Systems biology, models, and concurrency

Walter Fontana

Systems Biology
Harvard Medical School

with **Vincent Danos, Jean Krivine, Jerome Feret,** and **Russ Harmer**

The team

The concepts, software instruments, and models that I will talk about are the work of

Vincent Danos

computer science, math

Edinburgh & CNRS Paris

Jean Krivine

computer science

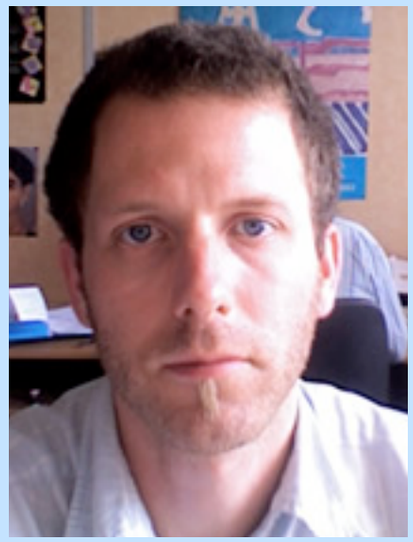
Edinburgh

Jérôme Feret

computer science

Harvard

Russ Harmer



computer science

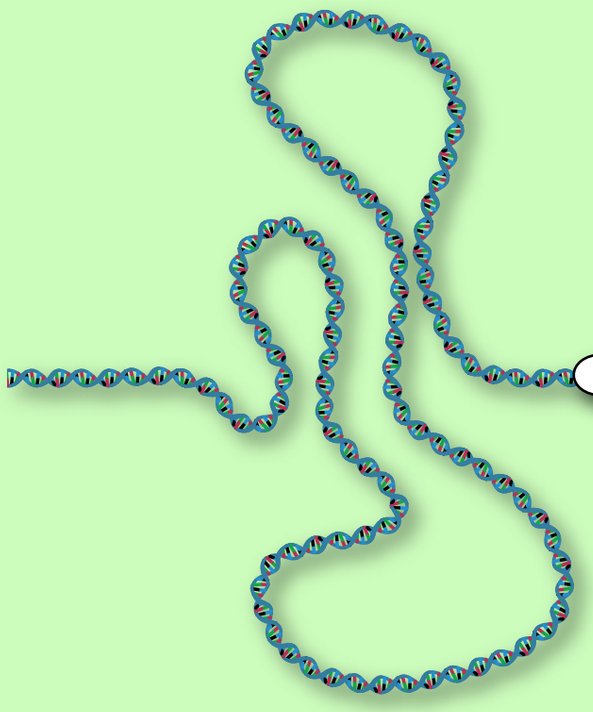
CNRS Paris

- changing the conditions for observability
  digitally isolate genes without using a phenotype

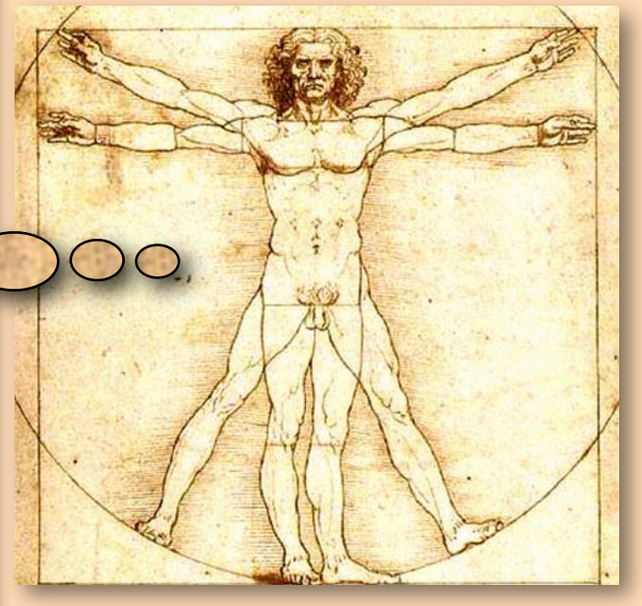
- making observable what was not observable before
  mechanisms of evolution

levels of analysis

mature	**missing**	?
tools for representing, managing, and analyzing sequence information	tools for representing, managing, and analyzing concurrent systems of molecular agents	tools for healthcare
• **genotype**	• molecular **processes**	• **phenotype**



metabolic systems
signaling systems
neural systems
structural systems
...

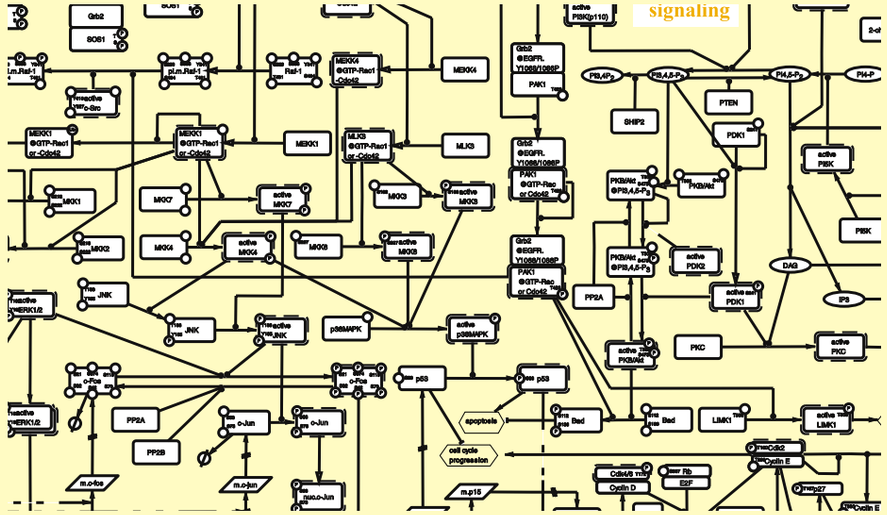


"easy" to understand; too hard a target for intervention	hard to understand; more accessible targets for intervention	

systematic biology

"... can be considered the large-scale, high-throughput collection of specific data sets and their organization and interpretation, usually through the application of advanced bioinformatics tools."

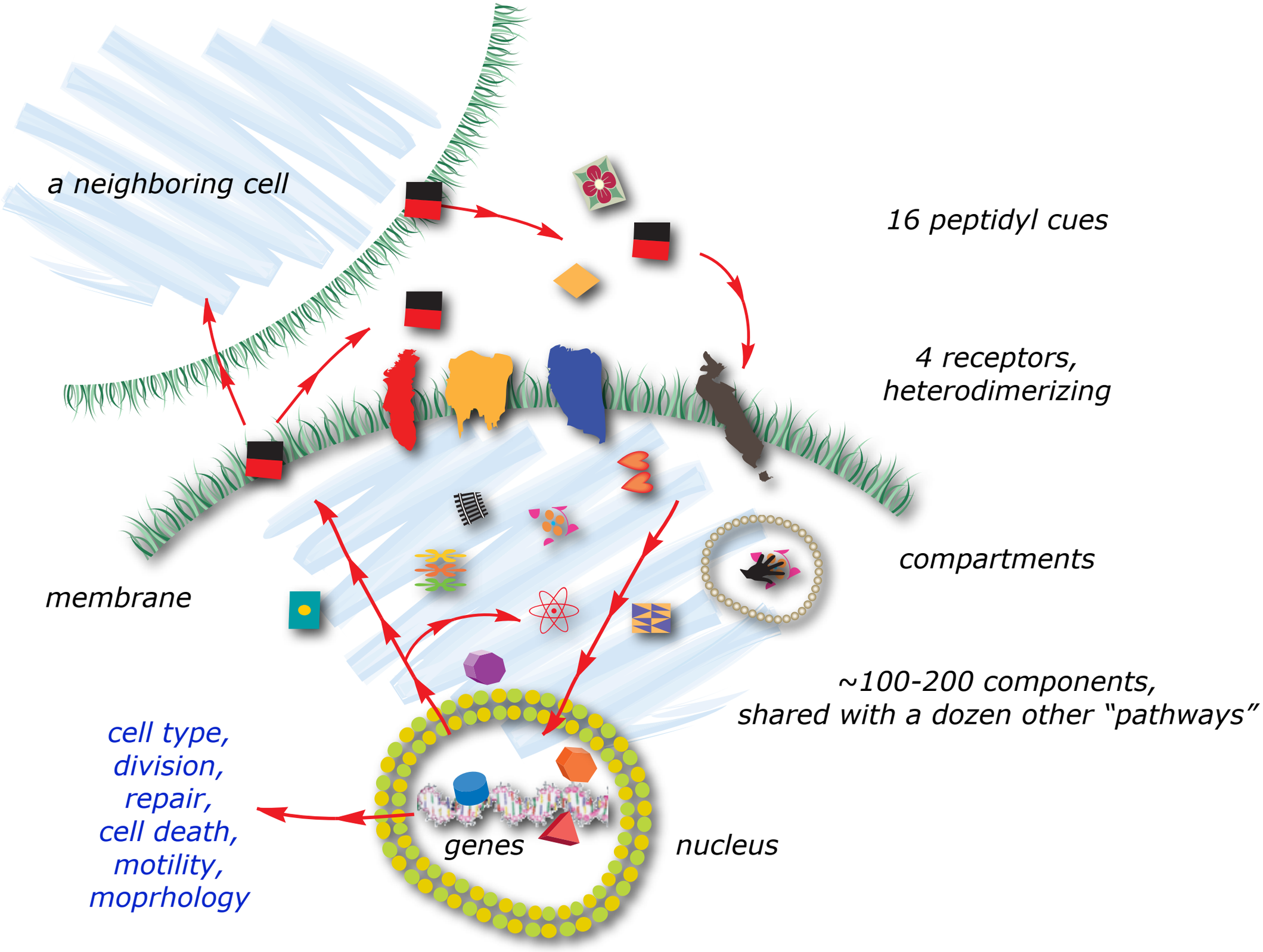
systems biology



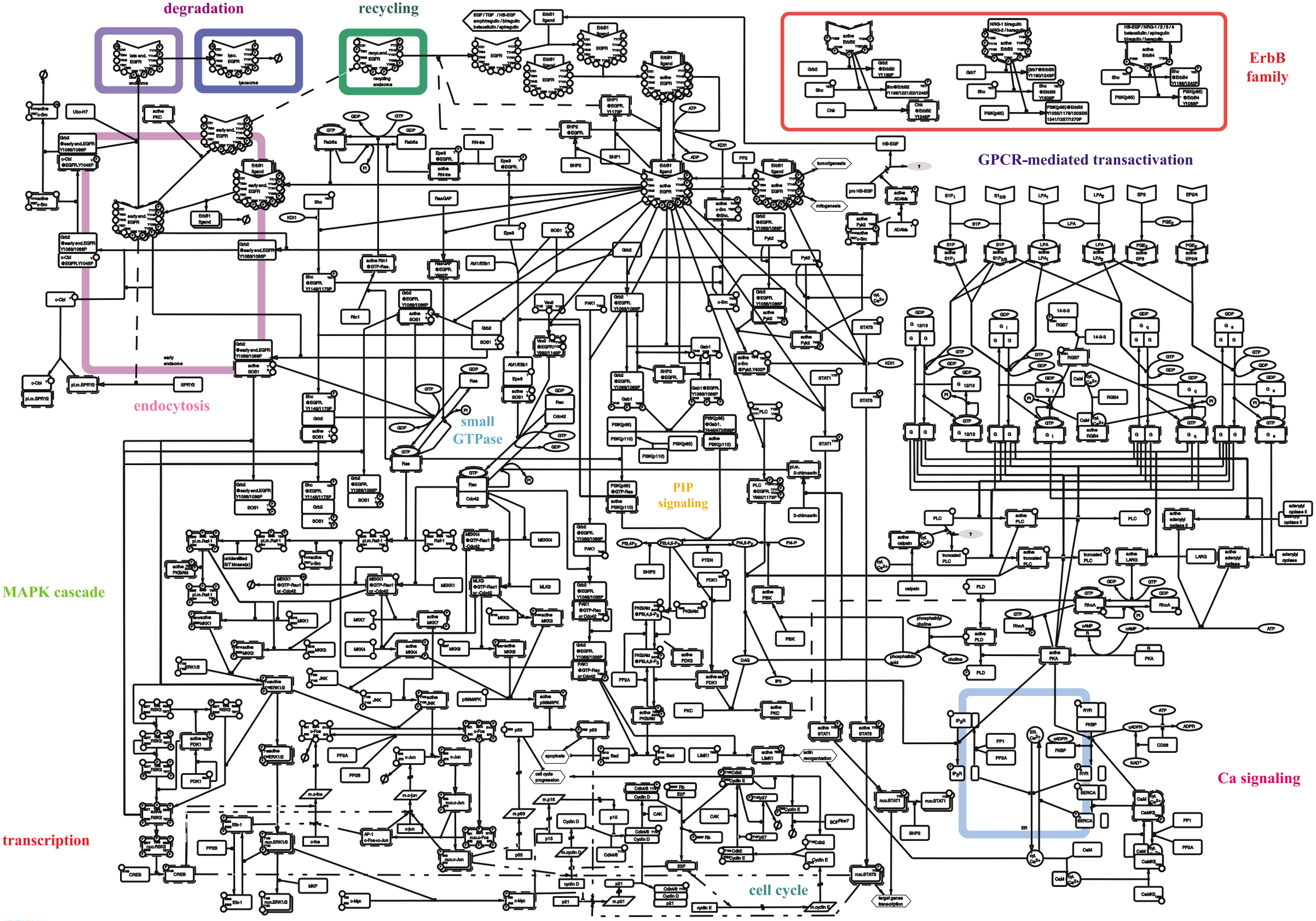
"... understanding [...] network behavior, and in particular [its] dynamic aspects, which requires the utilization of mathematical modeling tightly linked to experiment."

Marvin Cassman, WTEC Study on International Research and Development in Systems Biology, Chapter 1, p.1 (2005) [http://www.wtec.org/sysbio/]

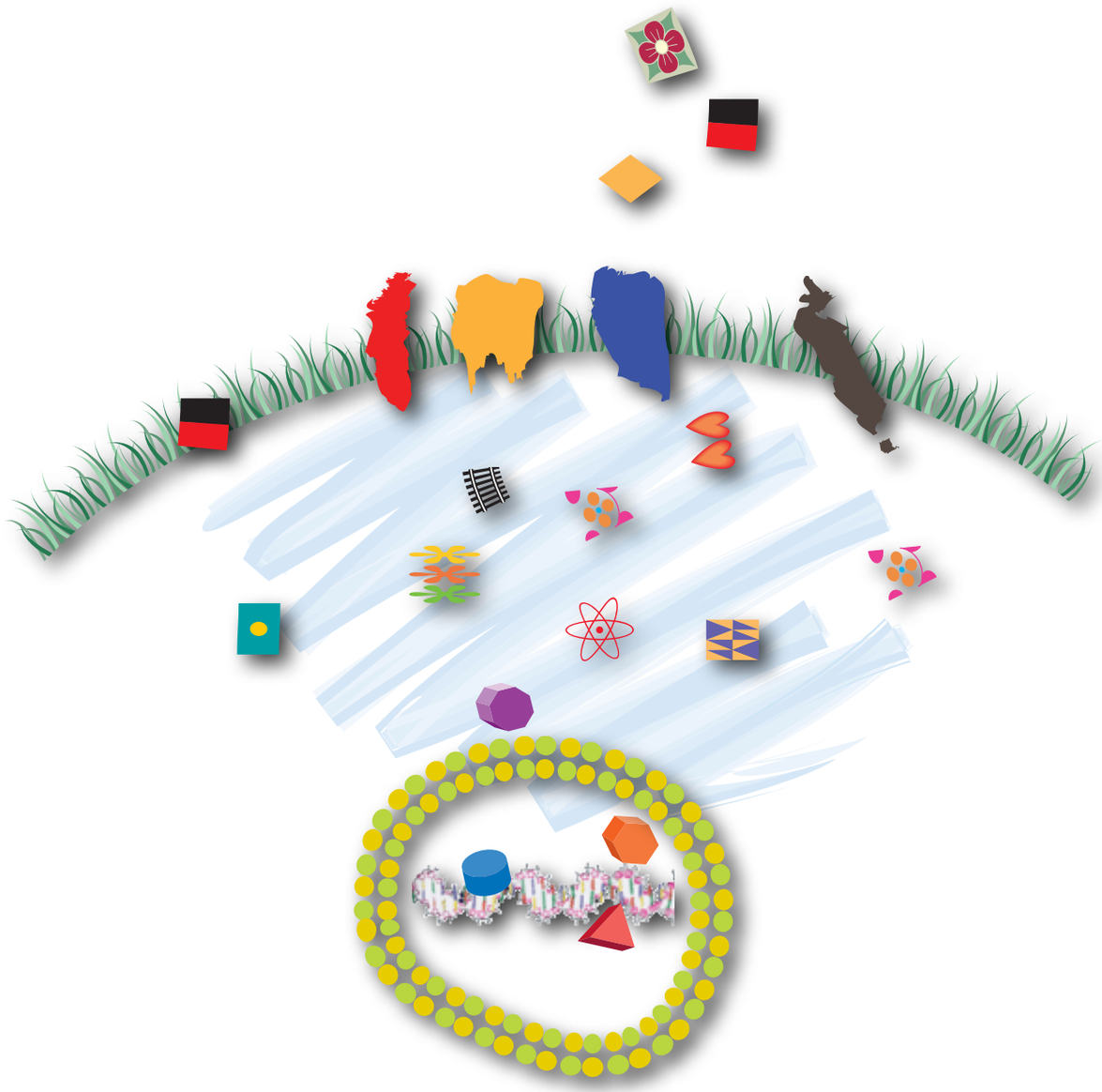
EGF / ErbB signaling (or any signaling system for that matter)



EGF map by Kitano et al

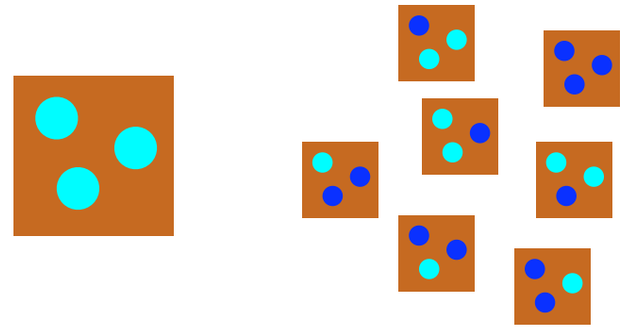


two fundamental problems that must be addressed

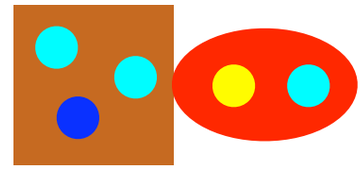


1. combinatorial state

post-translational modifications



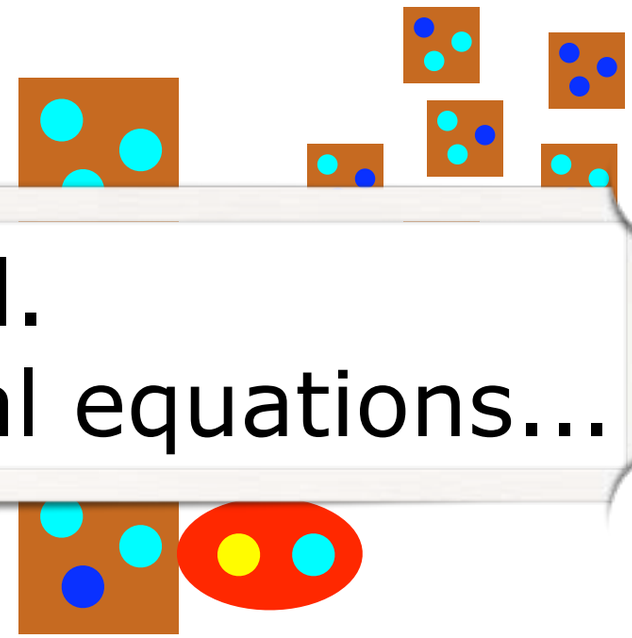
complex formation



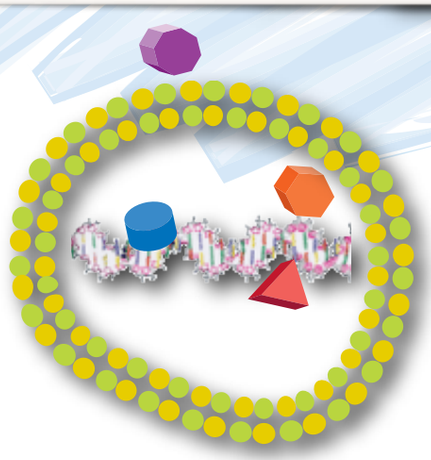
2. empirical facts are rapidly changing and scattered across research communities

1. combinatorial state

post-translational modifications

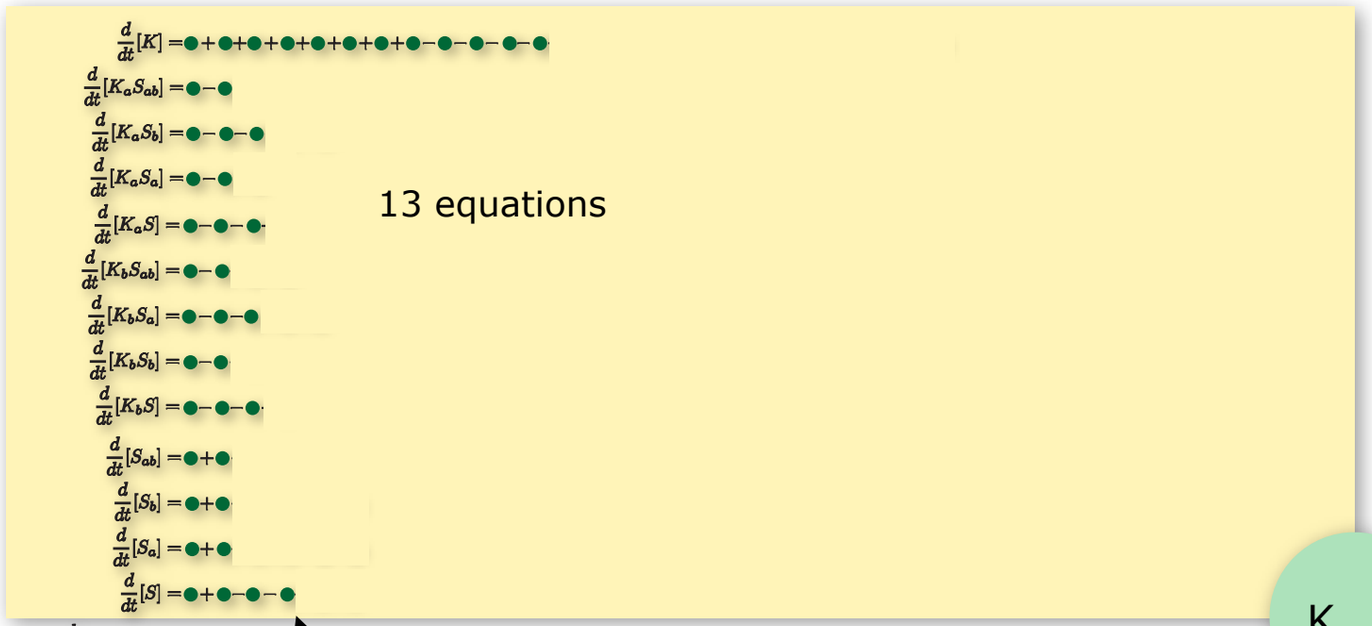


"Flat models" fail.
Take, for example, differential equations...

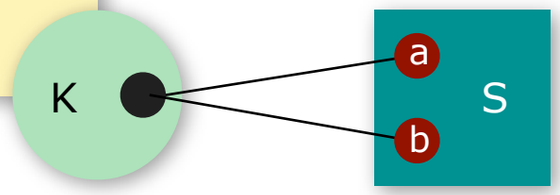


2. empirical facts are rapidly changing and scattered across research communities

flat models: differential equations



kinase-related terms



each dot is a particular reaction that contributes to the overall change of the molecular species (“state”) referred to on the left of the equation

flat models explode: differential equations

$$\frac{d}{dt}[K] = \bullet + \bullet + \bullet + \bullet + \bullet + \bullet + \bullet + \bullet - \bullet - \bullet - \bullet - \bullet - \bullet + \bullet + \bullet + \bullet + \bullet + \bullet + \bullet + \bullet + \bullet + \bullet - \bullet - \bullet - \bullet - \bullet$$

$$\frac{d}{dt}[K_a S_{ab}] = \bullet - \bullet + \bullet + \bullet$$

$$\frac{d}{dt}[K_a S_b] = \bullet - \bullet - \bullet + \bullet + \bullet$$

$$\frac{d}{dt}[K_a S_a] = \bullet - \bullet + \bullet$$

$$\frac{d}{dt}[K_a S] = \bullet - \bullet - \bullet + \bullet$$

$$\frac{d}{dt}[K_b S_{ab}] = \bullet - \bullet + \bullet + \bullet$$

$$\frac{d}{dt}[K_b S_a] = \bullet - \bullet - \bullet + \bullet + \bullet$$

$$\frac{d}{dt}[K_b S_b] = \bullet - \bullet + \bullet$$

$$\frac{d}{dt}[K_b S] = \bullet - \bullet - \bullet + \bullet$$

$$\frac{d}{dt}[S_{ab}] = \bullet + \bullet + \bullet + \bullet$$

$$\frac{d}{dt}[S_b] = \bullet + \bullet + \bullet + \bullet - \bullet - \bullet$$

$$\frac{d}{dt}[S_a] = \bullet + \bullet + \bullet + \bullet - \bullet - \bullet$$

$$\frac{d}{dt}[S] = \bullet + \bullet - \bullet - \bullet + \bullet + \bullet$$

13 equations, each of which had to be changed

$$\frac{d}{dt}[K_a P_b S_{ab}] = \bullet + \bullet - \bullet - \bullet - \bullet$$

$$\frac{d}{dt}[K_a P_b S_b] = \bullet + \bullet - \bullet - \bullet - \bullet - \bullet$$

$$\frac{d}{dt}[K_a P_b S_a] = \bullet + \bullet - \bullet - \bullet$$

$$\frac{d}{dt}[K_a P_b S] = \bullet + \bullet - \bullet - \bullet - \bullet$$

$$\frac{d}{dt}[K_b P_a S_{ab}] = \bullet + \bullet - \bullet - \bullet - \bullet$$

$$\frac{d}{dt}[K_b P_a S_b] = \bullet + \bullet - \bullet - \bullet$$

$$\frac{d}{dt}[K_b P_a S_a] = \bullet + \bullet - \bullet - \bullet - \bullet - \bullet$$

$$\frac{d}{dt}[K_b P_a S] = \bullet + \bullet - \bullet - \bullet - \bullet$$

$$\frac{d}{dt}[P] = \bullet + \bullet - \bullet - \bullet - \bullet - \bullet - \bullet - \bullet - \bullet - \bullet - \bullet - \bullet - \bullet - \bullet$$

$$\frac{d}{dt}[P_a P_b S_{ab}] = \bullet + \bullet - \bullet - \bullet - \bullet - \bullet$$

$$\frac{d}{dt}[P_a P_b S_b] = \bullet + \bullet - \bullet - \bullet - \bullet$$

$$\frac{d}{dt}[P_a P_b S_a] = \bullet + \bullet - \bullet - \bullet - \bullet$$

$$\frac{d}{dt}[P_a P_b S] = \bullet + \bullet - \bullet - \bullet$$

$$\frac{d}{dt}[P_a S_{ab}] = \bullet - \bullet - \bullet + \bullet$$

$$\frac{d}{dt}[P_a S_b] = \bullet - \bullet + \bullet + \bullet$$

$$\frac{d}{dt}[P_a S_a] = \bullet - \bullet - \bullet + \bullet$$

$$\frac{d}{dt}[P_a S] = \bullet - \bullet + \bullet$$

$$\frac{d}{dt}[P_b S_{ab}] = \bullet - \bullet - \bullet + \bullet$$

$$\frac{d}{dt}[P_b S_b] = \bullet - \bullet - \bullet + \bullet$$

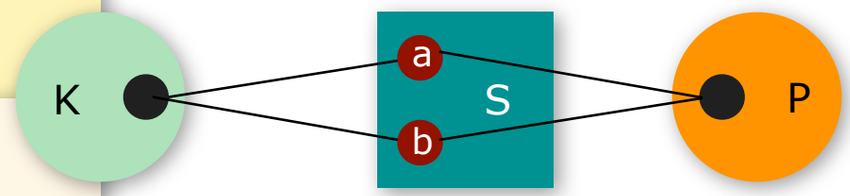
$$\frac{d}{dt}[P_b S_a] = \bullet - \bullet + \bullet$$

$$\frac{d}{dt}[P_b S] = \bullet - \bullet + \bullet$$

+ 21 new equations

kinase-related terms

phosphatase-related terms

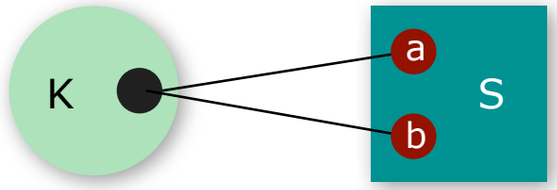


adding a phosphatase

flat models are opaque: differential equations

$$\frac{d}{dt}x_1 = x_5 + x_9 + x_4 + x_3 + x_7 + x_8 + x_2 + x_6 - x_1x_{13} - x_1x_{13} - x_1x_{12} - x_1x_{11}$$
$$\frac{d}{dt}x_2 = x_3 - x_2$$
$$\frac{d}{dt}x_3$$
$$\frac{d}{dt}x_5 = x_1x_{13} - x_5 - x_5$$
$$\frac{d}{dt}x_6 = x_7 - x_6$$
$$\frac{d}{dt}x_7 = x_1x_{12} - x_7 - x_7$$
$$\frac{d}{dt}x_8 = x_9 - x_8$$
$$\frac{d}{dt}x_9 = x_1x_{13} - x_9 - x_9$$
$$\frac{d}{dt}x_{10} = x_2 + x_6$$
$$\frac{d}{dt}x_{11} = x_3 + x_8 - x_1x_{11}$$
$$\frac{d}{dt}x_{12} = x_4 + x_7 - x_1x_{12}$$
$$\frac{d}{dt}x_{13} = x_5 + x_9 - x_1x_{13} - x_1x_{13}$$

hard to decipher what the facts are that went into this model...



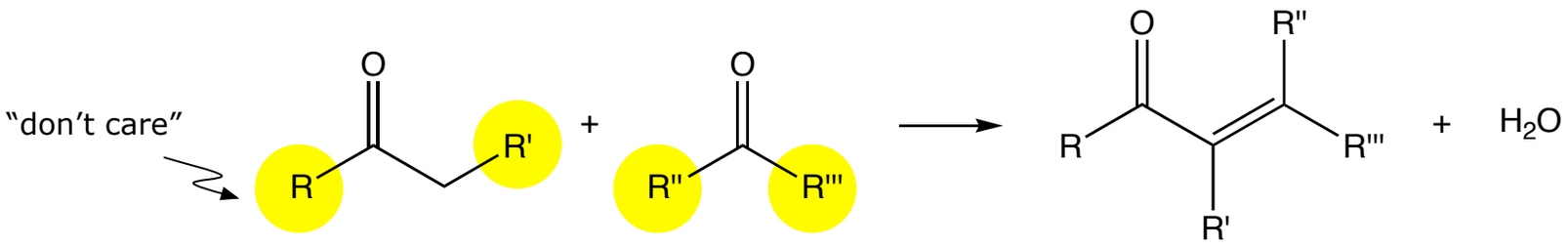
$$x_1 = [K]$$
$$x_2 = [K_a S_{ab}]$$
$$x_3 = [K_a S_b]$$
$$x_4 = [K_a S_a]$$
$$x_5 = [K_a S]$$
$$x_{10} = [S_{ab}]$$
$$x_{11} = [S_b]$$
$$x_{12} = [S_a]$$
$$x_{13} = [S]$$

need key... but the names in the key are only mnemonic - they are not part of the formalism.

We've been there before. The transition from Alchemy to chemistry was a formal language and a system of rules to codify transformations.

rule

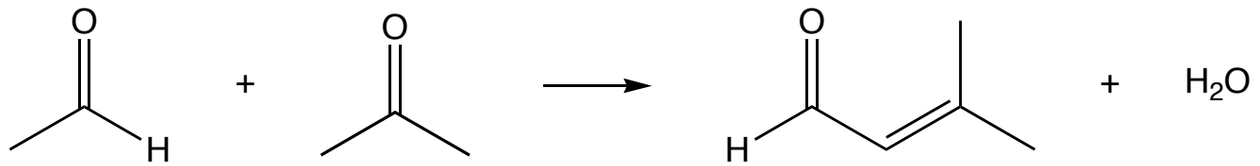
the rule (pattern) of an aldol condensation



instance

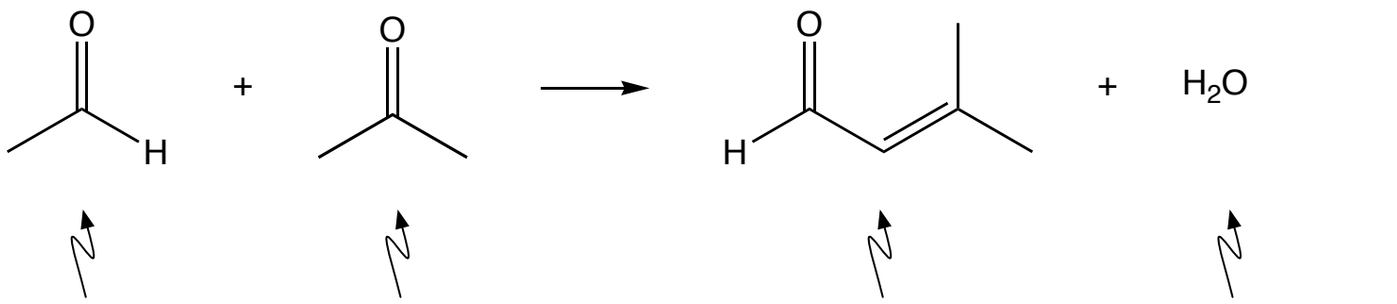
an instance of an aldol condensation (a "flat" or "fully contextualized" reaction)

R=H, R'=H, R''=CH_3,
R'''=CH_3



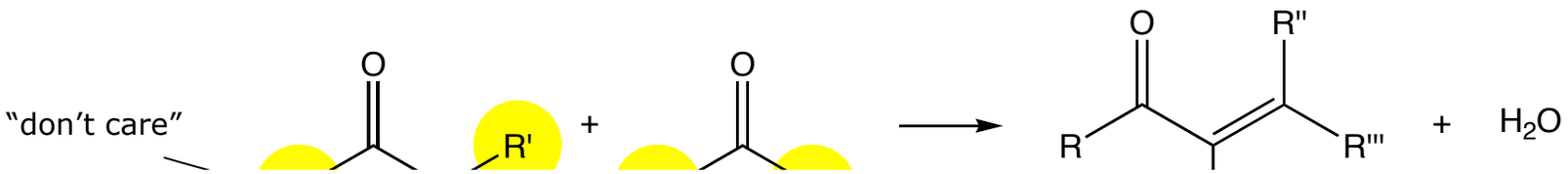
event

an event of an aldol condensation refers to particular molecular instances



rule

the rule (pattern) of an aldol condensation

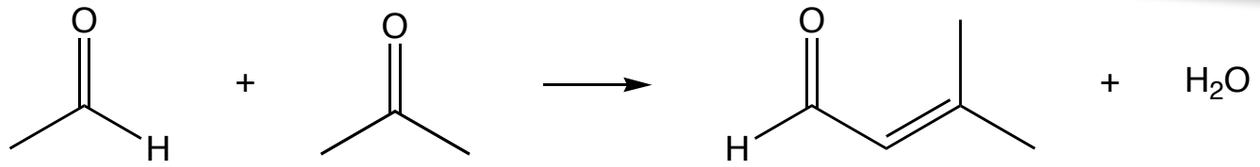


But what is an appropriate language for molecular biology?

instance

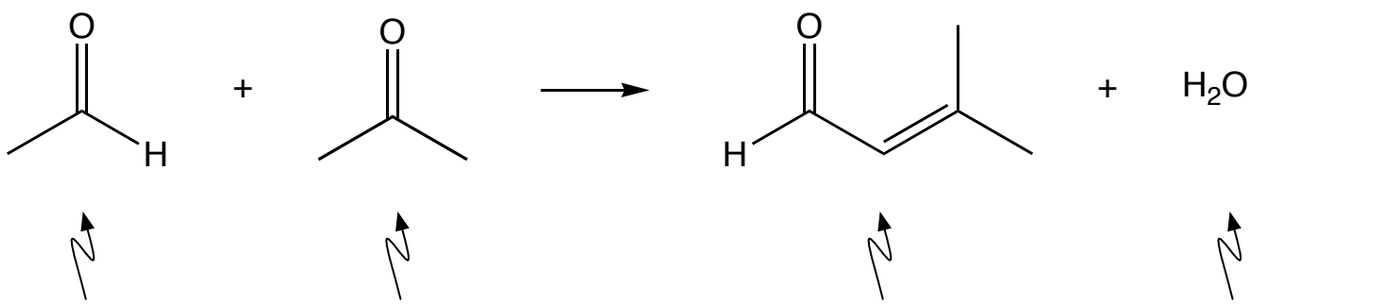
an

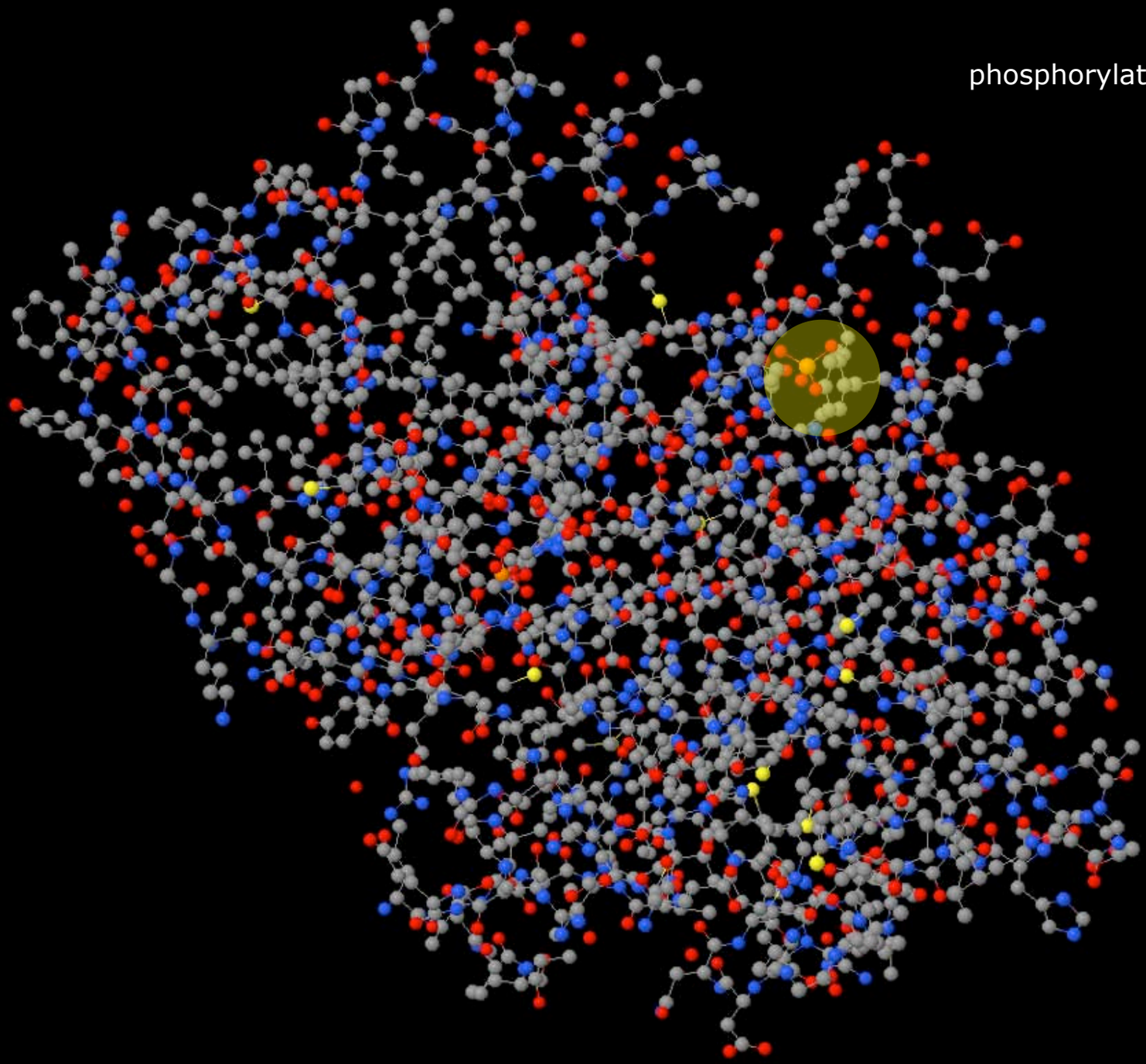
R=H, R'=H, R''=CH_3,
R'''=CH_3



event

an event of an aldol condensation refers to particular molecular instances

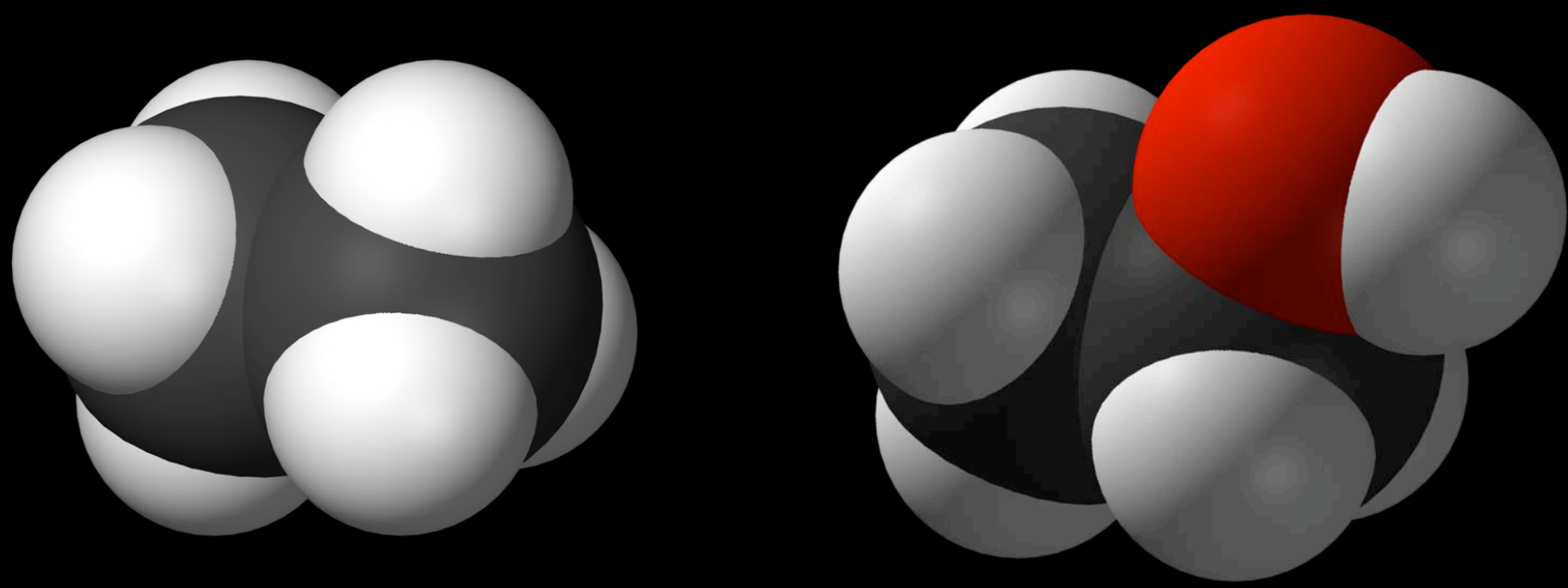




phosphorylated RET tyrosine kinase domain

It seems justified to speak of “the same molecule in a different state”.

agent identity



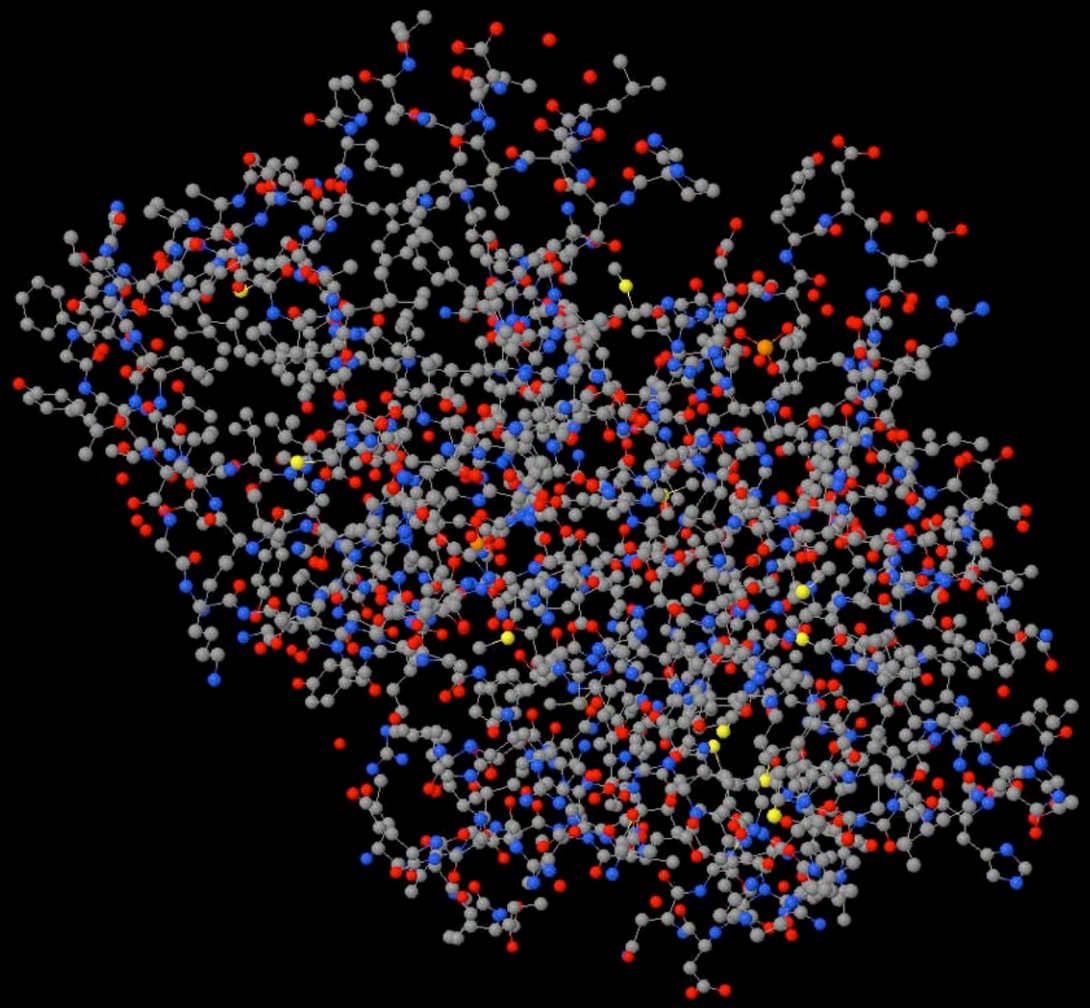
ethane

ethanol

Not “the same molecule in a different state”, but rather two different types of molecules!

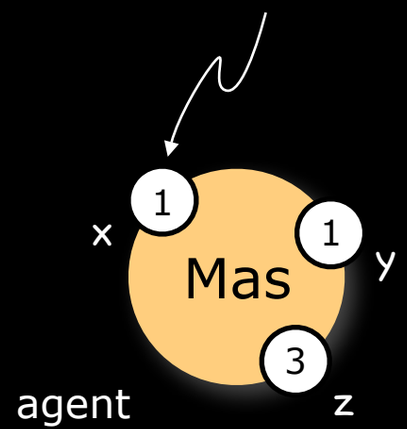
a spherical cow...

the world to a structural biologist



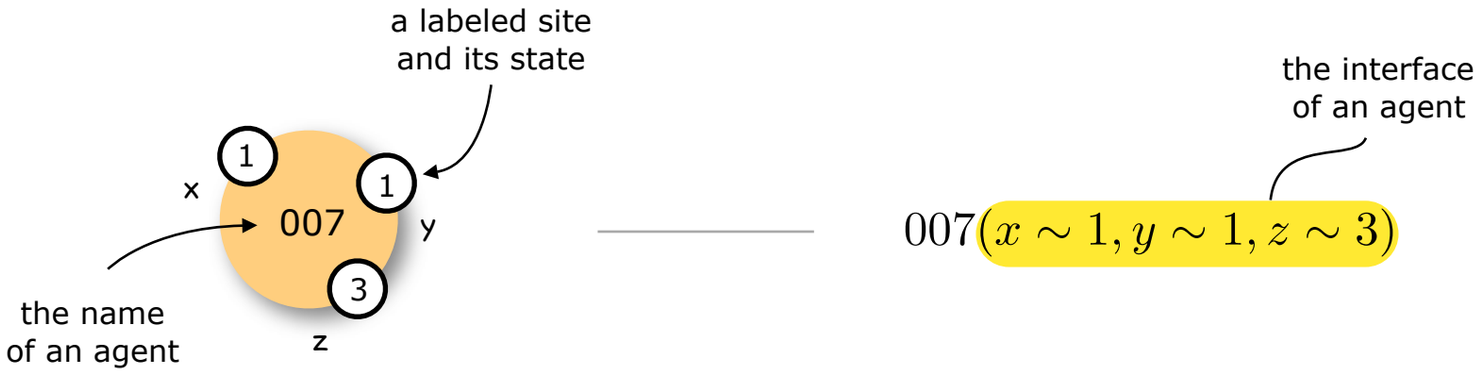
the world in Kappa

a *site with a state* represents
an interaction capability

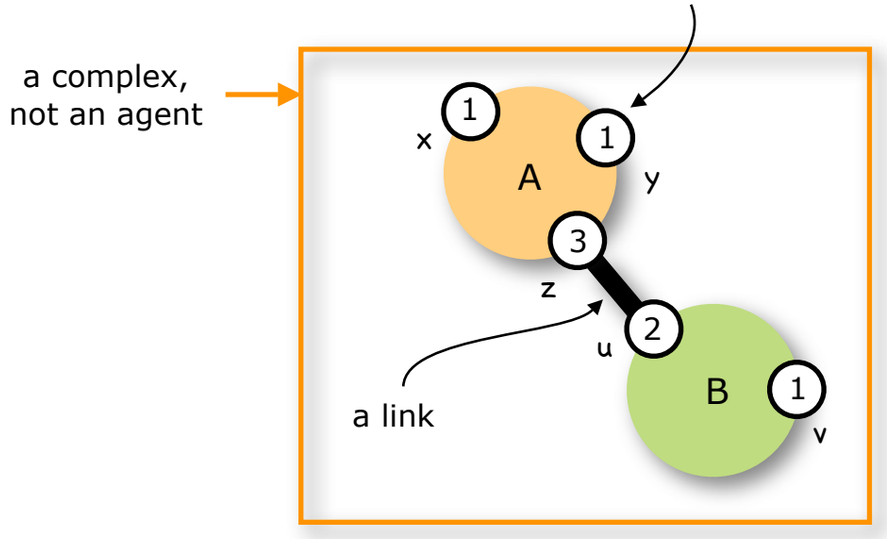


V. Danos and C. Laneve, “Formal molecular biology”,
Theoretical Computer Science, **325**, 69-110 (2004)

Kappa: fully specified agents and complexes



a free site in state 1

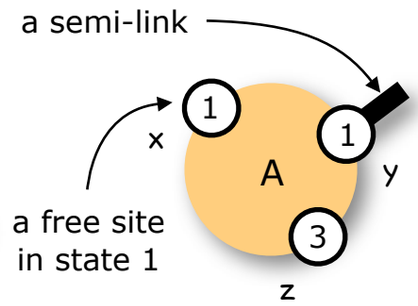


V. Danos and C. Laneve, "Formal molecular biology", *Theoretical Computer Science*, **325**, 69-110 (2004)

Kappa: partially specified agents and complexes - don’t care, don’t write

patterns

binding partner is unspecified

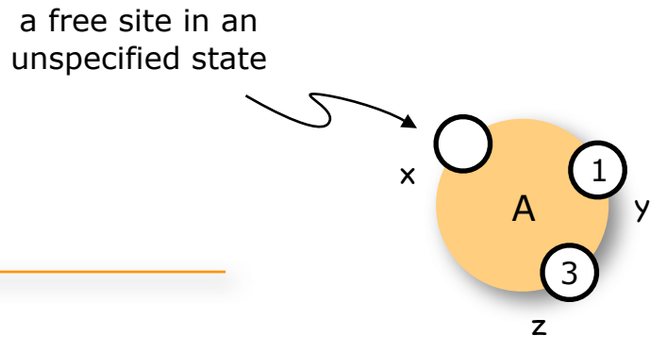


$A(x \sim 1, y \sim 1!\_, z \sim 3)$

binding state is unspecified

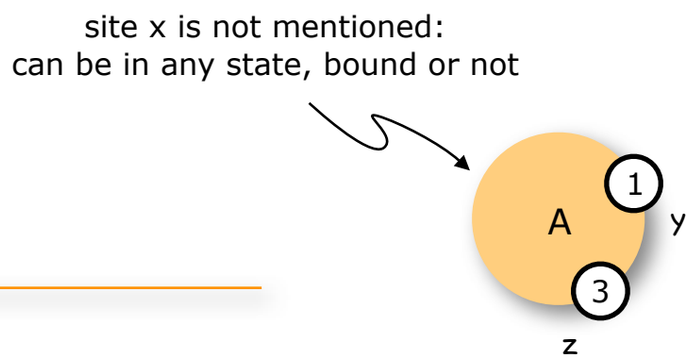
$A(x \sim 1, y \sim 1?, z \sim 3)$

state is unspecified



$A(x, y \sim 1, z \sim 3)$

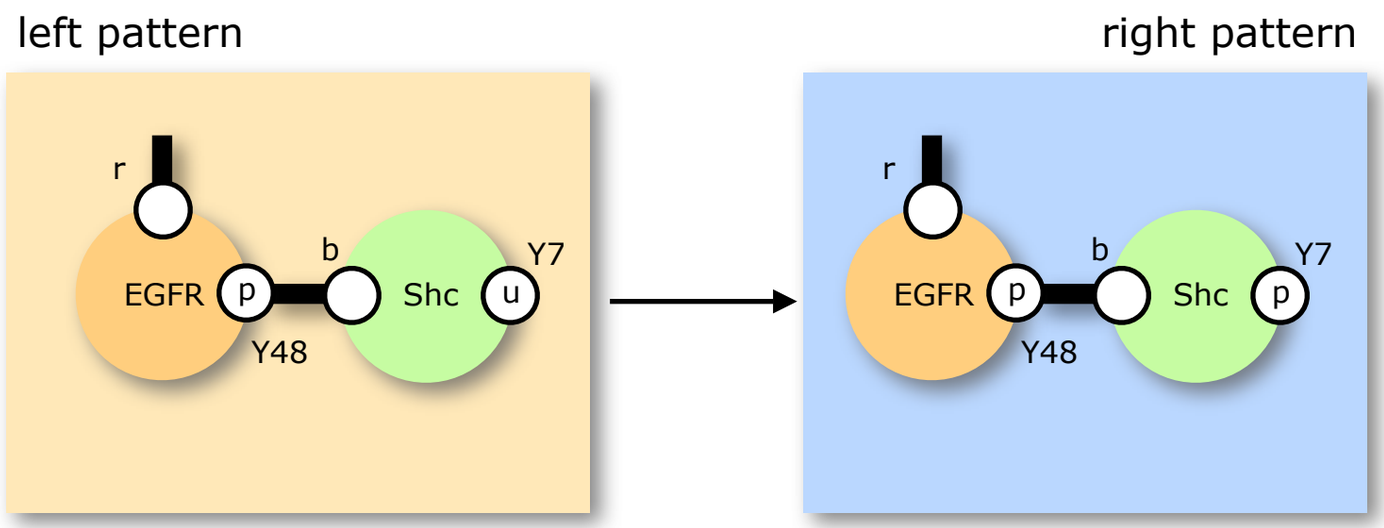
site is not mentioned



$A(y \sim 1, z \sim 3)$

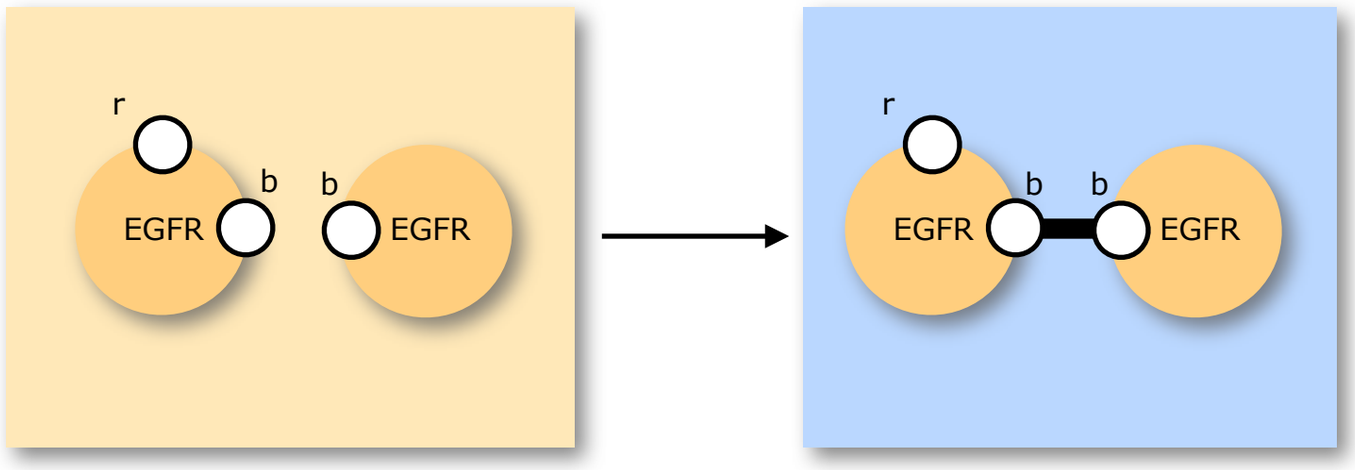
Kappa: rules

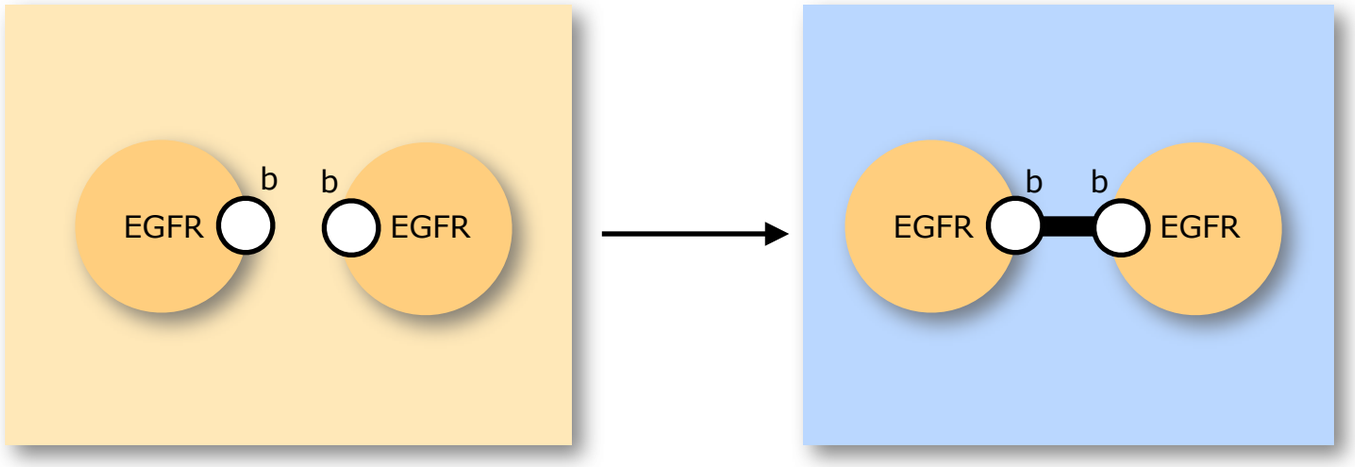
a rule rewrites a pattern:



$$EGFR(r!\_, Y48 \sim p!1), Shc(b!1, Y7 \sim u) \longrightarrow EGFR(r!\_, Y48 \sim p!1), Shc(b!1, Y7 \sim p)$$

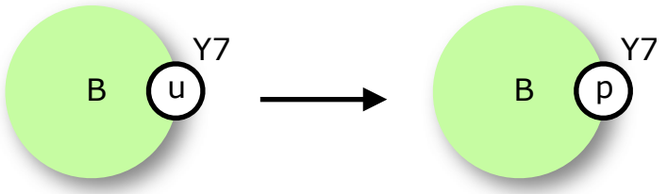
two different rules (remember: don’t care, don’t write)





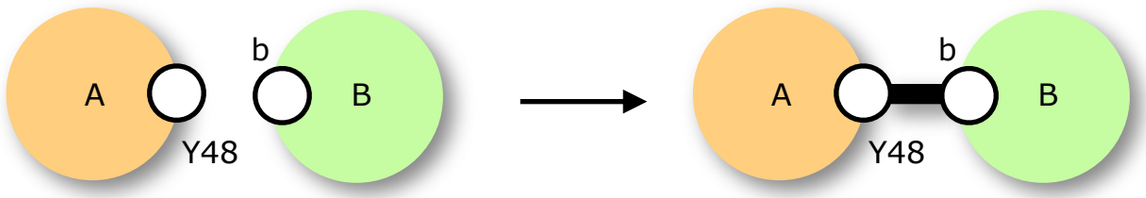
the language (kappa) : elementary actions

modify state



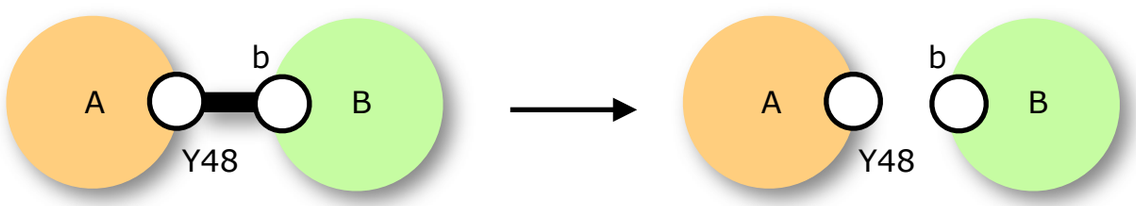
$$B(Y7 \sim u) \longrightarrow B(Y7 \sim p)$$

bind



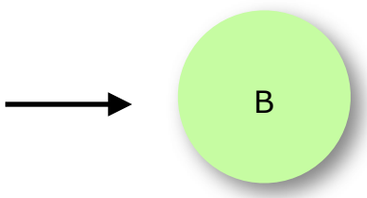
$$A(Y48), B(b) \longrightarrow A(Y48!0), B(b!0)$$

unbind



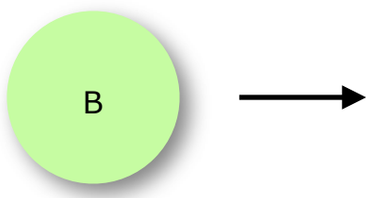
$$A(Y48!0), B(b!0) \longrightarrow A(Y48), B(b)$$

introduce



$$\longrightarrow B()$$

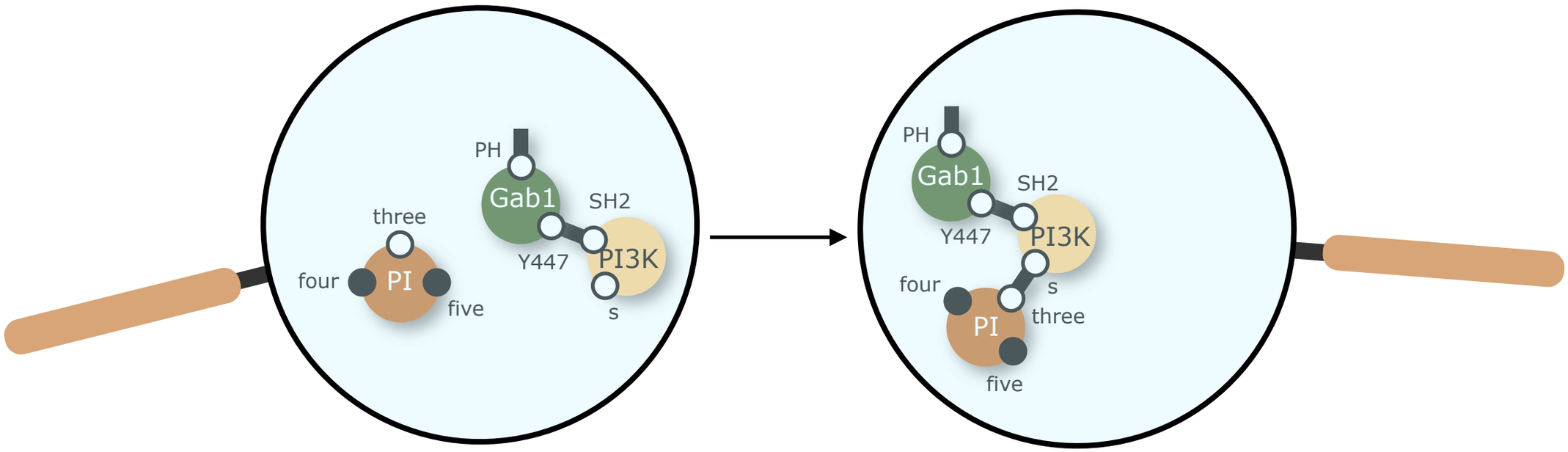
delete

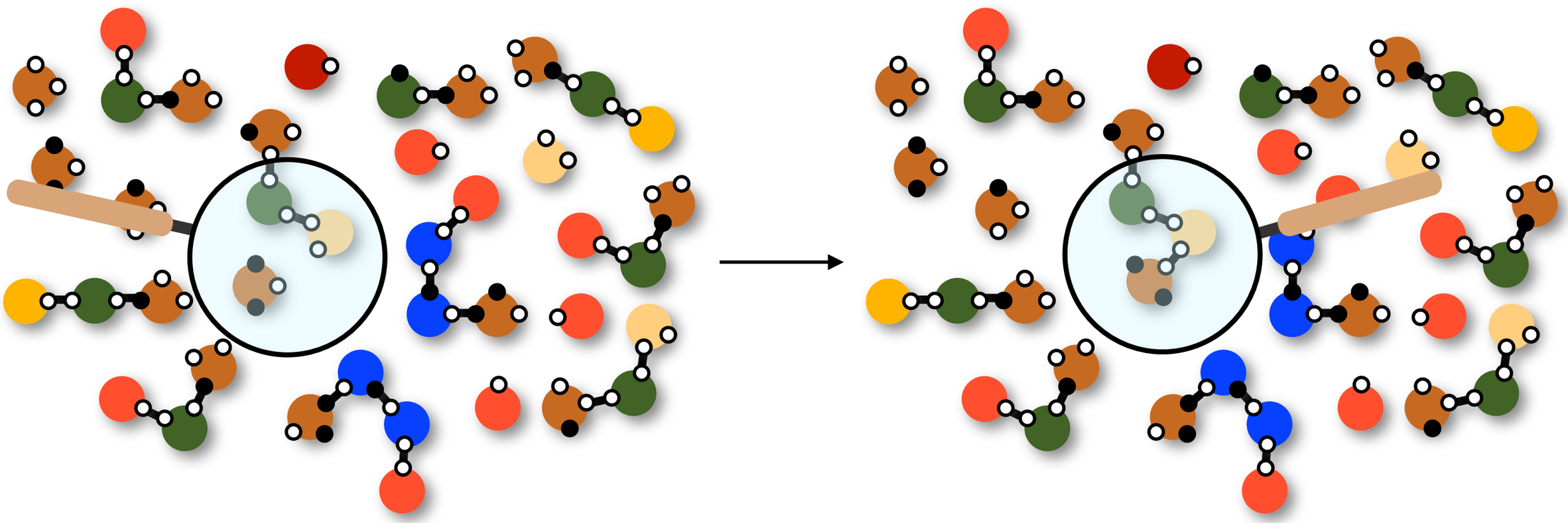


$$B() \longrightarrow$$

rule application

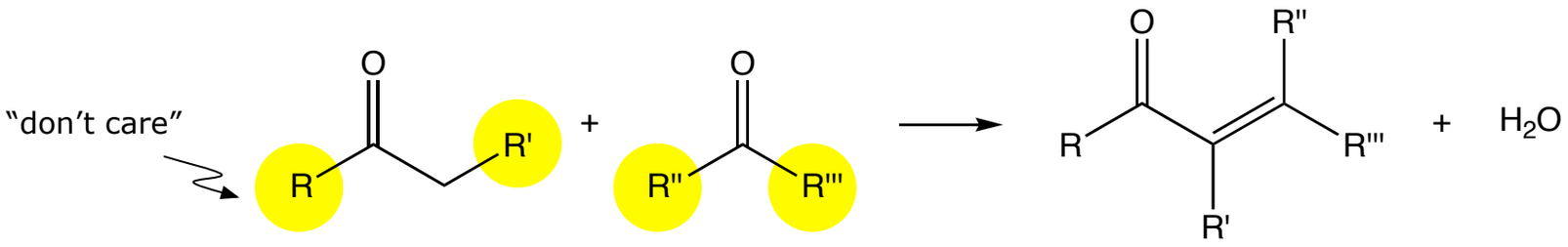
Gab1(PH!_,Y447!1),PI3K(SH2!1,s),PI(three~u,four~p,five~p) -> Gab1(PH!_,Y447!1),PI3K(SH2!1,s!2),PI(three~u!2,four~p,five~p)





rule

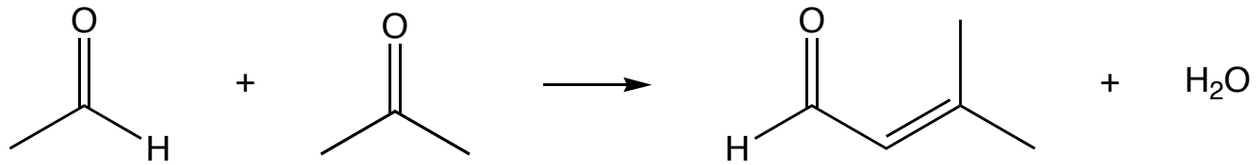
the rule (pattern) of an aldol condensation



instance

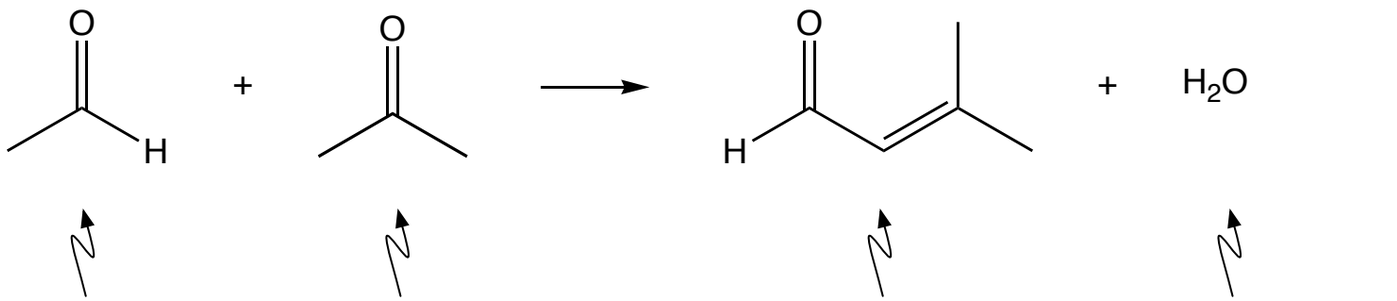
an instance of an aldol condensation (a "flat" or "fully contextualized" reaction)

R=H, R'=H, R''=CH_3,
R'''=CH_3



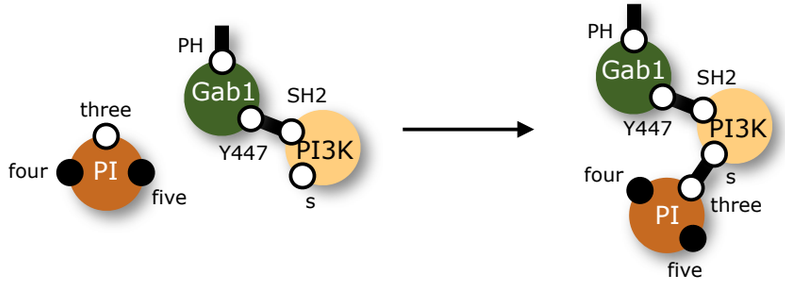
event

an event of an aldol condensation refers to particular molecular instances

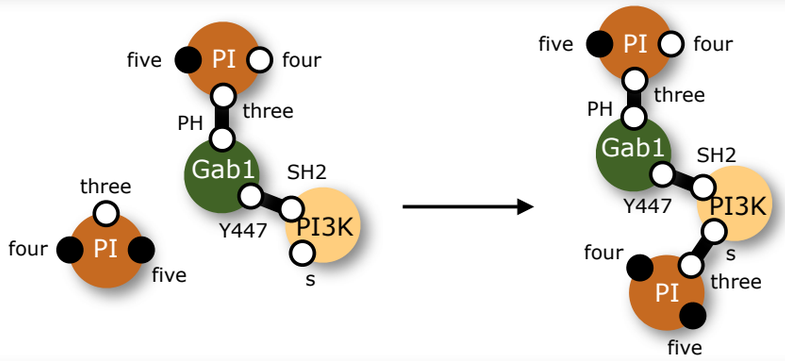


much like the familiar: rules, instances, and events in Kappa

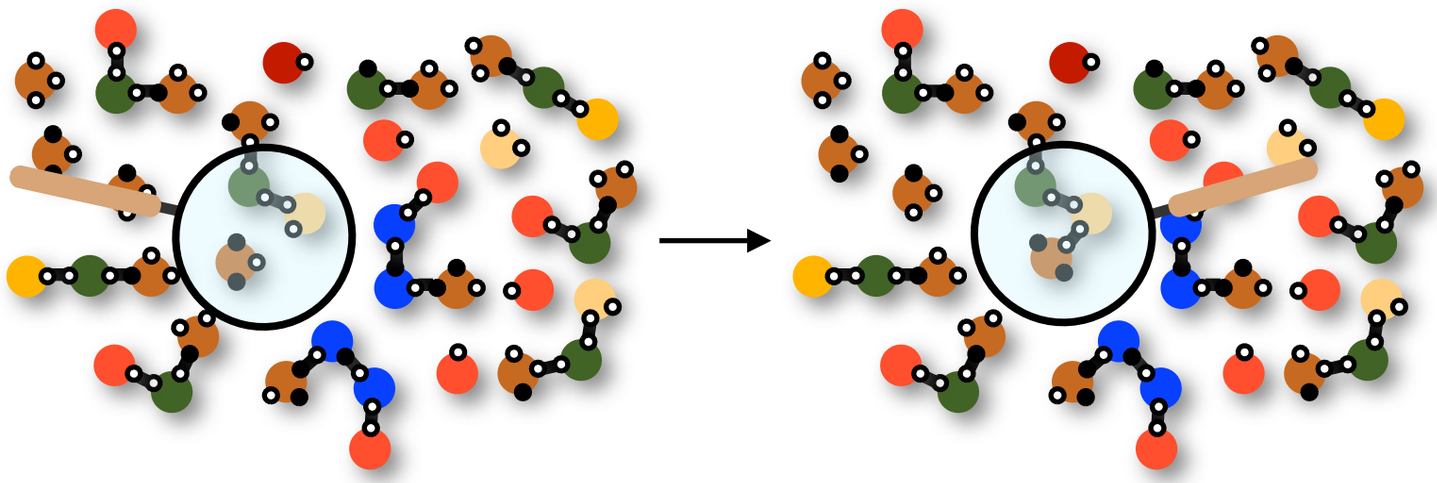
rule



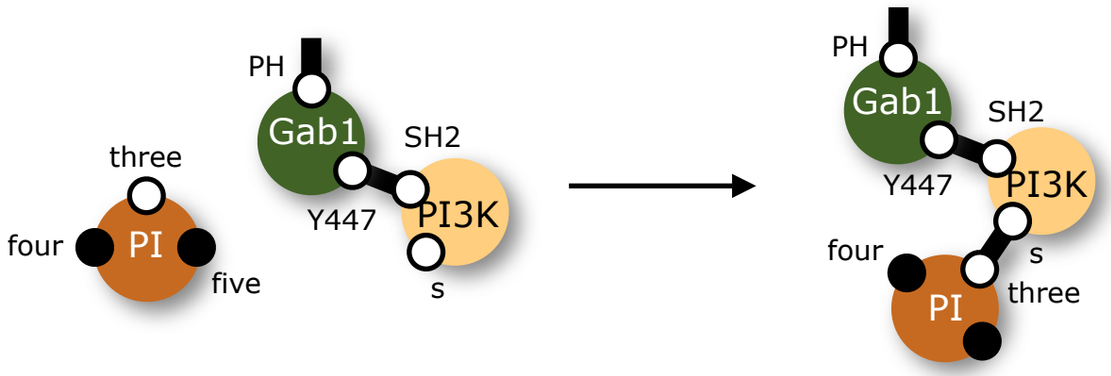
instance

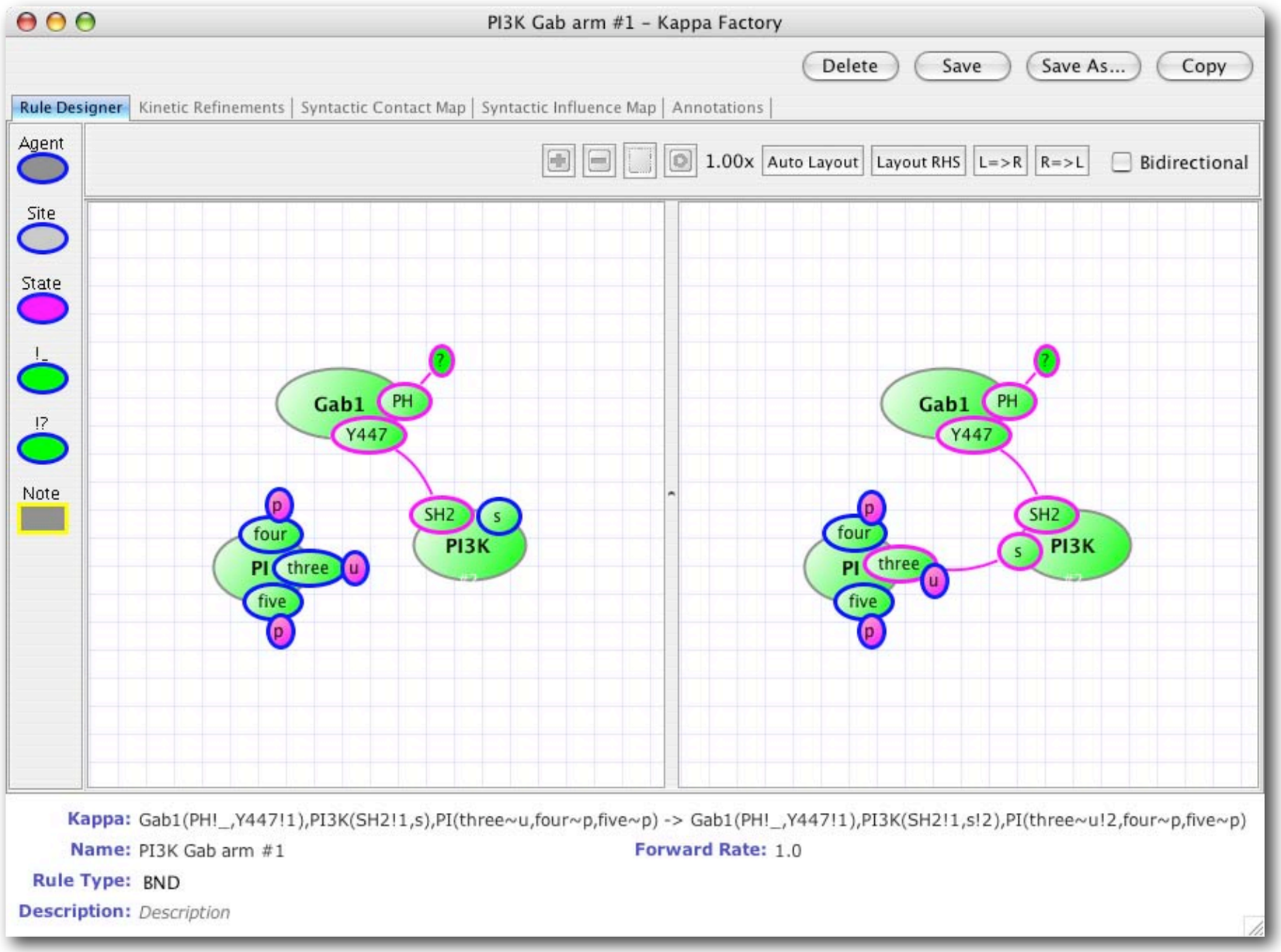


event



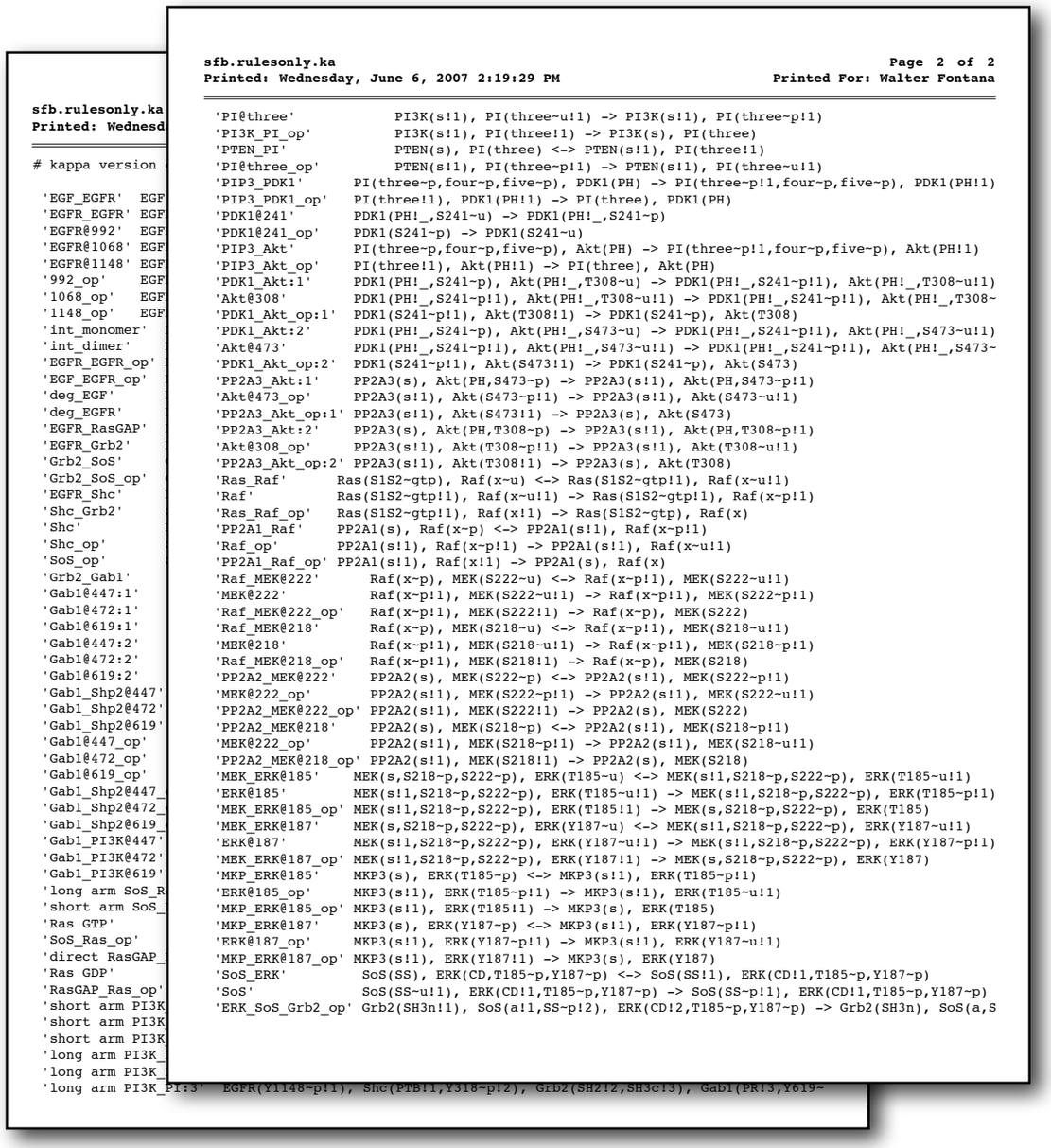
Kappa with GUI: the *KappaFactory*





Russ Harmer’s EGFR system with Akt

18 868 902 333 942 991 917 533 566 435 344 349 possible molecular species



sfb.rulesonly.ka Page 2 of 2
Printed: Wednesday, June 6, 2007 2:19:29 PM Printed For: Walter Fontana

```
'PI@three'            PI3K(s!1), PI(three~u!1) -> PI3K(s!1), PI(three~p!1)
'PI3K_PI_op'          PI3K(s!1), PI(three!1) -> PI3K(s), PI(three)
'PTEN_PI'             PTEN(s), PI(three) <-> PTEN(s!1), PI(three!1)
'PI@three_op'         PTEN(s!1), PI(three~p!1) -> PTEN(s!1), PI(three~u!1)
'PIP3_PDK1'      PI(three~p,four~p,five~p), PDK1(PH) -> PI(three~p!1,four~p,five~p), PDK1(PH!1)
'PIP3_PDK1_op'   PI(three!1), PDK1(PH!1) -> PI(three), PDK1(PH)
'PDK1@241'       PDK1(PH!_,S241~u) -> PDK1(PH!_,S241~p)
'PDK1@241_op'    PDK1(S241~p) -> PDK1(S241~u)
'PIP3_Akt'       PI(three~p,four~p,five~p), Akt(PH) -> PI(three~p!1,four~p,five~p), Akt(PH!1)
'PIP3_Akt_op'    PI(three!1), Akt(PH!1) -> PI(three), Akt(PH)
'PDK1_Akt:1'     PDK1(PH!_,S241~p), Akt(PH!_,T308~u) -> PDK1(PH!_,S241~p!1), Akt(PH!_,T308~u!1)
'Akt@308'        PDK1(PH!_,S241~p!1), Akt(PH!_,T308~u!1) -> PDK1(PH!_,S241~p!1), Akt(PH!_,T308~
'PDK1_Akt_op:1'  PDK1(S241~p!1), Akt(T308!1) -> PDK1(S241~p), Akt(T308)
'PDK1_Akt:2'     PDK1(PH!_,S241~p), Akt(PH!_,S473~u) -> PDK1(PH!_,S241~p!1), Akt(PH!_,S473~u!1)
'Akt@473'        PDK1(PH!_,S241~p!1), Akt(PH!_,S473~u!1) -> PDK1(PH!_,S241~p!1), Akt(PH!_,S473~
'PDK1_Akt_op:2'  PDK1(S241~p!1), Akt(S473!1) -> PDK1(S241~p), Akt(S473)
'PP2A3_Akt:1'    PP2A3(s), Akt(PH,S473~p) -> PP2A3(s!1), Akt(PH,S473~p!1)
'Akt@473_op'     PP2A3(s!1), Akt(S473~p!1) -> PP2A3(s!1), Akt(S473~u!1)
'PP2A3_Akt_op:1' PP2A3(s!1), Akt(S473!1) -> PP2A3(s), Akt(S473)
'PP2A3_Akt:2'    PP2A3(s), Akt(PH,T308~p) -> PP2A3(s!1), Akt(PH,T308~p!1)
'Akt@308_op'     PP2A3(s!1), Akt(T308~p!1) -> PP2A3(s!1), Akt(T308~u!1)
'PP2A3_Akt_op:2' PP2A3(s!1), Akt(T308!1) -> PP2A3(s), Akt(T308)
'Ras_Raf'      Ras(S1S2~gtp), Raf(x~u) <-> Ras(S1S2~gtp!1), Raf(x~u!1)
'Raf'          Ras(S1S2~gtp!1), Raf(x~u!1) -> Ras(S1S2~gtp!1), Raf(x~p!1)
'Ras_Raf_op'   Ras(S1S2~gtp!1), Raf(x!1) -> Ras(S1S2~gtp), Raf(x)
'PP2A1_Raf'    PP2A1(s), Raf(x~p) <-> PP2A1(s!1), Raf(x~p!1)
'Raf_op'       PP2A1(s!1), Raf(x~p!1) -> PP2A1(s!1), Raf(x~u!1)
'PP2A1_Raf_op' PP2A1(s!1), Raf(x!1) -> PP2A1(s), Raf(x)
'Raf_MEK@222'      Raf(x~p), MEK(S222~u) <-> Raf(x~p!1), MEK(S222~u!1)
'MEK@222'          Raf(x~p!1), MEK(S222~u!1) -> Raf(x~p!1), MEK(S222~p!1)
'Raf_MEK@222_op'   Raf(x~p!1), MEK(S222!1) -> Raf(x~p), MEK(S222)
'Raf_MEK@218'      Raf(x~p), MEK(S218~u) <-> Raf(x~p!1), MEK(S218~u!1)
'MEK@218'          Raf(x~p!1), MEK(S218~u!1) -> Raf(x~p!1), MEK(S218~p!1)
'Raf_MEK@218_op'   Raf(x~p!1), MEK(S218!1) -> Raf(x~p), MEK(S218)
'PP2A2_MEK@222'    PP2A2(s), MEK(S222~p) <-> PP2A2(s!1), MEK(S222~p!1)
'MEK@222_op'       PP2A2(s!1), MEK(S222~p!1) -> PP2A2(s!1), MEK(S222~u!1)
'PP2A2_MEK@222_op' PP2A2(s!1), MEK(S222!1) -> PP2A2(s), MEK(S222)
'PP2A2_MEK@218'    PP2A2(s), MEK(S218~p) <-> PP2A2(s!1), MEK(S218~p!1)
'MEK@222_op'       PP2A2(s!1), MEK(S218~p!1) -> PP2A2(s!1), MEK(S218~u!1)
'PP2A2_MEK@218_op' PP2A2(s!1), MEK(S218!1) -> PP2A2(s), MEK(S218)
'MEK_ERK@185'    MEK(s,S218~p,S222~p), ERK(T185~u) <-> MEK(s!1,S218~p,S222~p), ERK(T185~u!1)
'ERK@185'        MEK(s!1,S218~p,S222~p), ERK(T185~u!1) -> MEK(s!1,S218~p,S222~p), ERK(T185~p!1)
'MEK_ERK@185_op' MEK(s!1,S218~p,S222~p), ERK(T185!1) -> MEK(s,S218~p,S222~p), ERK(T185)
'MEK_ERK@187'    MEK(s,S218~p,S222~p), ERK(Y187~u) <-> MEK(s!1,S218~p,S222~p), ERK(Y187~u!1)
'ERK@187'        MEK(s!1,S218~p,S222~p), ERK(Y187~u!1) -> MEK(s!1,S218~p,S222~p), ERK(Y187~p!1)
'MEK_ERK@187_op' MEK(s!1,S218~p,S222~p), ERK(Y187!1) -> MEK(s,S218~p,S222~p), ERK(Y187)
'MKP_ERK@185'    MKP3(s), ERK(T185~p) <-> MKP3(s!1), ERK(T185~p!1)
'ERK@185_op'     MKP3(s!1), ERK(T185~p!1) -> MKP3(s!1), ERK(T185~u!1)
'MKP_ERK@185_op' MKP3(s!1), ERK(T185!1) -> MKP3(s), ERK(T185)
'MKP_ERK@187'    MKP3(s), ERK(Y187~p) <-> MKP3(s!1), ERK(Y187~p!1)
'ERK@187_op'     MKP3(s!1), ERK(Y187~p!1) -> MKP3(s!1), ERK(Y187~u!1)
'MKP_ERK@187_op' MKP3(s!1), ERK(Y187!1) -> MKP3(s), ERK(Y187)
'SoS_ERK'         SoS(SS), ERK(CD,T185~p,Y187~p) <-> SoS(SS!1), ERK(CD!1,T185~p,Y187~p)
'SoS'             SoS(SS~u!1), ERK(CD!1,T185~p,Y187~p) -> SoS(SS~p!1), ERK(CD!1,T185~p,Y187~p)
'ERK_SoS_Grb2_op' Grb2(SH3n!1), SoS(a!1,SS~p!2), ERK(CD!2,T185~p,Y187~p) -> Grb2(SH3n), SoS(a,S
```

EGFR with Akt pathway

110 rules

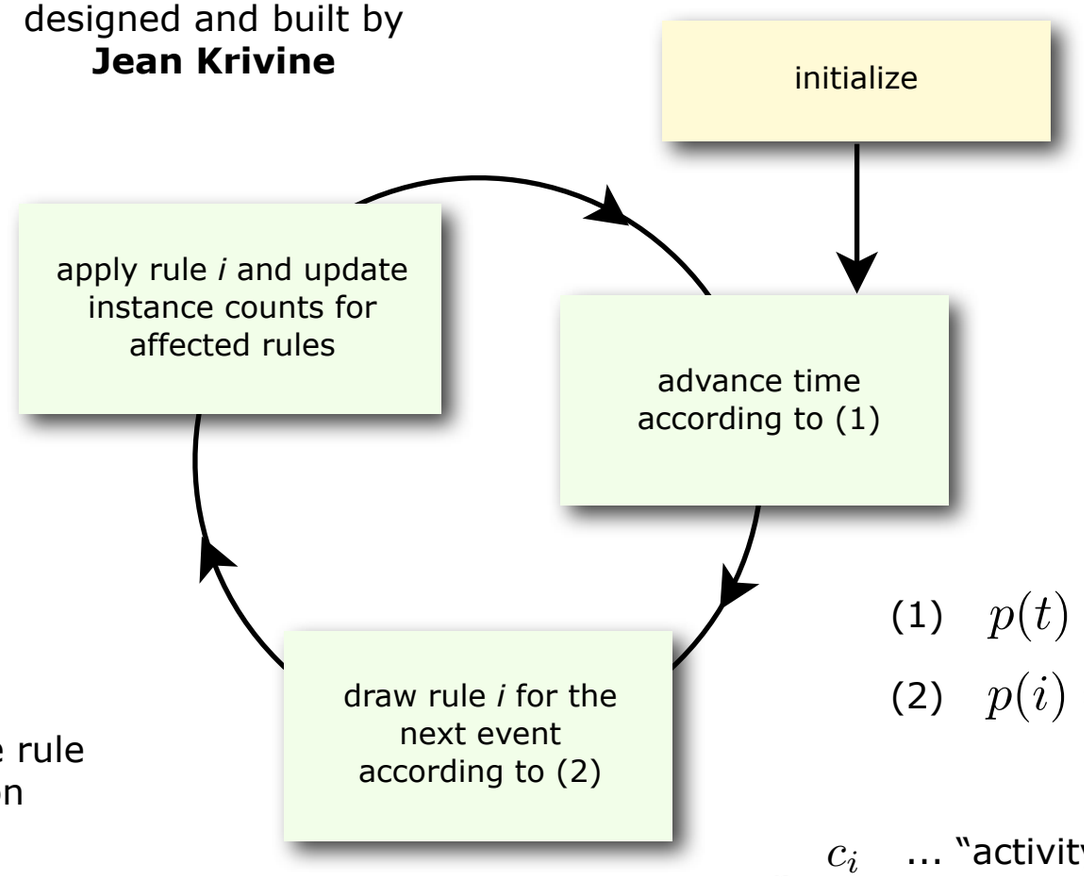
The simulator (using a generalization of Gillespie's scheme)

A fully scalable rule-based stochastic (CTMC) simulation engine.

The computational cost per update cycle is

- independent of the number of agents in the system (actual system size)
- independent of the number of possible agent combinations implied by rules (potential system size)
- bounded by the max degree of rule activation map

designed and built by
Jean Krivine



$$(1)\quad p(t) = \lambda e^{-\lambda t}$$

$$(2)\quad p(i) = \frac{c_i}{\lambda}$$

c_i ... "activity" of rule *i*

$\lambda = \sum_{i}^{n} c_i$ ... "system activity"

- At no point is a set of possible complexes generated. (No flattening ever.)
- Space complexity is linear rather than exponential in the rule arity; generates "clashes" that require a clever simulation time update (solved by **Vincent Danos**).

The only dominant time complexity in practice is log |R|, where R is the set of rules.

V.Danos, J.Feret, W.F, R.Harmer, and J.Krivine. Rule-based Modelling of Cellular Signalling. In: L. Caires and V.T.Vasconcelos (Eds.): Concurrency Theory 2007, *Lecture Notes in Computer Science*, **4703**:17-41 (2007)

Like ODEs, a set of rules is a formal object that can be analyzed mathematically...

$$\frac{d}{dt}x_1 = -k_1 \cdot x_2 \cdot x_1 + k_{-1} \cdot x_3$$
$$\frac{d}{dt}x_2 = -k_1 \cdot x_2 \cdot x_1 + k_{-1} \cdot x_3$$
$$\frac{d}{dt}x_3 = k_1 \cdot x_2 \cdot x_1 - k_{-1} \cdot x_3 - 2 \cdot (k_2 \cdot x_3 \cdot x_3 - k_{-2} \cdot x_4)$$
$$\frac{d}{dt}x_4 = k_2 \cdot x_3 \cdot x_3 - k_{-2} \cdot x_4 + \frac{v_4 \cdot x_5}{p_4 + x_5} - (k_3 \cdot x_4 - k_{-3} \cdot x_5)$$
$$\frac{d}{dt}x_5 = k_3 \cdot x_4 - k_{-3} \cdot x_5 + k_7 \cdot x_7 - k_{-7} \cdot x_5 \cdot x_{13} + k_{11} \cdot x_9 -$$
$$- k_{-11} \cdot x_5 \cdot x_9 + k_{15} \cdot x_{11} - k_{-15} \cdot x_5 \cdot x_{15} + k_{18} \cdot x_{12} -$$
$$- k_{-18} \cdot x_5 \cdot x_{16} + k_{20} \cdot x_{13} - k_{-20} \cdot x_{17} \cdot x_5 - (\frac{v_4 \cdot x_5}{p_4 + x_5} +$$
$$+ k_9 \cdot x_5 \cdot x_{15} - k_{-9} \cdot x_8 + k_5 \cdot x_5 \cdot x_{18} - k_{-5} \cdot x_6)$$
$$\frac{d}{dt}x_6 = \cdots$$
$$\vdots$$
$$\frac{d}{dt}x_n = \cdots$$

468,561 equations

76 rules

- dynamical systems theory
  - ✦ steady states, oscillations
  - ✦ bifurcations

static analysis

- “maps”
  - ✦ relationships between rules
  - ✦ relationships between agents
- concurrency theory
  - ✦ event structures (formal pathways)

- numerical integration

- stochastic simulation

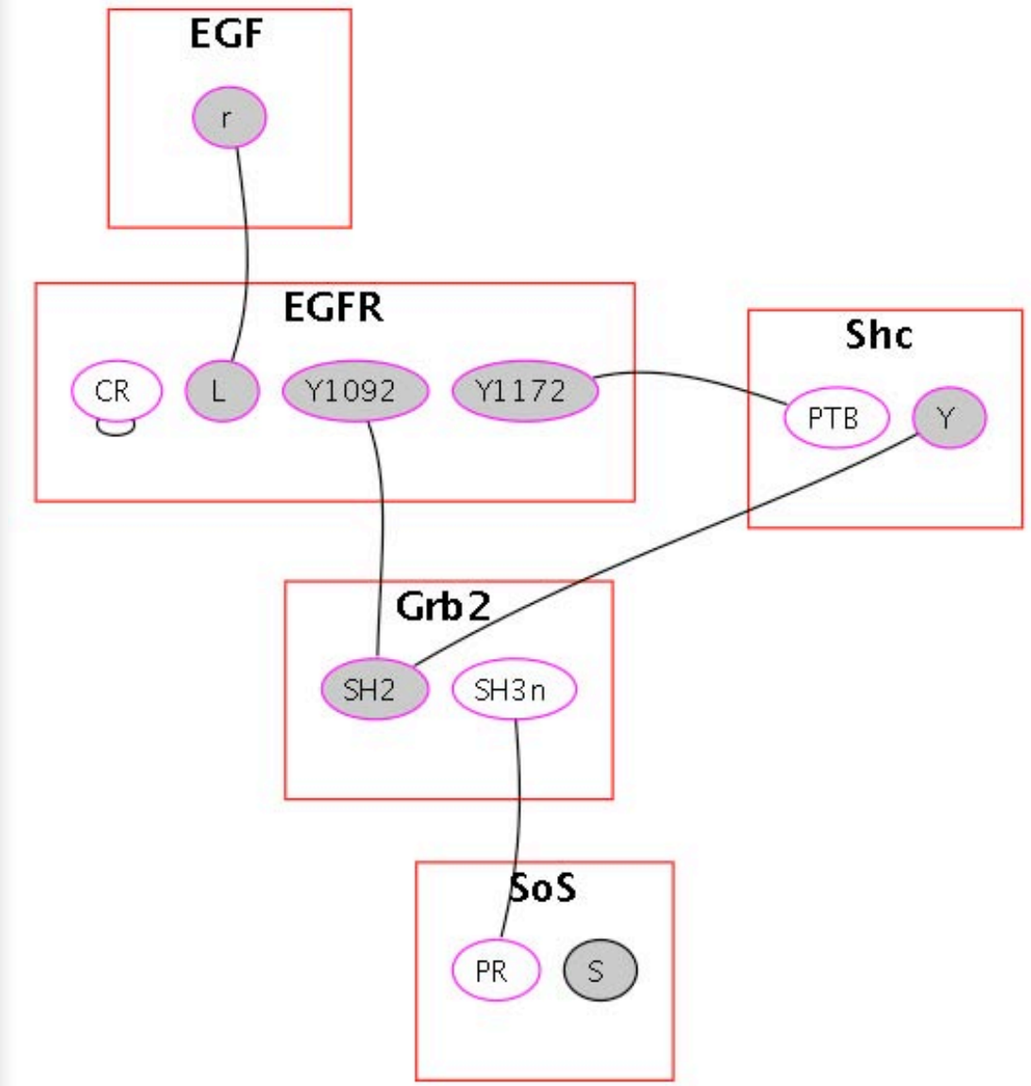
the contact map and the set of reachable species

```
'EGFR_EGFRe' EGFR(L~ext!_,CR),EGFR(L~ext!_,CR) <-> EGFR(L~ext!_,CR!1),EGFR(L~ext!_,CR!1)
'EGF_EGFRe'  EGF(r~ext),EGFR(L~ext,CR) <-> EGF(r~ext!1),EGFR(L~ext!1,CR)
'Shc_Grb2'   Shc(Y~p),Grb2(SH2) <-> Shc(Y~p!1),Grb2(SH2!1)
'EGFR_Grb2'  EGFR(Y1092~p),Grb2(SH2) <-> EGFR(Y1092~p!1),Grb2(SH2!1)
'EGFR_Shc'   EGFR(Y1172~p),Shc(PTB) <-> EGFR(Y1172~p!1),Shc(PTB!1)
'Grb2_SoS'   Grb2(SH3n),SoS(PR,S~u) -> Grb2(SH3n!1),SoS(PR!1,S~u)
'Grb2_SoS_op' Grb2(SH3n!1),SoS(PR!1) -> Grb2(SH3n),SoS(PR)
'auto Y1092'  EGFR(CR!1),EGFR(Y1092~u,CR!1) -> EGFR(CR!1),EGFR(Y1092~p,CR!1)
'auto Y1172'  EGFR(CR!1),EGFR(Y1172~u,CR!1) -> EGFR(CR!1),EGFR(Y1172~p,CR!1)
'depho Y1092' EGFR(Y1092~p) -> EGFR(Y1092~u)
'depho Y1172' EGFR(Y1172~p) -> EGFR(Y1172~u)
'Shc@Y'      Shc(PTB!_,Y~u) -> Shc(PTB!_,Y~p)
'Shc_op'     Shc(Y~p) -> Shc(Y~u)
```

these rules generate 356 possible molecular species

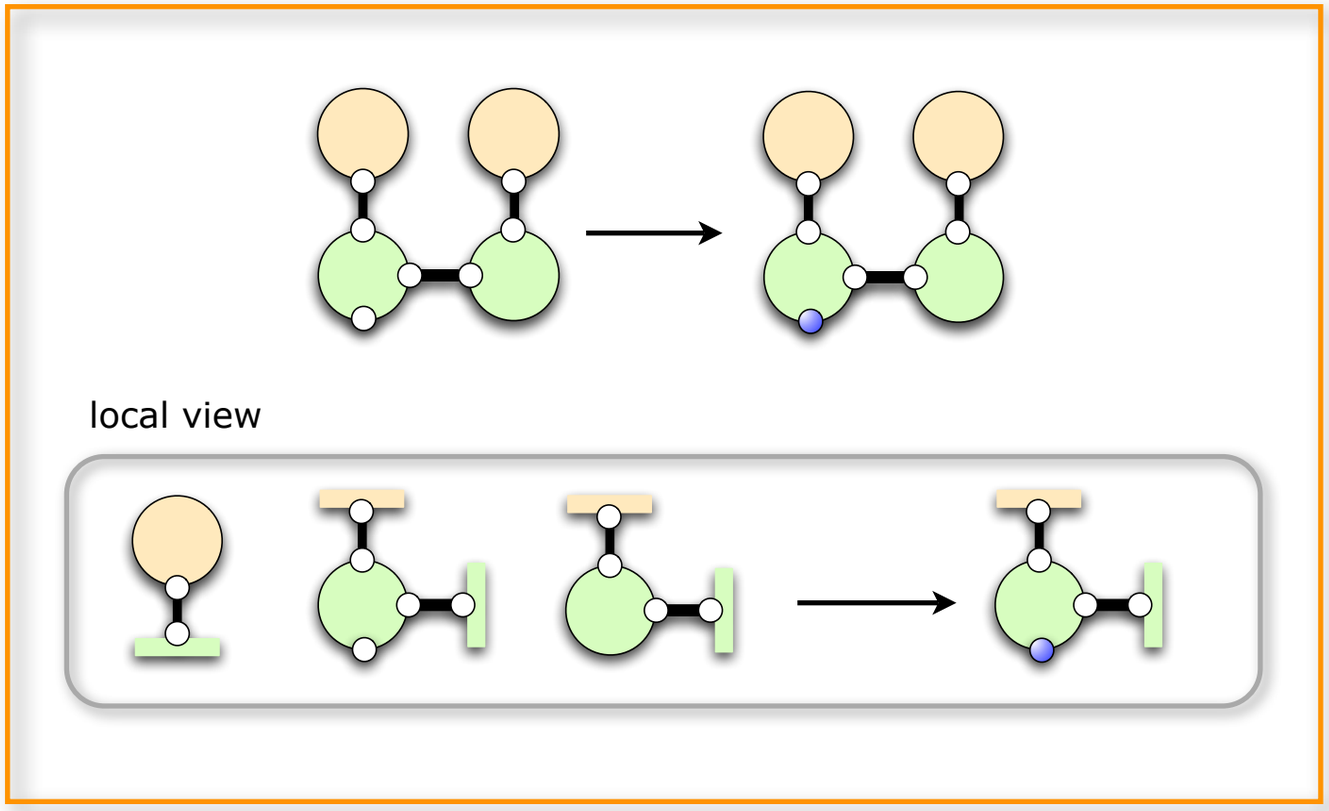
⋮

- EGFR(CR,L~ext,Y1092~p,Y1172~p!1),Grb2(SH2~u!2,SH3n),Shc(PTB!1,Y~p!2)
- EGFR(CR,L~ext,Y1092~p!1,Y1172~p!2),Grb2(SH2~u!1,SH3n),Shc(PTB!2,Y~p)
- EGFR(CR,L~ext,Y1092~p!1,Y1172~p!2),Grb2(SH2~u!1,SH3n),Shc(PTB!2,Y~u)
- EGFR(CR,L~ext,Y1092~u,Y1172~p!1),Grb2(SH2~u!2,SH3n),Shc(PTB!1,Y~p!2)
- EGFR(CR,L~ext,Y1092~p,Y1172~p!1),Grb2(SH2~u!2,SH3n!3),Shc(PTB!1,Y~p!2),SoS(PR!3,S~u)
- EGFR(CR,L~ext,Y1092~p!1,Y1172~p!3),Grb2(SH2~u!1,SH3n!2),Shc(PTB!3,Y~p),SoS(PR!2,S~u)
- EGFR(CR,L~ext,Y1092~p!1,Y1172~p!3),Grb2(SH2~u!1,SH3n!2),Shc(PTB!3,Y~u),SoS(PR!2,S~u)
- EGFR(CR,L~ext,Y1092~u,Y1172~p!1),Grb2(SH2~u!2,SH3n!3),Shc(PTB!1,Y~p!2),SoS(PR!3,S~u)
- EGFR(CR,L~ext,Y1092~p!1,Y1172~p),Grb2(SH2~u!1,SH3n!2),SoS(PR!2,S~u)
- EGFR(CR,L~ext,Y1092~p!1,Y1172~u),Grb2(SH2~u!1,SH3n!2),SoS(PR!2,S~u)
- EGFR(CR,L~ext,Y1092~p,Y1172~p!1),Shc(PTB!1,Y~p)
- EGFR(CR,L~ext,Y1092~p,Y1172~p!1),Shc(PTB!1,Y~u)
- EGFR(CR,L~ext,Y1092~u,Y1172~p!1),Shc(PTB!1,Y~p)
- EGFR(CR,L~ext,Y1092~u,Y1172~p!1),Shc(PTB!1,Y~u)
- Grb2(SH2~u,SH3n)
- Grb2(SH2~u!1,SH3n),Shc(PTB,Y~p!1)
- Grb2(SH2~u!1,SH3n!2),Shc(PTB,Y~p!1),SoS(PR!2,S~u)
- Grb2(SH2~u,SH3n!1),SoS(PR!1,S~u)
- Shc(PTB,Y~p)
- Shc(PTB,Y~u)
- SoS(PR,S~u)



Using abstract interpretation to...

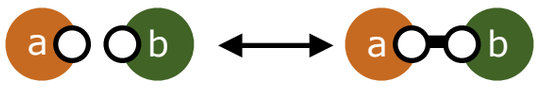
- compute reachable molecular species (states)
- detect dead rules (useful when developing large models)
- coarsen rules (eliminate superfluous conditions)
- determine whether a rule may activate another
- generate the underlying "flat" system



V.Danos, **J.Feret**, W.F., and J.Krivine. Abstract interpretation of cellular signalling networks. *VMCAI 2008*. To appear.

Cousot, P., Cousot, R. Abstract interpretation and application to logic programs. *Journal of Logic Programming,* 13, 103–179 (1992)

causality and the concept of story: example

ab, ab_op 

$$a(x), b(x) \longleftrightarrow a(x!1), b(x!1)$$

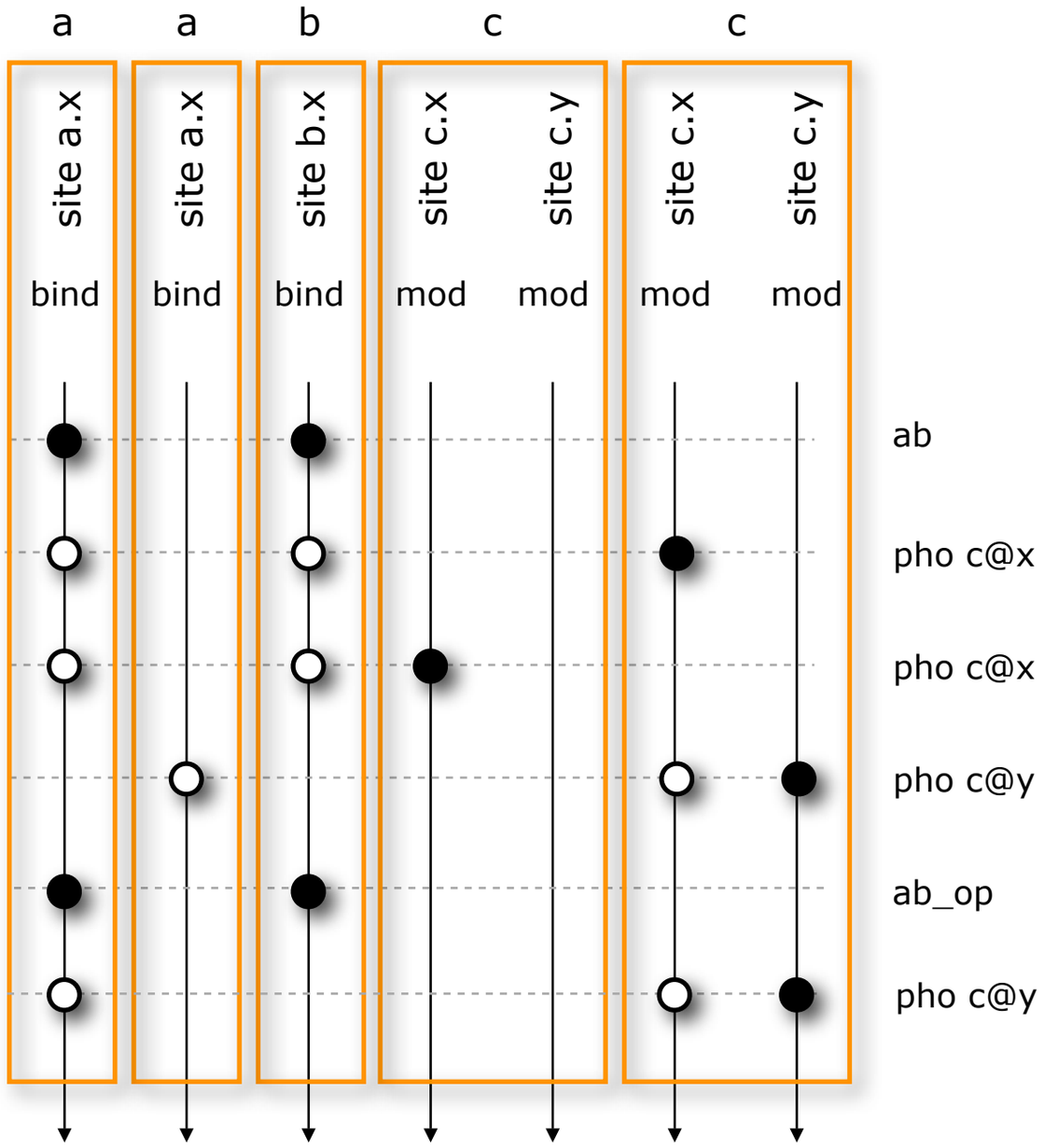
pho c@x 

$$a(x!1), b(x!1), c(x \sim u) \longrightarrow a(x!1), b(x!1), c(x \sim p)$$

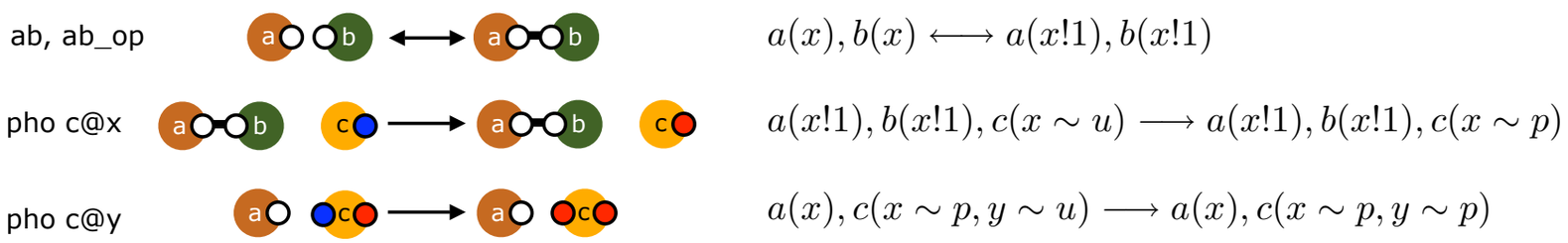
pho c@y 

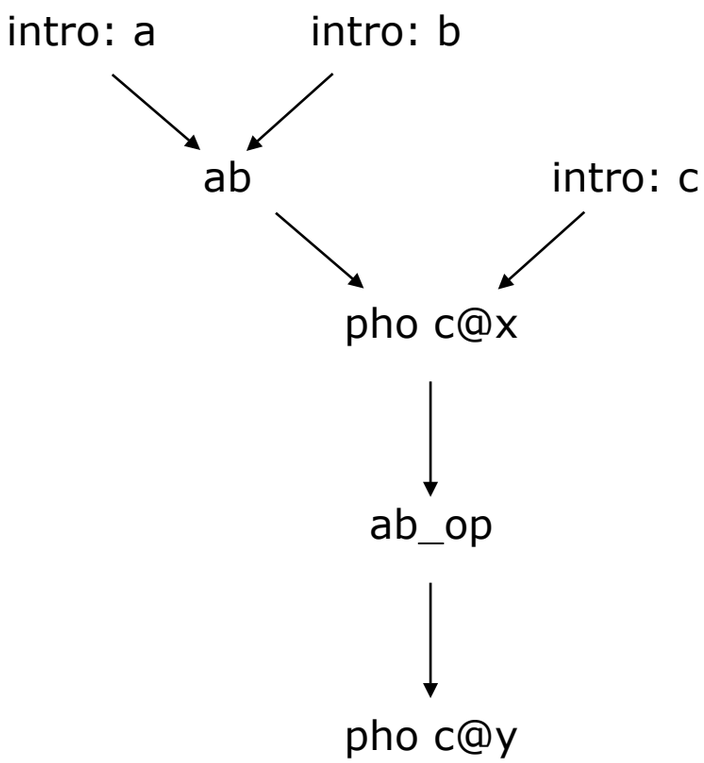
$$a(x), c(x \sim p, y \sim u) \longrightarrow a(x), c(x \sim p, y \sim p)$$

How does agent c become doubly phosphorylated?


a
a
b
c
c
site a.x
site a.x
site b.x
site c.x
site c.y
site c.x
site c.y
bind
bind
bind
mod
mod
mod
mod
ab
pho c@x
pho c@x
pho c@y
ab_op
pho c@y

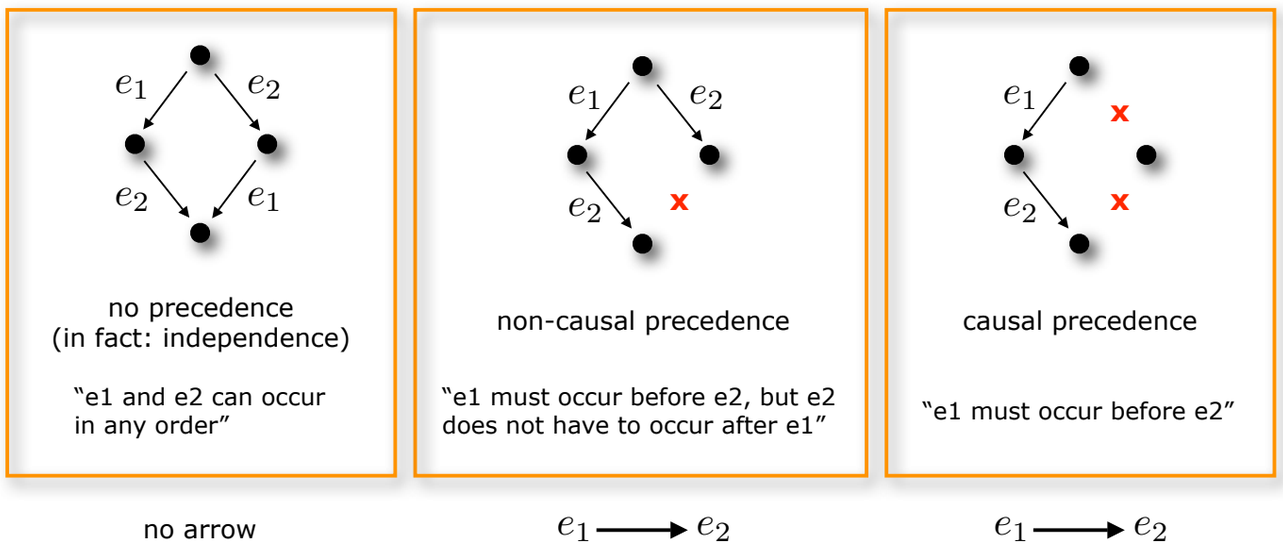
causality and the concept of "story"

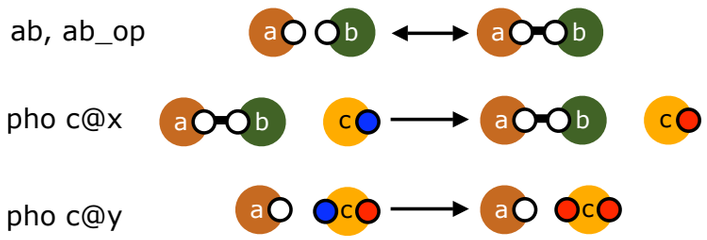




An arrow is a relation of precedence. Often, but not always, precedence reveals causal dependence. In addition, informative precedence is "one-step" precedence.

If there exists a (reachable) configuration such that...

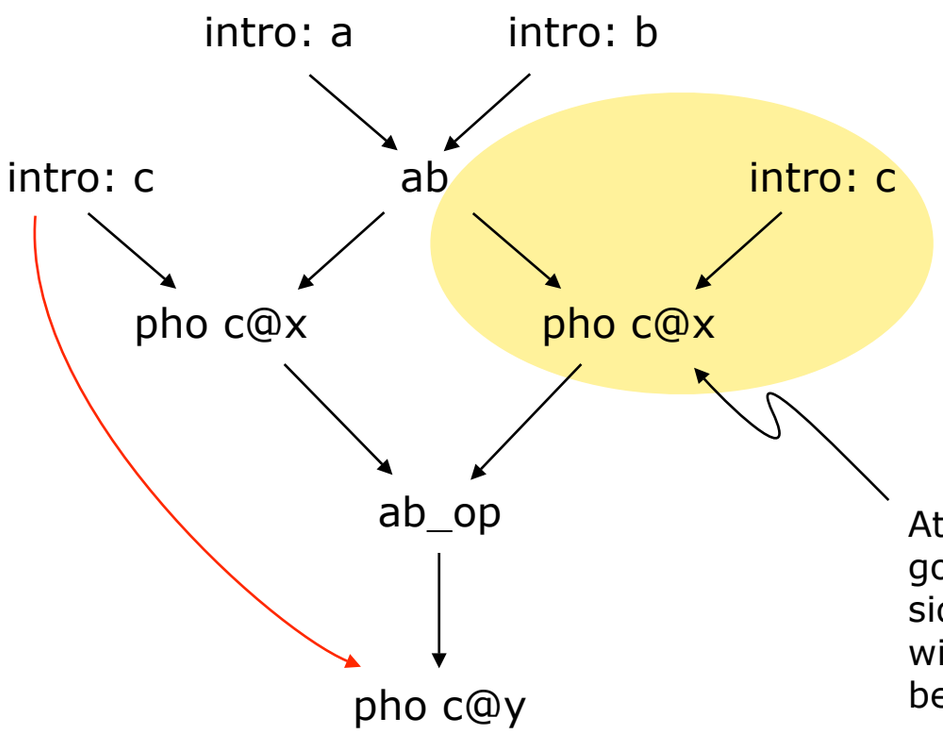




$$a(x), b(x) \longleftrightarrow a(x!1), b(x!1)$$

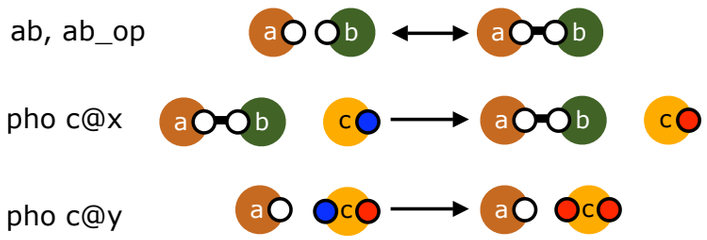
$$a(x!1), b(x!1), c(x \sim u) \longrightarrow a(x!1), b(x!1), c(x \sim p)$$

$$a(x), c(x \sim p, y \sim u) \longrightarrow a(x), c(x \sim p, y \sim p)$$



There is no ambiguity about which c is part of the observable because we always know the identity of all agents. The ambiguity is only an artifact of substituting events with the names of rules that gave rise to them.

At this point, we have made no progress (seen retrospectively) towards the goal "pho c@y". From the goal's perspective, this is a loop with an irrelevant side-effect. Removing such loops is called "**weak compression**". (2-loops with no side-effects, such as successive bind-unbind cycles have already been removed.)



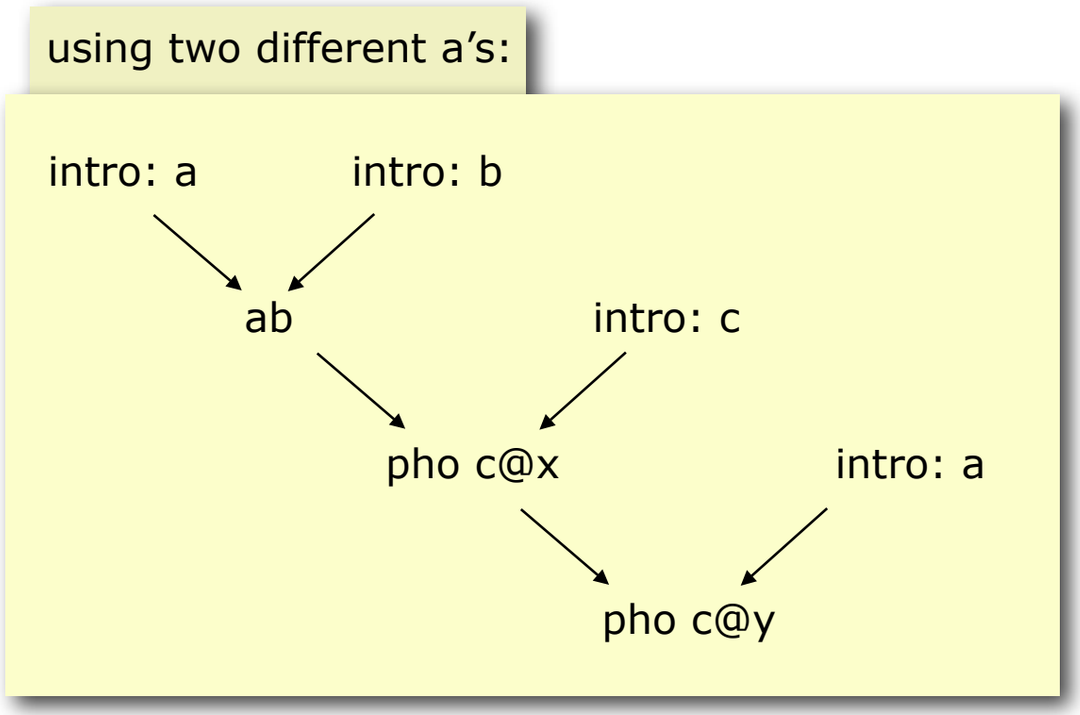
$$a(x), b(x) \longleftrightarrow a(x!1), b(x!1)$$

$$a(x!1), b(x!1), c(x \sim u) \longrightarrow a(x!1), b(x!1), c(x \sim p)$$

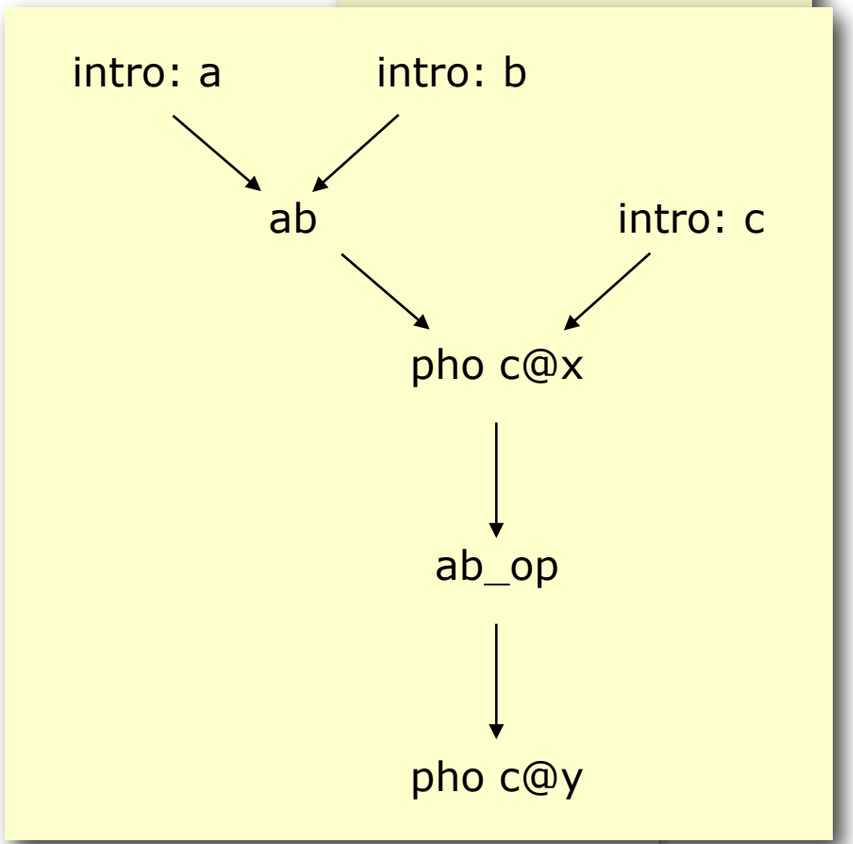
$$a(x), c(x \sim p, y \sim u) \longrightarrow a(x), c(x \sim p, y \sim p)$$

In sum, there are two basic ways - stories - of double-phoing c:

using two different a's:



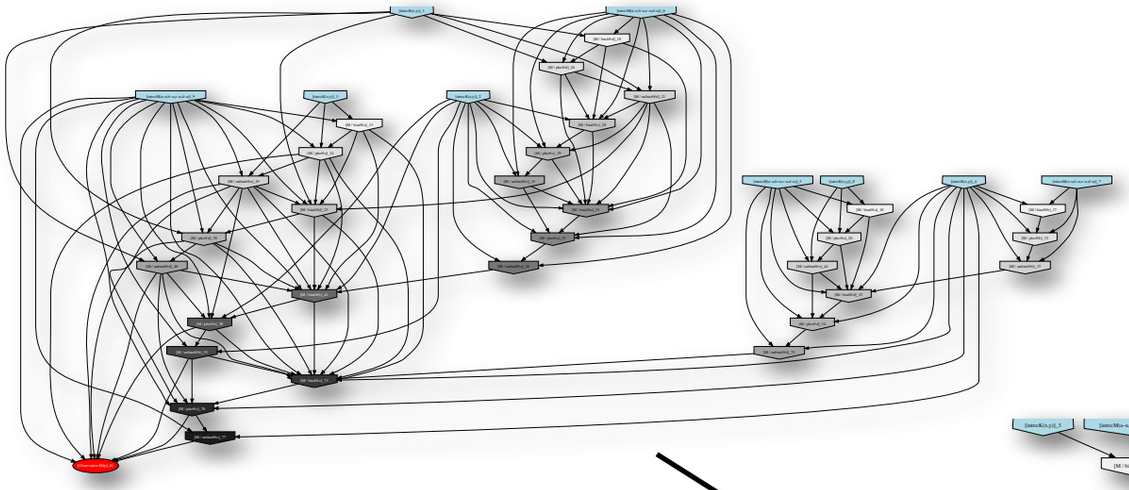
re-using the same a:



Intuitively, a **story** is:

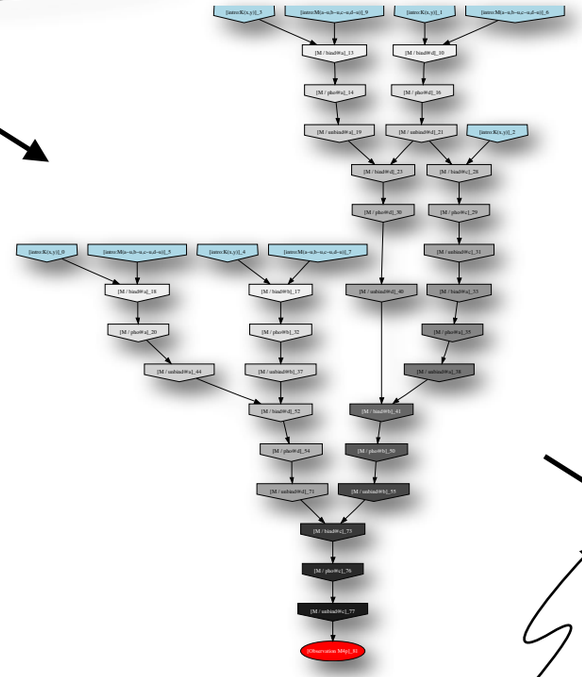
- a **causal lineage** of an observable in logical time
- a **mechanism of action**
- a **pathway**
- a **distributed computation** (an equivalence class of executions or traces)

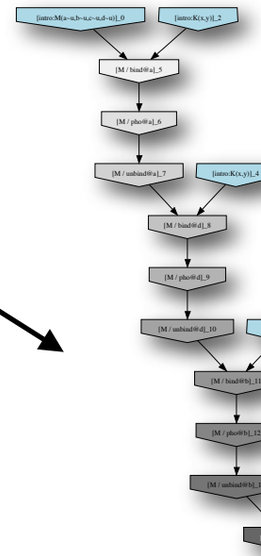
story compression: peeling the onion

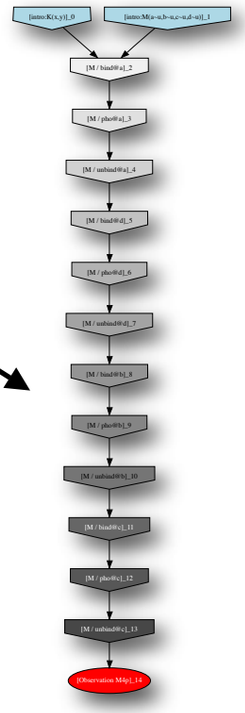


case: a kinase phosphorylates (distributively) a substrate at 4 sites

```
'unbind@a' M(a!1),K(x!1) -> M(a),K(x)
'unbind@b' M(b!1),K(x!1) -> M(b),K(x)
'unbind@c' M(c!1),K(x!1) -> M(c),K(x)
'unbind@d' M(d!1),K(x!1) -> M(d),K(x)
'bind@a' M(a~u,b,c,d),K(x) -> M(a~u!1,b,c,d),K(x!1)
'bind@b' M(a,b~u,c,d),K(x) -> M(a,b~u!1,c,d),K(x!1)
'bind@c' M(a,b,c~u,d),K(x) -> M(a,b,c~u!1,d),K(x!1)
'bind@d' M(a,b,c,d~u),K(x) -> M(a,b,c,d~u!1),K(x!1)
'pho@a' M(a~u!1,b,c,d),K(x!1) -> M(a~p!1,b,c,d),K(x!1)
'pho@b' M(a,c,d,b~u!1),K(x!1) -> M(a,c,d,b~p!1),K(x!1)
'pho@c' M(a,d,b,c~u!1),K(x!1) -> M(a,d,b,c~p!1),K(x!1)
'pho@d' M(a,b,c,d~u!1),K(x!1) -> M(a,b,c,d~p!1),K(x!1)
```





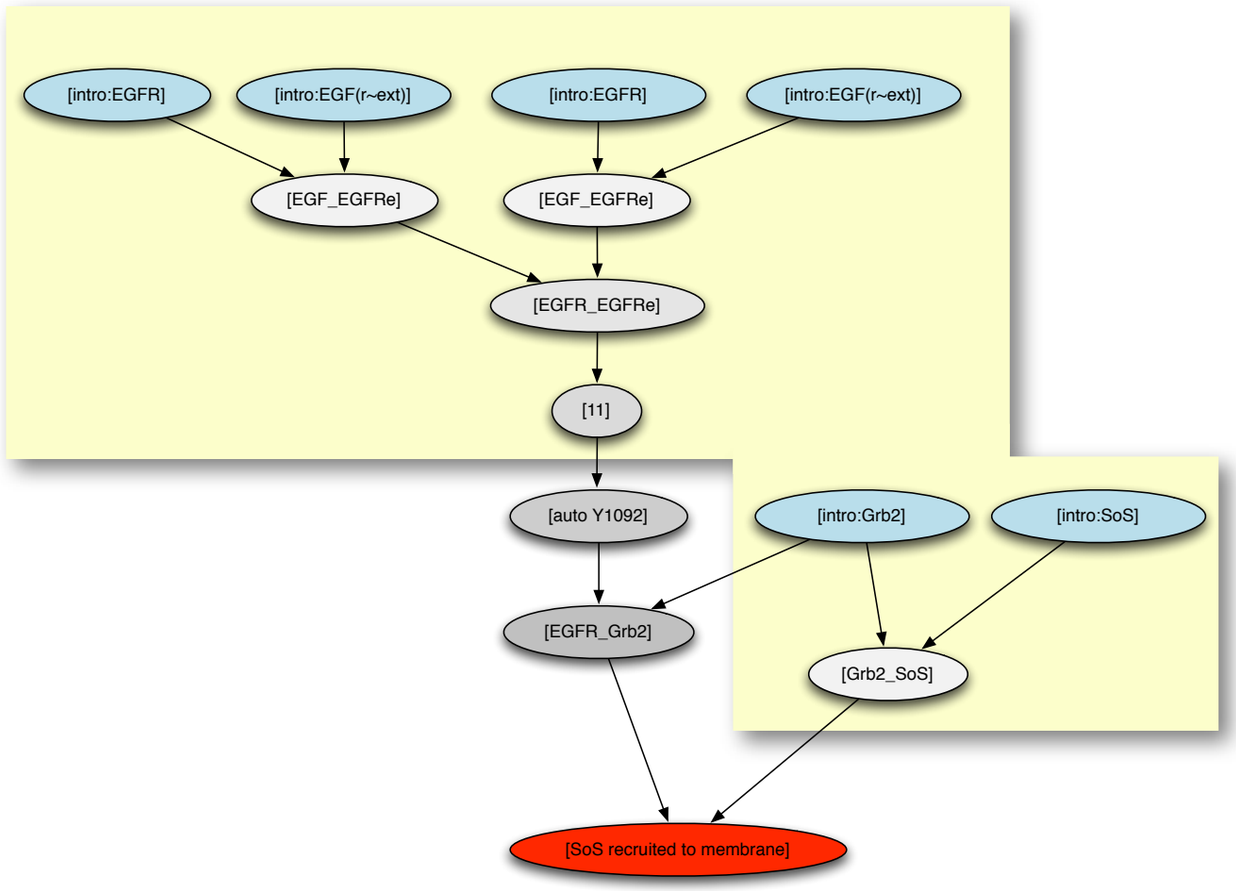


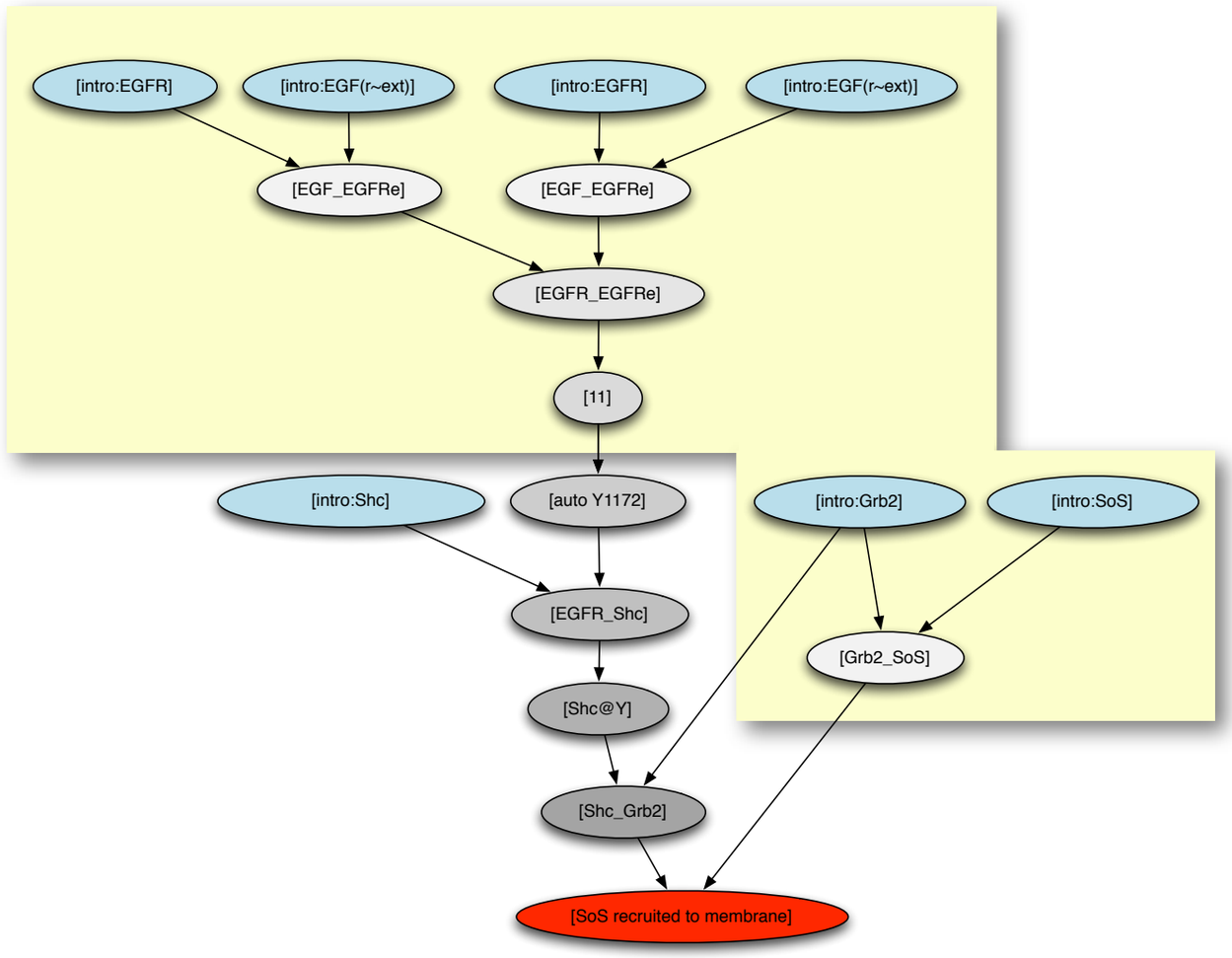
unclutter: reduced precedence relation (and removal of existence tests)

weak compression: remove loops with irrelevant side-effects

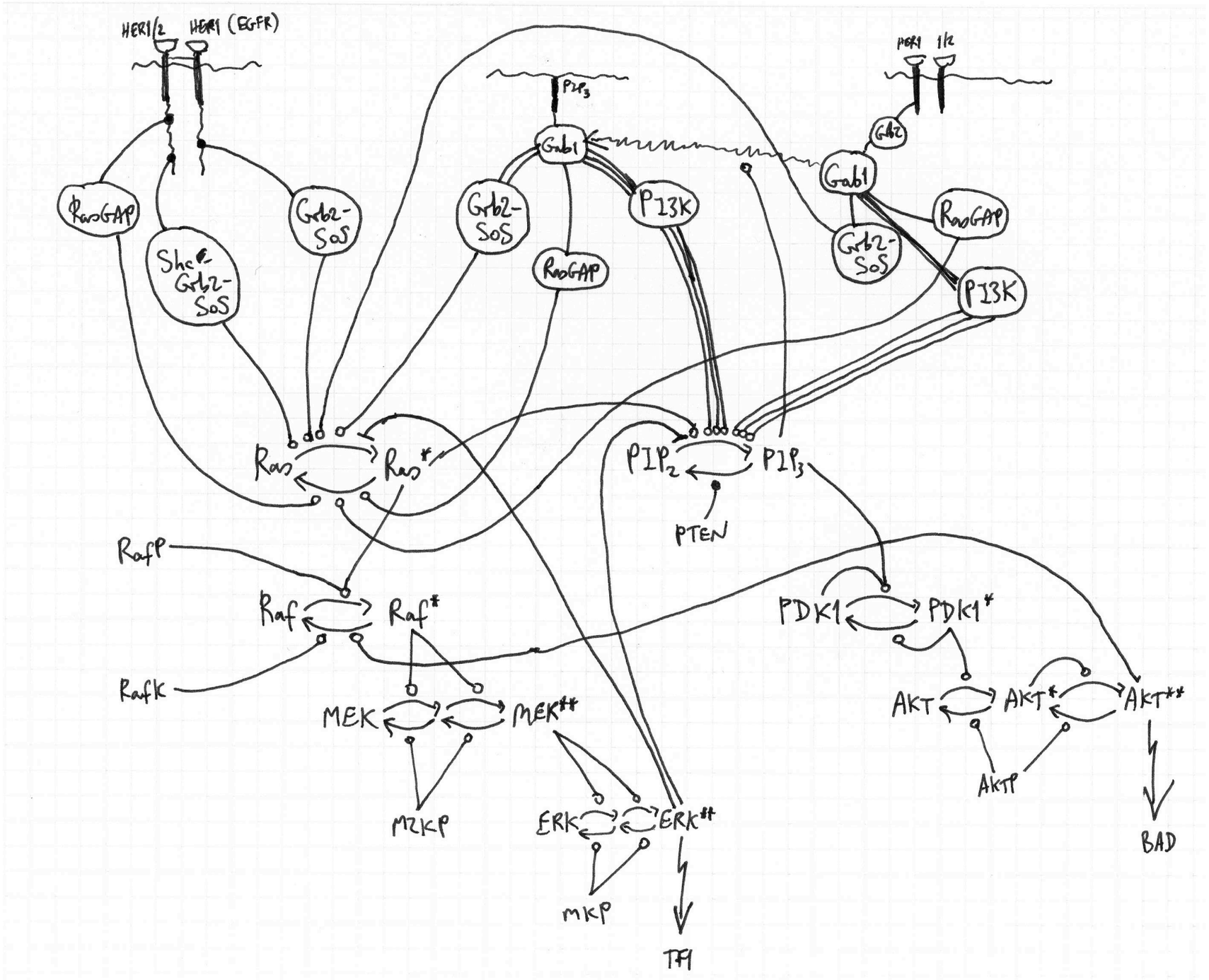
strong compression: apply agent automorphisms

necessary events to an observable

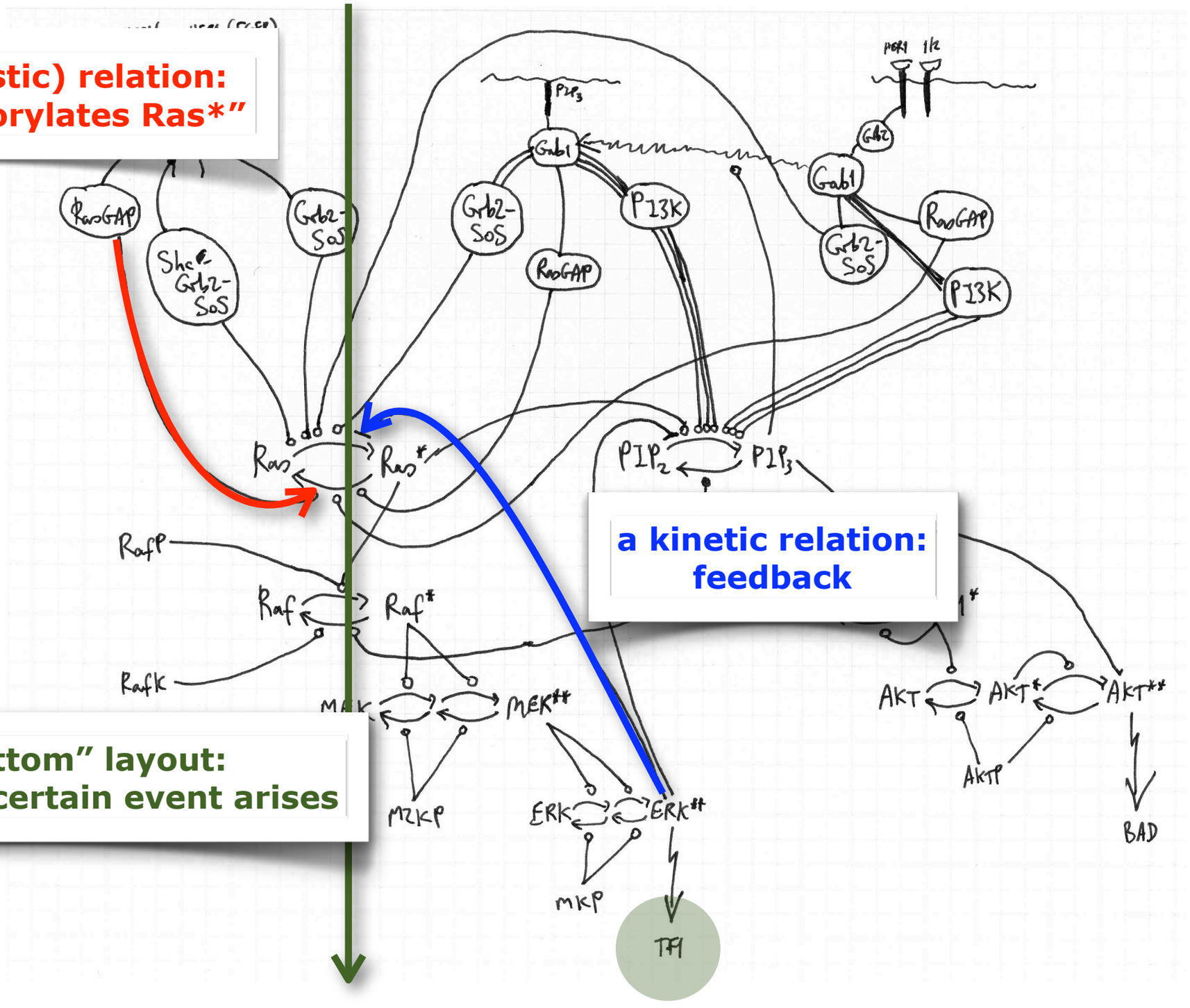




conceptual thumbnails on laboratory whiteboards



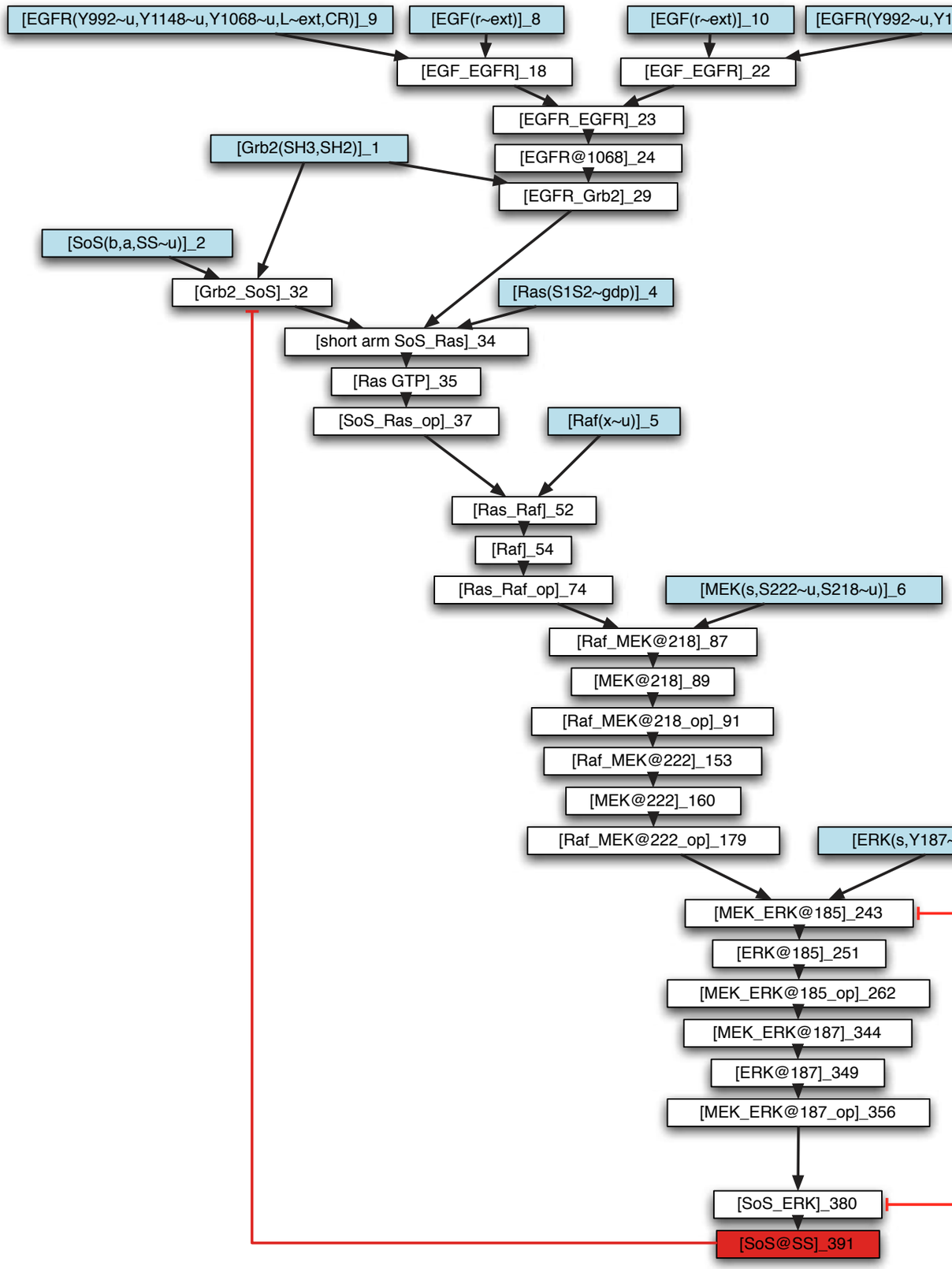
a causal (mechanistic) relation:
"RasGAP dephosphorylates Ras*"

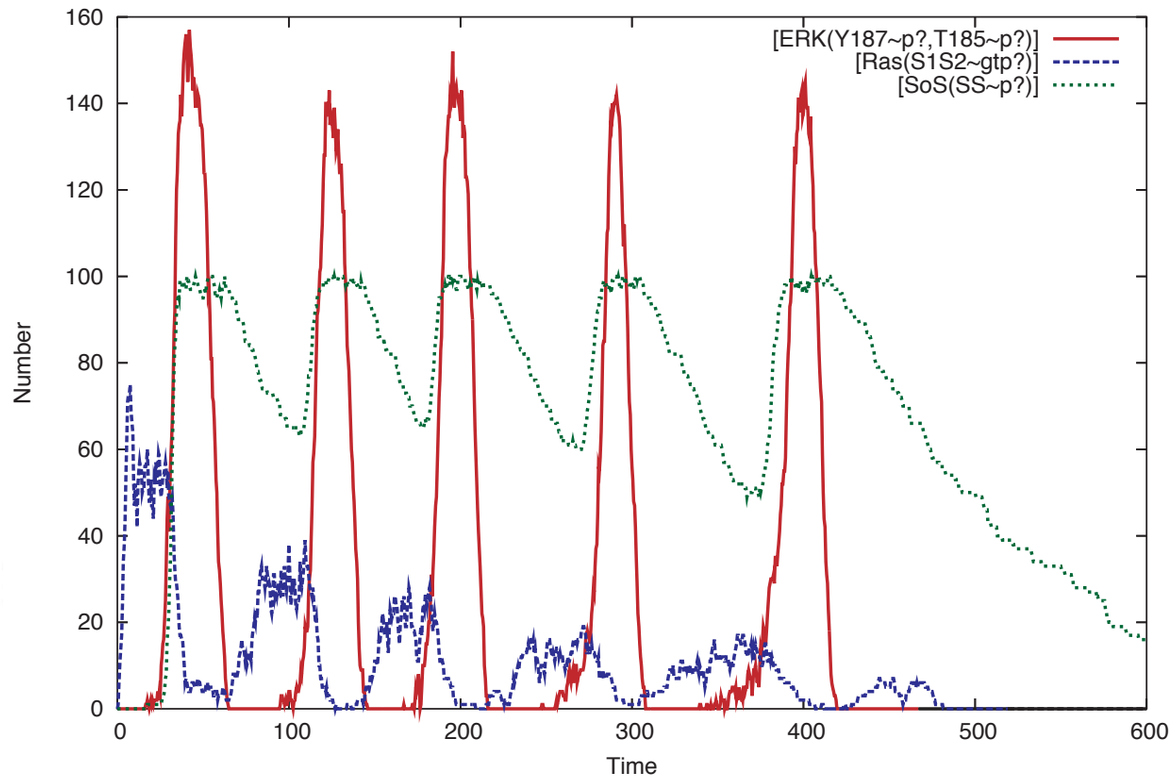


a kinetic relation:
feedback

"top-to-bottom" layout:
a story of how a certain event arises

superimposing precedence and causality/conflict within the same story - feedback example





models are communication devices

systems biology models *are* programs

"organization" refers to the causal structure of a concurrent system

pathways are special kinds of event structures

combinatorial complexity may be one source of variability *within* individuals

This is the beginning, not the end.

The foundations of systems biology need computer science!