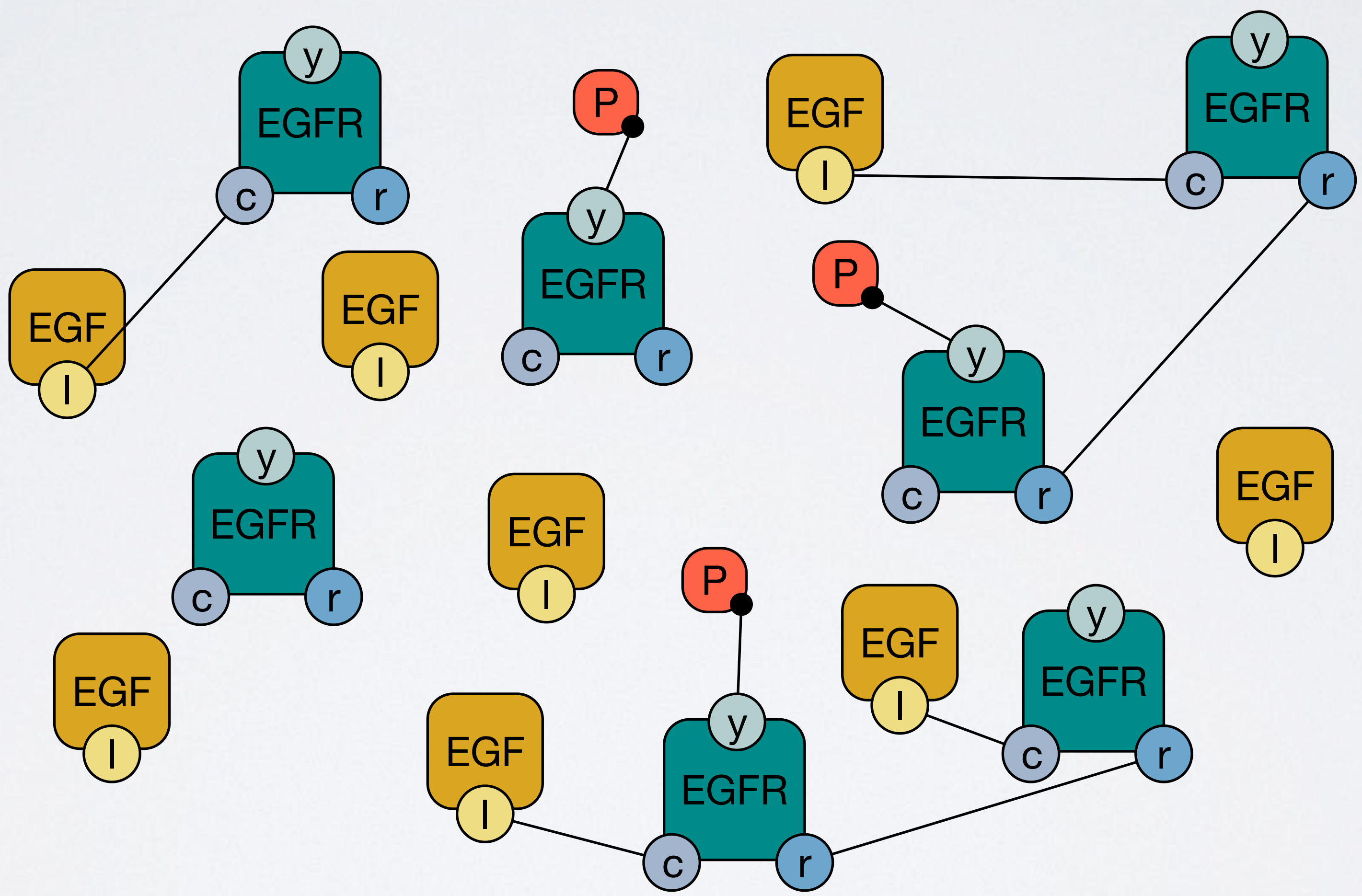
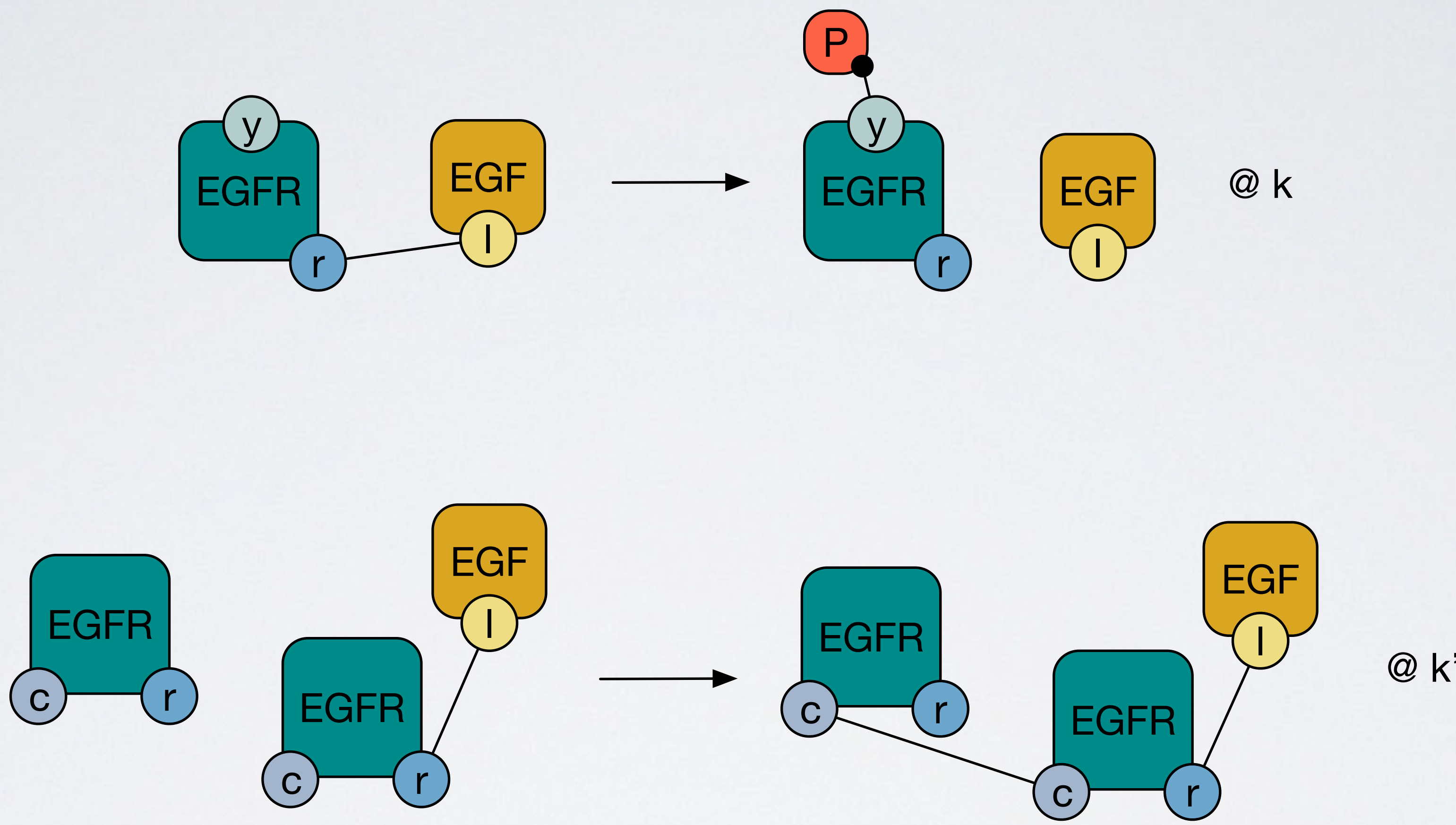
WHEN RULE-BASED MODELS NEED TO COUNT

Pierre Boutillier - Ioana Cristescu - Walter Fontana
http://kappalanguage.org

KAPPA MIXTURE

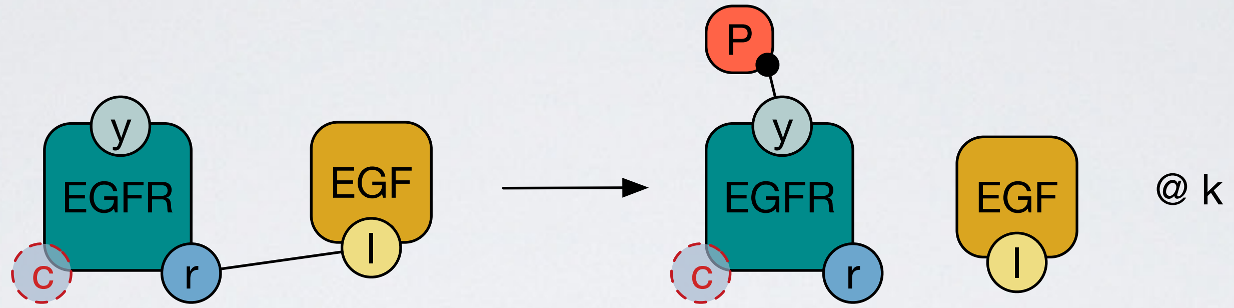


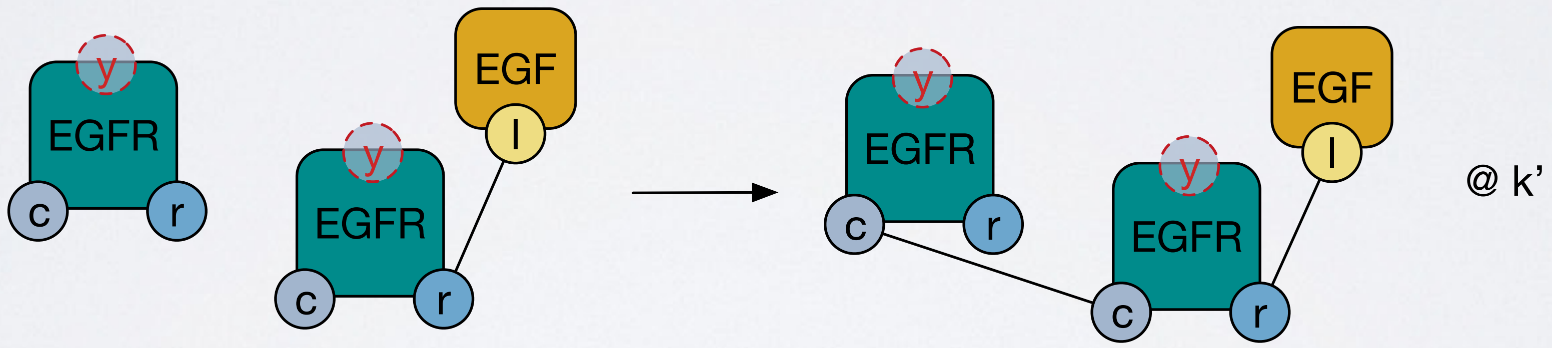
A KAPPA MODEL



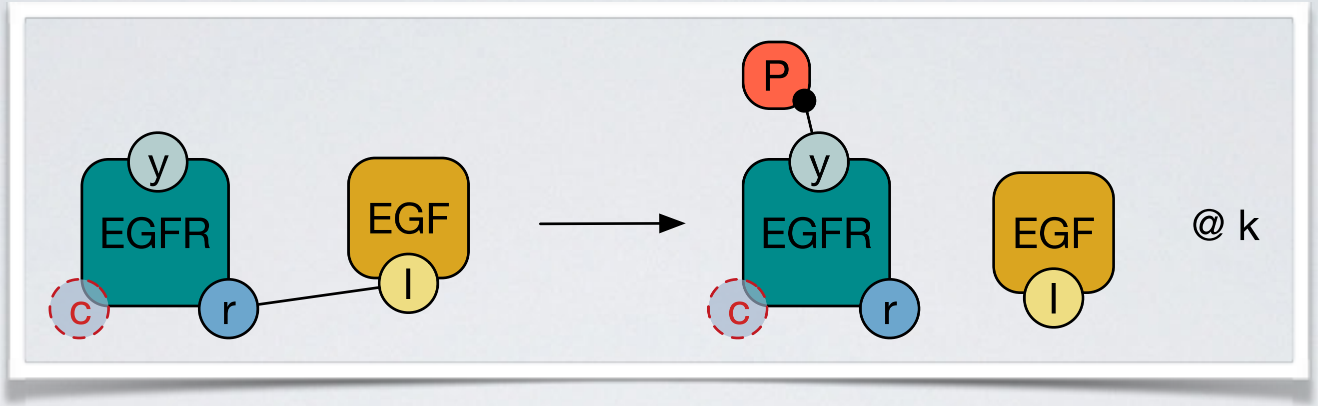
. . .

PATTERNS VS SPECIES

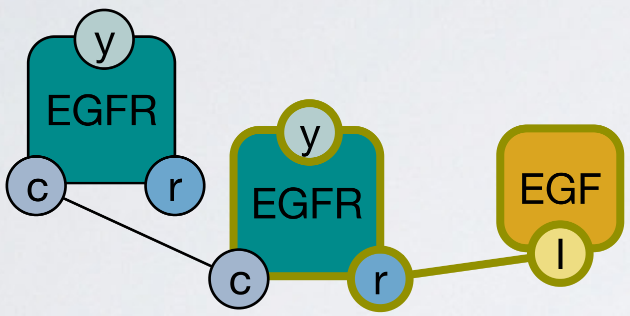




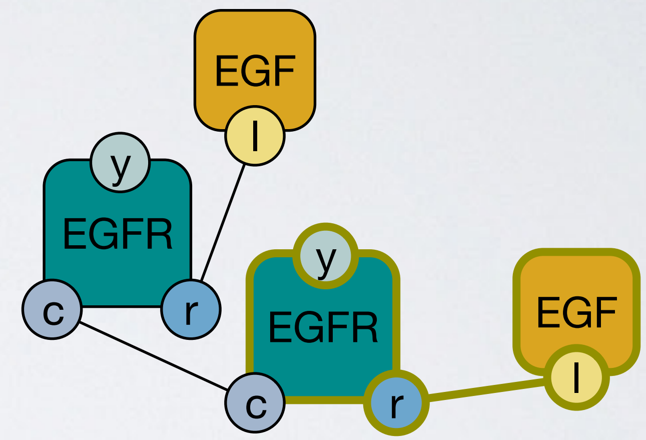
. . .


y
EGFR
c
r
EGF
l
P
y
EGFR
c
r
EGF
l
@ k

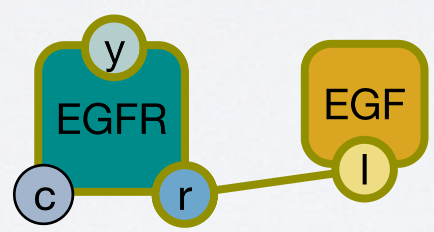
B


y
EGFR
c
r
y
EGFR
c
r
EGF
l

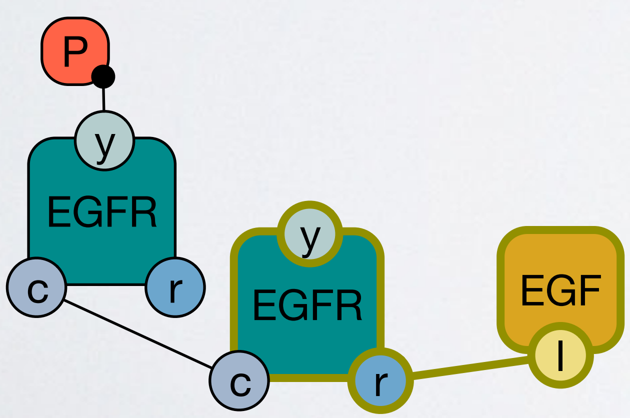
C


EGF
l
y
EGFR
c
r
y
EGFR
c
r
EGF
l

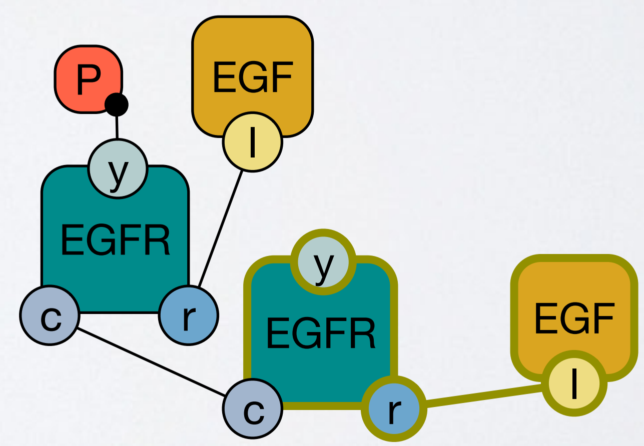
A

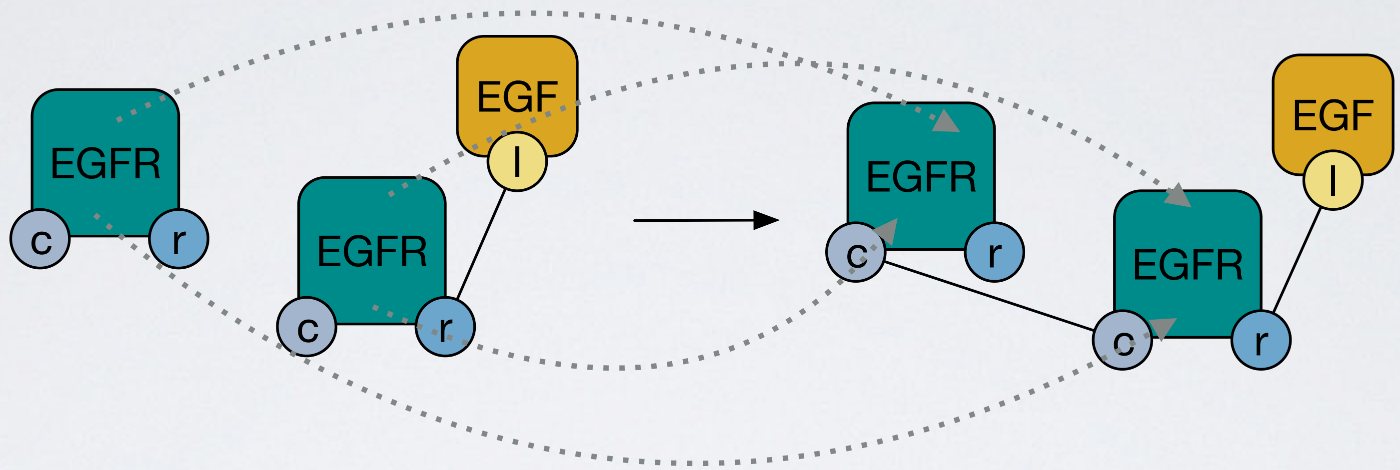

y
EGFR
c
r
EGF
l

D

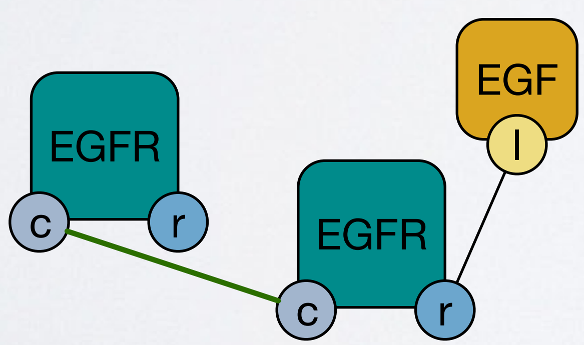

P
y
EGFR
c
r
y
EGFR
c
r
EGF
l

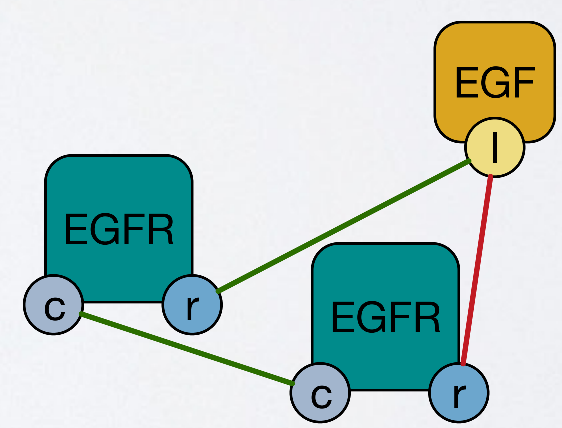
E


P
EGF
l
y
EGFR
c
r
y
EGFR
c
r
EGF
l


EGFR
c
r
EGF
l
EGFR
c
r
EGFR
c
r
EGF
l
EGFR
c
r

@ k’

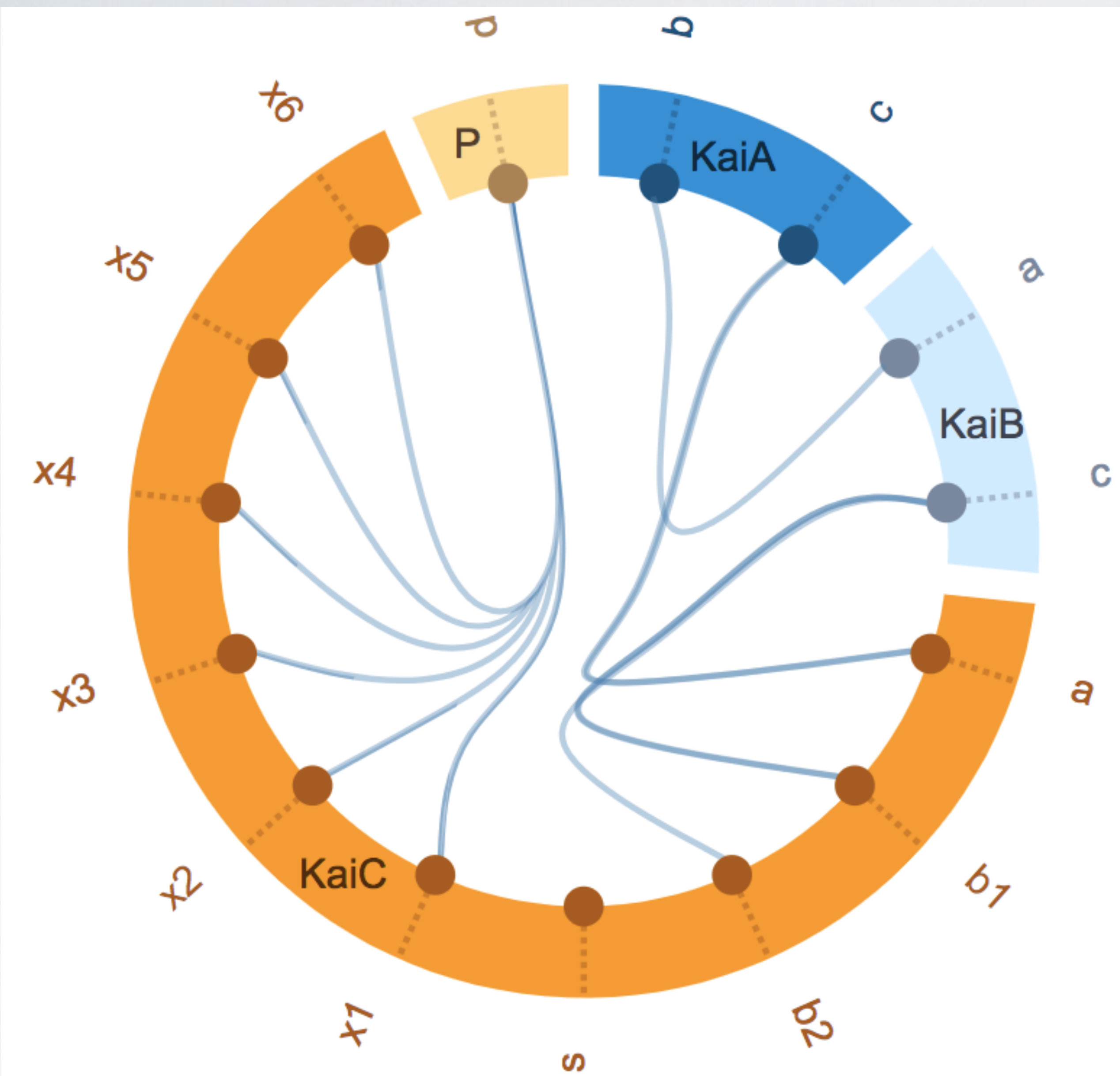

EGFR
c
r
EGF
l
EGFR
c
r


EGF
l
EGFR
c
r
EGFR
c
r

DISTINCT SITES

```
begin parameters
kp0 3
ku1 14
kp1 15
ku2 98
kp2 75
ku3 686
Stot 100
end parameters

begin molecule types
A(s~u~p,s~u~p,s~u~p,p~0~1~2~3~PLUS~MINUS)
end molecule types

begin seed species
A(s~u,s~u,s~u,p~0) Stot
end seed species

begin reaction rules
A(s~u,p~0) -> A(s~p,p~PLUS) kp0
A(s~u,p~1) -> A(s~p,p~PLUS) kp1
A(s~u,p~2) -> A(s~p,p~PLUS) kp2

A(s~p,p~1) -> A(s~u,p~MINUS) ku1
A(s~p,p~2) -> A(s~u,p~MINUS) ku2
A(s~p,p~3) -> A(s~u,p~MINUS) ku3
end reaction rules

```

CASE STUDY



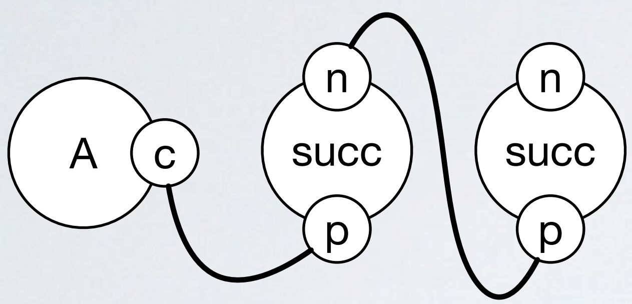

```
A(s1~p,s2~u,s3~p,s4~u) -> A(s1~p,s2~u,s3~p,s4~p) @'kp2'
A(s1~p,s2~u,s3~p,s4~u) -> A(s1~p,s2~u,s3~u,s4~u) @'ku2'
A(s1~p,s2~u,s3~p,s4~u) -> A(s1~p,s2~p,s3~p,s4~u) @'kp2'
A(s1~p,s2~u,s3~p,s4~u) -> A(s1~u,s2~u,s3~p,s4~u) @'ku2'
A(s1~p,s2~u,s3~p,s4~p) -> A(s1~p,s2~u,s3~p,s4~u) @'ku3'
A(s1~p,s2~u,s3~p,s4~p) -> A(s1~p,s2~u,s3~u,s4~p) @'ku3'
A(s1~p,s2~u,s3~p,s4~p) -> A(s1~p,s2~p,s3~p,s4~p) @'kp3'
A(s1~p,s2~u,s3~p,s4~p) -> A(s1~u,s2~u,s3~p,s4~p) @'ku3'
A(s1~p,s2~u,s3~u,s4~u) -> A(s1~p,s2~u,s3~u,s4~p) @'kp1'
A(s1~p,s2~u,s3~u,s4~u) -> A(s1~p,s2~u,s3~p,s4~u) @'kp1'
A(s1~p,s2~u,s3~u,s4~u) -> A(s1~p,s2~p,s3~u,s4~u) @'kp1'
A(s1~p,s2~u,s3~u,s4~u) -> A(s1~u,s2~u,s3~u,s4~u) @'ku1'
A(s1~p,s2~u,s3~u,s4~p) -> A(s1~p,s2~u,s3~u,s4~u) @'ku2'
A(s1~p,s2~u,s3~u,s4~p) -> A(s1~p,s2~u,s3~p,s4~p) @'kp2'
A(s1~p,s2~u,s3~u,s4~p) -> A(s1~p,s2~p,s3~u,s4~p) @'kp2'
A(s1~p,s2~u,s3~u,s4~p) -> A(s1~u,s2~u,s3~u,s4~p) @'ku2'
A(s1~p,s2~p,s3~p,s4~u) -> A(s1~p,s2~p,s3~p,s4~p) @'kp3'
A(s1~p,s2~p,s3~p,s4~u) -> A(s1~p,s2~p,s3~u,s4~u) @'ku3'
A(s1~p,s2~p,s3~p,s4~u) -> A(s1~p,s2~u,s3~p,s4~u) @'ku3'
A(s1~p,s2~p,s3~p,s4~u) -> A(s1~u,s2~p,s3~p,s4~u) @'ku3'
A(s1~p,s2~p,s3~p,s4~p) -> A(s1~p,s2~p,s3~p,s4~u) @'ku4'
A(s1~p,s2~p,s3~p,s4~p) -> A(s1~p,s2~p,s3~u,s4~p) @'ku4'
```

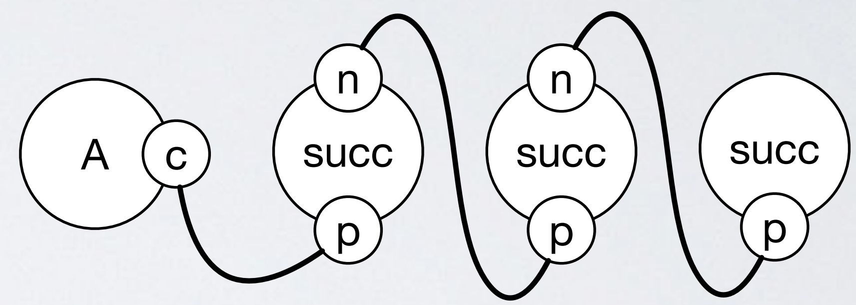
COUNTERS

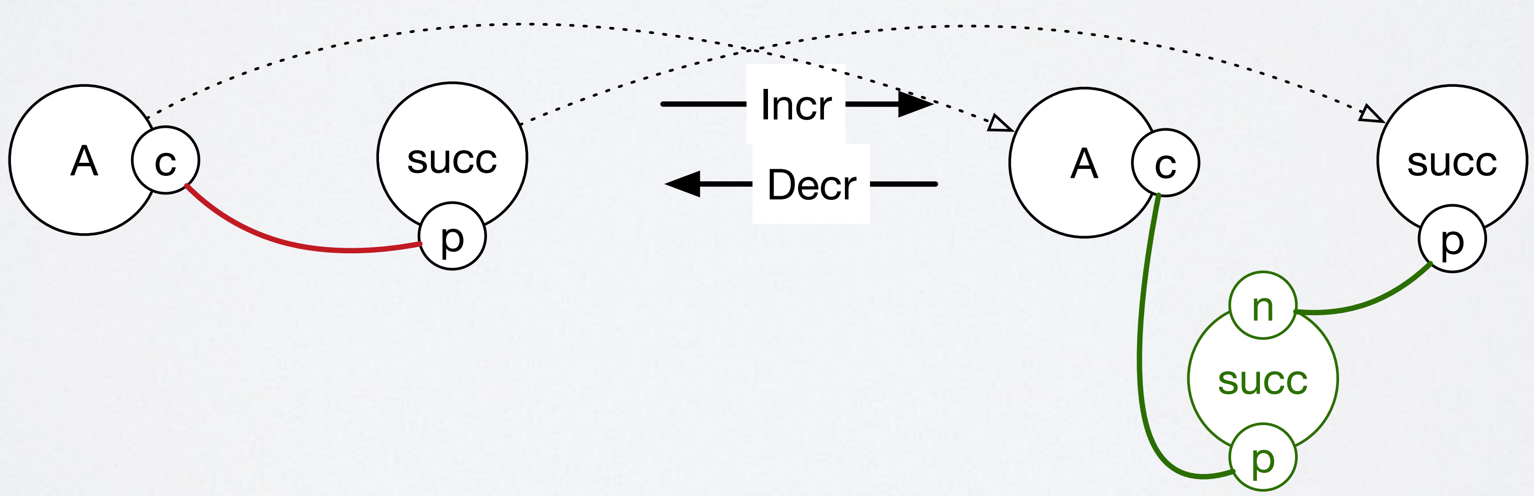
- Declaration: C(p: 0 += 8)
- Equality test: C(p: 1)
- Inequality test: C(p:> 2)
- Increment/Decrement: C(p += -1)

```
%agent: A(s1~u~p, s2~u~p, s3~u~p, s4~u~p, c:0 +=5)

%init: 100 A()

A(s1~u,c:x) -> A(s1~p,c +=1) @ 3*7^x
A(s2~u,c:x) -> A(s2~p,c +=1) @ 3*7^x
A(s3~u,c:x) -> A(s3~p,c +=1) @ 3*7^x
A(s4~u,c:x) -> A(s4~p,c +=1) @ 3*7^x

A(s1~p,c:x) -> A(s1~u,c += -1) @ 2*5^x
A(s2~p,c:x) -> A(s2~u,c += -1) @ 2*5^x
A(s3~p,c:x) -> A(s3~u,c += -1) @ 2*5^x
A(s4~p,c:x) -> A(s4~u,c += -1) @ 2*5^x

%plot: |A(c:0)|
%plot: |A(c:1)|
%plot: |A(c:2)|
%plot: |A(c:3)|
%plot: |A(c:4)|
```

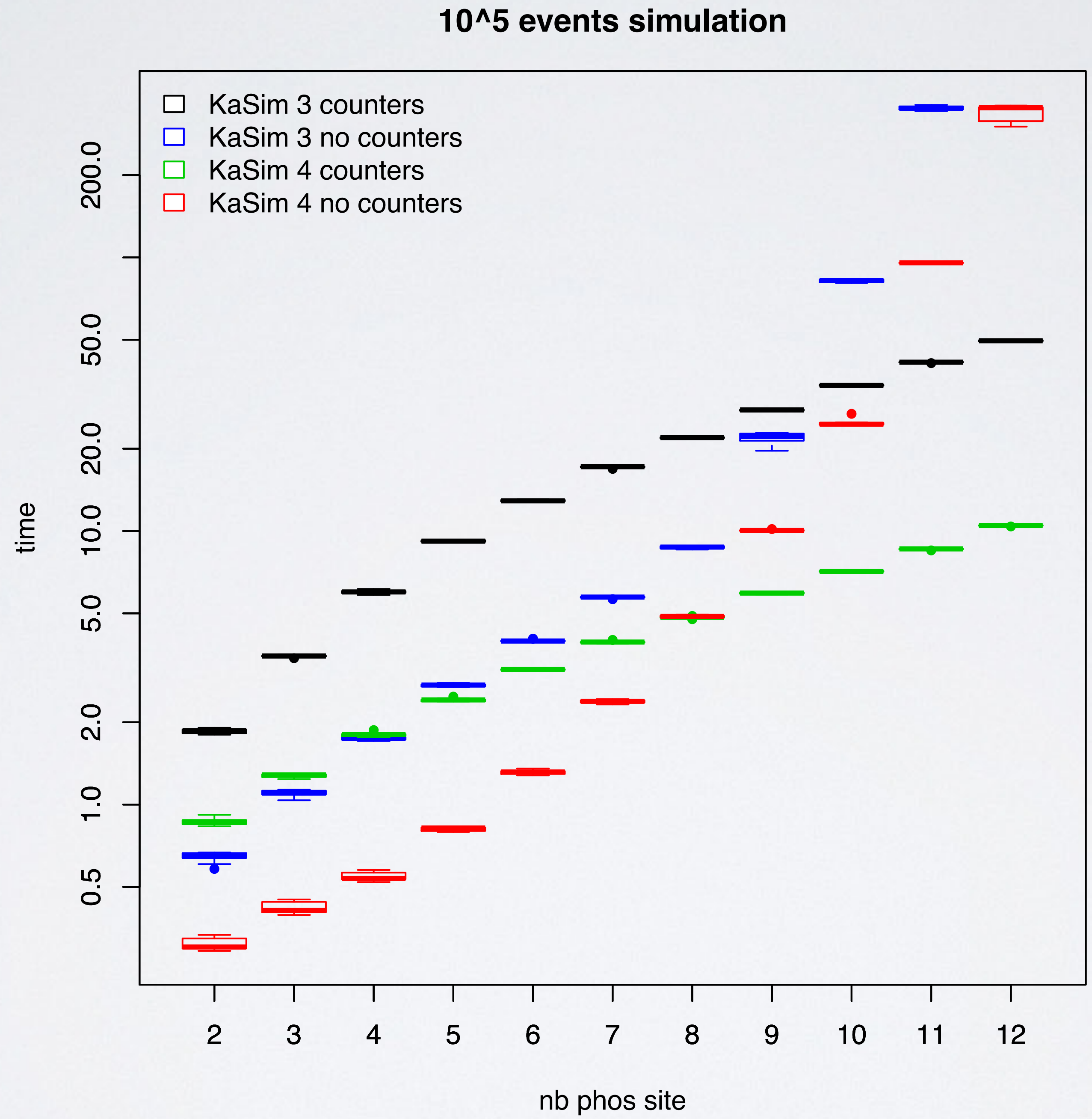
COUNTERS MACHINERY







EFFICIENCY



10^5 events simulation
KaSim 3 counters
KaSim 3 no counters
KaSim 4 counters
KaSim 4 no counters
time
0.5
1.0
2.0
5.0
10.0
20.0
50.0
200.0
2
3
4
5
6
7
8
9
10
11
12
nb phos site

THOUGHTS ON TRIVIALITY

- a posteriori triviality is not a priory triviality
- an easy encoding is not an easy language extension