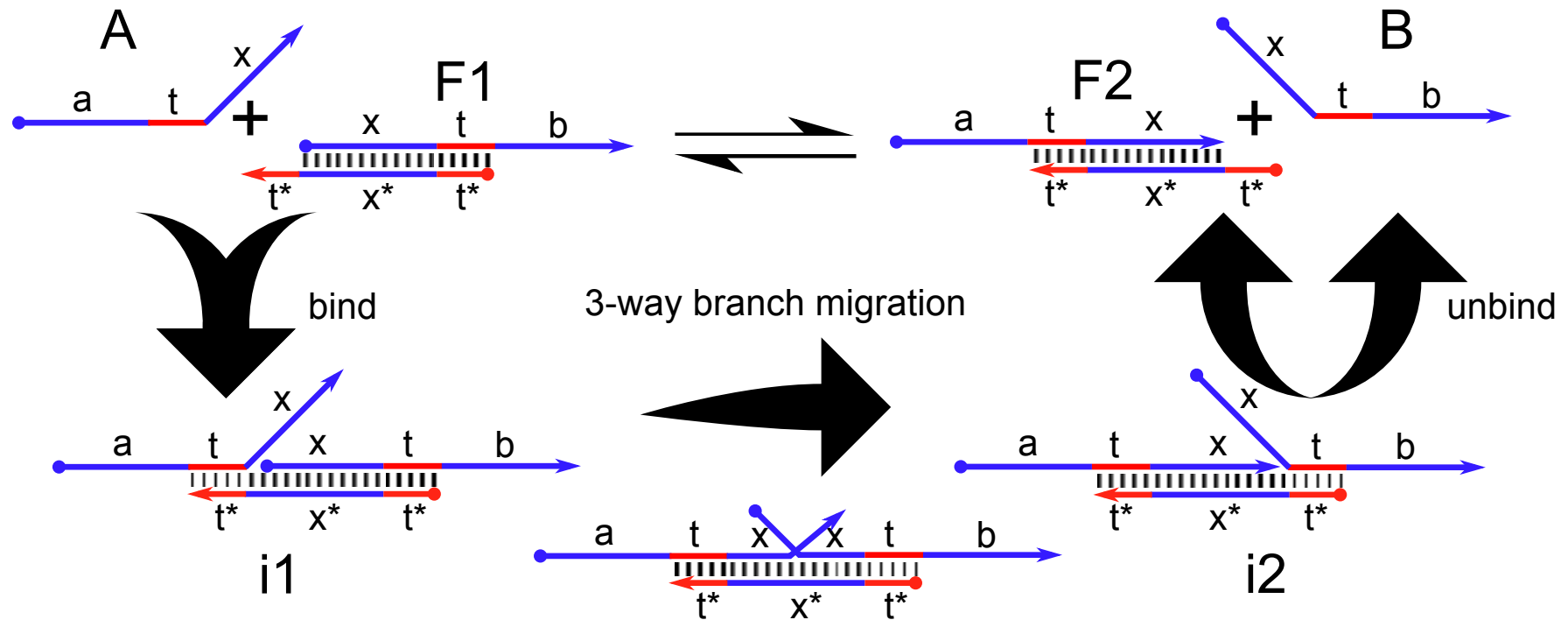
— long (branch-migration) domain: binds irreversibly
— short (toehold) domain: binds reversibly



DOMAIN-LEVEL STRAND DISPLACEMENT

— long (branch-migration) domain: binds irreversibly
— short (toehold) domain: binds reversibly

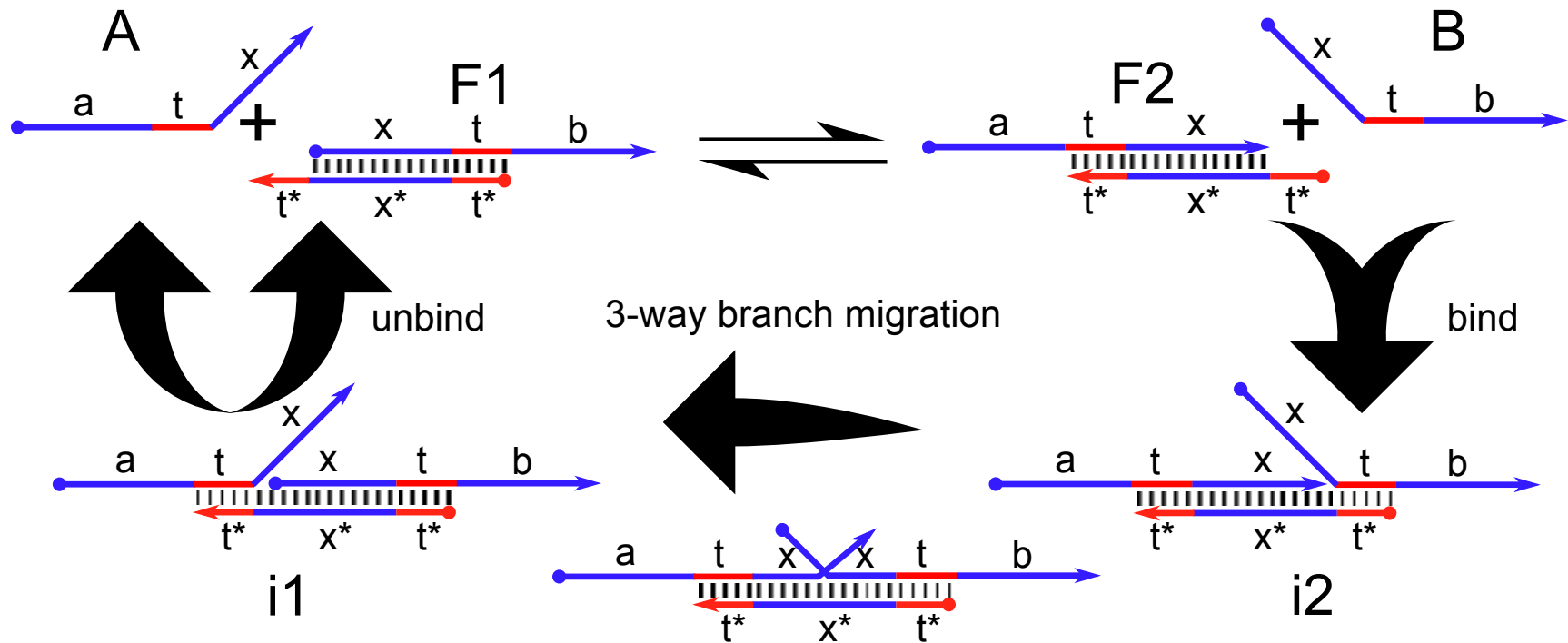
→ detailed network
⇌ condensed network



DOMAIN-LEVEL STRAND DISPLACEMENT

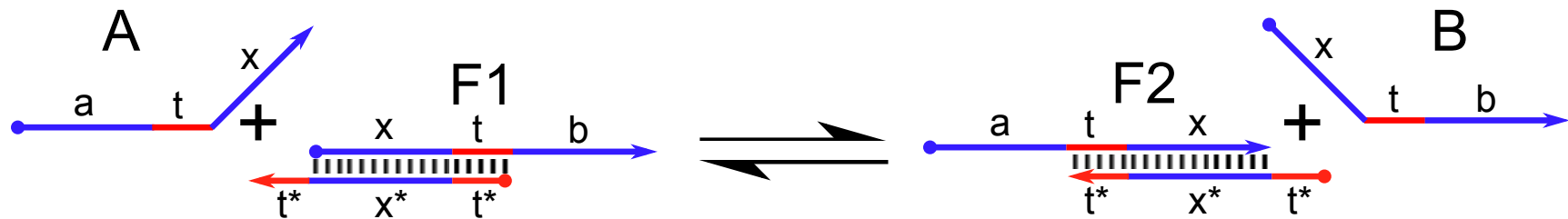
— long (branch-migration) domain: binds irreversibly
— short (toehold) domain: binds reversibly

→ detailed network
⇌ condensed network

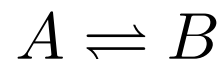


DOMAIN-LEVEL STRAND DISPLACEMENT

— long (branch-migration) domain: binds irreversibly
— short (toehold) domain: binds reversibly

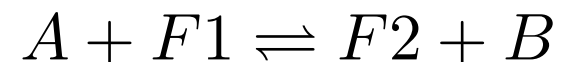


formal CRN



formal species: {A, B}

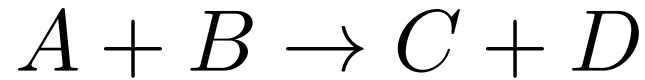
DSD system specification



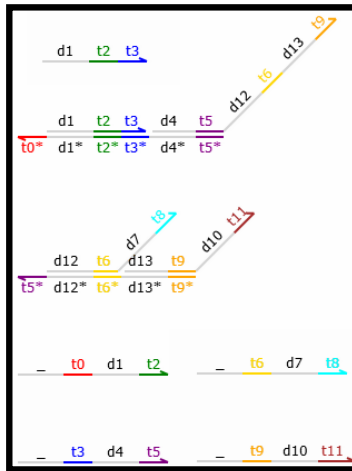
signal species (low concentration): {A, B}

fuel species (high concentration): {F1, F2}

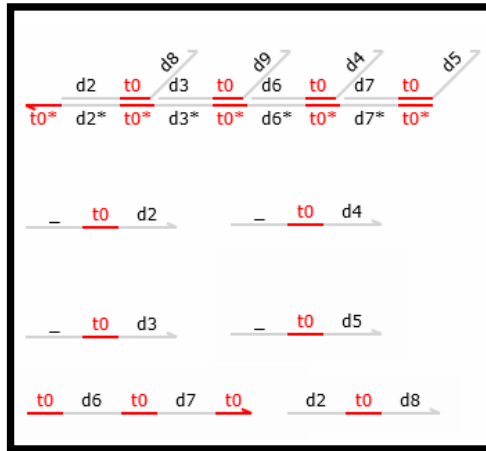
FROM CRN TO DSD SYSTEMS



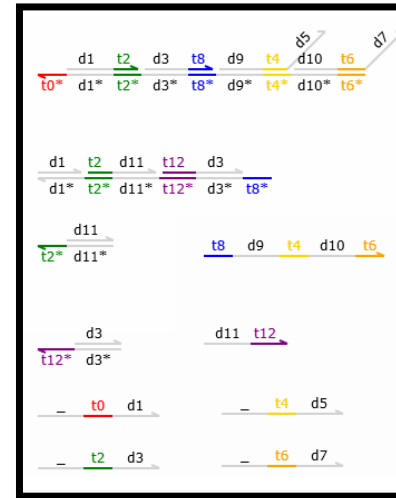
Soloveichik
et al. (2010)



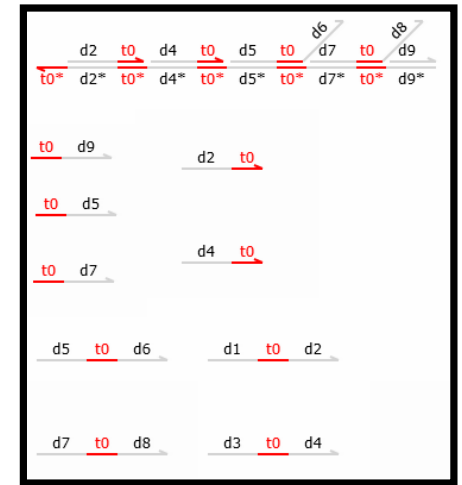
Lakin
et al. (2012)



Cardelli (2011)



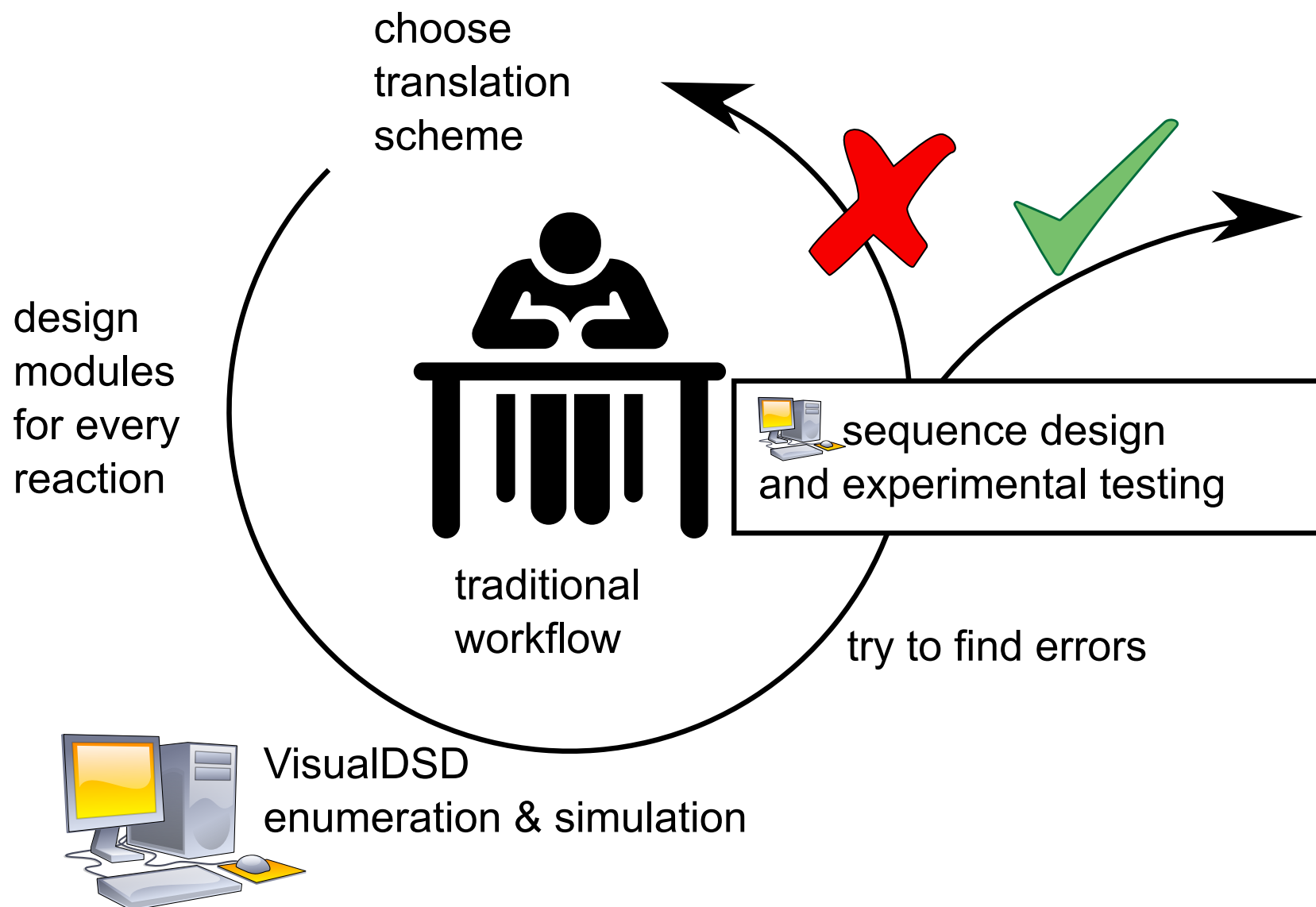
Qian et al. (2011)



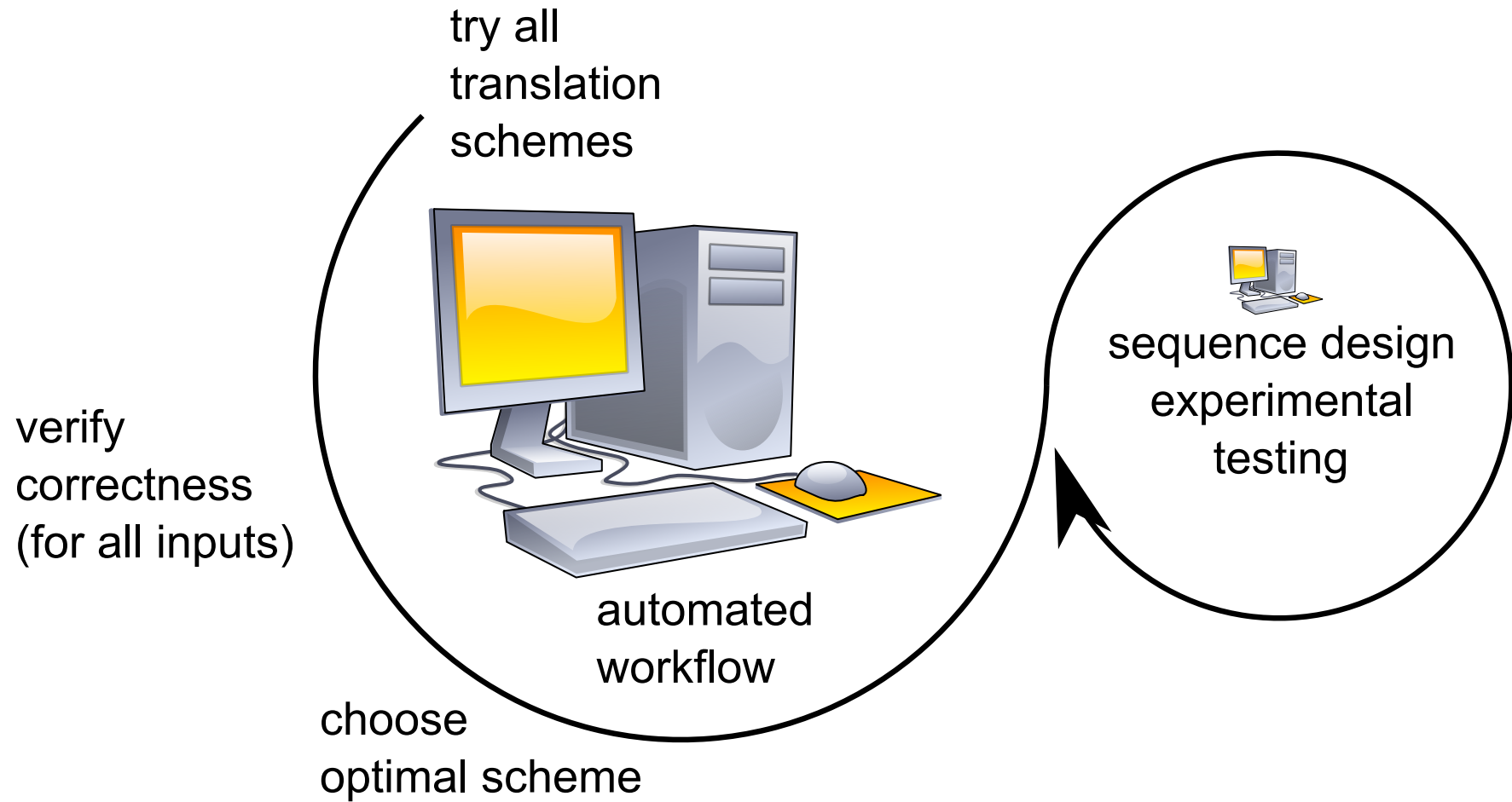
Chen et al. (2012), Cardelli (2013), Srinivas (2015), Lakin et al. (2016), ...

Images drawn using VisualDSD, Lakin et al. (2012)

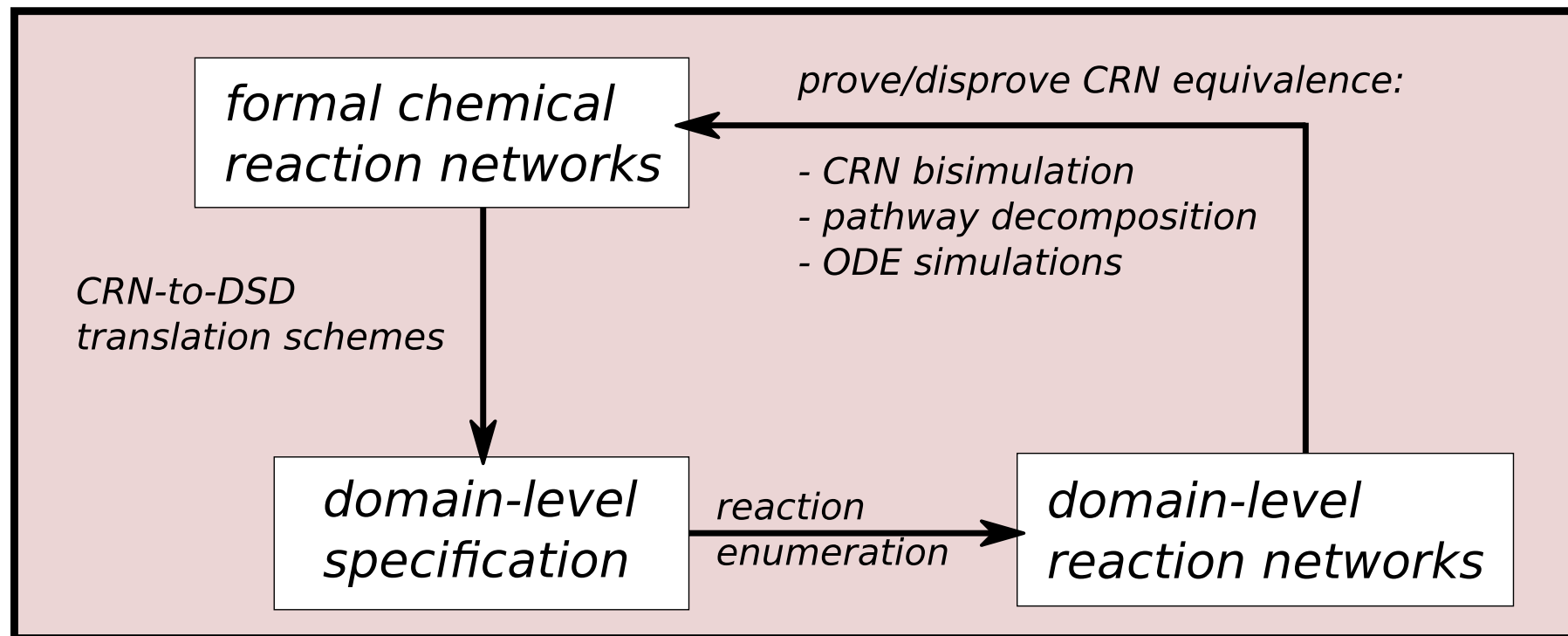
TRADITIONAL WORKFLOW



A CRN-TO-DSD COMPILER



THE NUSKELL COMPILER PROJECT



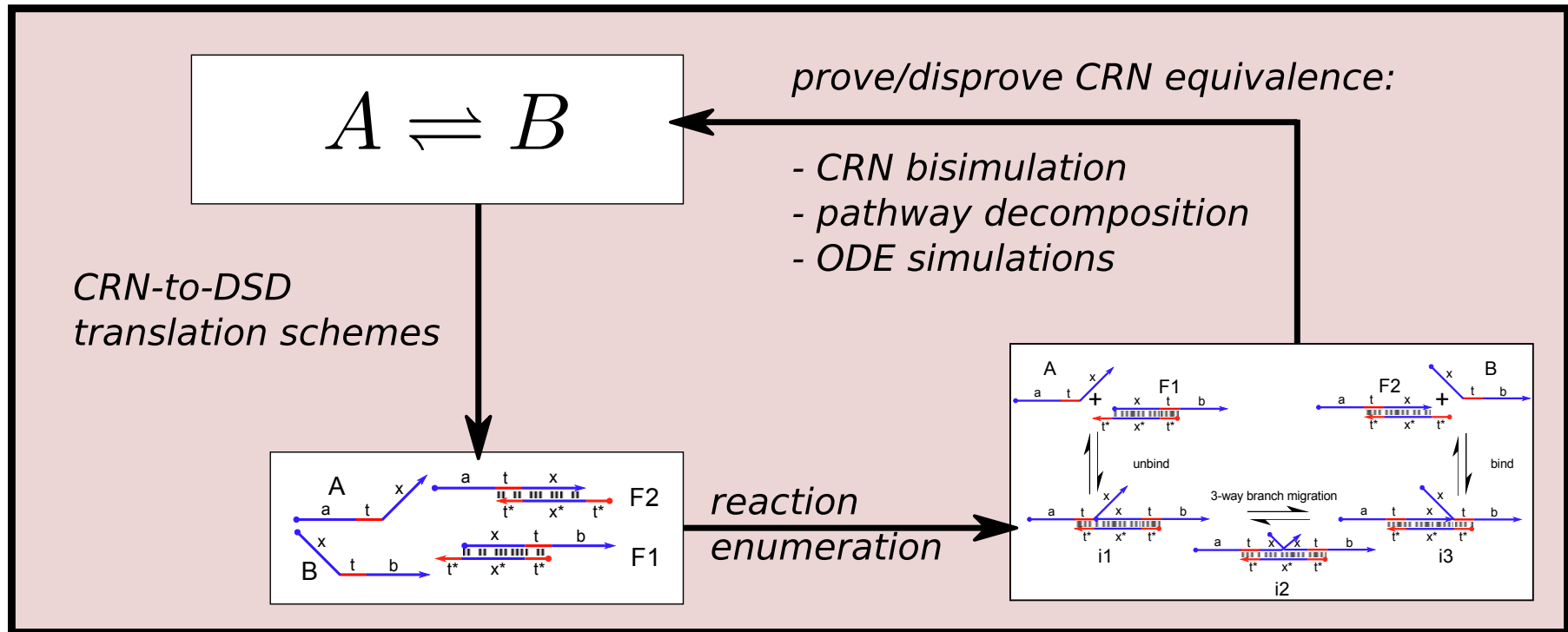
Badelt et al. (2017) - compiler framework and CRN-to-DSD translation

Grun et al. (2014) - reaction enumeration

Shin et al. (2014) - CRN pathway decomposition equivalence

Johnson et al. (2016) - CRN bisimulation equivalence

THE NUSKELL COMPILER PROJECT



Badelt et al. (2017) - compiler framework and CRN-to-DSD translation

Grun et al. (2014) - reaction enumeration

Shin et al. (2014) - CRN pathway decomposition equivalence

Johnson et al. (2016) - CRN bisimulation equivalence

TRANSLATION SCHEMES

```
# Translate formal reactions with two reactants and two products.  
# Lakin et. al (2012) "Abstractions for DNA circuit design." [Figure 5]
```

```
# Define a global short toehold domain  
global toehold = short();  
  
# Define domains and structure of signal species  
class formal(s) = "? t f" | ". . ."  
  where { t = toehold; f = long() };  
  
# Define fuel complexes for bimolecular reactions  
class binary_fuels(r, p) =  
  [ "a t i + b t k + ch t c + dh t d + t* dh* t* ch* t* b* t* a* t*" |  
    "( ( . + ( ( . + ( ( . + ( ( . + ) ) ) ) ) ) ) ) . ",  
    "a t i" | " . . . ", "t ch t dh t" | " . . . . " ]  
  where {  
    a = r[0].f;  
    b = r[1].f;  
    c = p[0].f; ch = long();  
    d = p[1].f; dh = long();  
    i = long(); k = long();  
    t = toehold };  
  
# Module *rxn* applies the fuel production to every bimolecular reaction  
module rxn(r) = sum(map(infty, fuels))  
  where fuels =  
    if len(r.reactants) != 2 or len(r.products) != 2 then  
      abort('Reaction type not implemented')  
    else  
      binary_fuels(r.reactants, r.products);  
  
# Module *main* applies *rxn* to the crn  
module main(crn) = sum(map(rxn, crn))  
  where crn = irrev_reactions(crn);
```

simplified variant of translation scheme: lakin2012.ts

TRANSLATION SCHEMES

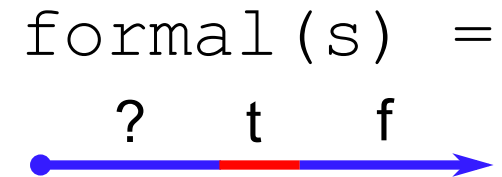
```
# Translate formal reactions with two reactants and two products.
# Lakin et. al (2012) "Abstractions for DNA circuit design." [Figure 5]
```

```
# Define a global short toehold domain
```

```
global toehold = short();
```

```
# Define domains and structure of signal species
```

```
class formal(s) = "? t f" | ". . ."
  where { t = toehold; f = long() };
```



```
# Define fuel complexes for bimolecular reactions
```

```
class binary_fuels(r, p) =
  [ "a t i + b t k + ch t c + dh t d + t* dh* t* ch* t* b* t* a* t*"
  | "( ( . + ( ( . + ( ( . + ( ( . + ) ) ) ) ) ) ) ) . ",
    "a t i" | " . . . ", "t ch t dh t" | " . . . . " ]
```

```
where {
```

```
  a = r[0].f;
```

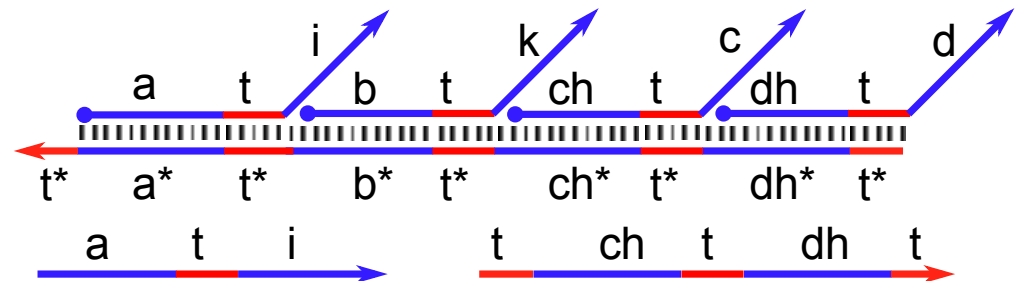
```
  b = r[1].f;
```

```
  c = p[0].f; ch = long();
```

```
  d = p[1].f; dh = long();
```

```
  i = long(); k = long();
```

```
  t = toehold };
```



simplified variant of translation scheme: lakin2012.ts

TRANSLATION SCHEMES

```
# Translate formal reactions with two reactants and two products.
# Lakin et. al (2012) "Abstractions for DNA circuit design." [Figure 5]
```

```
# Define a global short toehold domain
```

```
global toehold = short();
```

```
# Define domains and structure of signal species
```

```
class formal(s) = "? t f" | ". . ."
  where { t = toehold; f = long() };
```

```
# Define fuel complexes for bimolecular reactions
```

```
class binary_fuels(r, p) =
  [ "a t i + b t k + ch t c + dh t d + t* dh* t* ch* t* b* t* a* t*"
  | "( ( . + ( ( . + ( ( . + ( ( . + ) ) ) ) ) ) ) ) . ",
    "a t i" | " . . . ", "t ch t dh t" | " . . . . " ]
```

```
where {
```

```
  a = r[0].f;
```

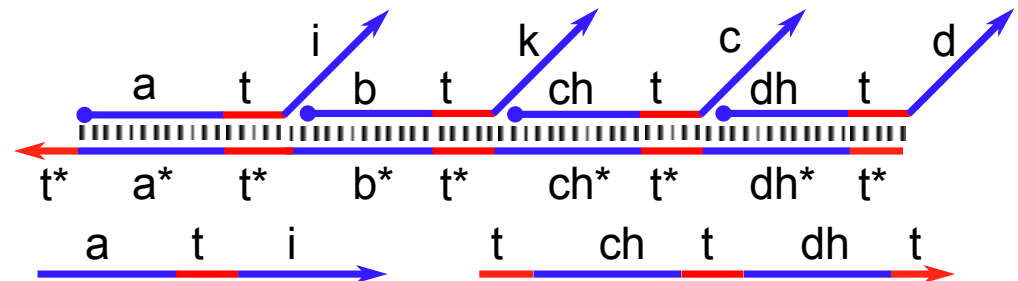
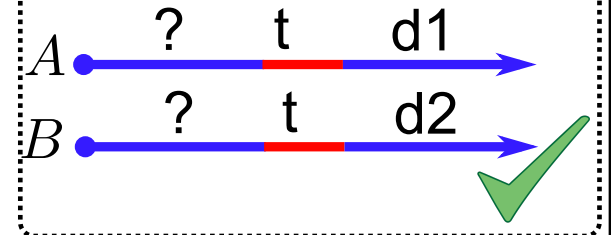
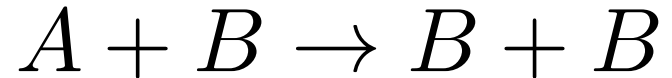
```
  b = r[1].f;
```

```
  c = p[0].f; ch = long();
```

```
  d = p[1].f; dh = long();
```

```
  i = long(); k = long();
```

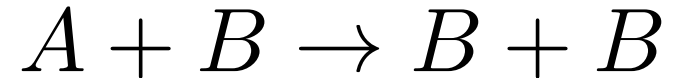
```
  t = toehold };
```



simplified variant of translation scheme: lakin2012.ts

TRANSLATION SCHEMES

```
# Translate formal reactions with two reactants and two products.
# Lakin et. al (2012) "Abstractions for DNA circuit design." [Figure 5]
```

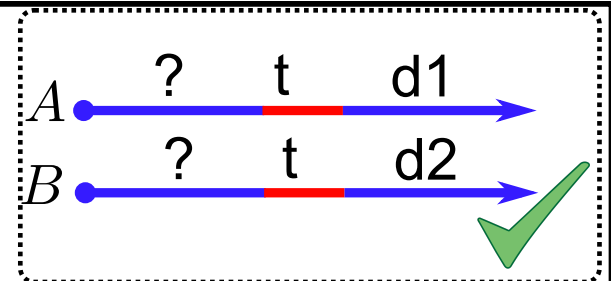


```
# Define a global short toehold domain
```

```
global toehold = short();
```

```
# Define domains and structure of signal species
```

```
class formal(s) = "? t f" | ". . ."
  where { t = toehold; f = long() };
```



```
# Define fuel complexes for bimolecular reactions
```

```
class binary_fuels(r, p) =
  [ "a t i + b t k + ch t c + dh t d + t* dh* t* ch* t* b* t* a* t*"
  | "( ( . + ( ( . + ( ( . + ( ( . + ) ) ) ) ) ) ) ) . ",
    "a t i" | " . . . ", "t ch t dh t" | " . . . . " ]
```

```
where {
```

```
  a = r[0].f;
```

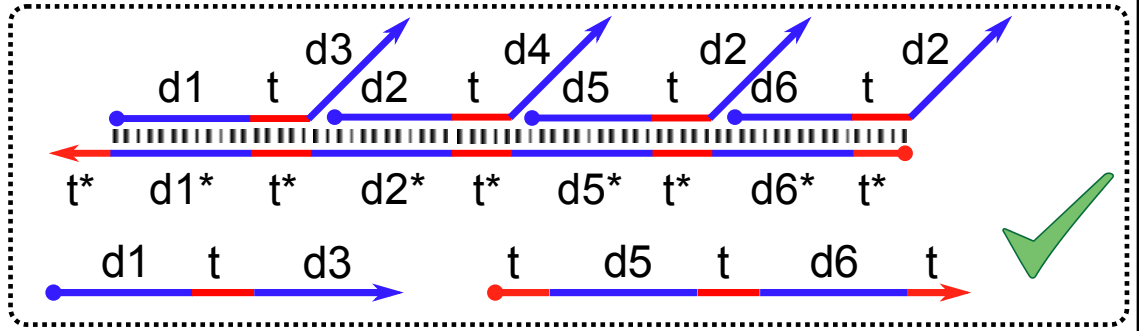
```
  b = r[1].f;
```

```
  c = p[0].f; ch = long();
```

```
  d = p[1].f; dh = long();
```

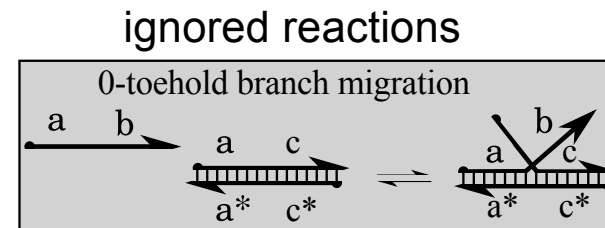
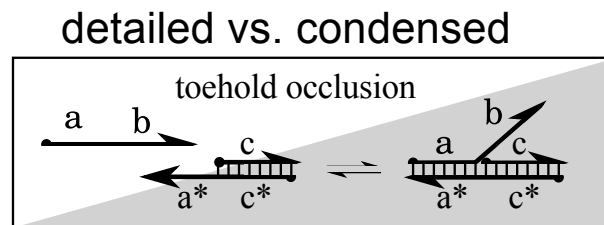
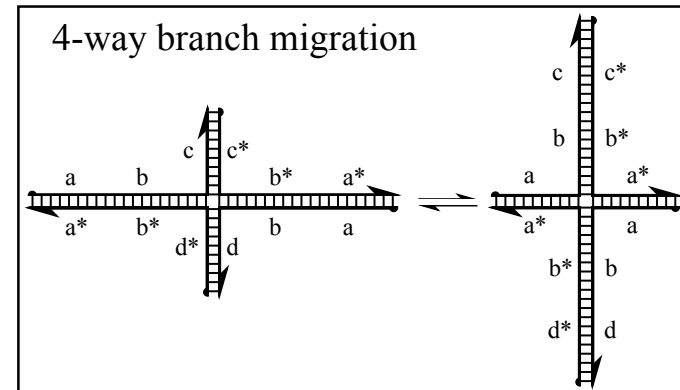
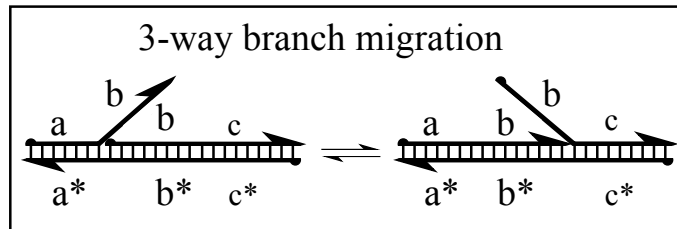
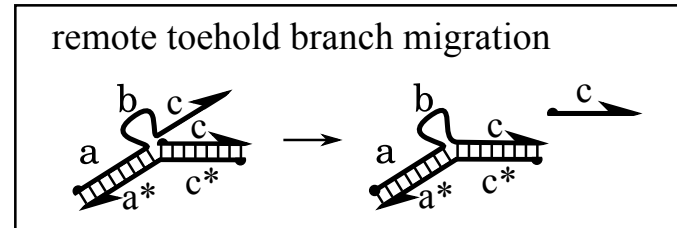
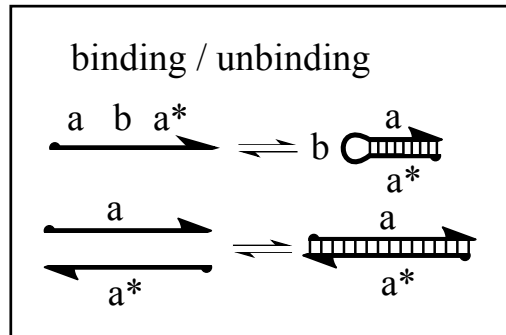
```
  i = long(); k = long();
```

```
  t = toehold };
```



simplified variant of translation scheme: lakin2012.ts

REACTION ENUMERATION



<http://www.github.com/DNA-and-Natural-Algorithms-Group/peppercornenumerator>

Grun et al. 2014

CRN EQUIVALENCE

formal input CRN

3 species

7 reactions

```
A      -> A + A
A + A  -> A
A + B  -> B + B
B      ->
A + C  ->
C      -> C + C
C + C  -> C
```

enumerated CRN

360 species

668 reactions

translation scheme: qian2011_3D_var1.ts

CRN EQUIVALENCE

formal input CRN

3 species

7 reactions

```
A      -> A + A
A + A -> A
A + B -> B + B
B      ->
A + C ->
C      -> C + C
C + C -> C
```

condensed CRN

42 species

32 reactions

```
f14 + C -> e1428 + f15
e853 + f12 -> C + f13
A + f4 -> f3 + e71
f2 + e25 -> A + f1
A + e25 -> f2 + e7
e996 + f3 -> A + f10
e1428 + f15 -> f14 + C
f3 + e71 -> A + f4
e465 + B -> e418 + f6
e614 + f9 -> e611 + e730
e996 + C -> e1040 + f12
e465 + f5 -> e514 + e368
e308 + f7 -> f8 + B
e418 + B -> e371 + f6
C + f13 -> e853 + f12
A + f1 -> f2 + e25
B + e71 -> e319 + f7
f2 + e7 -> A + e25
e1040 + f11 -> e1162 + e1163 + e1158
e319 + f7 -> B + e71
e308 + f9 -> e614 + e615 + e611
e371 + f6 -> e418 + B
e1040 + f12 -> e996 + C
f8 + B -> e308 + f7
e319 + f5 -> e372 + e371 + e368
f3 + e7 -> A + f0
e853 + f15 -> e1428 + C
e1428 + C -> e853 + f15
A + f10 -> e996 + f3
e1163 + f11 -> e1158 + e1246
e418 + f6 -> e465 + B
A + f0 -> f3 + e7
```

enumerated CRN

360 species

668 reactions

translation scheme: qian2011_3D_var1.ts

CRN EQUIVALENCE

formal input CRN

3 species
7 reactions

```
A      -> A + A
A + A  -> A
A + B  -> B + B
B      ->
A + C  ->
C      -> C + C
C + C  -> C
```

verification CRN

26 species (no fuel species)
32 reactions

```
C -> e1428
e853 -> C
A -> e71
e25 -> A
A + e25 -> e7
e996 -> A
e1428 -> C
e71 -> A
e465 + B -> e418
e614 -> e611 + e730
e996 + C -> e1040
e465 -> e514 + e368
e308 -> B
e418 + B -> e371
C -> e853
A -> e25
B + e71 -> e319
e7 -> A + e25
e1040 -> e1162 + e1163 + e1158
e319 -> B + e71
e308 -> e614 + e615 + e611
e371 -> e418 + B
e1040 -> e996 + C
B -> e308
e319 -> e372 + e371 + e368
e7 -> A
e853 -> e1428 + C
e1428 + C -> e853
A -> e996
e1163 -> e1158 + e1246
e418 -> e465 + B
A -> e7
```

condensed CRN

42 species
32 reactions

```
f14 + C -> e1428 + f15
e853 + f12 -> C + f13
A + f4 -> f3 + e71
f2 + e25 -> A + f1
A + e25 -> f2 + e7
e996 + f3 -> A + f10
e1428 + f15 -> f14 + C
f3 + e71 -> A + f4
e465 + B -> e418 + f6
e614 + f9 -> e611 + e730
e996 + C -> e1040 + f12
e465 + f5 -> e514 + e368
e308 + f7 -> f8 + B
e418 + B -> e371 + f6
C + f13 -> e853 + f12
A + f1 -> f2 + e25
B + e71 -> e319 + f7
f2 + e7 -> A + e25
e1040 + f11 -> e1162 + e1163 + e1158
e319 + f7 -> B + e71
e308 + f9 -> e614 + e615 + e611
e371 + f6 -> e418 + B
e1040 + f12 -> e996 + C
f8 + B -> e308 + f7
e319 + f5 -> e372 + e371 + e368
f3 + e7 -> A + f0
e853 + f15 -> e1428 + C
e1428 + C -> e853 + f15
A + f10 -> e996 + f3
e1163 + f11 -> e1158 + e1246
e418 + f6 -> e465 + B
A + f0 -> f3 + e7
```

enumerated CRN

360 species
668 reactions

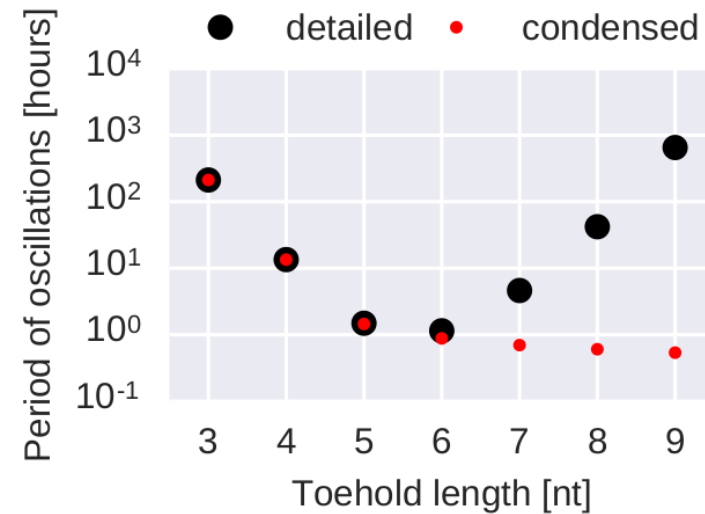
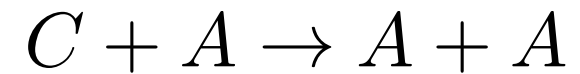
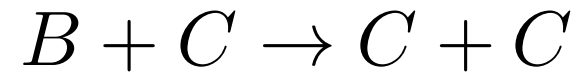
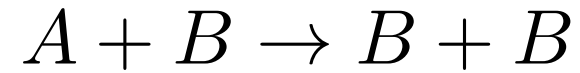
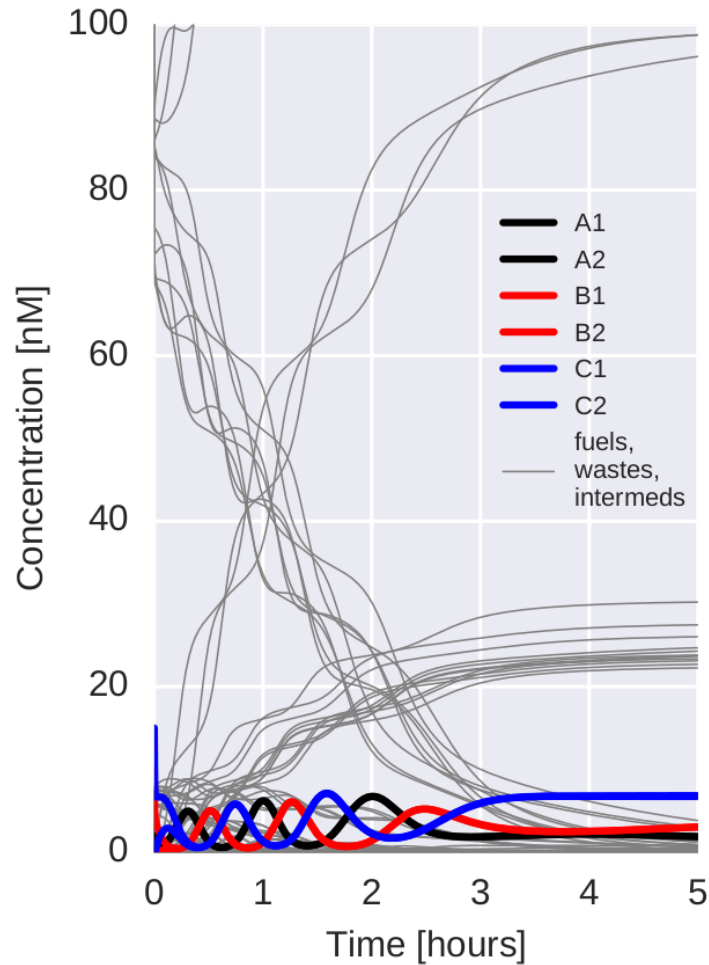
translation scheme: qian2011_3D_var1.ts

CRN EQUIVALENCE

	formal input CRN	interpreted CRN	verification CRN	condensed CRN	enumerated CRN
	3 species 7 reactions	3 species 7 non-trivial reactions	26 species (no fuel species) 32 reactions	42 species 32 reactions	360 species 668 reactions
Interpretation (CRN-bisimulation):	A $\rightarrow$ A + A	C $\rightarrow$ C	C $\rightarrow$ e1428	f14 + C $\rightarrow$ e1428 + f15	
	A + A $\rightarrow$ A	C + C $\rightarrow$ C	e853 $\rightarrow$ C	e853 + f12 $\rightarrow$ C + f13	
	A + B $\rightarrow$ B + B	A $\rightarrow$ A	A $\rightarrow$ e71	A + f4 $\rightarrow$ f3 + e71	
	B $\rightarrow$	A $\rightarrow$ A	e25 $\rightarrow$ A	f2 + e25 $\rightarrow$ A + f1	
	A + C $\rightarrow$	A + A $\rightarrow$ A + A	A + e25 $\rightarrow$ e7	A + e25 $\rightarrow$ f2 + e7	
	C $\rightarrow$ C + C	A $\rightarrow$ A	e996 $\rightarrow$ A	e996 + f3 $\rightarrow$ A + f10	
	C + C $\rightarrow$ C	C $\rightarrow$ C	e1428 $\rightarrow$ C	e1428 + f15 $\rightarrow$ f14 + C	
		A $\rightarrow$ A	e71 $\rightarrow$ A	f3 + e71 $\rightarrow$ A + f4	
	A $\Rightarrow$ A	B $\rightarrow$ B	e465 + B $\rightarrow$ e418	e465 + B $\rightarrow$ e418 + f6	
	B $\Rightarrow$ B	$\rightarrow$	e614 $\rightarrow$ e611 + e730	e614 + f9 $\rightarrow$ e611 + e730	
	C $\Rightarrow$ C	A + C $\rightarrow$ A + C	e996 + C $\rightarrow$ e1040	e996 + C $\rightarrow$ e1040 + f12	
	e1040 $\Rightarrow$ A, C	$\rightarrow$	e465 $\rightarrow$ e514 + e368	e465 + f5 $\rightarrow$ e514 + e368	
	e1158 $\Rightarrow$	B $\rightarrow$ B	e308 $\rightarrow$ B	e308 + f7 $\rightarrow$ f8 + B	
	e1162 $\Rightarrow$	B + B $\rightarrow$ B + B	e418 + B $\rightarrow$ e371	e418 + B $\rightarrow$ e371 + f6	
	e1163 $\Rightarrow$	C $\rightarrow$ C + C	C $\rightarrow$ e853	C + f13 $\rightarrow$ e853 + f12	
	e1246 $\Rightarrow$	A $\rightarrow$ A	A $\rightarrow$ e25	A + f1 $\rightarrow$ f2 + e25	
	e1428 $\Rightarrow$ C	B + A $\rightarrow$ A + B	B + e71 $\rightarrow$ e319	B + e71 $\rightarrow$ e319 + f7	
	e25 $\Rightarrow$ A	A + A $\rightarrow$ A + A	e7 $\rightarrow$ A + e25	f2 + e7 $\rightarrow$ A + e25	
	e308 $\Rightarrow$ B	A + C $\rightarrow$	e1040 $\rightarrow$ e1162 + e1163 + e1158	e1040 + f11 $\rightarrow$ e1162 + e1163 + e1158	
	e319 $\Rightarrow$ A, B	A + B $\rightarrow$ B + A	e319 $\rightarrow$ B + e71	e319 + f7 $\rightarrow$ B + e71	
	e368 $\Rightarrow$	B $\rightarrow$	e308 $\rightarrow$ e614 + e615 + e611	e308 + f9 $\rightarrow$ e614 + e615 + e611	
	e371 $\Rightarrow$ B, B	B + B $\rightarrow$ B + B	e371 $\rightarrow$ e418 + B	e371 + f6 $\rightarrow$ e418 + B	
	e372 $\Rightarrow$	A + C $\rightarrow$ A + C	e1040 $\rightarrow$ e996 + C	e1040 + f12 $\rightarrow$ e996 + C	
	e418 $\Rightarrow$ B	B $\rightarrow$ B	B $\rightarrow$ e308	f8 + B $\rightarrow$ e308 + f7	
	e465 $\Rightarrow$	A + B $\rightarrow$ B + B	e319 $\rightarrow$ e372 + e371 + e368	e319 + f5 $\rightarrow$ e372 + e371 + e368	
	e514 $\Rightarrow$	A + A $\rightarrow$ A	e7 $\rightarrow$ A	f3 + e7 $\rightarrow$ A + f0	
	e611 $\Rightarrow$	C + C $\rightarrow$ C + C	e853 $\rightarrow$ e1428 + C	e853 + f15 $\rightarrow$ e1428 + C	
	e614 $\Rightarrow$	C + C $\rightarrow$ C + C	e1428 + C $\rightarrow$ e853	e1428 + C $\rightarrow$ e853 + f15	
	e615 $\Rightarrow$	A $\rightarrow$ A	A $\rightarrow$ e996	A + f10 $\rightarrow$ e996 + f3	
	e7 $\Rightarrow$ A, A	$\rightarrow$	e1163 $\rightarrow$ e1158 + e1246	e1163 + f11 $\rightarrow$ e1158 + e1246	
	e71 $\Rightarrow$ A	B $\rightarrow$ B	e418 $\rightarrow$ e465 + B	e418 + f6 $\rightarrow$ e465 + B	
	e730 $\Rightarrow$	A $\rightarrow$ A + A	A $\rightarrow$ e7	A + f0 $\rightarrow$ f3 + e7	
	e853 $\Rightarrow$ C, C				
	e996 $\Rightarrow$ A				

Johnson et al. (2016) - CRN bisimulation equivalence
translation scheme: qian2011_3D_var1.ts

DETAILED VS. CONDENSED ENUMERATION



translation scheme: srinivas2015.ts

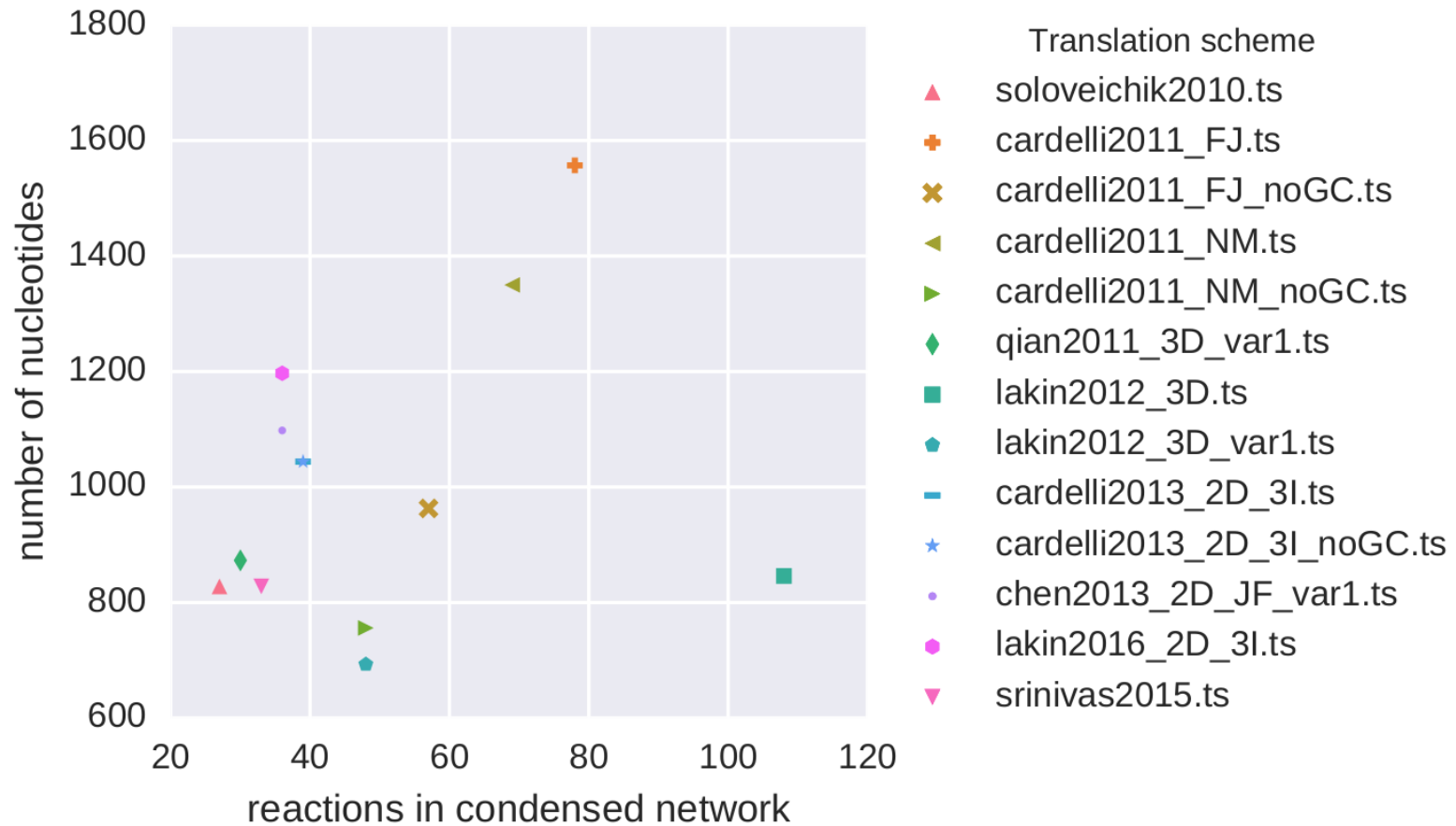
OSCILLATOR ANALYSIS 1/2

Translation scheme	pathway decompositon	CRN bisimulation
soloveichik2010.ts	True	True
cardelli2011_FJ.ts	False	False
cardelli2011_FJ_noGC.ts	False	True
cardelli2011_NM.ts	timeout	False
cardelli2011_NM_noGC.ts	False	True
qian2011_3D_var1.ts	True	True
lakin2012_3D.ts	False	False
lakin2012_3D_var1.ts	False	True
cardelli2013_2D_3I.ts	False	True
cardelli2013_2D_3I_noGC.ts	timeout	True
chen2013_2D_JF_var1.ts	timeout	True
lakin2016_2D_3I.ts	timeout	False
srinivas2015.ts	True	True

CRN bisimulation equivalence: Johnson et al. (2016)

CRN pathway decomposition equivalence: Shin et al. (2014)

OSCILLATOR ANALYSIS 2/2



FROM A DIGITAL CIRCUIT TO DSD

$y_2 y_1 = \lfloor \sqrt{x_4 x_3 x_2 x_1} \rfloor$

Qian et al. (2011)

fanout:

$x_{on} \rightarrow i_{on} + j_{on} + k_{on}$
 $x_{off} \rightarrow i_{off} + j_{off} + k_{off}$

AND gate:

$i_{off} + j_{off} \rightarrow y_{off}$
 $i_{off} + j_{on} \rightarrow y_{off}$
 $i_{on} + j_{off} \rightarrow y_{off}$
 $i_{on} + j_{on} \rightarrow y_{on}$

The CRN computes the floor of the squareroot of a four-bit binary number.

```

# Fanout X3 -> F5 + F6 + F7
X3_OFF -> F5_OFF + F6_OFF + F7_OFF
X3_ON -> F5_ON + F6_ON + F7_ON
# Fanout X4 -> F8 + F9 + F10
X4_OFF -> F8_OFF + F9_OFF + F10_OFF
X4_ON -> F8_ON + F9_ON + F10_ON
# G11 = NOT(X1 OR X2) + Fanout G11 -> F14 + F15
X1_OFF + X2_OFF -> F14_OFF + F15_ON
X1_OFF + X2_ON -> F14_OFF + F15_OFF
X1_ON + X2_OFF -> F14_OFF + F15_OFF
X1_ON + X2_ON -> F14_OFF + F15_OFF
# G12 = F6 AND (NOT F9)
F6_OFF + F9_OFF -> G12_OFF
F6_OFF + F9_ON -> G12_OFF
F6_ON + F9_OFF -> G12_ON
F6_ON + F9_ON -> G12_OFF
# Y2 = F7 OR F10
F7_OFF + F10_OFF -> Y2_OFF
F7_OFF + F10_ON -> Y2_ON
F7_ON + F10_OFF -> Y2_ON
F7_ON + F10_ON -> Y2_ON

# G16b = F5 AND F8
F5_OFF + F8_OFF -> G16b_OFF
F5_OFF + F8_ON -> G16b_OFF
F5_ON + F8_OFF -> G16b_OFF
F5_ON + F8_ON -> G16b_ON
# G16 = NOT(F14 AND G16b)
F14_OFF + G16b_OFF -> G16_ON
F14_OFF + G16b_ON -> G16_ON
F14_ON + G16b_OFF -> G16_ON
F14_ON + G16b_ON -> G16_OFF
# G17 = F15 OR G12
F15_OFF + G12_OFF -> G17_OFF
F15_OFF + G12_ON -> G17_ON
F15_ON + G12_OFF -> G17_ON
F15_ON + G12_ON -> G17_ON
# Y1 = NOT(G16 AND G17)
G16_OFF + G17_OFF -> Y1_ON
G16_OFF + G17_ON -> Y1_ON
G16_ON + G17_OFF -> Y1_ON
G16_ON + G17_ON -> Y1_OFF
                    
```

Nuskell

Input: 32 formal reactions.

soloveichik2010.ts: 52 signal species, 92 fuel species, 172 intermediate species, 180 reactions.

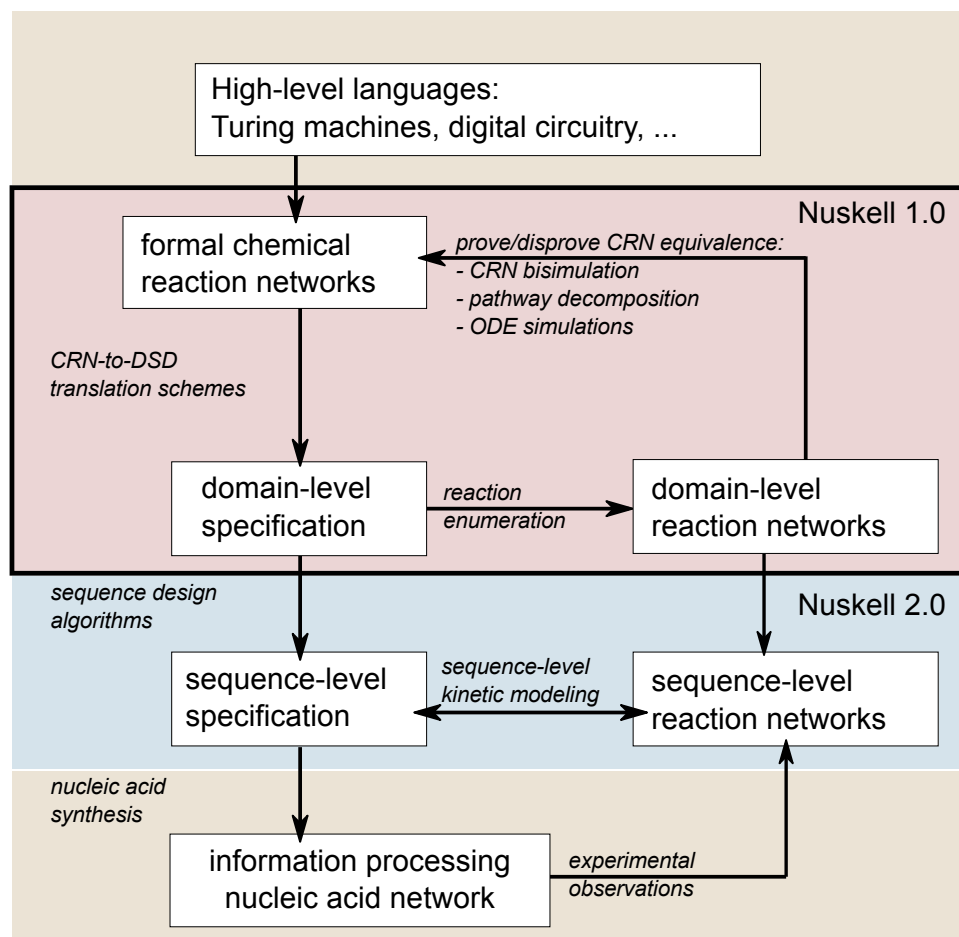
verifies as **correct** according to the **pathway decomposition** and **CRN bisimulation** equivalence

SUMMARY

The CRN-to-DSD compiler Nuskell

- translates formal CRNs to signal and fuel species
 - with choice of translation scheme (design algorithm)
- enumerates reactions given signal and fuel species
 - detailed vs. condensed semantics
 - approximate DNA reaction rates
- verifies equivalence of formal and enumerated CRN
 - CRN bisimulation equivalence
 - CRN pathway decomposition equivalence
- can be coupled with automated sequence-level design

THE NUSKELL COMPILER PROJECT



THANKS TO



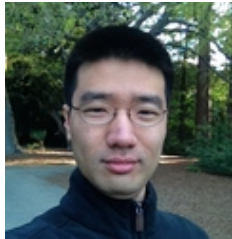
Erik Winfree



Robert F. Johnson



Chris Thachuk



Seung Woo Shin



you



Qing Dong

<http://www.github.com/DNA-and-Natural-Algorithms-Group/nuskell>

This research was funded in parts by:

The Caltech Biology and Biological Engineering Division Fellowship.

The U.S. National Science Foundation NSF Grant CCF-1213127 and NSF Grant CCF-1317694.

The Gordon and Betty Moore Foundation's Programmable Molecular Technology Initiative (PMTI).