

Rule-based modeling

Part II

Computational Biology 1 & 2

FDV 2009-2010

Programs as models

A language for the cell
(or how to express
facts)

Syntax

Software engineering a
model
(or how to trust a
model)

Compilation

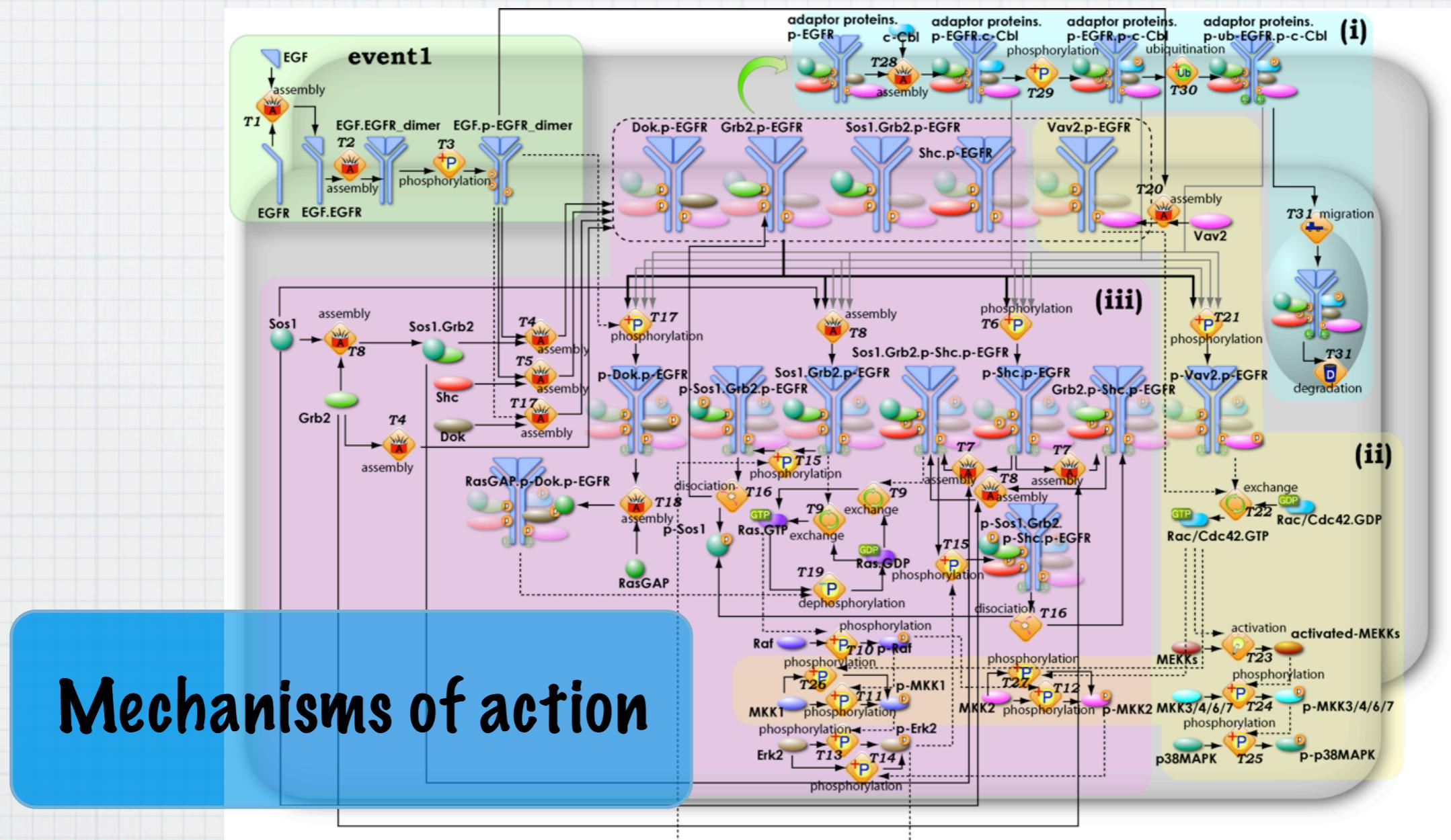
A model

Program

Motivations

Quantification vs. representation

Trying to combine:

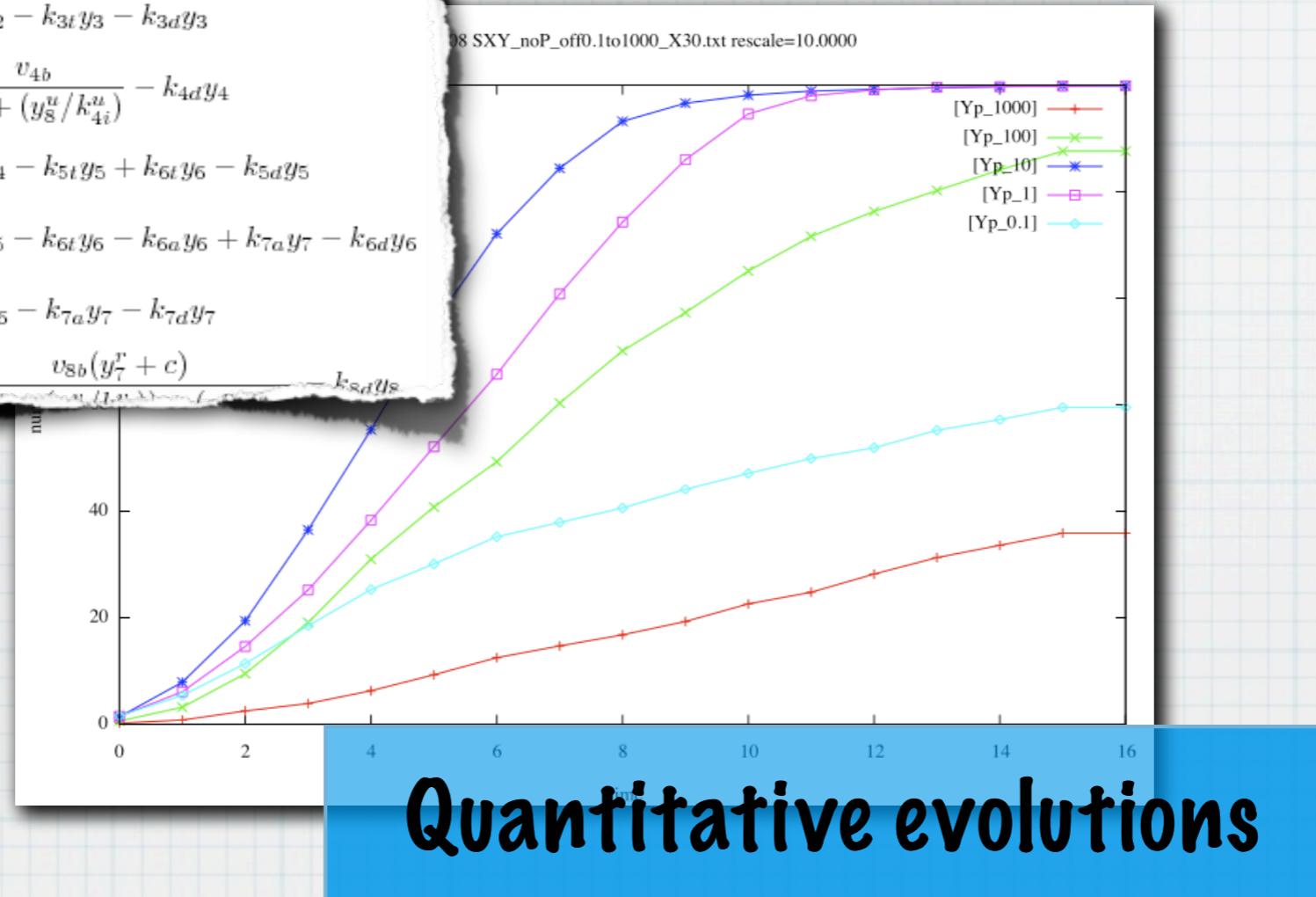


Motivations

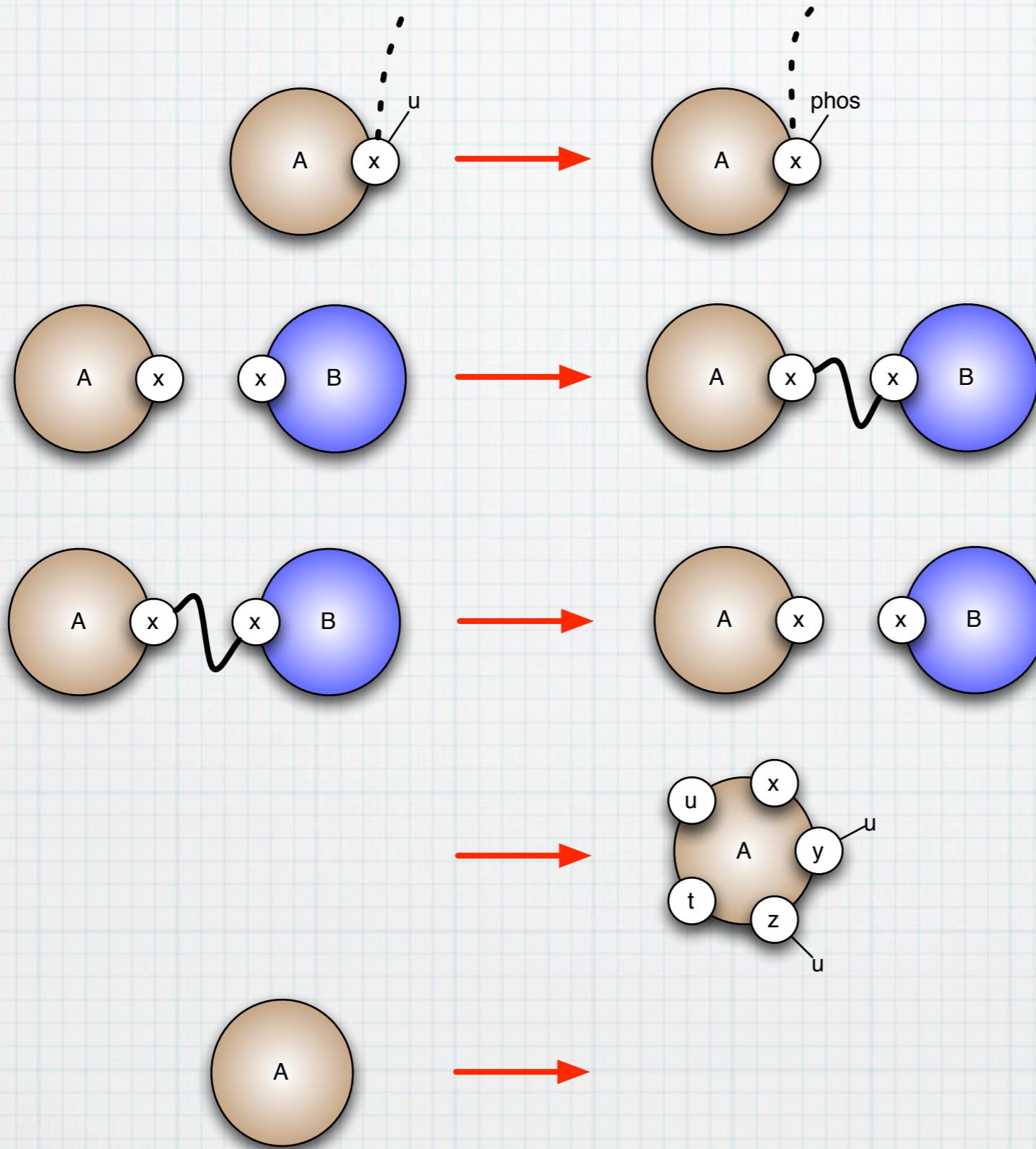
Quantification vs. representation

Trying to combine:

$$\begin{aligned}\dot{y}_1 &= \frac{v_{1b}(y_3^p/k_{1i}^p)}{k_{1b}^s(1+(y_3^p/k_{1i}^p))+(y_7^s+c)} - k_{1d}y_1 \\ \dot{y}_2 &= k_{2b}y_1^q - k_{2t}y_2 + k_{3t}y_3 - k_{2d}y_2 \\ \dot{y}_3 &= k_{2t}y_2 - k_{3t}y_3 - k_{3d}y_3 \\ \dot{y}_4 &= \frac{v_{4b}}{k_{4b}+(y_8^u/k_{4i}^u)} - k_{4d}y_4 \\ \dot{y}_5 &= k_{5b}y_4 - k_{5t}y_5 + k_{6t}y_6 - k_{5d}y_5 \\ \dot{y}_6 &= k_{5t}y_5 - k_{6t}y_6 - k_{6a}y_6 + k_{7a}y_7 - k_{6d}y_6 \\ \dot{y}_7 &= k_{6a}y_6 - k_{7a}y_7 - k_{7d}y_7 \\ \dot{y}_8 &= \frac{v_{8b}(y_7^r+c)}{k_{8d}y_8}\end{aligned}$$



Kappa overview



Site modification

Binding

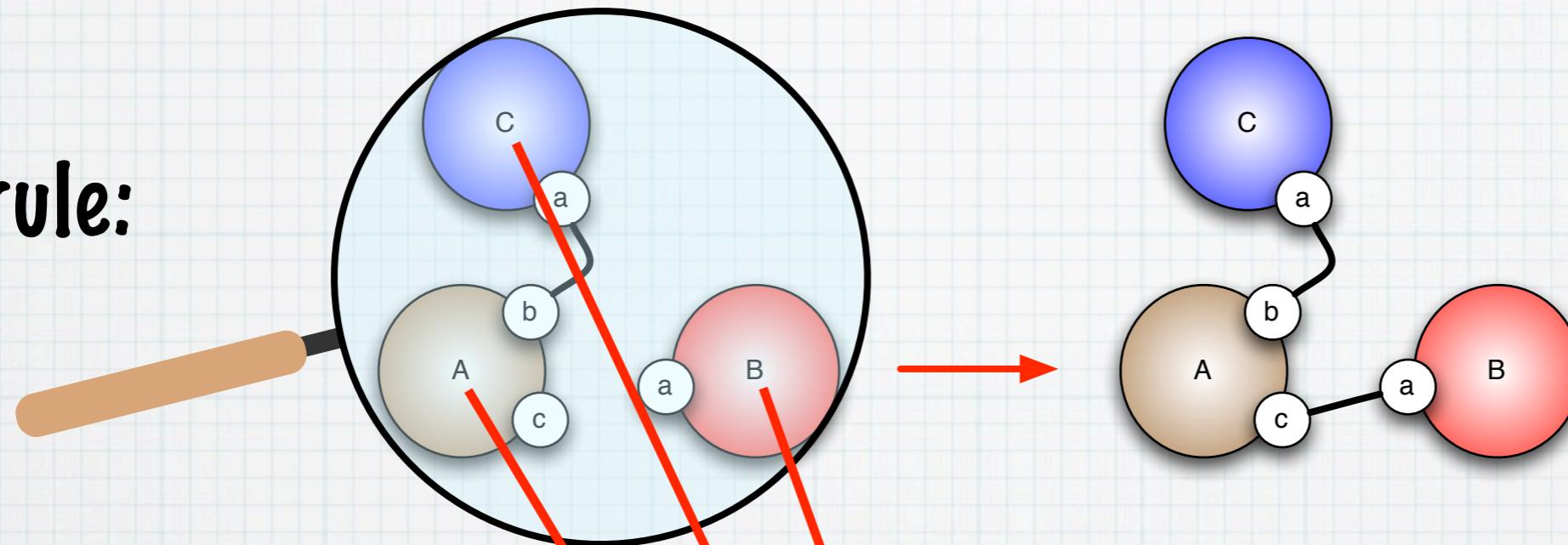
Unbinding

Synthesis

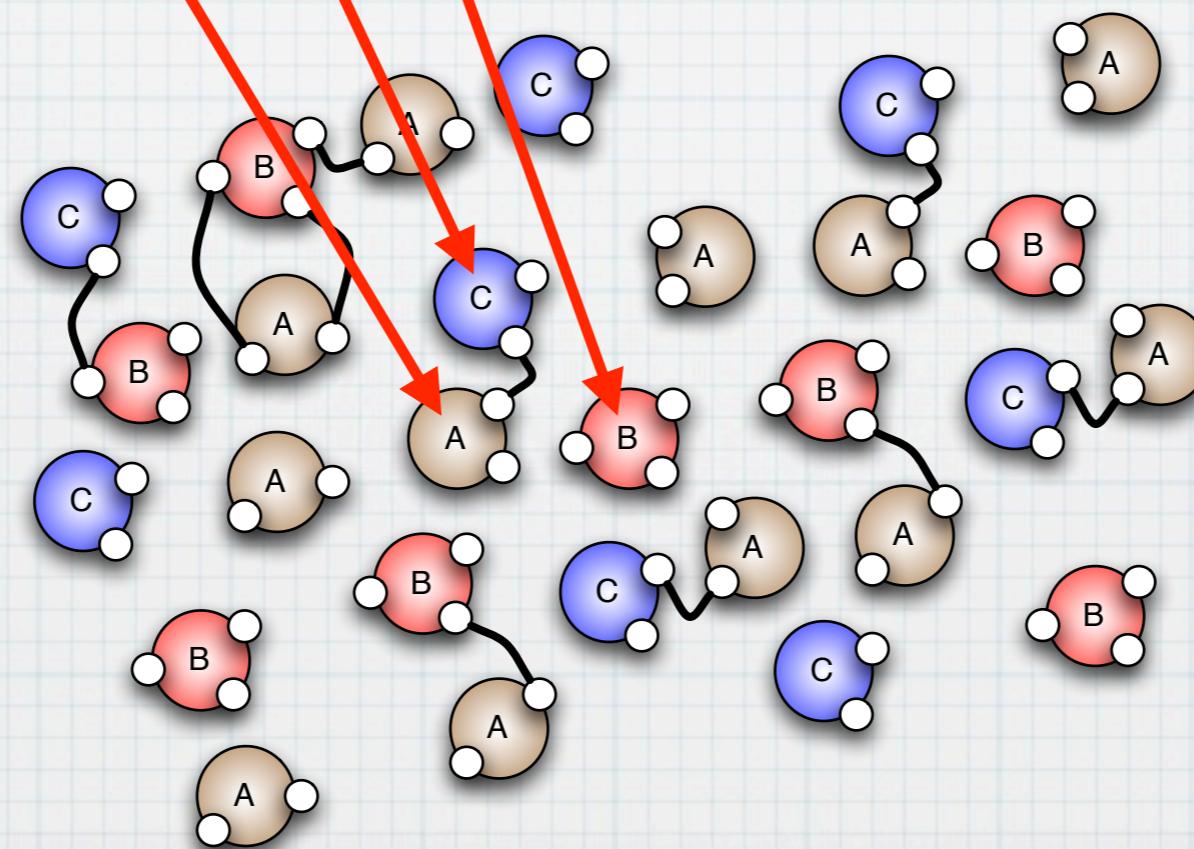
Degradation

Rule application

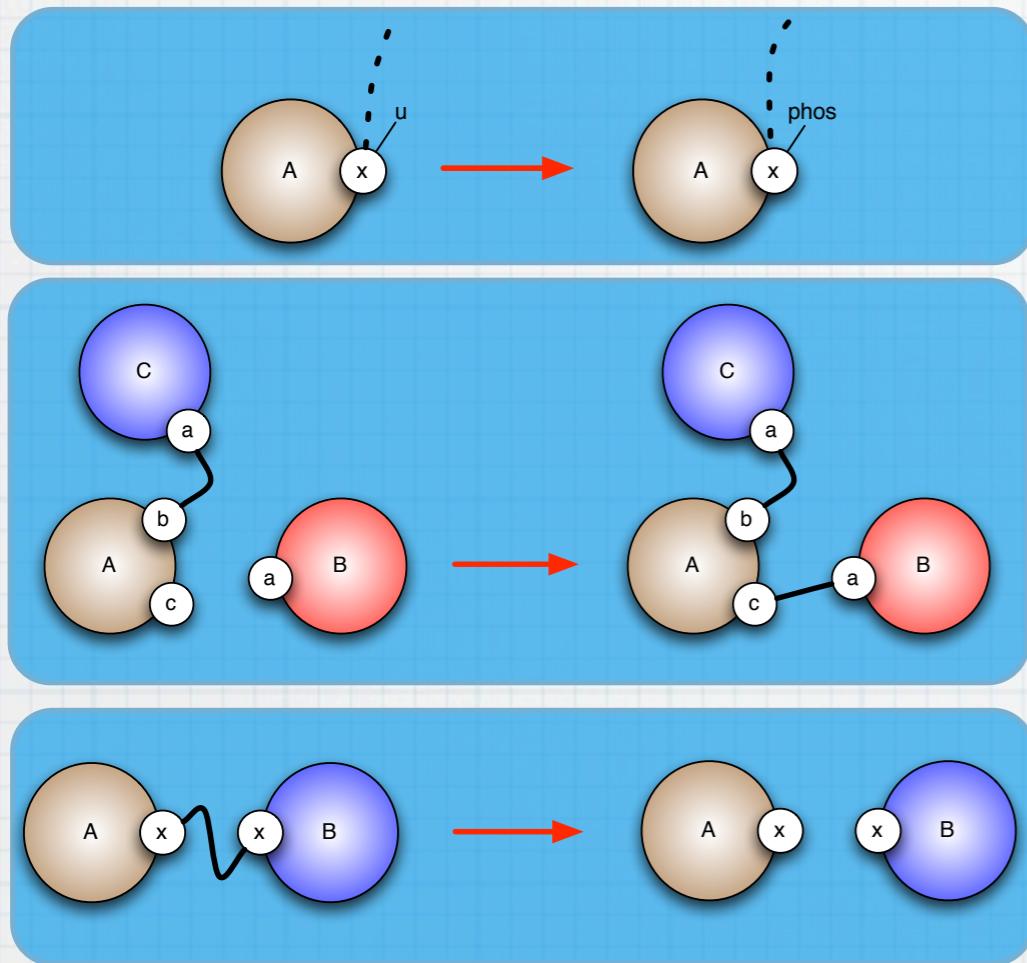
A rule:



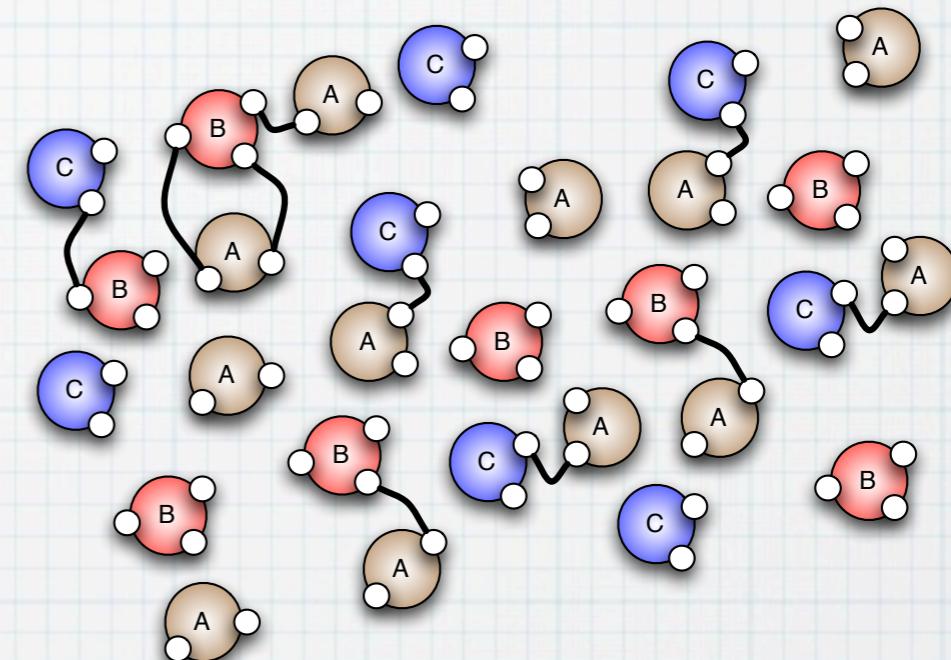
A match:



A Kappa-model

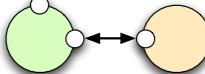
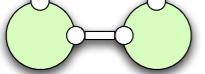
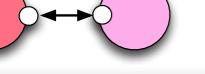
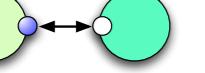


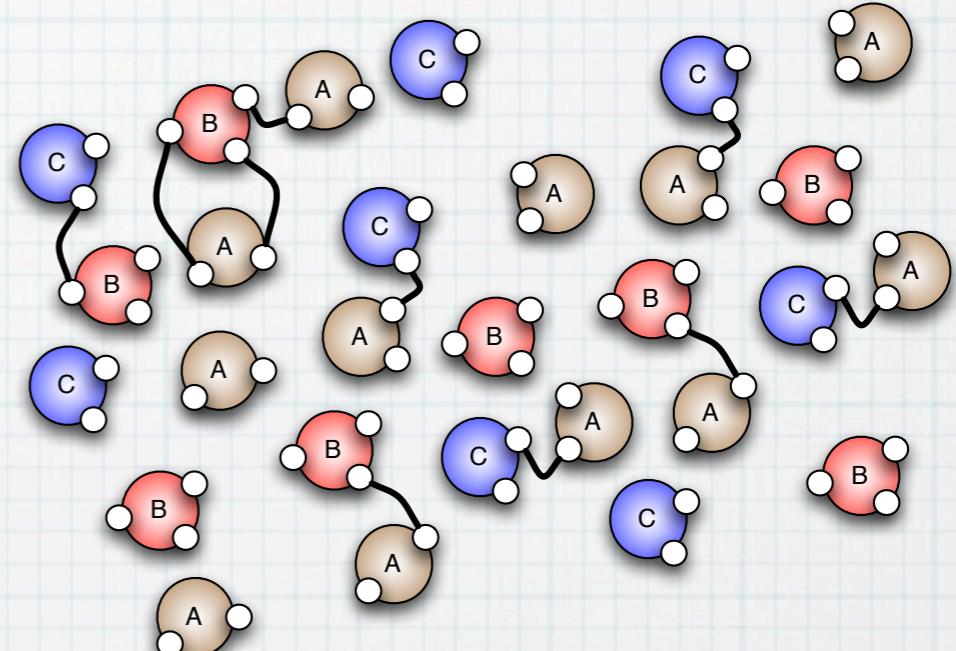
A rule set



An initial solution

A Kappa-model

| | | |
|----------|---|--|
| r_1 |  | $R(l), E(k) \longrightarrow R(l!1), E(k!1)$ $R(l!1), E(k!1) \longrightarrow R(l), E(k)$ |
| r_2 |  | $R(l!-, r), R(l!-, r) \longrightarrow R(l!-, r!1), R(l!-, r!1)$ |
| r_3 |  | $R(r!1), R(r!1) \longrightarrow R(r), R(r)$ |
| r_4 |  | $R(r!-, Y68 \sim u) \longrightarrow R(r!-, Y68 \sim p)$ |
| r_5 |  | $R(Y68 \sim p) \longrightarrow R(Y68 \sim u)$ |
| r_6 |  | $R(r!-, Y48 \sim u) \longrightarrow R(r!-, Y48 \sim p)$ |
| r_7 |  | $R(Y48 \sim p) \longrightarrow R(Y48 \sim u)$ |
| r_8 |  | $Shc(Y7 \sim p), Grb(a) \longrightarrow Shc(Y7 \sim p!1), Grb(a!1)$ $Shc(Y7 \sim p!1), Grb(a!1) \longrightarrow Shc(Y7 \sim p), Grb(a)$ |
| r_9 |  | $R(Y68 \sim p), Grb(a) \longrightarrow R(Y68 \sim p!1), Grb(a!1)$ $R(Y68 \sim p!1), Grb(a!1) \longrightarrow R(Y68 \sim p), Grb(a)$ |
| r_{10} |  | $Sos(d), Grb(b) \longrightarrow Sos(d!1), Grb(b!1)$ $Sos(d!1), Grb(b!1) \longrightarrow Sos(d), Grb(b)$ |
| r_{11} |  | $R(Y48 \sim p), Shc(b) \longrightarrow R(Y48 \sim p!1), Shc(b!1)$ $R(Y48 \sim p!1), Shc(b!1) \longrightarrow R(Y48 \sim p), Shc(b)$ |
| r_{12} |  | $R(r!-, Y48 \sim p!1), Shc(b!1), Y7 \sim u) \longrightarrow R(r!-, Y48 \sim p!1), Shc(b!1), Y7 \sim p)$ |
| r_{13} |  | $Shc(Y7 \sim p) \longrightarrow Shc(Y7 \sim u)$ |



Is this model correct?

A Kappa-model

| | | |
|--|---|-------|
| <input checked="" type="checkbox"/> C base deamination | DNA(Base~C,status~ok) => DNA(Base~U,status~dom) | 1.e-2 |
| <input checked="" type="checkbox"/> mC base deamination | DNA(Base~mC,status~ok) => DNA(Base~T,status~dom) | 1.e-2 |
| <input checked="" type="checkbox"/> Chromatin control | 6 rules | |
| <input checked="" type="checkbox"/> Chromatin compaction | MECP2(GmC!1),DNA(Base~mC,chr~opened!1) -> MECP2(GmC!1),DNA(Base~mC,chr~closed!1) | 1.e-2 |
| <input checked="" type="checkbox"/> Chromatin opening | DNA(chr~closed!1),CBP(chr!1) -> DNA(chr~opened),CBP(chr) | 1.e-2 |
| <input checked="" type="checkbox"/> DNA.chr--GmC.MECP2 | MECP2(GmC),DNA(Base~mC,chr~opened) -> MECP2(GmC!1),DNA(Base~mC,chr~opened!1) | 1.0 |
| <input checked="" type="checkbox"/> DNA(T:G)+MECP2 | MECP2(GmC!1),DNA(Base~T,chr!1) -> MECP2(GmC),DNA(Base~T,chr) | 1.e+5 |
| <input checked="" type="checkbox"/> DNA.chr--chr.CBP | DNA(chr~closed),CBP(chr) -> DNA(chr~closed!1),CBP(chr!1) | 1.0 |
| <input checked="" type="checkbox"/> DNA.chr--chr.CBP Ref1 | DNA(chr~closed),CBP(chr,tdg) -> DNA(chr~closed!1),CBP(chr!1,tdg) | 1.0 |
| <input checked="" type="checkbox"/> DNA.chr--chr.CBP Ref2 | DNA(chr~closed),CBP(chr,tdg!1),TDG(rd!1,N140,dnmt3a) -> DNA(chr~closed!1),CBP(chr!1,tdg!2),TDG(rd!2,N140,... | 1.0 |
| <input checked="" type="checkbox"/> DNA.chr--chr.CBP Ref3 | CBP(chr,tdg!1),DNA(chr~closed),TDG(rd!1,N140,dnmt3a!2),Dnmt3a(dna,tdg!2) -> CBP(chr!1,tdg!2),DNA(chr~close...) | 1.0 |
| <input checked="" type="checkbox"/> DNA.chr+chr.CBP | DNA(chr!1),CBP(chr!1) -> DNA(chr),CBP(chr) | 1.e-3 |
| <input checked="" type="checkbox"/> DNA repair | 17 rules | |
| <input checked="" type="checkbox"/> DNA.base--N140.TDG | DNA(Base,status~dom,chr~opened),TDG(N140) -> DNA(Base!1,status~dom,chr~opened),TDG(N140!1) | 1.0 |
| <input checked="" type="checkbox"/> DNA.base--N140.TDG Ref1 | DNA(Base,chr~opened,status~dom,rd),TDG(N140,rd,dnmt3a,lig) -> DNA(Base!1,chr~opened,status~dom,rd),TDG... 1.0 | |
| <input checked="" type="checkbox"/> DNA.base--N140.TDG Ref2 | DNA(Base,status~dom,chr~opened),TDG(N140,rd!1,dnmt3a,lig),CBP(tdg!1,chr) -> DNA(Base!1,status~dom,chr~o... 1.0 | |
| <input checked="" type="checkbox"/> DNA.base--N140.TDG Ref3 | DNA(status~dom,chr~opened,Base~T,rd!1),TDG(N140,rd!1,dnmt3a,lig) -> DNA(status~dom,chr~opened,Base~T!... 1.e+3 | |
| <input checked="" type="checkbox"/> DNA.base--N140.TDG Ref4 | DNA(Base,chr~opened,status~dom,rd),TDG(dnmt3a!1,rd,N140,lig),Dnmt3a(tdg!1,dna) -> DNA(Base!2,chr~opene... 1.0 | |
| <input checked="" type="checkbox"/> DNA.base--N140.TDG Ref5 | DNA(Base,status~dom,chr~opened),CBP(tdg!1,chr),TDG(N140,rd!1,dnmt3a!2,lig),Dnmt3a(dna,tdg!2) -> DNA(Base... 1.0 | |
| <input checked="" type="checkbox"/> DNA.base--N140.TDG Ref6 | DNA(status~dom,chr~opened,Base~T,rd!1),TDG(N140,rd!1,dnmt3a!2,lig),Dnmt3a(tdg!2,dna) -> DNA(status~dom... 1.e+3 | |
| <input checked="" type="checkbox"/> DNA.base--N140.TDG Ref7 | DNA(Base,chr~opened,status~dom,rd),TDG(N140,rd,dnmt3a,lig!1),APE1_PoB(tdg!1,CAT) -> DNA(Base!2,chr~ope... 1.0 | |
| <input checked="" type="checkbox"/> DNA.base--N140.TDG Ref8 | DNA(Base,status~dom,chr~opened),CBP(tdg!1,chr),TDG(N140,rd!1,dnmt3a,lig!2),APE1_PoB(tdg!2,CAT) -> DNA(B... 1.0 | |
| <input checked="" type="checkbox"/> DNA.base--N140.TDG Ref9 | DNA(status~dom,chr~opened,Base~T,rd!1),TDG(rd!1,lig!2,dnmt3a,N140),APE1_PoB(tdg!2,CAT) -> DNA(status~d... 1.0 | |
| <input checked="" type="checkbox"/> DNA.base--N140.TDG Ref10 | DNA(Base,chr~opened,status~dom,rd),Dnmt3a(tdg!1,dna),TDG(dnmt3a!1,N140,rd,lig!2),APE1_PoB(tdg!2,CAT) -> ... 1.0 | |
| <input checked="" type="checkbox"/> DNA.base--N140.TDG Ref11 | DNA(Base,chr~opened,status~dom,rd),CBP(tdg!1,chr),Dnmt3a(dna,tdg!2),TDG(N140,rd!1,dnmt3a!2,lig!3),APE1_PoB(... 1.0 | |
| <input checked="" type="checkbox"/> DNA.base--N140.TDG Ref12 | DNA(Base,chr~opened,status~dom,rd),CBP(tdg!1,chr),Dnmt3a(tdg!2,dna),TDG(rd!1,lig!3,dnmt3a!2,N140),APE1_PoB(tdg!3,C... 1.0 | |
| <input checked="" type="checkbox"/> DNA.base+N140.TDG | DNA(Base),TDG(N140) | 1.e-5 |
| <input checked="" type="checkbox"/> DNA(T:G).rd--rd.TDG core | DNA(rd),TDG(rd) -> DNA(rd!),TDG(rd!1) | 1.e+3 |
| <input checked="" type="checkbox"/> DNA(T:G).rd--rd.TDG | DNA(Base~T!1,rd),TDG(N140!1,rd) -> DNA(Base~T!2,rd!1),TDG(N140!2,rd!1) | 1.e+5 |
| <input checked="" type="checkbox"/> DNA.rd+rd.TDG | DNA(rd!1),TDG(rd!1,nc~act,N140) -> DNA(rd),TDG(rd,nc~act,N140) | 1.0 |
| <input checked="" type="checkbox"/> DNA.rd+rd.TDG Ref1 | DNA(rd!1),TDG(nc~act,N140,rd~Ac!1) -> DNA(rd),TDG(nc~act,N140,rd~Ac) | 1.0 |
| <input checked="" type="checkbox"/> DNA.rd+rd.TDG Ref2 | DNA(rd!1,Base!2),TDG(rd!1,N140!2,nc~act) -> DNA(rd,Base!1),TDG(rd,N140!1,nc~act) | 1.e-5 |
| <input checked="" type="checkbox"/> DNA.rd+rd.TDG Ref3 | DNA(rd!1),TDG(rd!1,nc~act,N140) -> DNA(rd),TDG(rd,nc~act,N140) | 1.0 |
| <input checked="" type="checkbox"/> Dnmt3a+TDG | Dnmt3a(tdg!2),TDG(dnmt3a!2) -> Dnmt3a(tdg),TDG(dnmt3a) | 1.e-3 |
| <input checked="" type="checkbox"/> Dnmt3a+TDG Ref1 | Dnmt3a(tdg!2),TDG(dnmt3a!2,rd) -> Dnmt3a(tdg),TDG(dnmt3a,rd) | 1.0 |
| <input checked="" type="checkbox"/> Dnmt3a+TDG Ref2 | Dnmt3a(tdg!2),TDG(dnmt3a!2,rd!) -> Dnmt3a(tdg),TDG(dnmt3a,rd!) | 1.e-3 |
| <input checked="" type="checkbox"/> Dnmt3a--TDG | Dnmt3a(tdg, dna),TDG(dnmt3a) -> Dnmt3a(tdg!2, dna),TDG(dnmt3a!2) | 1.0 |
| <input checked="" type="checkbox"/> Dnmt3a--TDG Ref1 | Dnmt3a(tdg, dna),TDG(dnmt3a,rd) -> Dnmt3a(tdg!2, dna),TDG(dnmt3a!2,rd) | 1.0 |
| <input checked="" type="checkbox"/> Dnmt3a--TDG Ref2 | Dnmt3a(tdg, dna),TDG(dnmt3a,rd!) -> Dnmt3a(tdg!2, dna),TDG(dnmt3a!2,rd!) | 100.0 |
| <input checked="" type="checkbox"/> TDG.lig--tdg.APE1 | TDG(lig),APE1_PoB(tdg) -> TDG(lig!1),APE1_PoB(tdg!1) | 1.0 |
| <input checked="" type="checkbox"/> TDG.lig--tdg.APE1 Ref1 | TDG(lig),APE1_PoB(tdg,CAT) -> TDG(lig!1),APE1_PoB(tdg!1,CAT) | 1.0 |
| <input checked="" type="checkbox"/> TDG.lig--tdg.APE1 Ref2 | TDG(lig,rd,N140),APE1_PoB(tdg,CAT!1),DNA(Base!1,rd) -> TDG(lig!1,rd,N140),APE1_PoB(tdg!1,CAT!2),DNA(Base... 1.0 | |
| <input checked="" type="checkbox"/> TDG.lig--tdg.APE1 Ref3 | DNA(Base!1,rd!2),TDG(rd!2,lig,N140),APE1_PoB(CAT!1,tdg) -> DNA(Base!1,rd!2),TDG(rd!2,lig!3,N140),APE1_Po... 1.e+5 | |
| <input checked="" type="checkbox"/> TDG.lig+tdg.APE1 | TDG(lig!1),APE1_PoB(tdg!1) -> TDG(lig),APE1_PoB(tdg) | 1.e-3 |
| <input checked="" type="checkbox"/> Repair T | DNA(status~dom,Base~T!1,rd!2),TDG(lig,rd!2,N140!1) => DNA(status~hole,Base~x,rd!1),TDG(lig,rd!1,N140) | 1.e+5 |
| <input checked="" type="checkbox"/> Repair T with swap | DNA(Base~T!1,status~dom,rd!2),TDG(N140!1,lig!3,rd!2),APE1_PoB(tdg!3,CAT) => DNA(Base~x!3,status~hole,rd!... 1.e+4 | |
| <input checked="" type="checkbox"/> Repair U | DNA(Base~U!1,status~dom),TDG(N140!1,lig) -> DNA(Base~x,status~hole),TDG(N140,lig) | 1.e+5 |
| <input checked="" type="checkbox"/> Repair U with swap | DNA(Base~U!1,status~dom),TDG(N140!1,lig!2),APE1_PoB(tdg!2,CAT) => DNA(Base~x!2,status~hole),TDG(N140,li... 1.e+4 | |
| <input checked="" type="checkbox"/> DNA(x:G)--CAT.APE1_PoB1 | APE1_PoB(CAT),DNA(Base~x,chr~opened) -> APE1_PoB(CAT!1),DNA(Base~x!1,chr~opened) | 1.0 |
| <input checked="" type="checkbox"/> DNA(x:G)--CAT.APE1_PoB1 Ref1 | APE1_PoB(CAT,tdg),DNA(Base~x,chr~opened) -> APE1_PoB(CAT!1,tdg),DNA(Base~x!1,chr~opened) | 1.0 |
| <input checked="" type="checkbox"/> DNA(x:G)--CAT.APE1_PoB1 Ref2 | APE1_PoB(CAT,tdg!1),DNA(Base~x,chr~opened),TDG(lig!1,N140,rd) -> APE1_PoB(CAT!1,tdg!2),DNA(Base~x!1,ch... 1.0 | |
| <input checked="" type="checkbox"/> DNA(x:G)--CAT.APE1_PoB1 Ref3 | APE1_PoB(CAT,tdg!1,TDG(rd!1,lig!1)),DNA(Base~x,chr~opened,rd!1,TDG(rd!1,lig!1)) -> APE1_PoB(CAT!1,tdg!2),DNA(Base~x!1,chr~open... 1.0 | |

And that one?

Outline

Outline

1. Accessibility or how to debug a model?

| |
|---|
| <input checked="" type="checkbox"/> C base deamination |
| <input checked="" type="checkbox"/> mC base deamination |
| <input checked="" type="checkbox"/> Chromatin control |
| <input checked="" type="checkbox"/> Chromatin compaction |
| <input checked="" type="checkbox"/> Chromatin opening |
| <input checked="" type="checkbox"/> DNA.chr--GmC.MECP2 |
| <input checked="" type="checkbox"/> DNA(T:G)+MECP2 |
| <input checked="" type="checkbox"/> DNA.chr--chr.CBP |
| <input checked="" type="checkbox"/> DNA.chr--chr.CBP Ref1 |
| <input checked="" type="checkbox"/> DNA.chr--chr.CBP Ref2 |
| <input checked="" type="checkbox"/> DNA.chr--chr.CBP Ref3 |
| <input checked="" type="checkbox"/> DNA.chr+chr.CBP |
| <input checked="" type="checkbox"/> DNA repair |
| <input checked="" type="checkbox"/> DNA.base--N140.TDG |
| <input checked="" type="checkbox"/> DNA.base--N140.TDG Ref1 |
| <input checked="" type="checkbox"/> DNA.base--N140.TDG Ref2 |
| <input checked="" type="checkbox"/> DNA.base--N140.TDG Ref3 |
| <input checked="" type="checkbox"/> DNA.base--N140.TDG Ref4 |
| <input checked="" type="checkbox"/> DNA.base--N140.TDG Ref5 |
| <input checked="" type="checkbox"/> DNA.base--N140.TDG Ref6 |
| <input checked="" type="checkbox"/> DNA.base--N140.TDG Ref7 |
| <input checked="" type="checkbox"/> DNA.base--N140.TDG Ref8 |
| <input checked="" type="checkbox"/> DNA.base--N140.TDG Ref9 |
| <input checked="" type="checkbox"/> DNA.base--N140.TDG Ref10 |
| <input checked="" type="checkbox"/> DNA.base--N140.TDG Ref11 |
| <input checked="" type="checkbox"/> DNA.base--N140.TDG Ref12 |
| <input checked="" type="checkbox"/> DNA.base+N140.TDG |
| <input checked="" type="checkbox"/> DNA(T:G).rd--rd.TDG core |
| <input checked="" type="checkbox"/> DNA(T:G).rd--rd.TDG |
| <input checked="" type="checkbox"/> DNA.rd+rd.TDG |
| <input checked="" type="checkbox"/> DNA.rd+rd.TDG Ref1 |
| <input checked="" type="checkbox"/> DNA.rd+rd.TDG Ref2 |
| <input checked="" type="checkbox"/> DNA.rd+rd.TDG Ref3 |
| <input checked="" type="checkbox"/> Dnmt3a+TDG |
| <input checked="" type="checkbox"/> Dnmt3a+TDG Ref1 |
| <input checked="" type="checkbox"/> Dnmt3a+TDG Ref2 |
| <input checked="" type="checkbox"/> Dnmt3a--TDG |
| <input checked="" type="checkbox"/> Dnmt3a--TDG Ref1 |
| <input checked="" type="checkbox"/> Dnmt3a--TDG Ref2 |
| <input checked="" type="checkbox"/> TDG.lig--tdg.APE1 |
| <input checked="" type="checkbox"/> TDG.lig--tdg.APE1 Ref1 |
| <input checked="" type="checkbox"/> TDG.lig--tdg.APE1 Ref2 |
| <input checked="" type="checkbox"/> TDG.lig--tdg.APE1 Ref3 |
| <input checked="" type="checkbox"/> TDG.lig+tdg.APE1 |
| <input checked="" type="checkbox"/> Repair T |
| <input checked="" type="checkbox"/> Repair T with swap |
| <input checked="" type="checkbox"/> Repair U |
| <input checked="" type="checkbox"/> Repair U with swap |
| <input checked="" type="checkbox"/> DNA(x;G)--CAT.APE1_PolB1 |
| <input checked="" type="checkbox"/> DNA(x;G)--CAT.APE1_PolB1 Ref1 |
| <input checked="" type="checkbox"/> DNA(x;G)--CAT.APE1_PolB1 Ref2 |

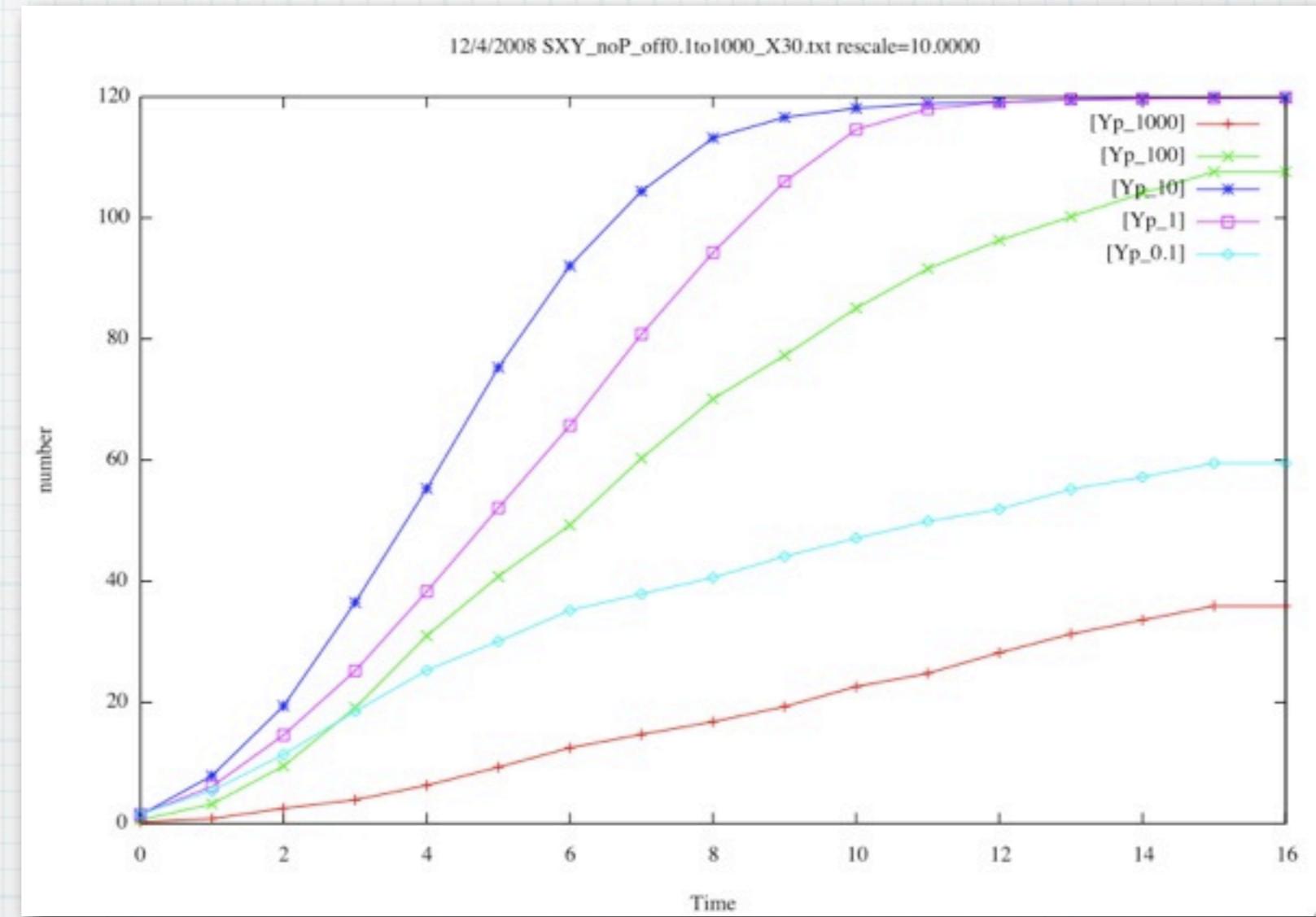
| | |
|---|-------|
| DNA(Base-C,status-ok) => DNA(Base-U,status-dom) | 1.e-2 |
| DNA(Base-mC,status-ok) => DNA(Base-T,status-dom) | 1.e-2 |
| 6 rules | |
| MECP2(GmC!1),DNA(Base-mC,chr~opened!1) -> MECP2(GmC!1),DNA(Base-mC,chr~closed!1) | 1.e-2 |
| DNA(chr~closed!1),CBP(chr!1) -> DNA(chr~opened),CBP(chr) | 1.e-2 |
| MECP2(GmC!),DNA(Base-mC,chr~opened) -> MECP2(GmC!),DNA(Base-mC,chr~opened!1) | 1.0 |
| MECP2(GmC!1),DNA(Base-T,chr!1) -> MECP2(GmC!),DNA(Base-T,chr) | 1.e+5 |
| DNA(chr~closed),CBP(chr) -> DNA(chr~closed!1),CBP(chr!1) | 1.0 |
| DNA(chr~closed),CBP(chr,tdg) -> DNA(chr~closed!1),CBP(chr!1,tdg) | 1.0 |
| DNA(chr~closed),CBP(chr,tdg!1),TDG(rd!1,N140,dnmt3a) -> DNA(chr~closed!1),CBP(chr!1,tdg!2),TDG(rd!2,N140,... | 1.0 |
| CBP(chr,tdg!1),DNA(chr~closed),TDG(rd!1,N140,dnmt3a!2),Dnmt3a(dna,tdg!2) -> CBP(chr!1,tdg!2),DNA(chr~close... 1.0 | |
| DNA(chr!1),CBP(chr!1) -> DNA(chr),CBP(chr) | 1.e-3 |
| 1 rules | |
| DNA(Base,status-dom,chr~opened),TDG(N140) -> DNA(Base!1,status-dom,chr~opened),TDG(N140!1) | 1.0 |
| DNA(Base,chr~opened,status-dom,rd),TDG(N140,rd,dnmt3a,lig) -> DNA(Base!1,chr~opened,status-dom,rd),TDG(... 1.0 | |
| DNA(Base,status-dom,chr~opened),TDG(N140,rd!1,dnmt3a,lig),CBP(tdg!1,chr) -> DNA(Base!1,status-dom,chr~o... 1.0 | |
| DNA(status-dom,chr~opened,Base-T,rd!1),TDG(N140,rd!1,dnmt3a,lig) -> DNA(status-dom,chr~opened,Base-T!... 1.e+3 | |
| DNA(Base,chr~opened,status-dom,rd),TDG(dnmt3a!1,rd,N140,lig),Dnmt3a(tdg!1,dna) -> DNA(Base!2,chr~opene... 1.0 | |
| DNA(Base,status-dom,chr~opened),CBP(tdg!1,chr),TDG(N140,rd!1,dnmt3a!2,lig),Dnmt3a(dna,tdg!2) -> DNA(Base... 1.0 | |
| DNA(status-dom,chr~opened,Base-T,rd!1),TDG(N140,rd!1,dnmt3a!2,lig),Dnmt3a(tdg!2,dna) -> DNA(status-dom... 1.e+3 | |
| DNA(Base,chr~opened,status-dom,rd),TDG(N140,rd,dnmt3a,lig!1),APE1_PolB(tdg!1,CAT) -> DNA(Base!2,chr~ope... 1.0 | |
| DNA(Base,status-dom,chr~opened),CBP(tdg!1,chr),TDG(N140,rd!1,dnmt3a,lig!2),APE1_PolB(tdg!2,CAT) -> DNA(... 1.0 | |
| DNA(status-dom,chr~opened,Base-T,rd!1),TDG(rd!1,lig!2,dnmt3a,N140),APE1_PolB(tdg!2,CAT) -> DNA(status-d... 1.0 | |
| DNA(Base,chr~opened,status-dom,rd),Dnmt3a(tdg!1,dna),TDG(dnmt3a!1,N140,rd,lig!2),APE1_PolB(tdg!2,CAT) -> ... 1.0 | |
| DNA(Base,status-dom,chr~opened),CBP(tdg!1,chr),Dnmt3a(dna,tdg!2),TDG(N140,rd!1,dnmt3a!2,lig!3),APE1_PolB(... 1.0 | |
| DNA(status-dom,chr~opened,Base-T,rd!1),Dnmt3a(tdg!2,dna),TDG(rd!1,lig!3,dnmt3a!2,N140),APE1_PolB(tdg!3,C... 1.0 | |
| DNA(Base!1),TDG(N140!1) -> DNA(Base),TDG(N140) | 1.e-5 |
| DNA(rd),TDG(rd) -> DNA(rd!1),TDG(rd!1) | 1.e+3 |
| DNA(Base-T!1,rd),TDG(N140!1,rd) -> DNA(Base-T!2,rd!1),TDG(N140!2,rd!1) | 1.e+5 |
| DNA(rd!1),TDG(rd!1,nc-act,N140) -> DNA(rd),TDG(rd,nc-act,N140) | 1.0 |
| DNA(rd!1),TDG(nc-act,N140,rd~Ac!1) -> DNA(rd),TDG(nc-act,N140,rd~Ac) | 1.0 |
| DNA(rd!1,Base!2),TDG(rd!1,N140!2,nc-act) -> DNA(rd,Base!1),TDG(rd,N140!1,nc-act) | 1.e-5 |
| DNA(rd!1),TDG(rd!1,nc-act,N140) -> DNA(rd),TDG(rd,nc-act,N140) | 1.0 |
| Dnmt3a(tdg!2),TDG(dnmt3a!2) -> Dnmt3a(tdg),TDG(dnmt3a) | 1.e-3 |
| Dnmt3a(tdg!2),TDG(dnmt3a!2,rd) -> Dnmt3a(tdg),TDG(dnmt3a,rd) | 1.0 |
| Dnmt3a(tdg!2),TDG(dnmt3a!2,rd!) -> Dnmt3a(tdg),TDG(dnmt3a,rd!) | 1.e-3 |
| Dnmt3a(tdg,dna),TDG(dnmt3a) -> Dnmt3a(tdg,dna),TDG(dnmt3a!2) | 1.0 |
| Dnmt3a(tdg,dna),TDG(dnmt3a,rd) -> Dnmt3a(tdg!2,dna),TDG(dnmt3a!2,rd) | 1.0 |
| Dnmt3a(tdg,dna),TDG(dnmt3a,rd!) -> Dnmt3a(tdg!2,dna),TDG(dnmt3a!2,rd!) | 100.0 |
| TDG(lig),APE1_PolB(tdg) -> TDG(lig!1),APE1_PolB(tdg!1) | 1.0 |
| TDG(lig),APE1_PolB(tdg,CAT) -> TDG(lig!1),APE1_PolB(tdg!1,CAT) | 1.0 |
| TDG(lig,rd,N140),APE1_PolB(tdg,CAT!1),DNA(Base!1,rd) -> TDG(lig!1,rd,N140),APE1_PolB(tdg!1,CAT!2),DNA(... 1.0 | |
| DNA(Base!1,rd!2),TDG(rd!2,lig,N140),APE1_PolB(CAT!1,tdg) -> DNA(Base!1,rd!2),TDG(rd!2,lig!3,N140),APE1_Pol... 1.e+5 | |
| TDG(lig!1),APE1_PolB(tdg!1) -> TDG(lig),APE1_PolB(tdg) | 1.e-3 |
| DNA(status-dom,Base-T!1,rd!2),TDG(lig,rd!2,N140!1) -> DNA(status-hole,Base-x,rd!1),TDG(lig,rd!1,N140) | 1.e+5 |
| DNA(Base-T!1,status-dom,rd!2),TDG(N140!1,lig!3,rd!2),APE1_PolB(tdg!3,CAT) -> DNA(Base-x!3,status-hole,rd!... 1.e+4 | |
| DNA(Base-U!1,status-dom),TDG(N140!1,lig) -> DNA(Base-x,status-hole),TDG(N140,lig) | 1.e+5 |
| DNA(Base-U!1,status-dom),TDG(N140!1,lig!2),APE1_PolB(tdg!2,CAT) -> DNA(Base-x!2,status-hole),TDG(N140,li... 1.e+4 | |
| APE1_PolB(CAT),DNA(Base-x,chr~opened) -> APE1_PolB(CAT!1),DNA(Base-x!1,chr~opened) | 1.0 |
| APE1_PolB(CAT,tdg),DNA(Base-x,chr~opened) -> APE1_PolB(CAT!1,tdg),DNA(Base-x!1,chr~opened) | 1.0 |
| APE1_PolB(CAT,tdg!1),DNA(Base-x,chr~opened),TDG(lig!1,N140,rd) -> APE1_PolB(CAT!1,tdg!2),DNA(Base-x!1,ch... 1.0 | |
| APE1_PolB(CAT) -> DNA(Base-x,chr~opened) | |



Outline

1. Accessibility or how to debug a model?

2. Simulation or how to execute a model?



Outline

1. Accessibility or how to debug a model?

2. Simulation or how to execute a model?

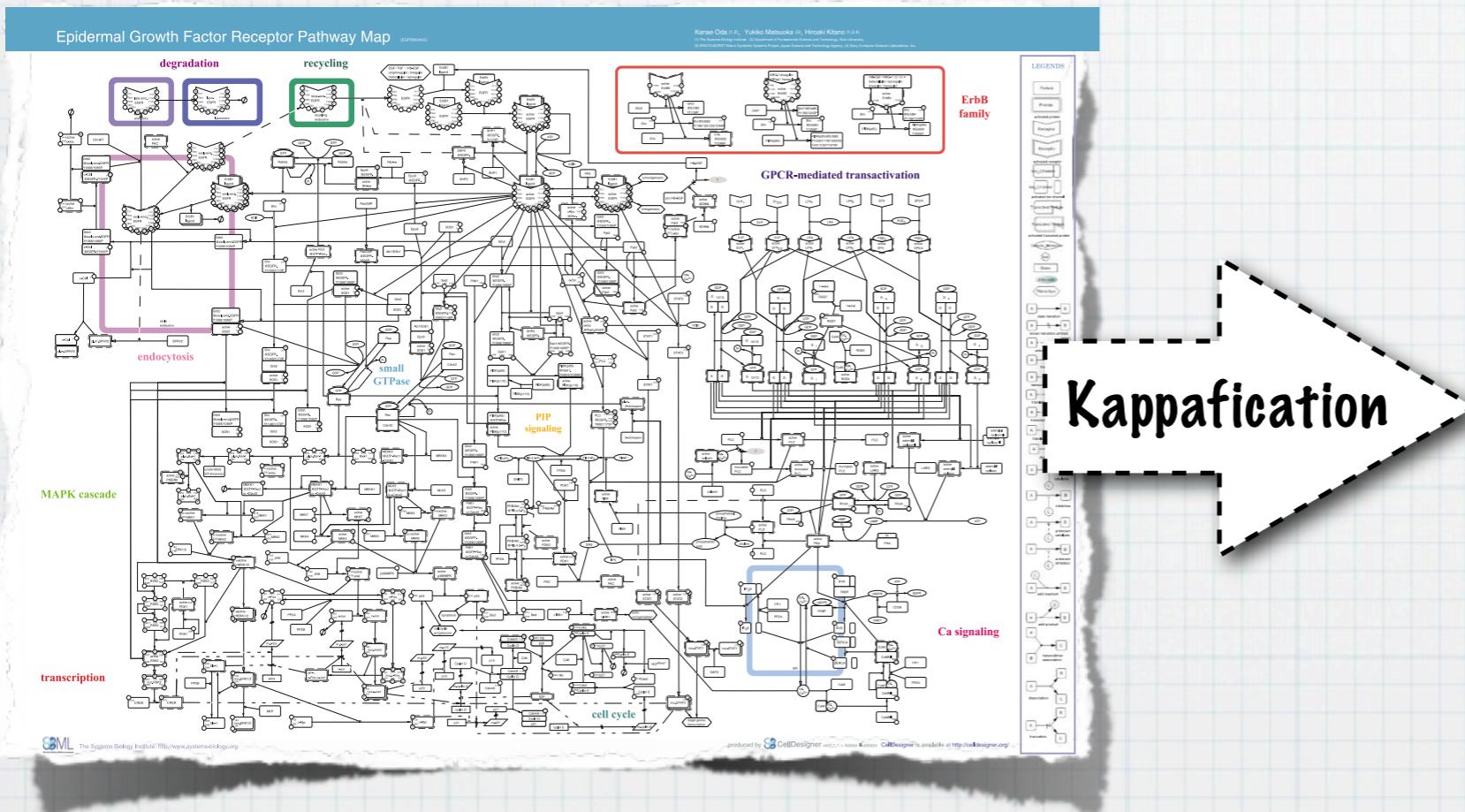
3. Producing pathways or how to understand a model?

Static analysis...

How to trust a model?

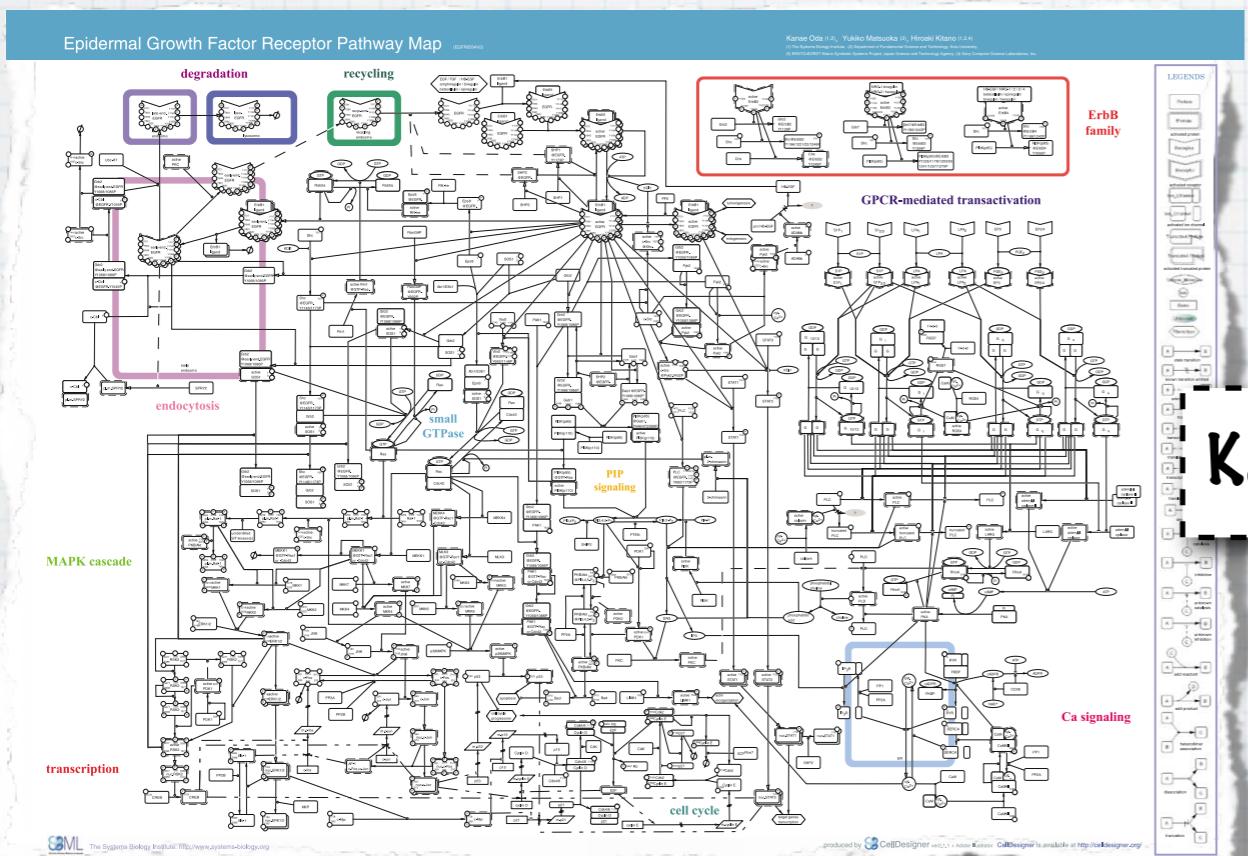
Programs as models: the good

A model can be wrong because of an incorrect biological hypothesis...

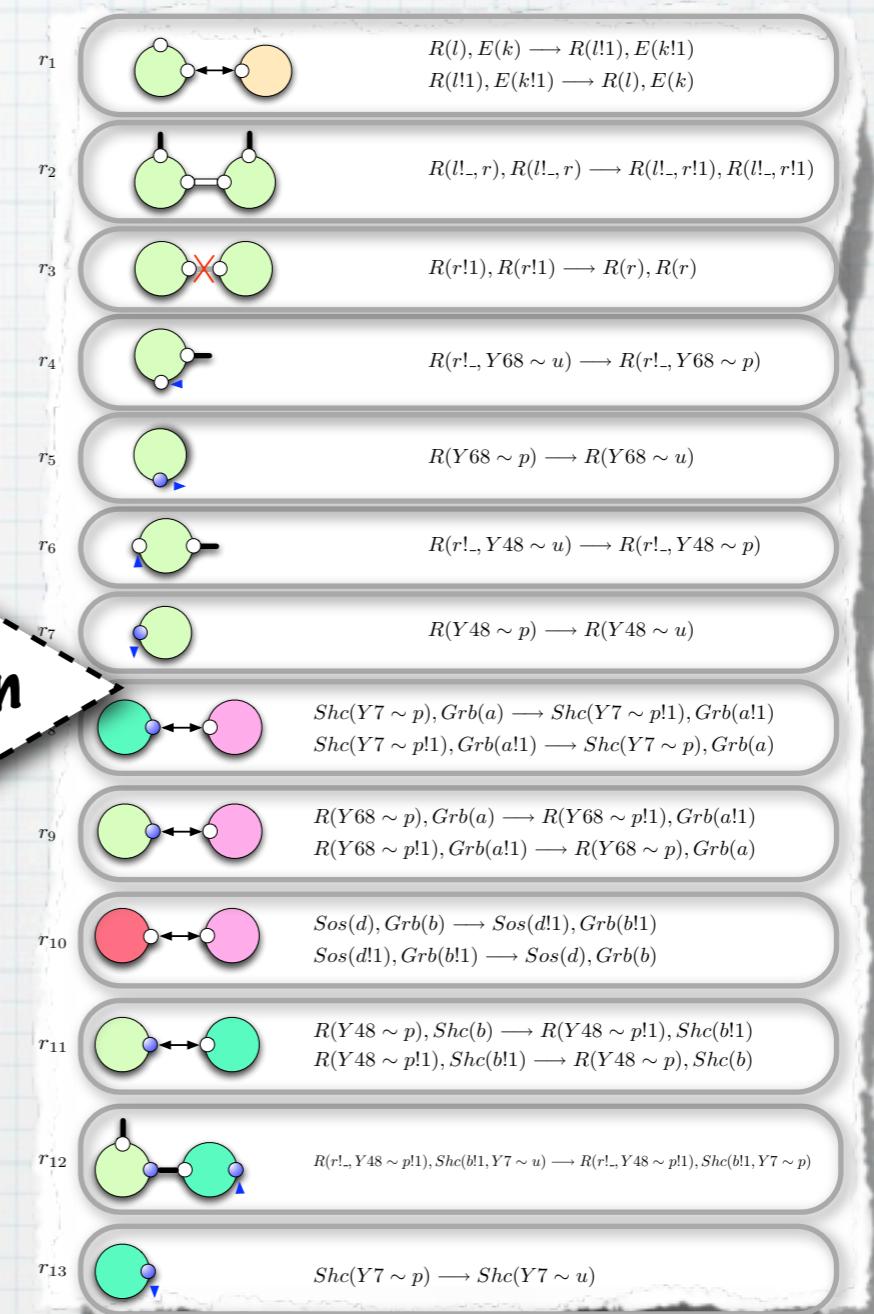


Programs as models: the good

A model can be wrong because of an incorrect biological hypothesis...

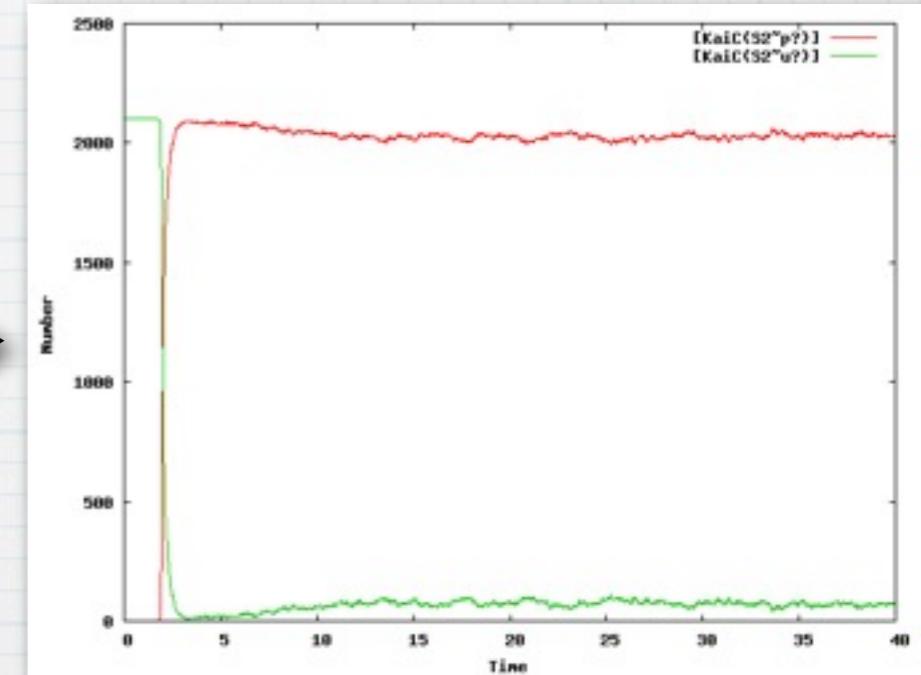
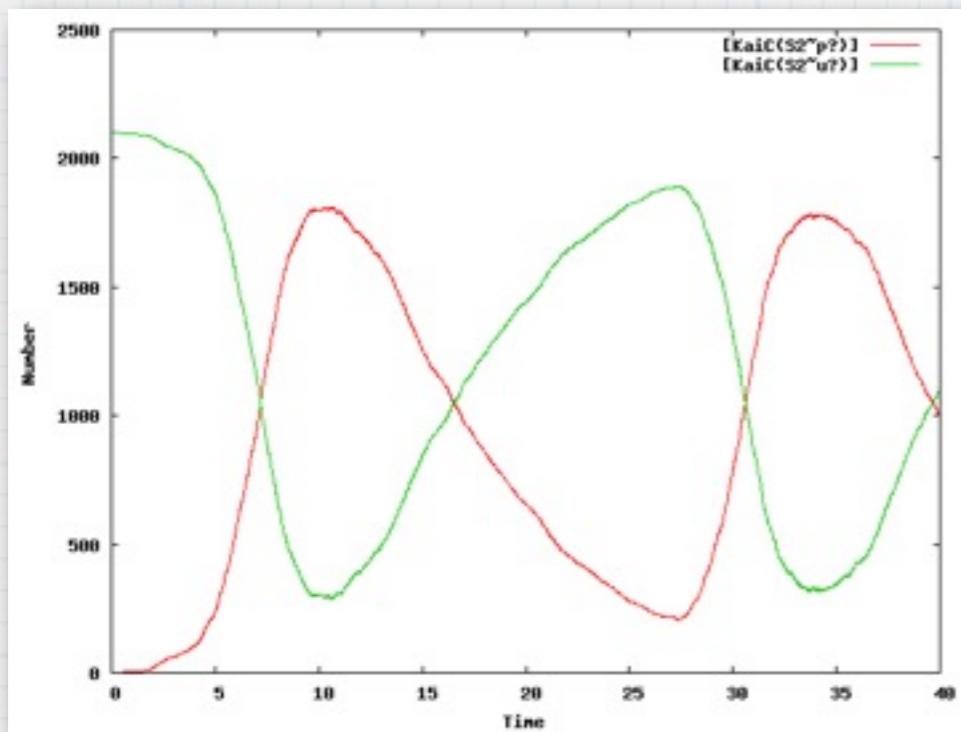


Kappaification



Programs as models: the good

A model can be wrong because of an incorrect biological hypothesis...



Programs as models: the bad

A model can be wrong because of a syntactic mistake...

Programs as models: the bad

A model can be wrong because of a syntactic mistake...

$$A(d^1, b^2), B(a^2, c^3), C(b^3, d^4), D(c^4, a^1) \rightarrow A(d^1, b^2), B(a, c^3), C(b^3, d^4), D(c^4, a^1)$$

Programs as models: the bad

A model can be wrong because of a syntactic mistake...

$A(d^1, b^2), B(a^2, c^3), C(b^3, d^4), D(c^4, a^1) \rightarrow A(d^1, b^{\textcircled{2}}), B(a, c^3), C(b^3, d^4), D(c^4, a^1)$

Error, Line 1029: syntax error, dangling link 2

Programs as models: the bad

A model can be wrong because of a syntactic mistake...

$A(d^1, b^2), B(a^2, c^3), C(b^3, d^4), D(c^4, a^1) \rightarrow A(d^1, b^{\textcircled{2}}), B(a, c^3), C(b^3, d^4), D(c^4, a^1)$

Error, Line 1029: syntax error, dangling link 2

$A(d^1, b^2), B(a^2, c_{\text{phos}}^3), C(b^3, d^4), D(c^4, a^1) \rightarrow A(d^1, b), B(a, c_{\text{Phos}}^3), C(b^3, d^4), D(c^4, a^1)$

Programs as models: the bad

A model can be wrong because of a syntactic mistake...

$A(d^1, b^2), B(a^2, c^3), C(b^3, d^4), D(c^4, a^1) \rightarrow A(d^1, b^2), B(a, c^3), C(b^3, d^4), D(c^4, a^1)$

Error, Line 1029: syntax error, dangling link 2

$A(d^1, b^2), B(a^2, c_{\text{phos}}^3), C(b^3, d^4), D(c^4, a^1) \rightarrow A(d^1, b), B(a, c_{\text{Phos}}^3), C(b^3, d^4), D(c^4, a^1)$

Warning, Line 733: rule is not atomic, actions are

- BRK(A#1,b)(B#2,a)
- MOD(B#2,c,phos,Phos)

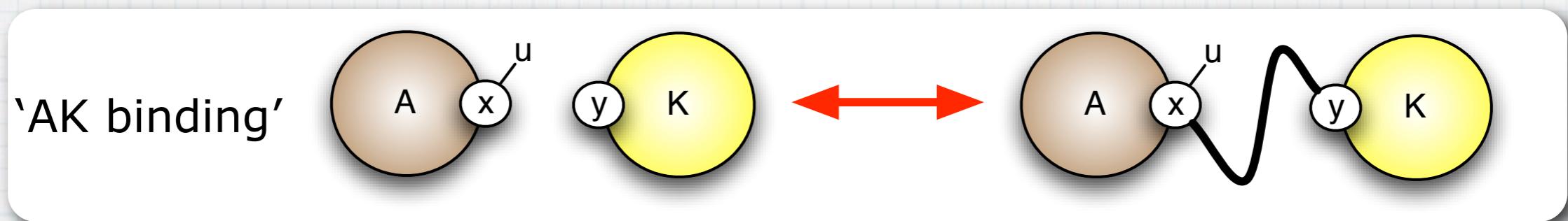
But a graphical interface may prevent these!

Programs as models: the ugly!

A model can be wrong because of a semantic mistake...

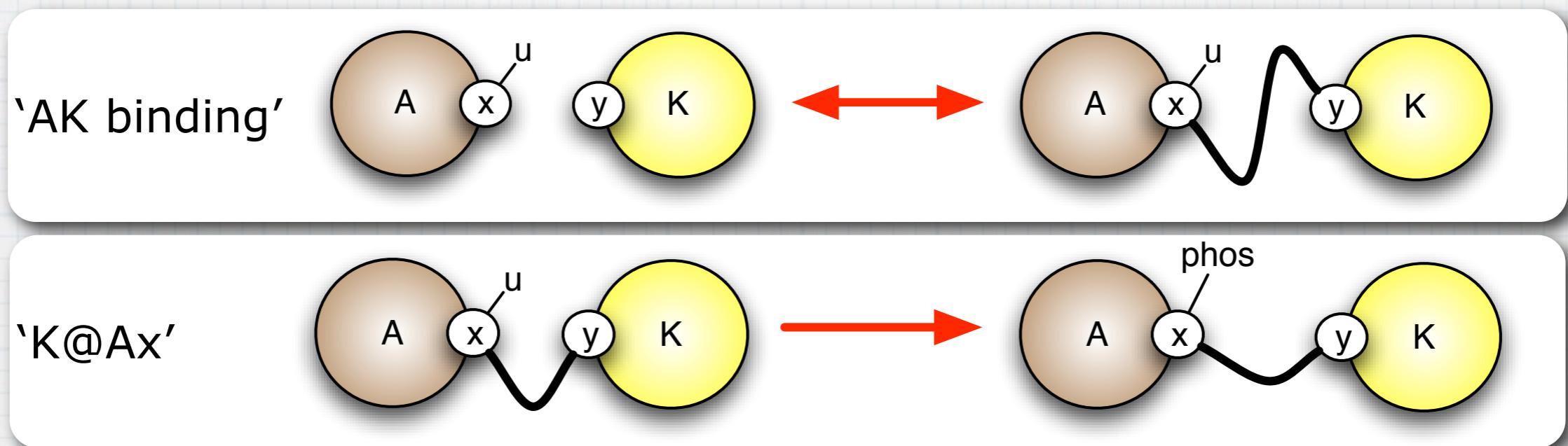
Programs as models: the ugly!

A model can be wrong because of a semantic mistake...



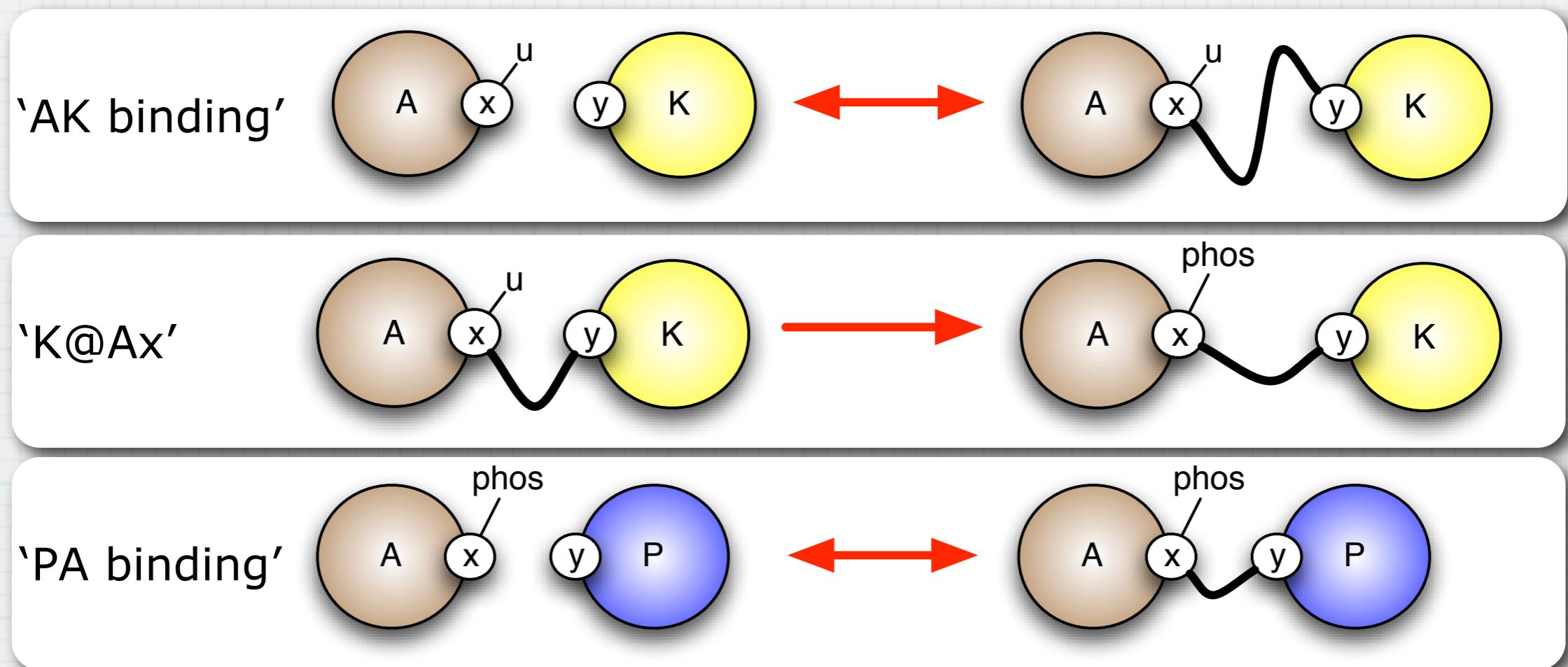
Programs as models: the ugly!

A model can be wrong because of a semantic mistake...



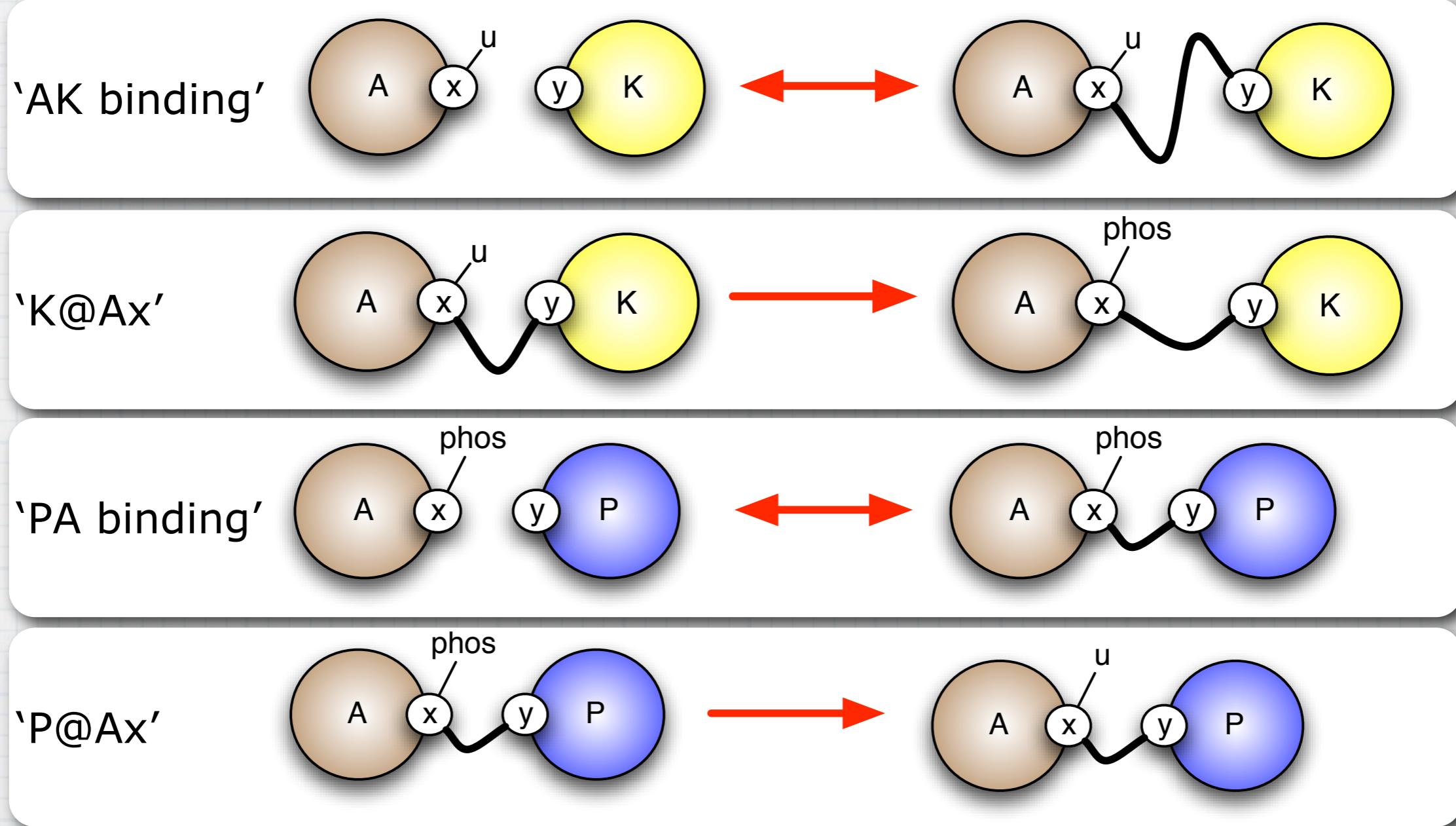
Programs as models: the ugly!

A model can be wrong because of a semantic mistake...



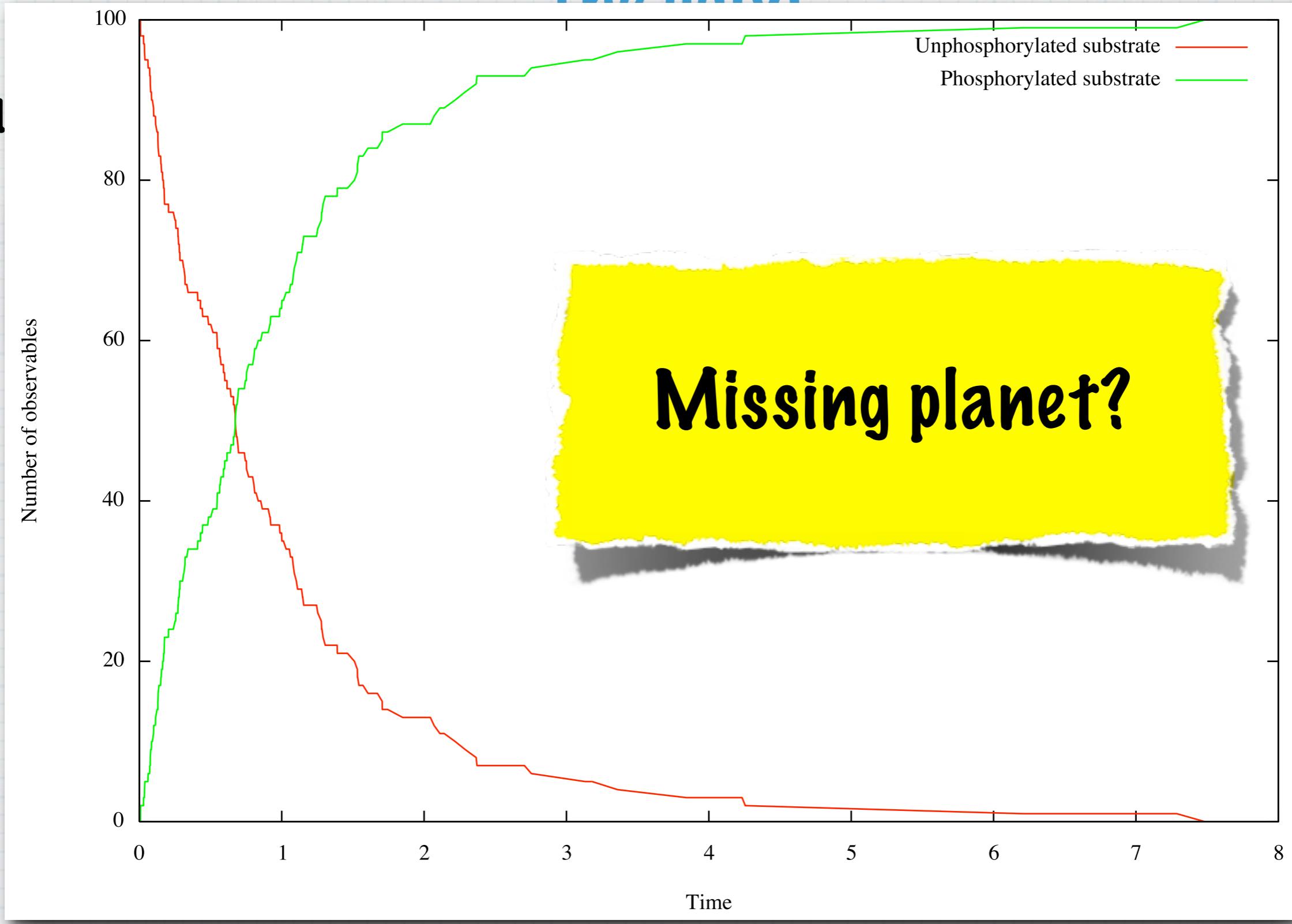
Programs as models: the ugly!

A model can be wrong because of a semantic mistake...



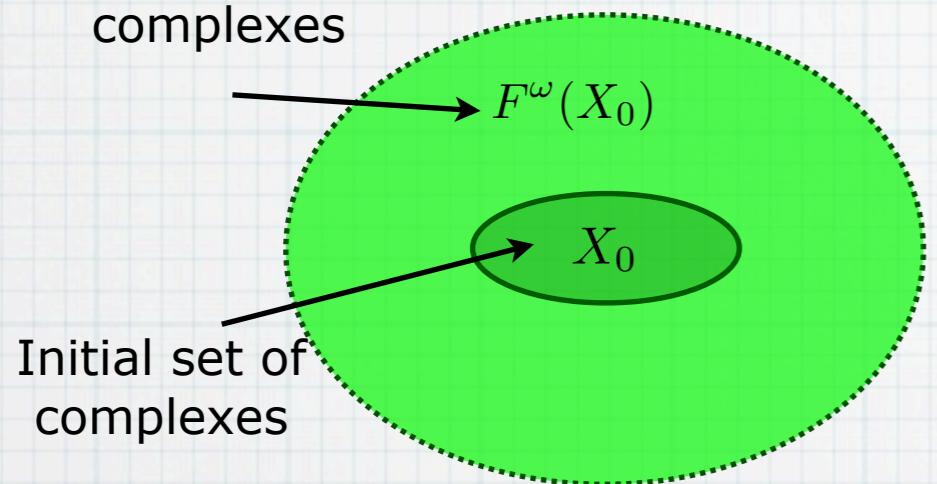
Programs as models: the value

Am



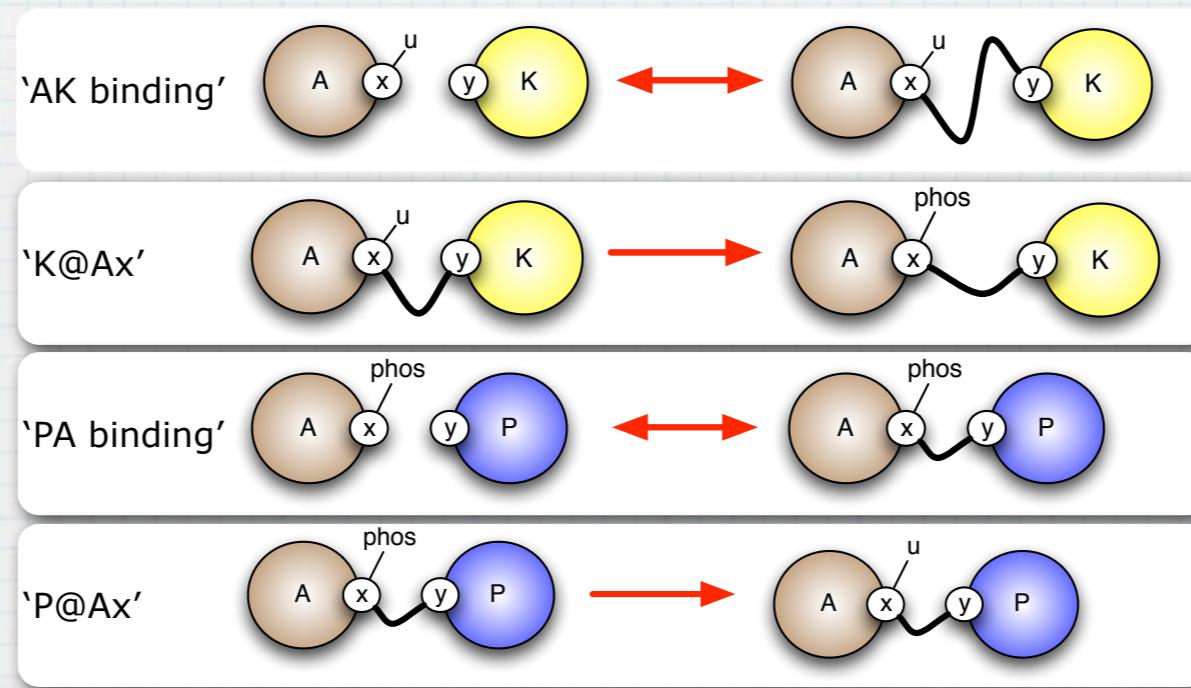
Exact solution...

Set of accessible complexes



$$F : X \rightarrow X$$

Accessibility function

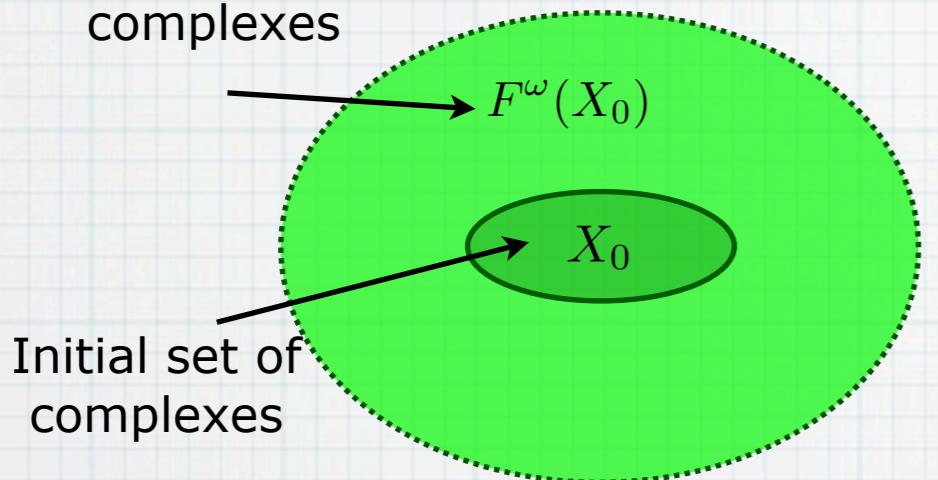


A rule is accessible if its left hand side matches an accessible complex

Exact solution...

... does not scale!

Set of accessible complexes



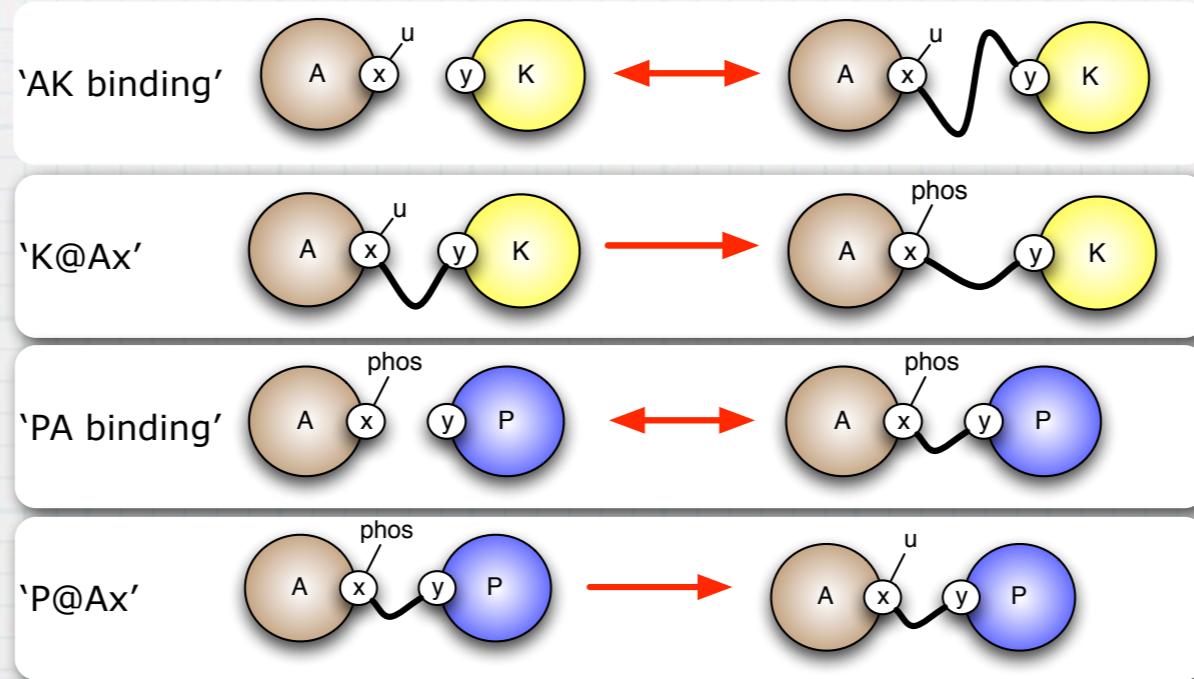
Initial set of complexes

Orders of magnitude

Early EGF: 400 complexes
EGF down to ERK: 80,000
EGF down to AKT: 500,000
EGF with receptor network: 10^{33}

$$F : X \rightarrow X$$

Accessibility function



A rule is accessible if its left hand side matches an accessible complex

Abstract interpretation

Correction: only correct answers are found

Completeness: all correct answers are found

Termination: program returns an answer in a reasonable time

Model Checking features

Abstract interpretation

Correctness: only correct answers are found

Completeness: all correct answers are found

Termination: program returns an answer in a reasonable time

Model Checking features

Abstract interpretation

Correction: only correct answers are found

Termination: program returns an answer in a reasonable time

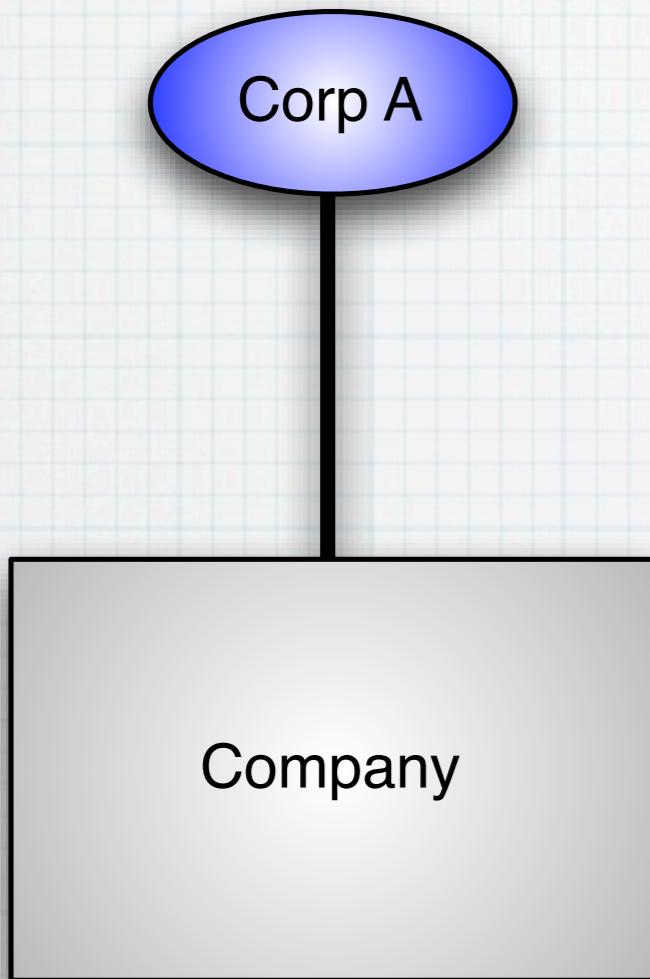
Abstract interpretation

Example



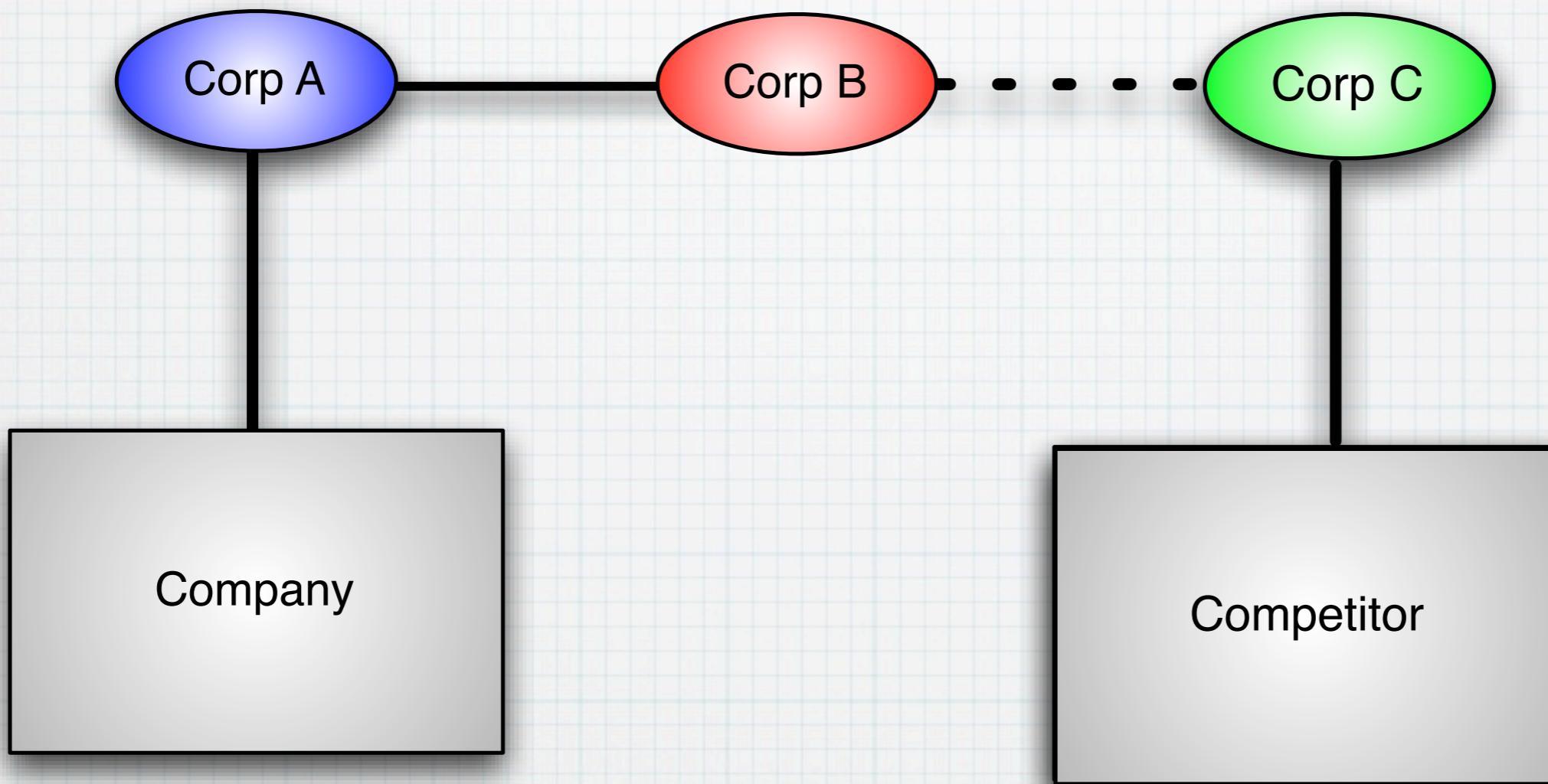
A company may sign deal with anonymous contractors

Example



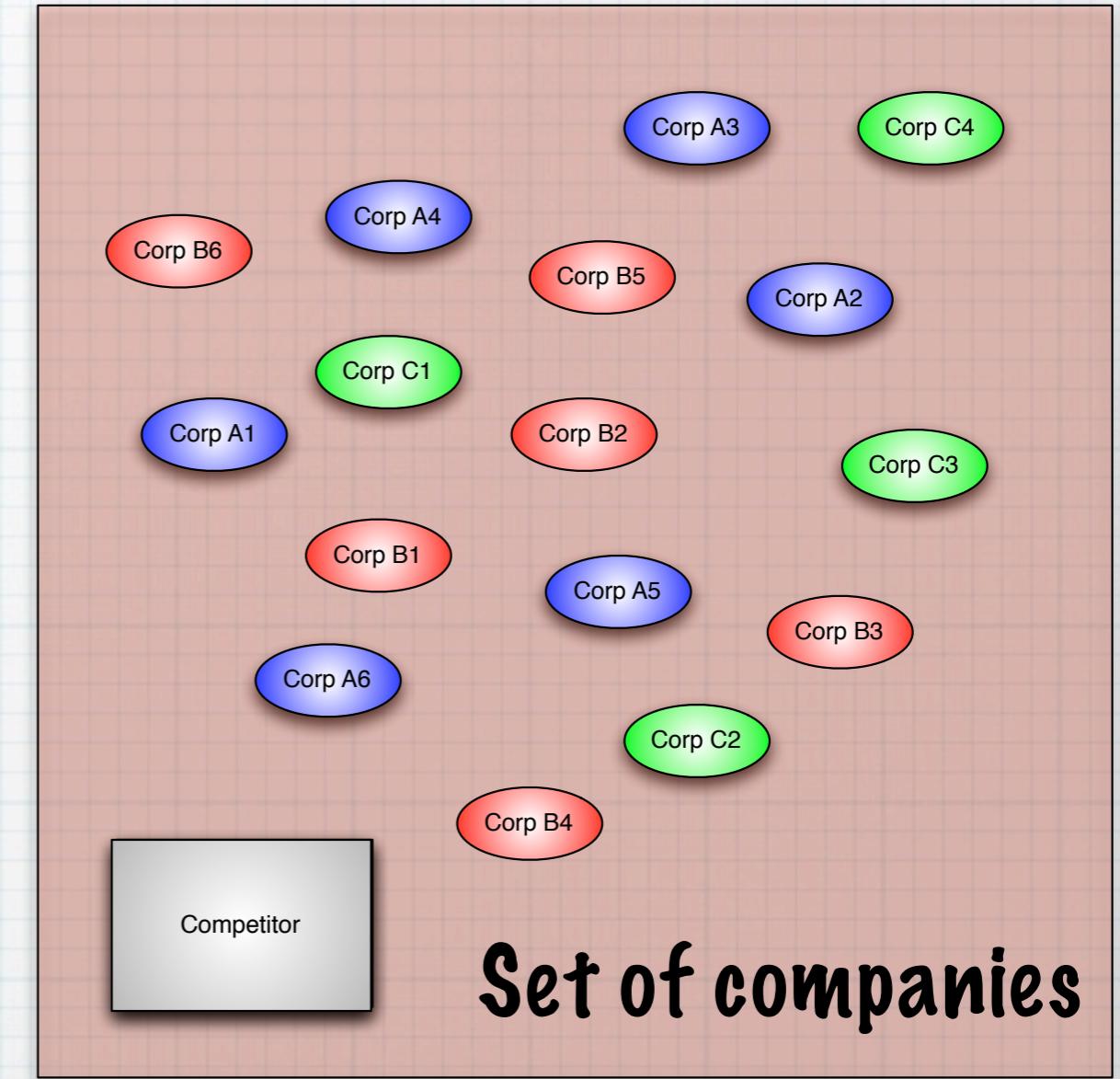
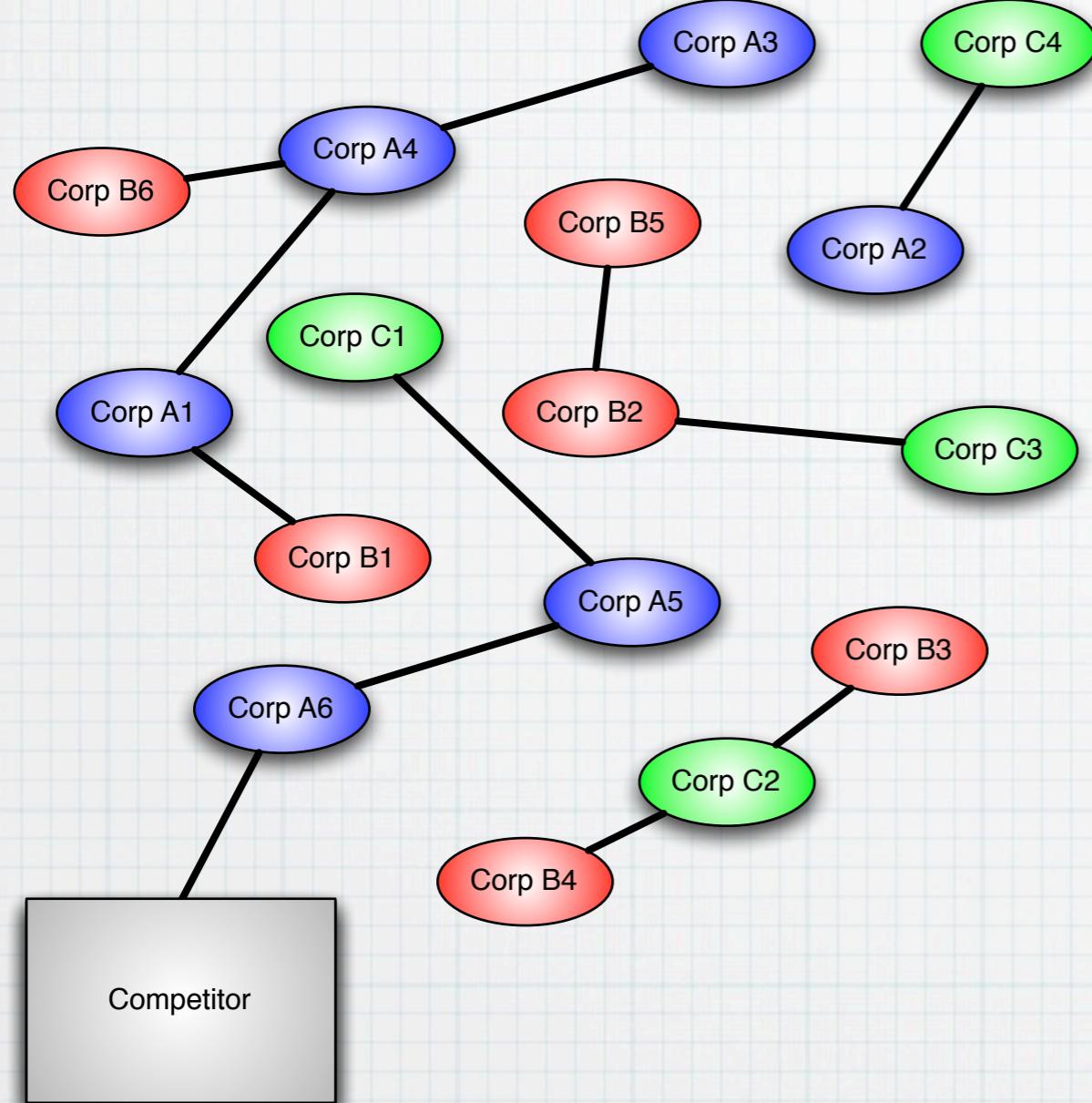
A company may sign deal with anonymous contractors

Example



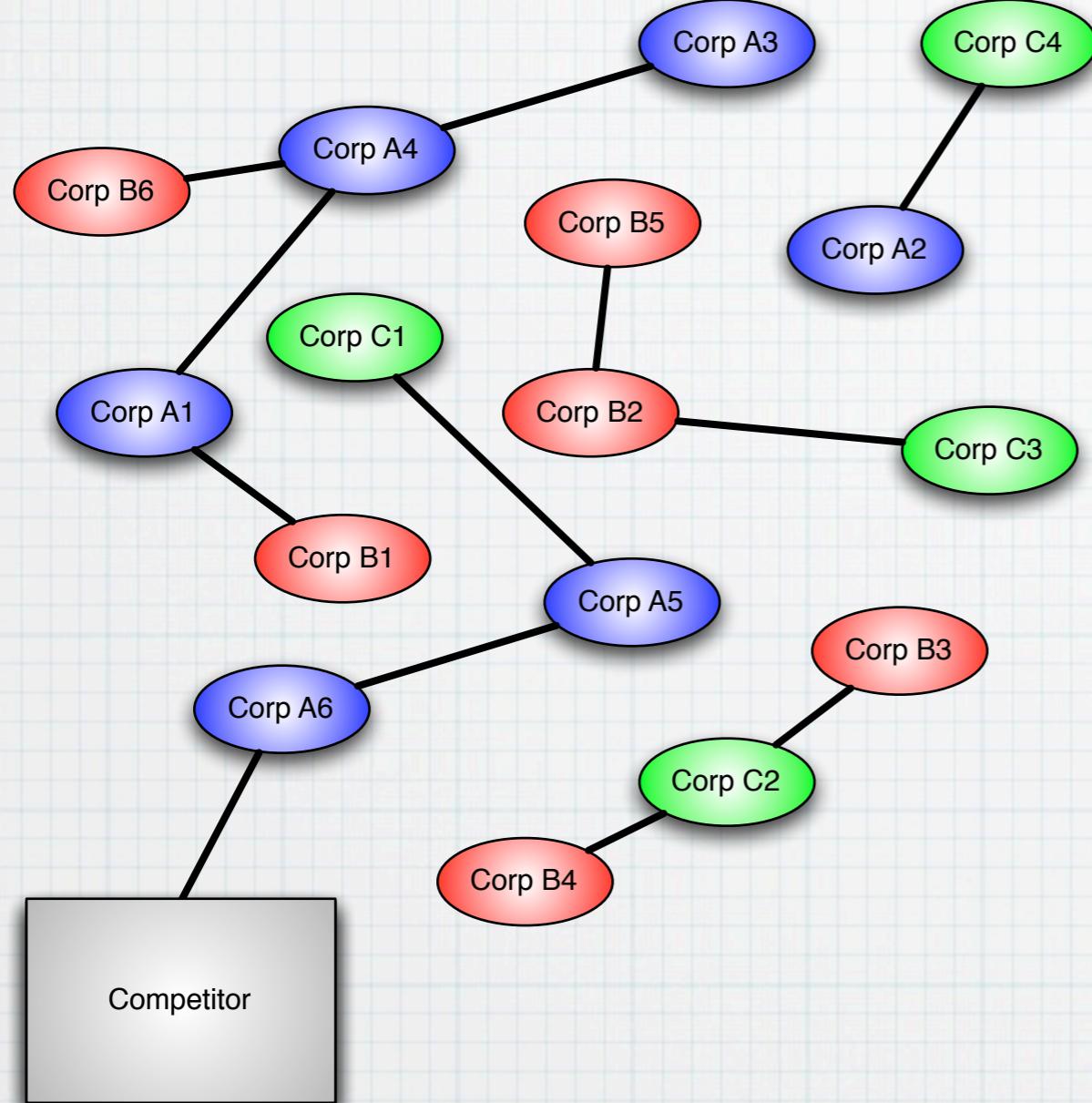
Which may be indirectly working for a competitor...

Model checking

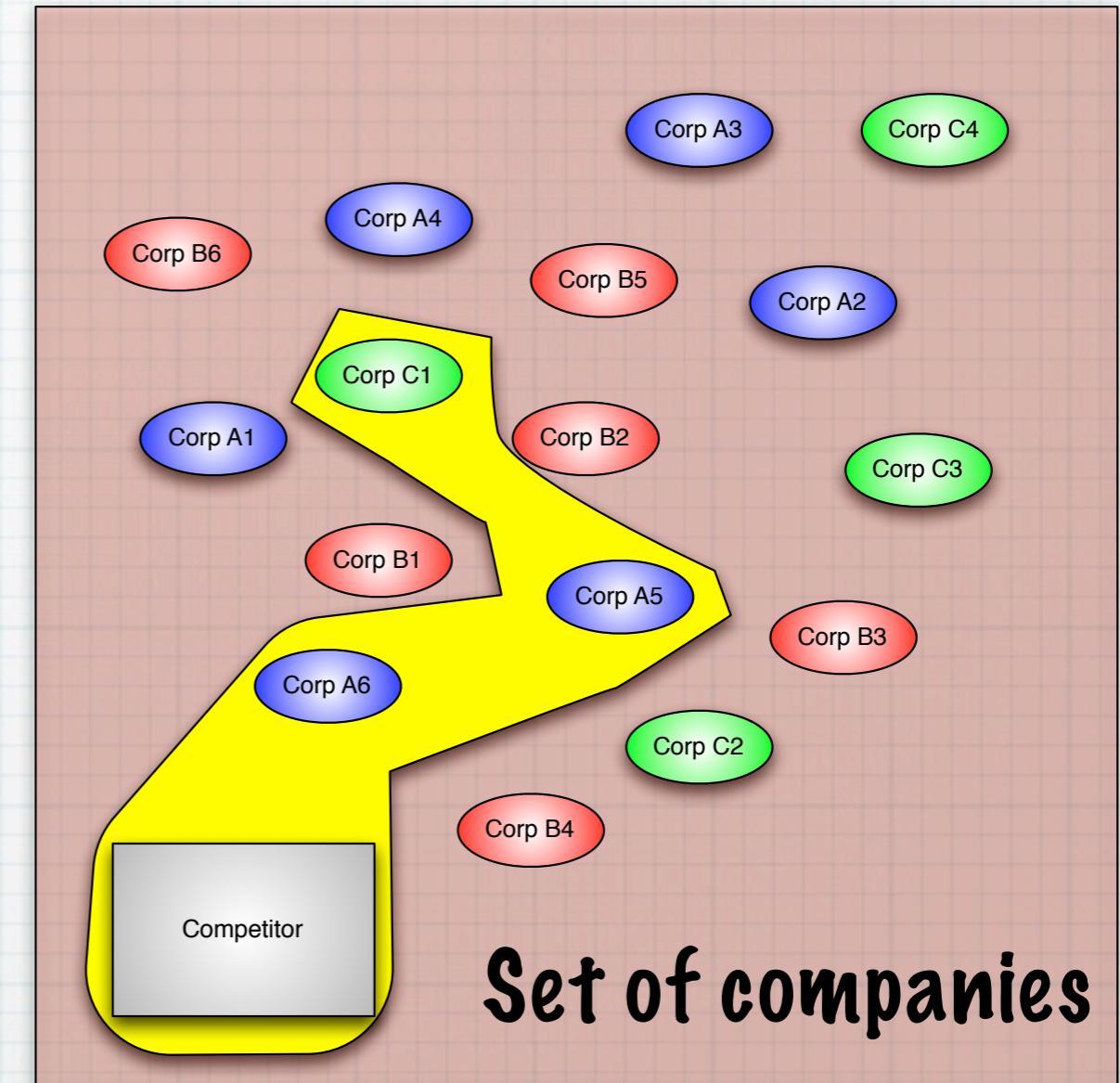


Fix point when one can
compute the exact
information

Model checking

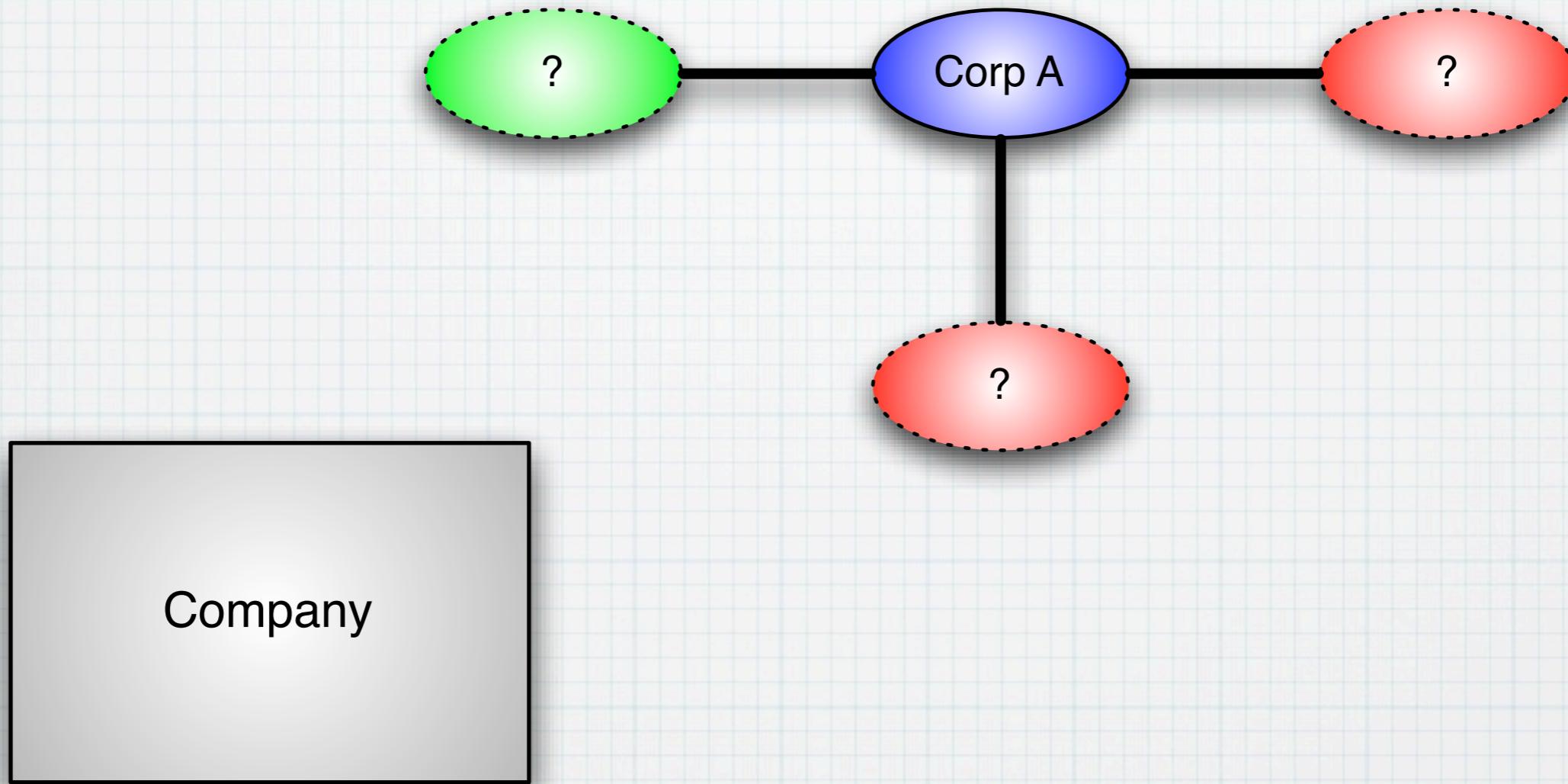


Fix point when one can
compute the exact
information



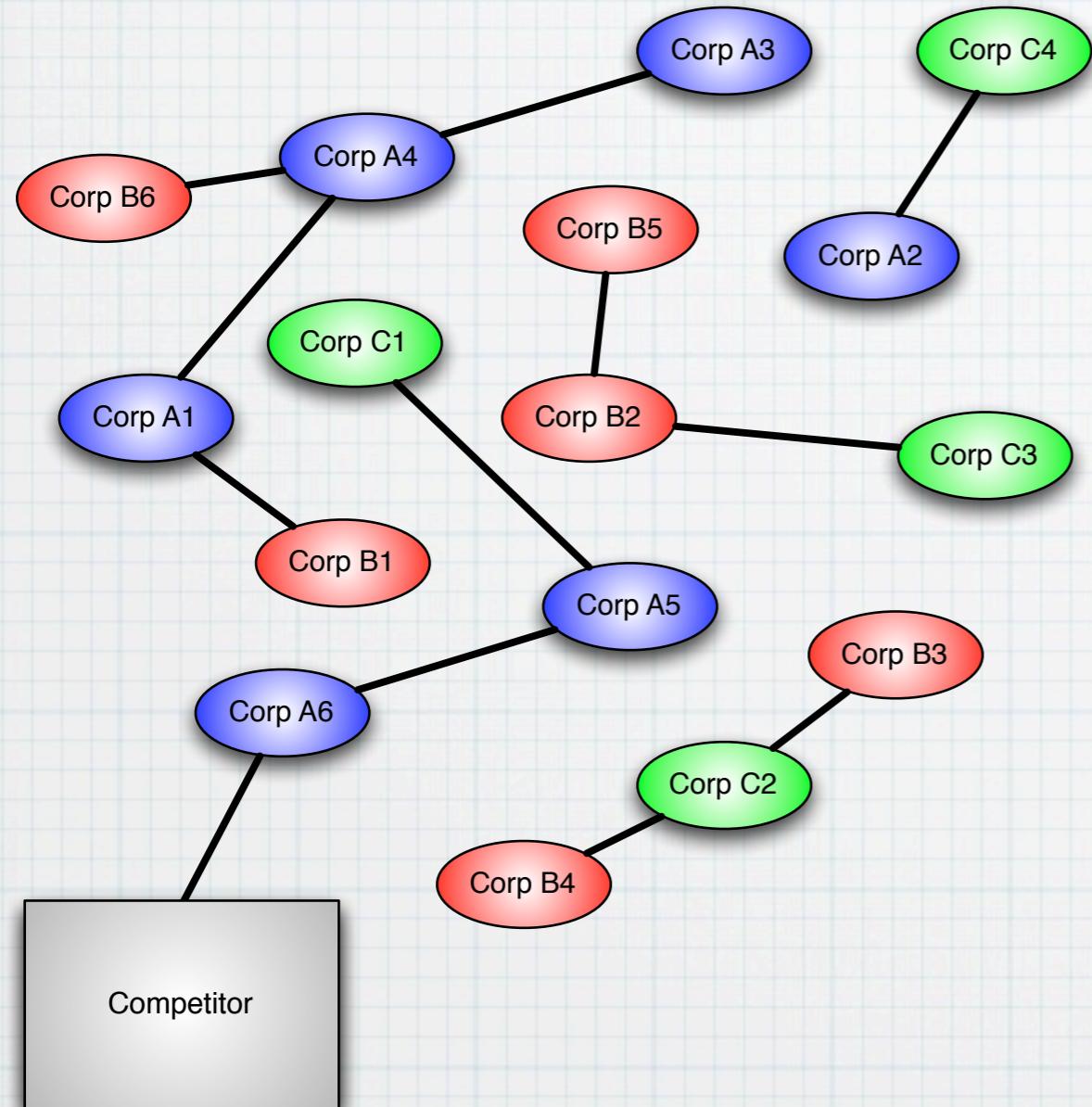
Exact set of companies you
don't want to deal with

Local information

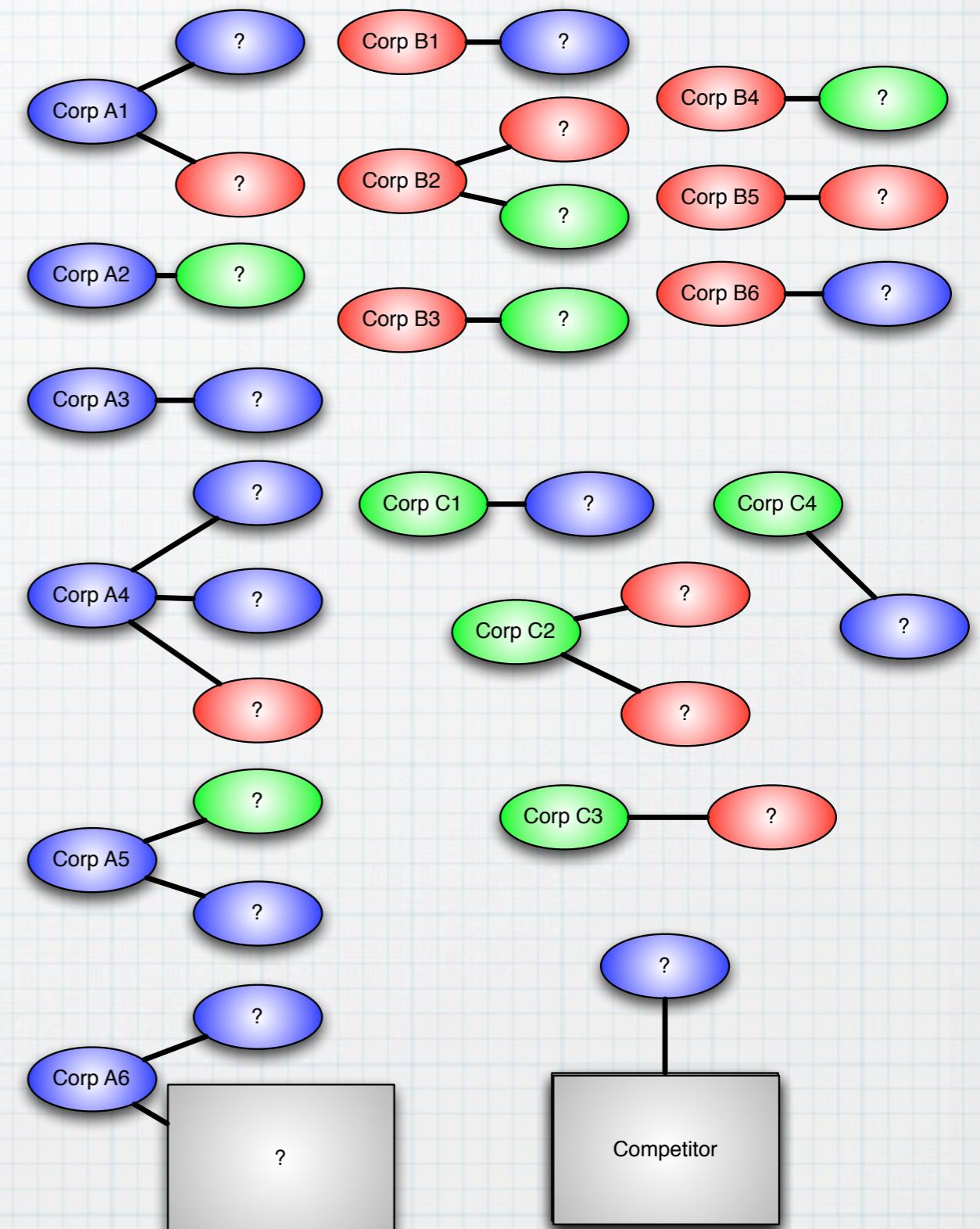


**For anonymity reasons, local information may be easier to obtain:
can I decide whether it is safe to deal with A?**

Abstraction

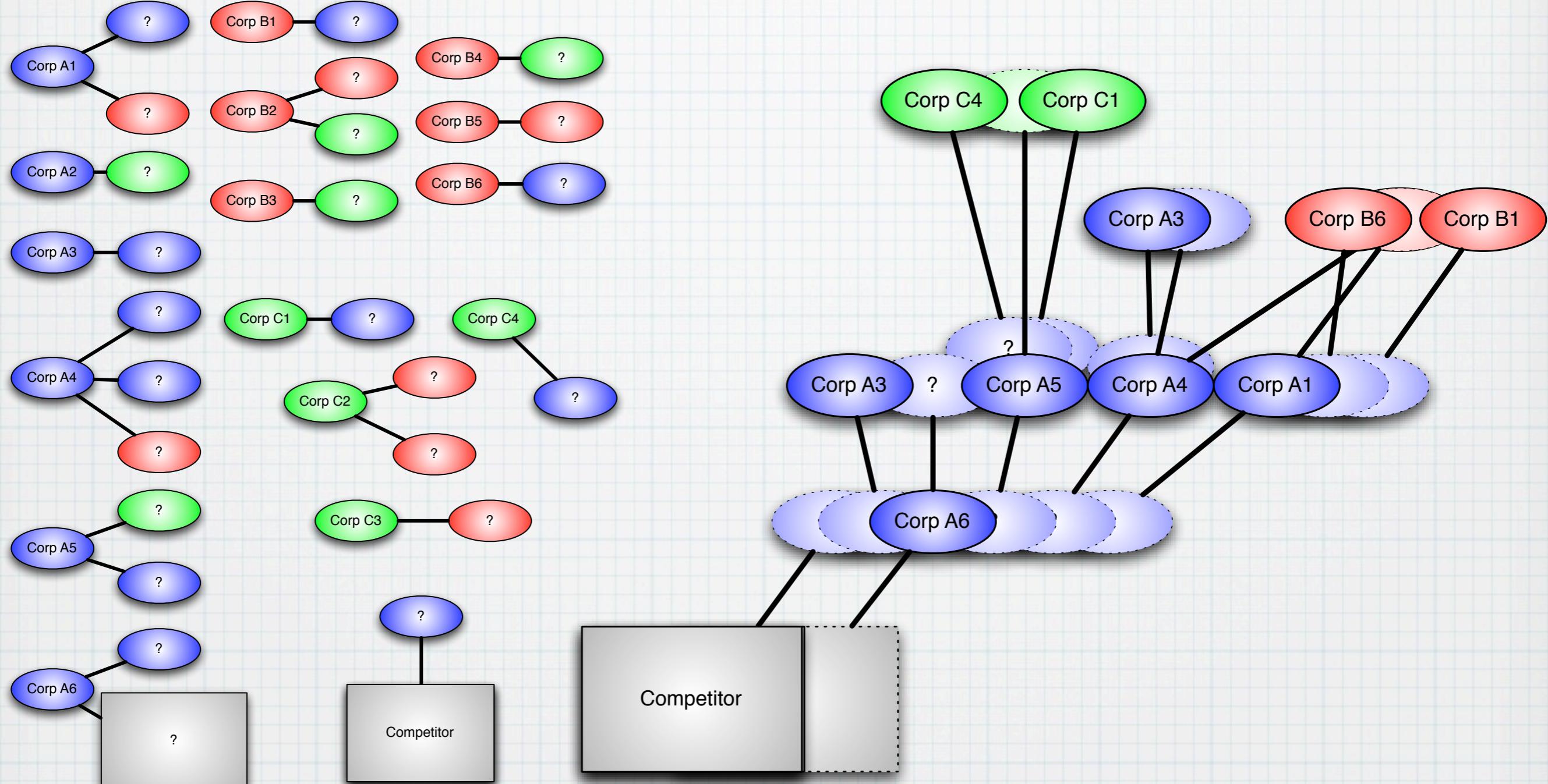


**Concrete fix point
(long)**



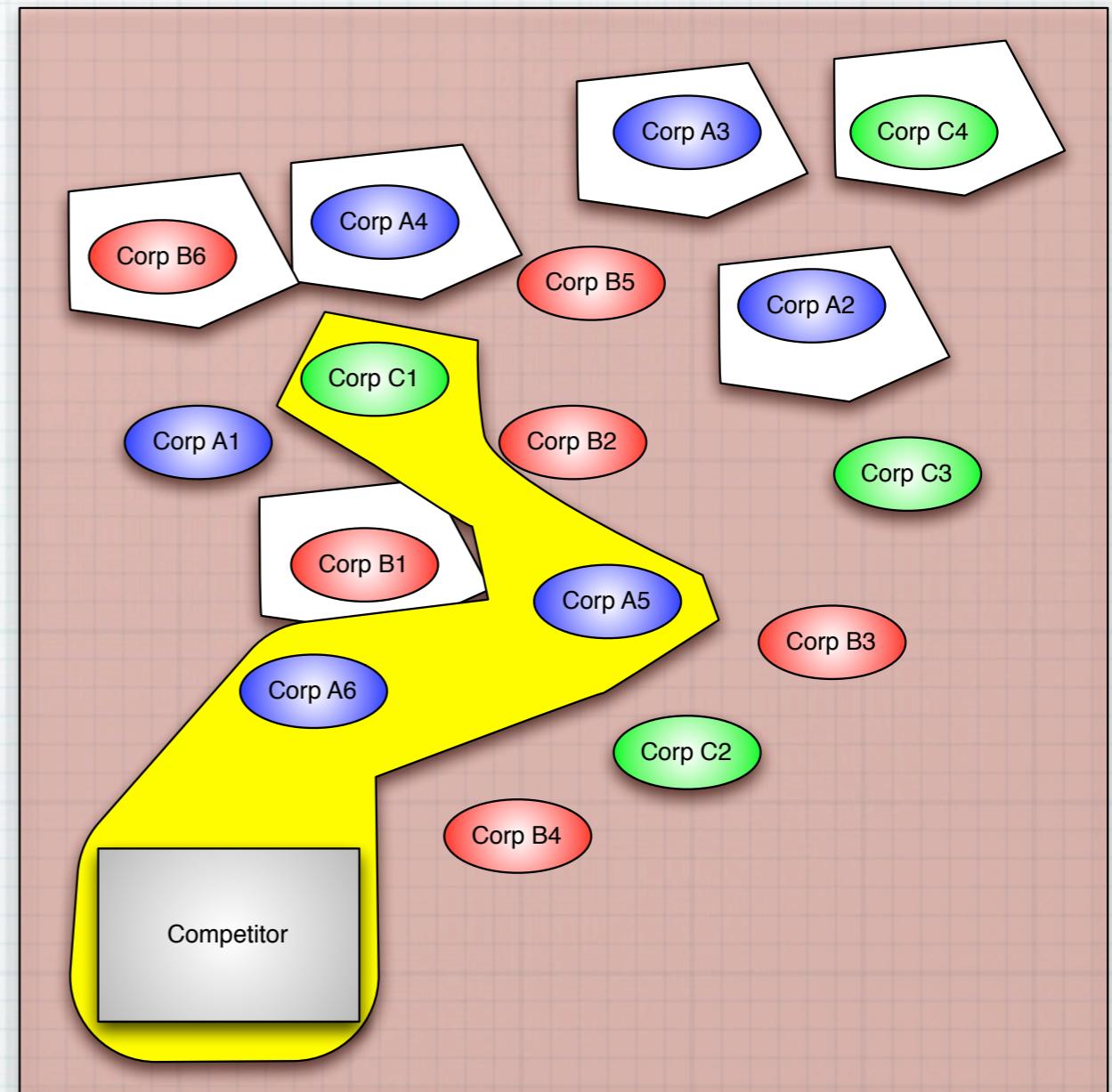
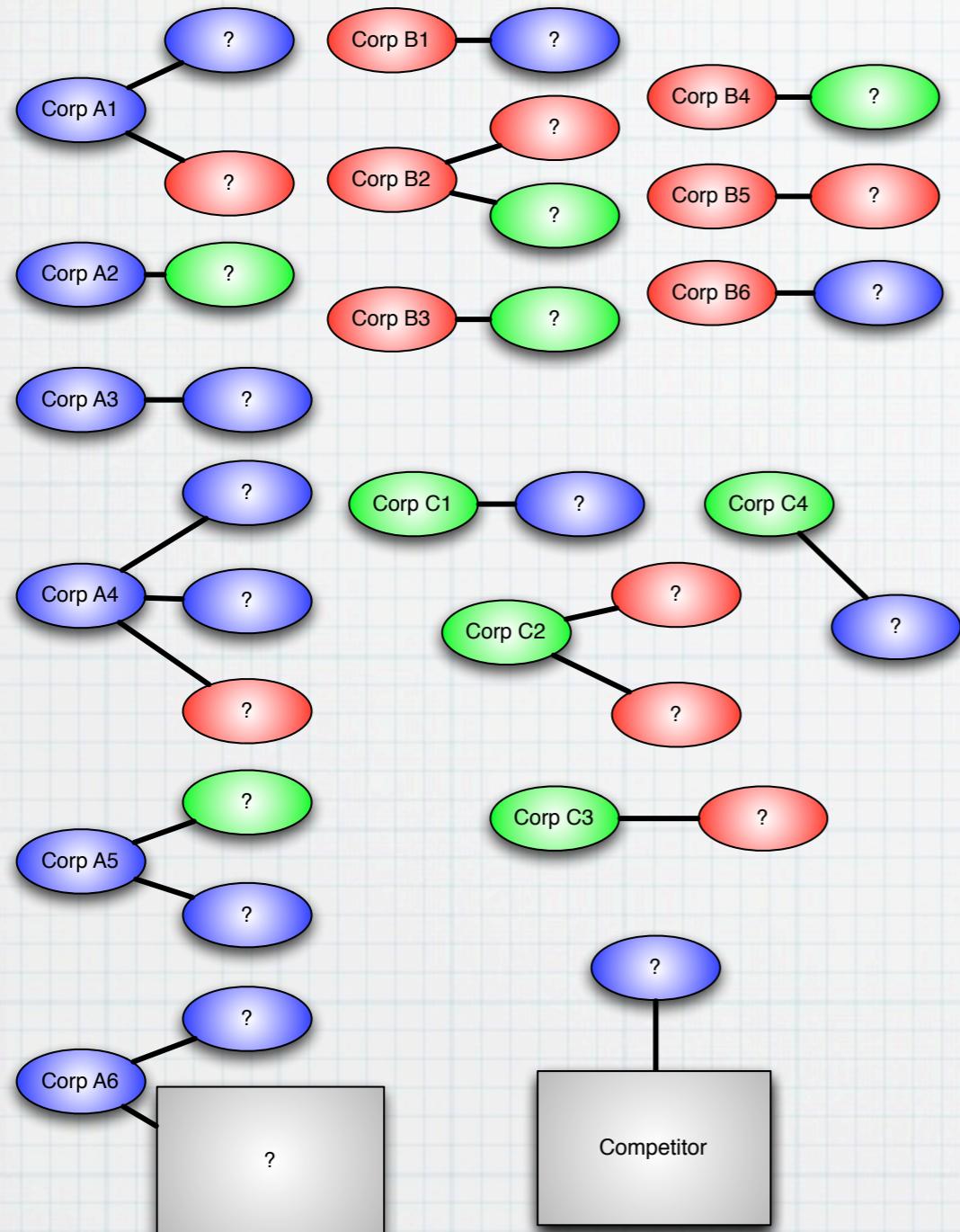
**Abstract fix point
(instantaneous)**

Concretization



One may compute an over approximation by
composing local views together

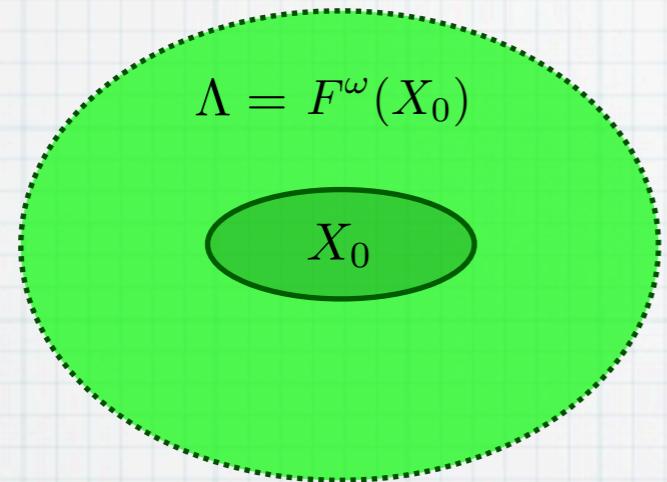
Over approximation



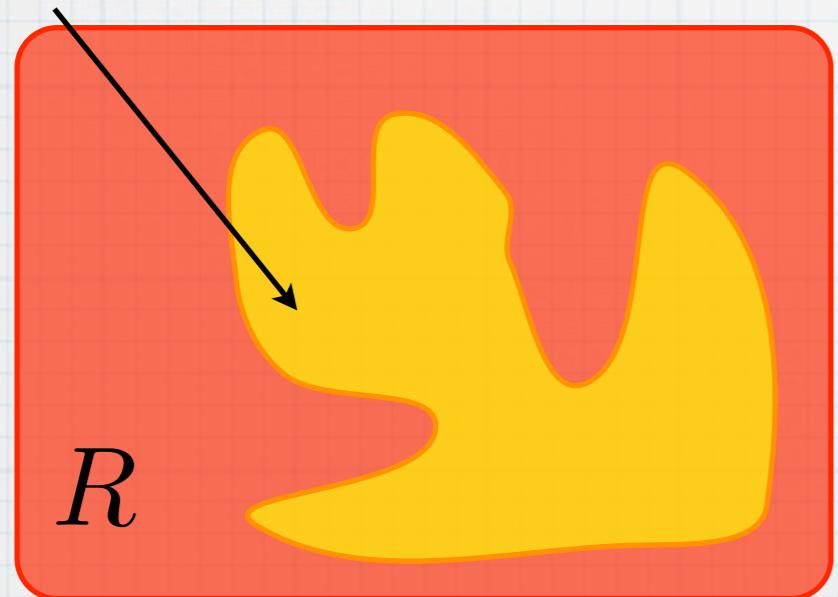
Concretization capture all untrusted companies
and some others (false positive)

Back to Kappa...

$$F : X \rightarrow X$$



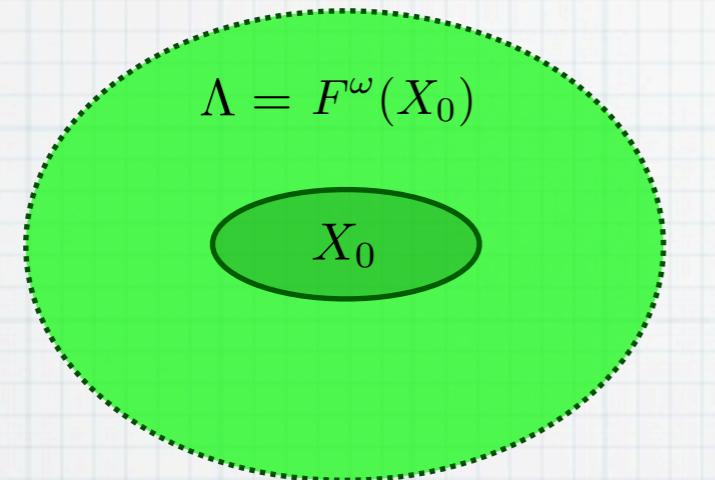
$$\{r \in R \mid P_\Lambda(r) = 1\}$$



$$\begin{aligned} P_\Lambda(r) &= 1 \text{ iff } \exists \varphi, x \in \Lambda \text{ st. } \text{lhs}(r) \xrightarrow{\varphi} x \\ P_\Lambda(r) &= 0 \text{ else} \end{aligned}$$

Back to Kappa...

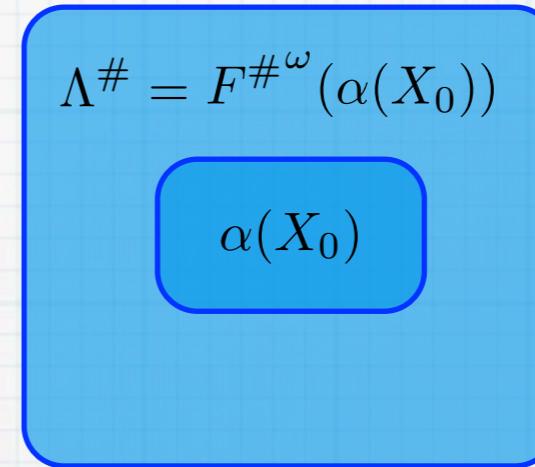
$$F : X \rightarrow X$$



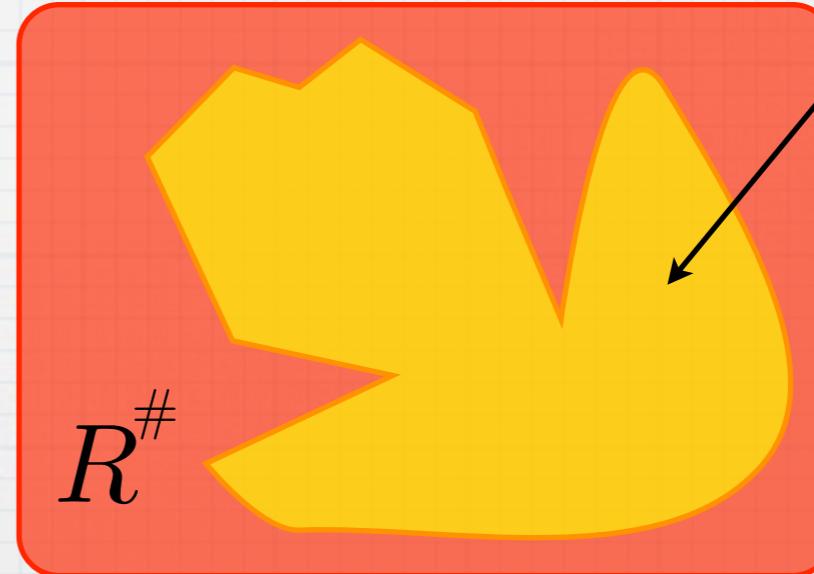
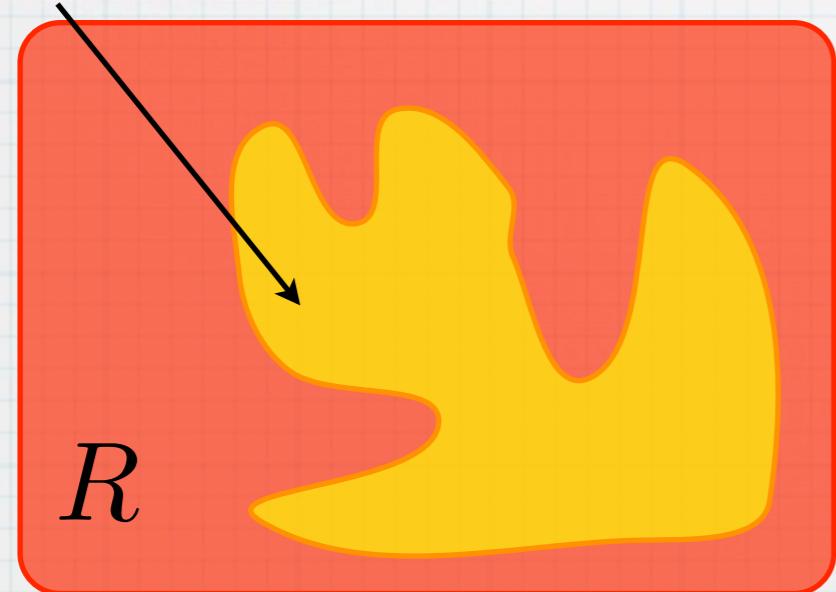
$$\{r \in R \mid P_\Lambda(r) = 1\}$$

$$\xrightarrow{\alpha} \quad \xleftarrow{\gamma}$$

$$F^\# : X^\# \rightarrow X^\#$$



$$\{r \in R \mid P_\Lambda^\#(r) = 1\}$$

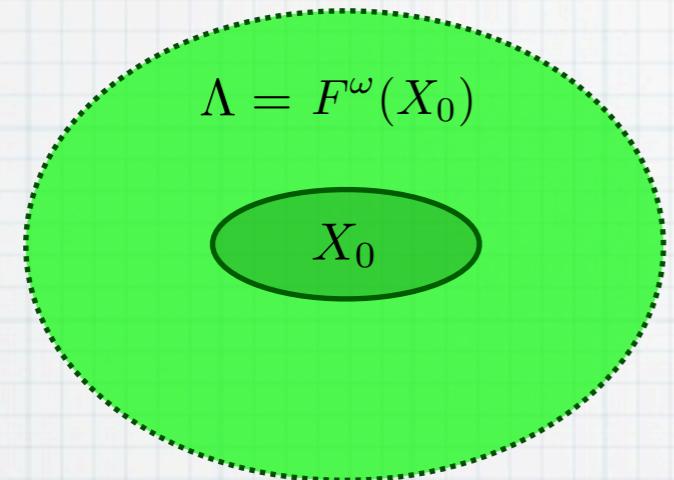


$$\begin{aligned} P_\Lambda(r) &= 1 \text{ iff } \exists \varphi, x \in \Lambda \text{ st. } \text{lhs}(r) \stackrel{\varphi}{\hookrightarrow} x \\ P_\Lambda(r) &= 0 \text{ else} \end{aligned}$$

$$\begin{aligned} P_{\Lambda^\#}(r) &= 1 \text{ iff } \exists \varphi, x \in \Lambda^\# \text{ st. } \text{lhs}(r) \stackrel{\varphi}{\hookrightarrow} \gamma(x) \\ P_{\Lambda^\#}(r) &= 0 \text{ else} \end{aligned}$$

Back to Kappa...

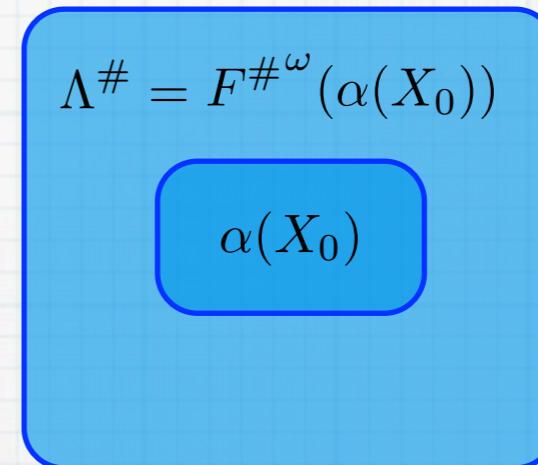
$$F : X \rightarrow X$$



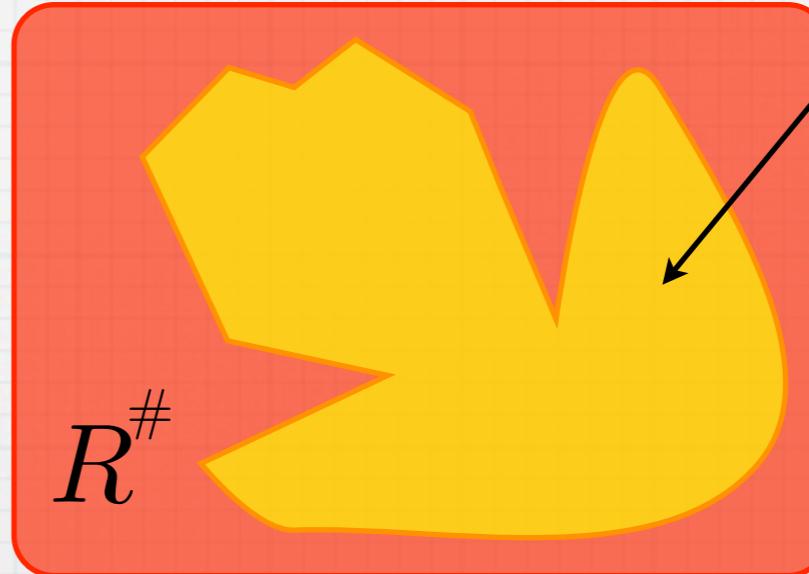
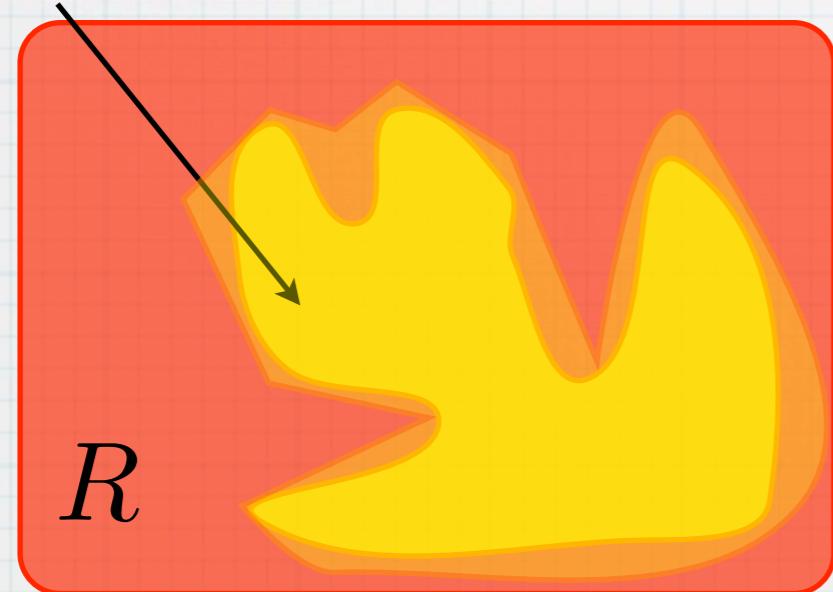
$$\{r \in R \mid P_\Lambda(r) = 1\}$$

$$\xrightarrow{\alpha} \quad \xleftarrow{\gamma}$$

$$F^\# : X^\# \rightarrow X^\#$$



$$\{r \in R \mid P_\Lambda^\#(r) = 1\}$$

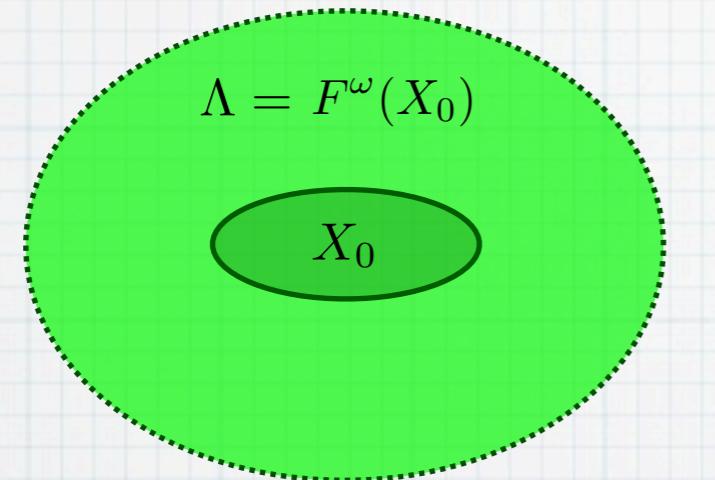


$$\begin{aligned} P_\Lambda(r) &= 1 \text{ iff } \exists \varphi, x \in \Lambda \text{ st. } \text{lhs}(r) \stackrel{\varphi}{\hookrightarrow} x \\ P_\Lambda(r) &= 0 \text{ else} \end{aligned}$$

$$\begin{aligned} P_{\Lambda^\#}(r) &= 1 \text{ iff } \exists \varphi, x \in \Lambda^\# \text{ st. } \text{lhs}(r) \stackrel{\varphi}{\hookrightarrow} \gamma(x) \\ P_{\Lambda^\#}(r) &= 0 \text{ else} \end{aligned}$$

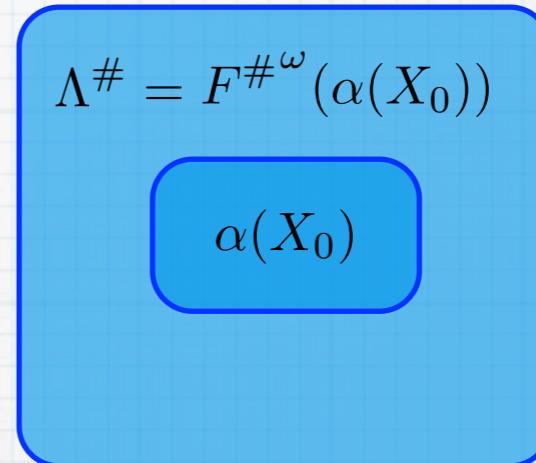
Back to Kappa...

$$F : X \rightarrow X$$



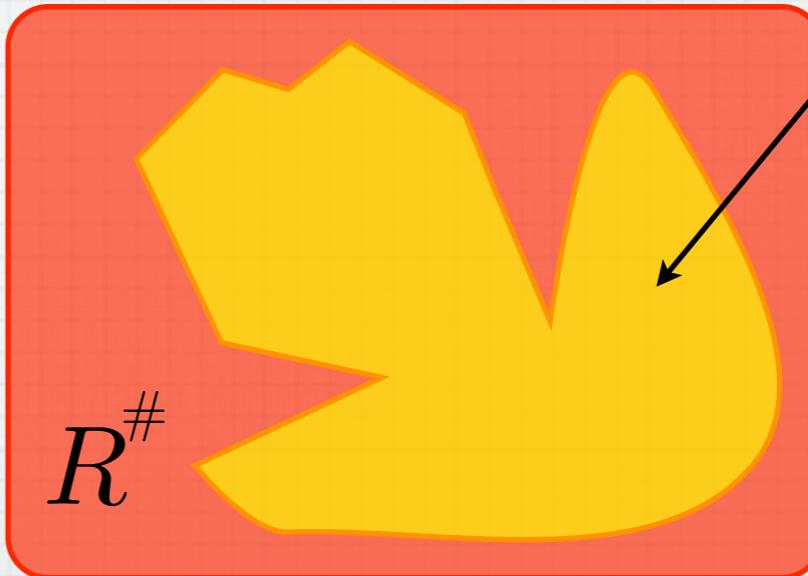
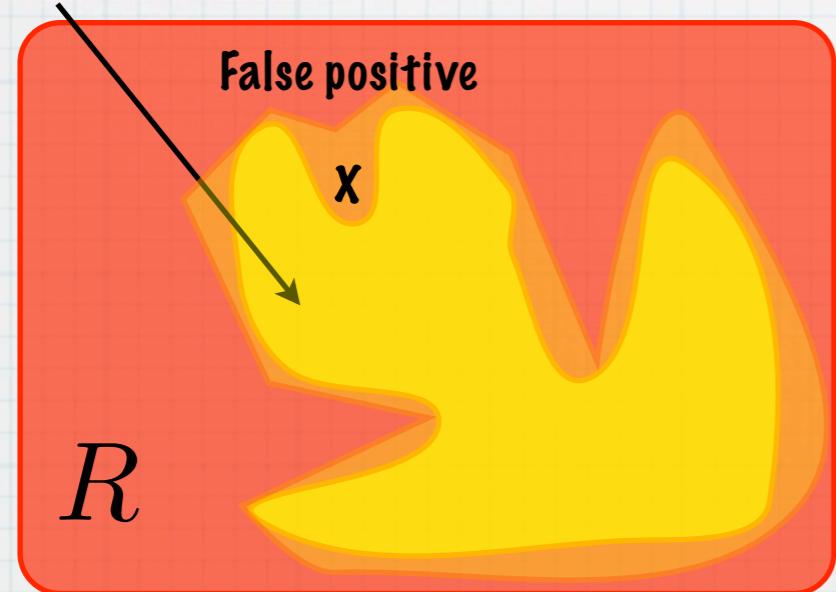
$$\xrightarrow{\alpha} \quad \xleftarrow{\gamma}$$

$$F^\# : X^\# \rightarrow X^\#$$



$$\{r \in R \mid P_\Lambda(r) = 1\}$$

$$\{r \in R \mid P_\Lambda^\#(r) = 1\}$$

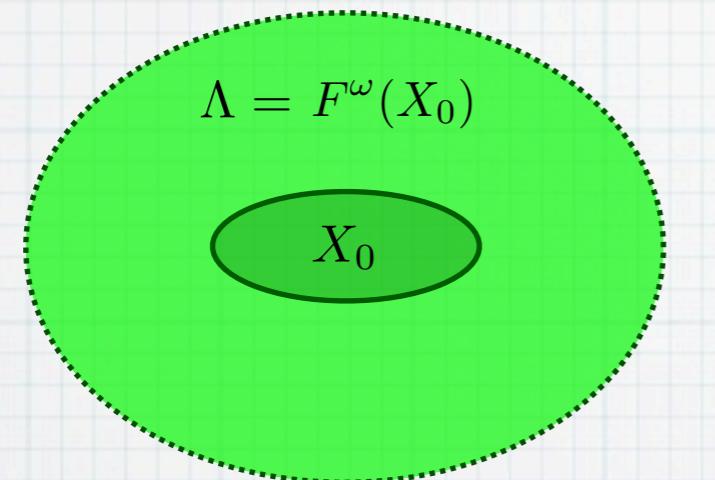


$$\begin{aligned} P_\Lambda(r) &= 1 \text{ iff } \exists \varphi, x \in \Lambda \text{ st. } \text{lhs}(r) \stackrel{\varphi}{\hookrightarrow} x \\ P_\Lambda(r) &= 0 \text{ else} \end{aligned}$$

$$\begin{aligned} P_{\Lambda^\#}(r) &= 1 \text{ iff } \exists \varphi, x \in \Lambda^\# \text{ st. } \text{lhs}(r) \stackrel{\varphi}{\hookrightarrow} \gamma(x) \\ P_{\Lambda^\#}(r) &= 0 \text{ else} \end{aligned}$$

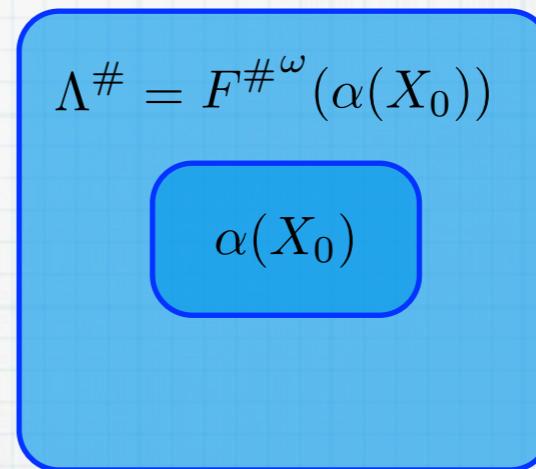
Back to Kappa...

$$F : X \rightarrow X$$

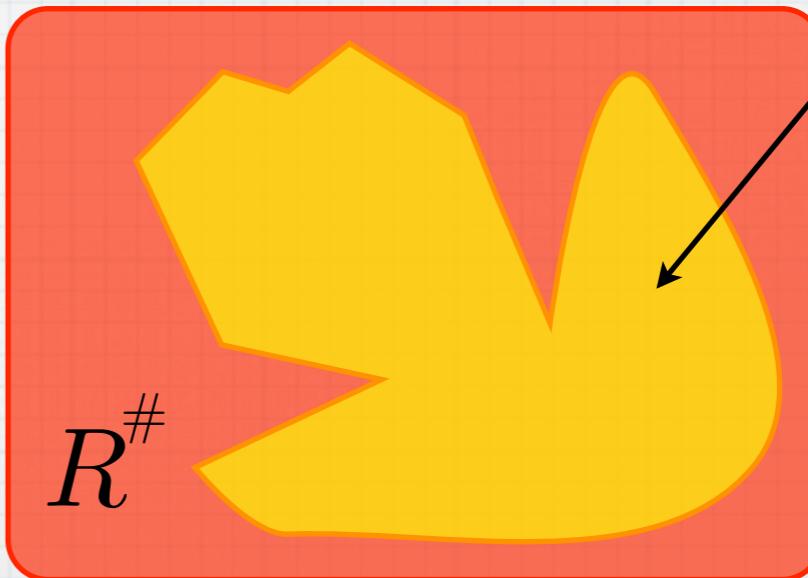
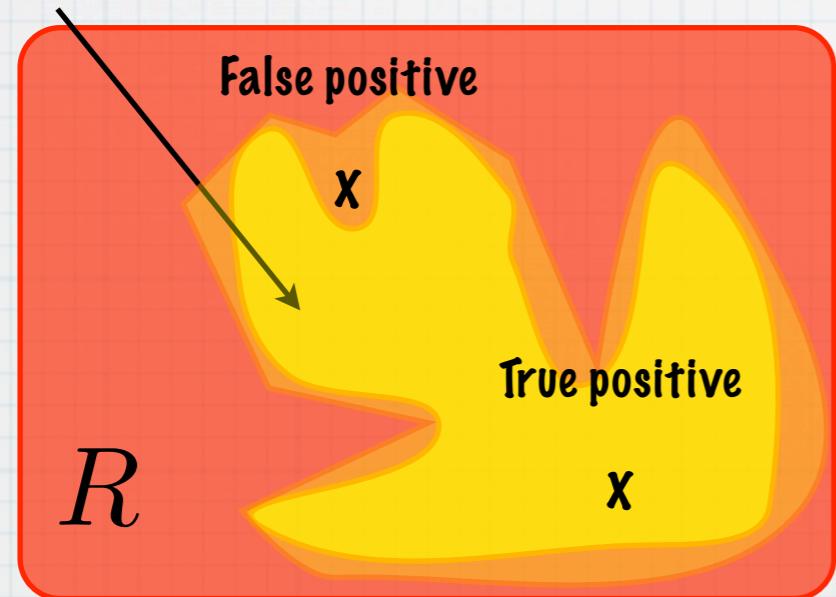


$$\{r \in R \mid P_\Lambda(r) = 1\}$$

$$F^\# : X^\# \rightarrow X^\#$$



$$\{r \in R \mid P_\Lambda^\#(r) = 1\}$$

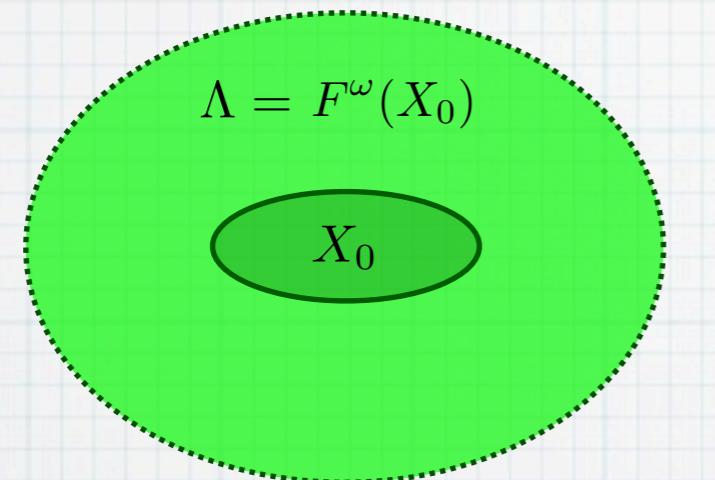


$$\begin{aligned} P_\Lambda(r) &= 1 \text{ iff } \exists \varphi, x \in \Lambda \text{ st. } \text{lhs}(r) \stackrel{\varphi}{\hookrightarrow} x \\ P_\Lambda(r) &= 0 \text{ else} \end{aligned}$$

$$\begin{aligned} P_{\Lambda^\#}(r) &= 1 \text{ iff } \exists \varphi, x \in \Lambda^\# \text{ st. } \text{lhs}(r) \stackrel{\varphi}{\hookrightarrow} \gamma(x) \\ P_{\Lambda^\#}(r) &= 0 \text{ else} \end{aligned}$$

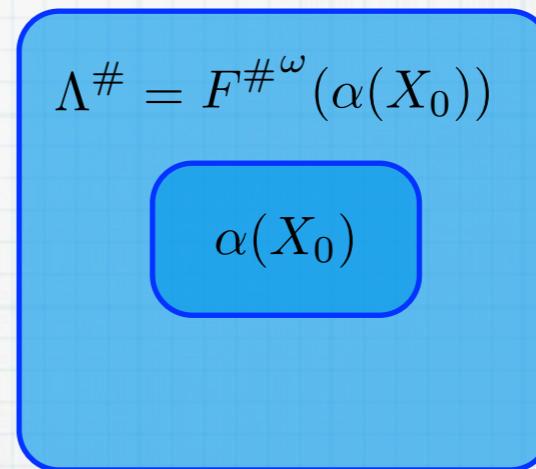
Back to Kappa...

$$F : X \rightarrow X$$

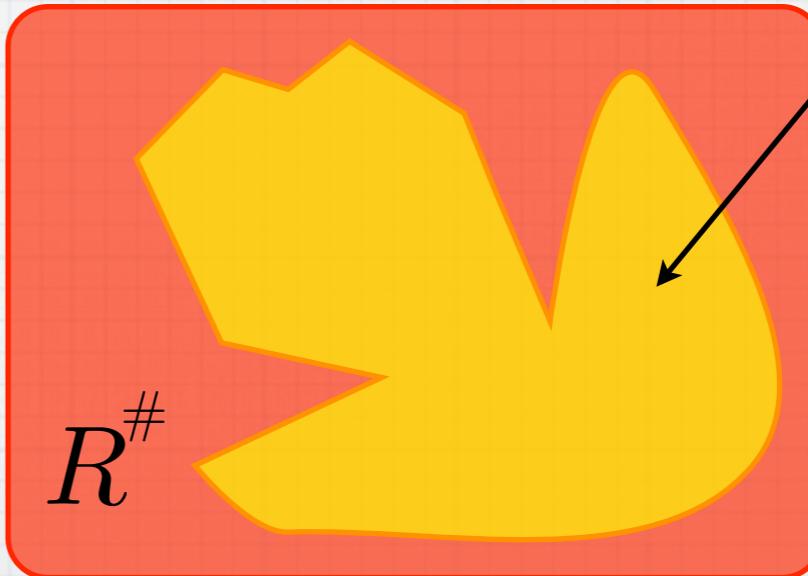
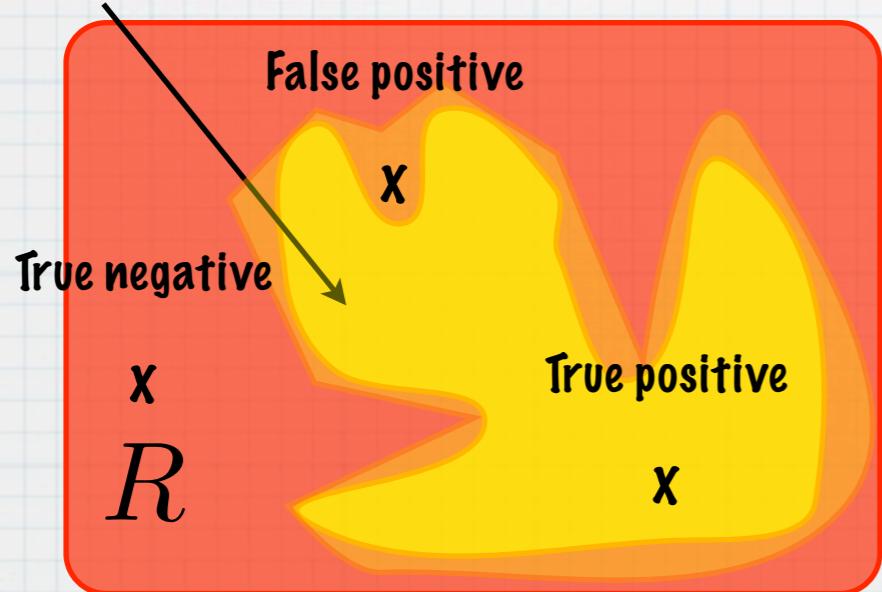


$$\{r \in R \mid P_\Lambda(r) = 1\}$$

$$F^\# : X^\# \rightarrow X^\#$$



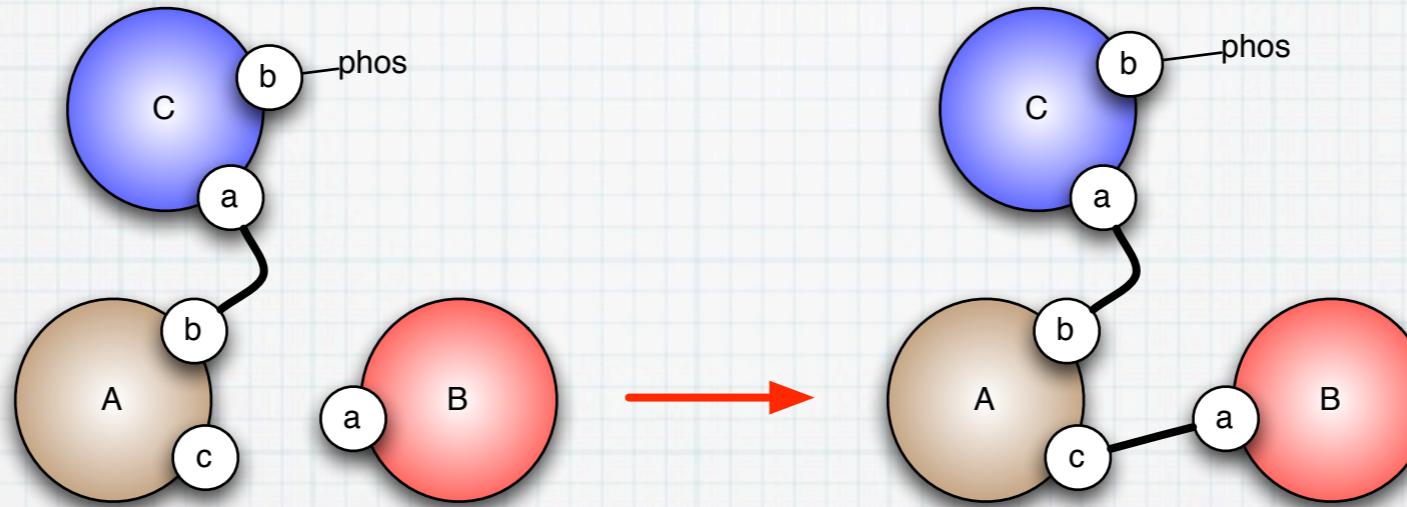
$$\{r \in R \mid P_{\Lambda^\#}(r) = 1\}$$



$$\begin{aligned} P_\Lambda(r) &= 1 \text{ iff } \exists \varphi, x \in \Lambda \text{ st. } \text{lhs}(r) \stackrel{\varphi}{\hookrightarrow} x \\ P_\Lambda(r) &= 0 \text{ else} \end{aligned}$$

$$\begin{aligned} P_{\Lambda^\#}(r) &= 1 \text{ iff } \exists \varphi, x \in \Lambda^\# \text{ st. } \text{lhs}(r) \stackrel{\varphi}{\hookrightarrow} \gamma(x) \\ P_{\Lambda^\#}(r) &= 0 \text{ else} \end{aligned}$$

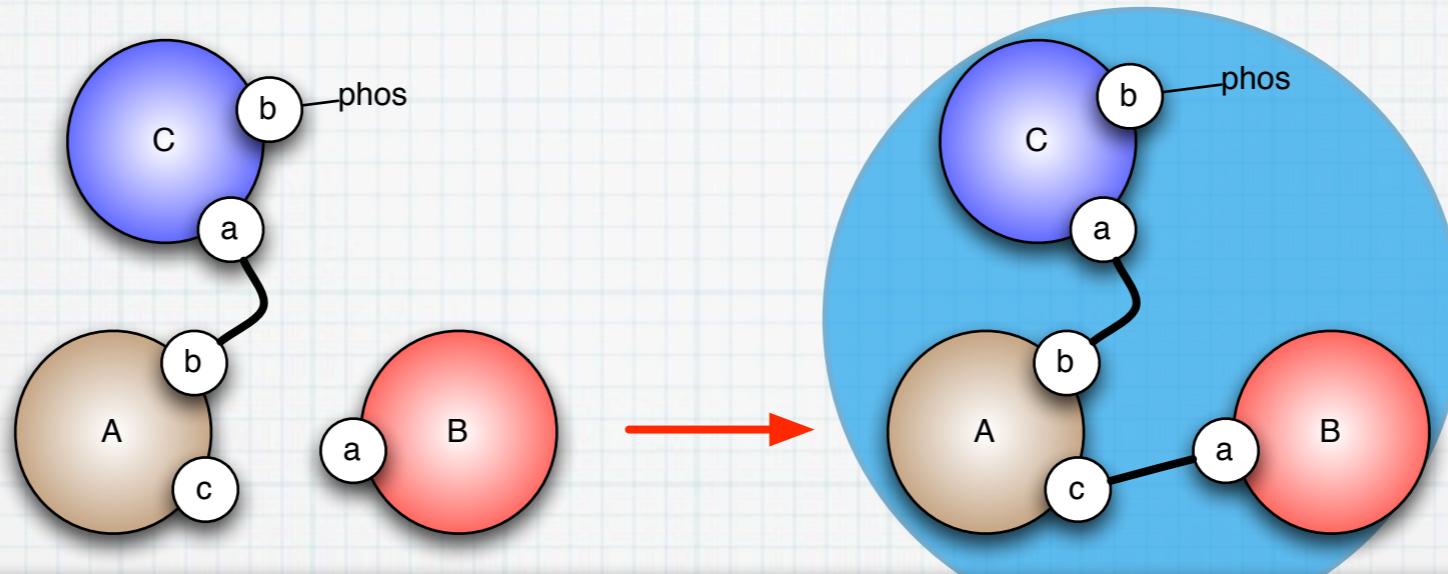
Local views



X_n : accessible complexes computed so far

$F(X_n) = X_{n+1}$: apply rule to complexes in X_n and add new complexes

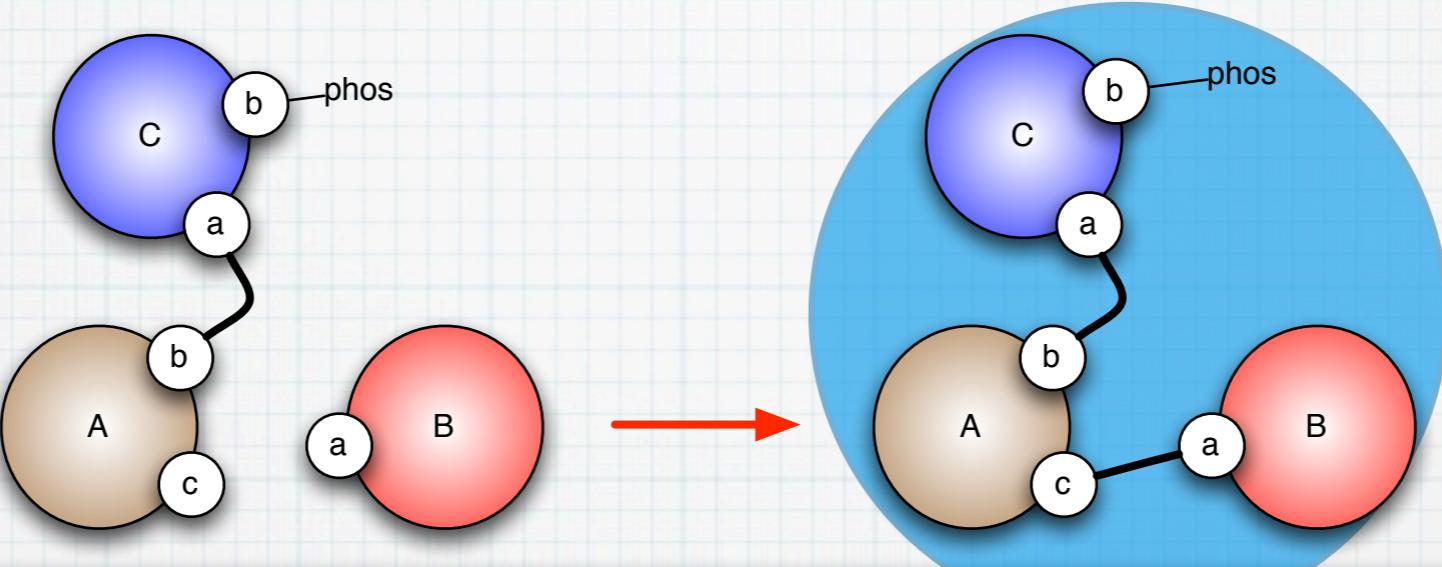
Local views



X_n : accessible complexes computed so far

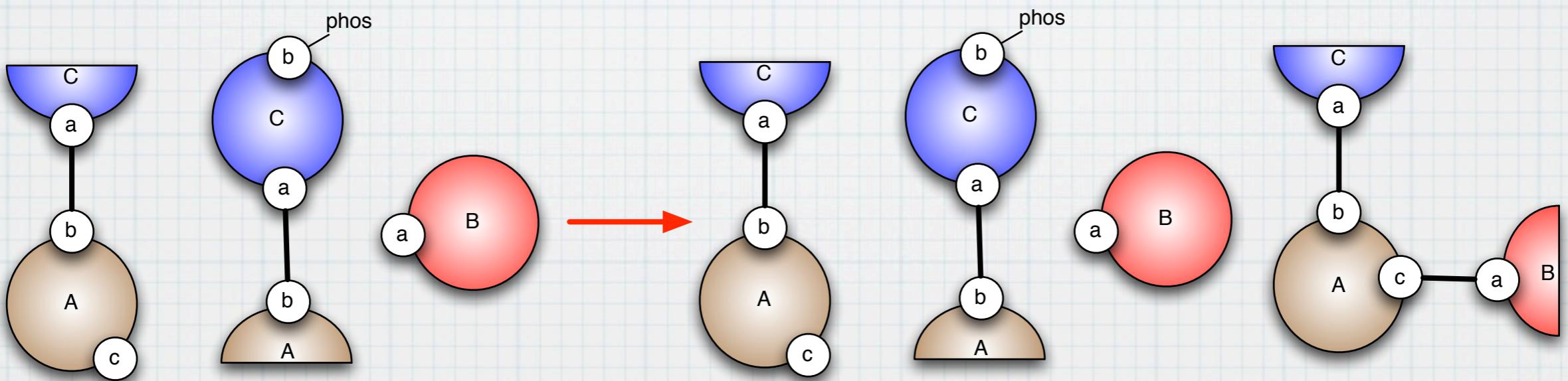
$F(X_n) = X_{n+1}$: apply rule to complexes in X_n and add new complexes

Local views



X_n : accessible complexes computed so far

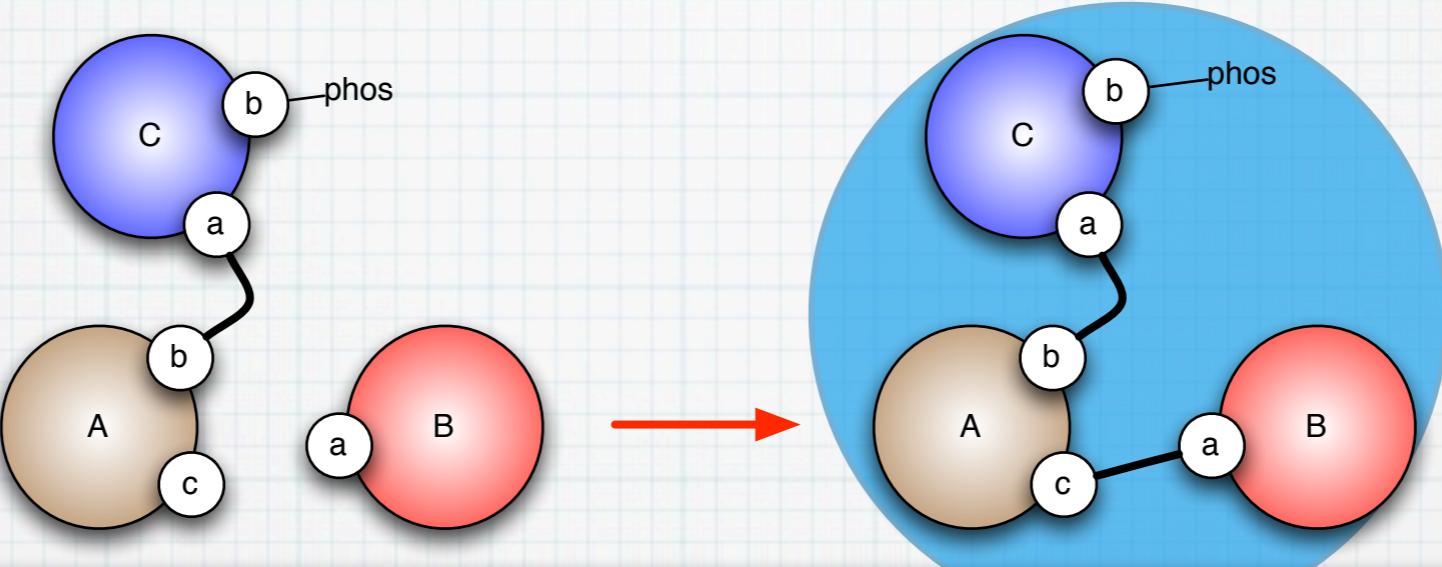
$F(X_n) = X_{n+1}$: apply rule to complexes in X_n and add new complexes



$X_n^\#$: accessible local views computed so far

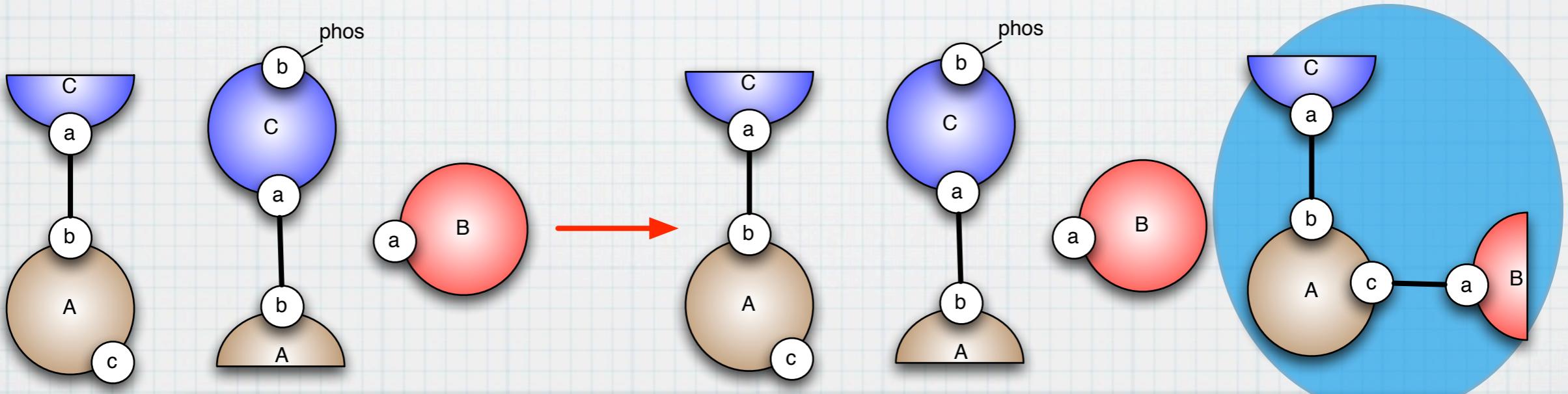
$F^\#(X_n^\#) = X_{n+1}^\#$: add new local views of rhs if local views of lhs are in $X_n^\#$

Local views



X_n : accessible complexes computed so far

$F(X_n) = X_{n+1}$: apply rule to complexes in X_n and add new complexes

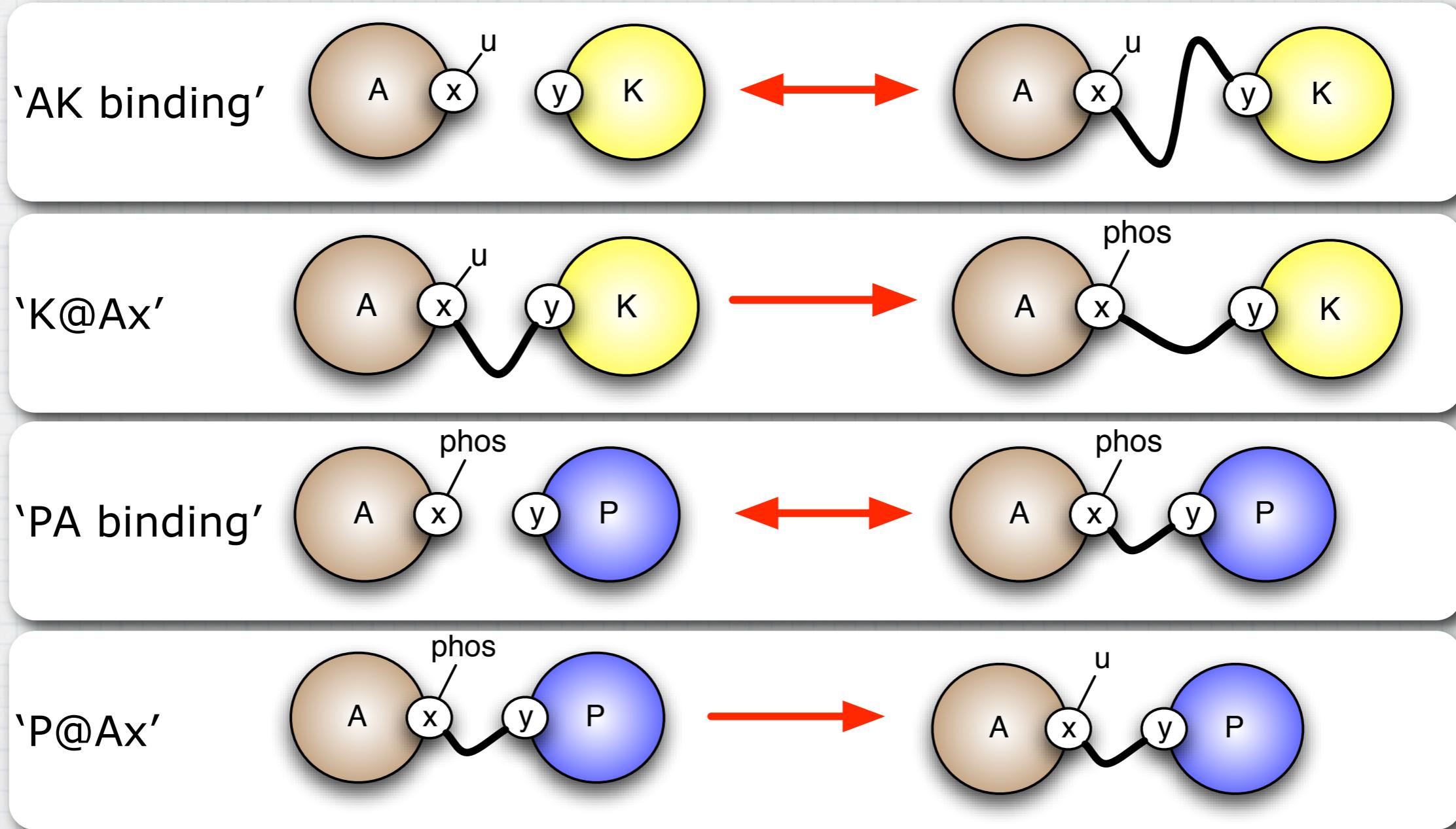


$X_n^\#$: accessible local views computed so far

$F^\#(X_n^\#) = X_{n+1}^\#$: add new local views of rhs if local views of lhs are in $X_n^\#$

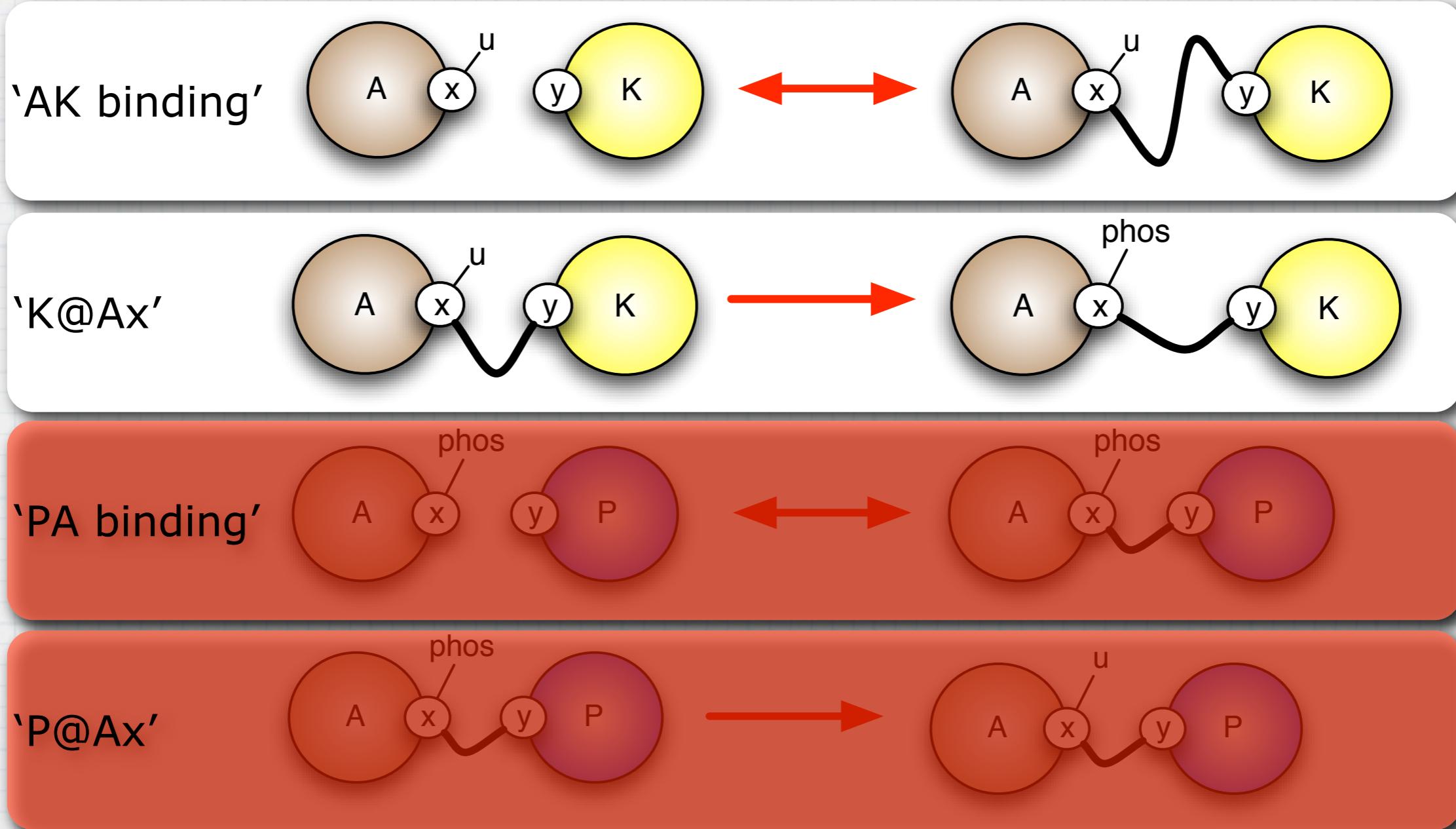
Application

Rule accessibility: compression



Application

Rule accessibility: compression



Application

Rule accessibility: compression

SIM1 - Kappa Factory

Sim1 New Sim Rename Sim Copy Sim Delete Sim Export to Kappa Delete Save Save As... Copy

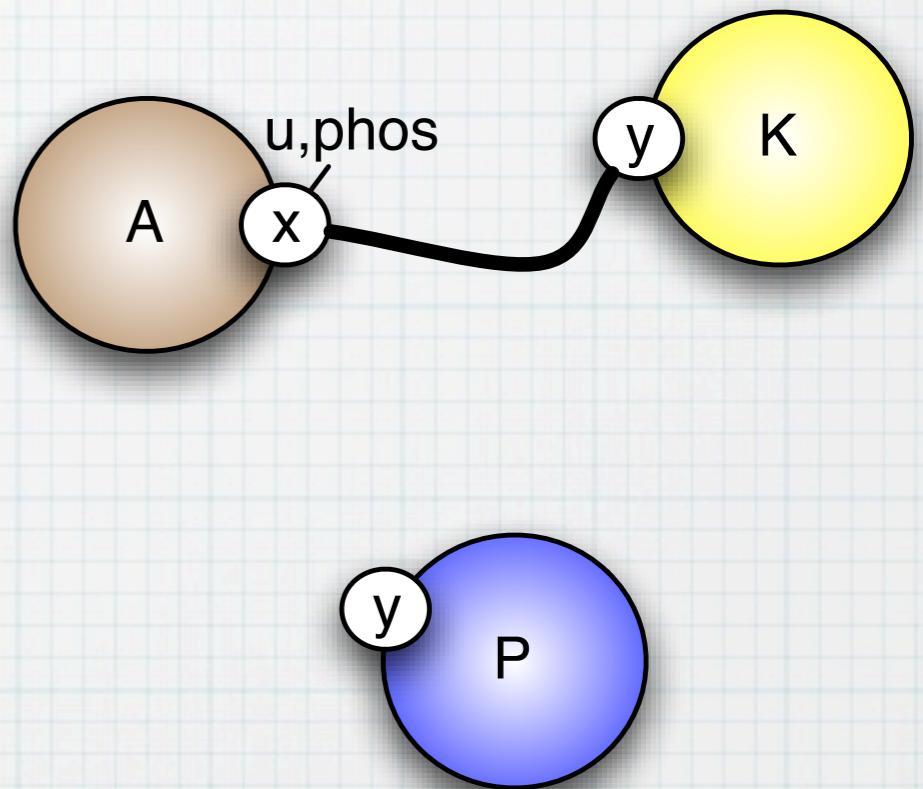
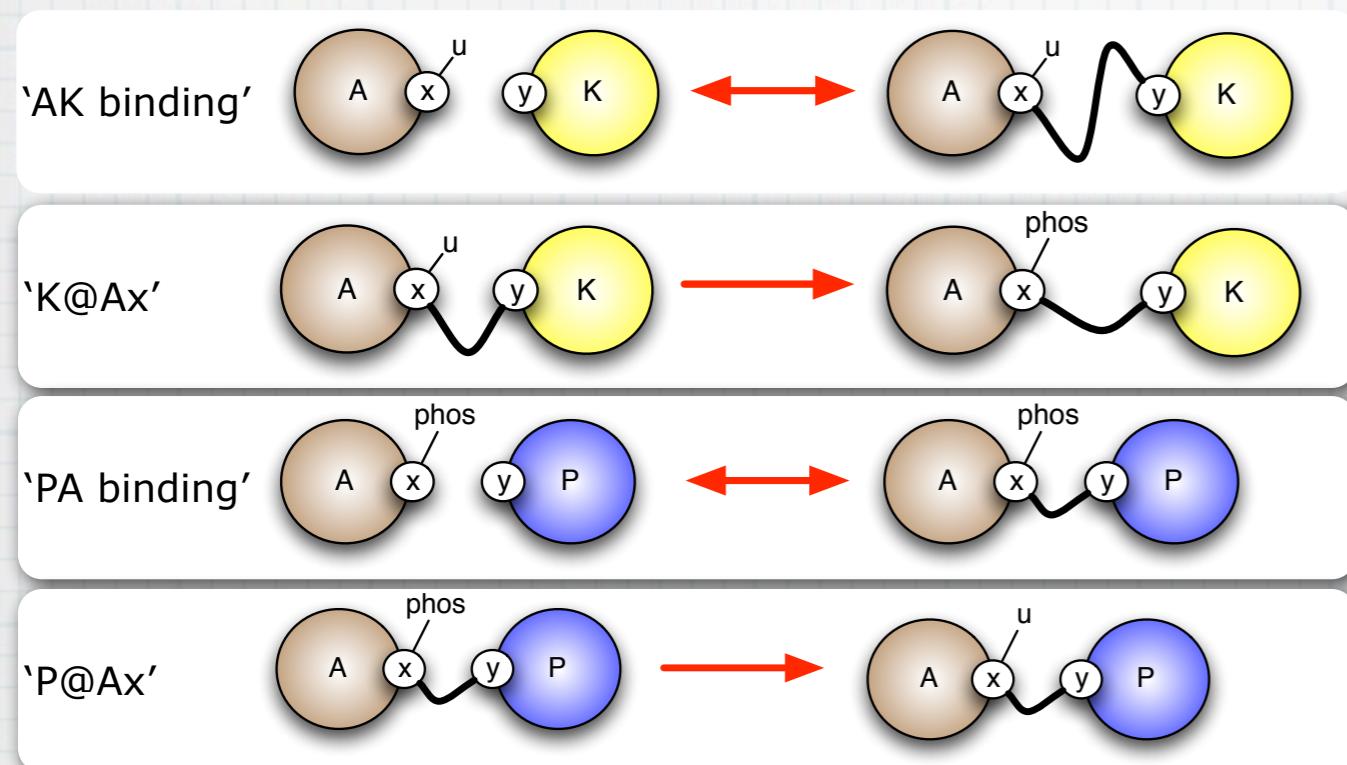
Rules | Initial Conditions | Contact Map | Influence Map | Reachables | **Compression** | Simulation | Stories | Results | Annotations | Kappa

Perform Compression | Compression Type: Qualitative Quantitative

| From | To |
|--|--|
| 'Akt...GSK3a' Akt(subs!0),GSK3a(a!0)->Akt(subs),GSK3a(a) | 'Akt...GSK3a' Akt(subs!1),GSK3a(a!1) -> Akt(subs),GSK3a(a) |
| 'Akt.PHLPP' Akt(S473~p),PHLPP(PDZ)->Akt(S473~p!0),PHLPP(PDZ!0) | 'Akt.PHLPP' Akt(S473~p),PHLPP(PDZ) -> Akt(S473~p!1),PHLPP(PDZ!1) |
| 'FKB38.Rheb' FKB38(a),Rheb(fkb,g-gtp)->FKB38(a!0),Rheb(fkb!0,g-gtp) | 'FKB38.Rheb' FKB38(a),Rheb(fkb,g-gtp) -> FKB38(a!1),Rheb(fkb!1,g-gtp) |
| 'mTOR.FKB38' FKB38(a,m),mTOR(fkb38)->FKB38(a,m!0),mTOR(fkb38!0) | 'mTOR.FKB38' FKB38(a,m),mTOR(fkb38) -> FKB38(a,m!1),mTOR(fkb38!1) |
| 'GSK3a@S21' Akt(T308~p,subs!0),GSK3a(S21~u,a!0)->Akt(T308~p,subs!0),GSK3a(S21~p,a!0) | 'GSK3a@S21' GSK3a(S21~u,a!1),Akt(T308~p,subs!1) -> GSK3a(S21~p,a!1),Akt(T308~p,subs!1) |
| 'Akt@S473' mTOR(fkb12!3,heat!2,m!1,sin!0),Rictor(m!1!2,m!3,sin!4),Akt(S473~u,sin!5),SIN1(akt!5,m!0,ric!4),mL | 'Akt@S473' Akt(S473~u,sin!6),mTOR(fkb12!4,heat!5,m!1,sin!2),Rictor(m!1!5,m!2,sin!3),SIN1(akt!6,m!2,ric!3),mL |
| 'Akt.GSK3a' Akt(T308~p,subs),GSK3a(a)->Akt(T308~p,subs!0),GSK3a(a!0) | 'Akt.GSK3a' Akt(T308~p,subs),GSK3a(a) -> Akt(T308~p,subs!1),GSK3a(a!1) |
| 'S6K1.S6@S235' S6K1(T389~p,subs!0),S6(S235~u!0)->S6K1(T389~p,subs!0),S6(S235~p!0) | 'S6K1.S6@S235' S6(S235~u!_) -> S6(S235~p!_) |
| 'Akt.PIP' Akt(PH),PIPthree~p)->Akt(PH!0),PIPthree~p!0) | 'Akt.PIP' Akt(PH),PIPthree~p) -> Akt(PH!1),PIPthree~p!1) |
| 'S6K1.Raptor' S6K1(tos),Raptor(s)->S6K1(tos!0),Raptor(s!0) | 'S6K1.Raptor' S6K1(tos),Raptor(s) -> S6K1(tos!1),Raptor(s!1) |
| '4E-BP1.Raptor' 4E-BP1(tos),Raptor(s)->4E-BP1(tos!0),Raptor(s!0) | '4E-BP1.Raptor' 4E-BP1(tos),Raptor(s) -> 4E-BP1(tos!1),Raptor(s!1) |
| Akt...FoxO3a@T32' Akt(subs!0),FoxO3a(T32~0)->Akt(subs),FoxO3a(T32) | Akt...FoxO3a@T32' Cannot be applied |
| 'GSK3b@S9' Akt(T308~p,subs!0),GSK3b(S9~u,a!0)->Akt(T308~p,subs!0),GSK3b(S9~p,a!0) | 'GSK3b@S9' GSK3b(S9~u,a!1),Akt(T308~p,subs!1) -> GSK3b(S9~p,a!1),Akt(T308~p,subs!1) |
| 'Akt.PP2A' Akt(T308~p),PP2A(B55~a)->Akt(T308~p!0),PP2A(B55~a!0) | 'Akt.PP2A' Akt(T308~p),PP2A(B55~a) -> Akt(T308~p!1),PP2A(B55~a!1) |
| 'FKB12.rapa' FKB12(a),rapamycin(a)->FKB12(a!0),rapamycin(a!0) | 'FKB12.rapa' FKB12(a),rapamycin(a) -> FKB12(a!1),rapamycin(a!1) |
| 'IRS1...PI3KE545K' IRS1(pi3k!0),PI3KE545K(SH2!0)->IRS1(pi3k),PI3KE545K(SH2) | 'IRS1...PI3KE545K' IRS1(pi3k!_) -> IRS1(pi3k) |
| 'PIP#three' PIPthree~p!0),PTEN(subs!0)->PIPthree~u!0),PTEN(subs!0) | 'PIP#three' PIPthree~p!1),PTEN(subs!1) -> PIPthree~u!1),PTEN(subs!1) |
| 'IRS1@S302' S6K1(subs!0),IRS1(S302~u!0)->S6K1(subs!0),IRS1(S302~p!0) | 'IRS1@S302' IRS1(S302~u!_) -> IRS1(S302~p!_) |
| 'TSC2@S1388' AMPK(a~p,tsc!0),TSC2(S1388~u,a!0)->AMPK(a~p,tsc!0),TSC2(S1388~p,a!0) | 'TSC2@S1388' TSC2(S1388~u,a!1),AMPK(tsc!1) -> TSC2(S1388~p,a!1),AMPK(tsc!1) |
| 'PDK1.Akt' PDK1(PH!0,akt),PIP(three~p!0),Akt(PH!1,T308~u),PIP(three~p!1)->PDK1(PH!1,akt!0),PIP(three~p!1),Akt'PDK1.Akt' PDK1(PH!_,akt),Akt(PH!_,T308~u) -> PDK1(PH!_,akt!1),Akt(PH!_,T308~u!1) | |
| Akt.FoxO3a@T32' Akt(S473~p,T308~p,subs),FoxO3a(T32~u)->Akt(S473~p,T308~p,subs!0),FoxO3a(T32~u!0) | Akt.FoxO3a@T32' Cannot be applied |
| 'S6K1@T229' PDK1(s6k1!0),S6K1(T229~u!0)->PDK1(s6k1!0),S6K1(T229~p!0) | 'S6K1@T229' S6K1(T229~u!_) -> S6K1(T229~p!_) |
| FoxO3a@T32' Akt(S473~p,T308~p,subs!0),FoxO3a(T32~u!0)->Akt(S473~p,T308~p,subs!0),FoxO3a(T32~p!0) | FoxO3a@T32' Cannot be applied |
| 'PDK1...Akt' PDK1(akt!0),Akt(T308!0)->PDK1(akt),Akt(T308) | 'PDK1...Akt' PDK1(akt!_) -> PDK1(akt) |
| 'Akt...TSC2@T1462' Akt(subs!0),TSC2(T1462,a!0)->Akt(subs),TSC2(T1462,a) | 'Akt...TSC2@S1086,Akt...TSC2@S939,Akt...TSC2@T1462' Akt(subs!1),TSC2(a!1) -> Akt(subs),TSC2(a) |
| 'Akt...TSC2@S939' Akt(subs!0),TSC2(S939,a!0)->Akt(subs),TSC2(S939,a) | |
| 'Akt...TSC2@S1086' Akt(subs!0),TSC2(S1086,a!0)->Akt(subs),TSC2(S1086,a) | |
| 'IRS1.S6K1' S6K1(subs),IRS1(S302~u)->S6K1(subs!0),IRS1(S302~u!0) | 'IRS1.S6K1' S6K1(subs),IRS1(S302~u) -> S6K1(subs!1),IRS1(S302~u!1) |
| 'Akt.TSC2@T1462' Akt(T308~p,subs),TSC2(T1462~u,a)->Akt(T308~p,subs!0),TSC2(T1462~u,a!0) | 'Akt.TSC2@T1462' Akt(T308~p,subs),TSC2(T1462~u,a) -> Akt(T308~p,subs!1),TSC2(T1462~u,a!1) |
| 'PI3KE545K.PIP@three' PI3KE545K(subs!0),PIPthree~u!0)->PI3KE545K(subs!0),PIP(three~p!0) | 'PI3KE545K.PIP@three' PI3KE545K(subs!1) -> PIP(three~p!1),PI3KE545K(subs!1) |
| 'Rheb.TSC2' Rheb(g-gtp),TSC2(gap)->Rheb(g-gtp!0),TSC2(gap!0) | 'Rheb.TSC2' Rheb(g-gtp),TSC2(gap) -> Rheb(g-gtp!1),TSC2(gap!1) |
| 'PDK1...PIP' PDK1(PH!0),PIP(three~p!0)->PDK1(PH),PIP(three~p) | 'PDK1...PIP' PDK1(PH!_) -> PDK1(PH) |
| 'RhebGDP' Rheb(g-gdp!0),TSC2(gap!0)->Rheb(g-gdp!0),TSC2(gap!0) | 'RhebGDP' Rheb(g-gdp!_) -> Rheb(g-gdp!_) |
| 'PTEN...PIP' PIP(three!0),PTEN(subs!0)->PIP(three),PTEN(subs) | 'PTEN...PIP' PIP(three!1),PTEN(subs!1) -> PIP(three),PTEN(subs) |
| 'mTOR.LST8' mTOR(),mLST8(m)->mTOR(l!0),mLST8(m!0) | 'mTOR.LST8' mTOR(),mLST8(m) -> mTOR(l!1),mLST8(m!1) |
| 'Akt...SIN1' Akt(sin!0),SIN1(akt!0)->Akt(sin),SIN1(akt) | 'Akt...SIN1' Akt(sin!_) -> Akt(sin) |
| 'AMPK...TSC2@S1388' TSC2(S1388,a!0),AMPK(a,tsc!0)->TSC2(S1388,a),AMPK(a,tsc) | 'AMPK...TSC2@S1388' TSC2(a!1),AMPK(tsc!1) -> TSC2(a),AMPK(tsc) |
| 'TSC2@T1462' Akt(T308~p,subs!0),TSC2(T1462~u,a!0)->Akt(T308~p,subs!0),TSC2(T1462~p,a!0) | 'TSC2@T1462' TSC2(T1462~u,a!1),Akt(T308~p,subs!1) -> TSC2(T1462~p,a!1),Akt(T308~p,subs!1) |
| 'PI3KE545K@SH2' IRS1(pi3k!0),PI3KE545K(SH2~u!0)->IRS1(pi3k!0),PI3KE545K(SH2~p!0) | 'PI3KE545K@SH2' PI3KE545K(SH2~u!_) -> PI3KE545K(SH2~p!_) |
| 'Akt@T308' PDK1(PH!1,akt!0),PIP(three~p!2),Akt(PH!2,T308~u!0),PIP(three~p!1)->PDK1(PH!1,akt!0),PIP(three~p!1),Akt(PH!1,T308~u!1),PDK1(PH!_,akt!1) -> Akt(PH!_,T308~p!1),PDK1(PH!_,akt!1) | |

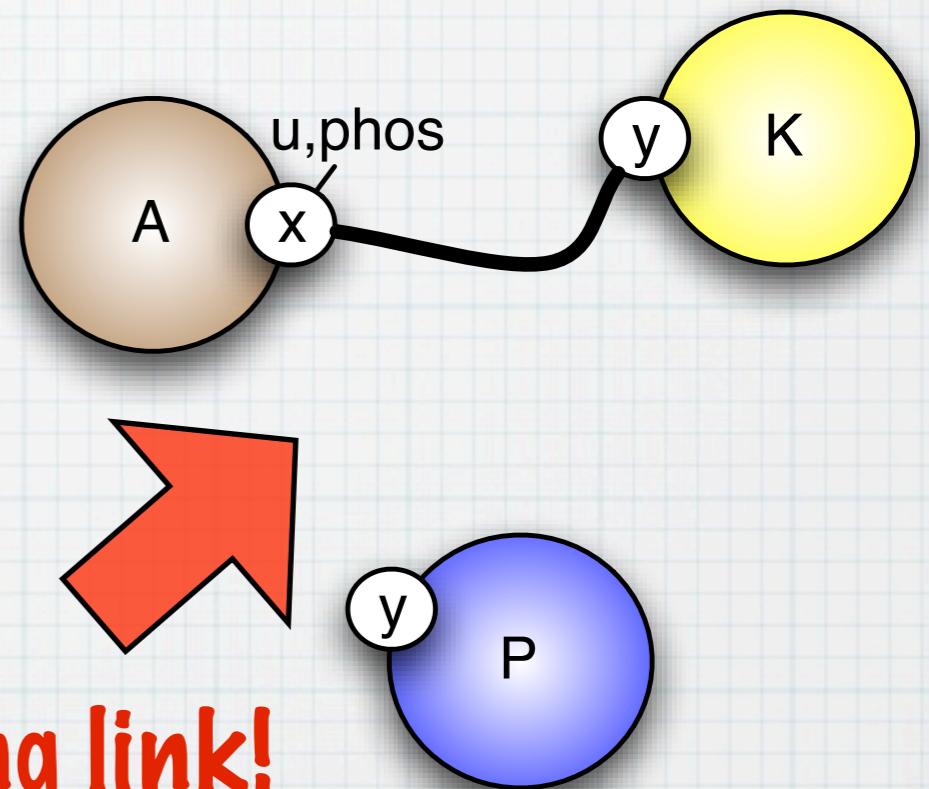
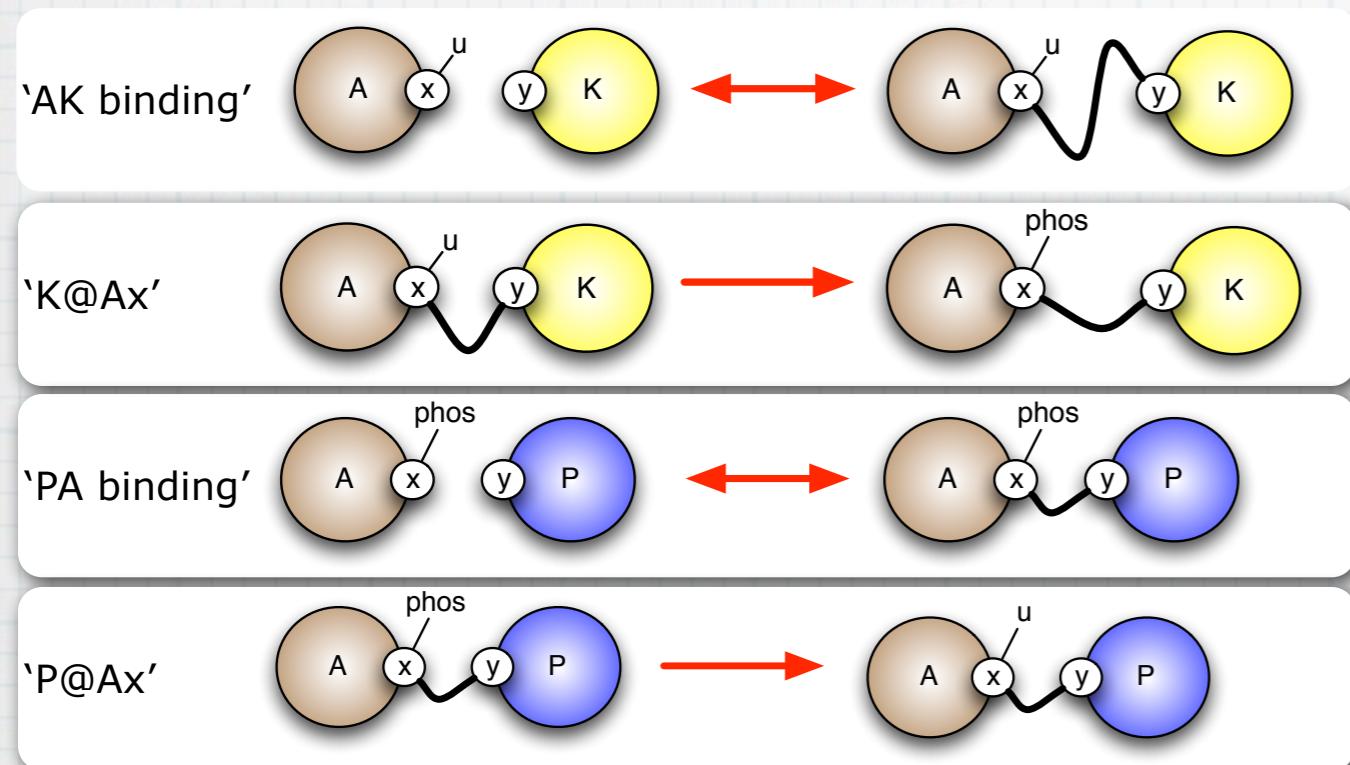
Application

Model control: contact map



Application

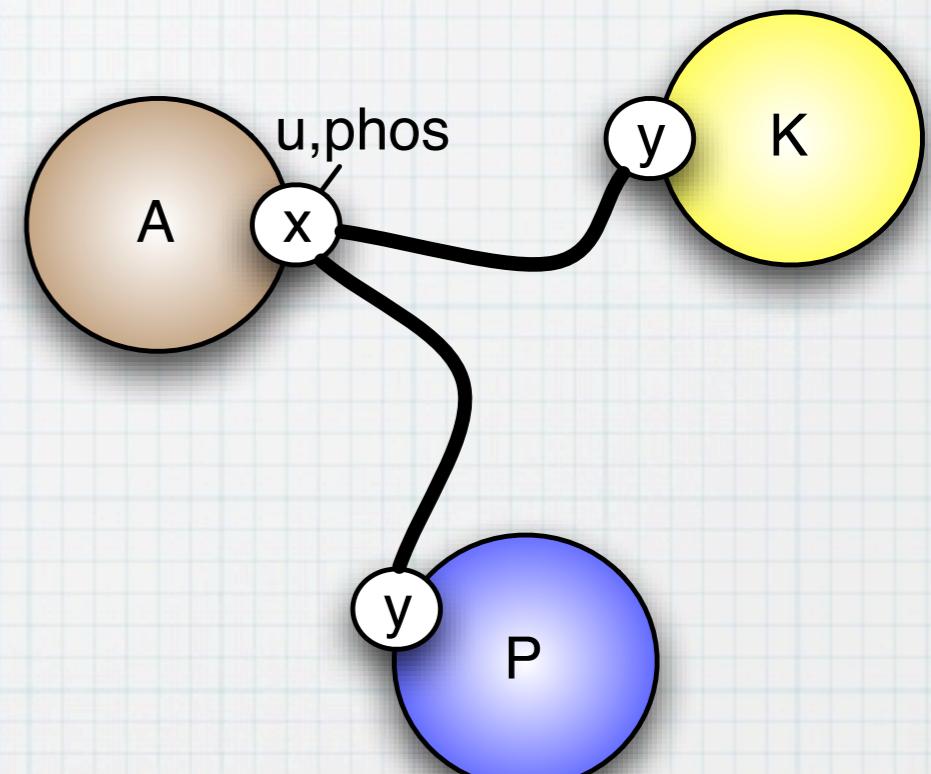
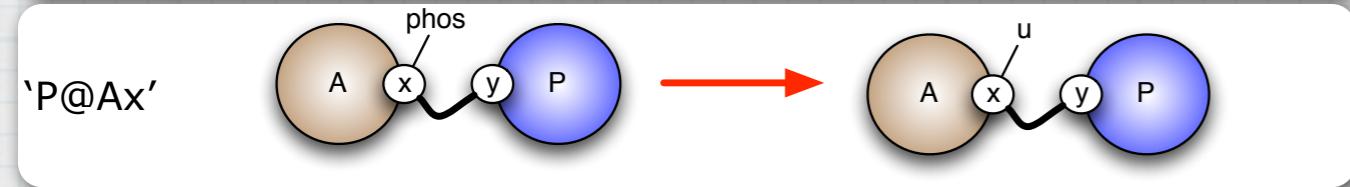
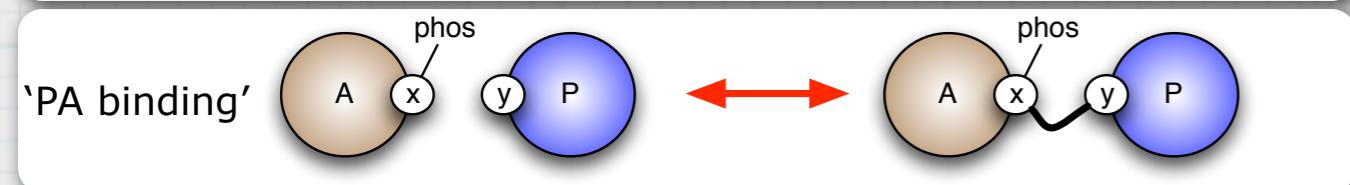
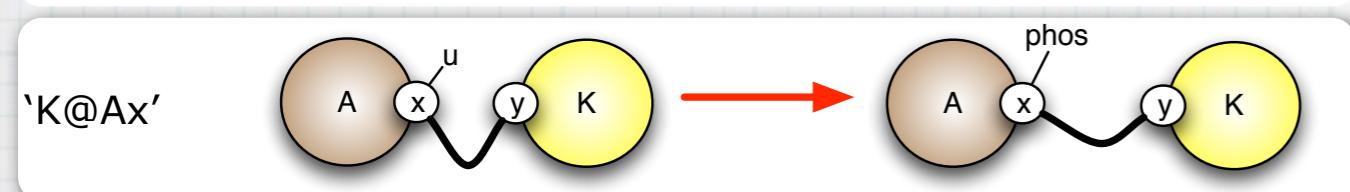
Model control: contact map



Missing link!

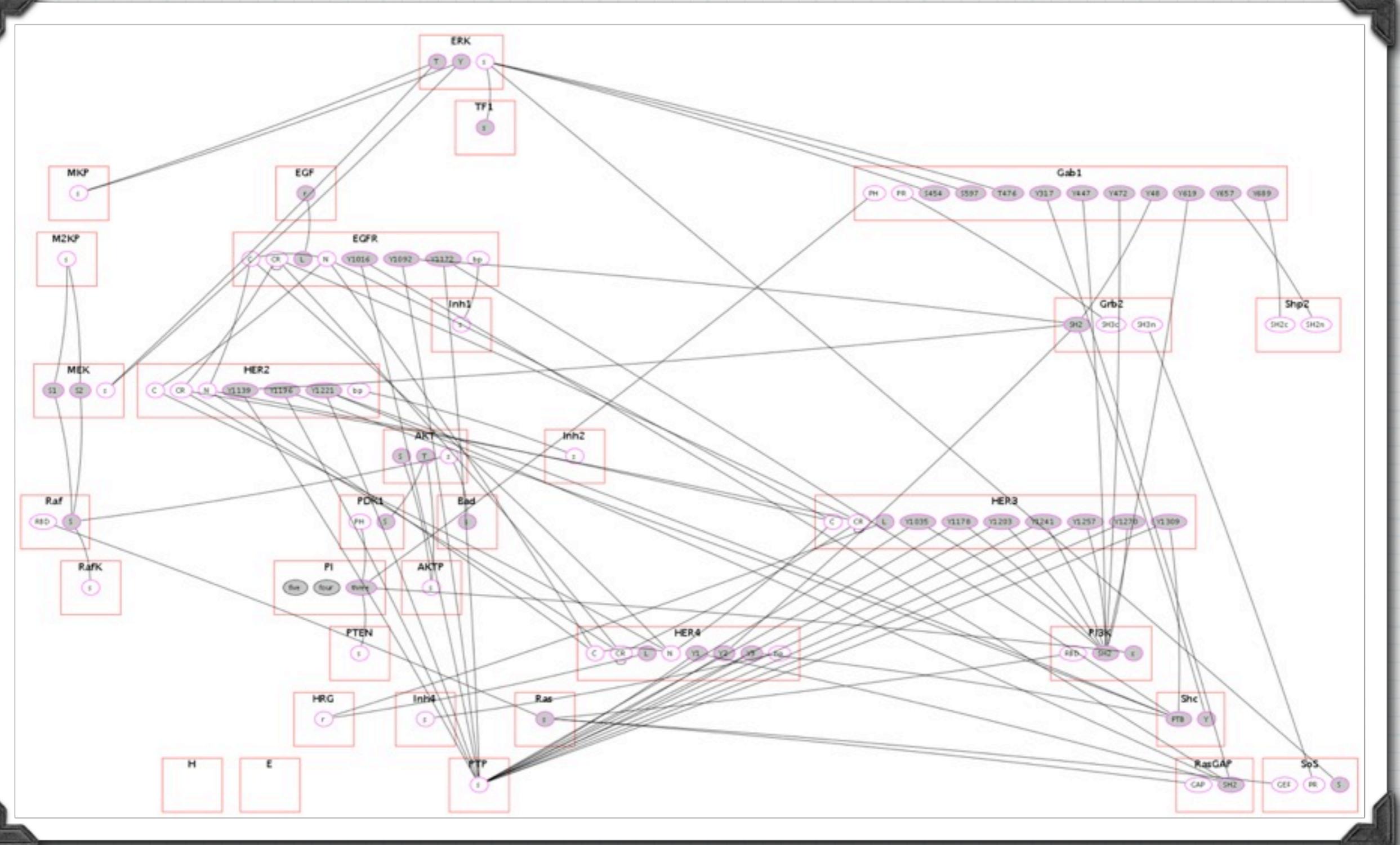
Application

Model control: contact map



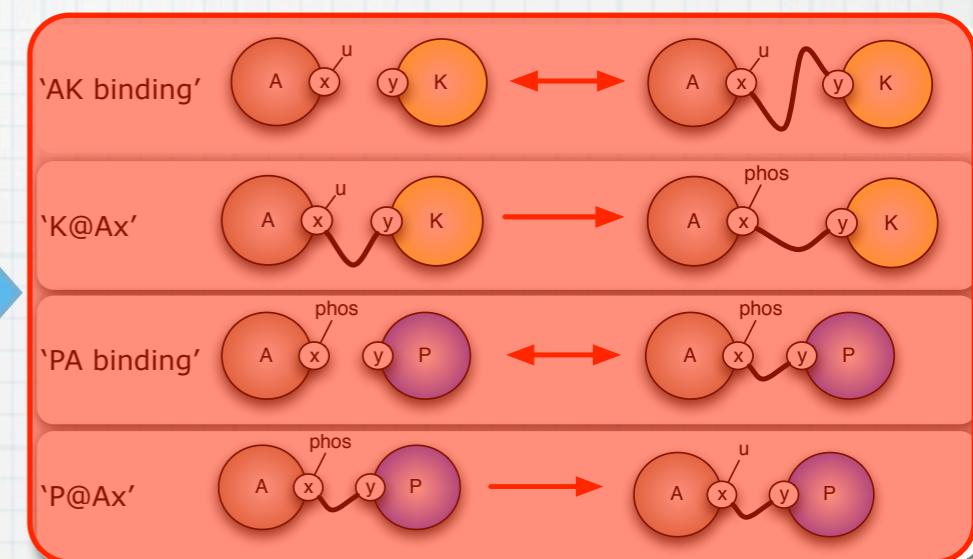
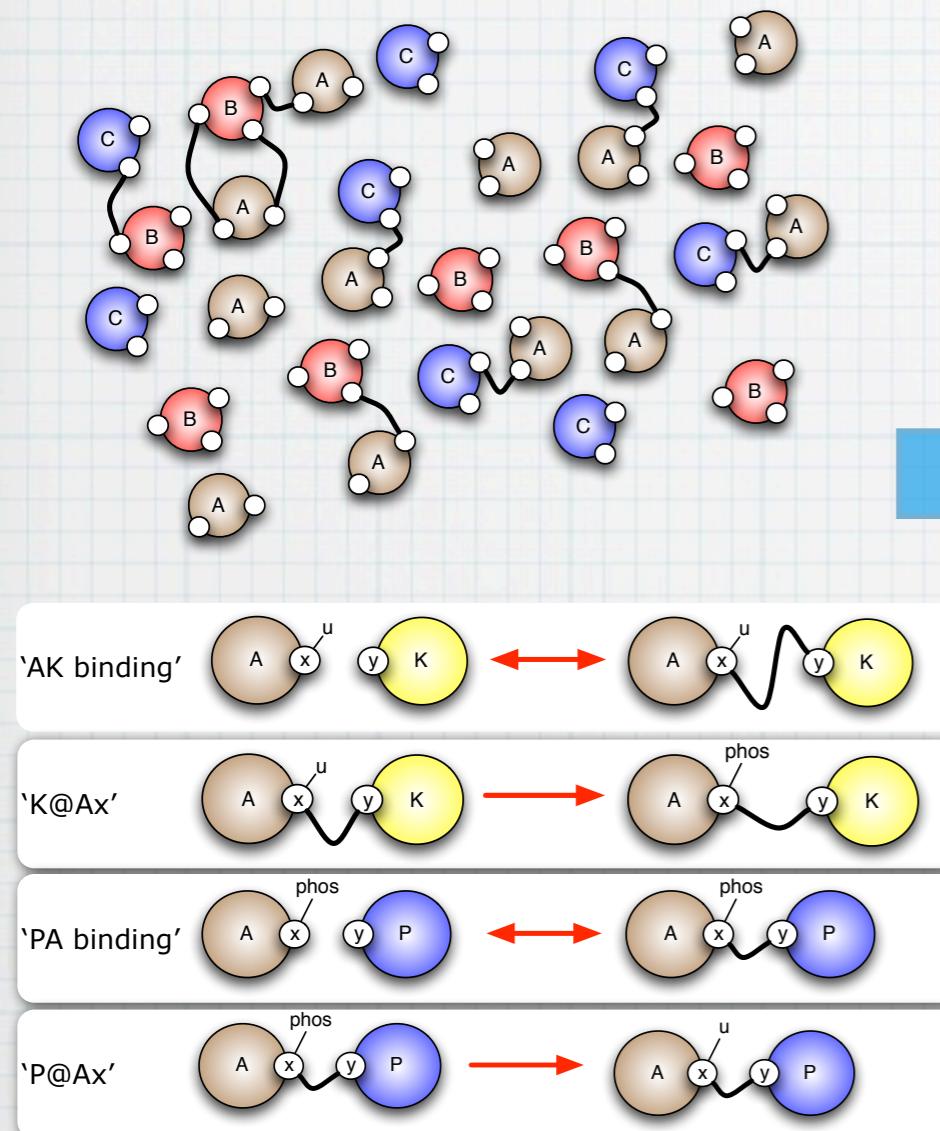
Application

Model control: contact map



NB...

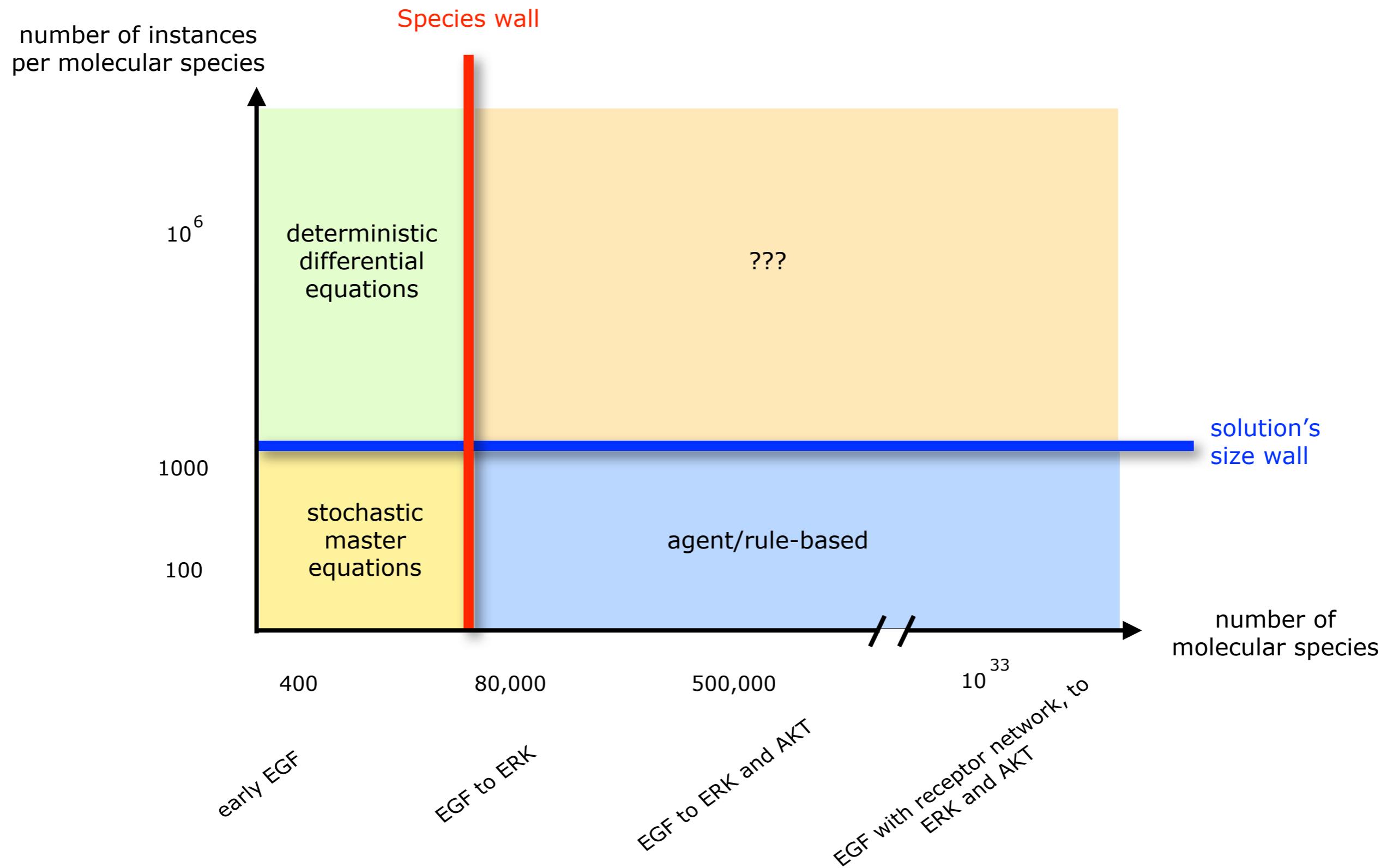
Accessibility is computed given an initial state



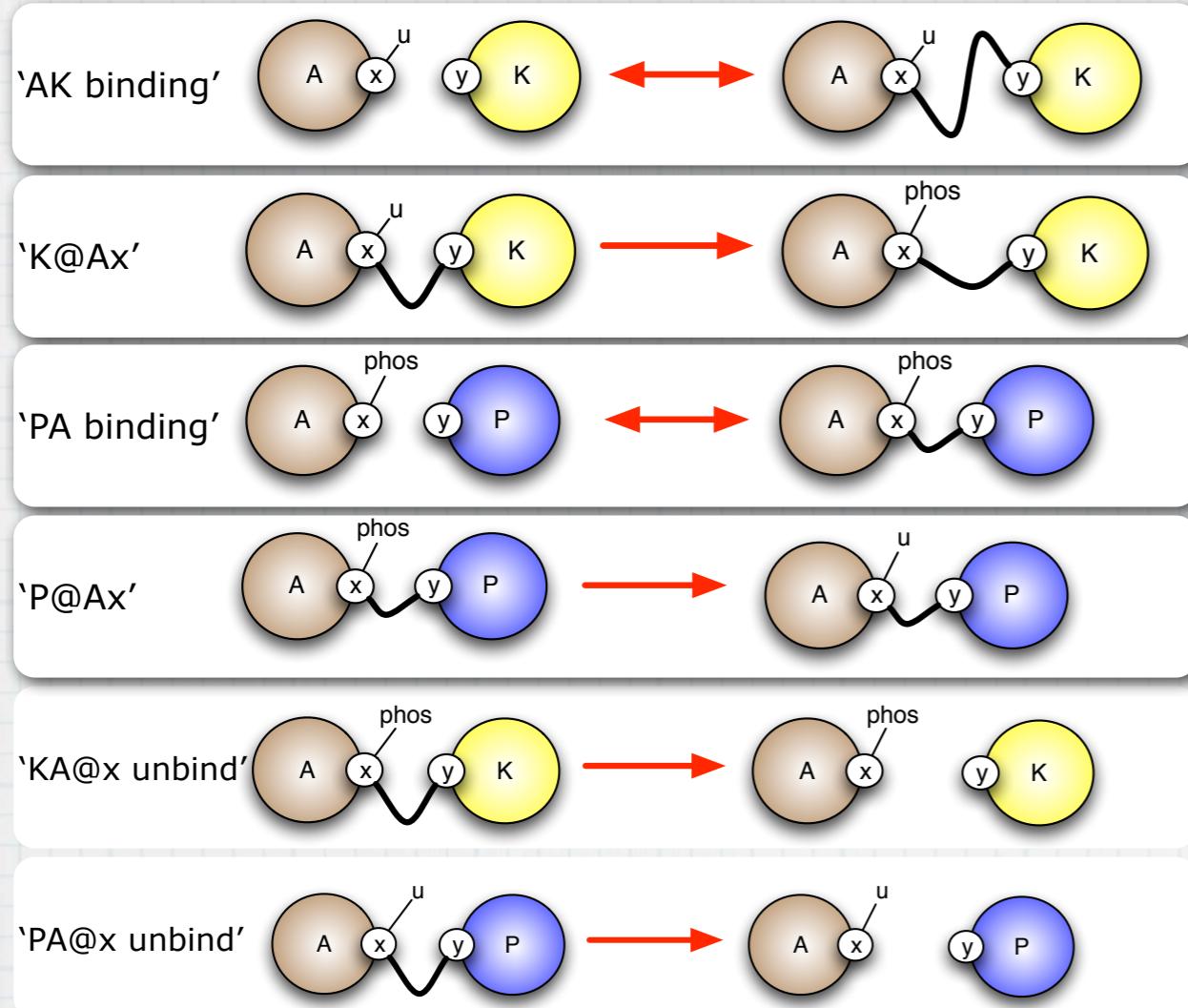
Simulation...

How to execute a model?

The axis of evil...

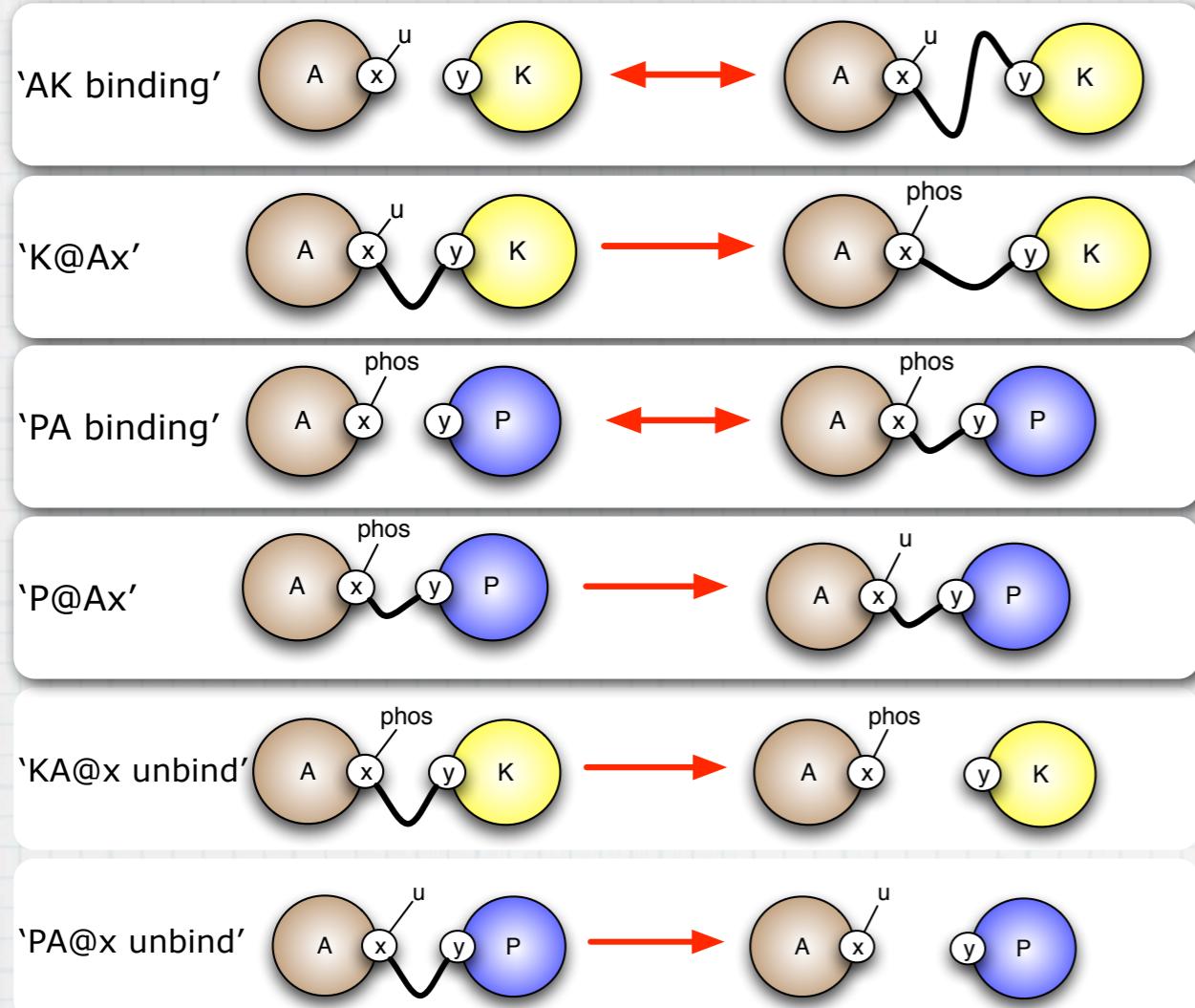


Simulation...

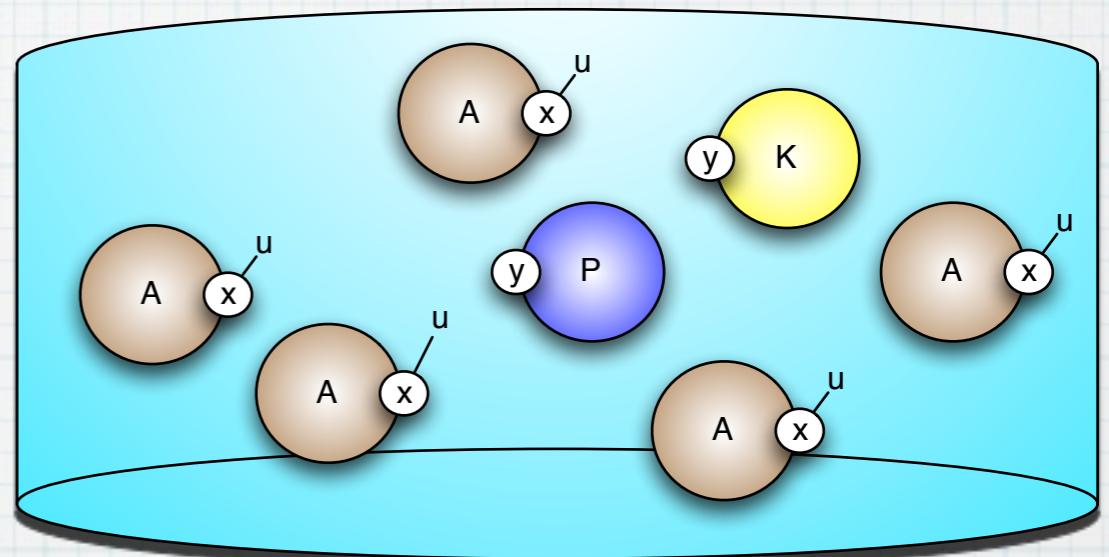


A rule set: defines the “laws” of the model

Simulation...

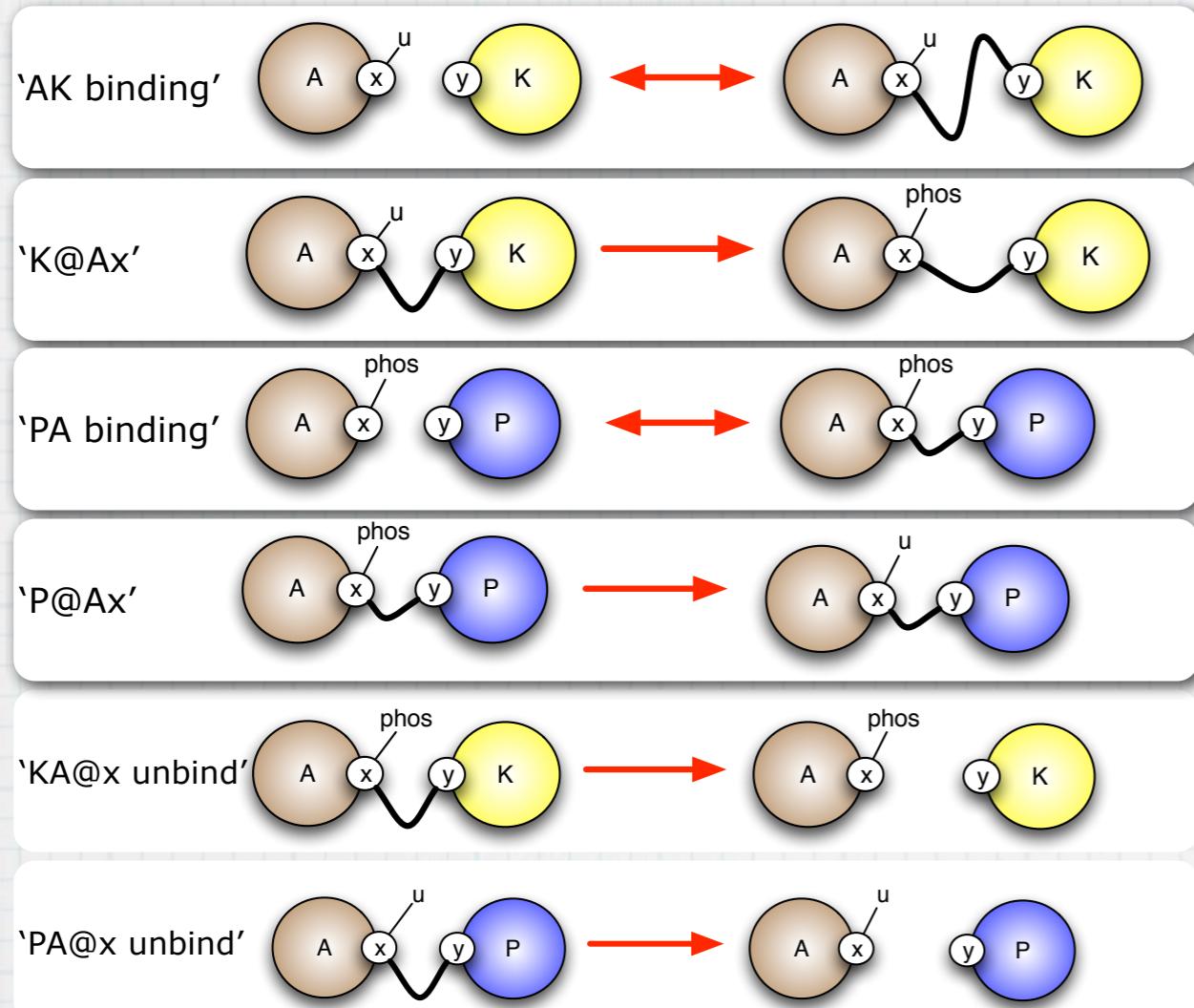


A rule set: defines the “laws” of the model

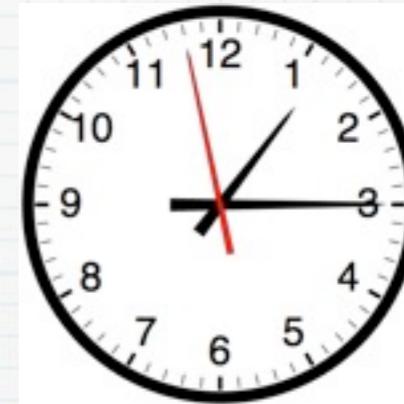


An initial solution: defines the starting state of the system.

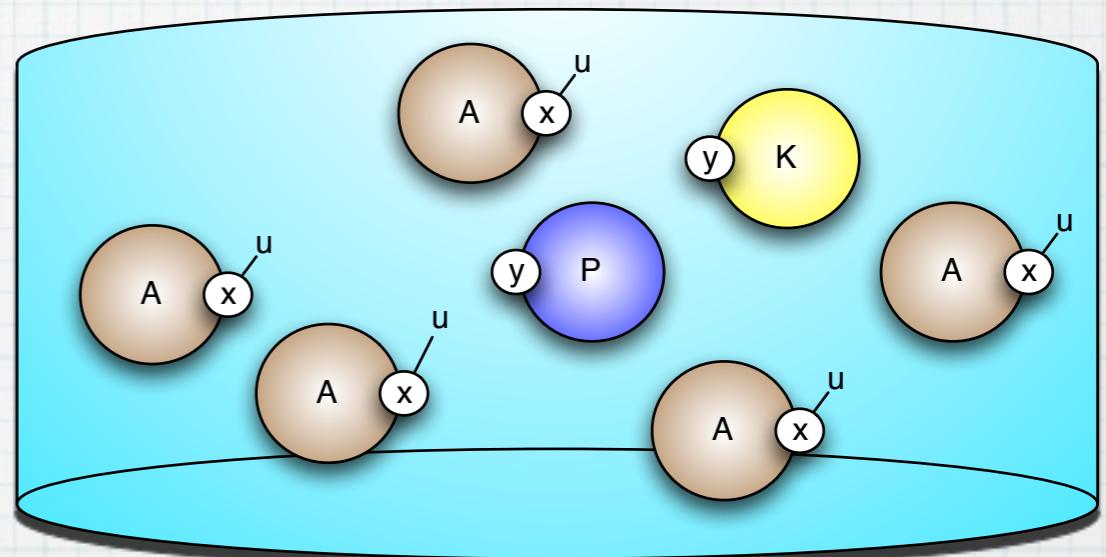
Simulation...



A rule set: defines the “laws” of the model

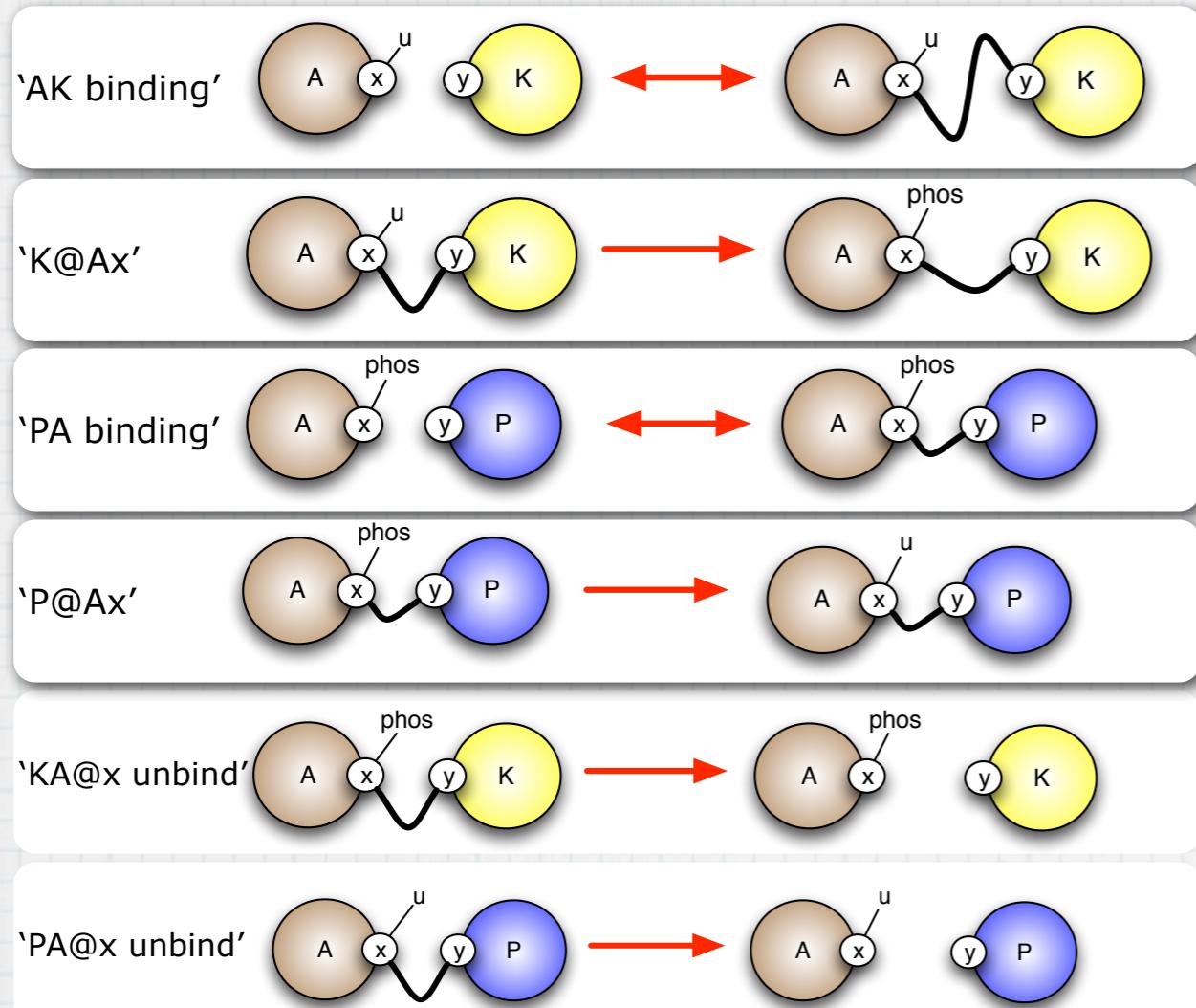


A global clock

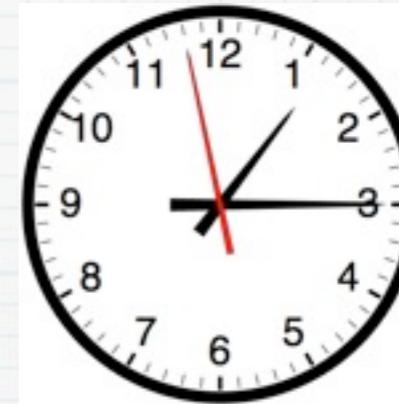


An initial solution: defines the starting state of the system.

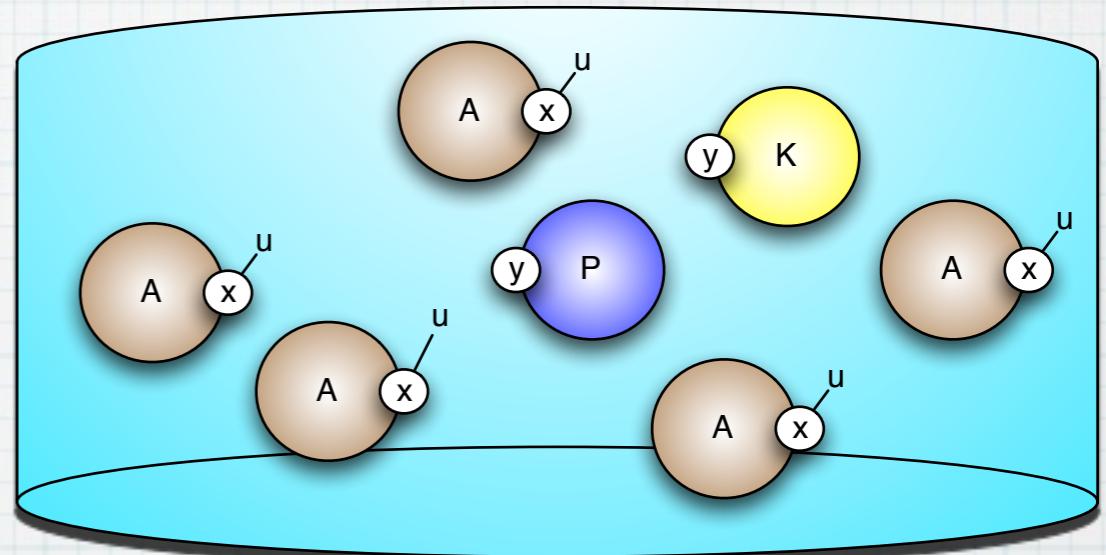
Simulation...



A rule set: defines the “laws” of the model



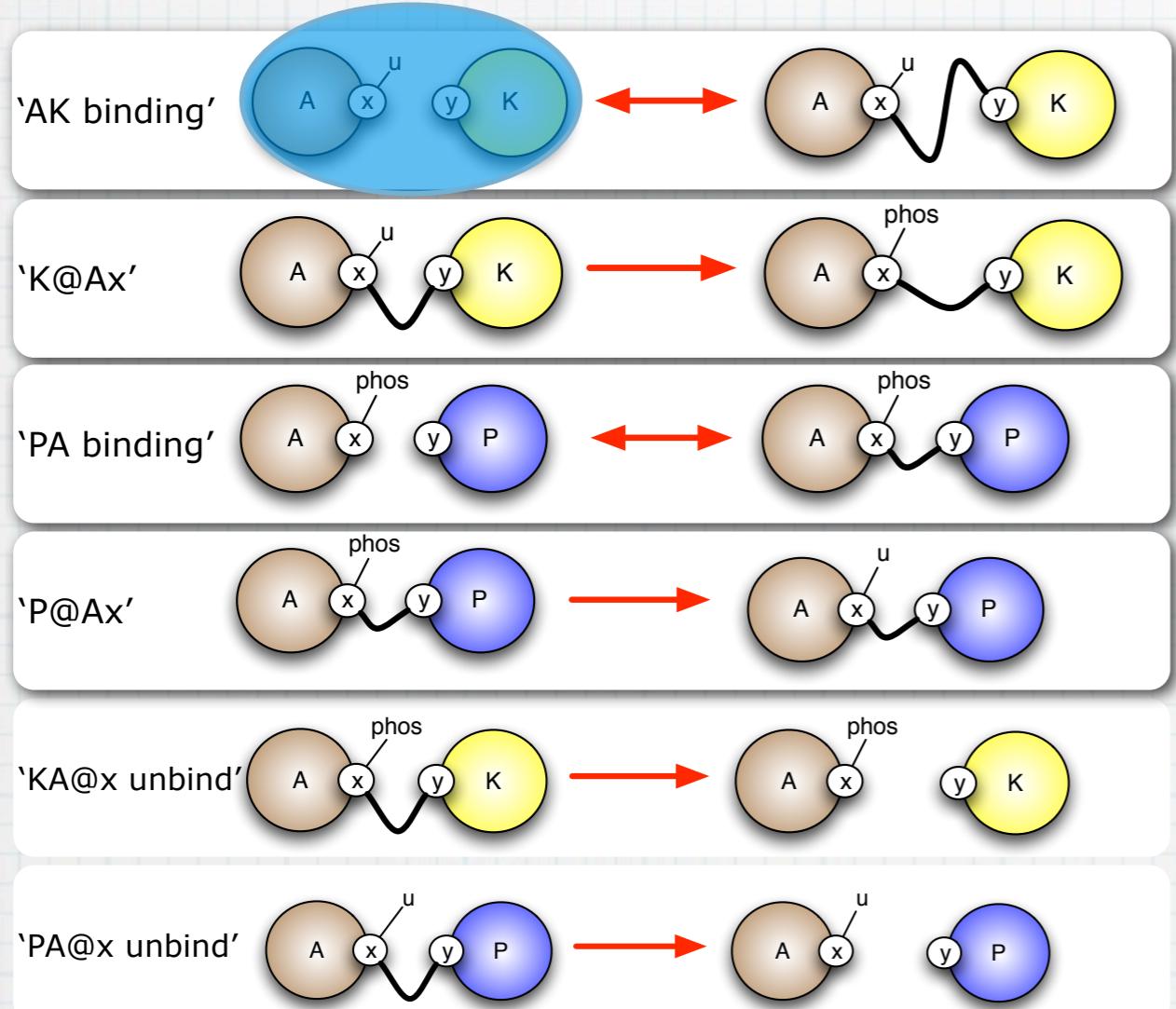
A global clock



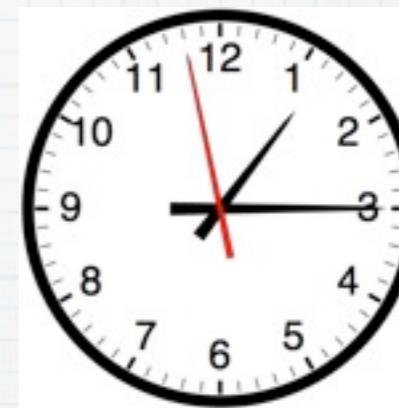
An initial solution: defines the starting state of the system.

1. Which rule can I apply next?

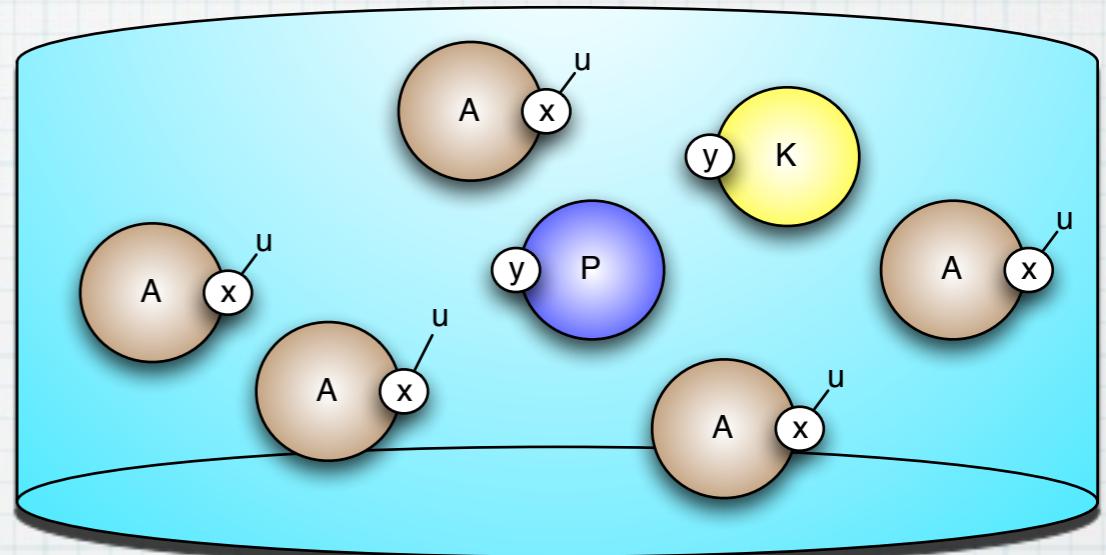
Simulation...



A rule set: defines the “laws” of the model



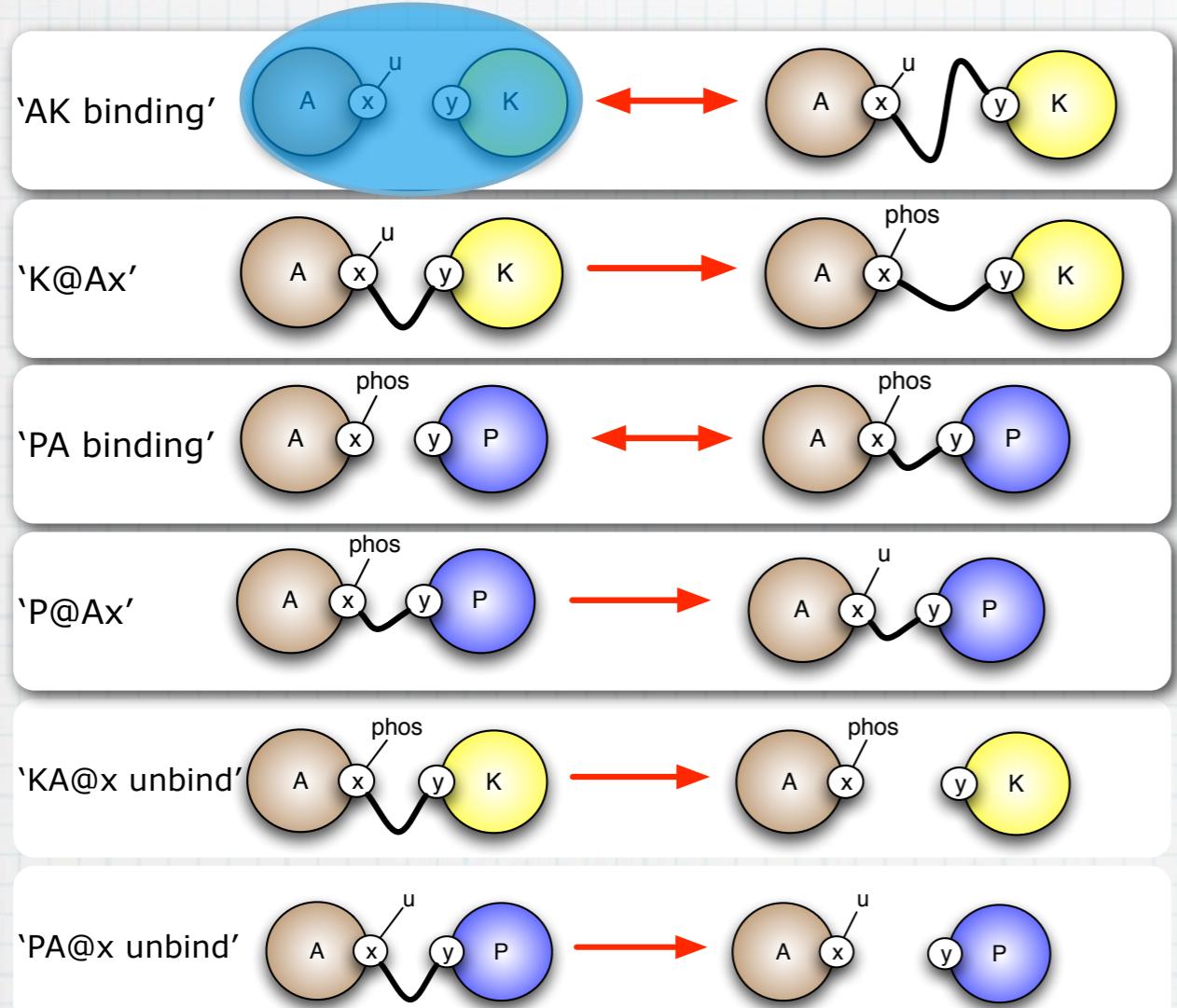
A global clock



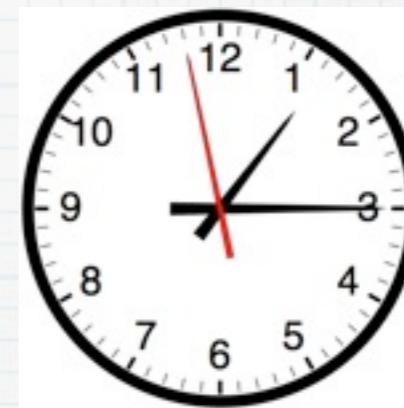
An initial solution: defines the starting state of the system.

1. Which rule can I apply next?

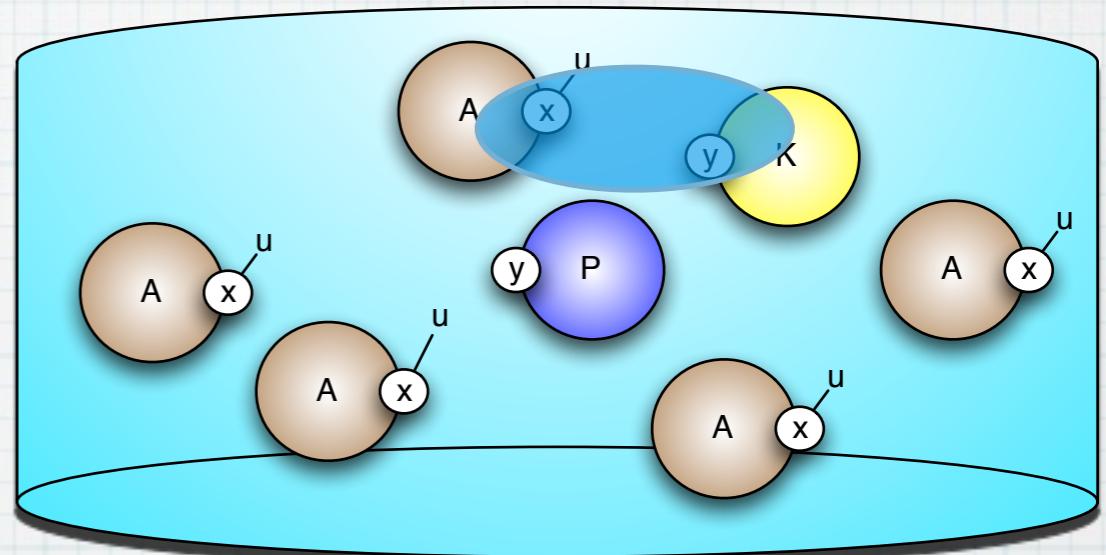
Simulation...



A rule set: defines the “laws” of the model



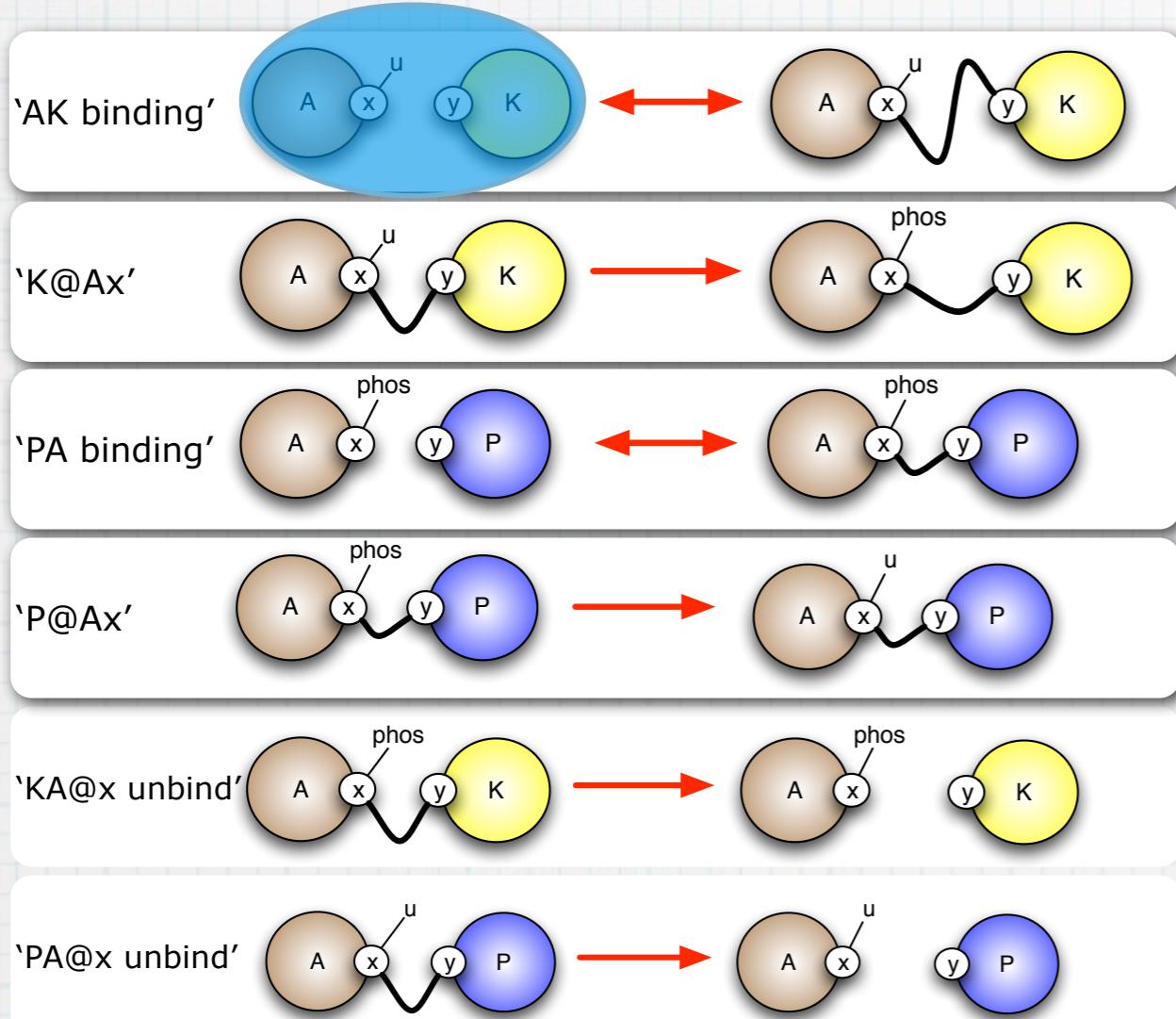
A global clock



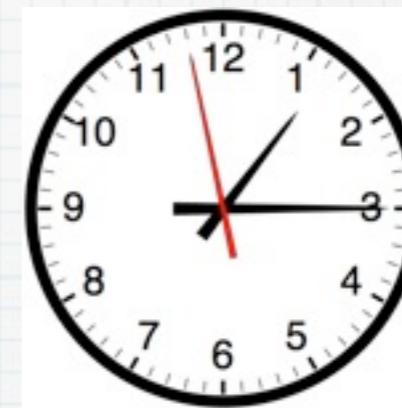
An initial solution: defines the starting state of the system.

1. Which rule can I apply next?

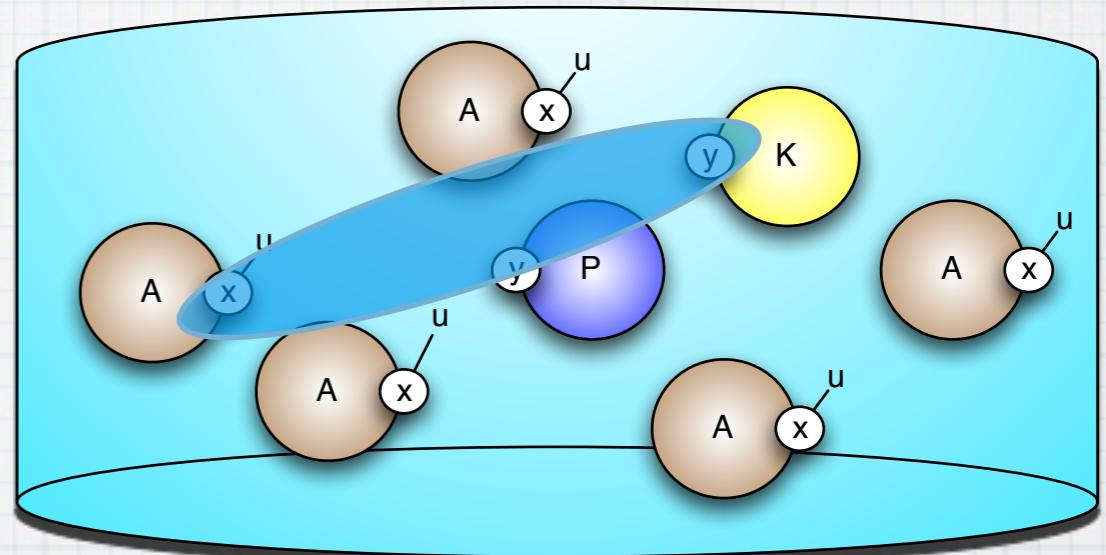
Simulation...



A rule set: defines the “laws” of the model



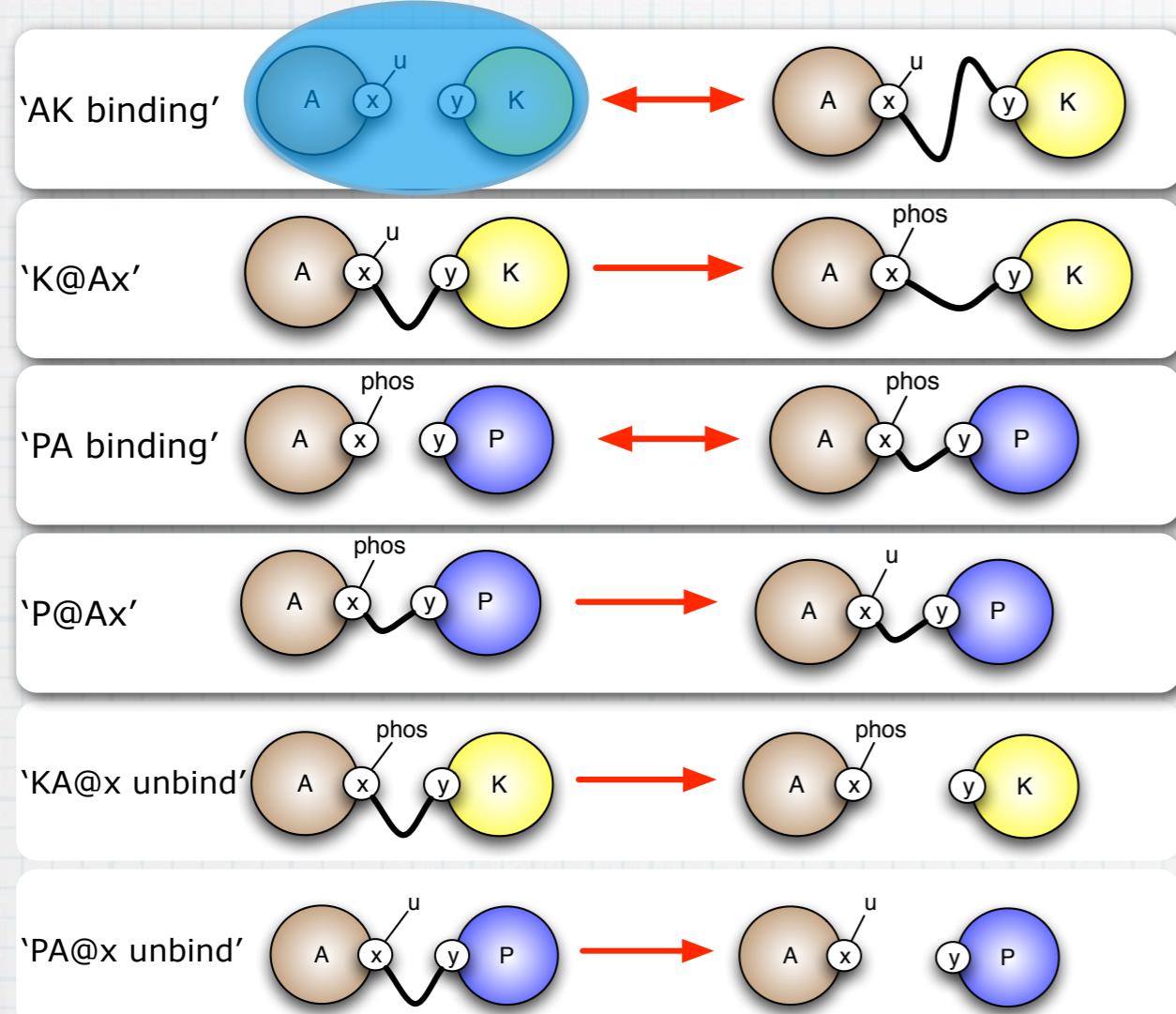
A global clock



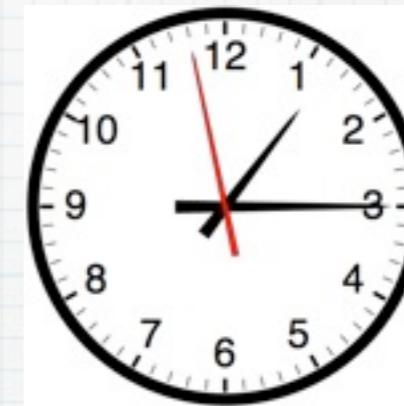
An initial solution: defines the starting state of the system.

1. Which rule can I apply next?

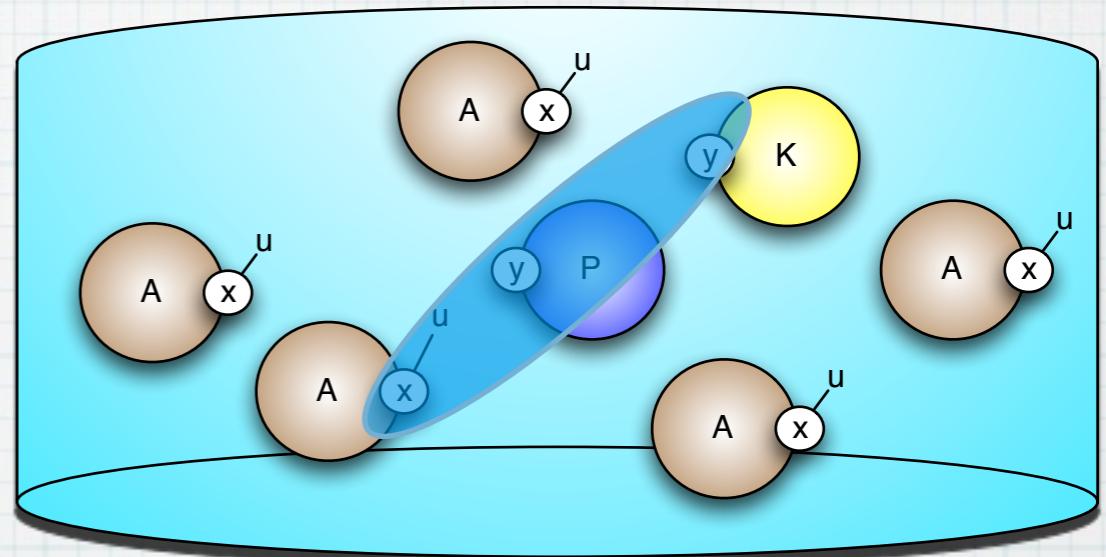
Simulation...



A rule set: defines the “laws” of the model



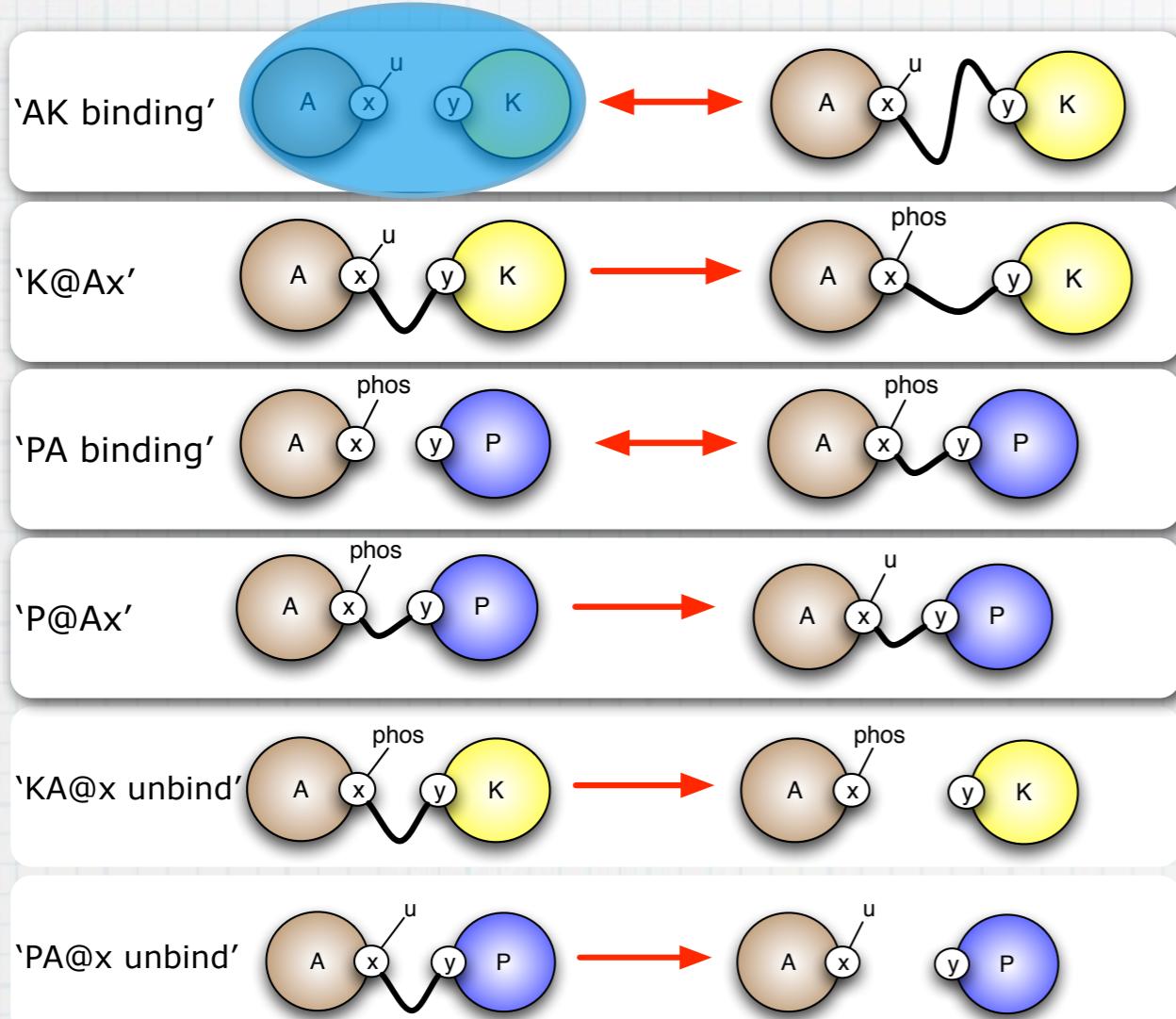
A global clock



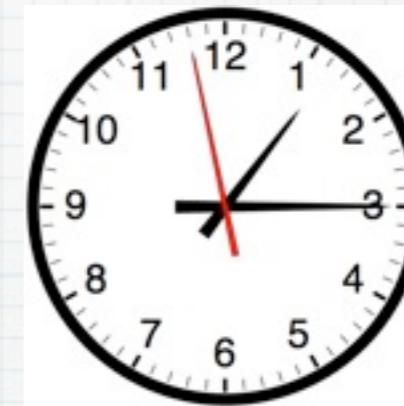
An initial solution: defines the starting state of the system.

1. Which rule can I apply next?

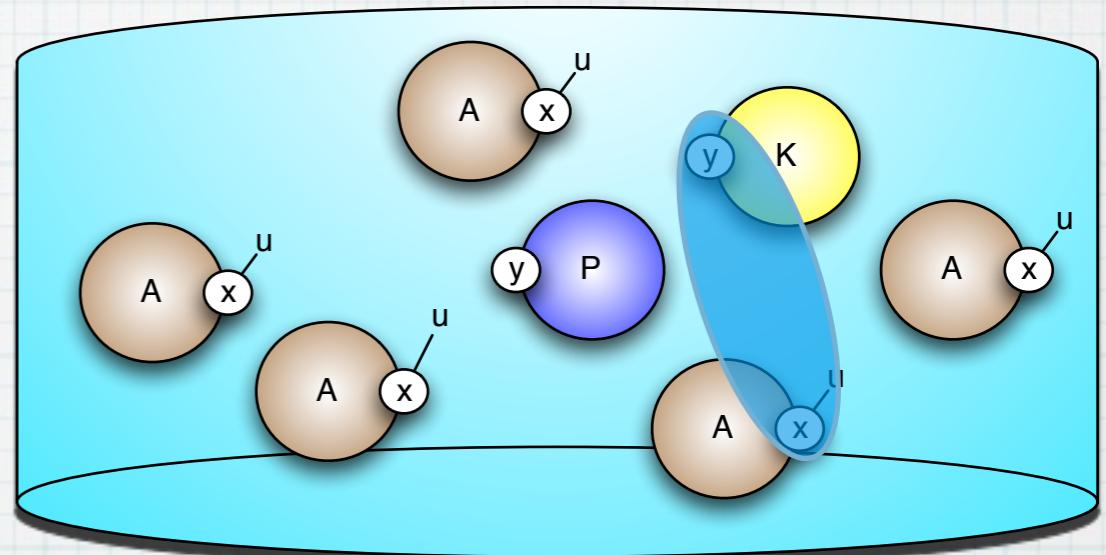
Simulation...



A rule set: defines the “laws” of the model



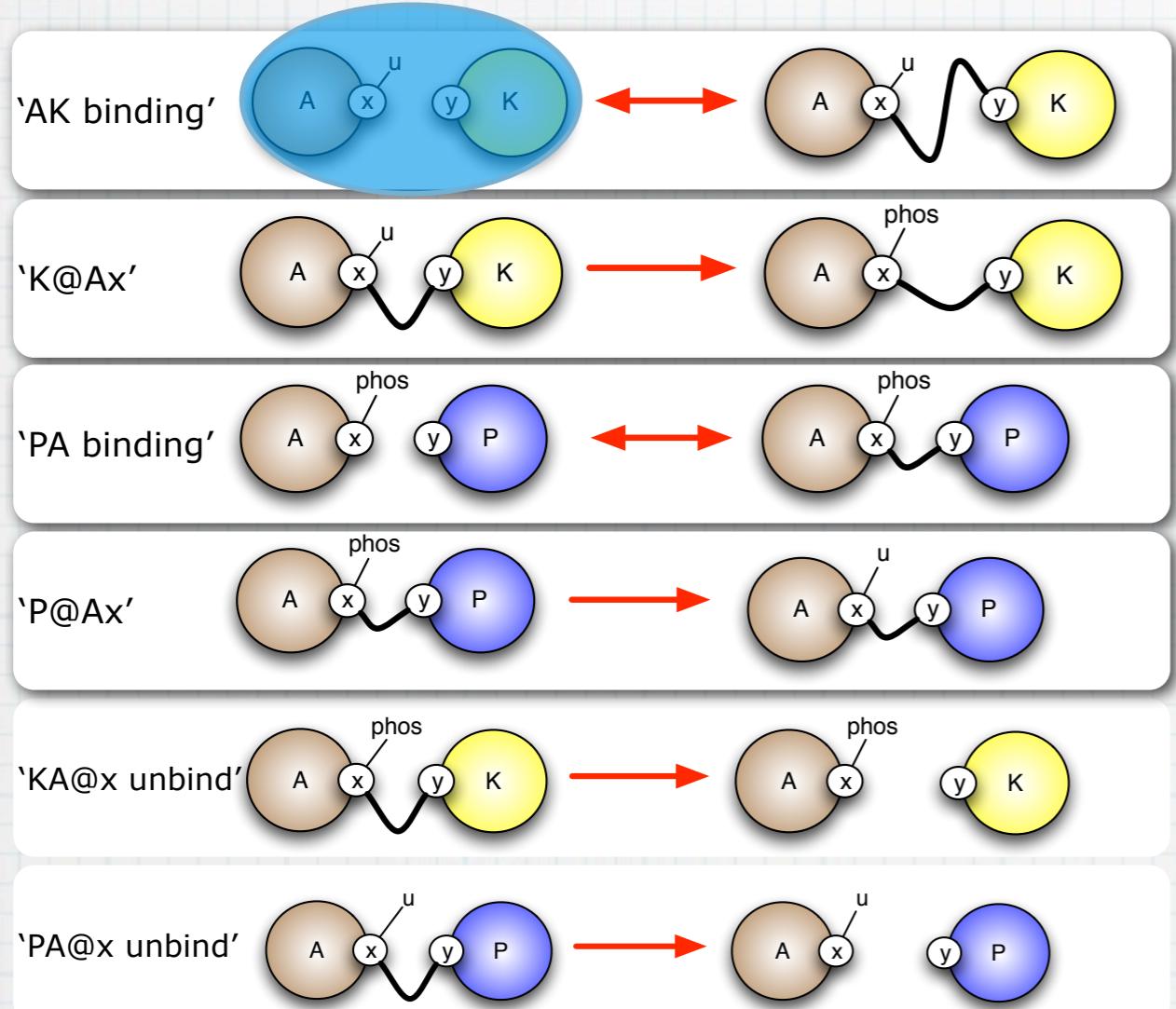
A global clock



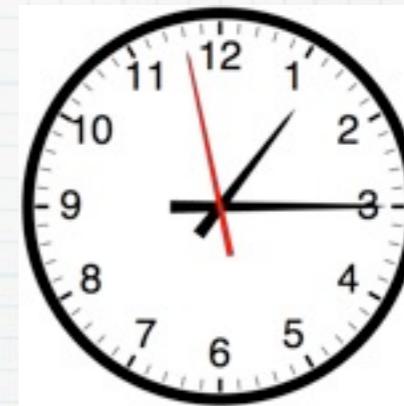
An initial solution: defines the starting state of the system.

1. Which rule can I apply next?

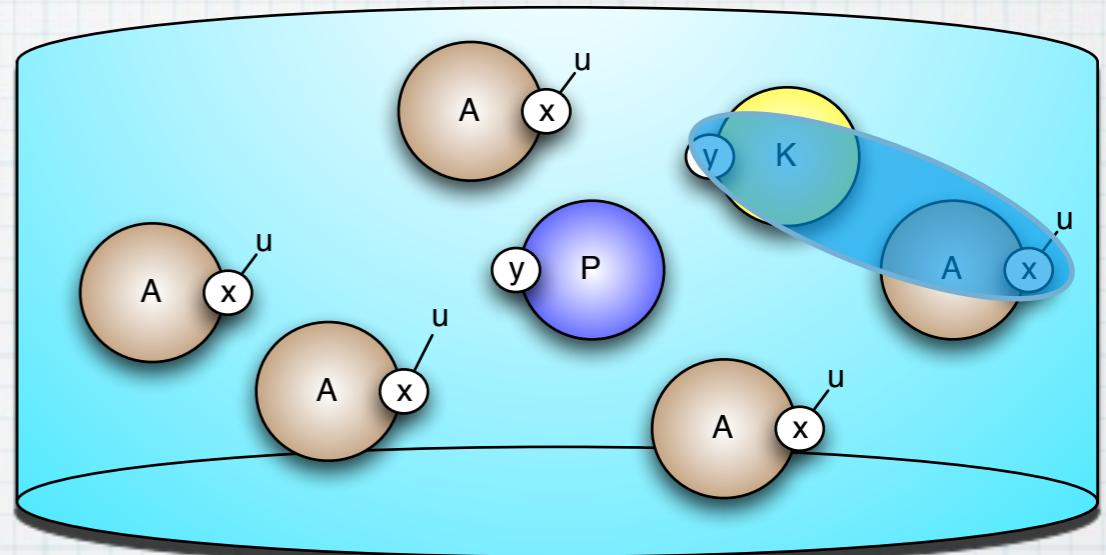
Simulation...



A rule set: defines the “laws” of the model



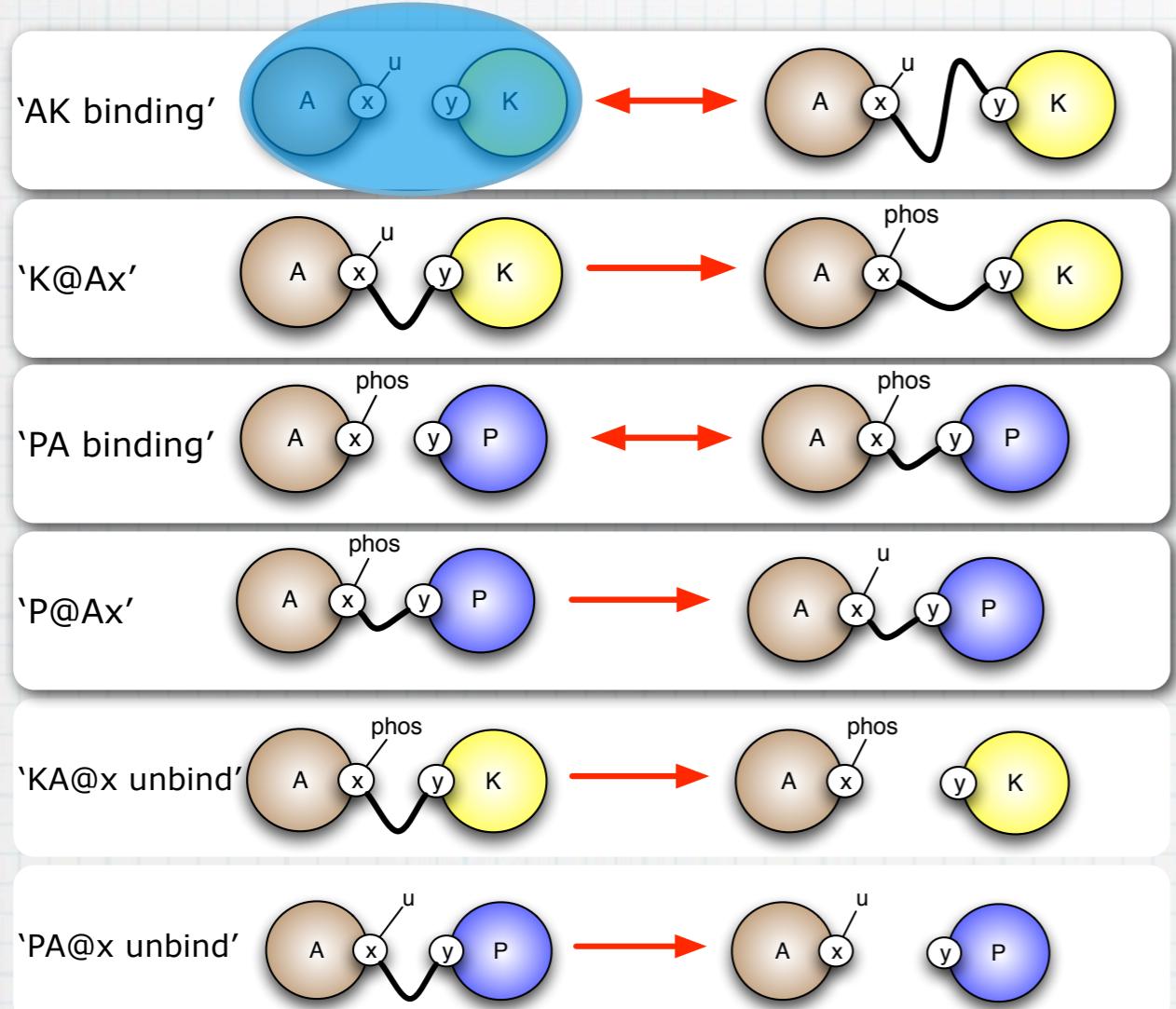
A global clock



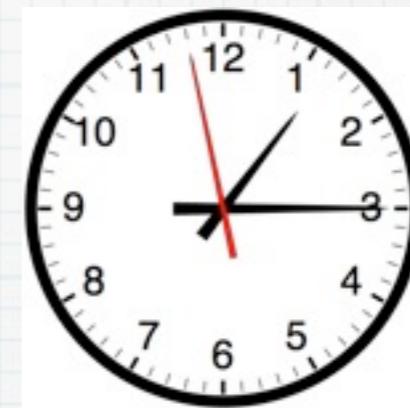
An initial solution: defines the starting state of the system.

1. Which rule can I apply next?

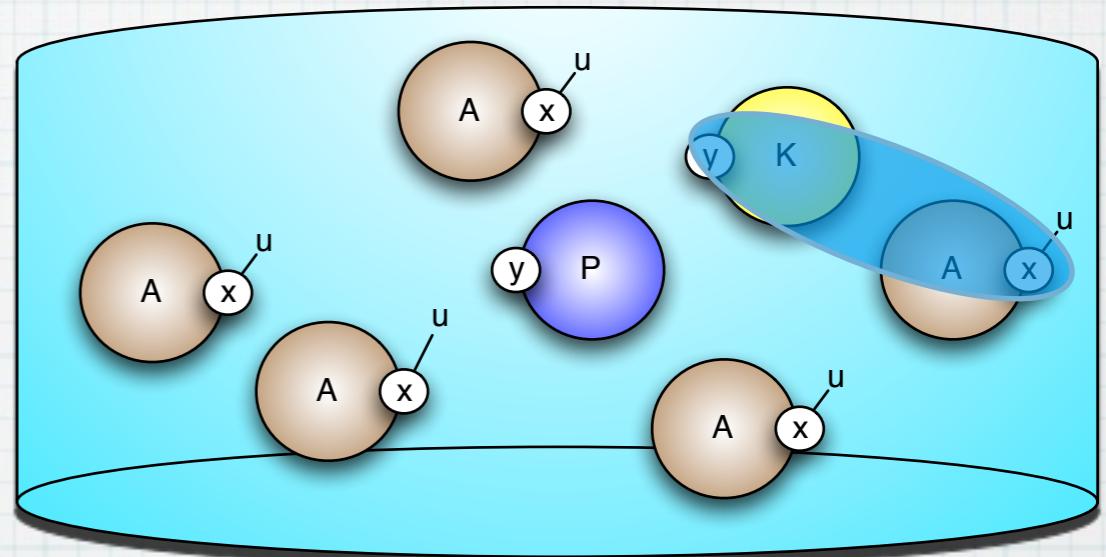
Simulation...



A rule set: defines the “laws” of the model



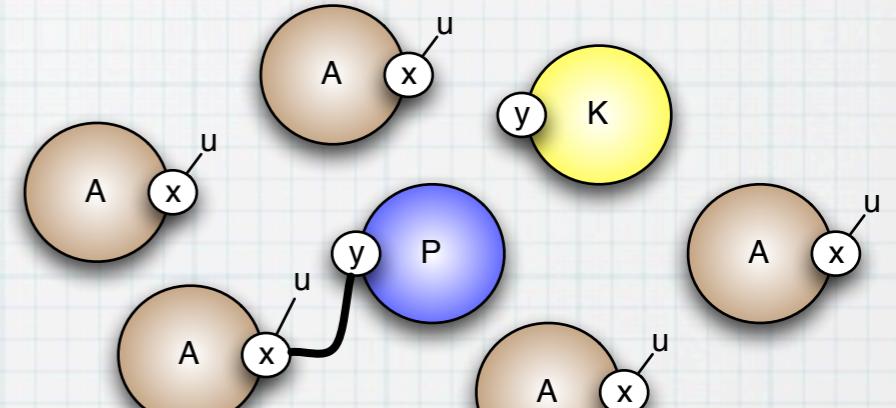
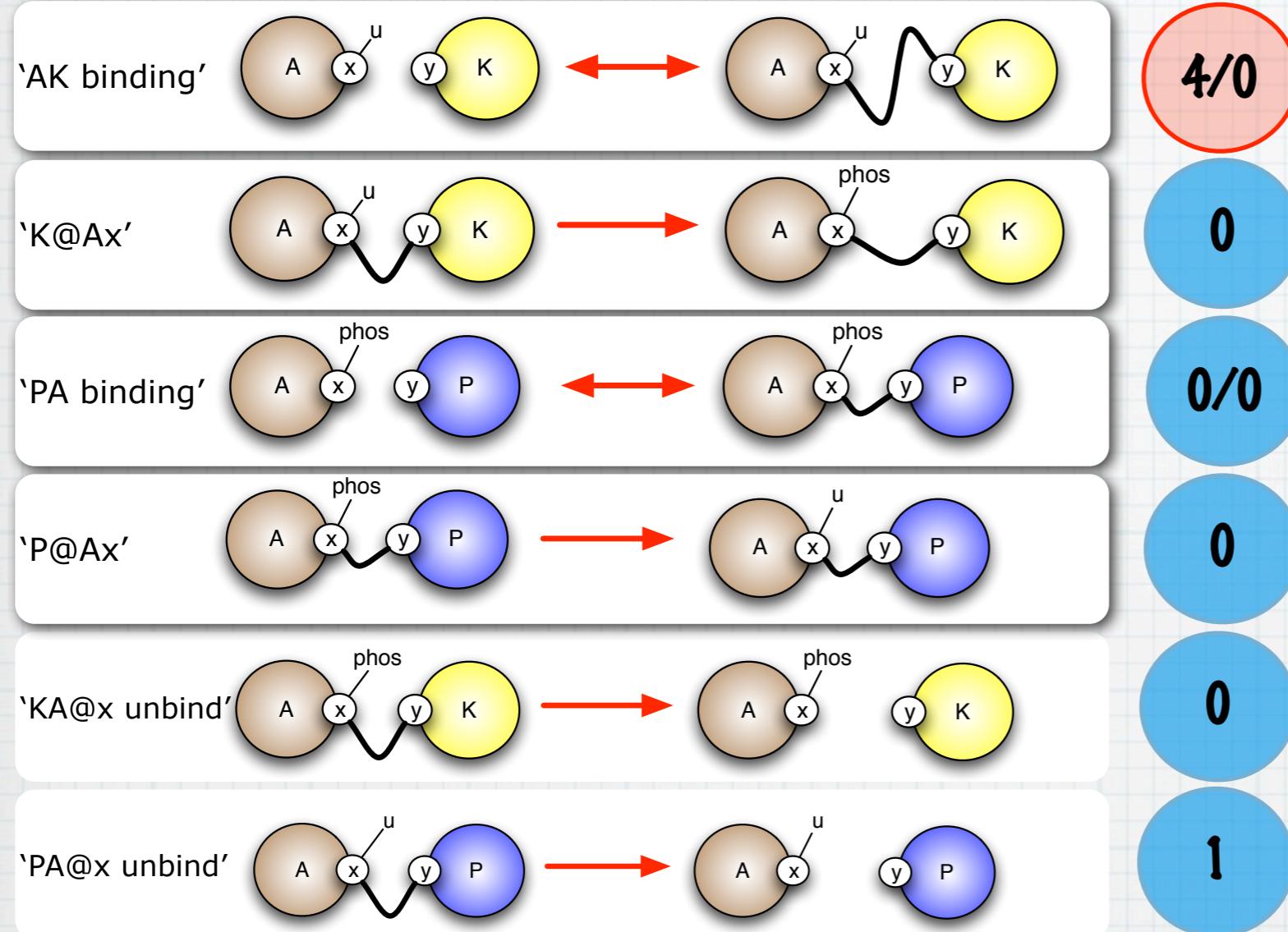
A global clock



An initial solution: defines the starting state of the system.

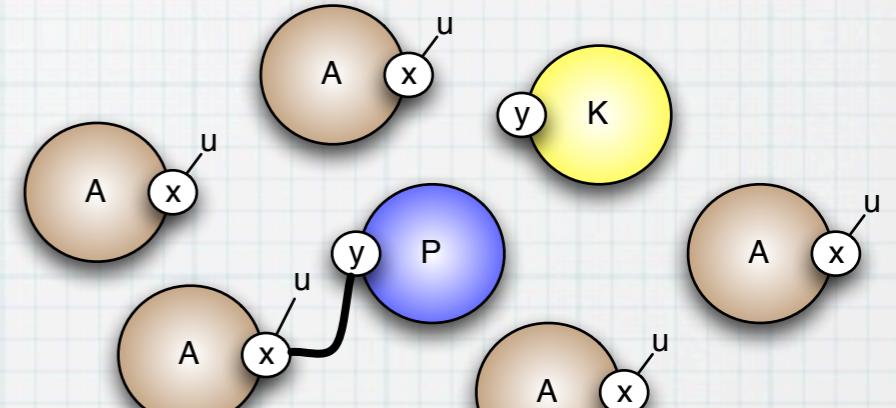
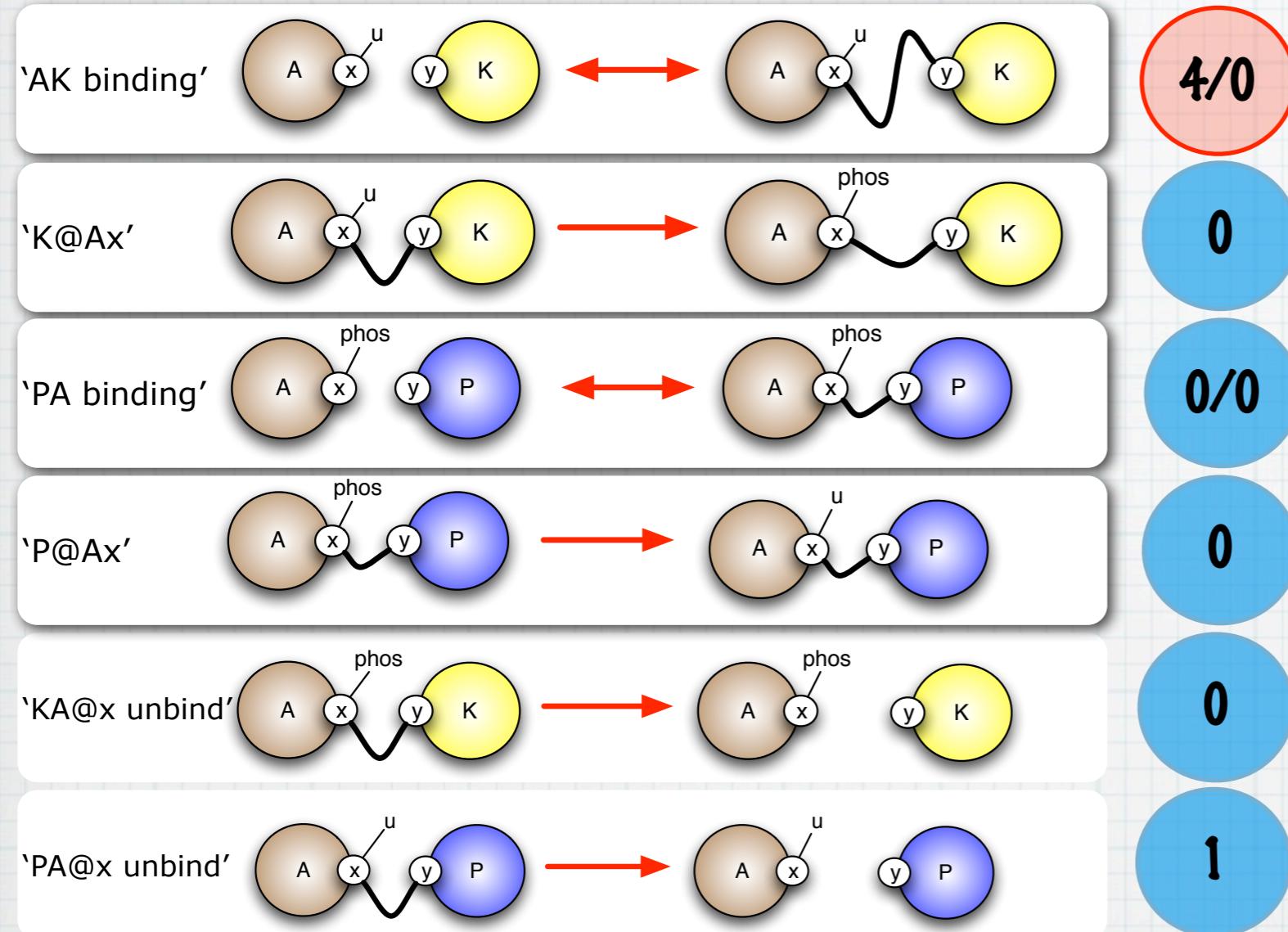
2. How long does it take to be observed?

Activity of a system



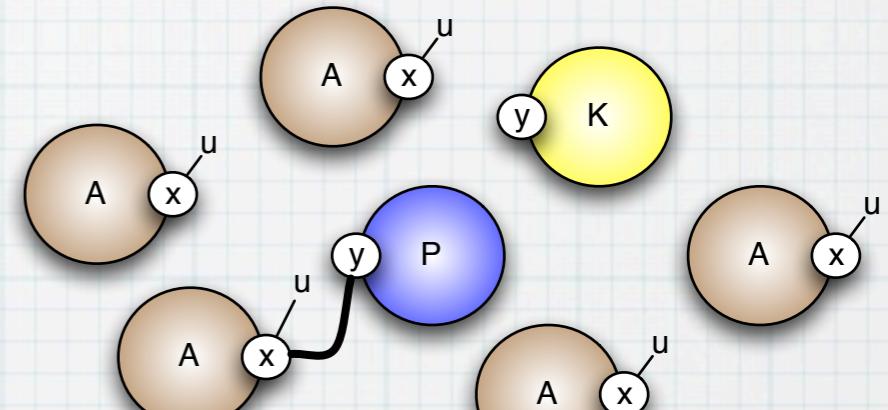
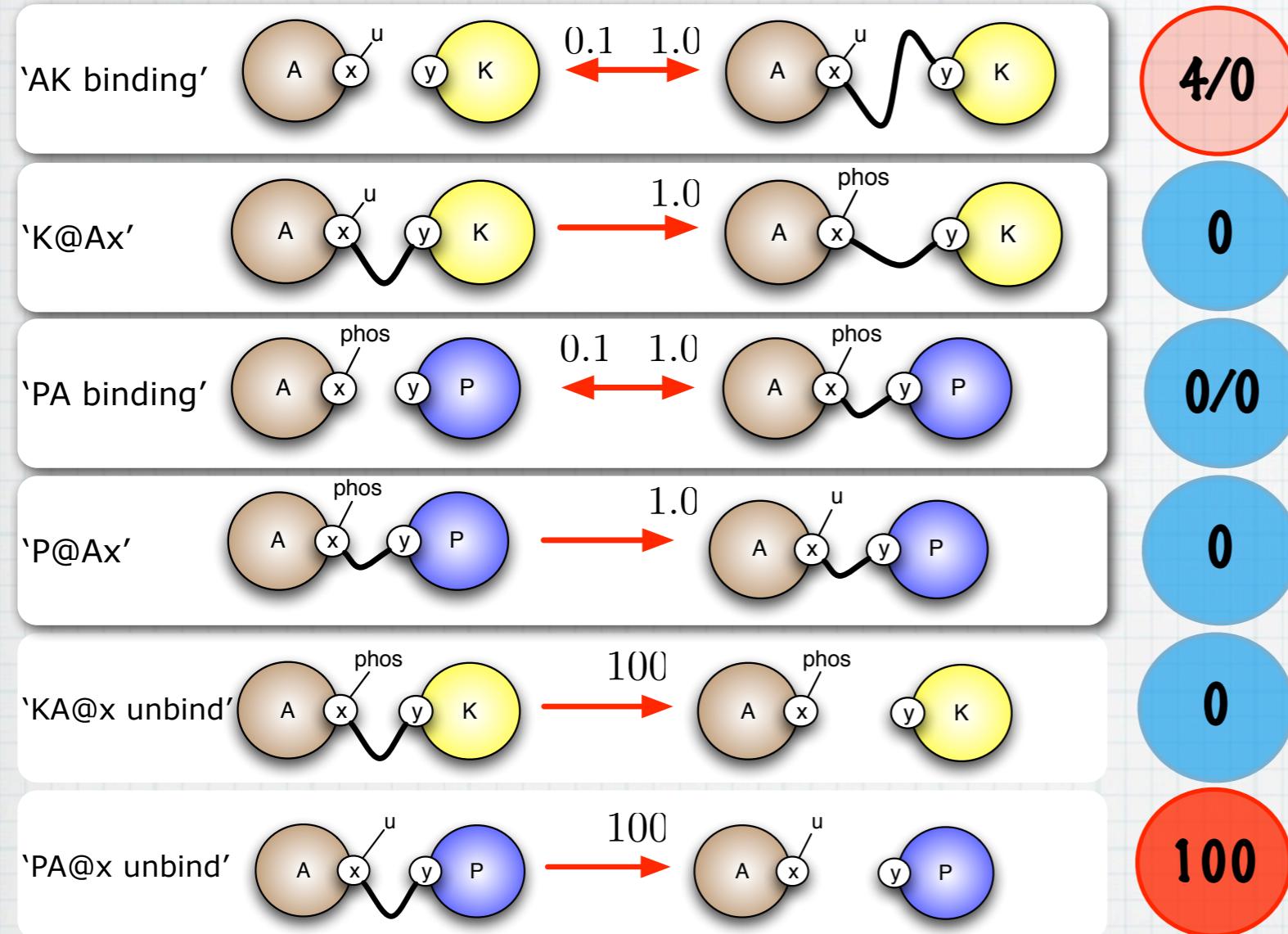
The activity of a rule in a solution S is the number of application it has in S

Activity of a system



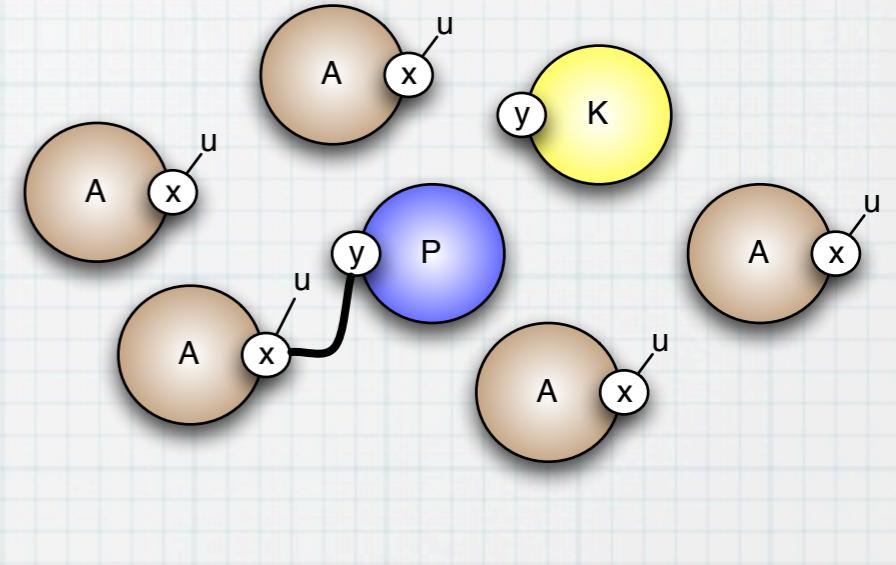
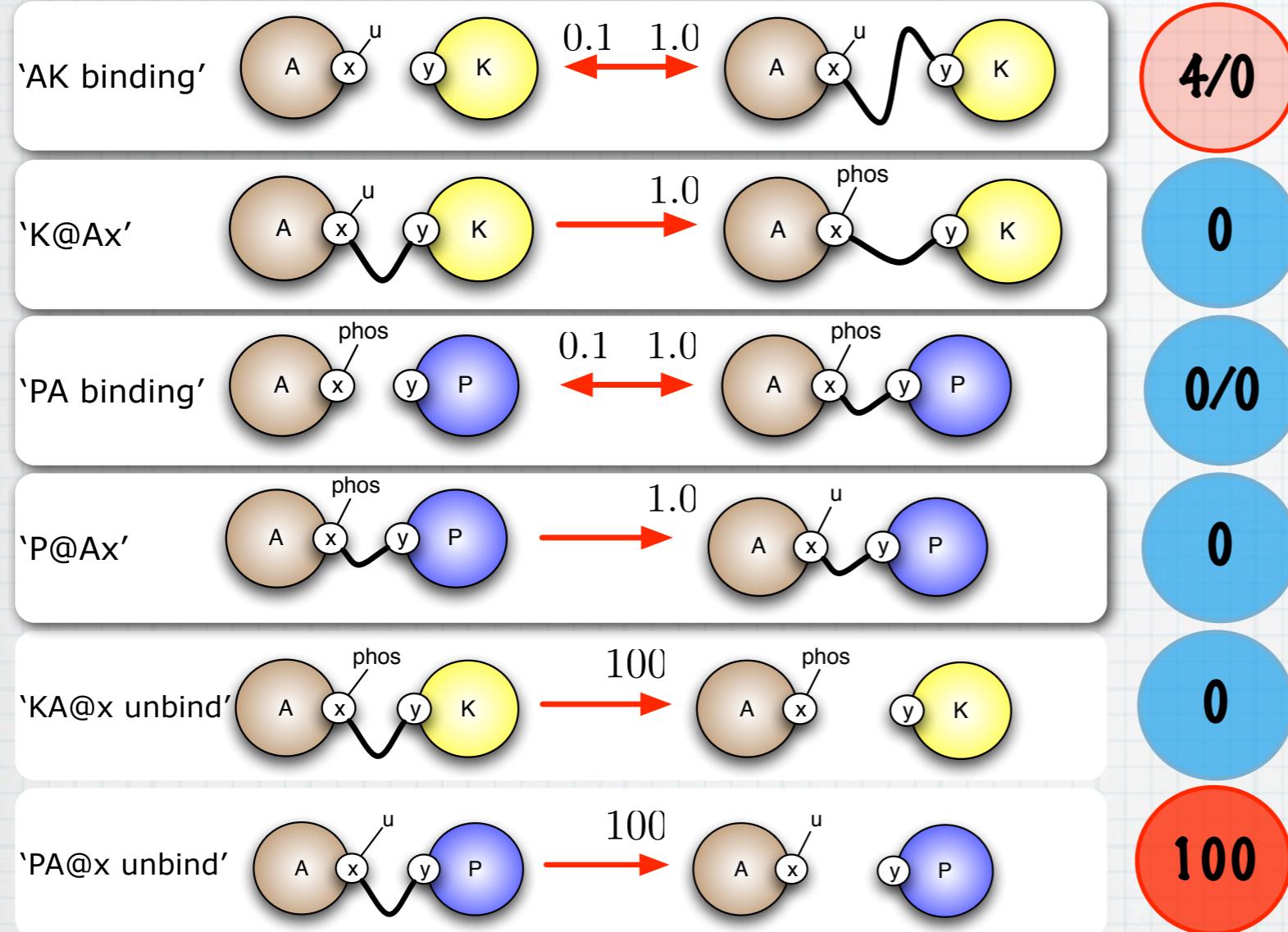
The activity of a rule in a solution S is the number of application it has in S times the rate of the rule

Activity of a system



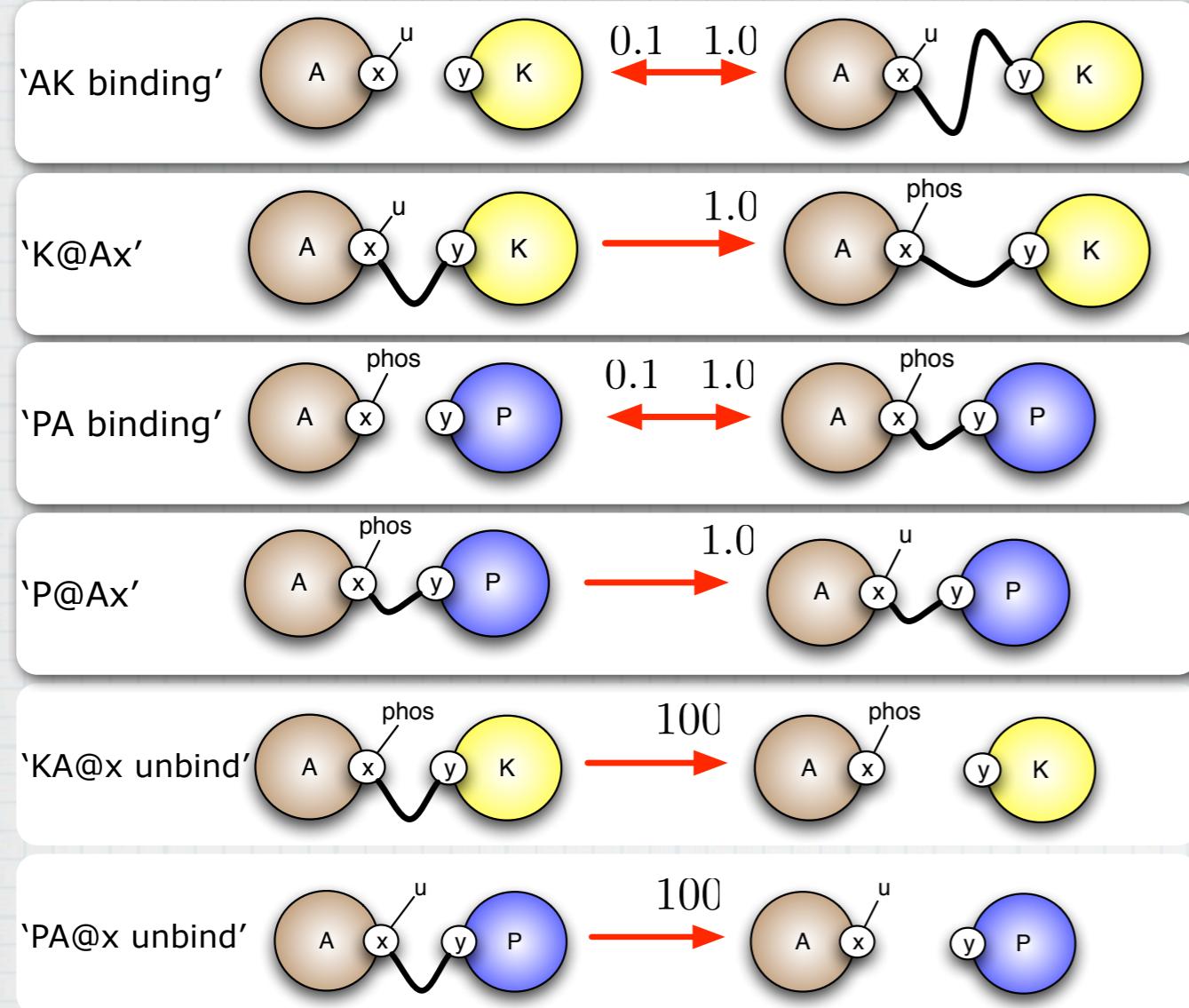
The activity of a rule in a solution S is the number of application it has in S times the rate of the rule

Activity of a system



The global activity of the system is the sum of rule activities

Activity of a system



4/0

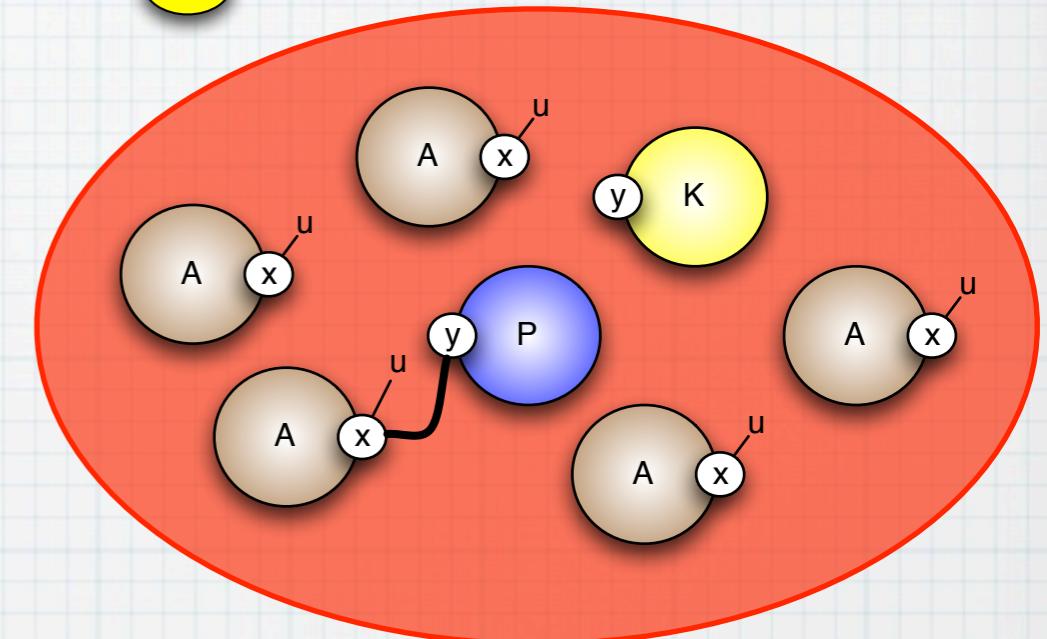
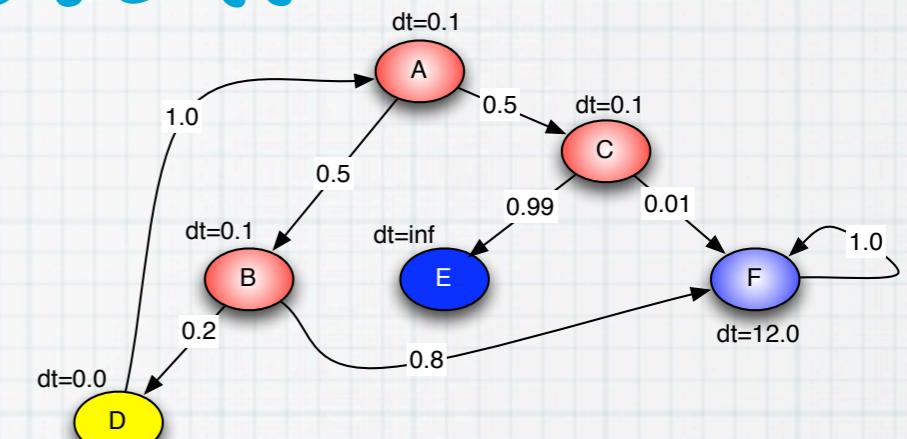
0

0/0

0

0

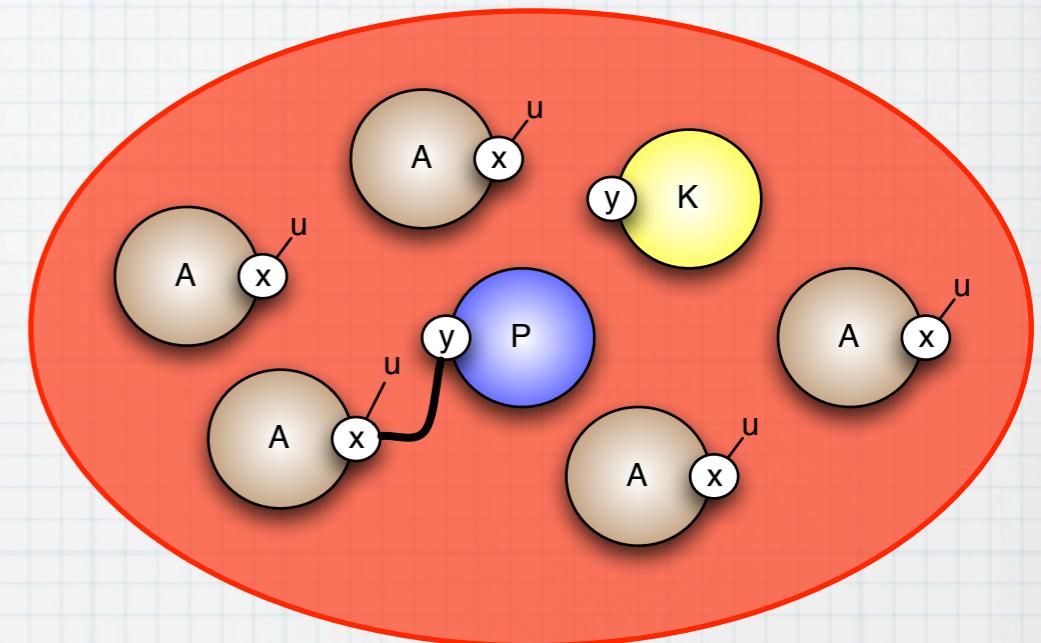
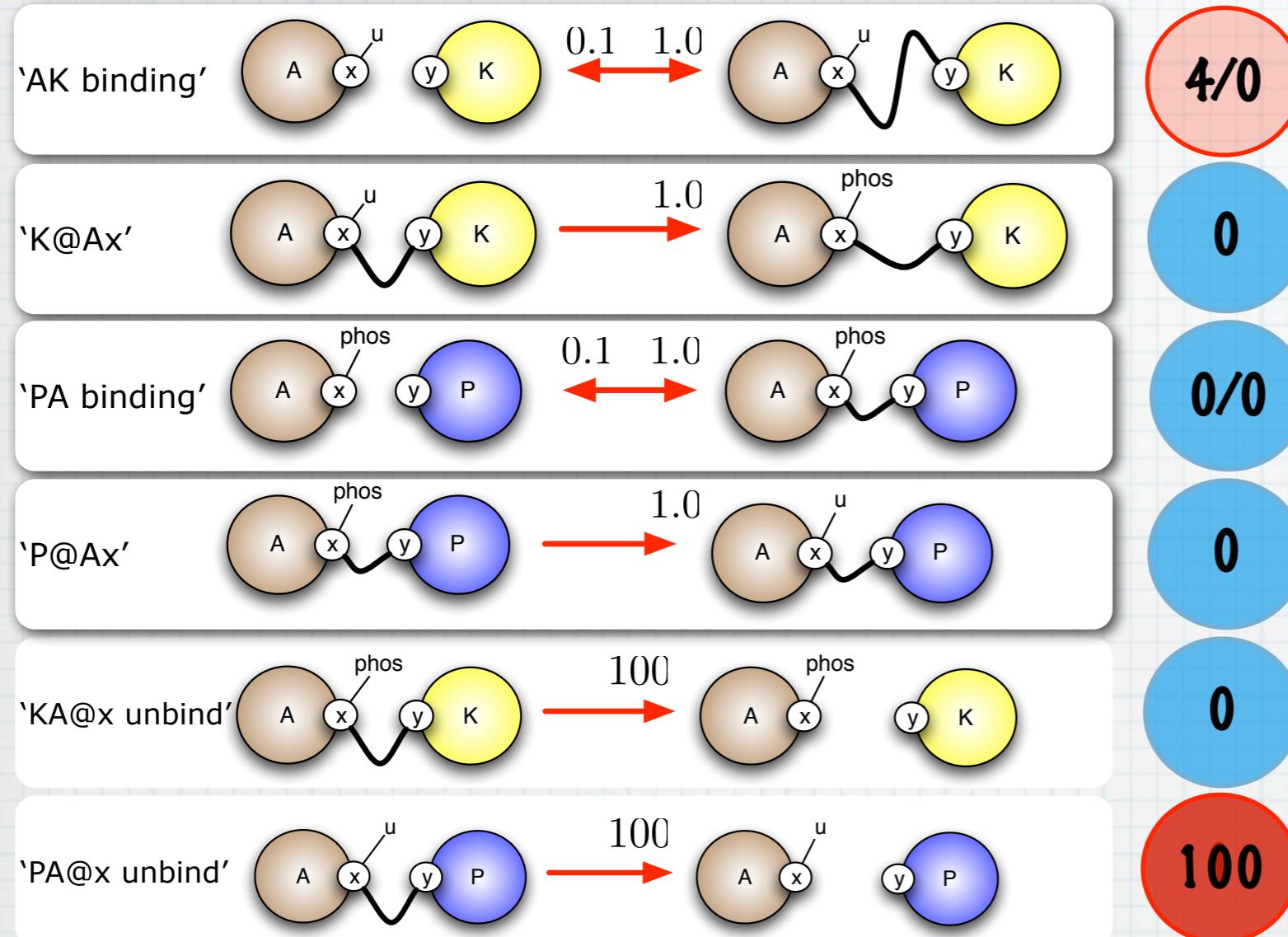
100



$$A = 104$$

The global activity of the system is the sum of rule activities

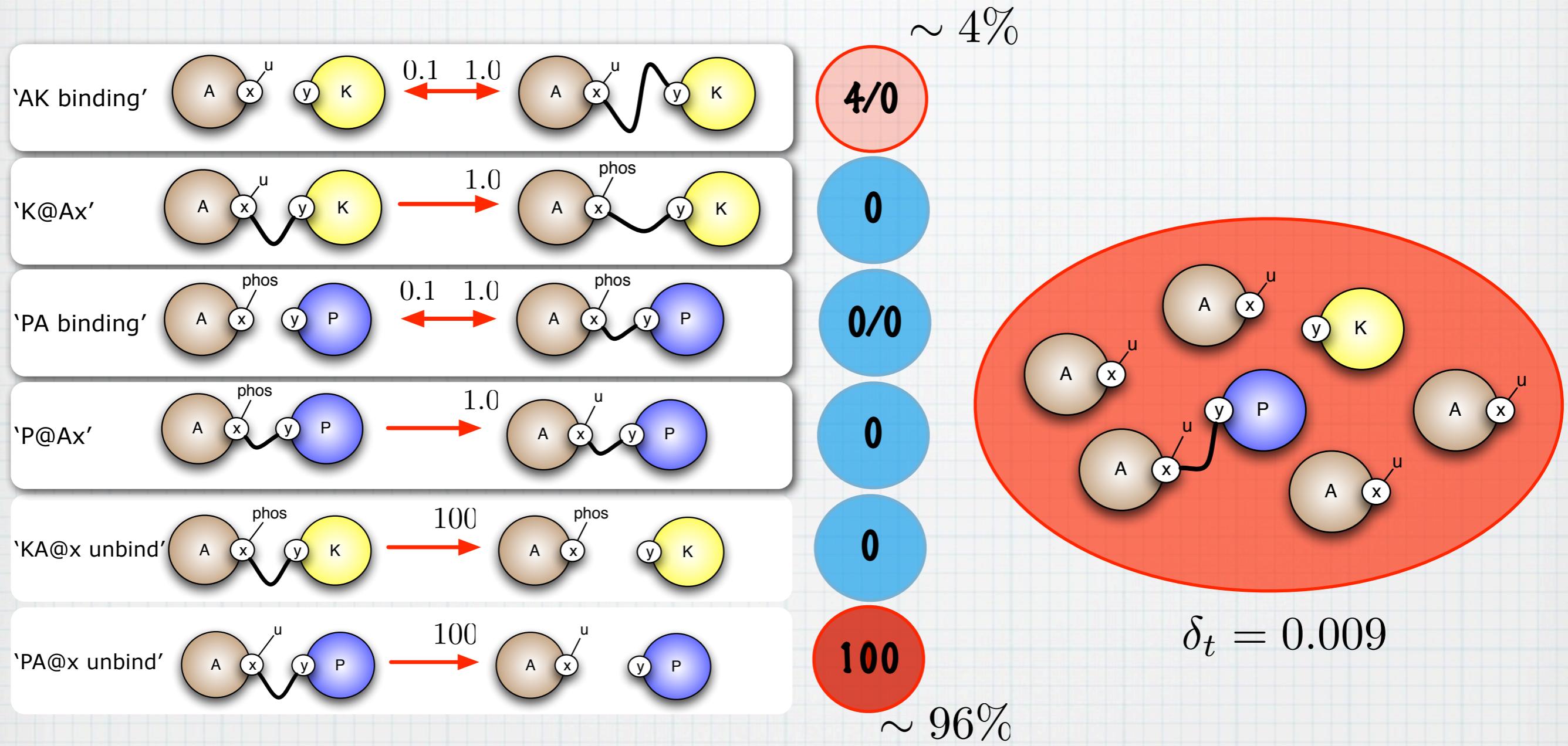
Application proba and time advance



$$\delta_t = \text{rand}[0, e^{-A}] \sim 1/A$$

$$p(r) = A_r / A$$

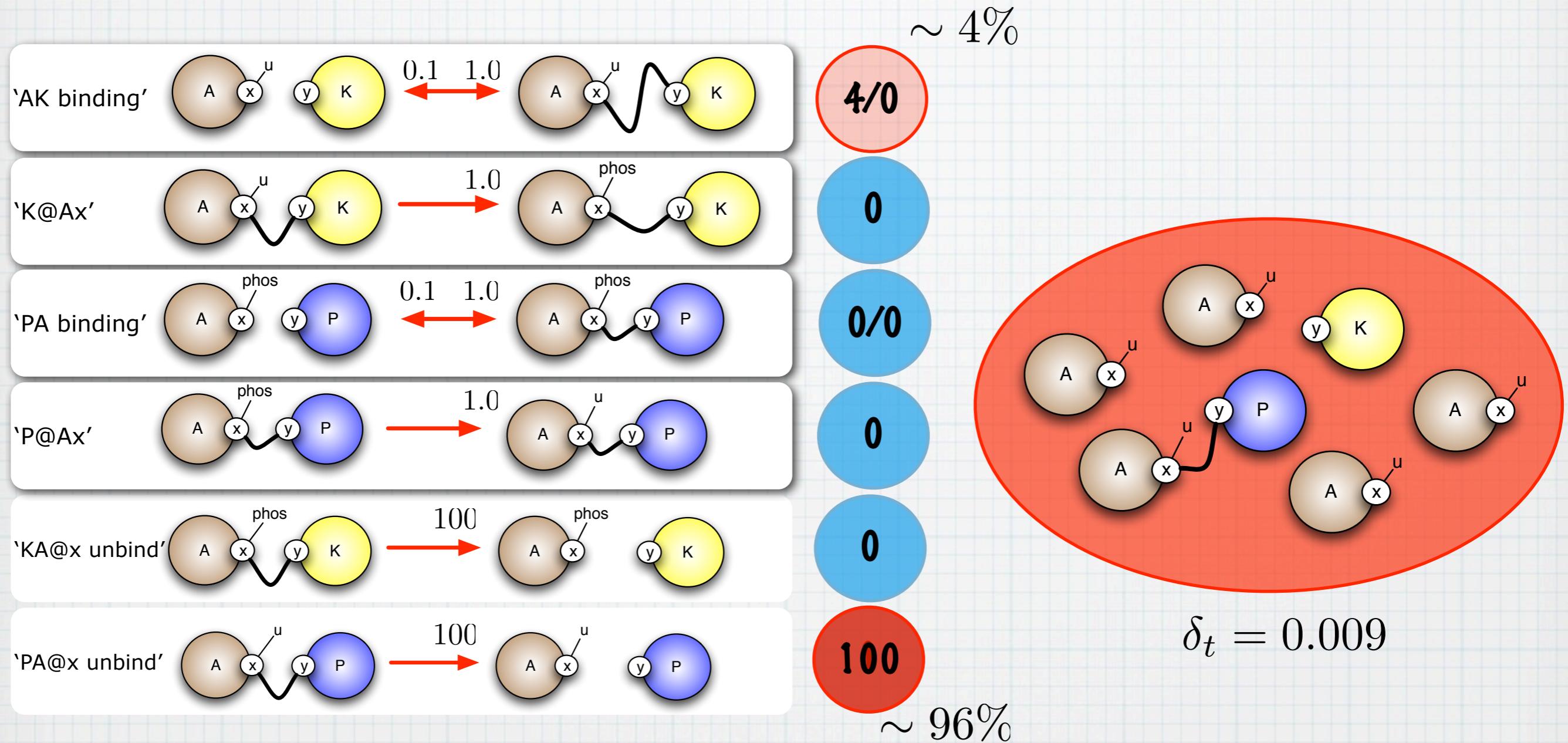
Application proba and time advance



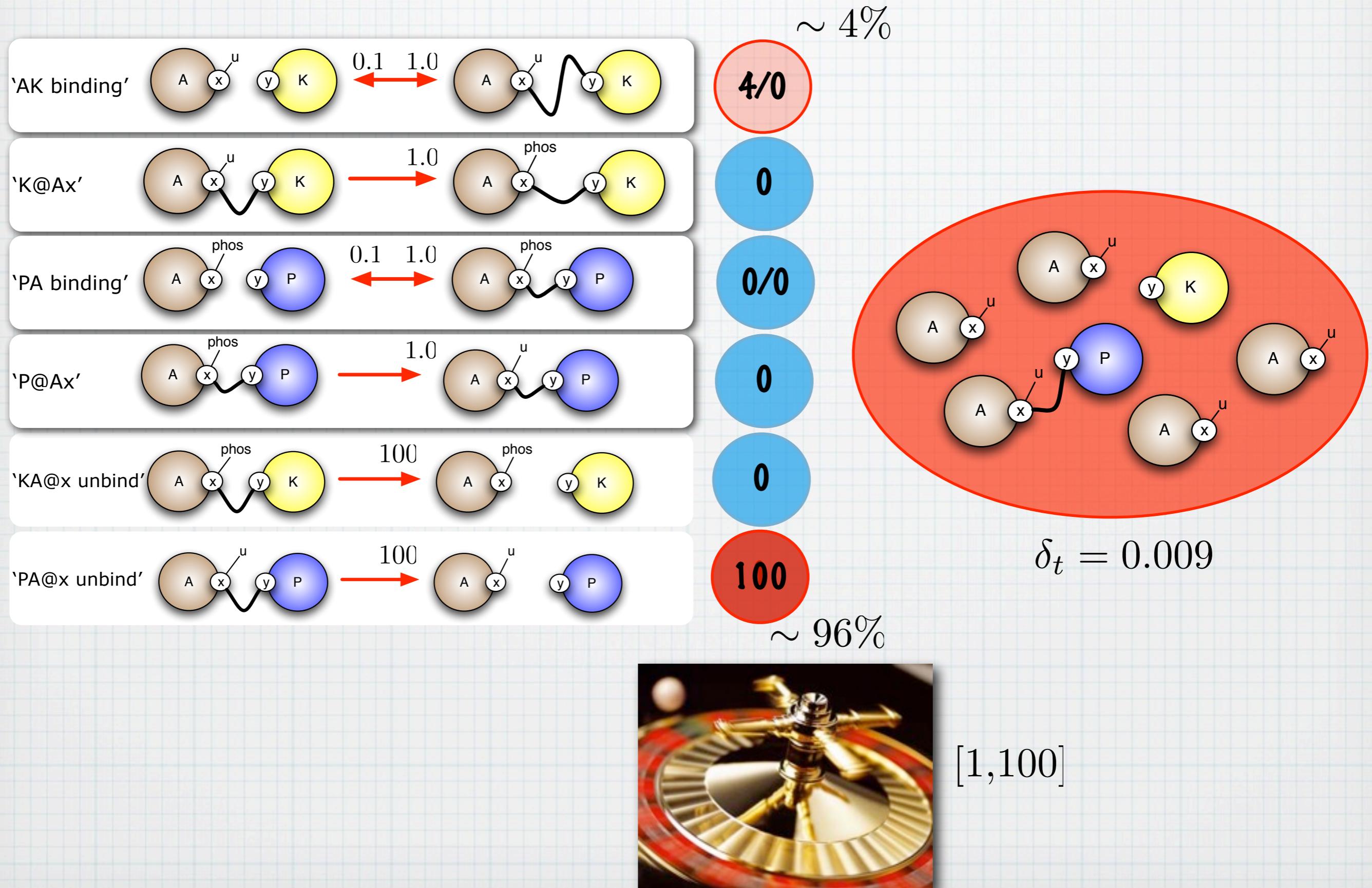
$$\delta_t = \text{rand}[0, e^{-A}] \sim 1/A$$

$$p(r) = A_r / A$$

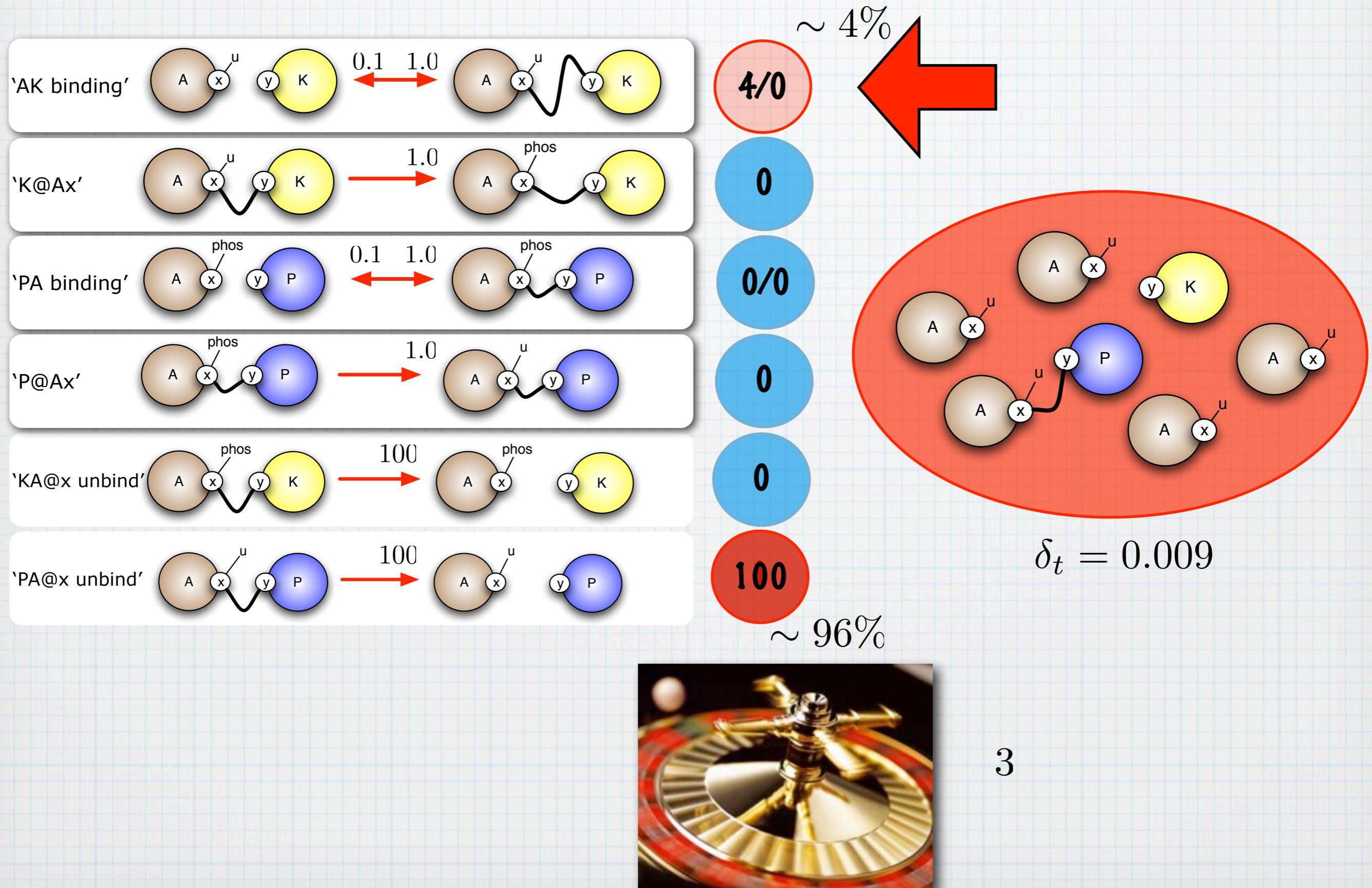
Choice of rule instance



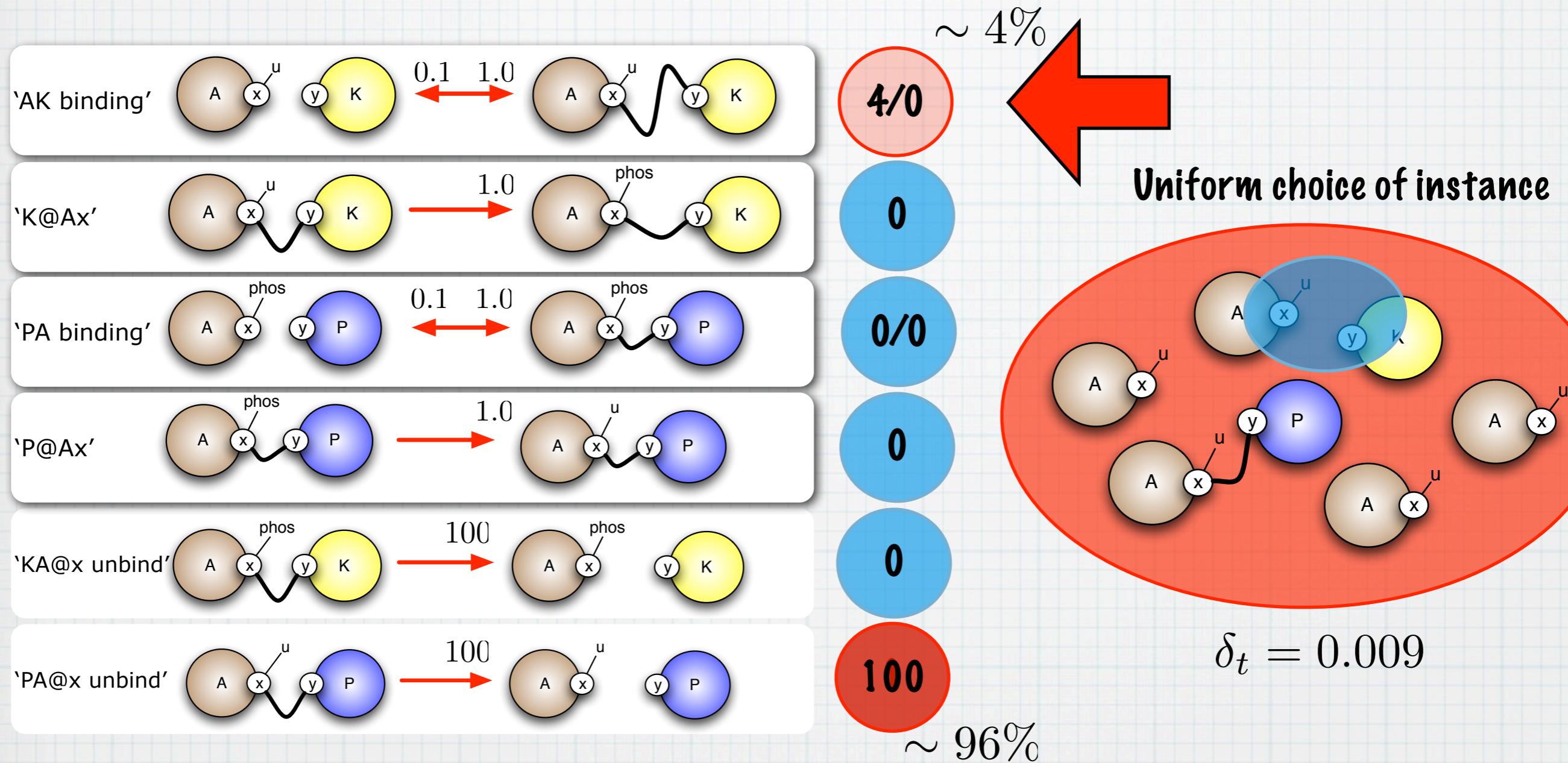
Choice of rule instance



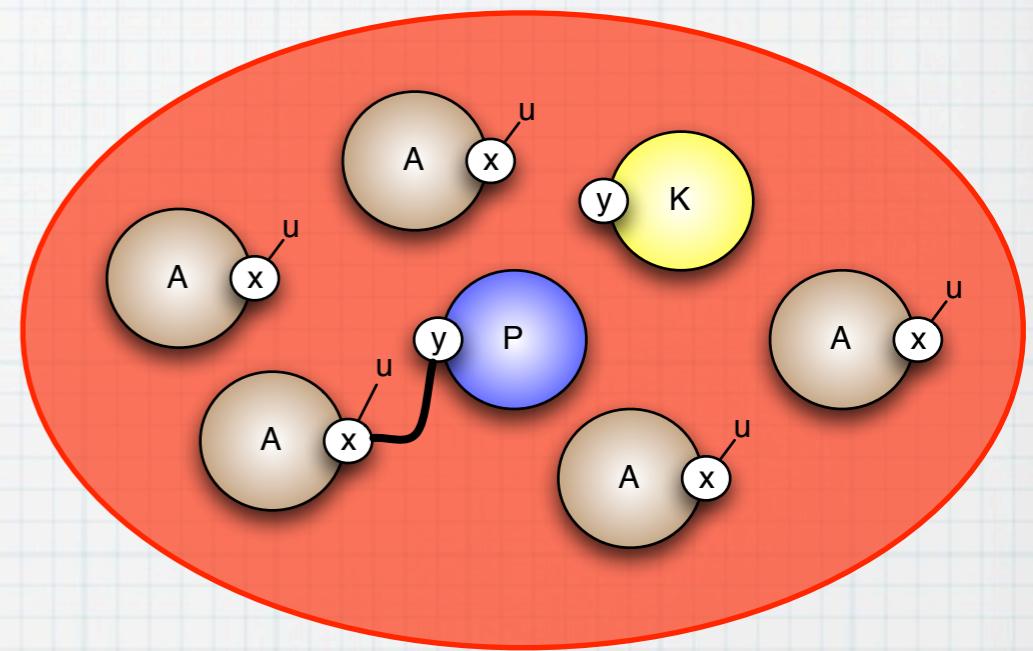
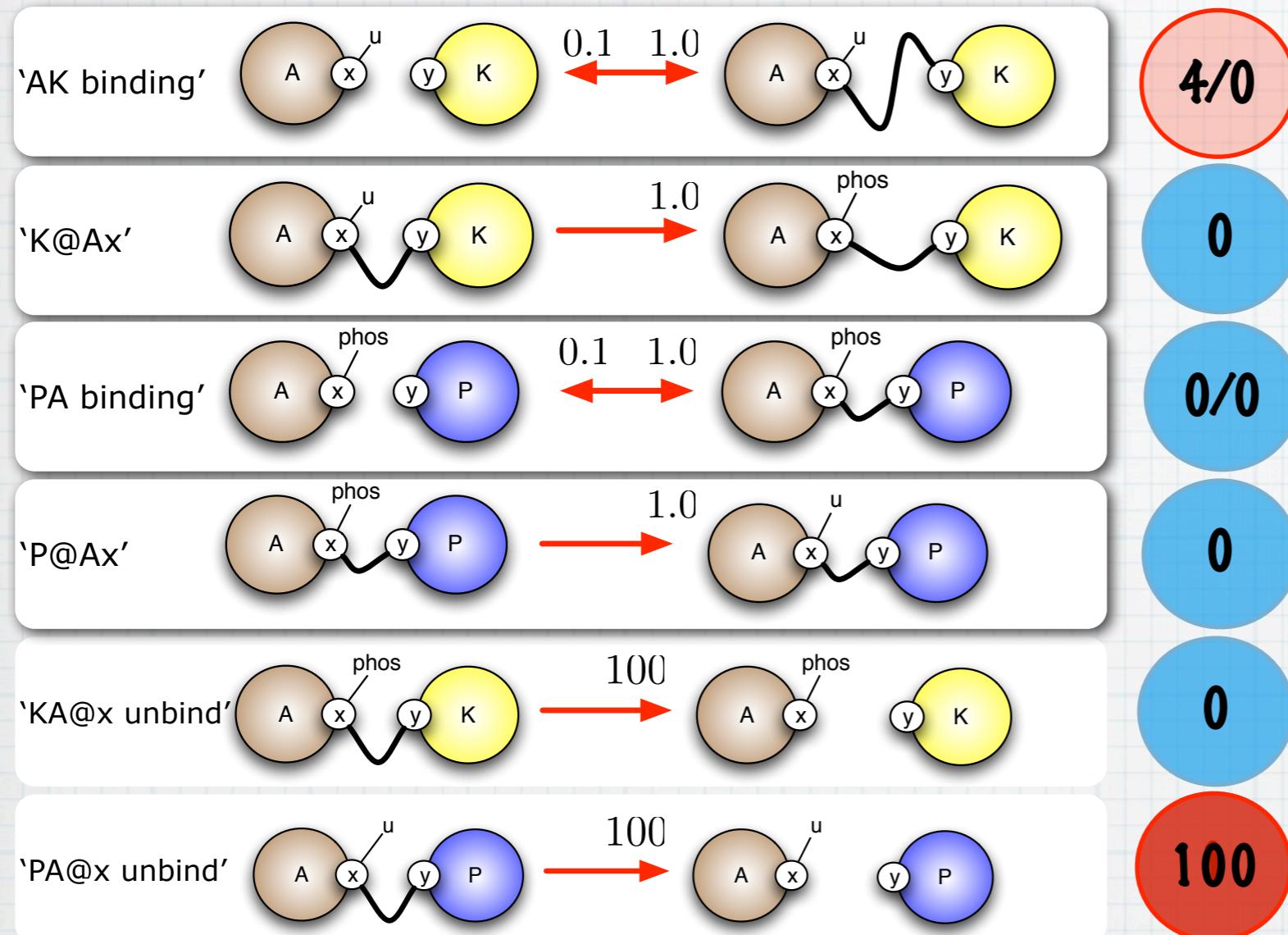
Choice of rule instance



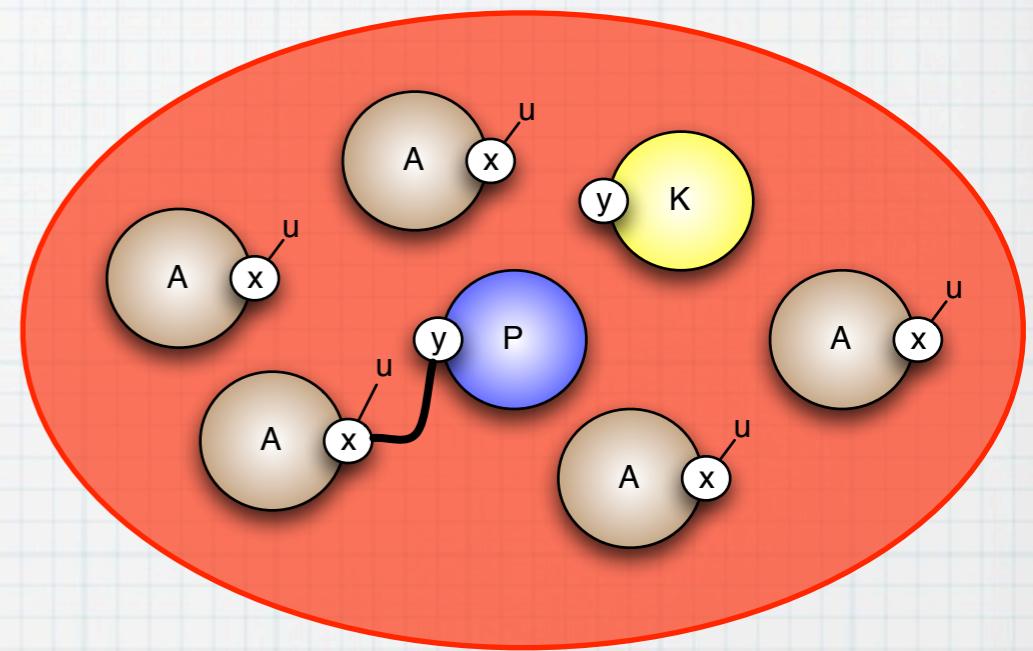
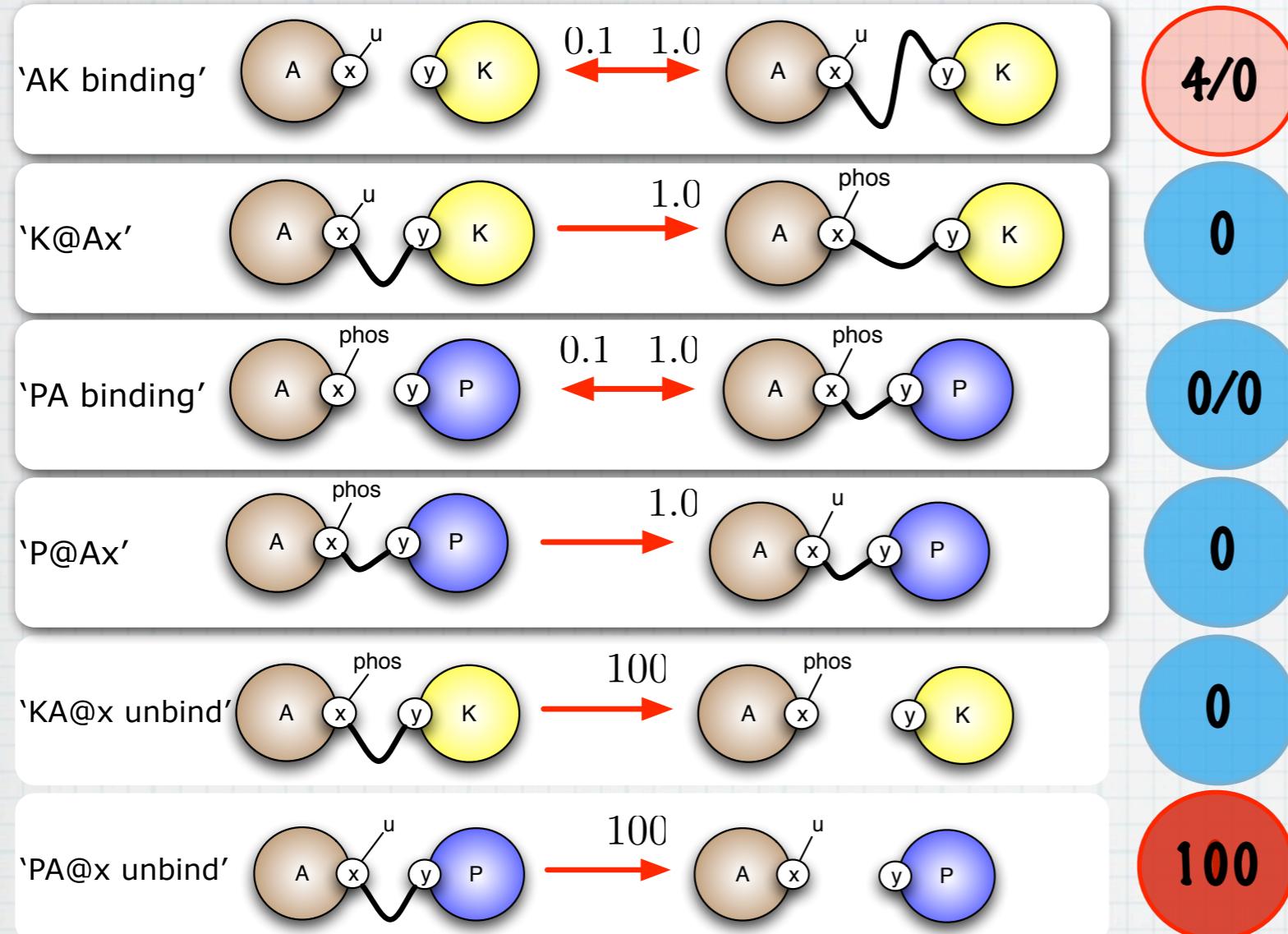
Choice of rule instance



System update



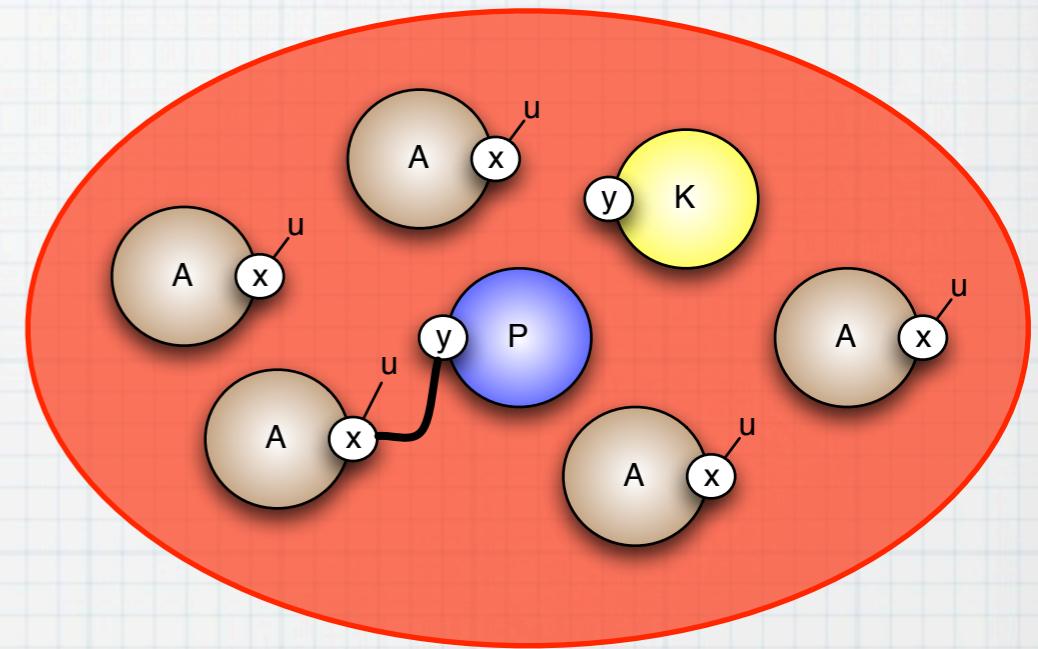
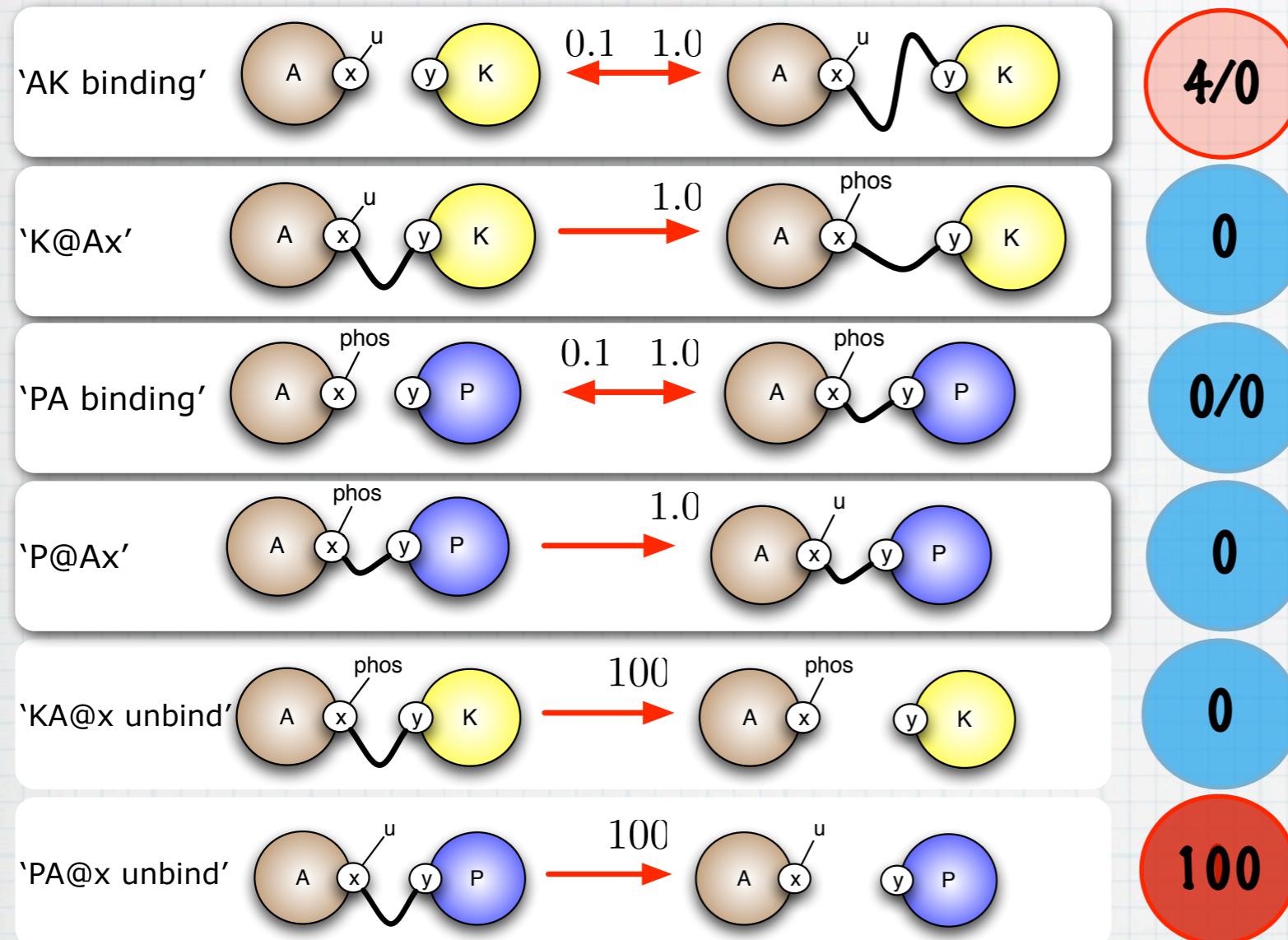
System update



$$\delta_t = 0.009$$

1. Advance time

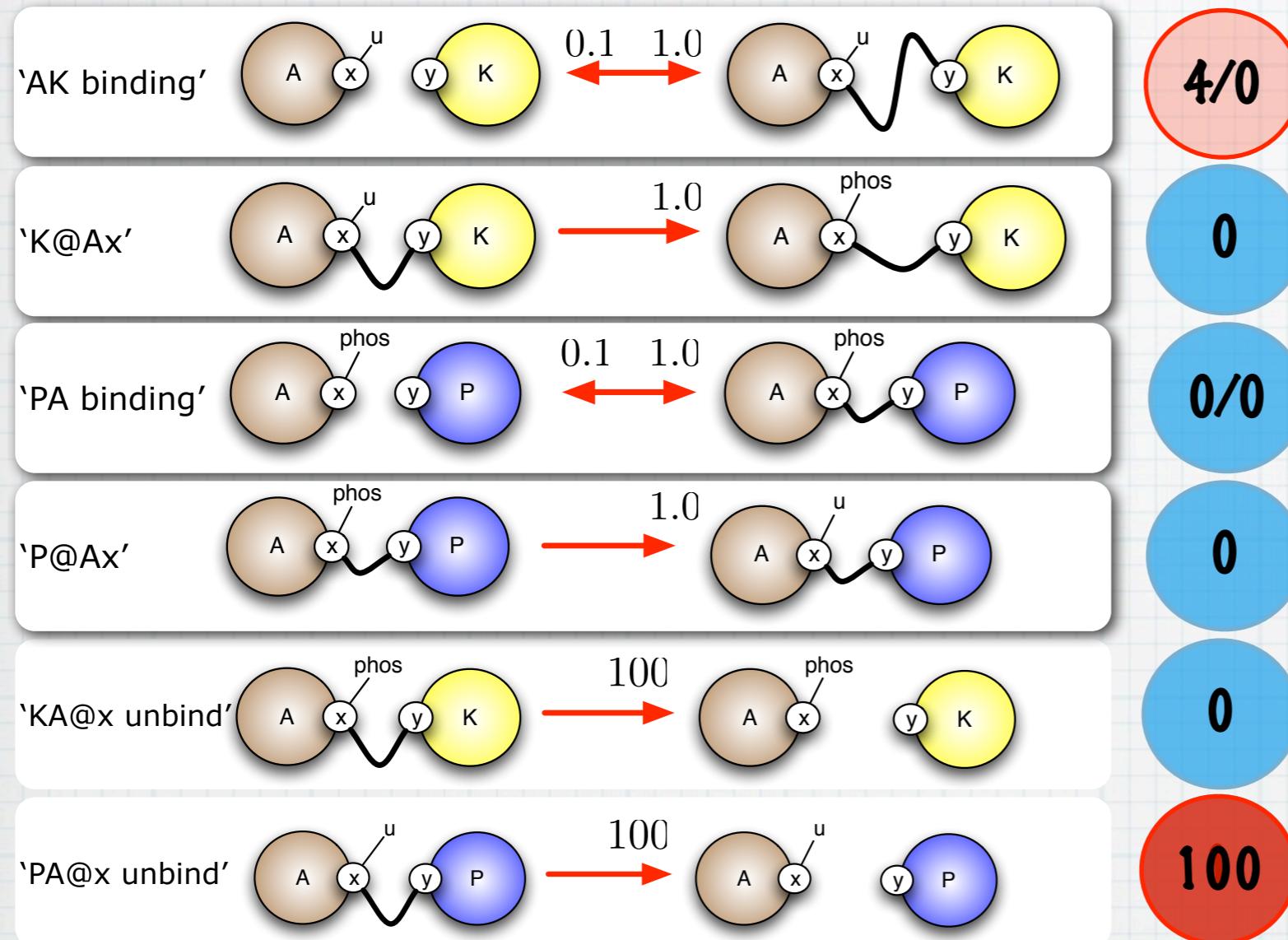
System update



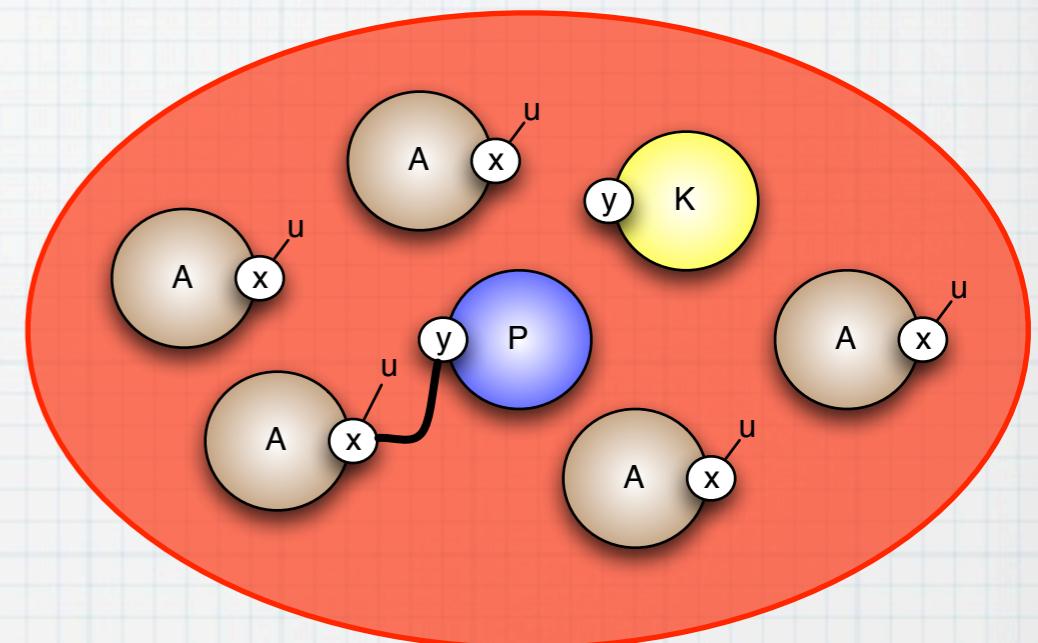
$$\delta_t = 0.009$$

1. Advance time

System update

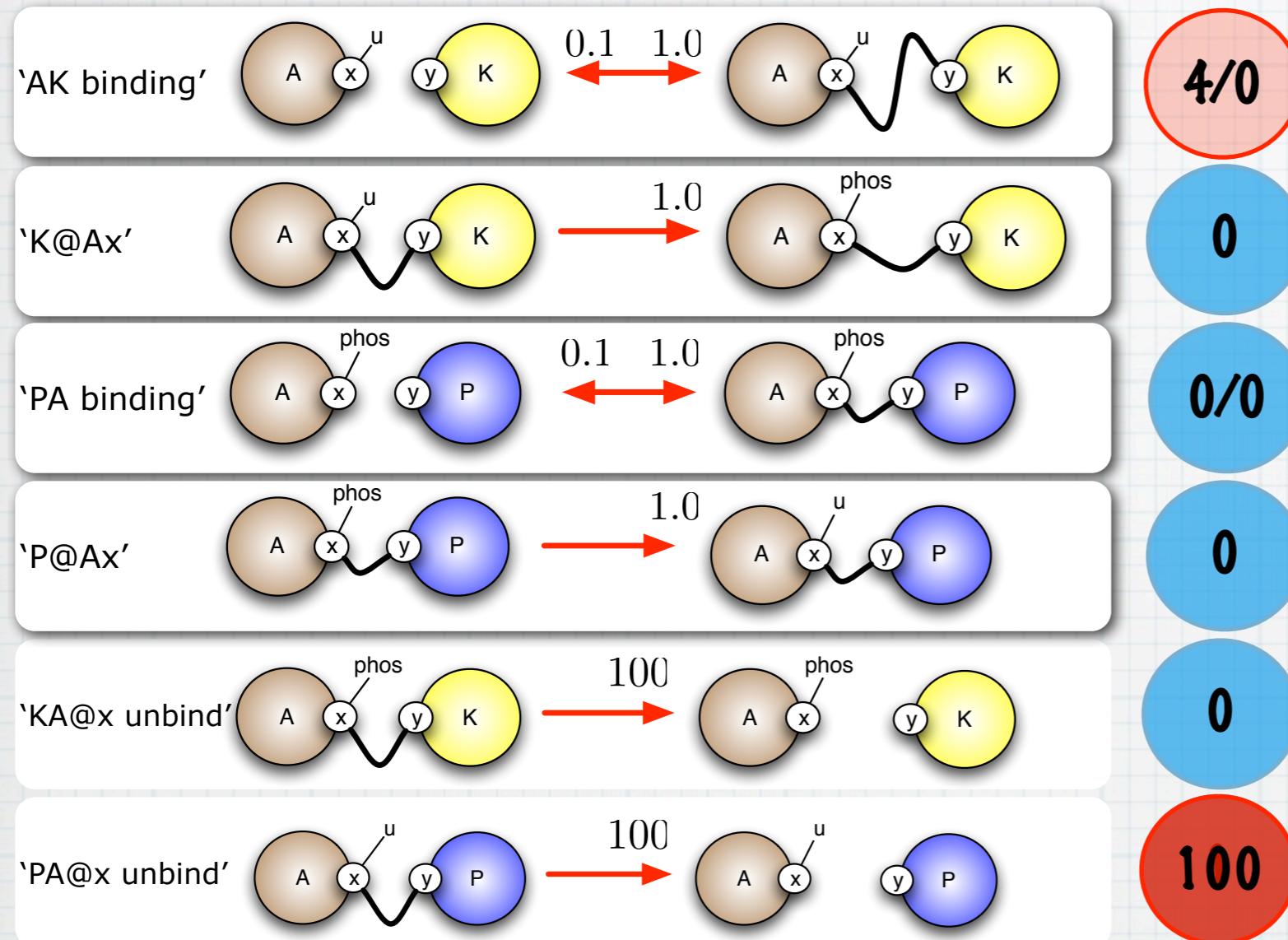


+ 0.009

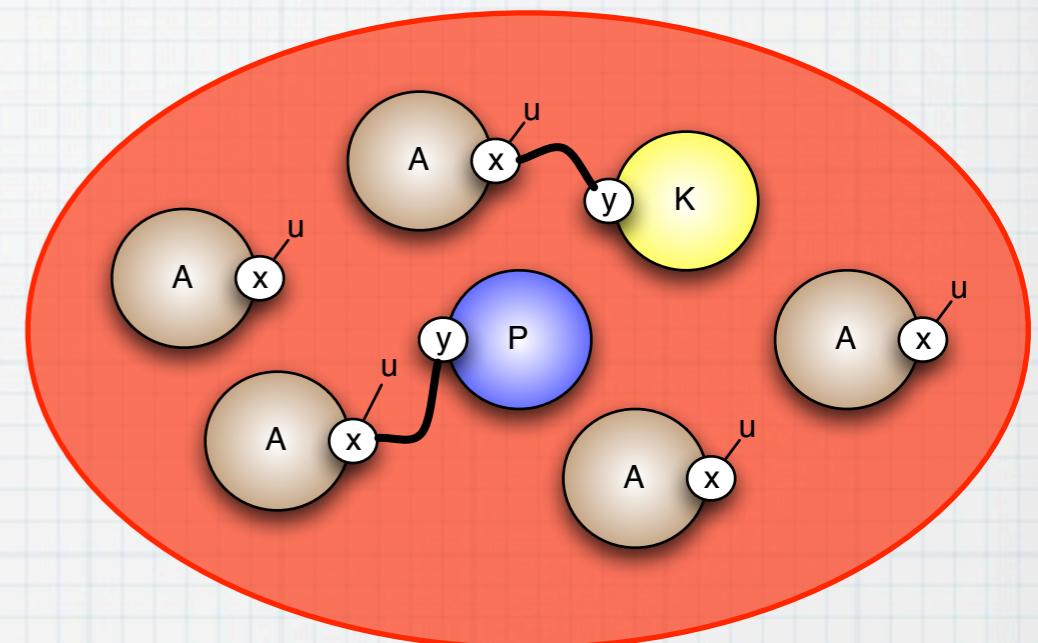


1. Advance time
2. Apply rule instance

System update

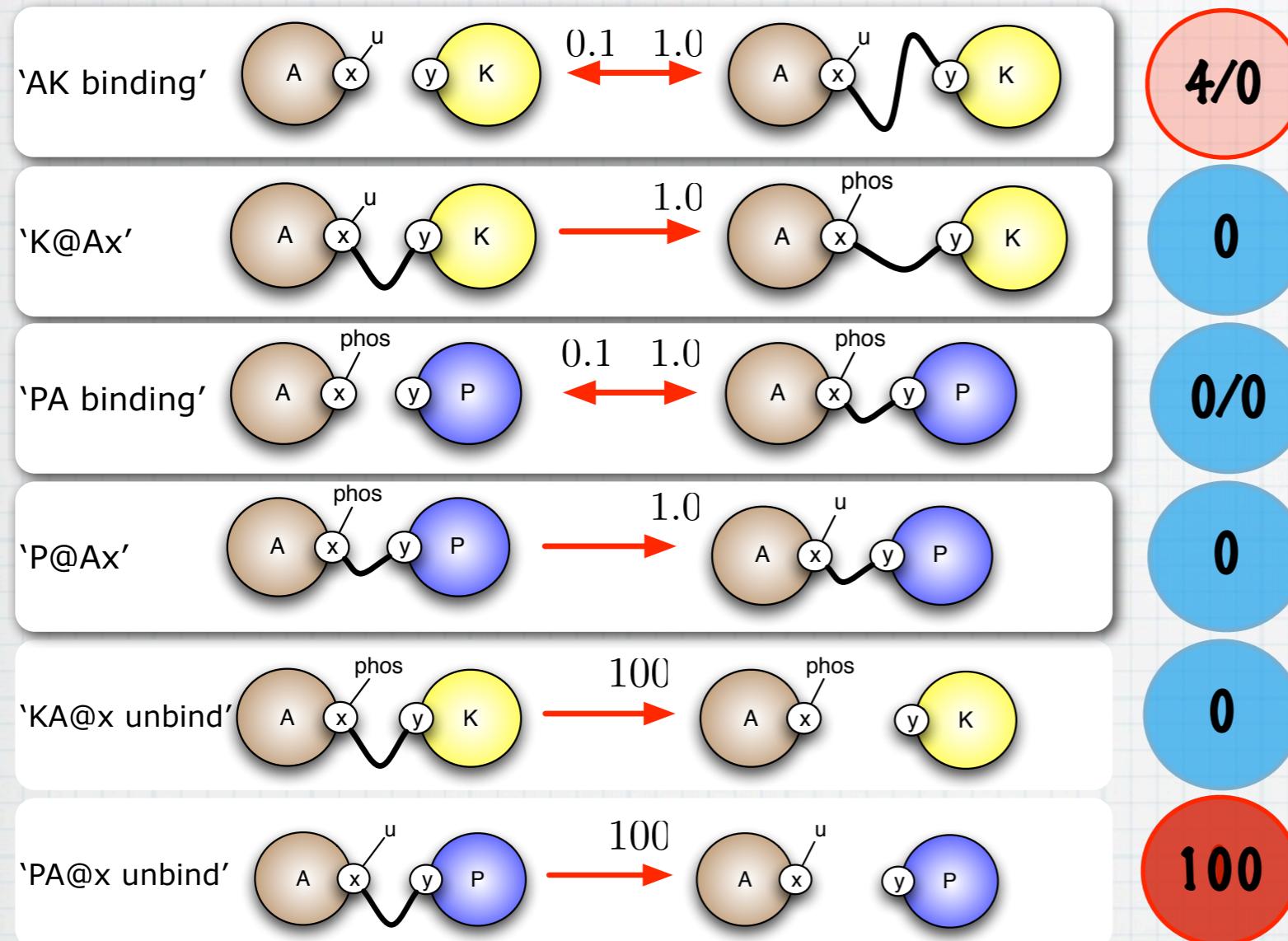


+ 0.009

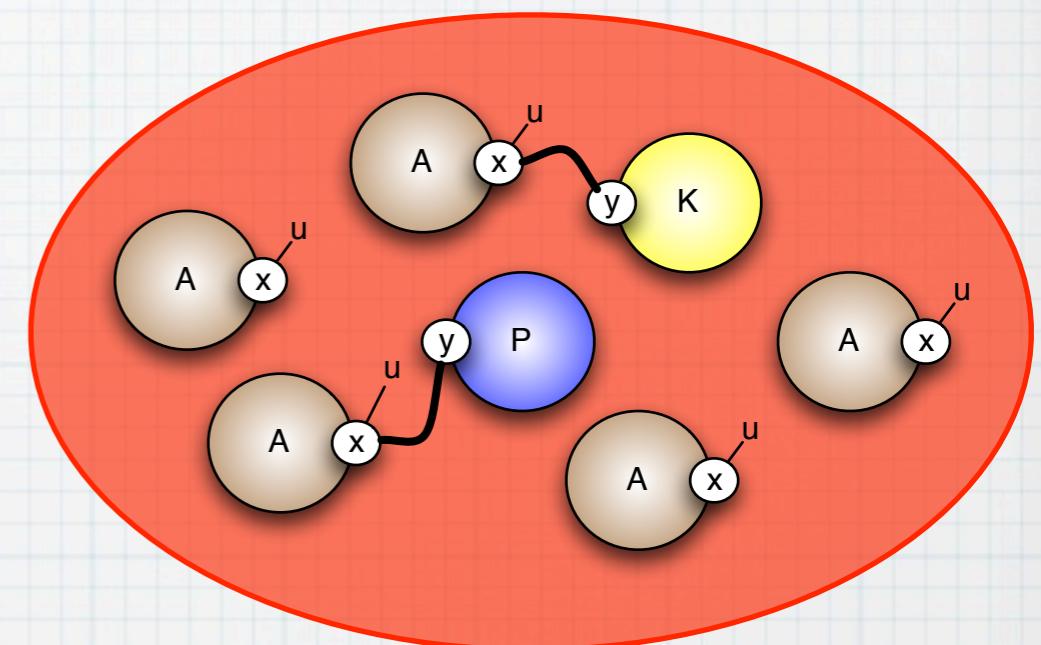


1. Advance time
2. Apply rule instance

System update

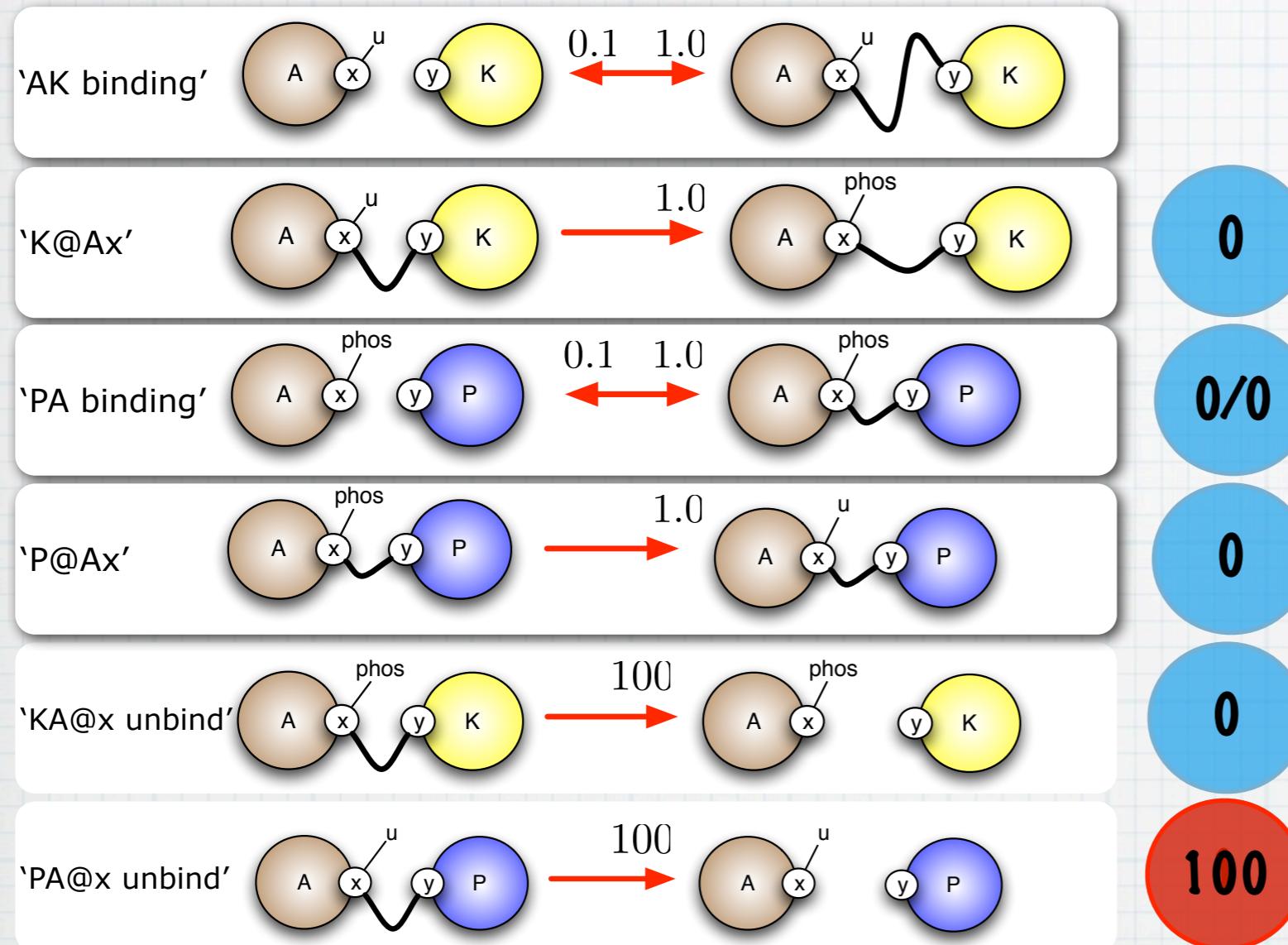


+ 0.009

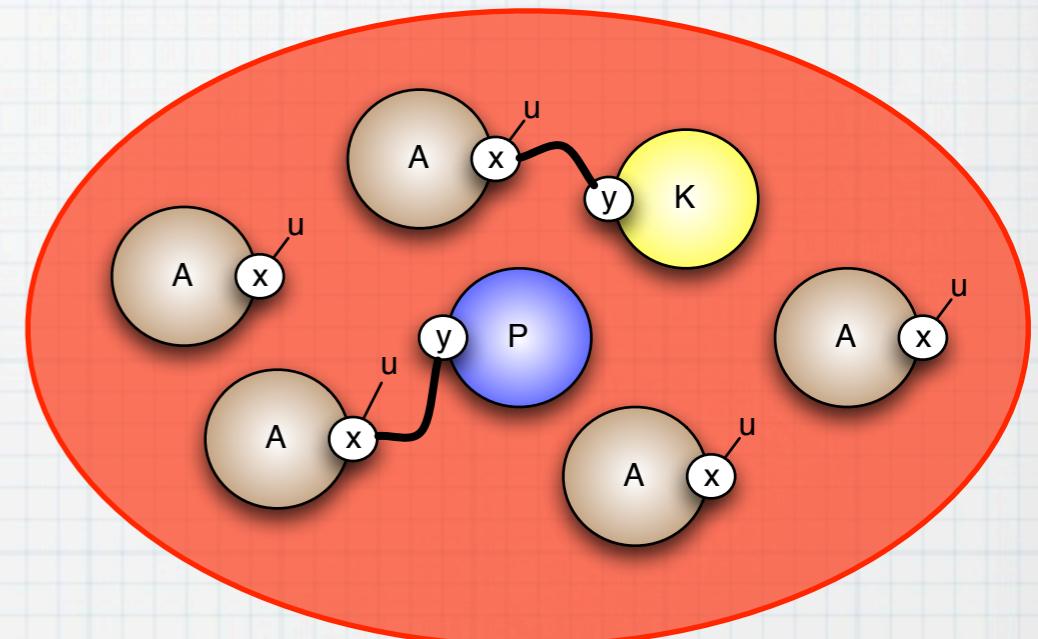


1. Advance time
2. Apply rule instance
3. Update rule and system's activities

System update

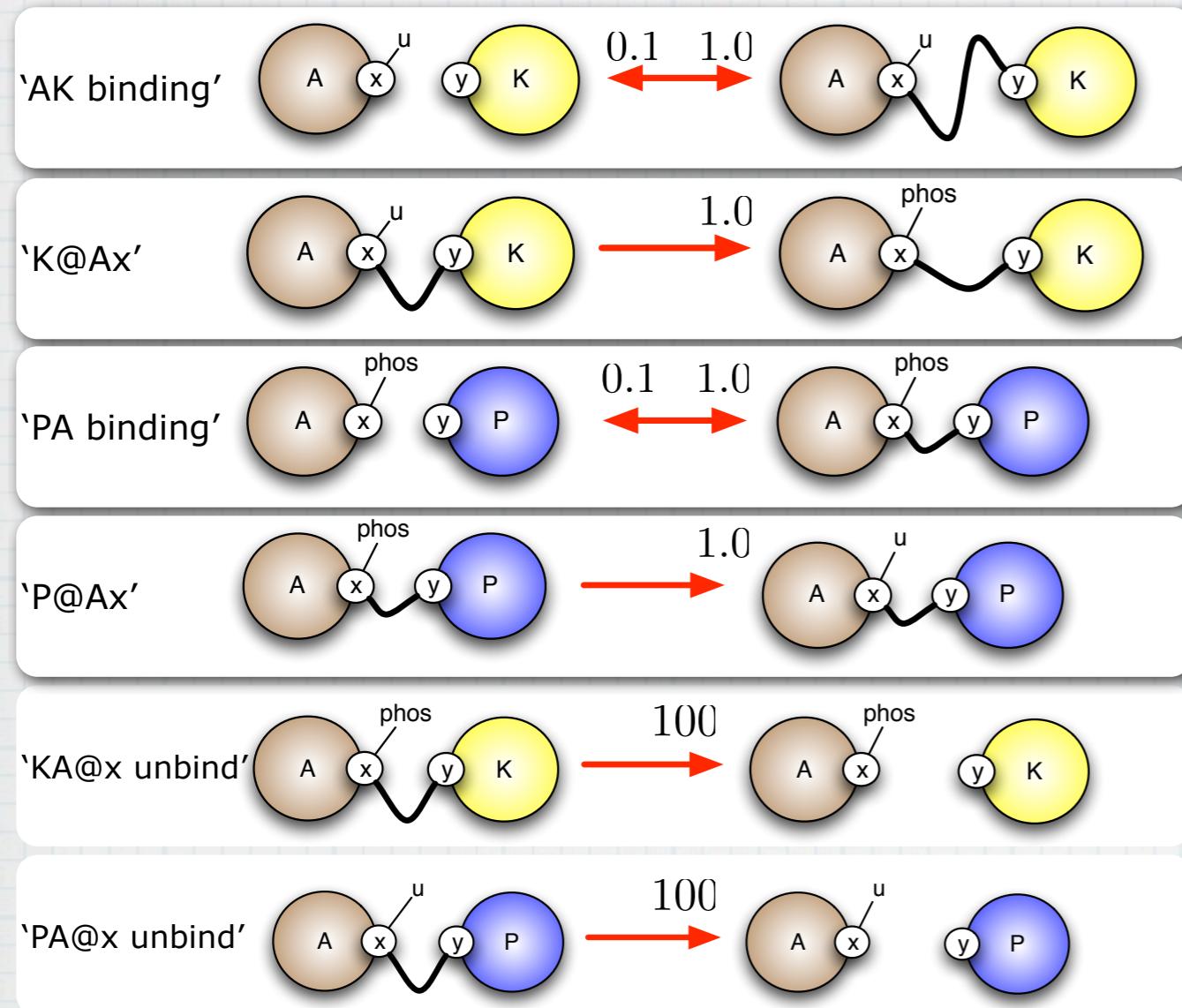


+ 0.009

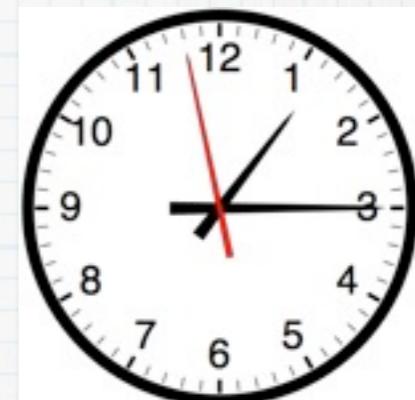


1. Advance time
2. Apply rule instance
3. Update rule and system's activities

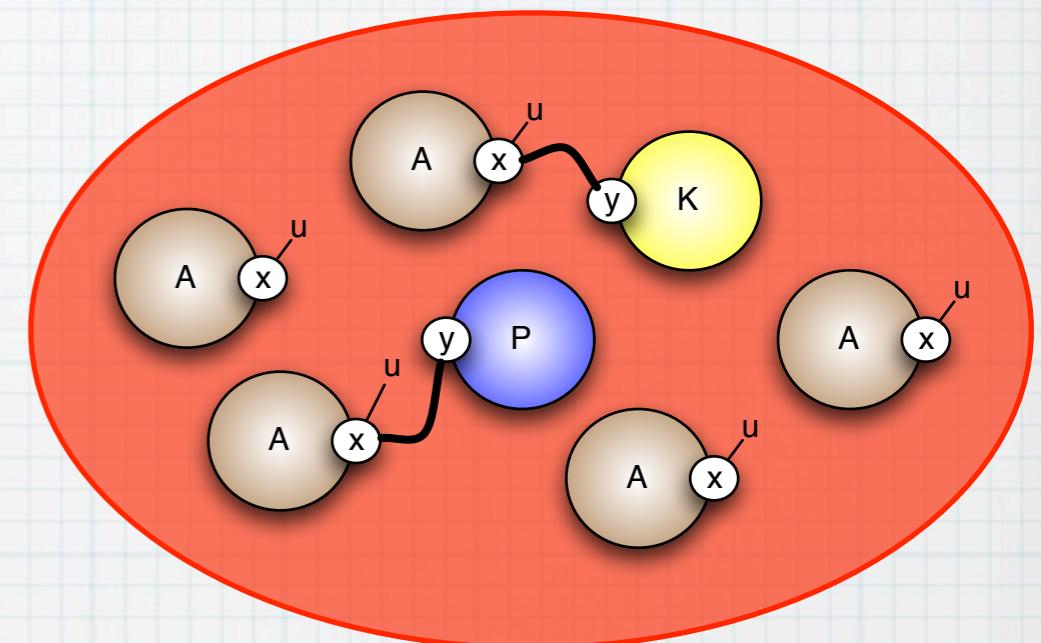
System update



0/0
0
0
100

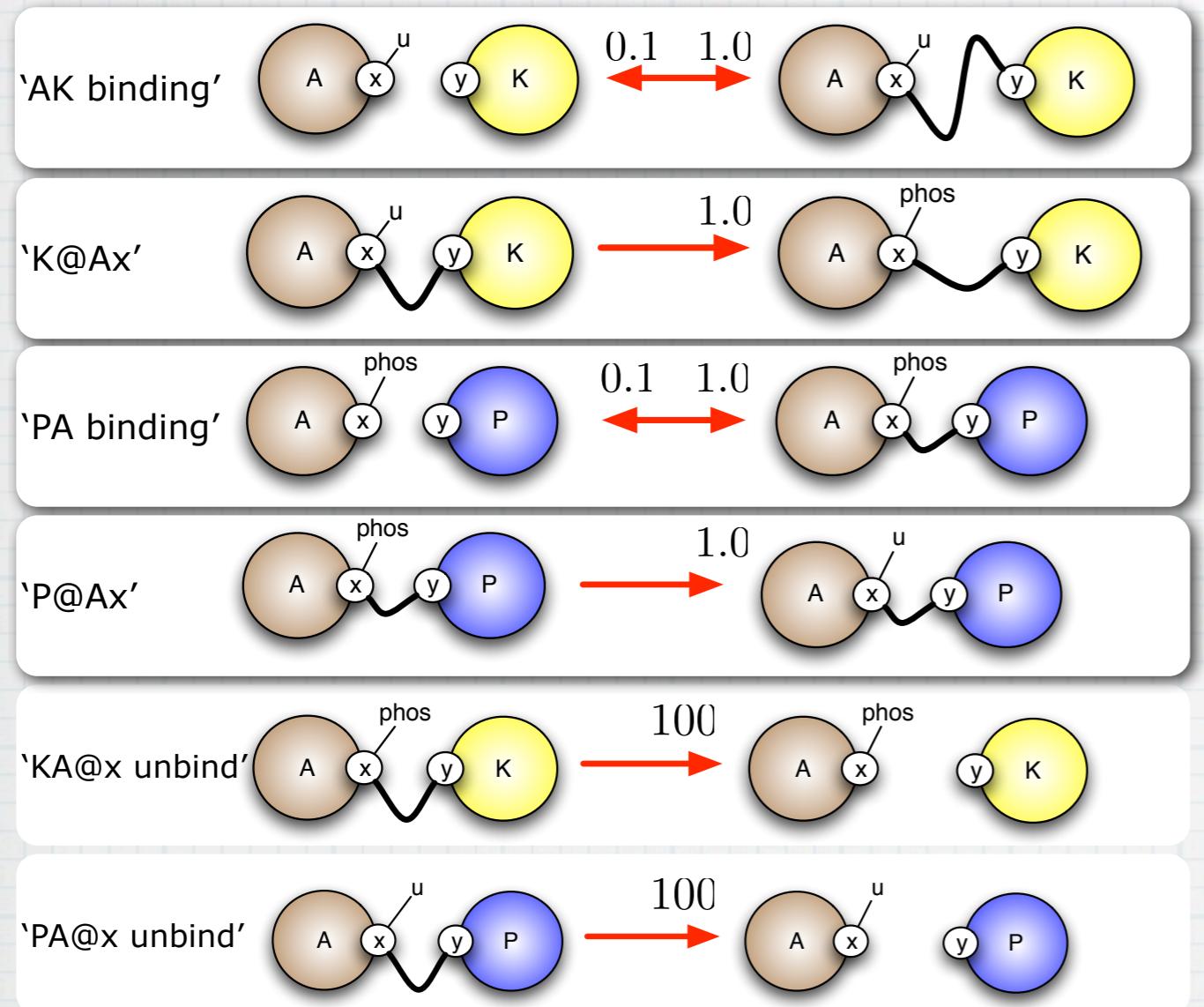


+ 0.009



1. Advance time
2. Apply rule instance
3. Update rule and system's activities

System update



0/0.1

0/0

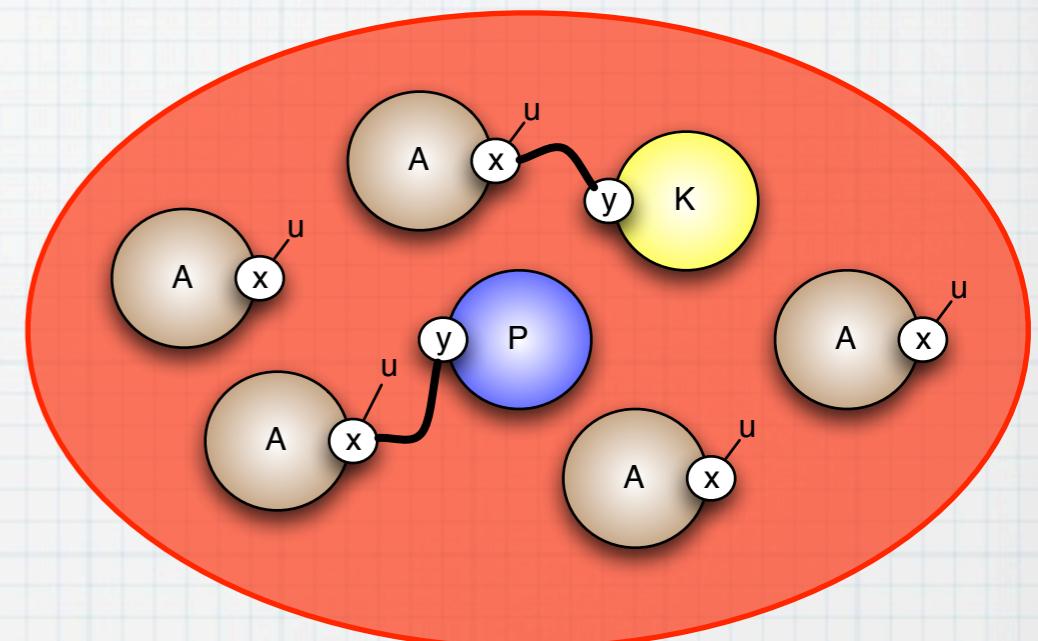
0

0

100

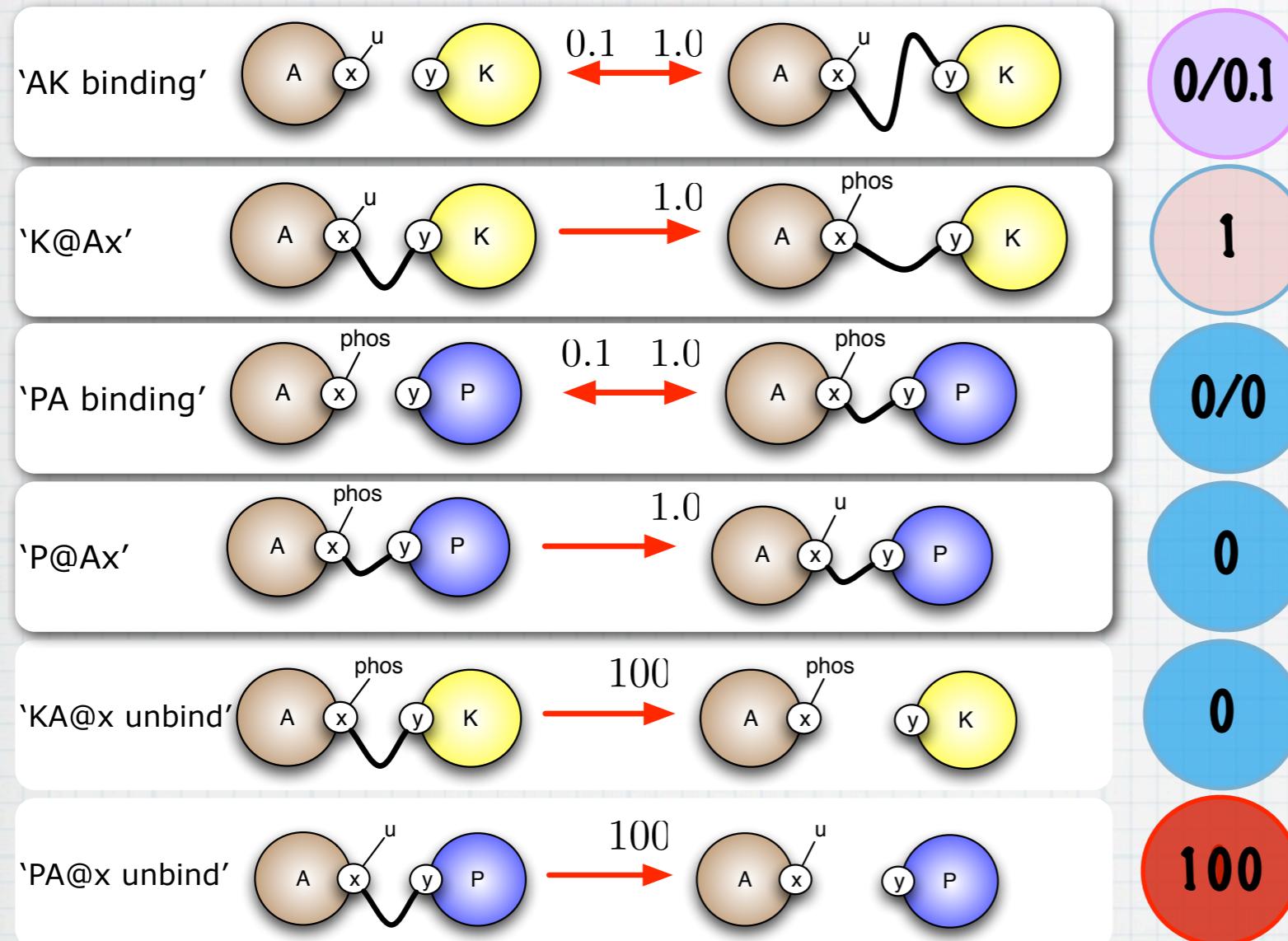


+ 0.009

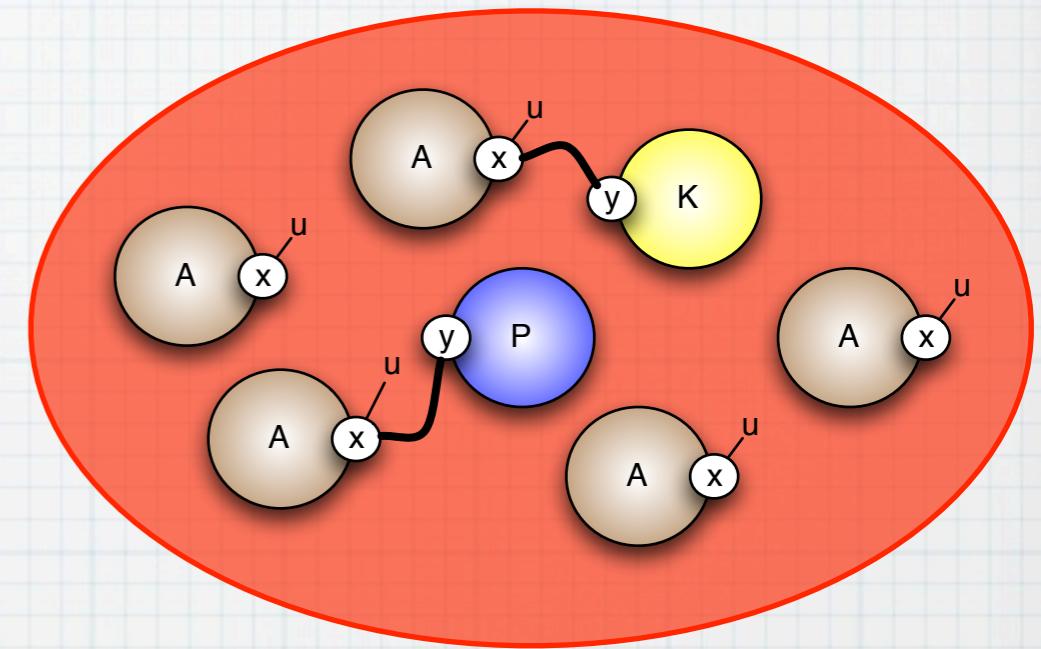


1. Advance time
2. Apply rule instance
3. Update rule and system's activities

System update



+ 0.009

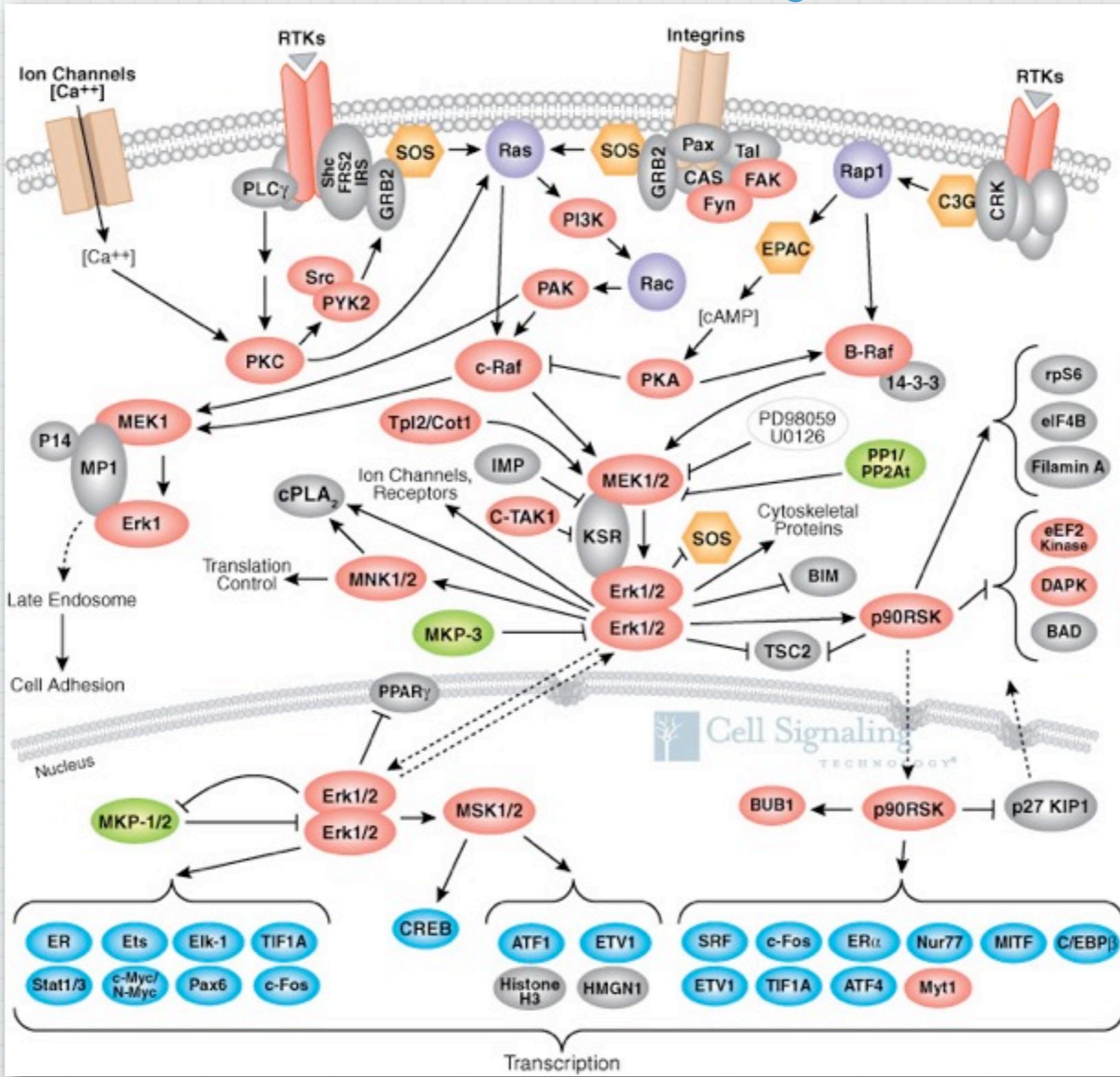


A=101.1

1. Advance time
2. Apply rule instance
3. Update rule and system's activities

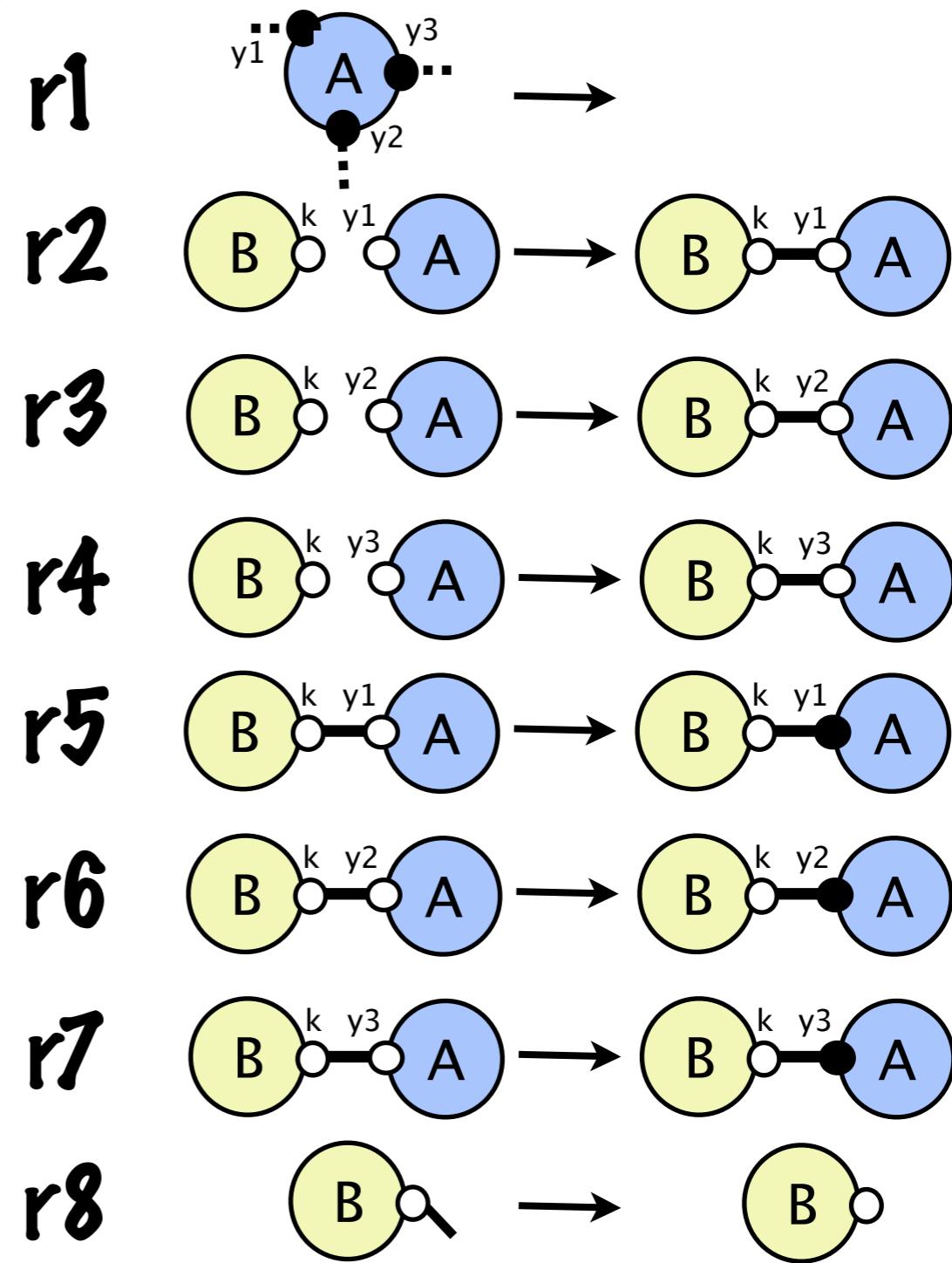
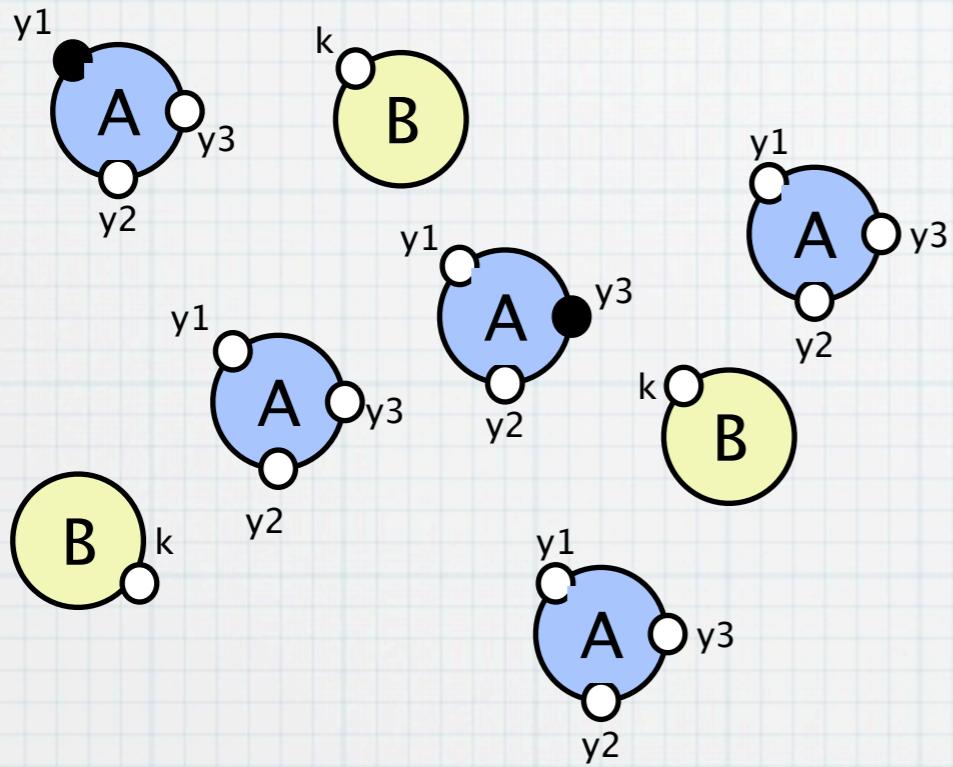
Generating
pathways...

Pathways

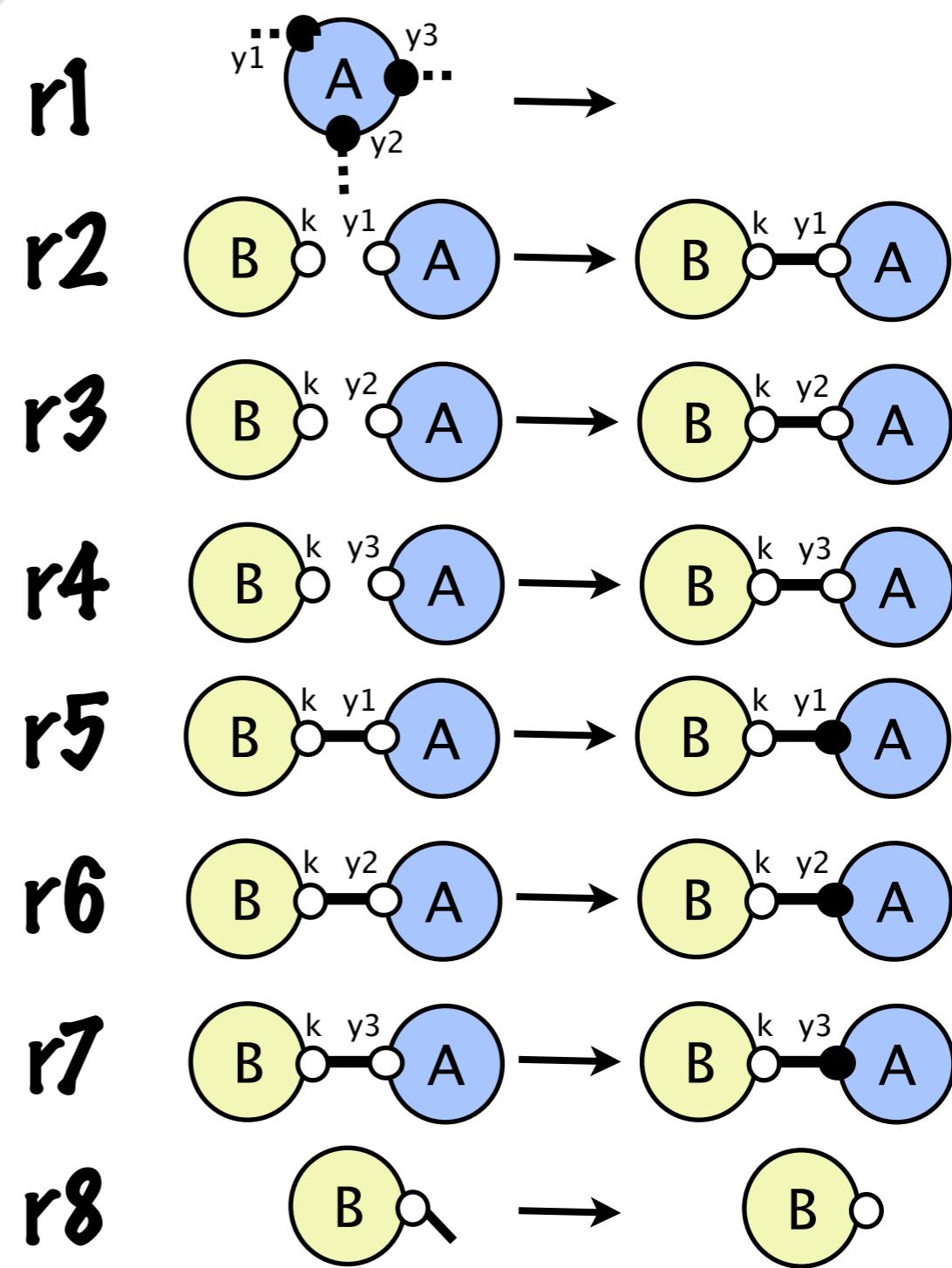
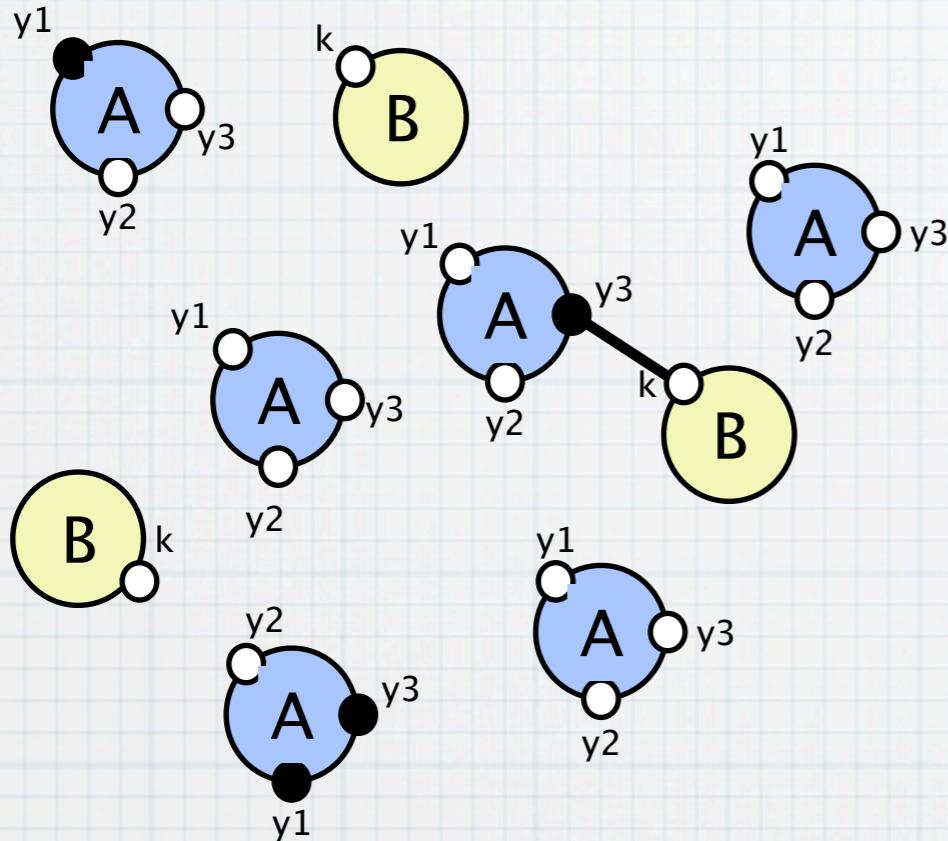


Can we produce such a map from a kappa model?

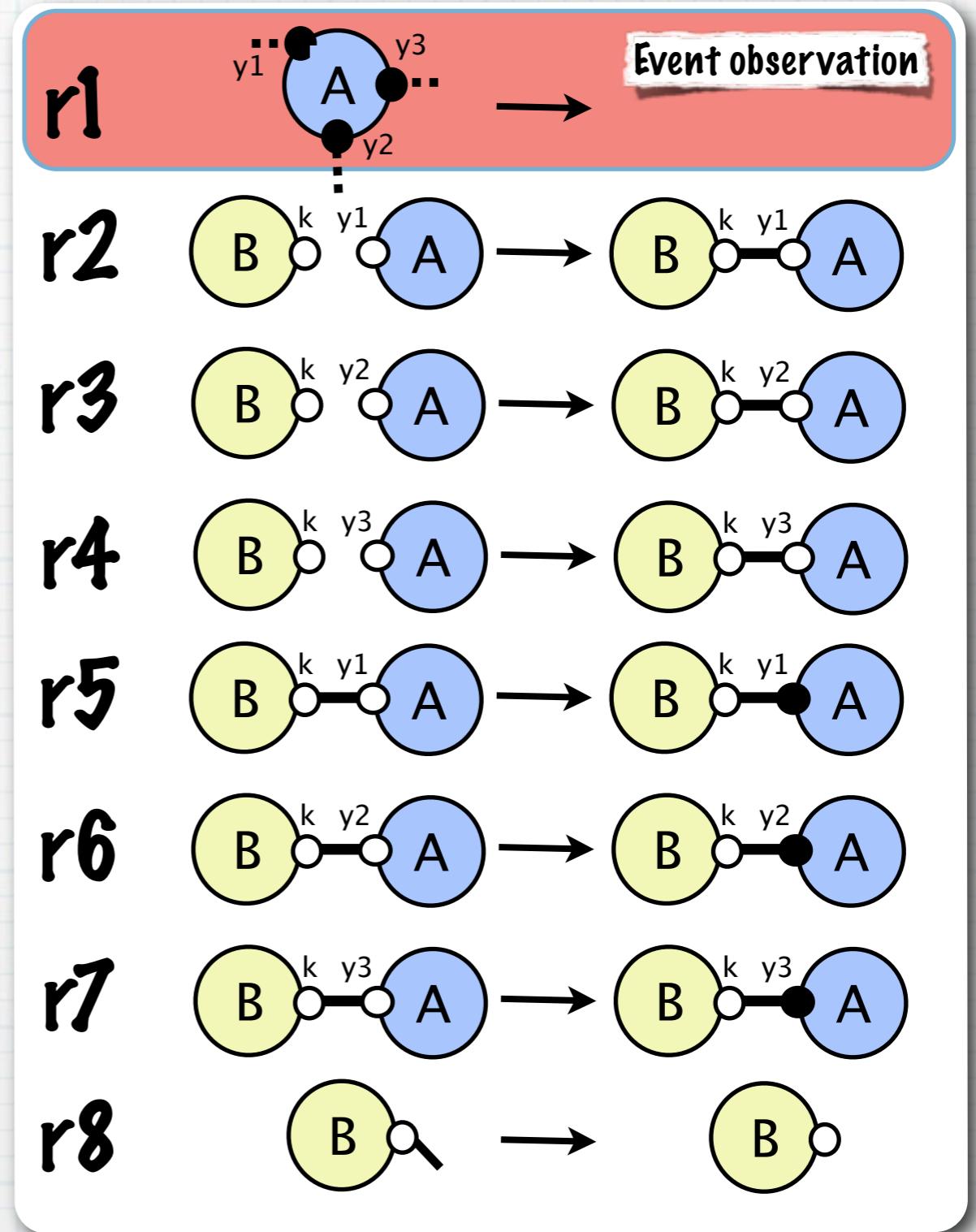
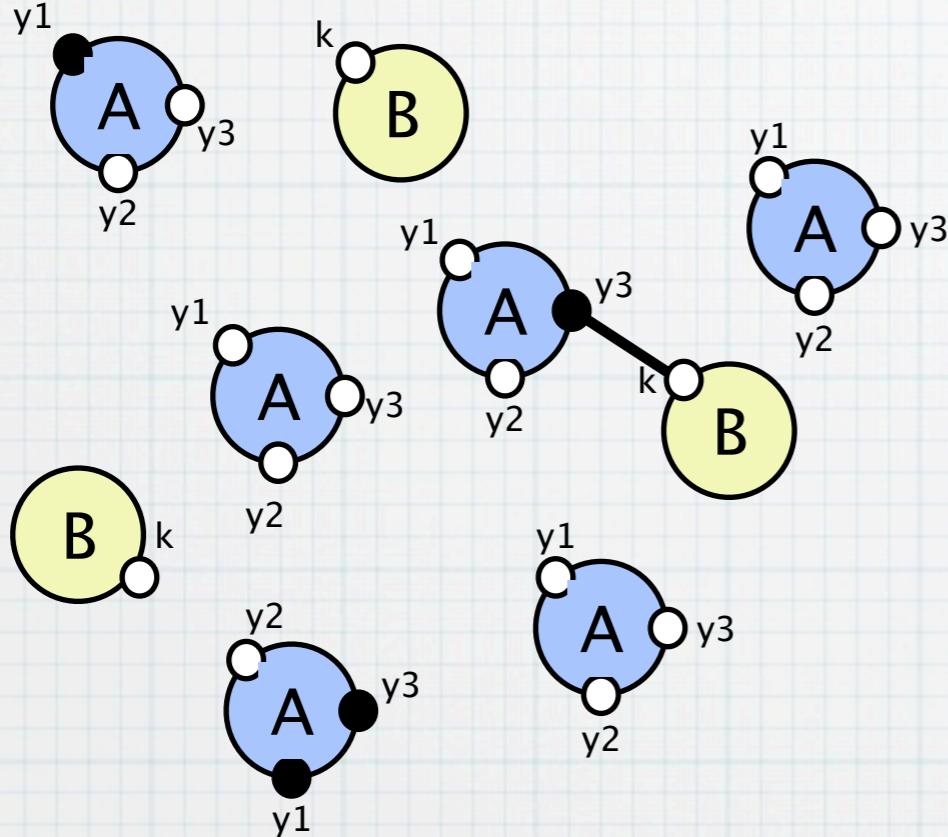
Interleaving...



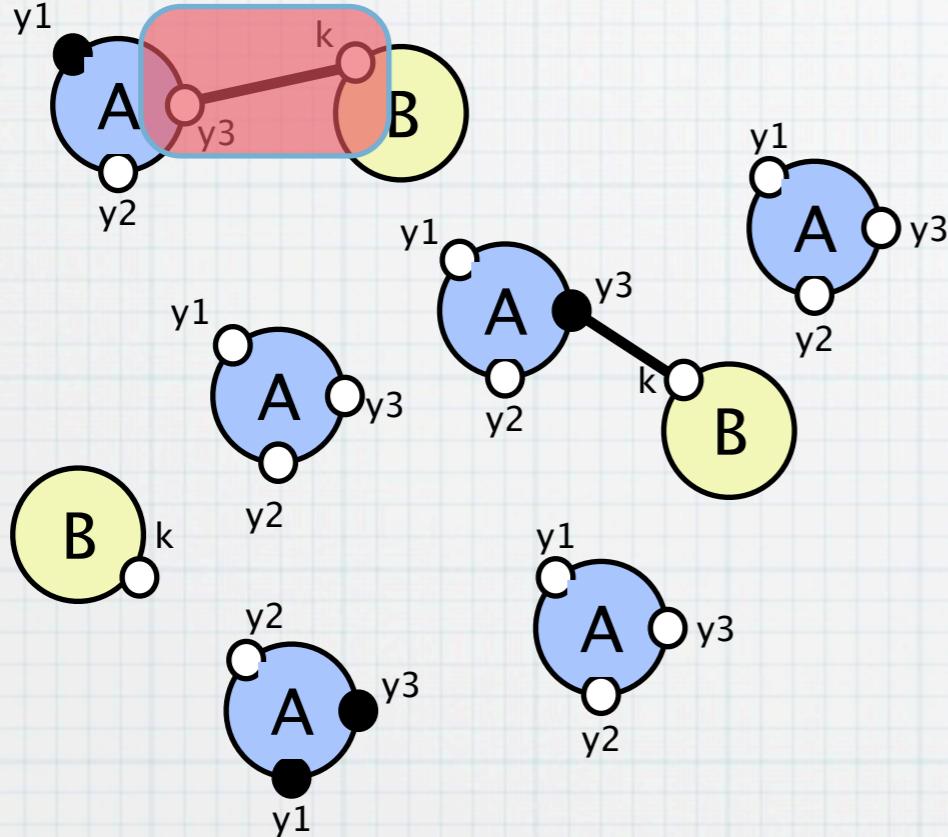
Interleaving...



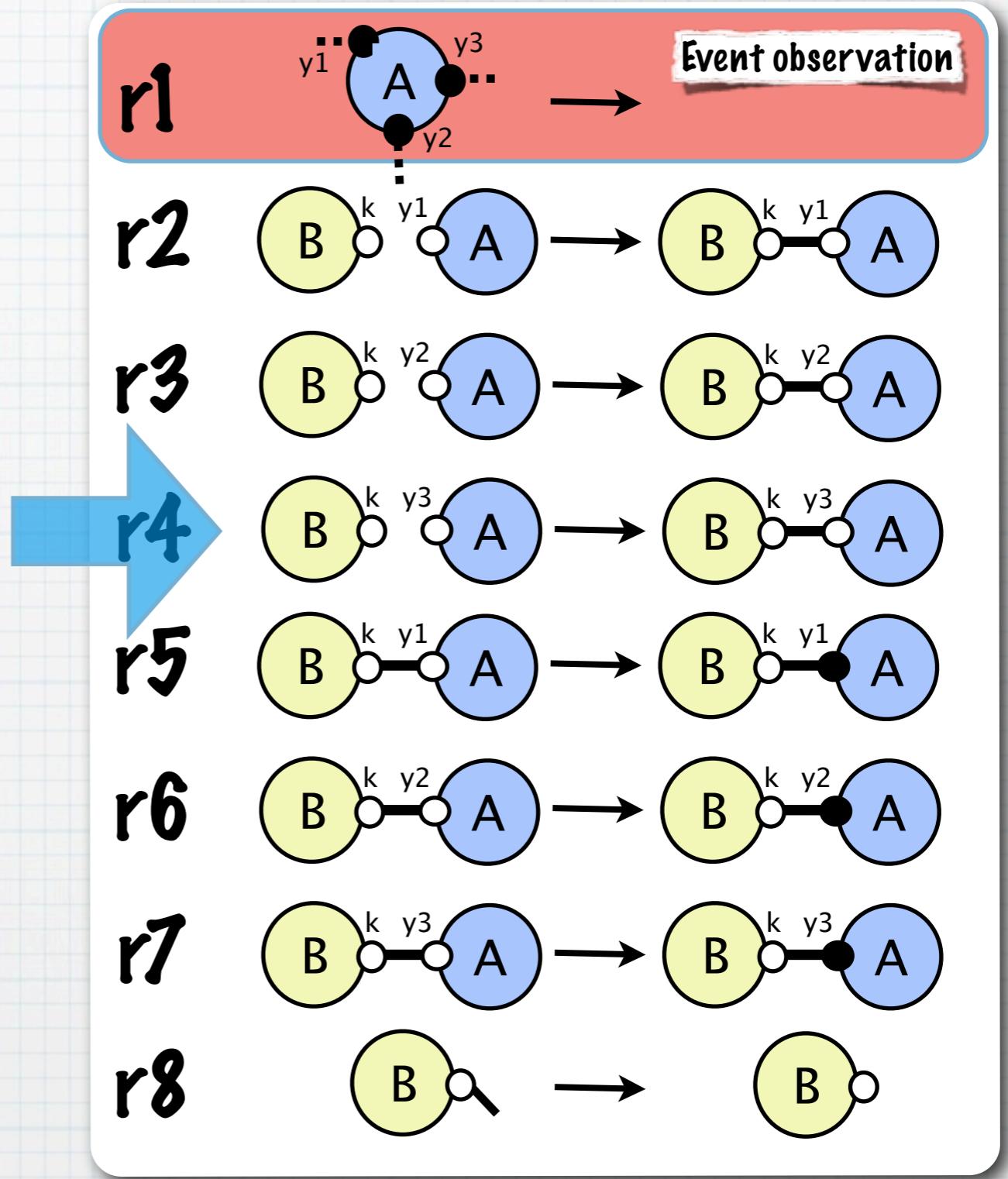
Interleaving...



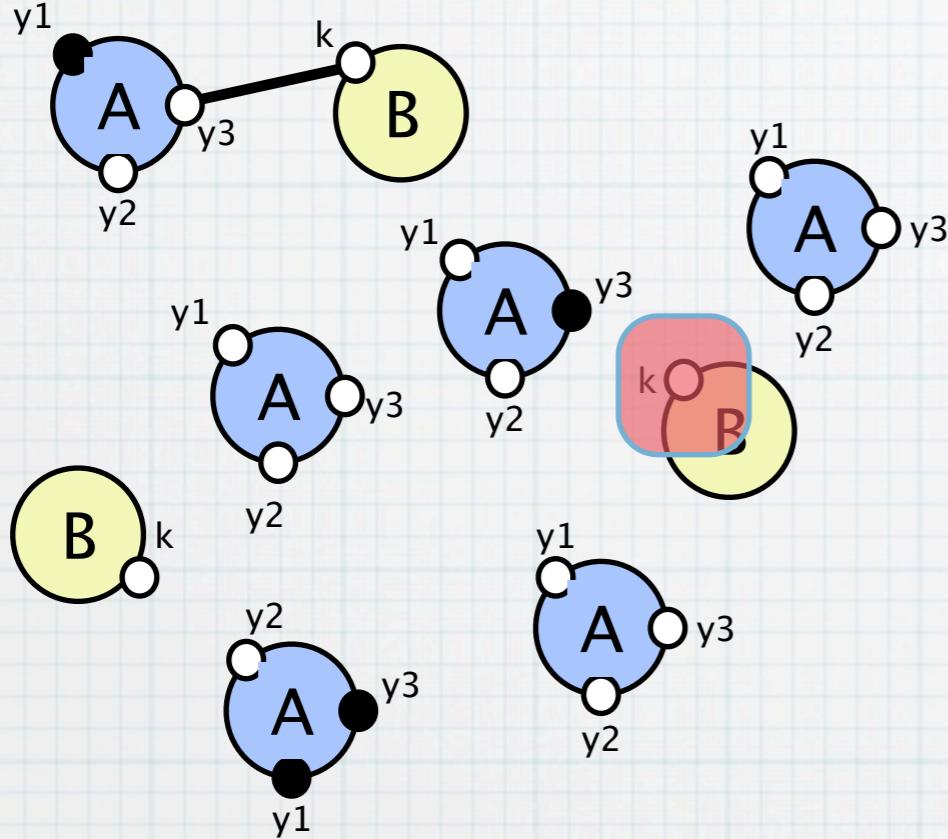
Interleaving...



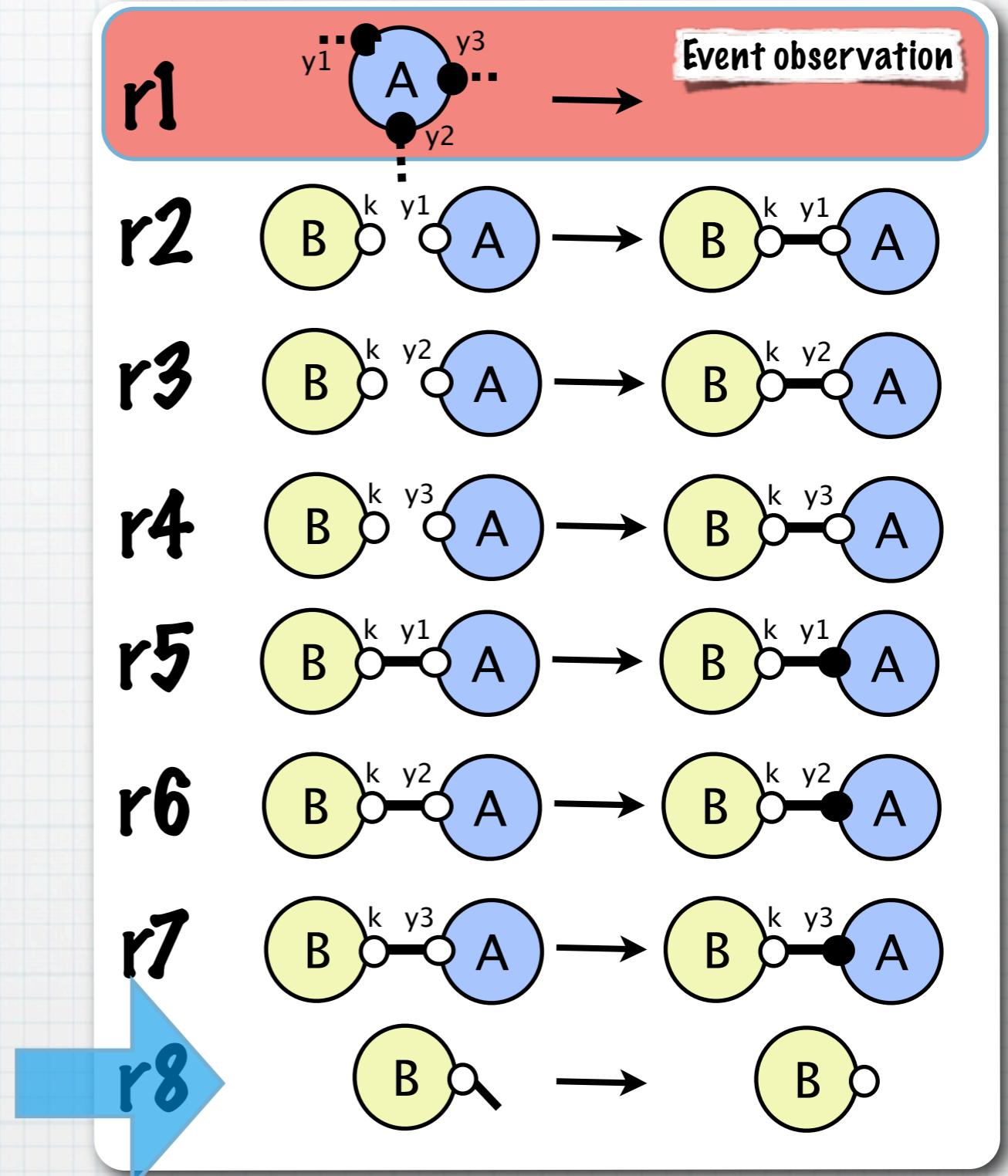
r_4



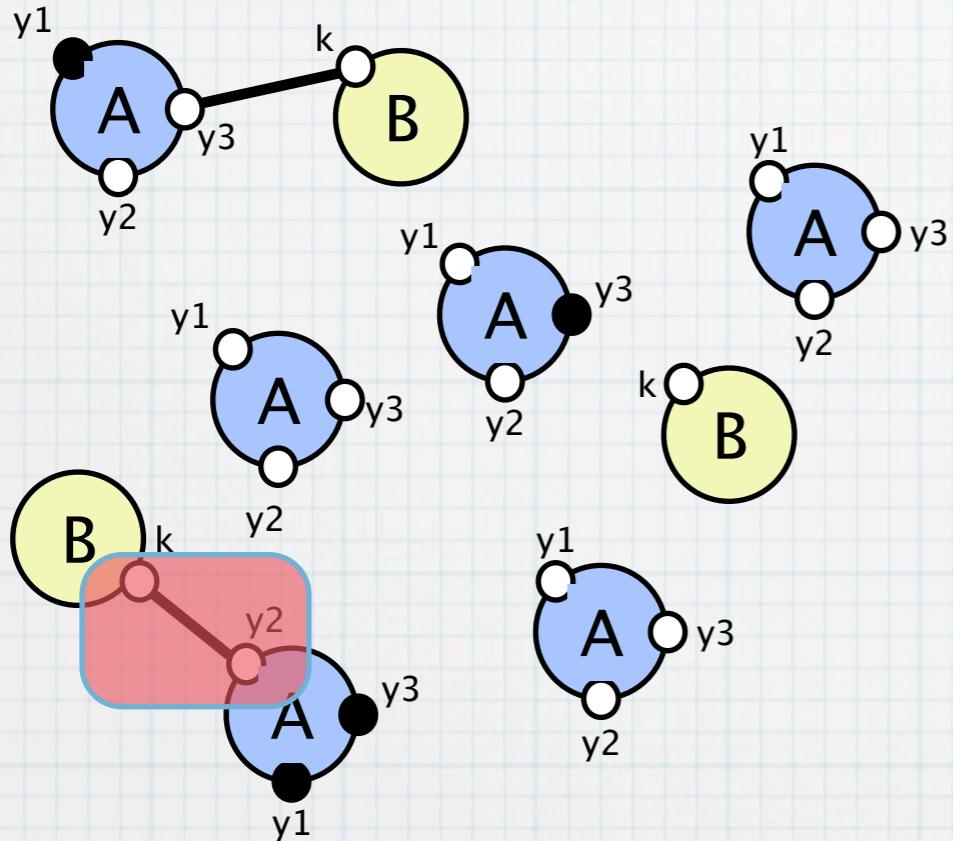
Interleaving...



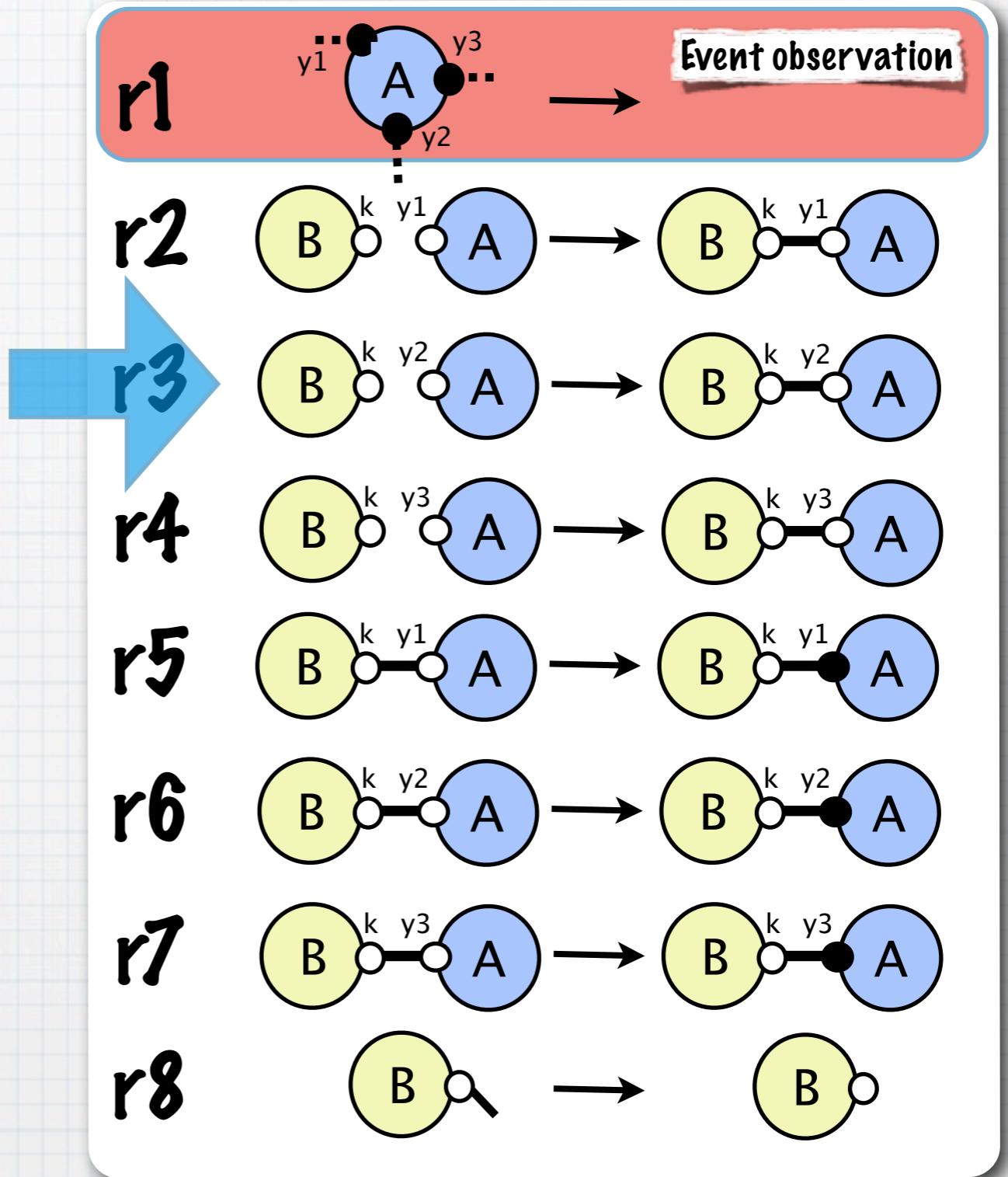
$r_4; r_8$



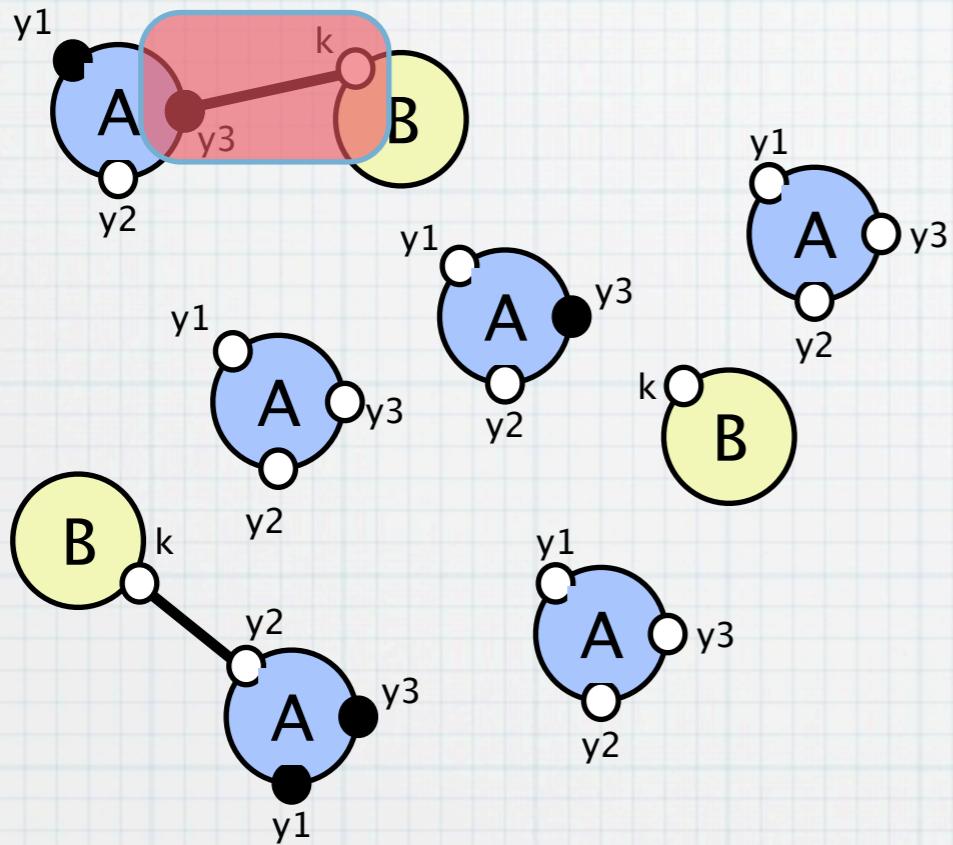
Interleaving...



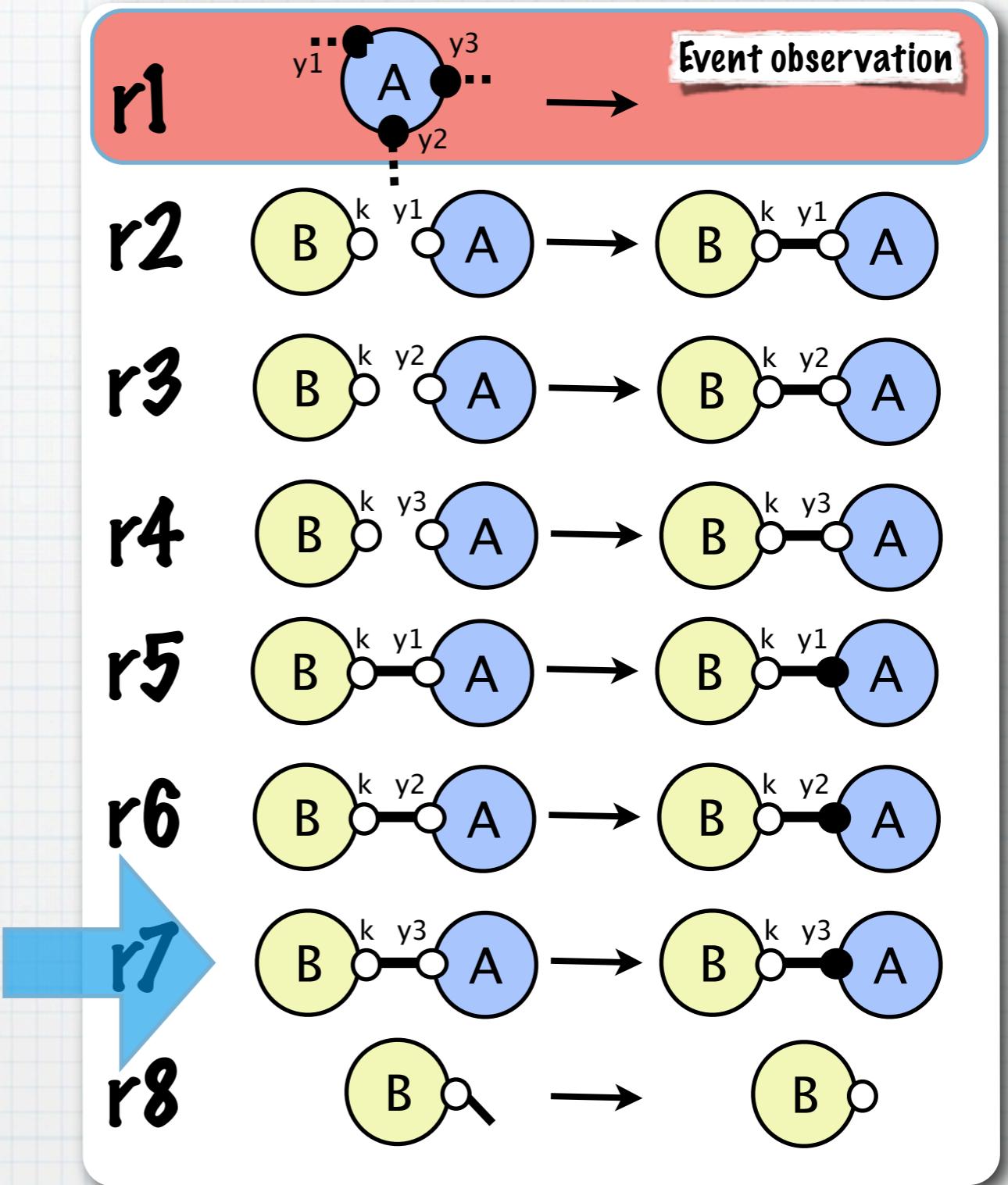
$r_4; r_8; r_3$



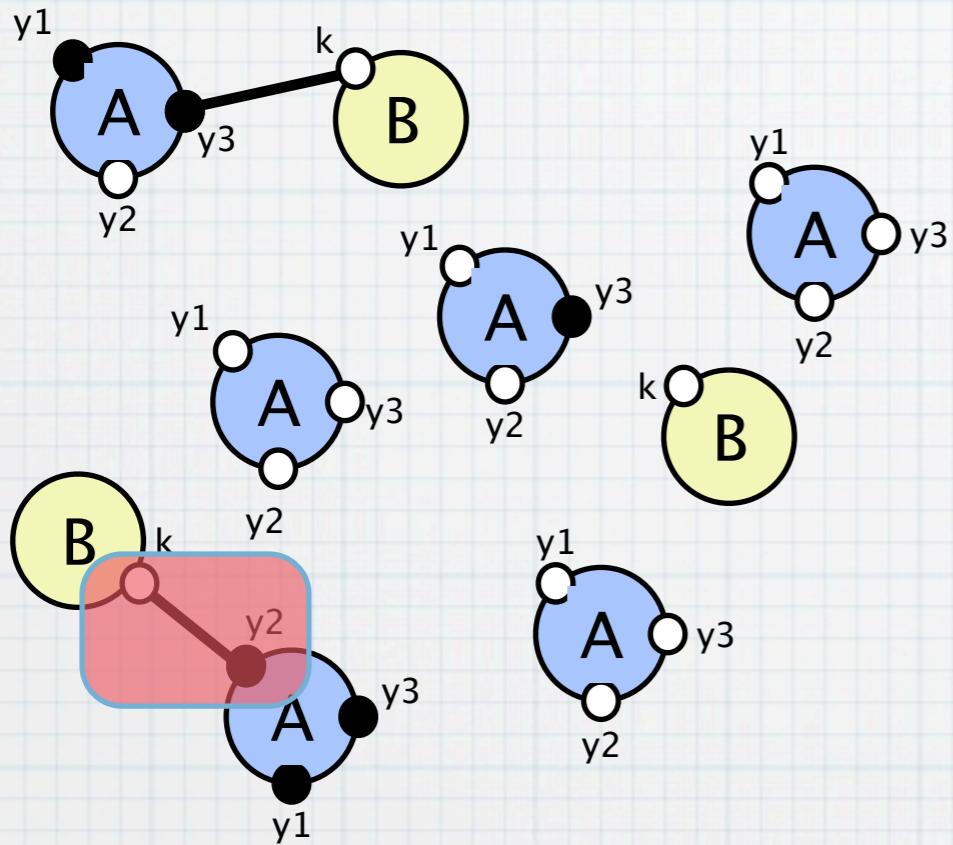
Interleaving...



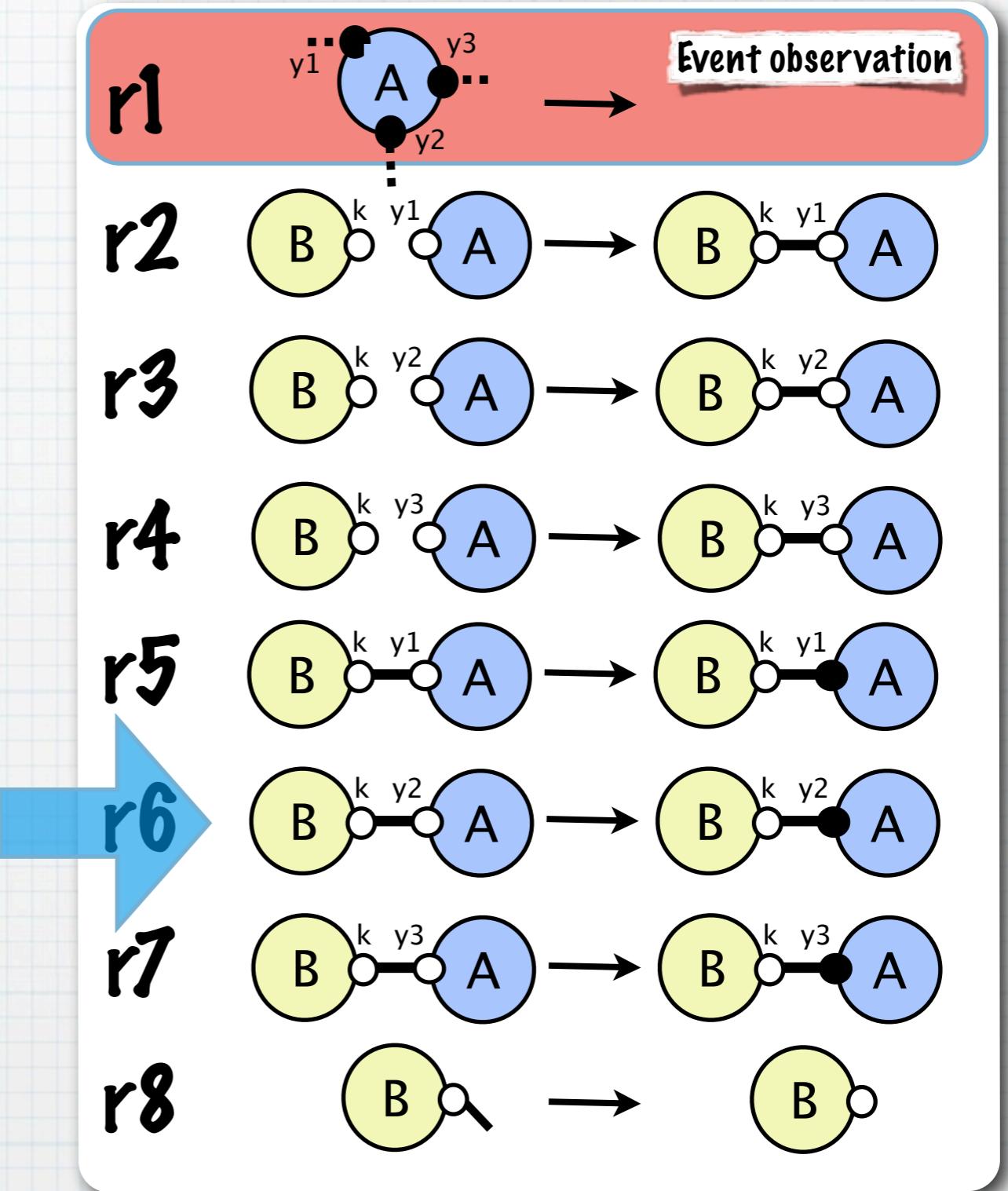
$r_4; r_8; r_3; r_7$



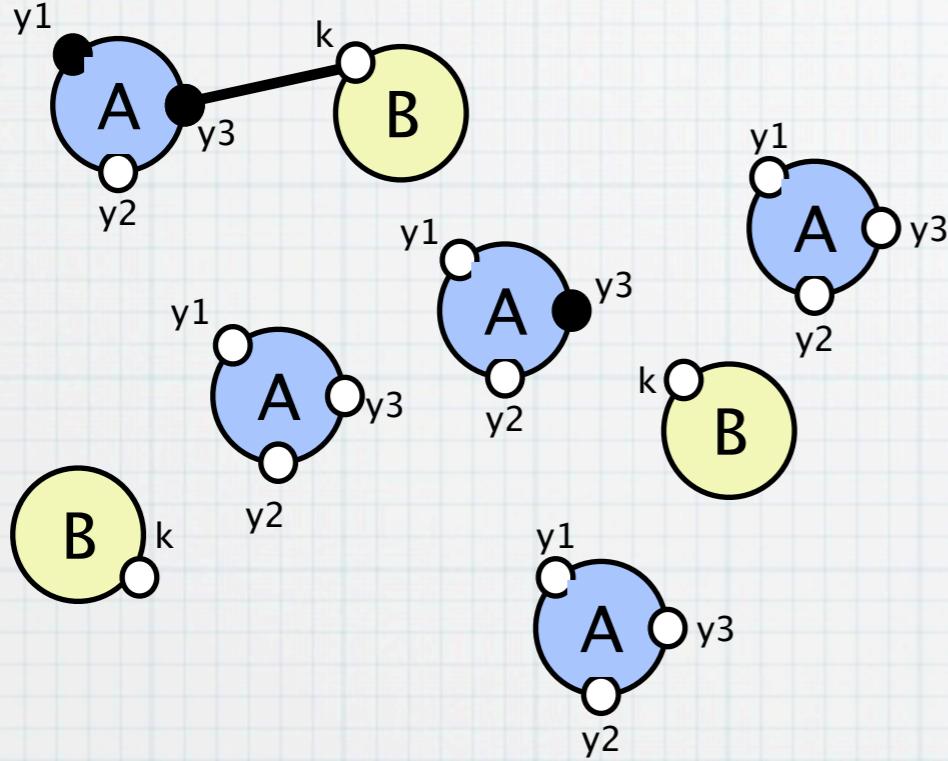
Interleaving...



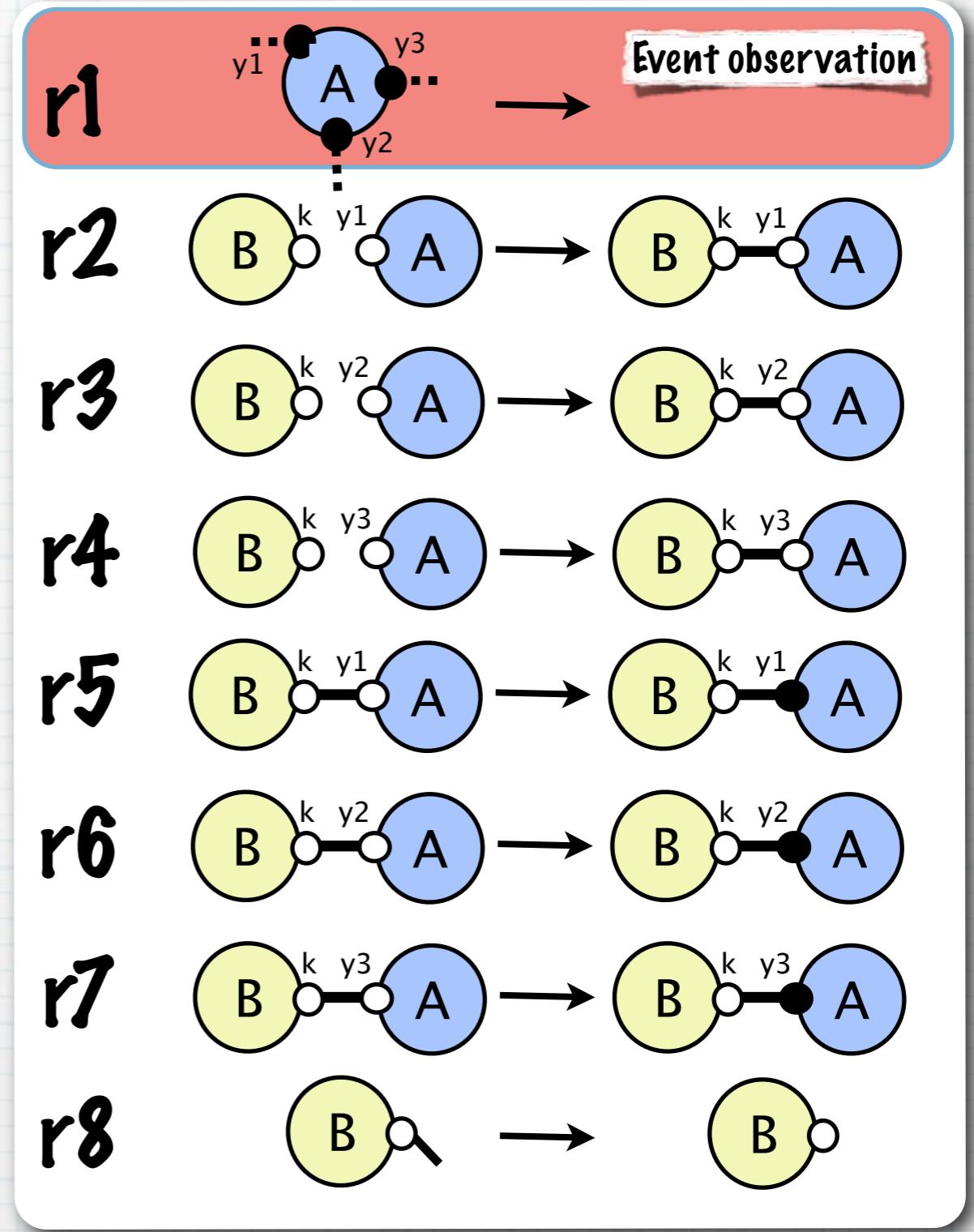
$r_4; r_8; r_3; r_7; r_6$



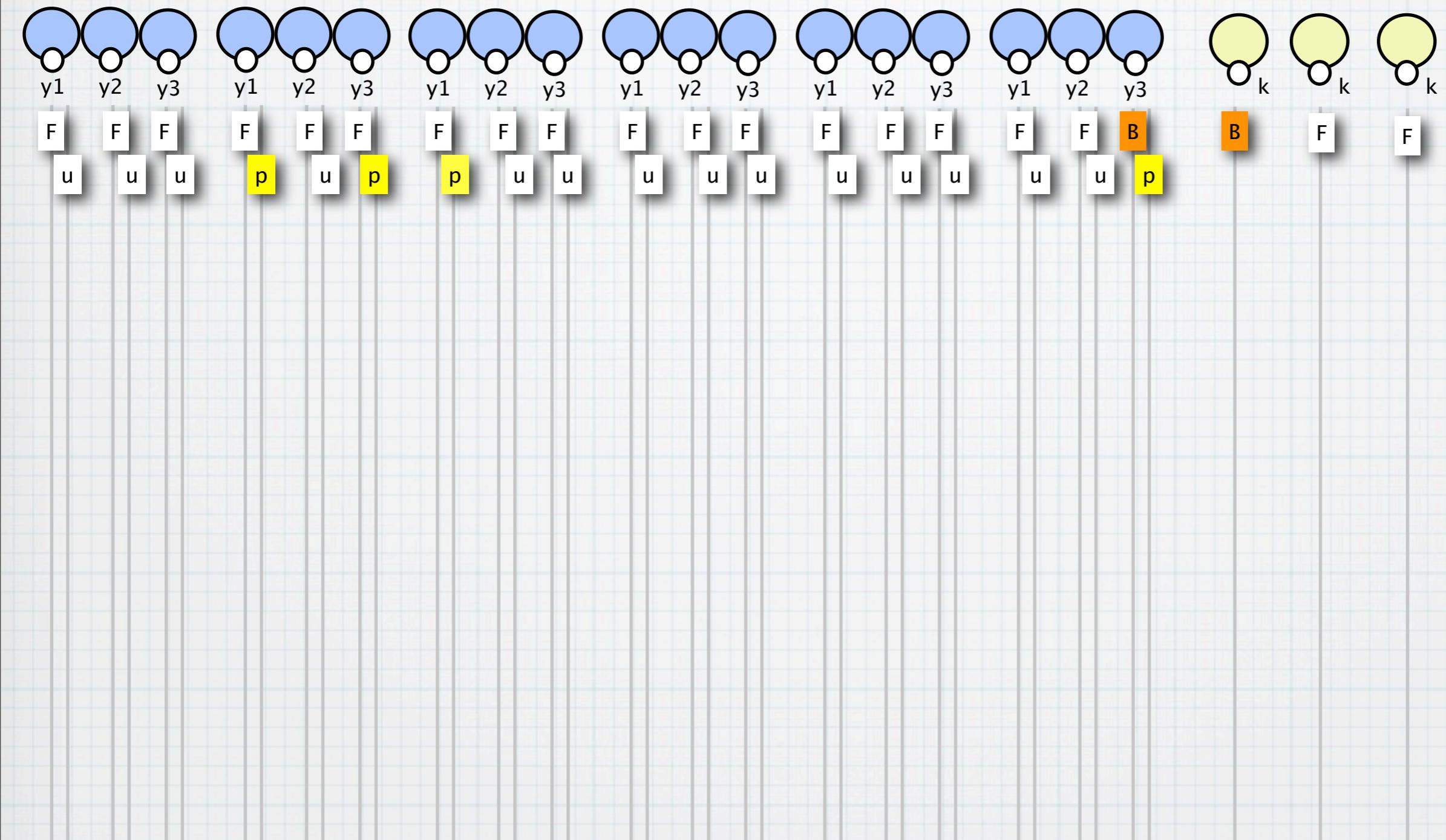
Interleaving...



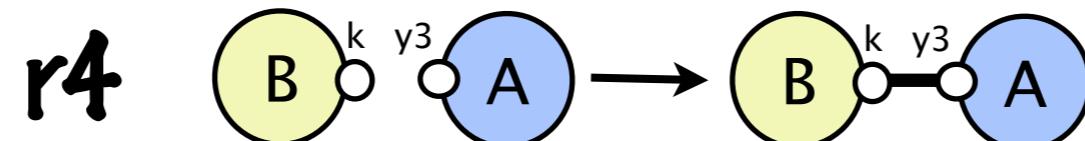
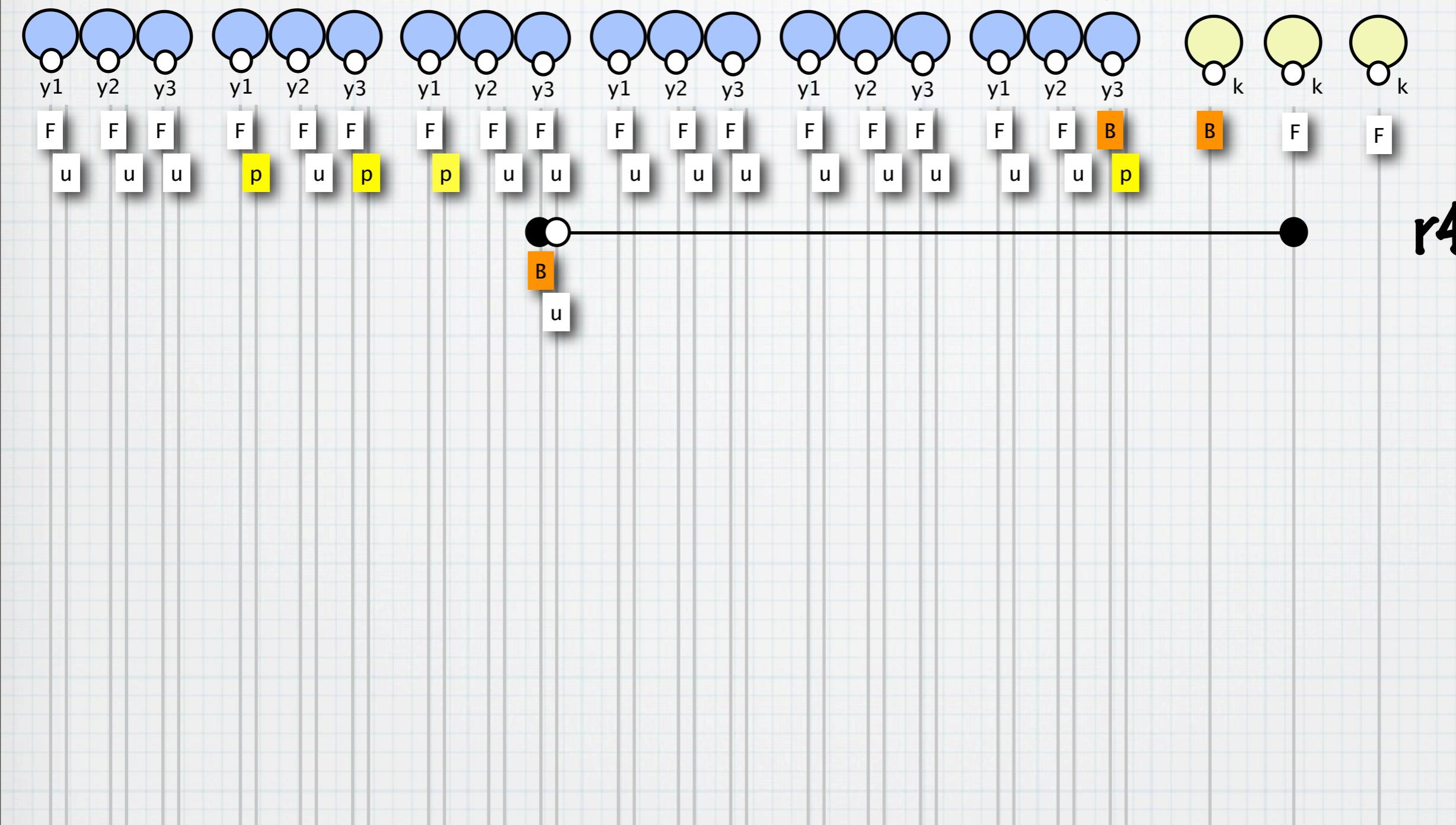
r4;r8;r3;r7;r6;r1



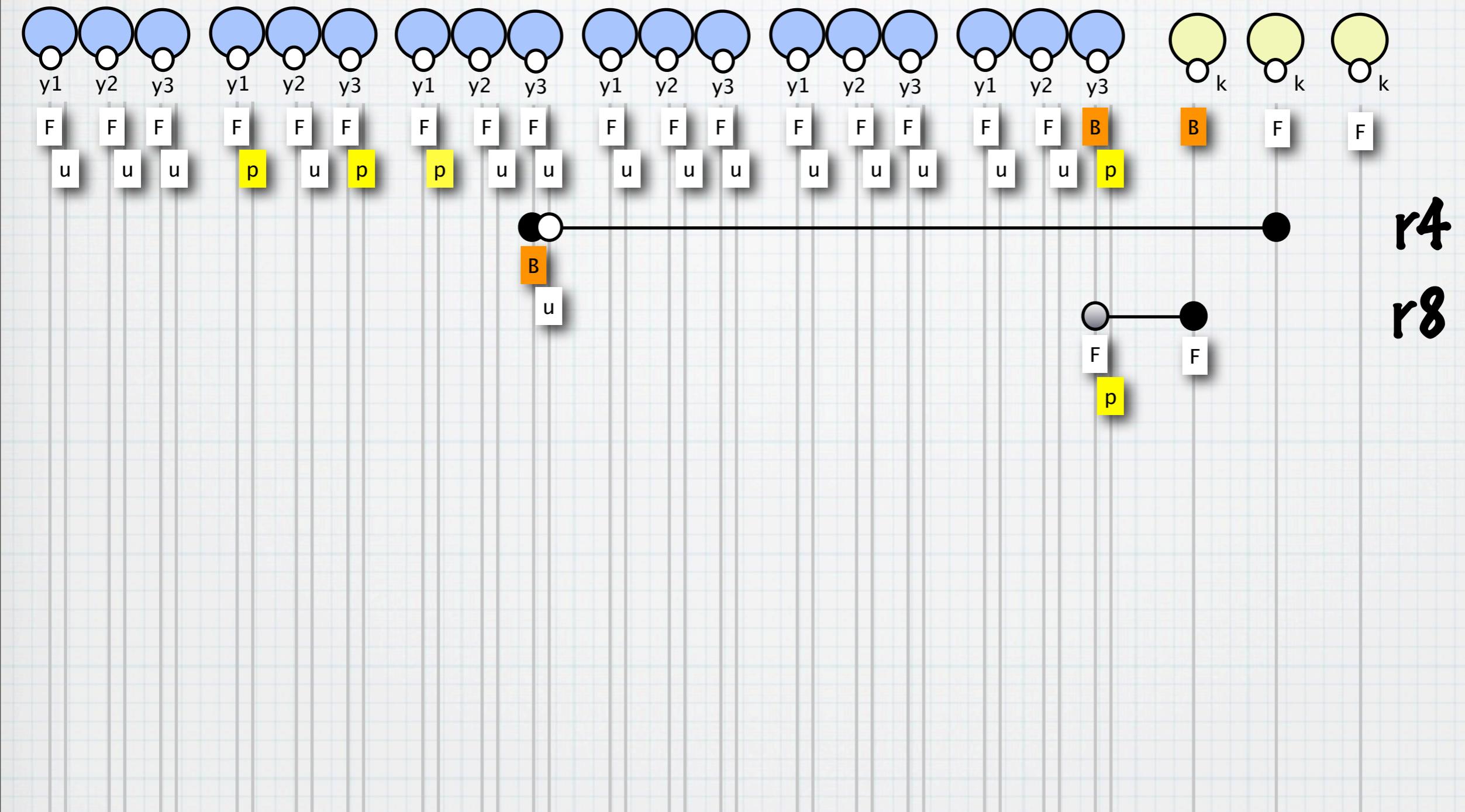
Concurrency



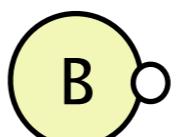
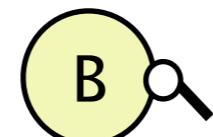
Concurrency



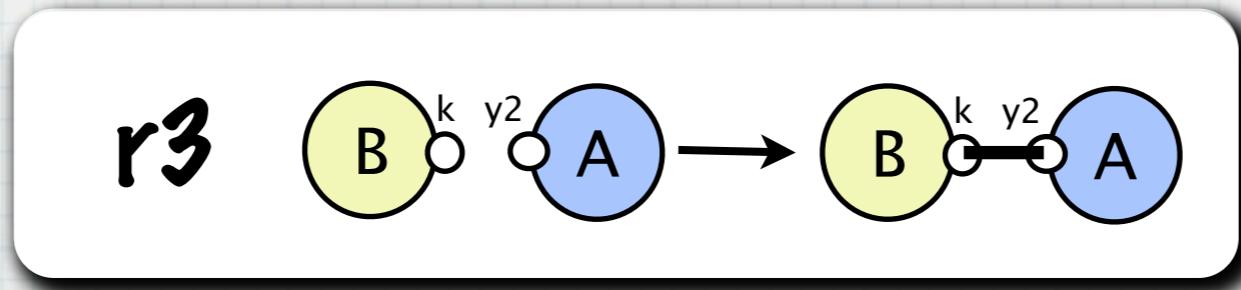
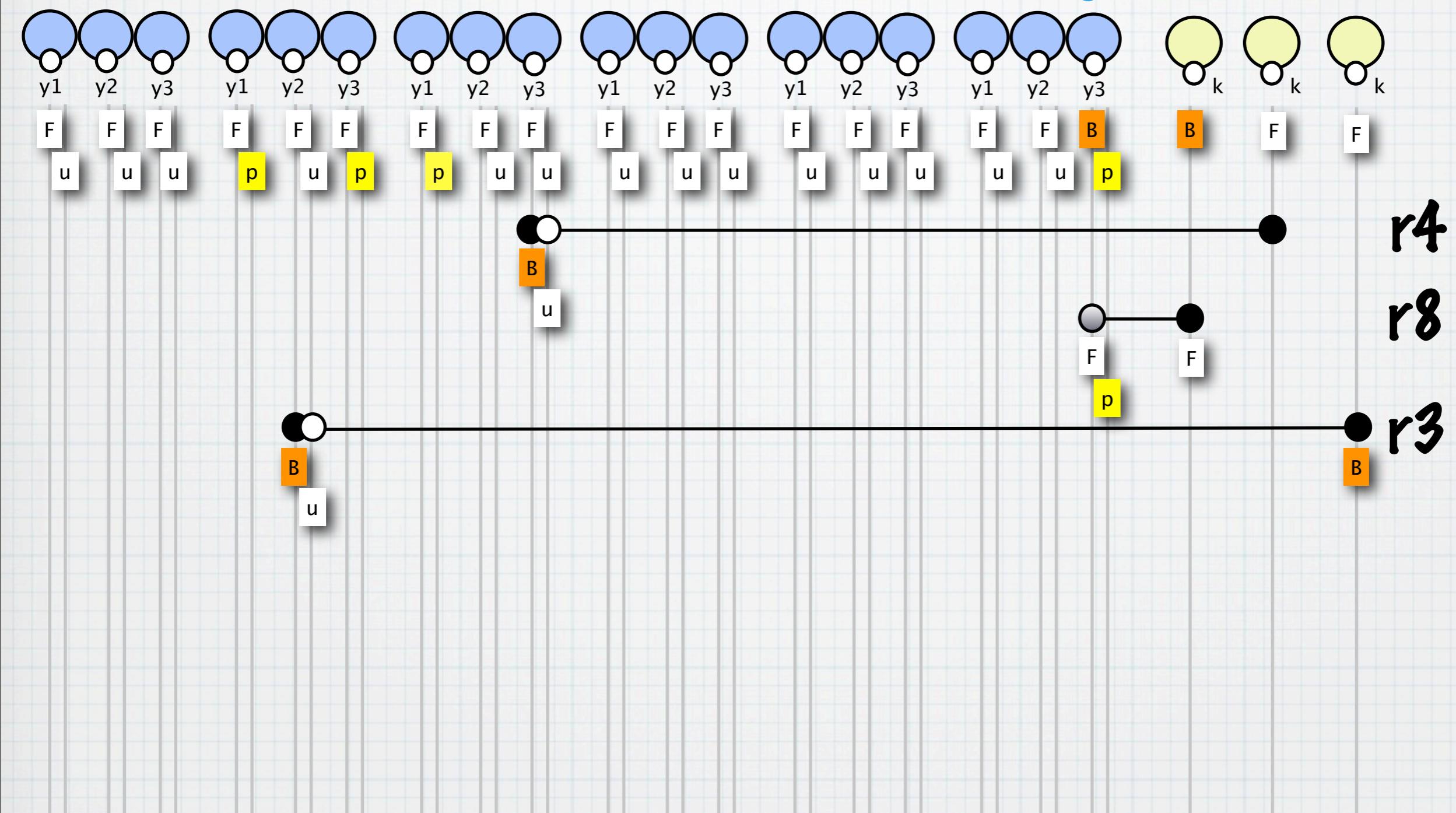
Concurrency



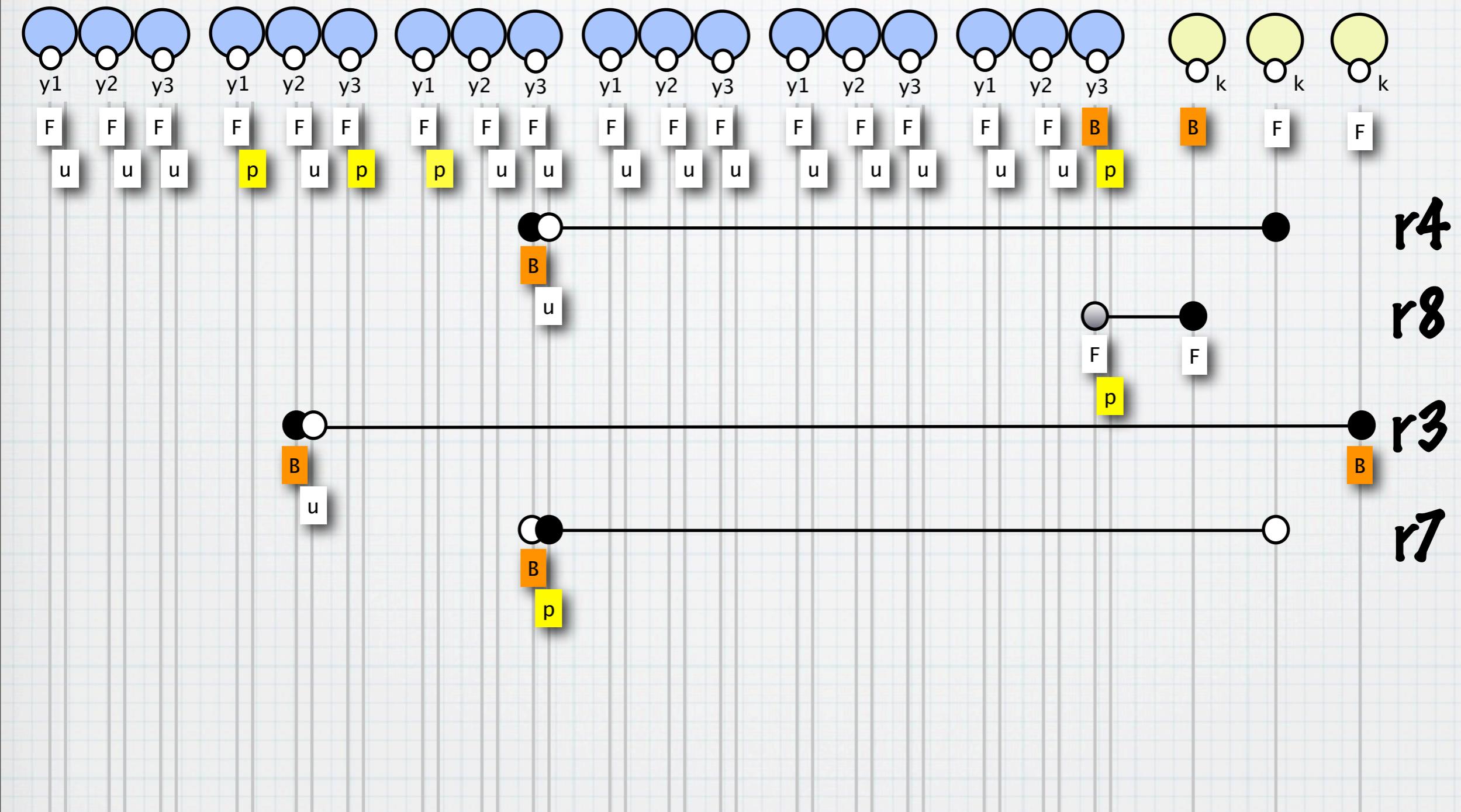
r8



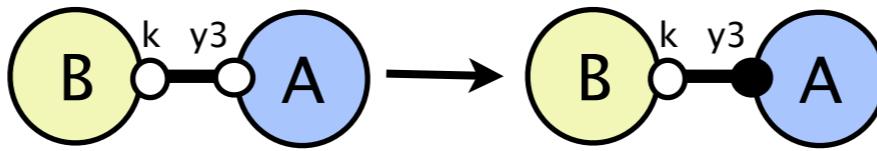
Concurrency



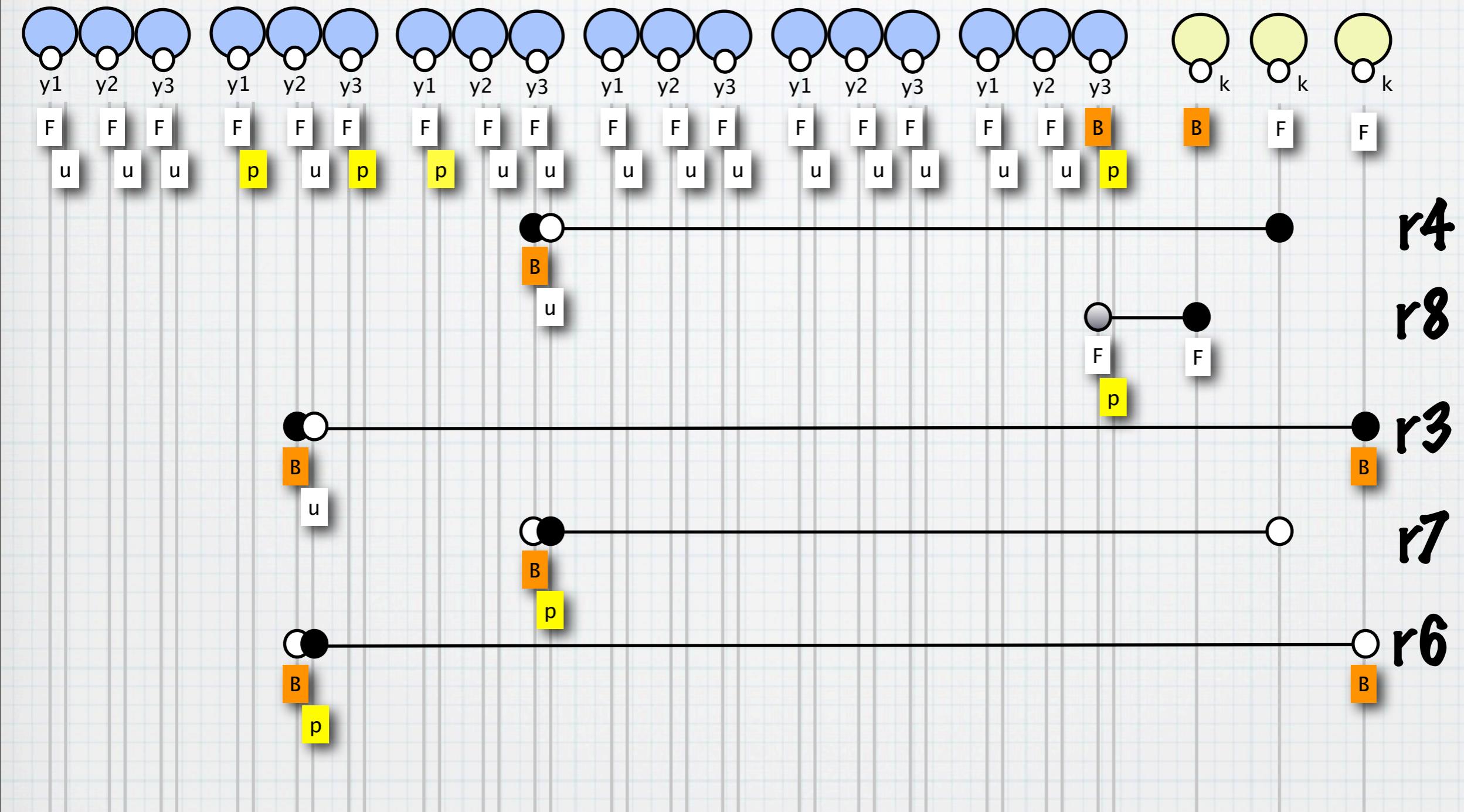
Concurrency



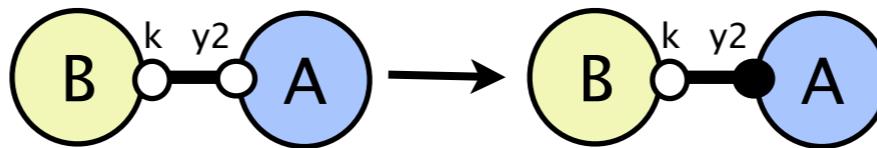
r7



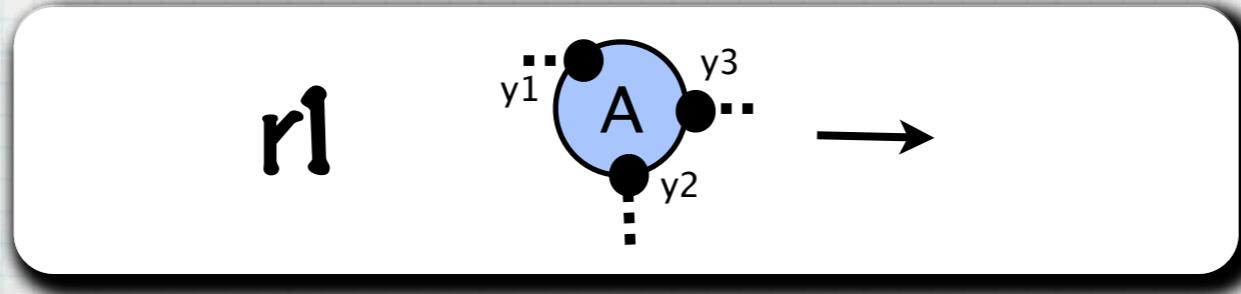
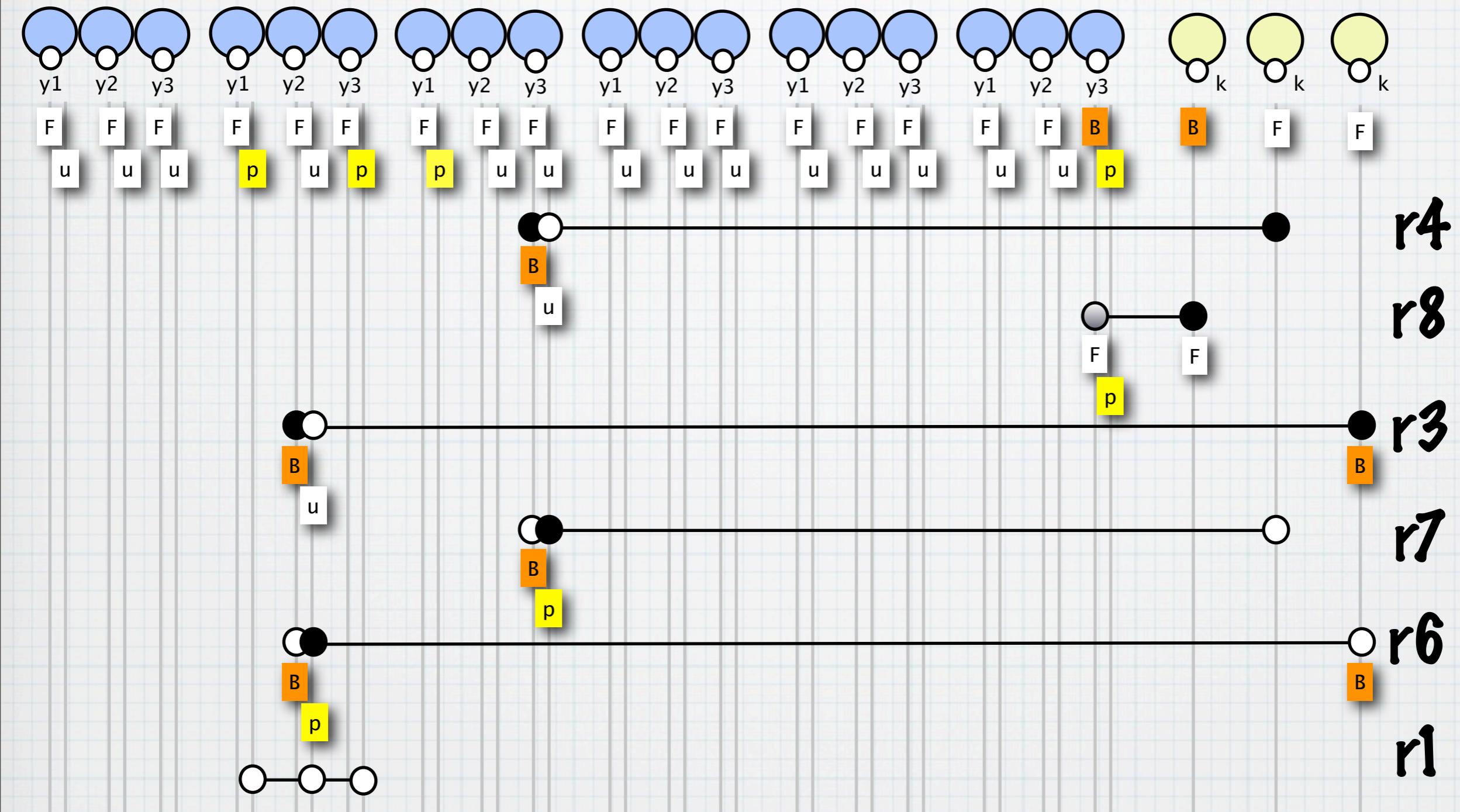
Concurrency



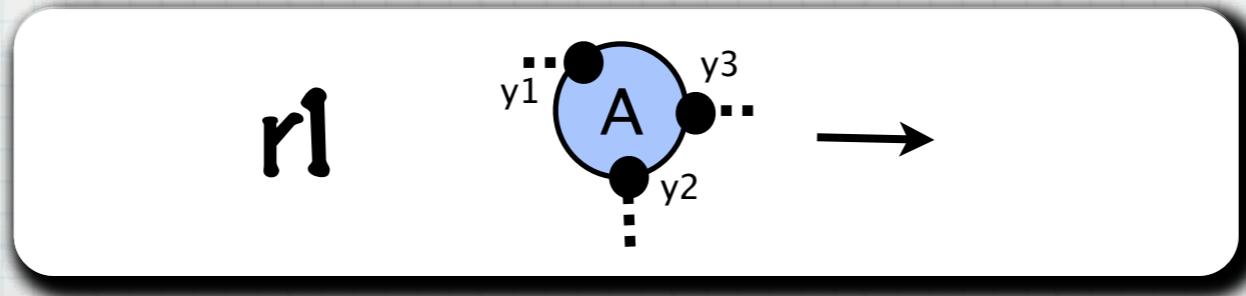
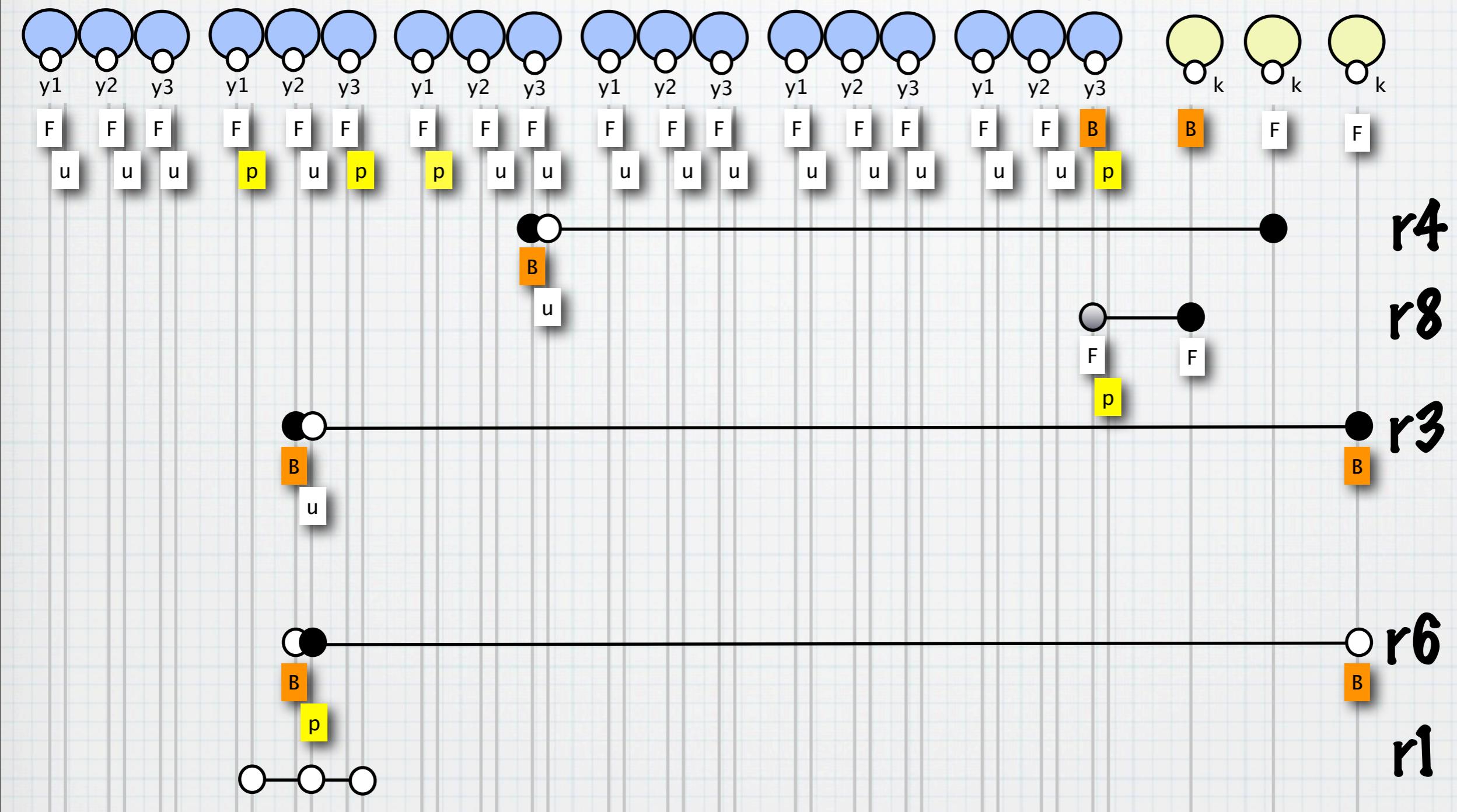
r6



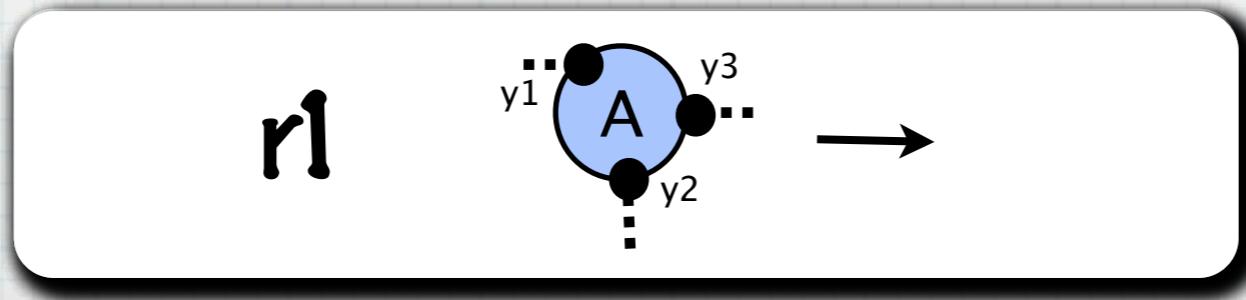
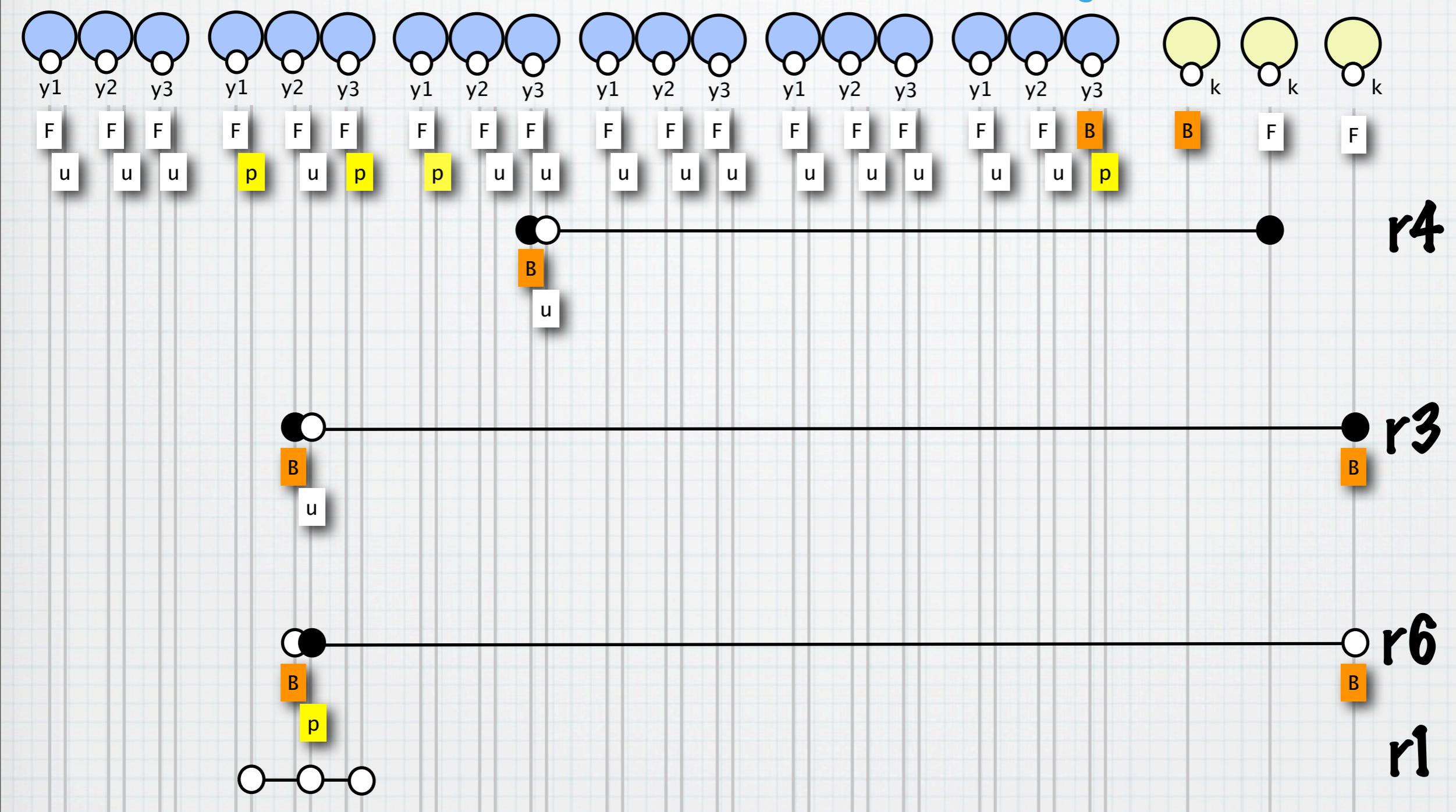
Concurrency



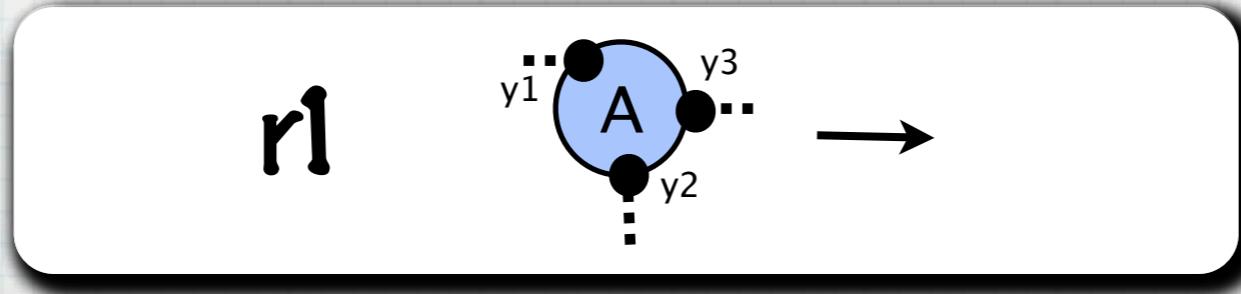
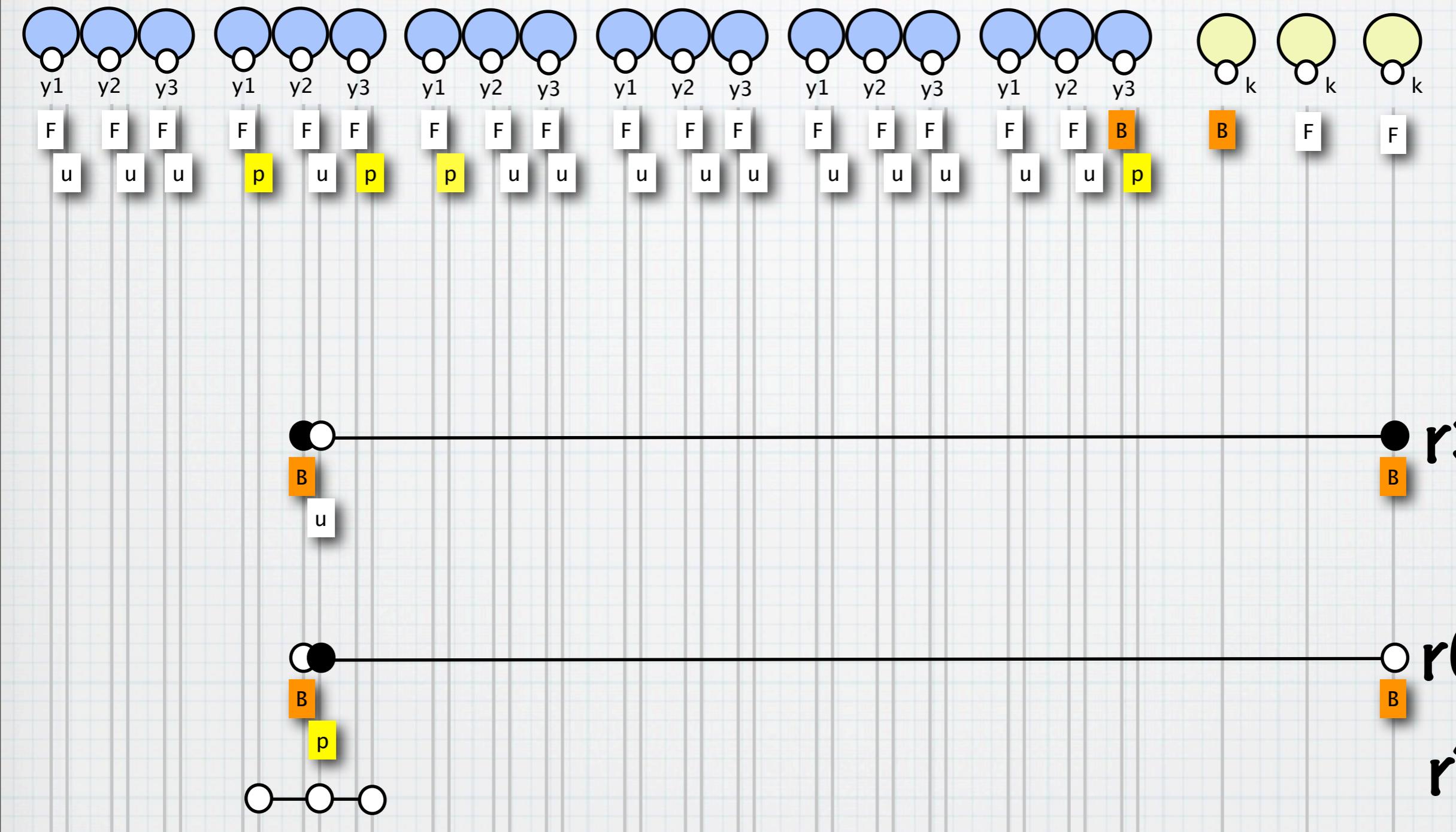
Concurrency



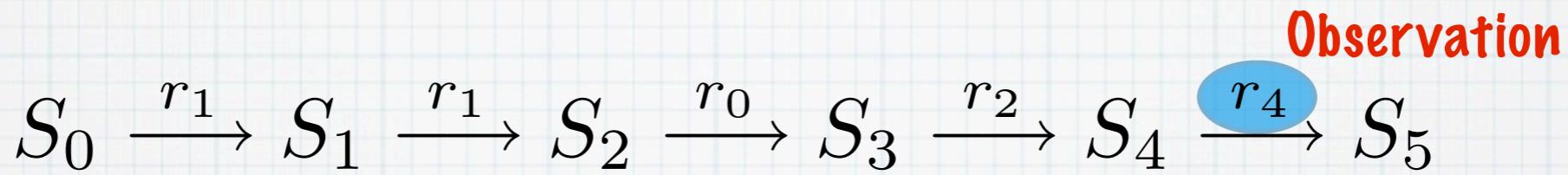
Concurrency



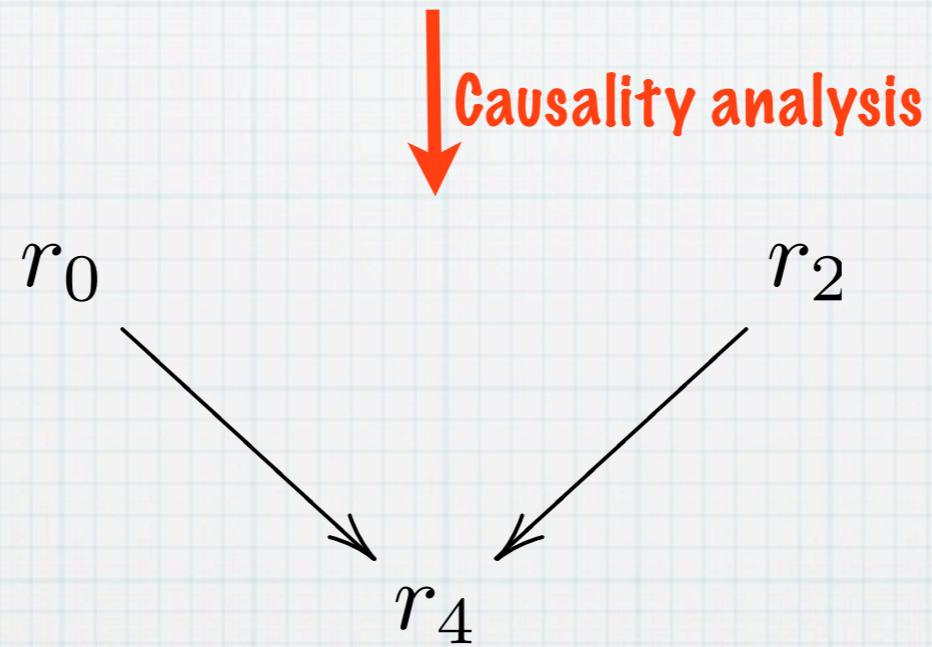
Concurrency



Story



A computation trace



The corresponding story

Stories as pathways

$$S_0 \xrightarrow{r_0} S_1 \xrightarrow{r_1} S_2 \xrightarrow{r_7} S_3 \xrightarrow{r_7} S_4 \xrightarrow{r_4} S_5$$

Trace 1

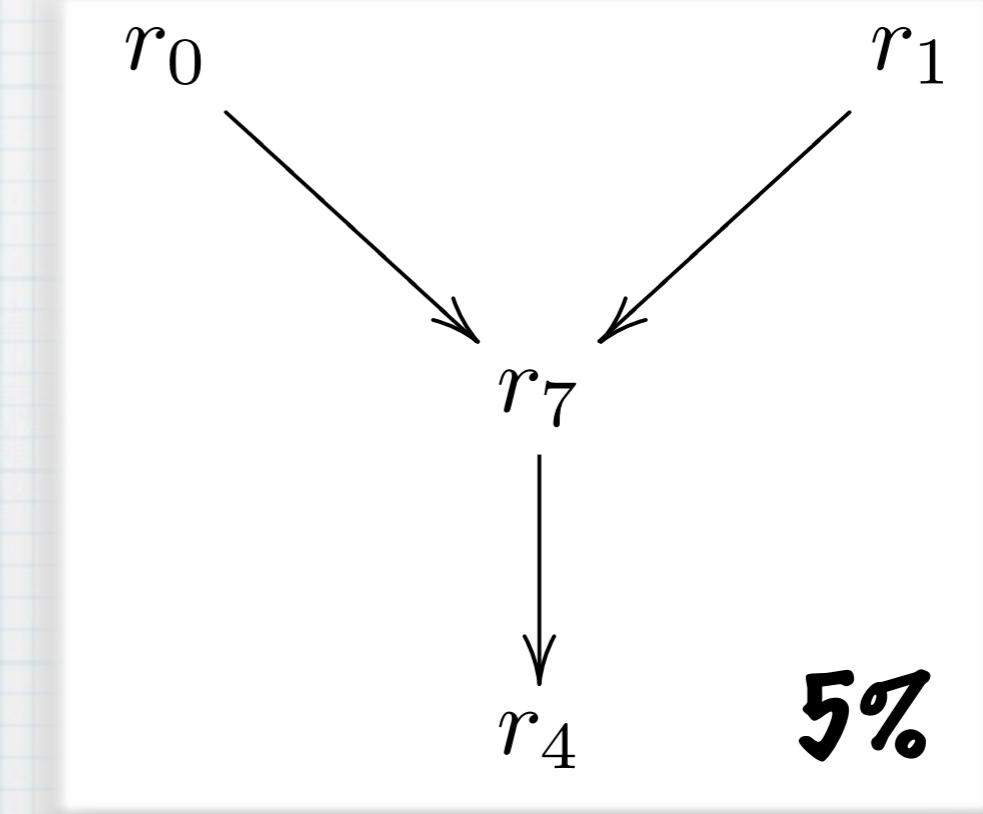
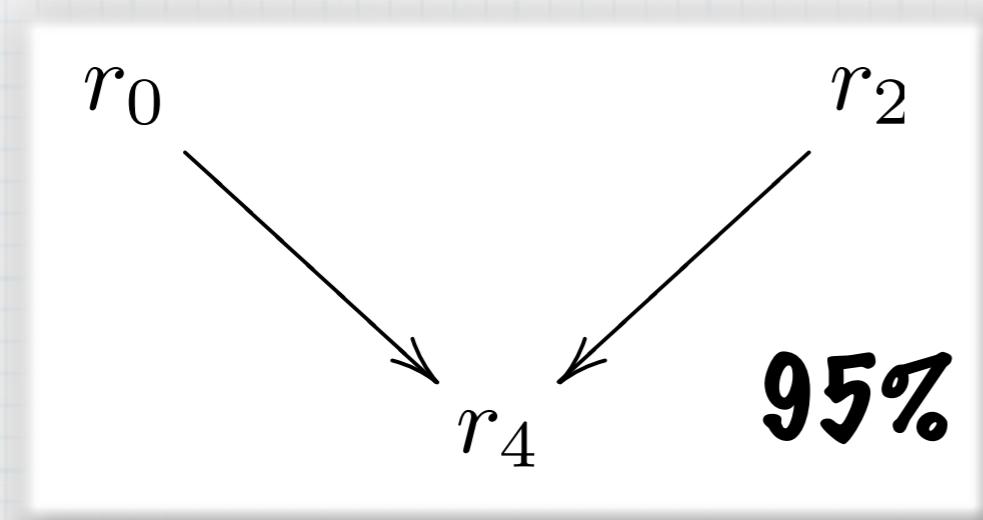
$$S_0 \xrightarrow{r_0} S_1 \xrightarrow{r_1} S_2 \xrightarrow{r_{12}} S_3 \xrightarrow{r_7} S_4 \xrightarrow{r_7} S_5 \xrightarrow{r_4} S_6$$

Trace 2

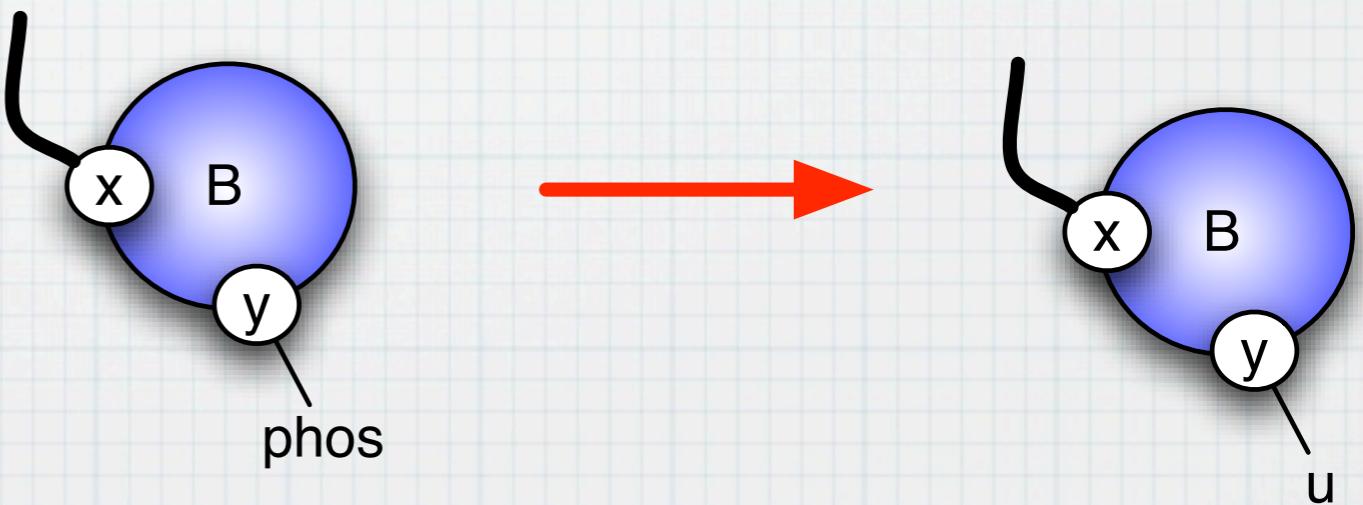
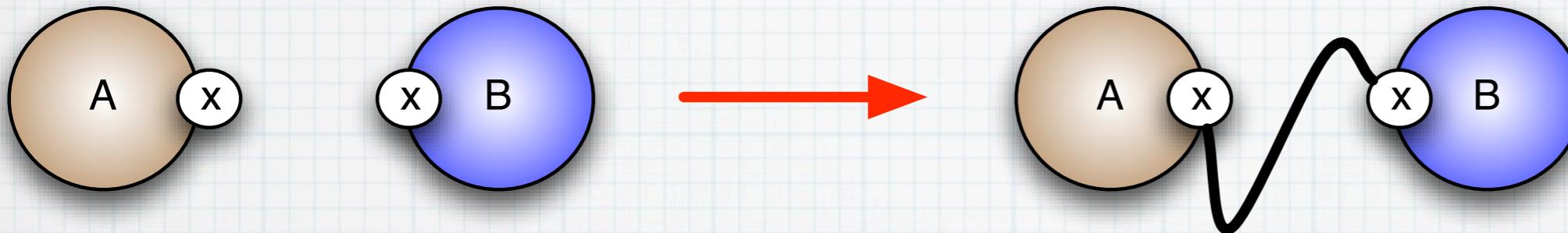
...

$$S_0 \xrightarrow{r_0} S_1 \xrightarrow{r_2} S_2 \xrightarrow{r_4} S_3$$

Trace n

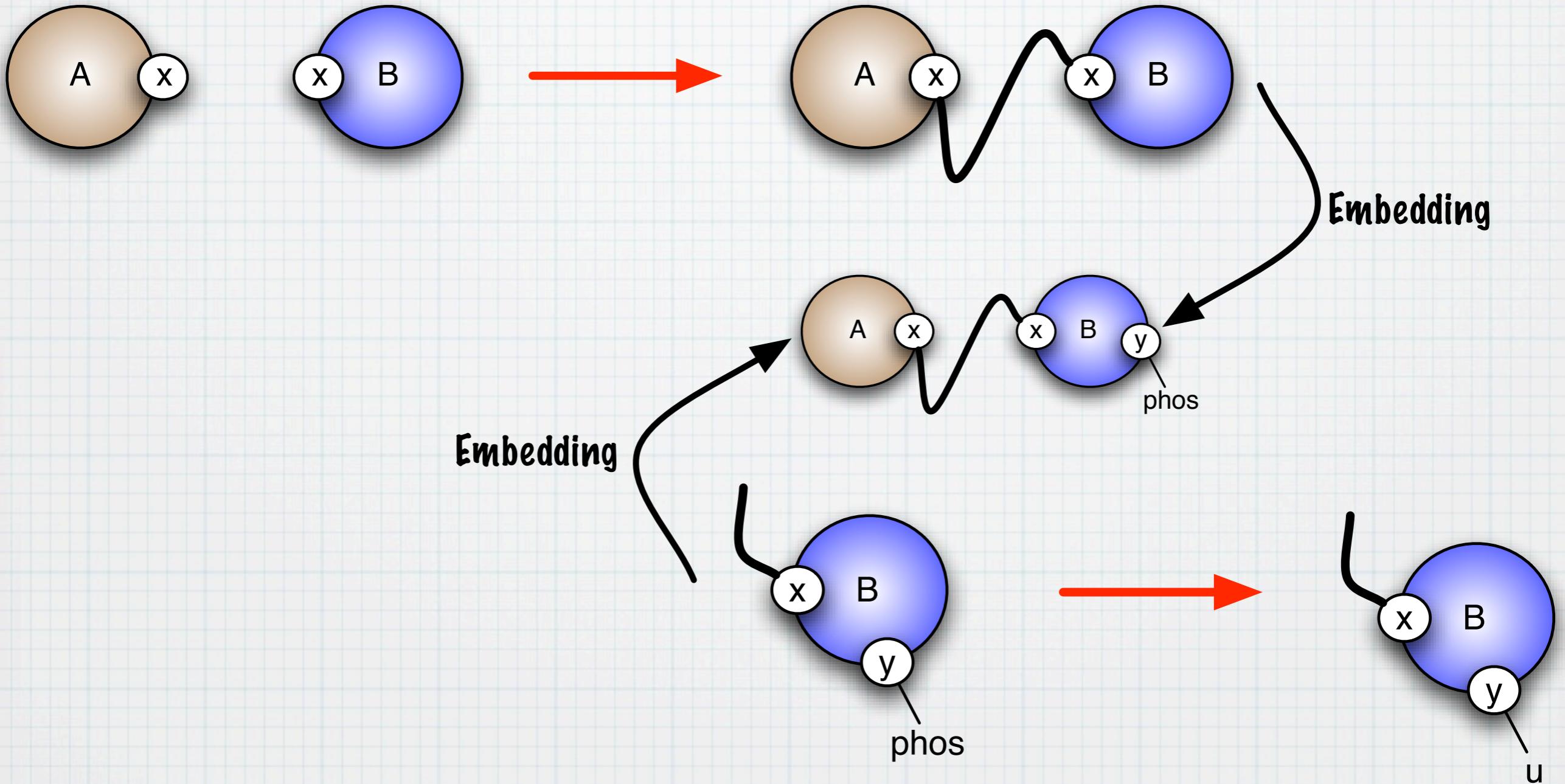


Activation



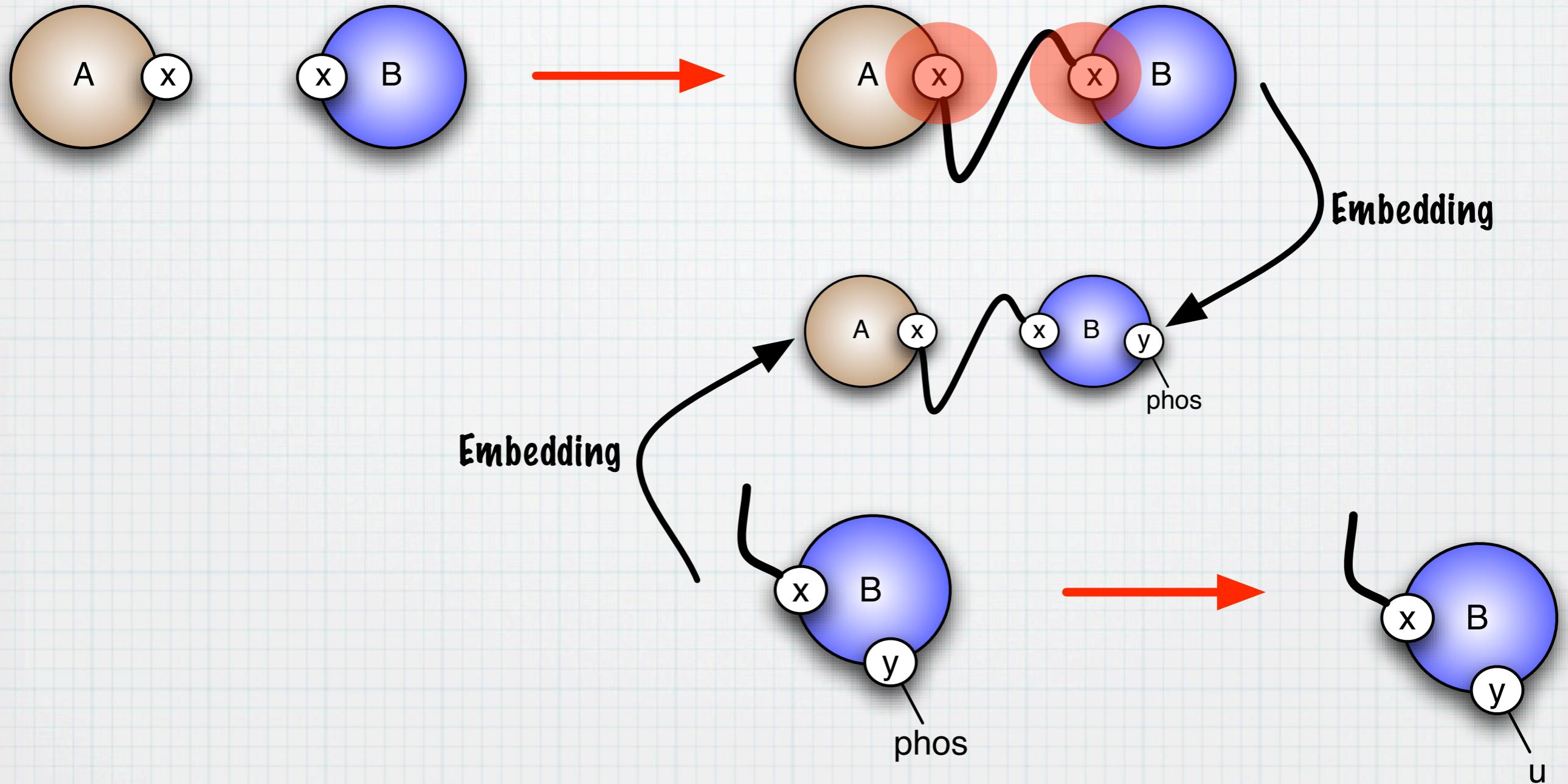
Can the upper rule increase the activity of the second one?

Activation



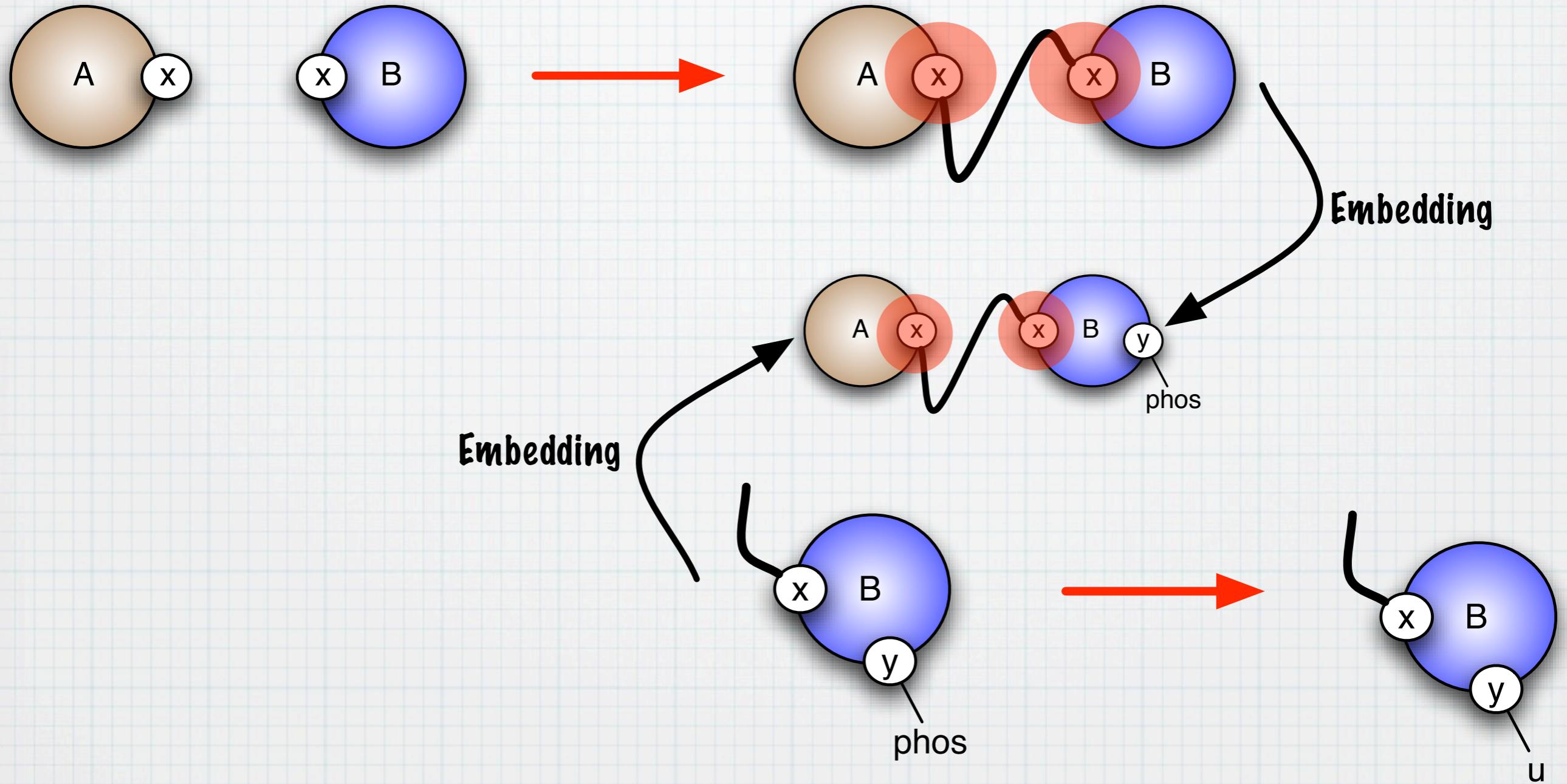
Can the upper rule increase the activity of the second one?

Activation



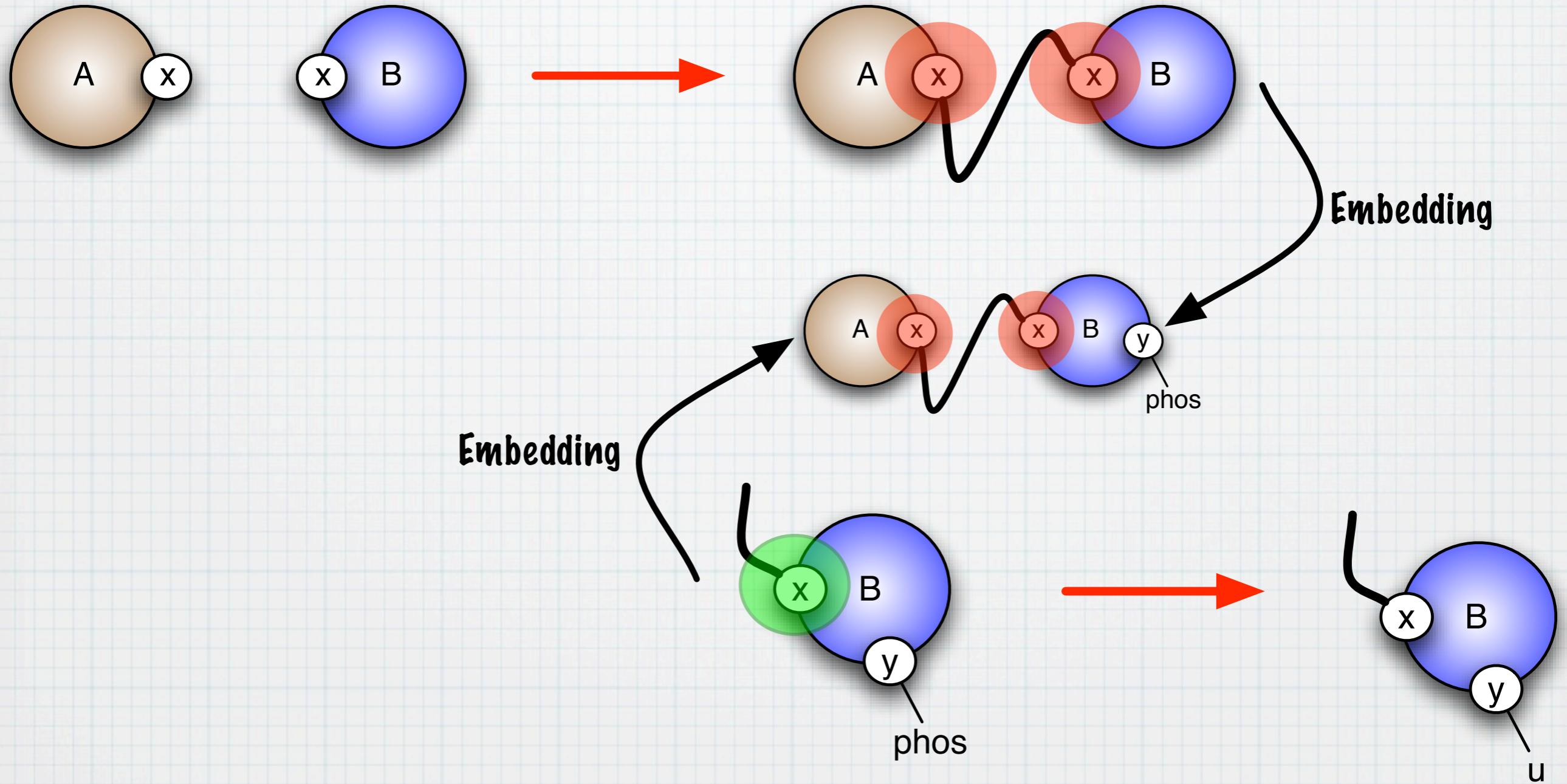
Can the upper rule increase the activity of the second one?

Activation



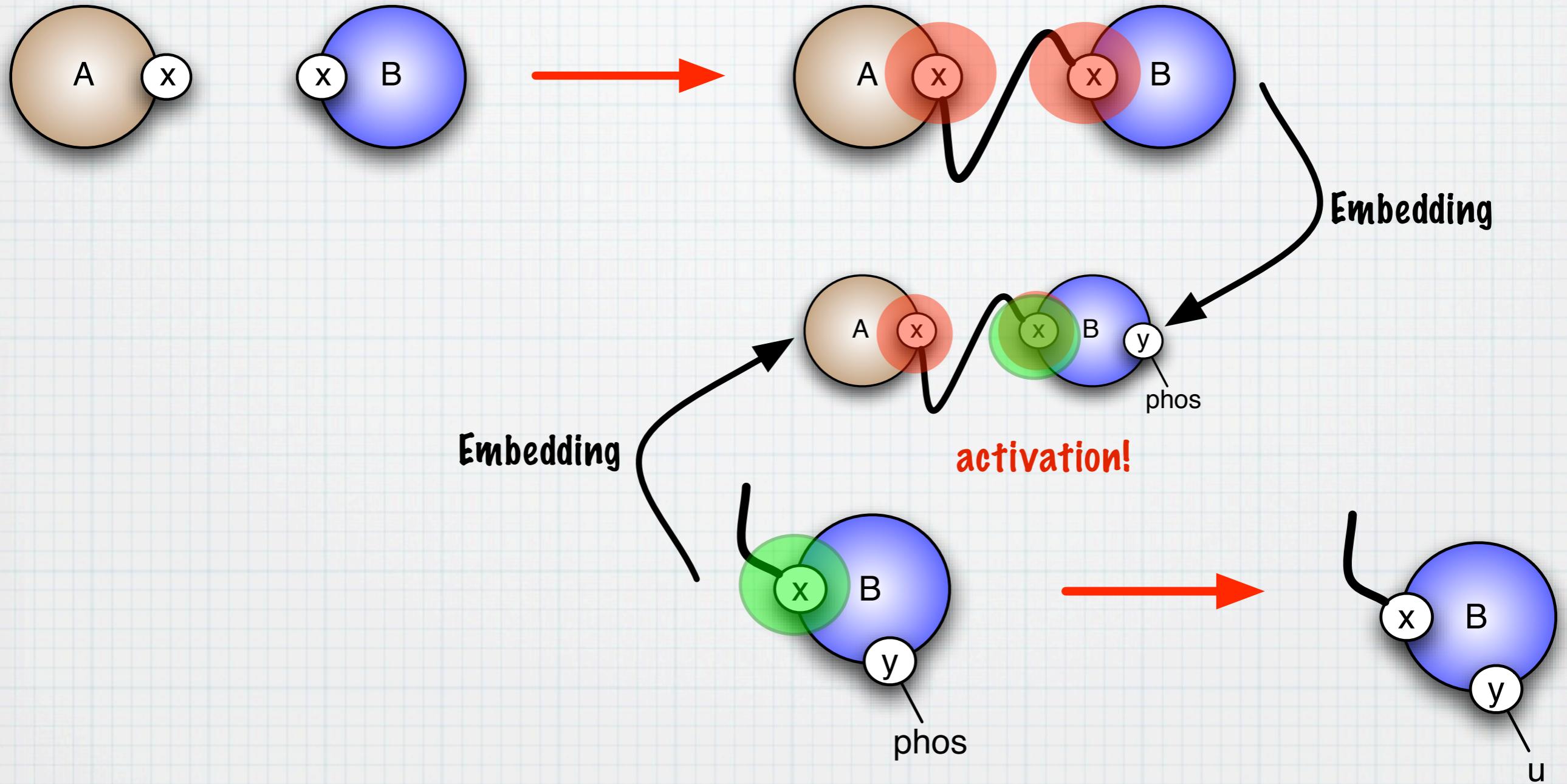
Can the upper rule increase the activity of the second one?

Activation



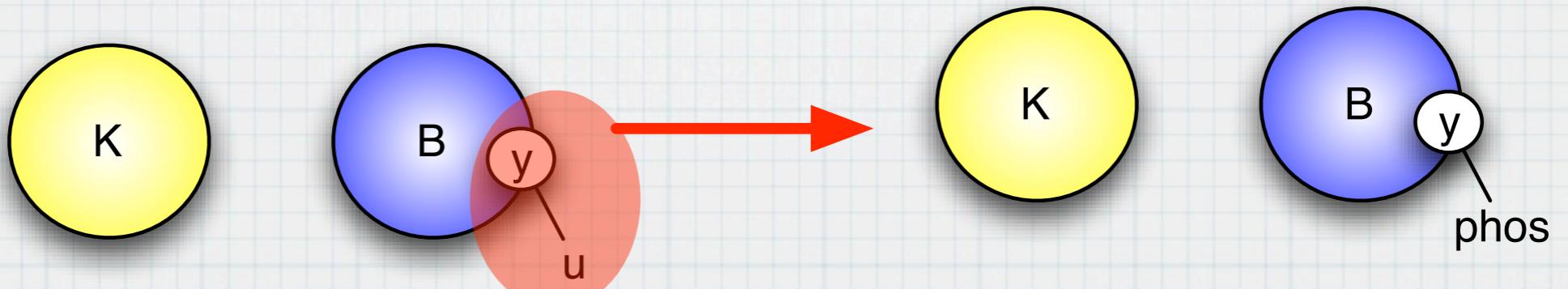
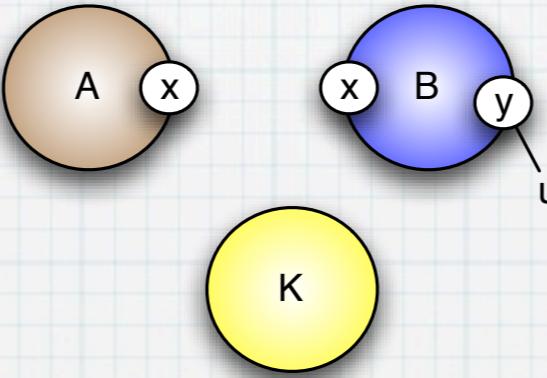
Can the upper rule increase the activity of the second one?

Activation

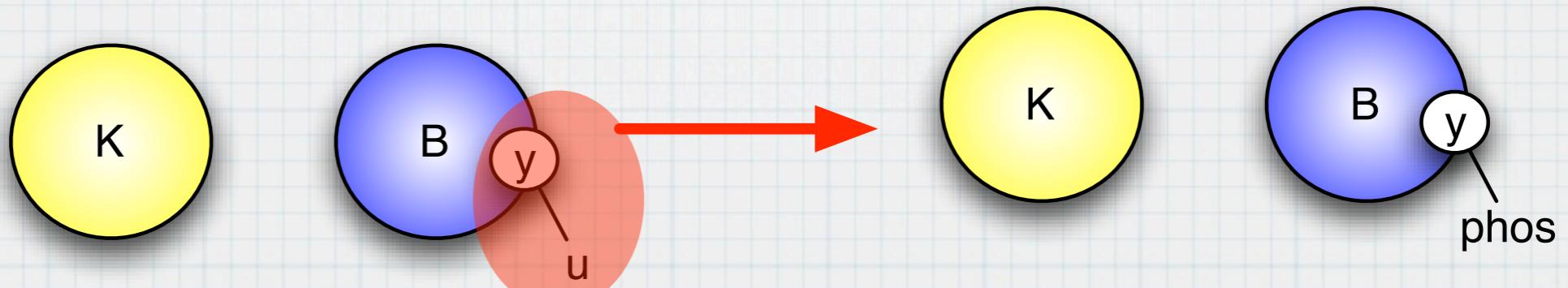
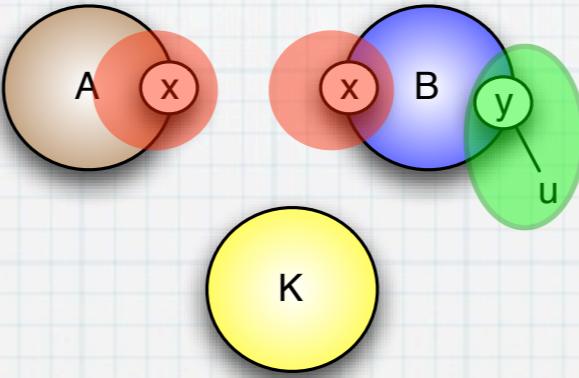
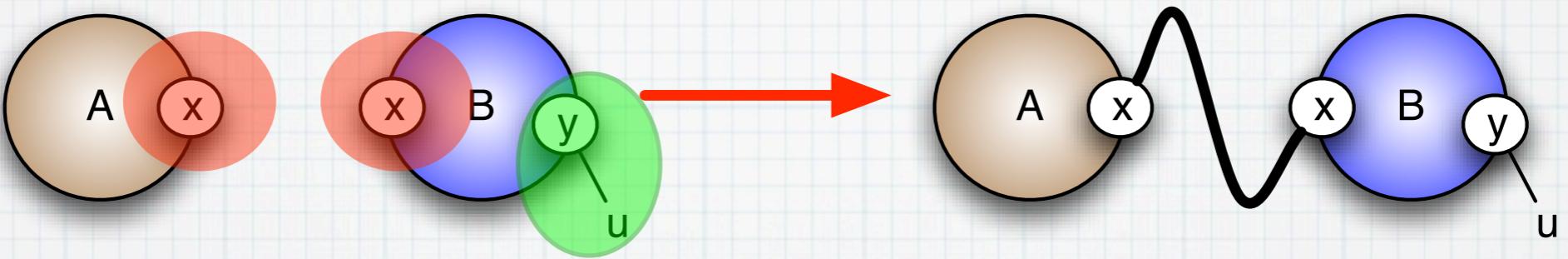


Can the upper rule increase the activity of the second one?

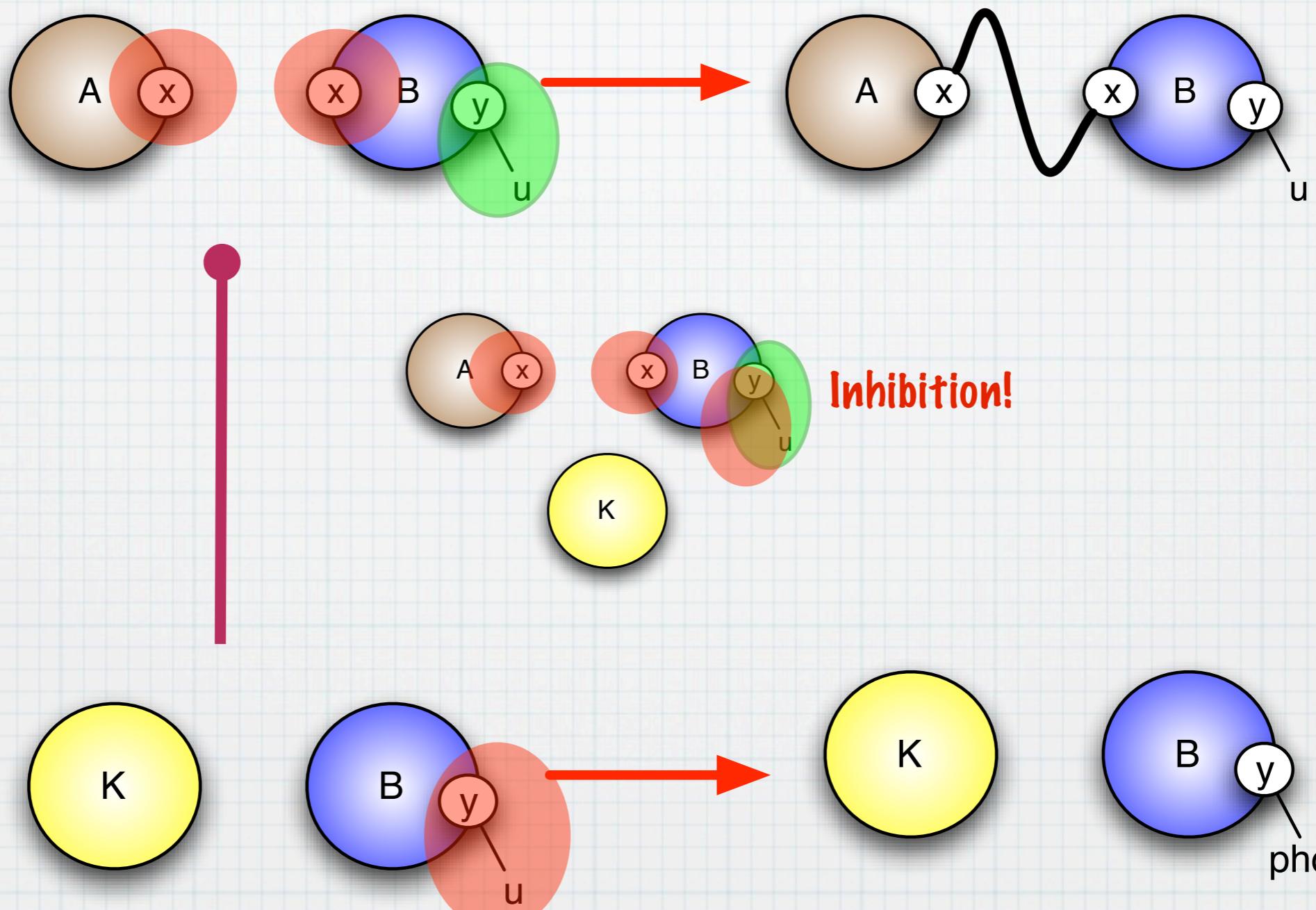
Inhibition



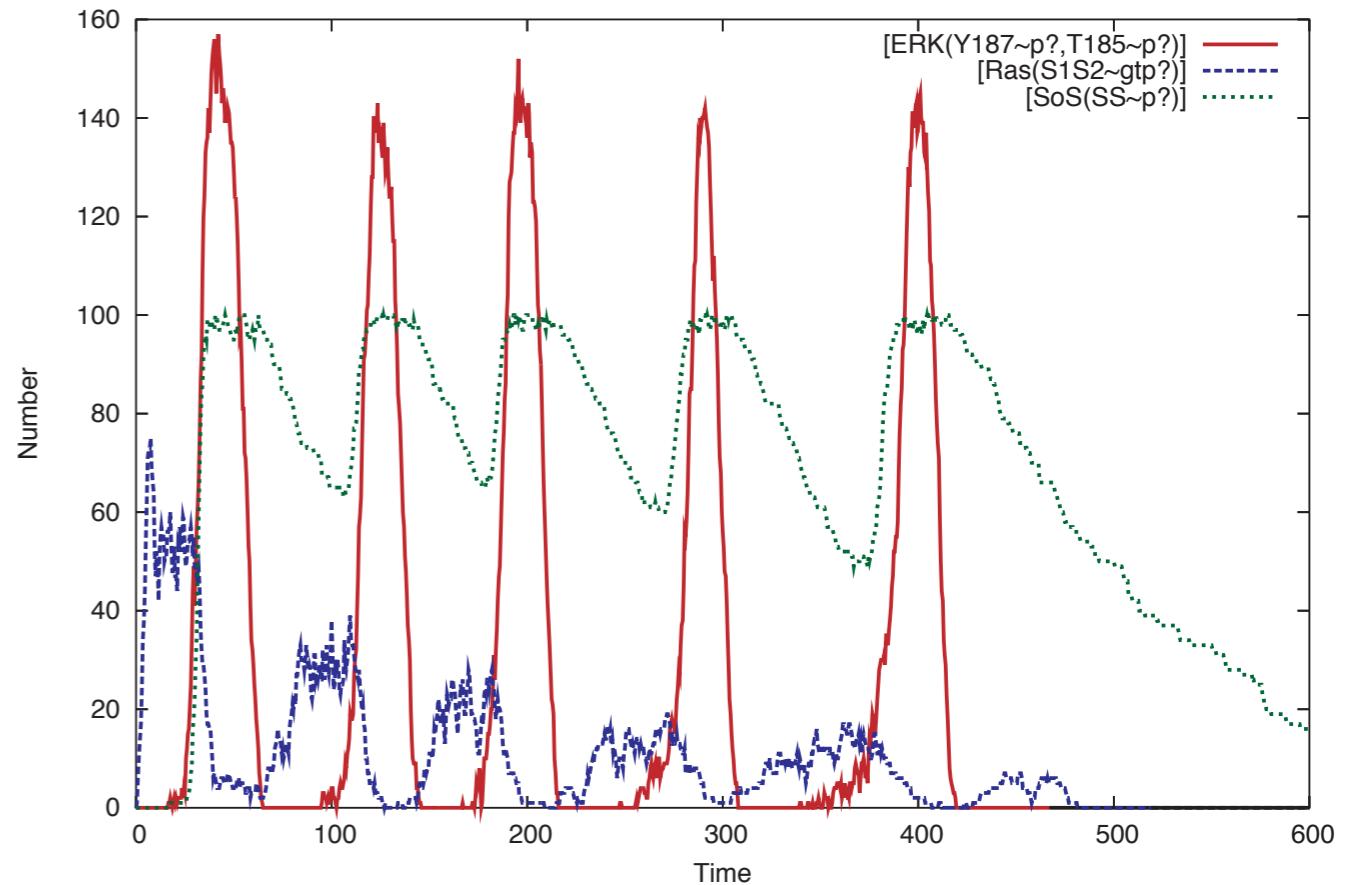
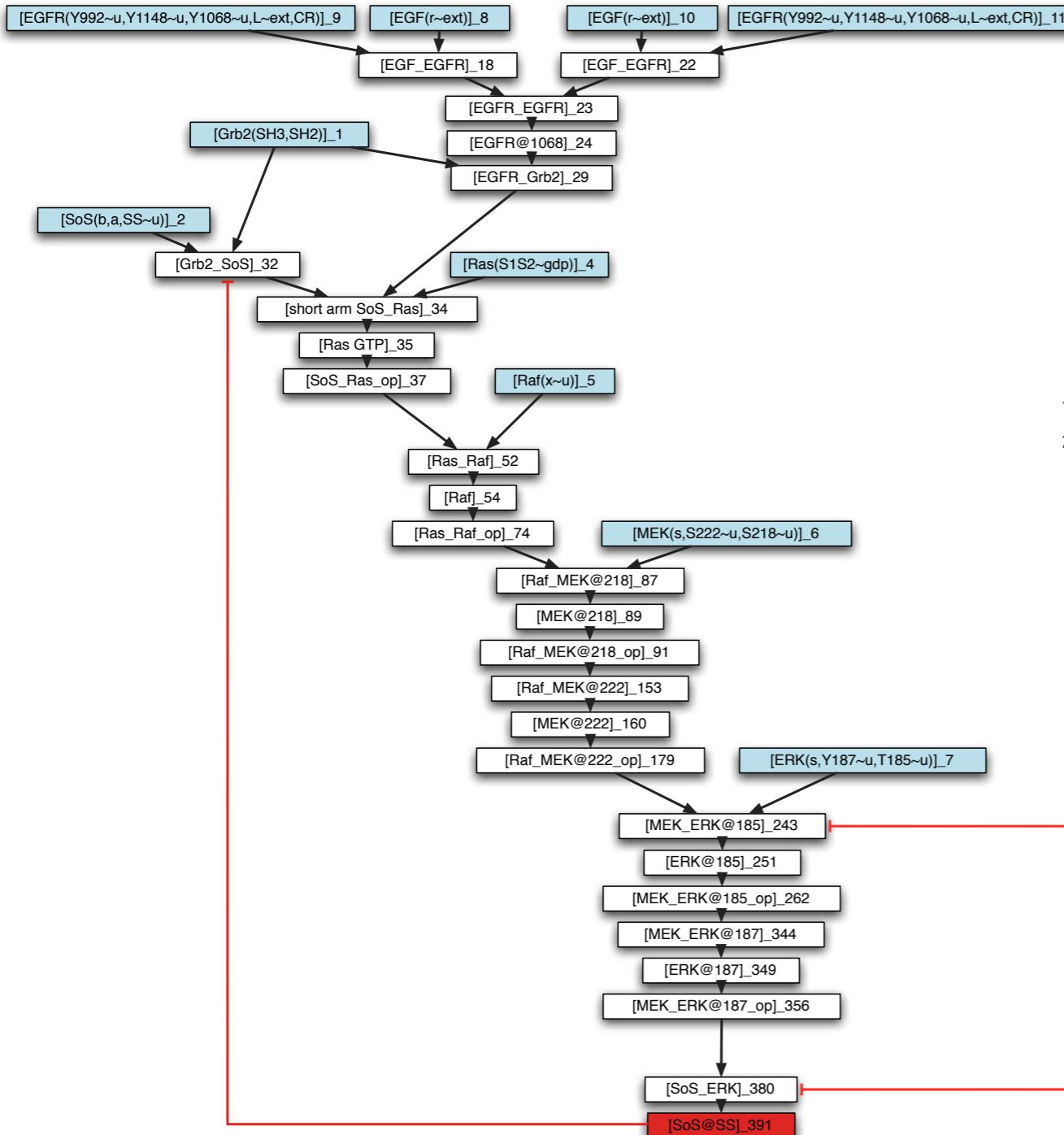
Inhibition



Inhibition



Causality and dynamics



The big picture

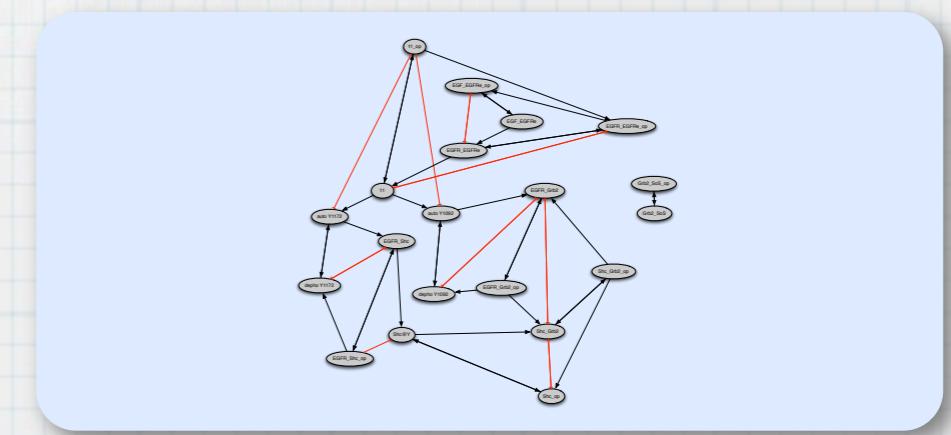
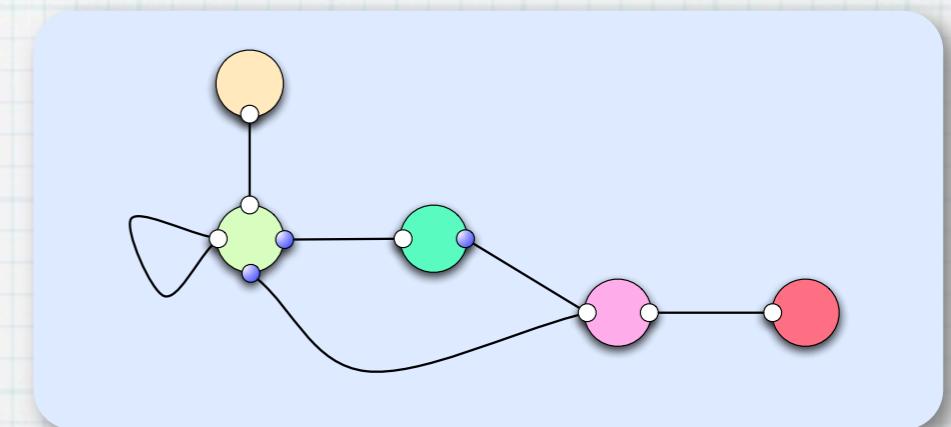
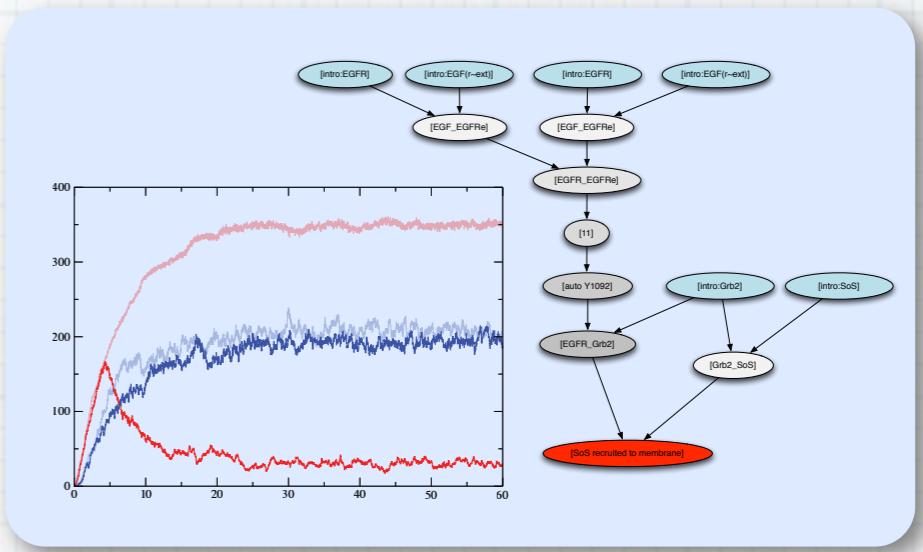
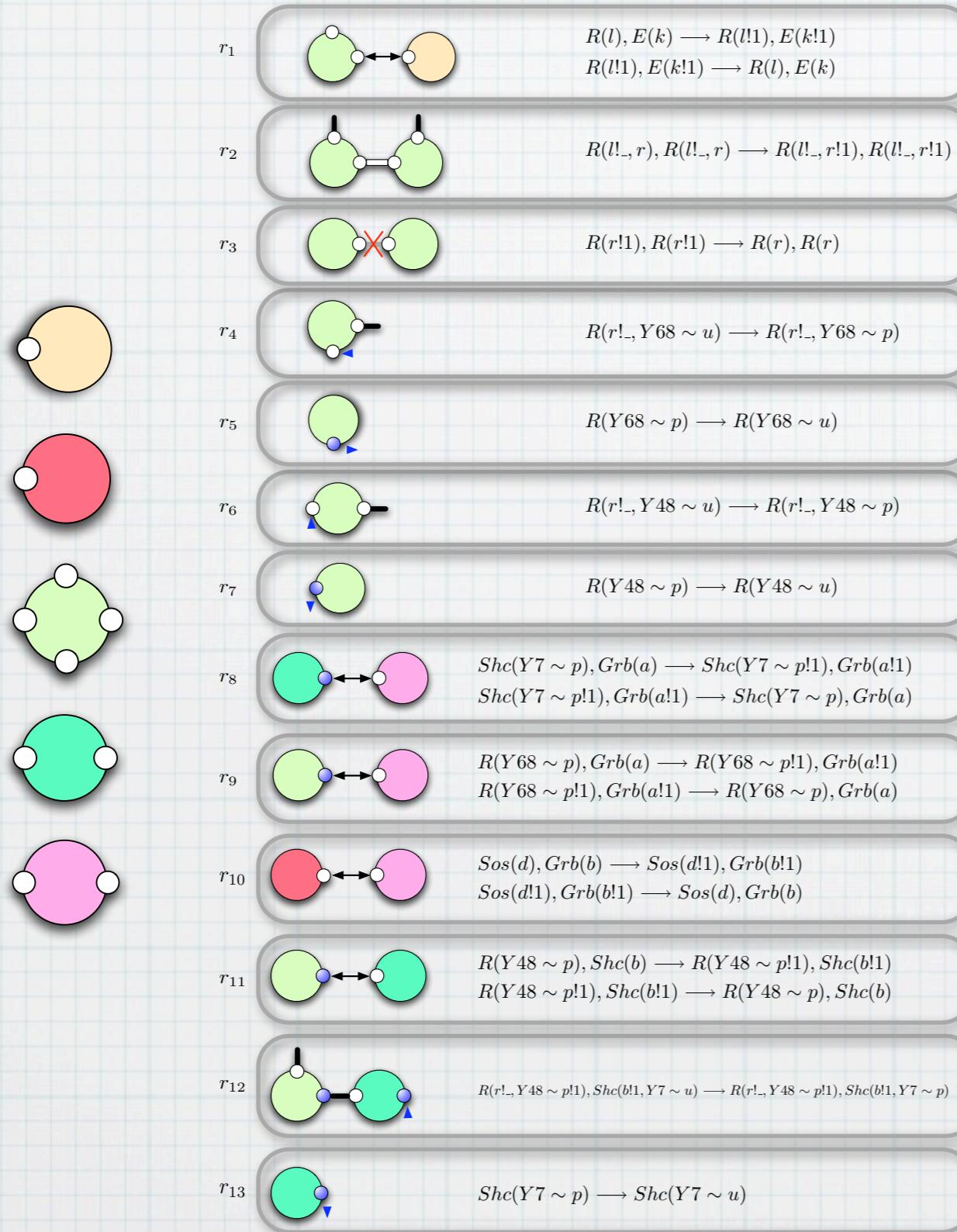
molecular
agents

plus

rules describing molecular actions

generate

maps:
system views of the possible





Vincent Danos



Walter Fontana



Jérôme Feret



Russ Harmer



Vincent Danos



Walter Fontana



Jérôme Feret



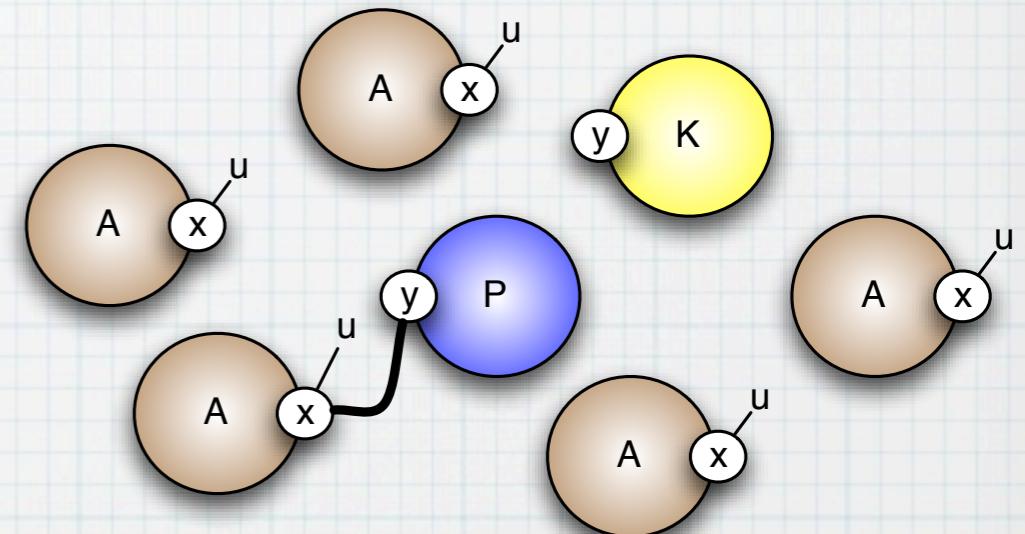
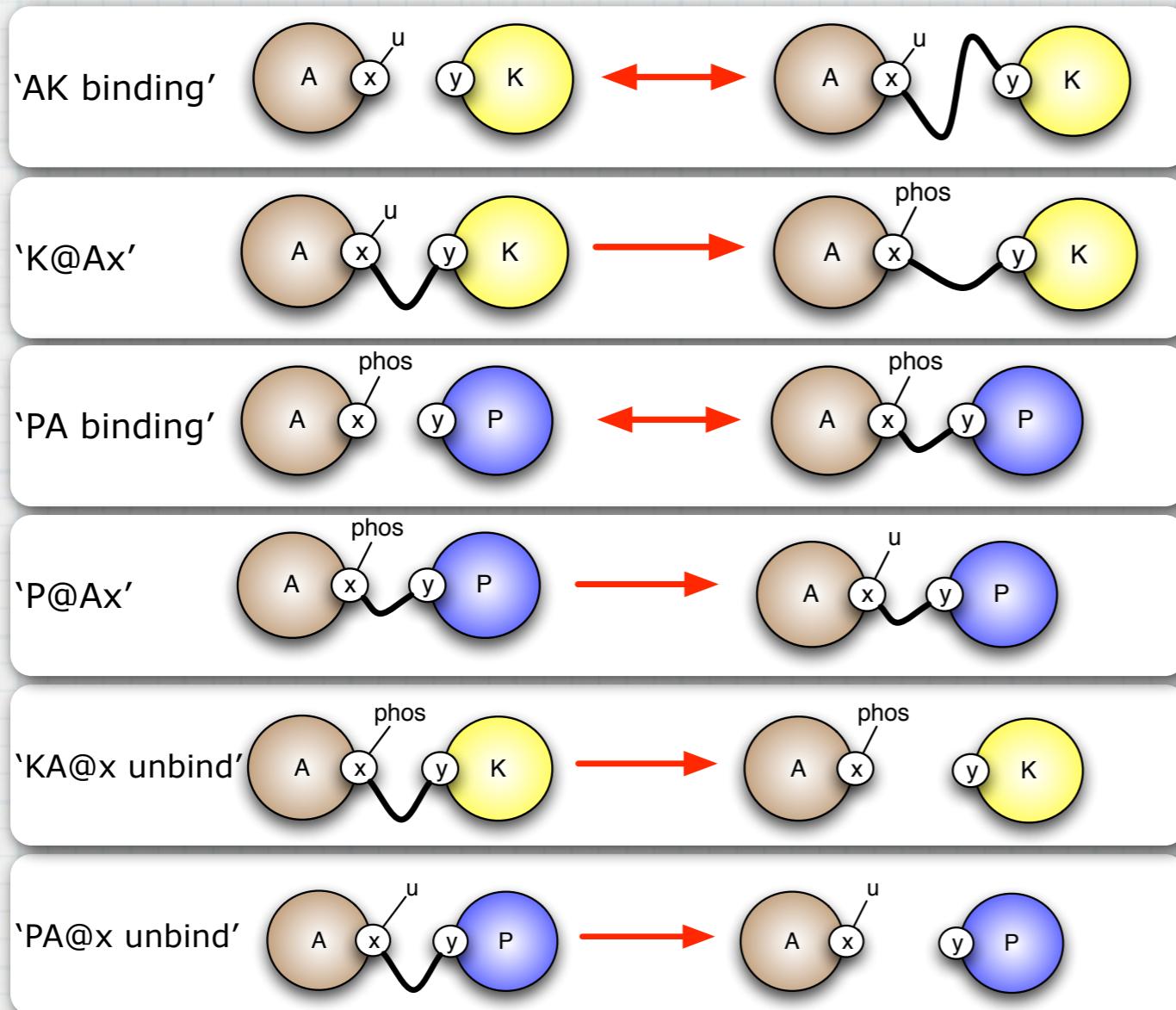
Russ Harmer



Scalability issue...

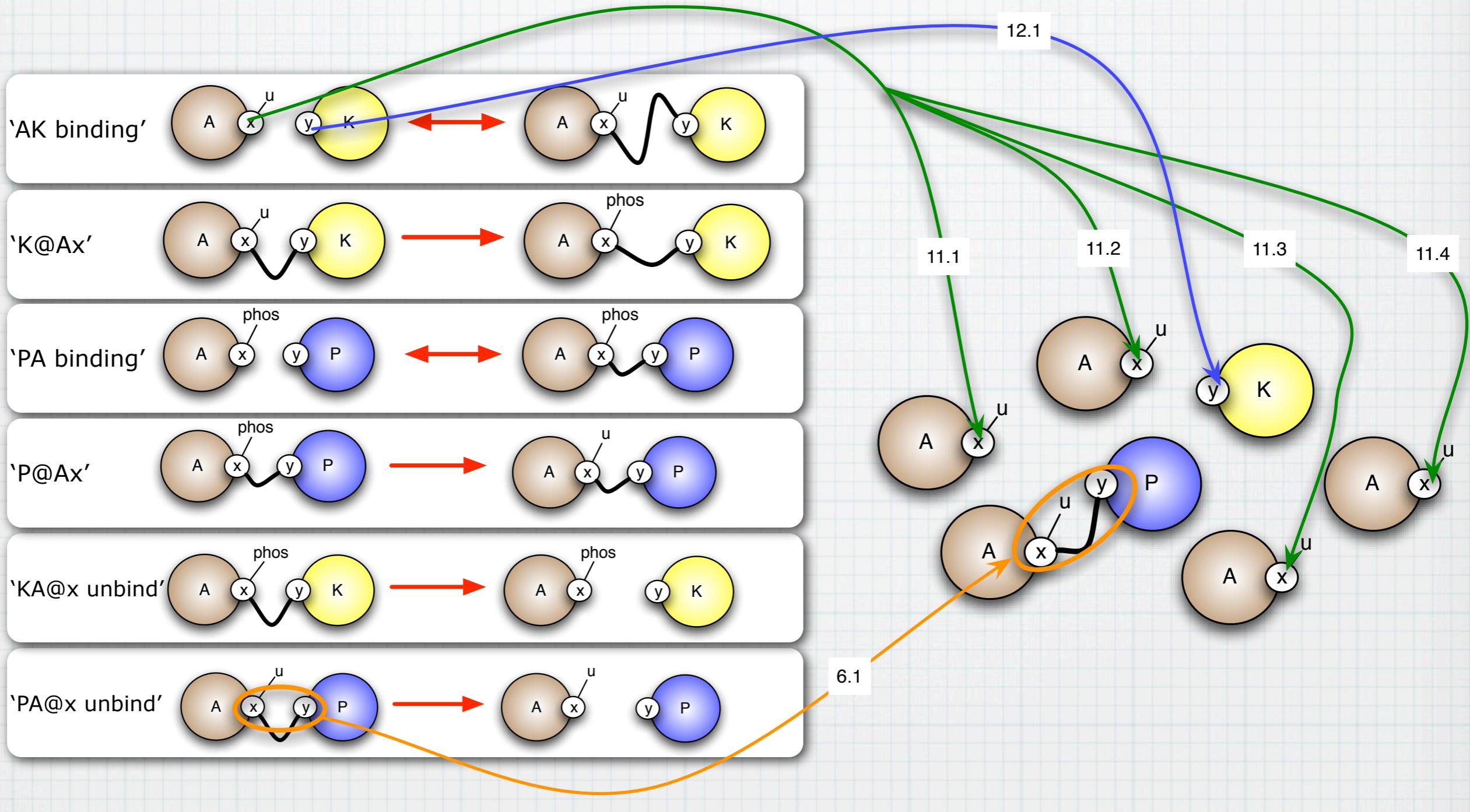
How not to crash your computer?

Cost of rule application



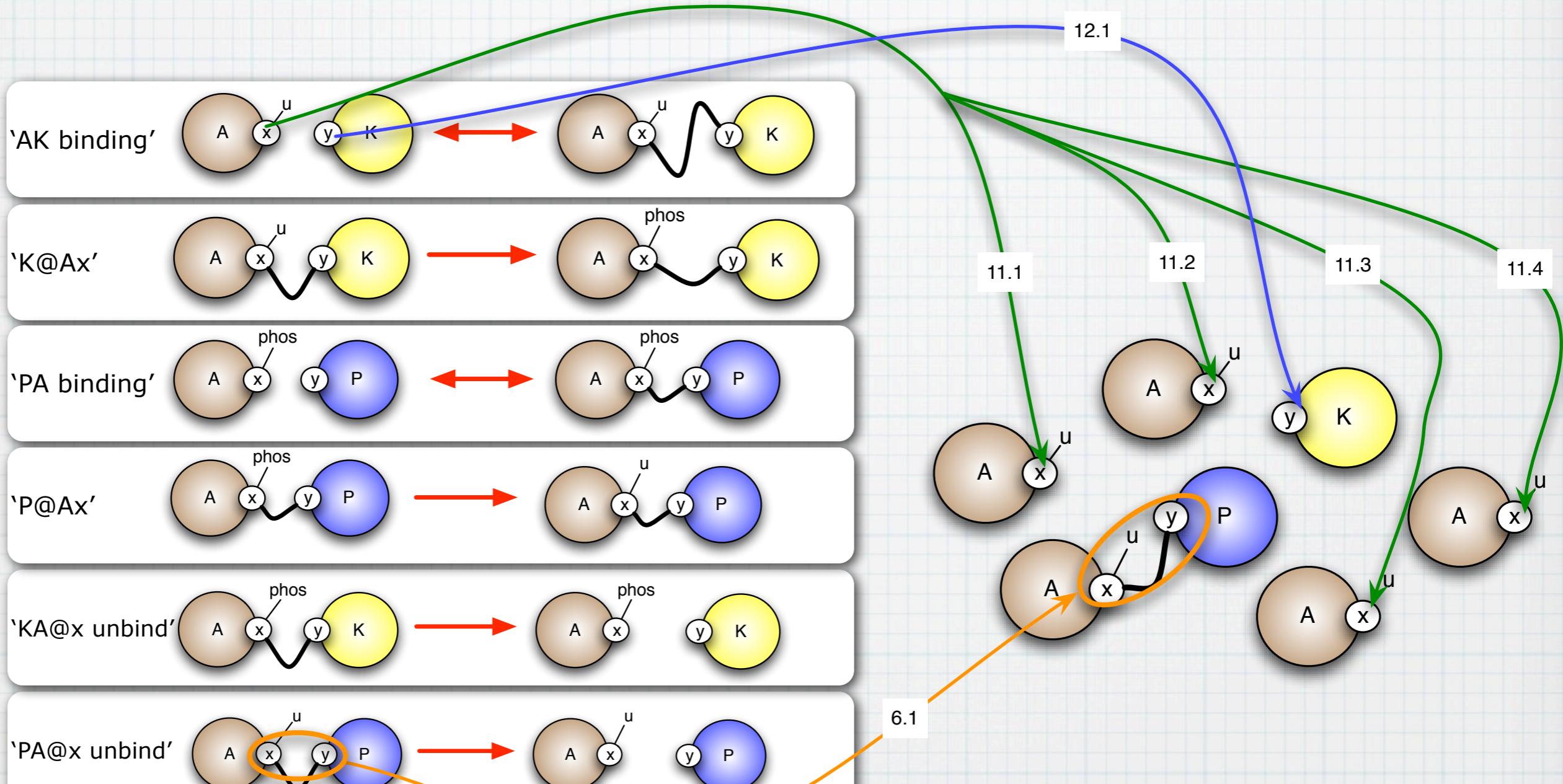
Depends on the number of rules (hundreds) and on the number of agents in the solution (hundreds of thousands!): not scalable...

Cost of rule application



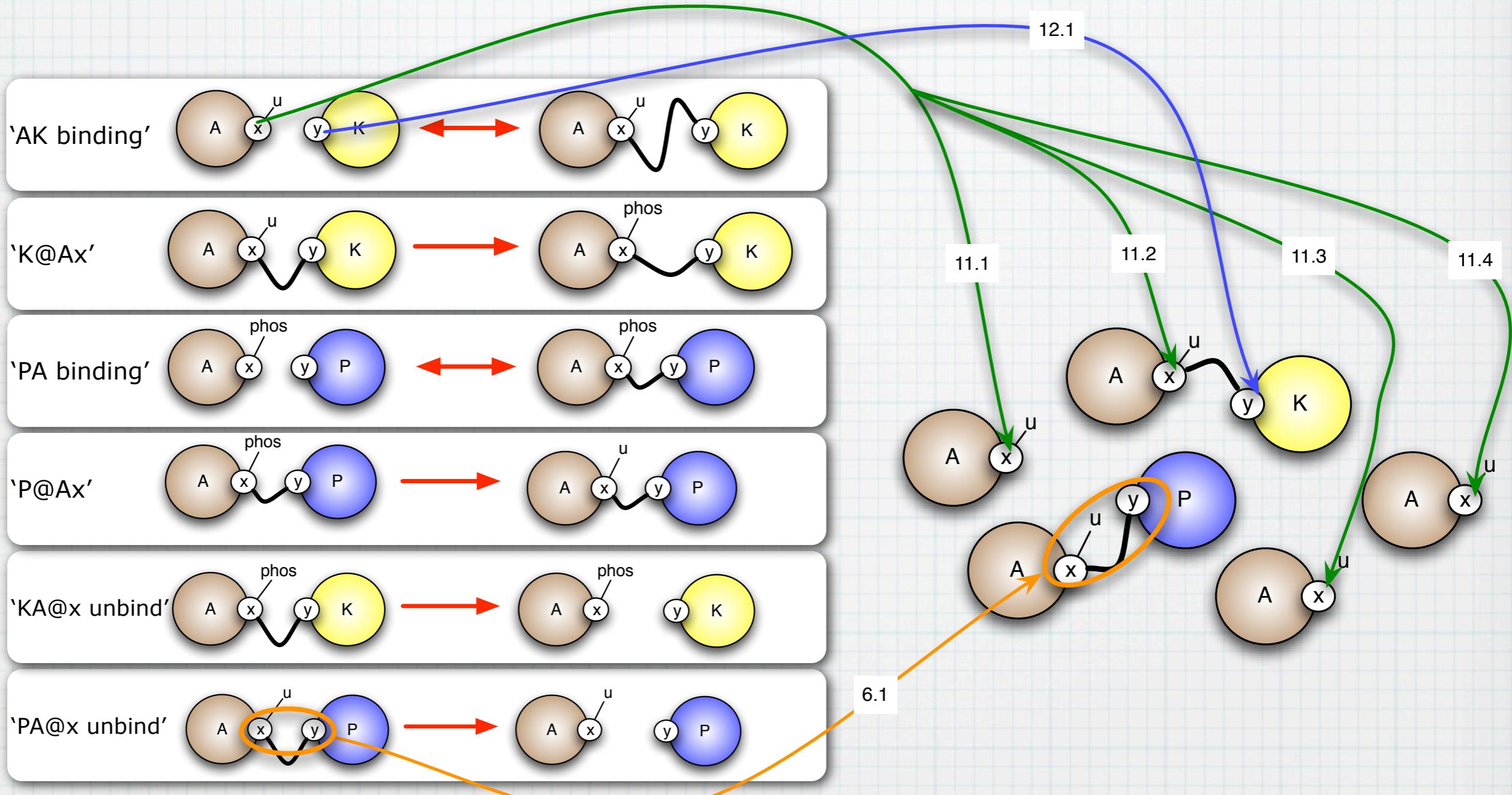
Idea: pre-compute every possible rule instances during initialization!

Cost of update



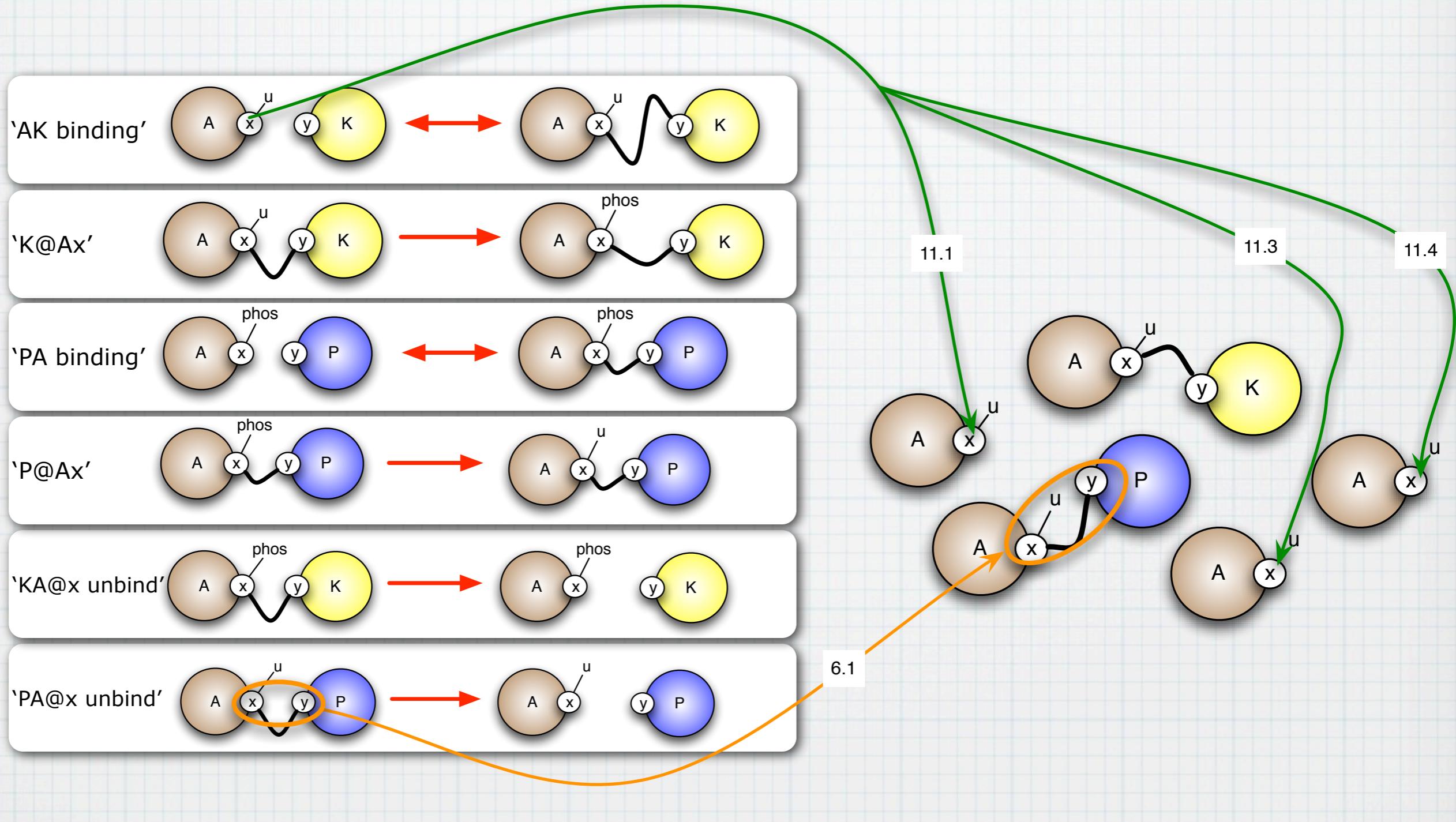
Negative update is purely local: just erase injections using the modified sites...

Cost of update



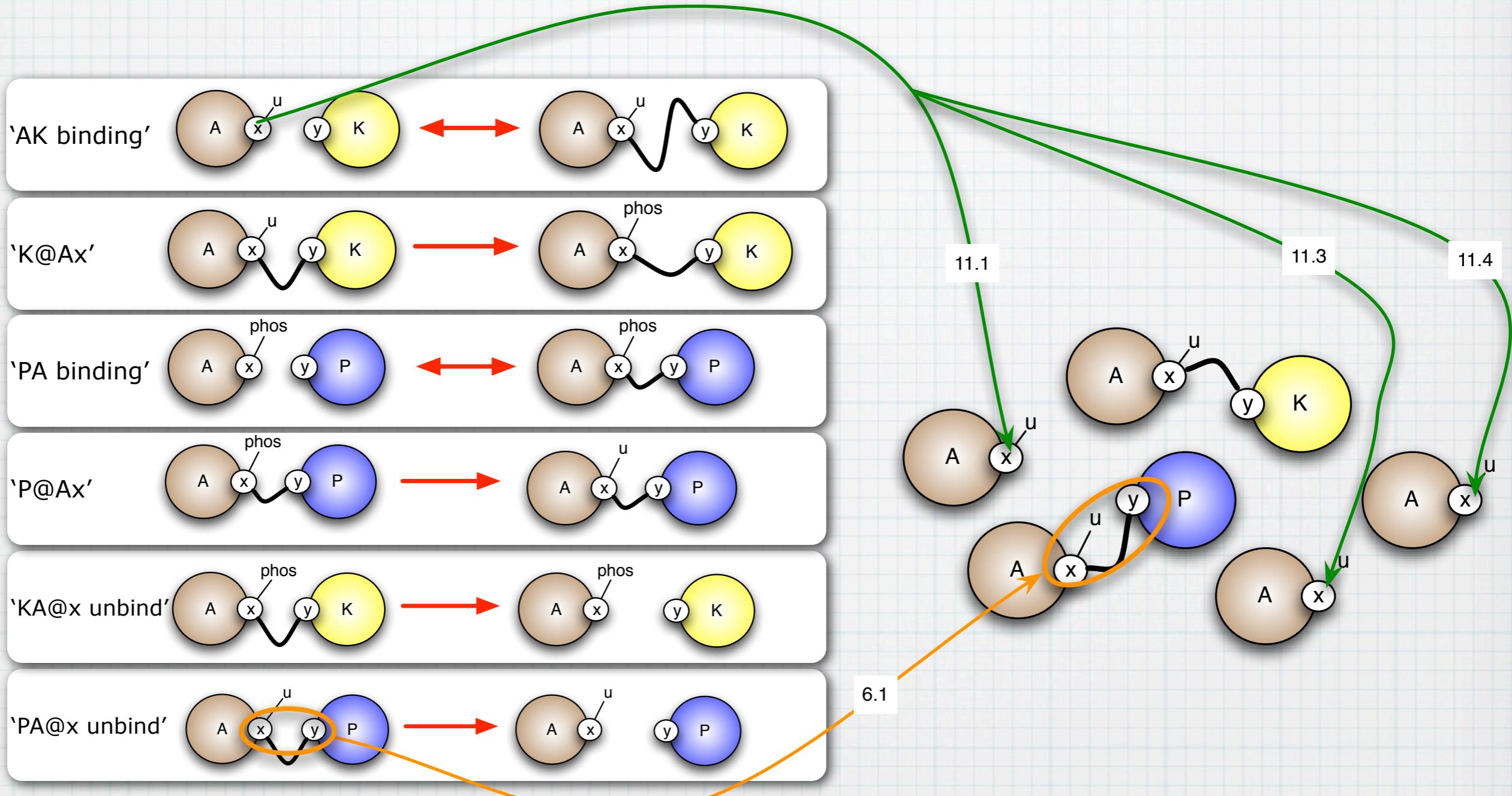
Negative update is purely local: just erase injections using the modified sites...

Cost of update



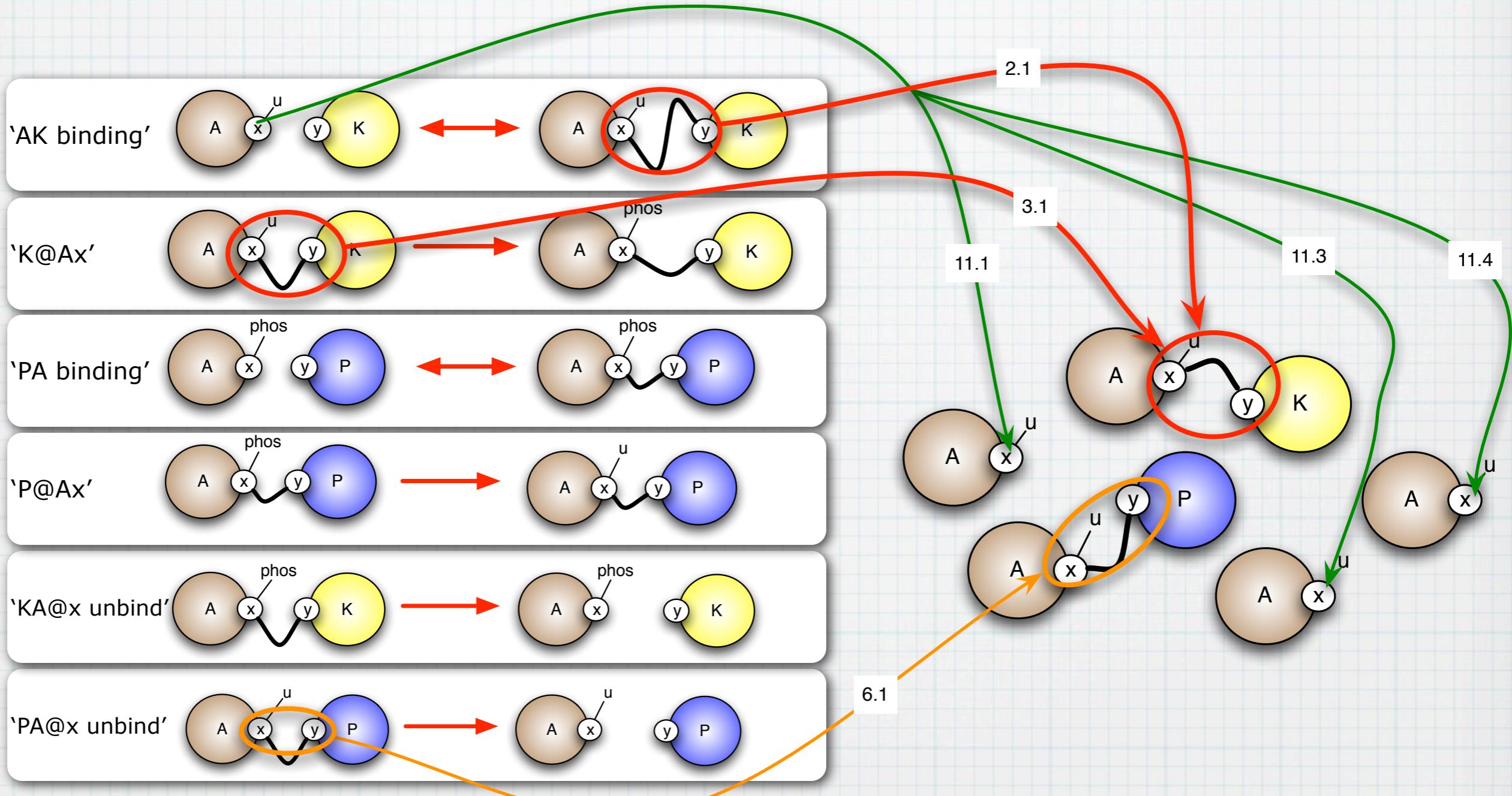
Negative update is purely local: just erase injections using the modified sites...

Cost of update



Positive update: we need now to update rules that may use the new state...

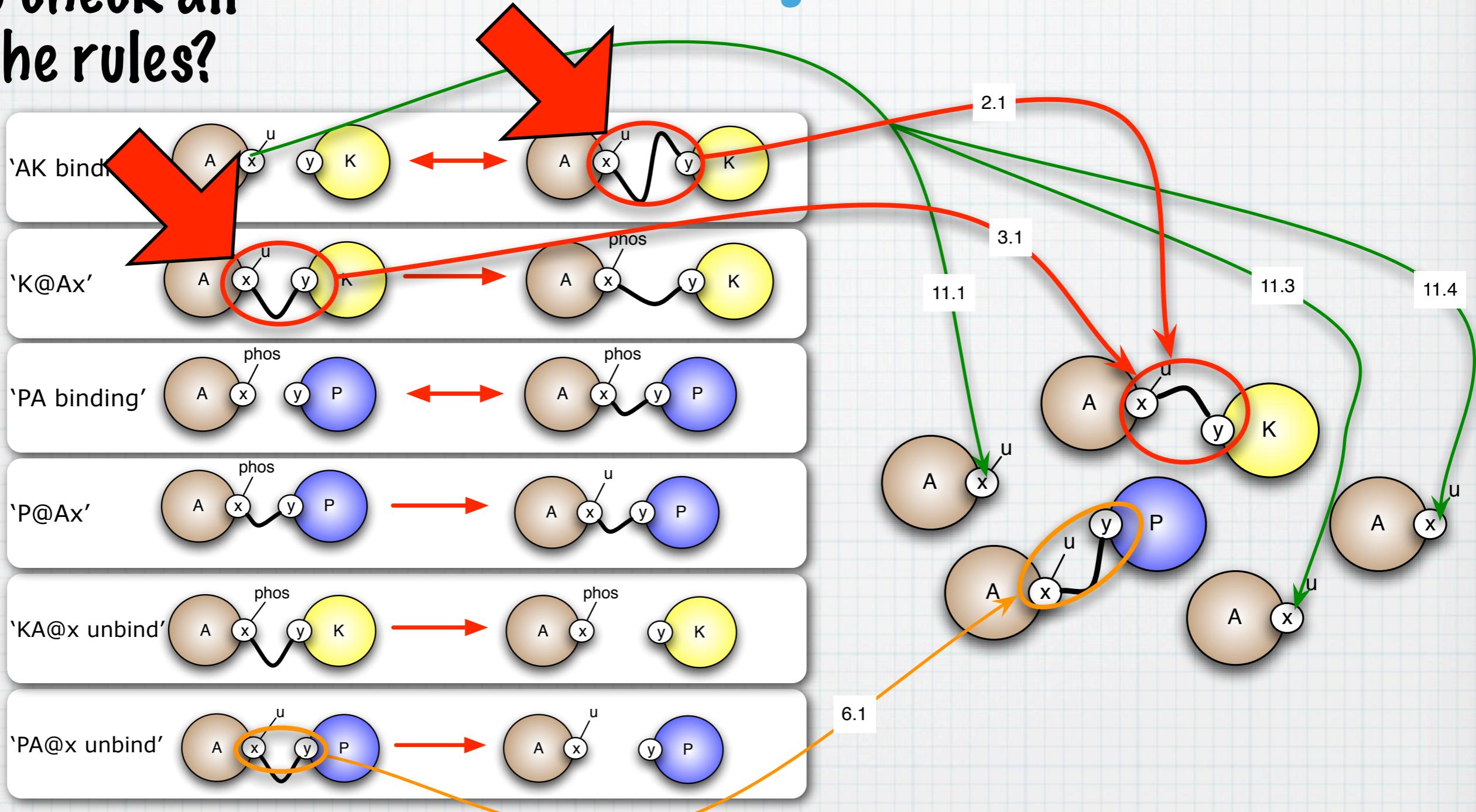
Cost of update



Positive update: we need now to update rules that may use the new state...

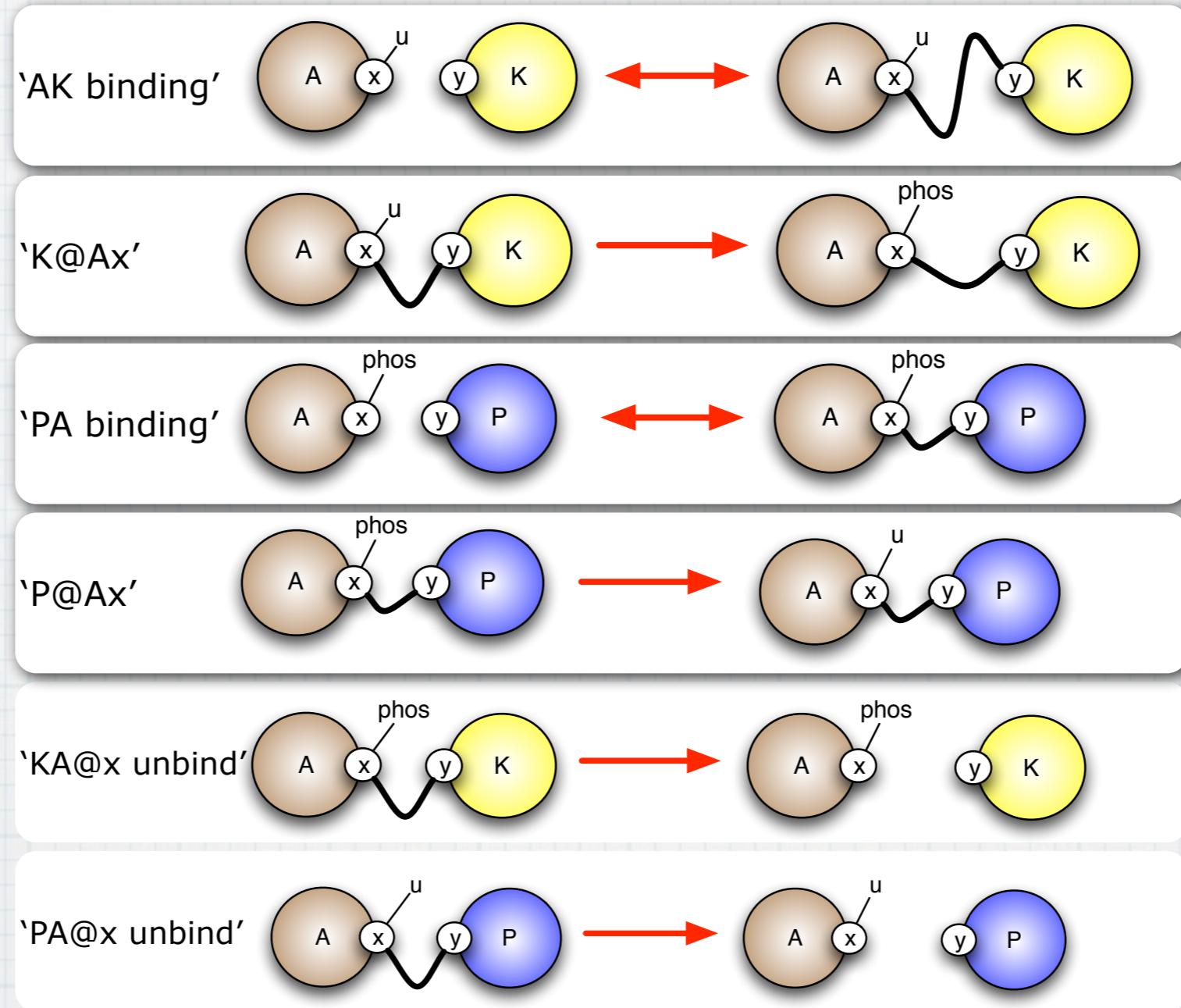
Do we have
to check all
the rules?

Cost of update



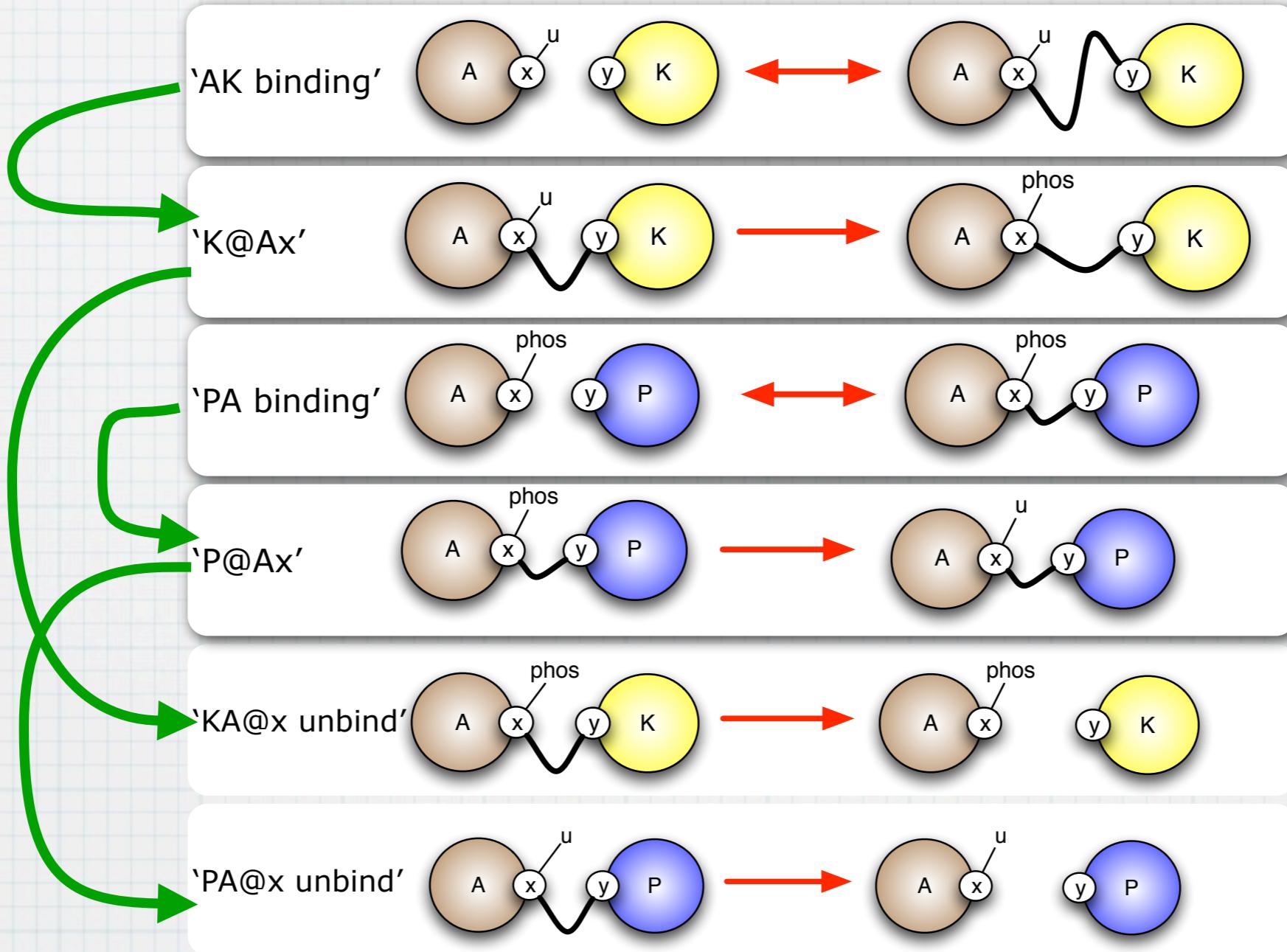
Positive update: we need now to update rules that may use the new state...

Wake-up map



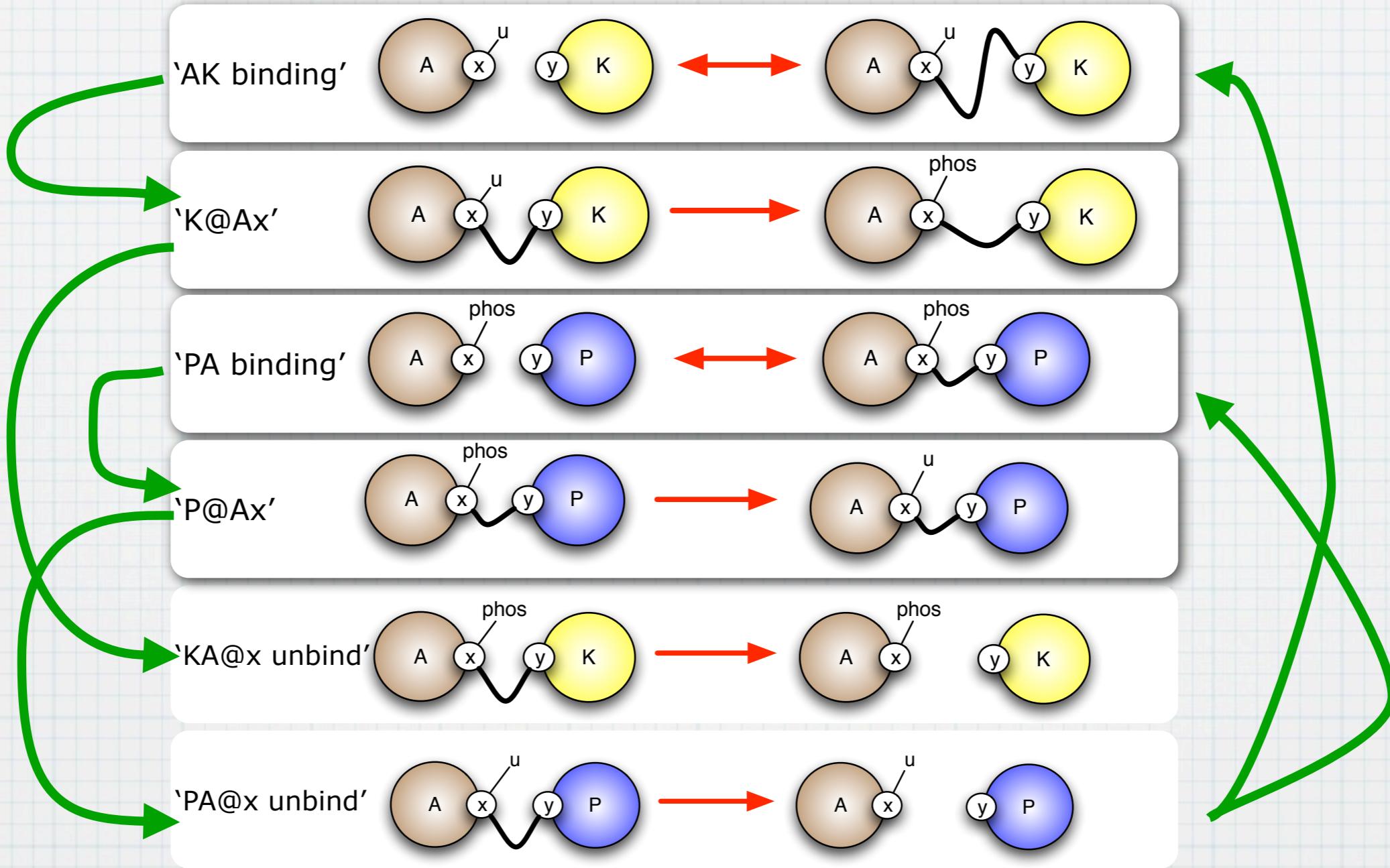
What is the potential impact (activation) of the application of a given rule on the activity of other rules?

Wake-up map



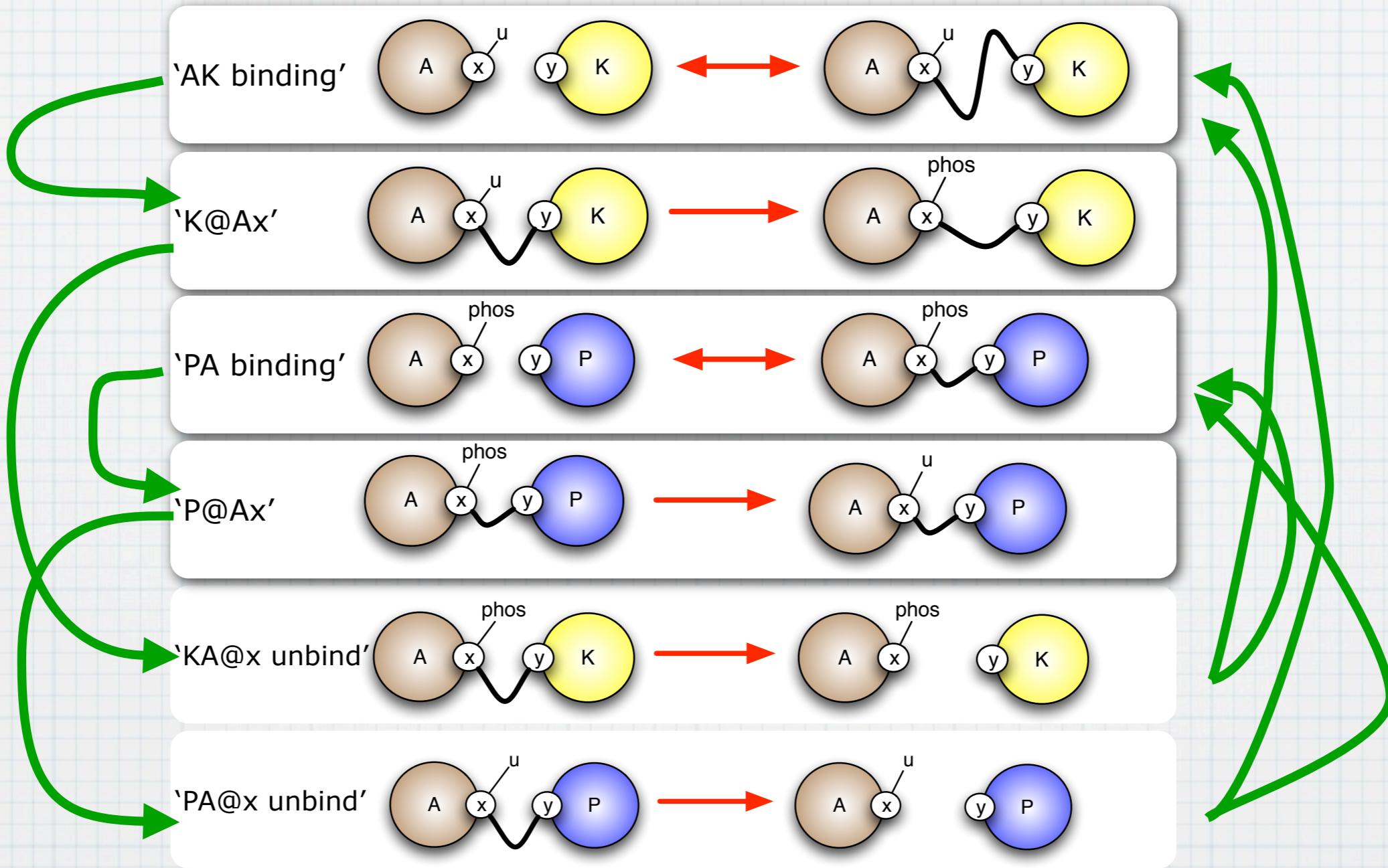
What is the potential impact (activation) of the application of a given rule on the activity of other rules?

Wake-up map



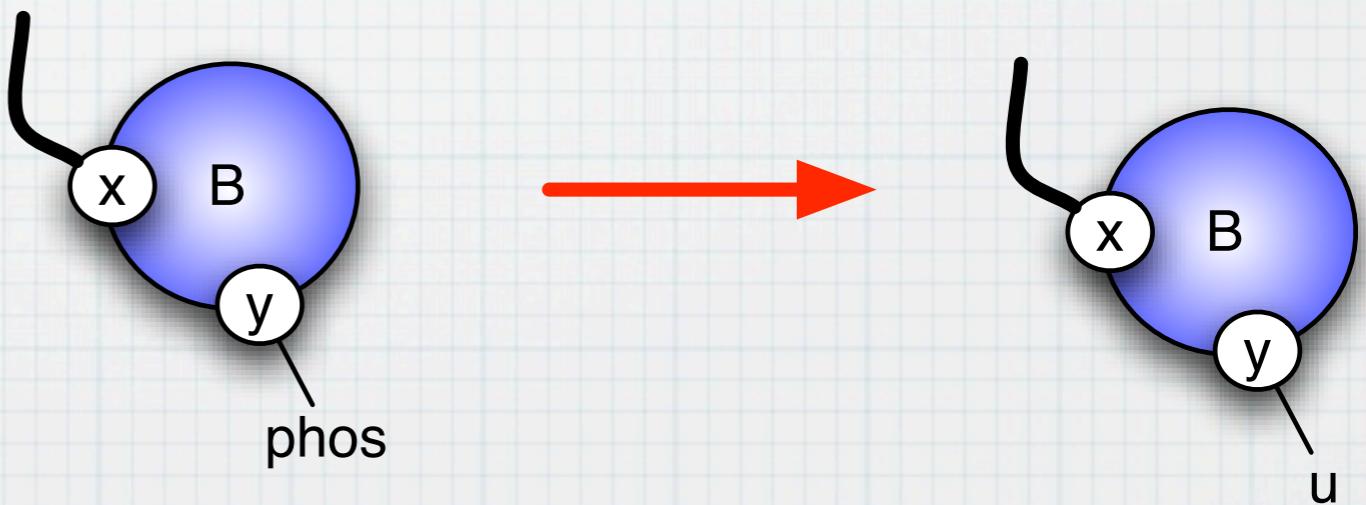
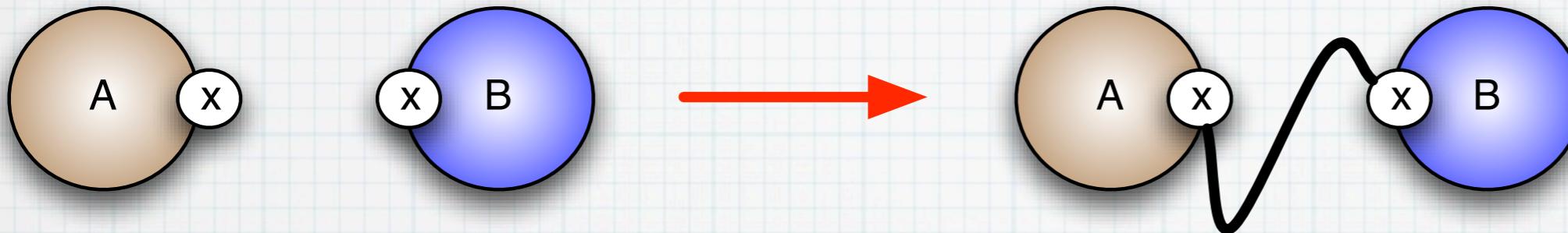
What is the potential impact (activation) of the application of a given rule on the activity of other rules?

Wake-up map



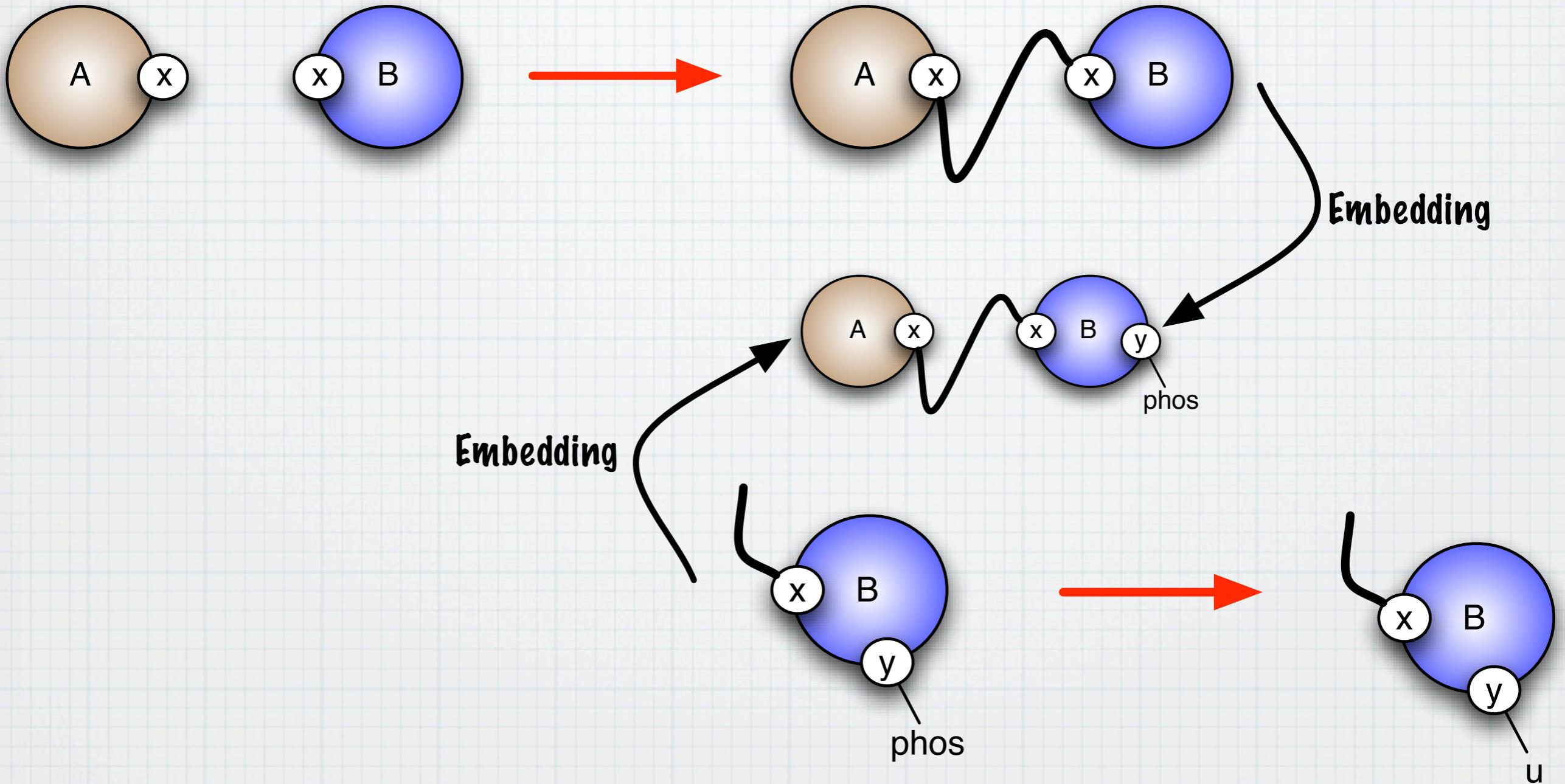
What is the potential impact (activation) of the application of a given rule on the activity of other rules?

Activation



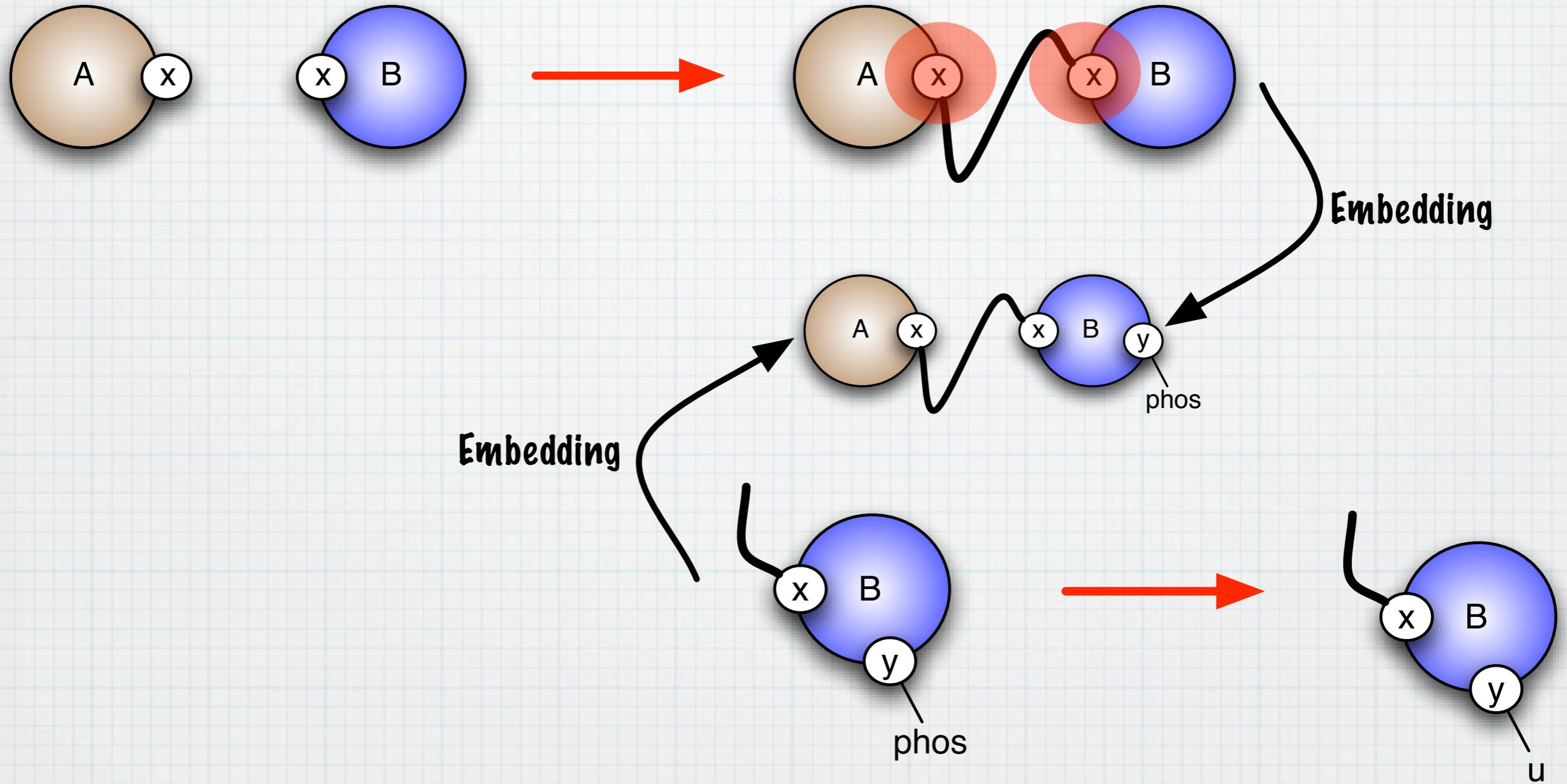
Can the upper rule increase the activity of the second one?

Activation



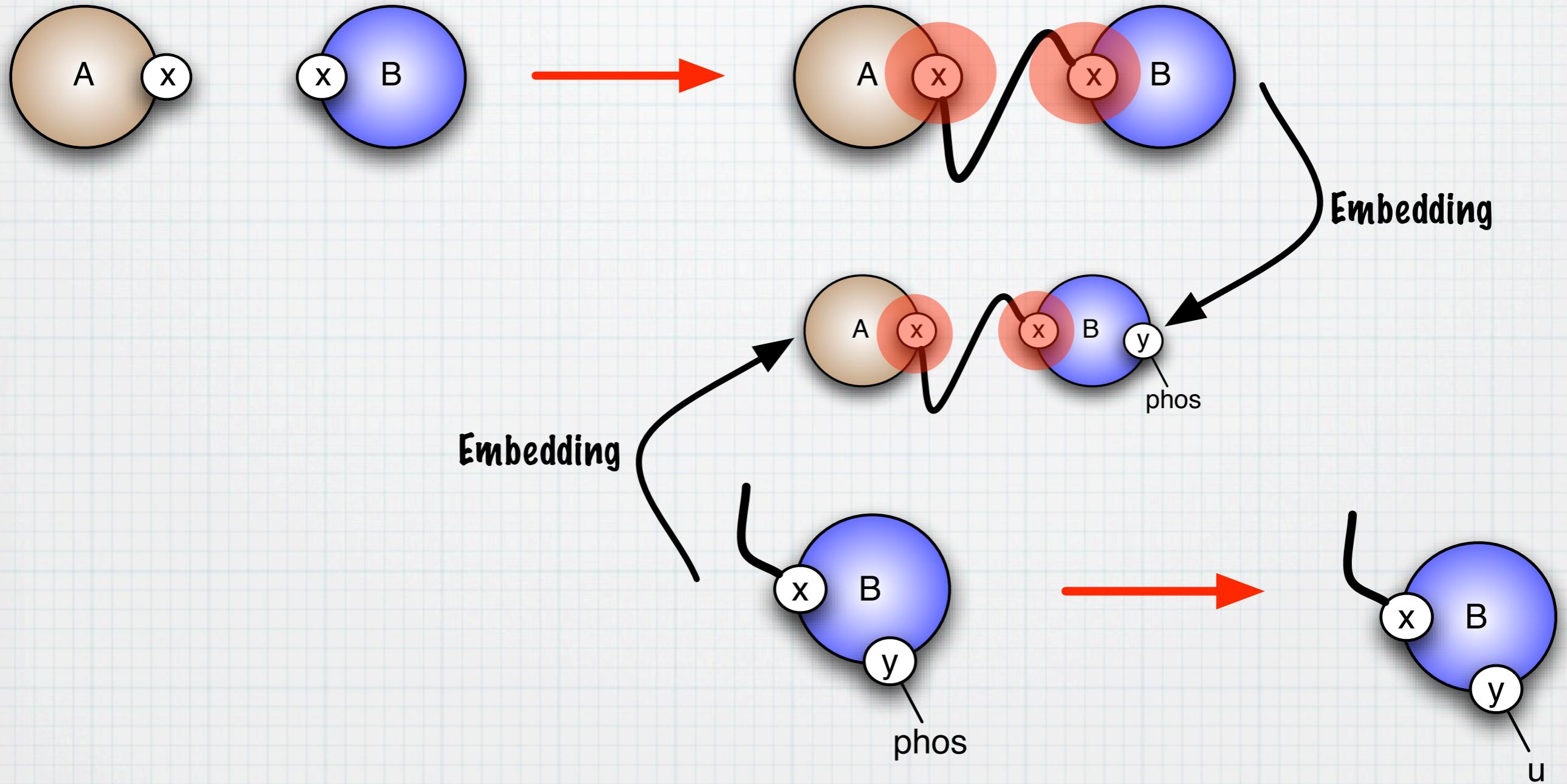
Can the upper rule increase the activity of the second one?

Activation



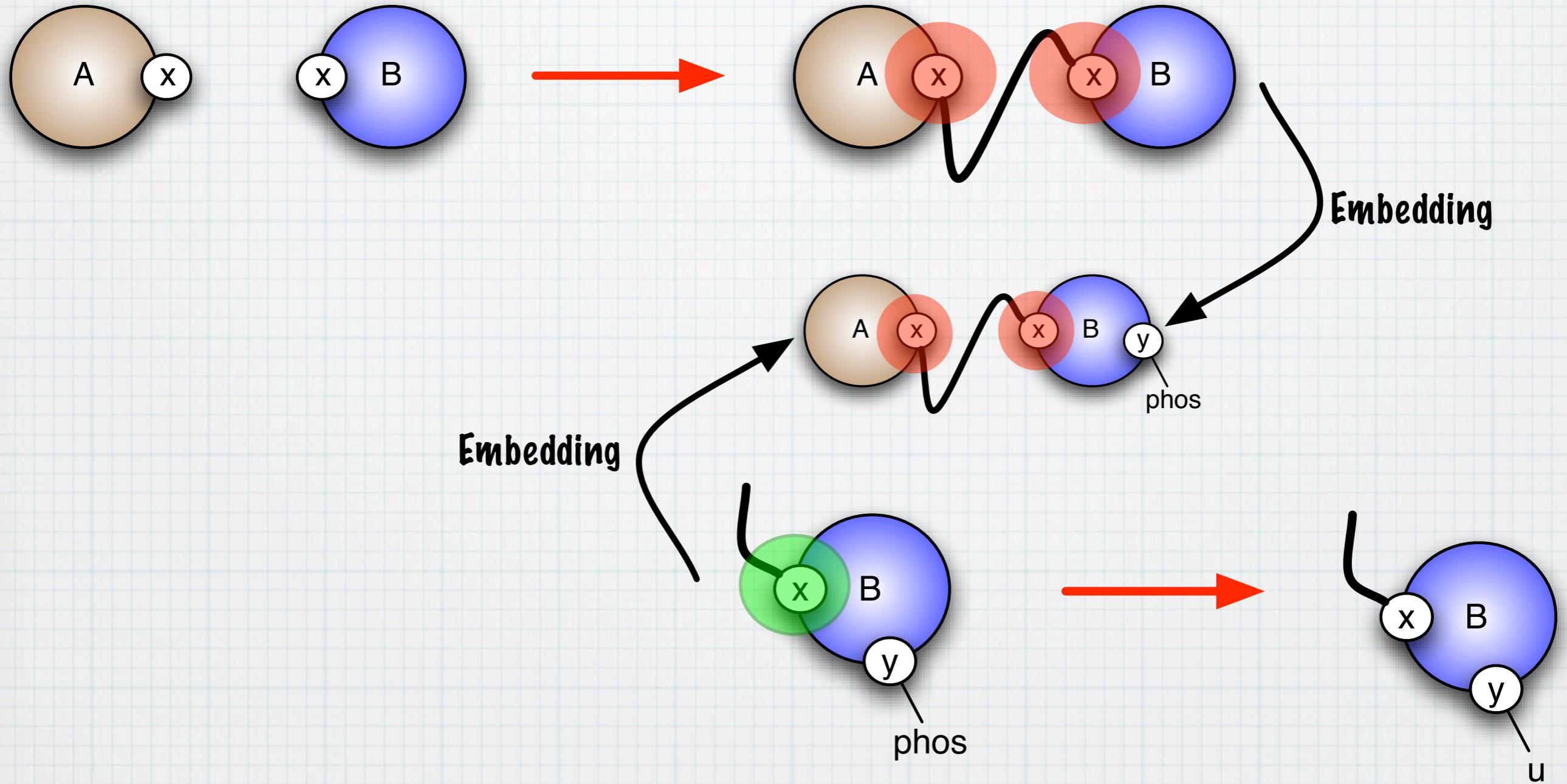
Can the upper rule increase the activity of the second one?

Activation



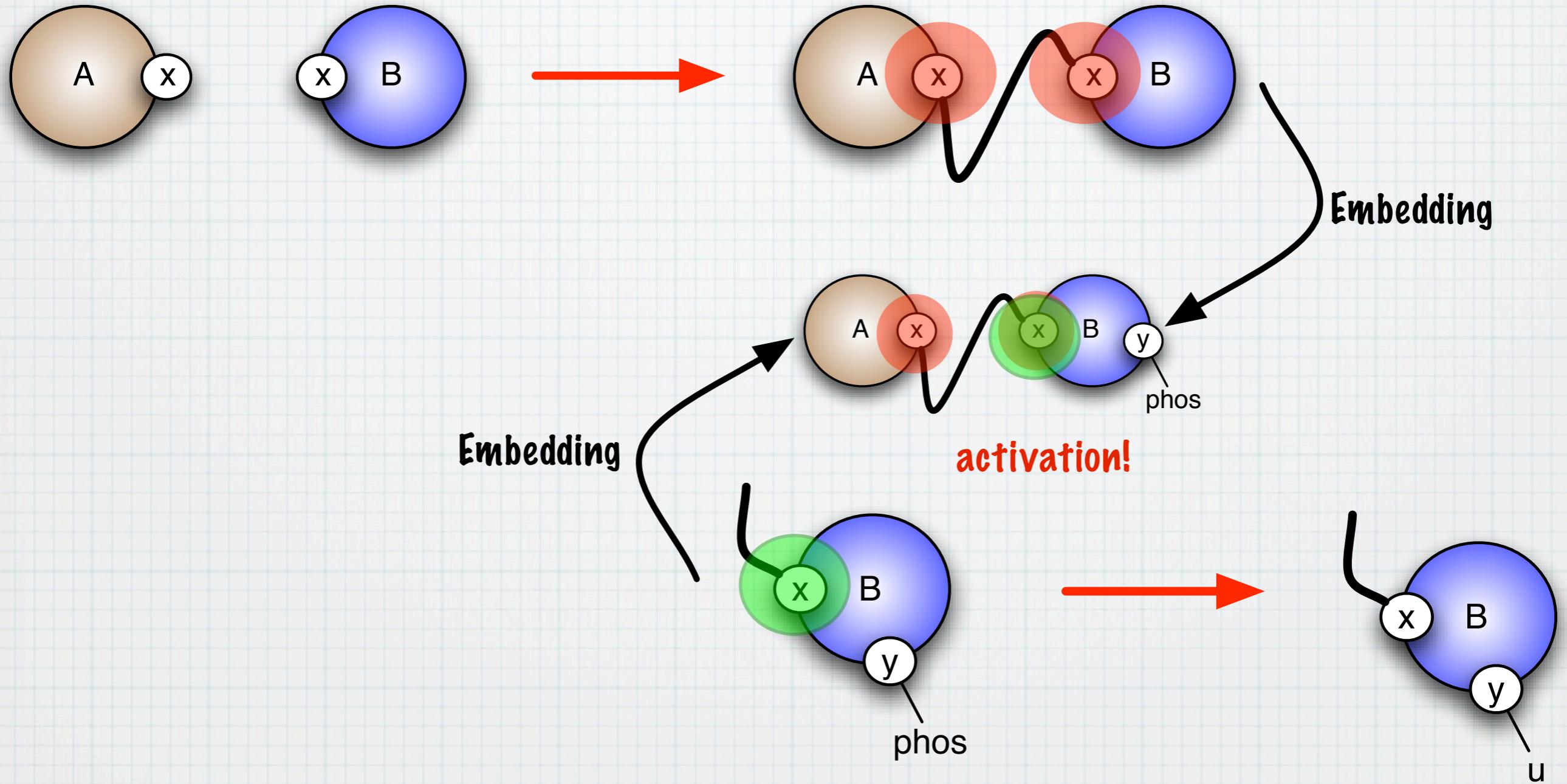
Can the upper rule increase the activity of the second one?

Activation

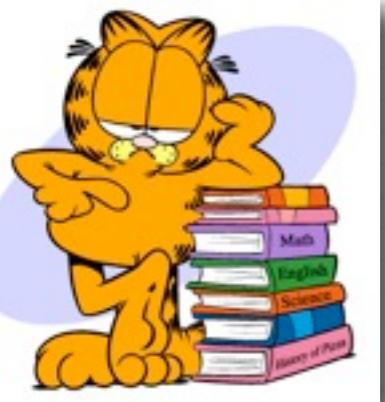


Can the upper rule increase the activity of the second one?

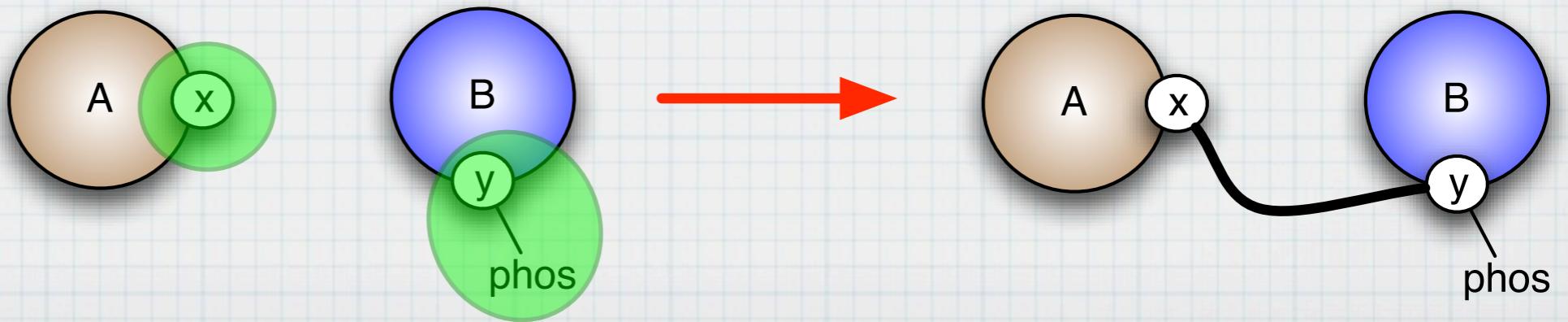
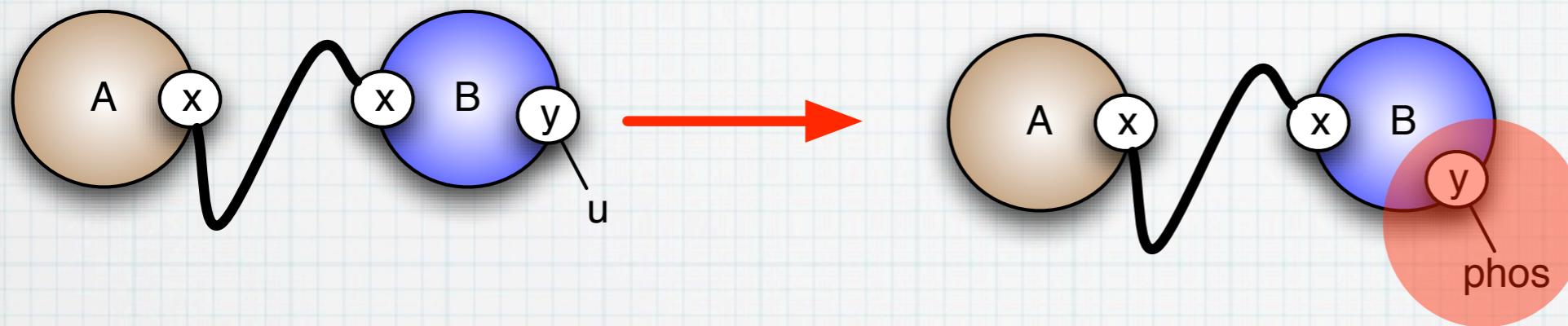
Activation

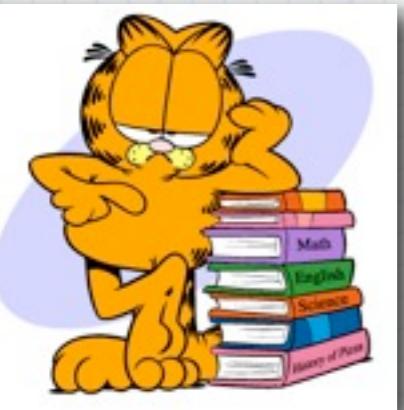


Can the upper rule increase the activity of the second one?

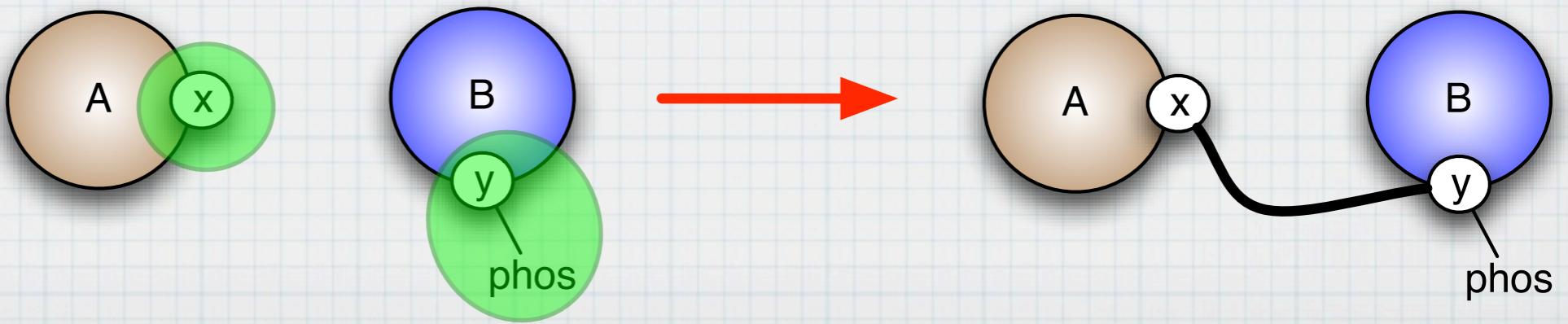
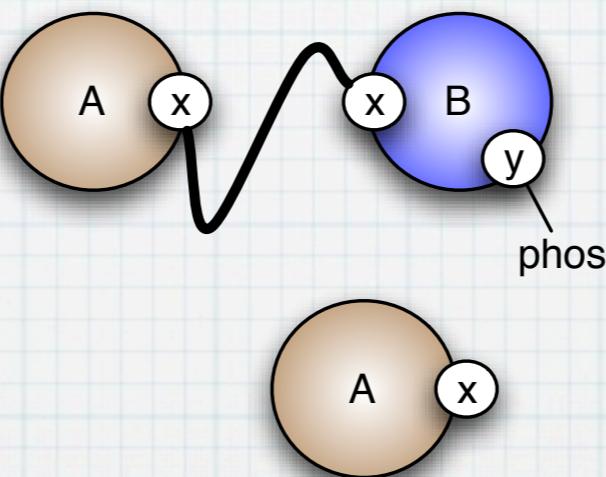
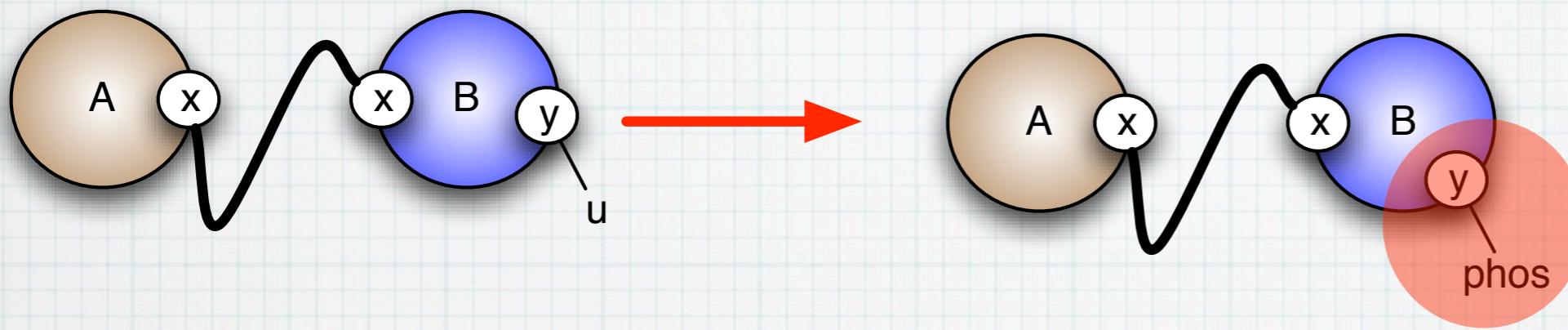


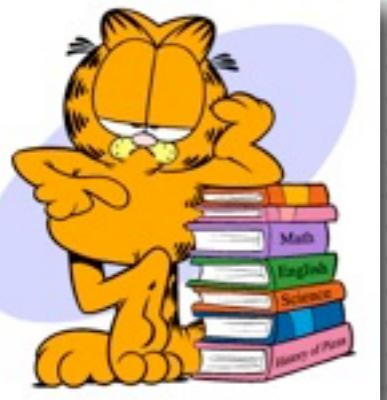
Quizz



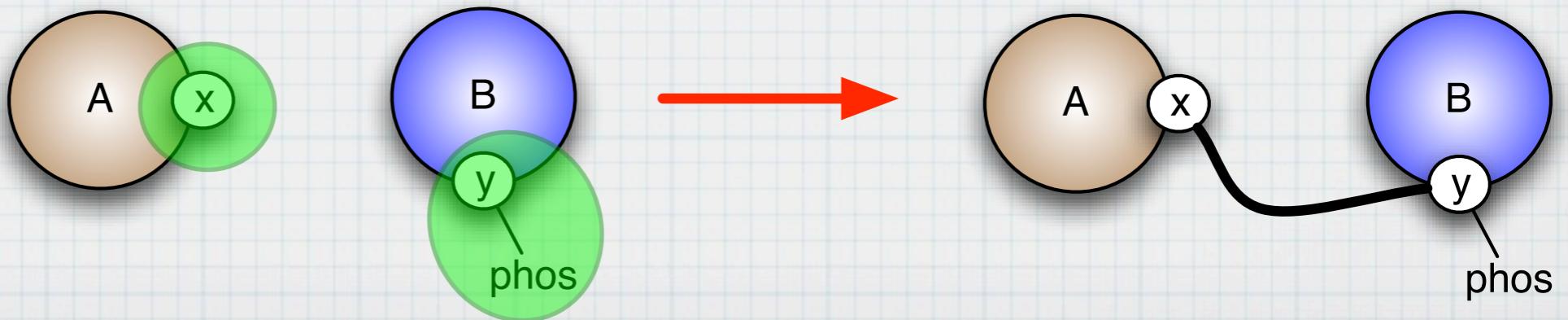
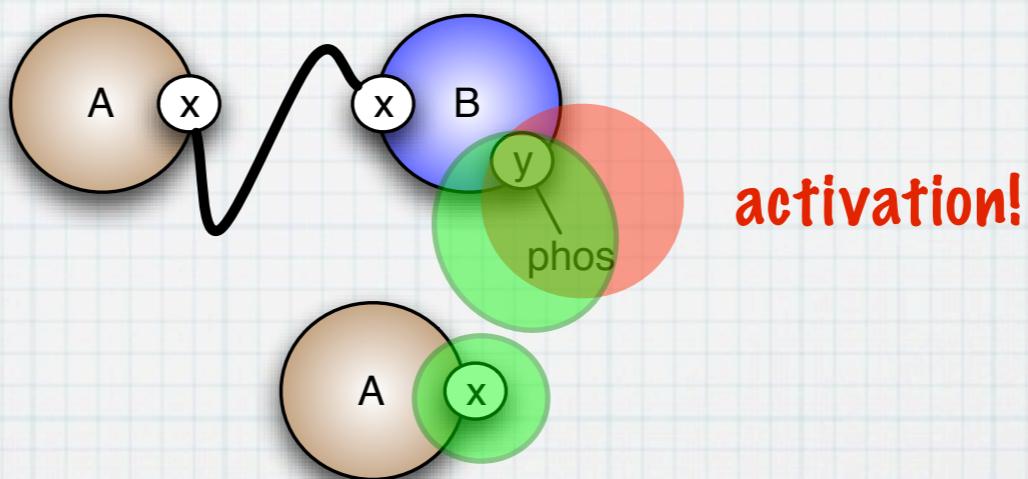
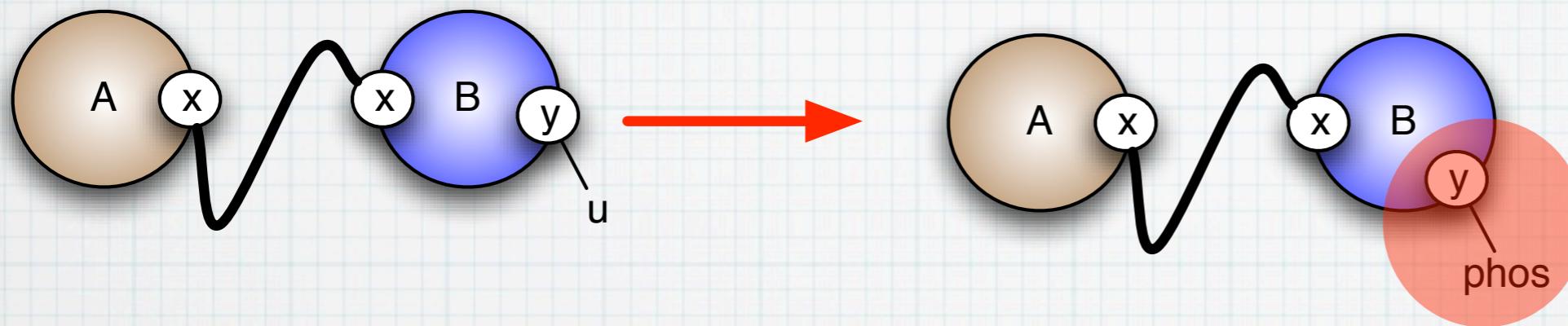


Quizz

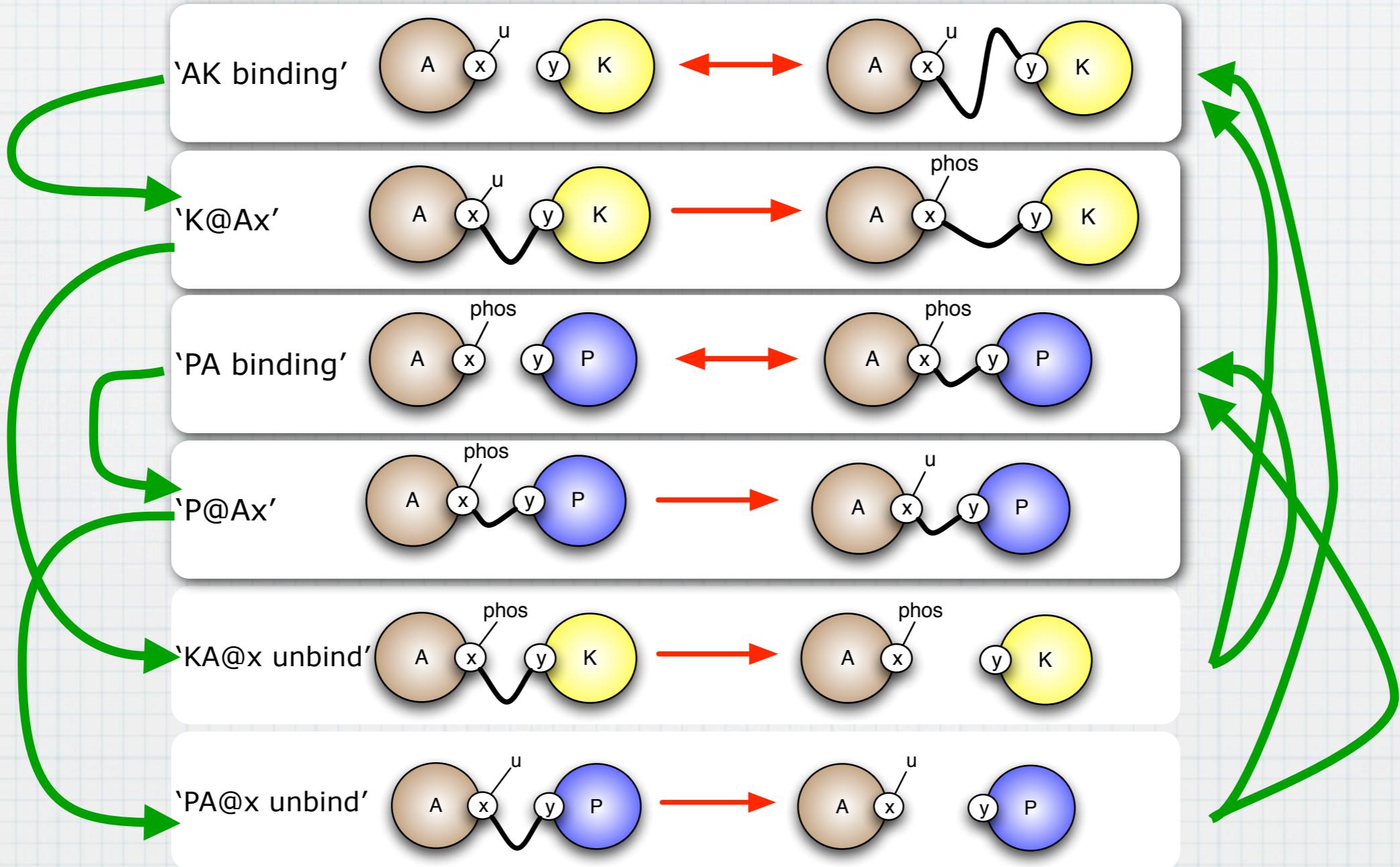




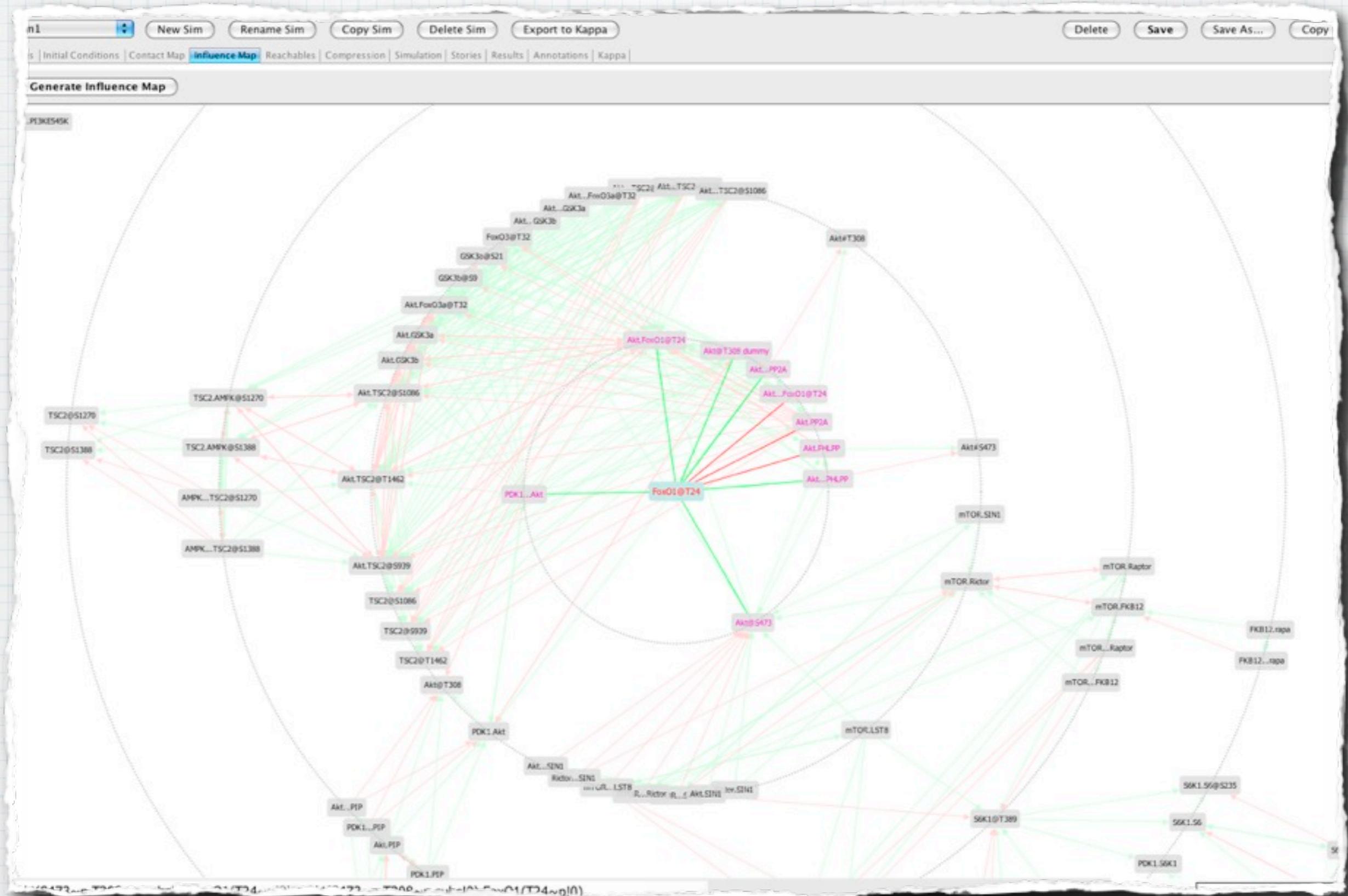
Quizz



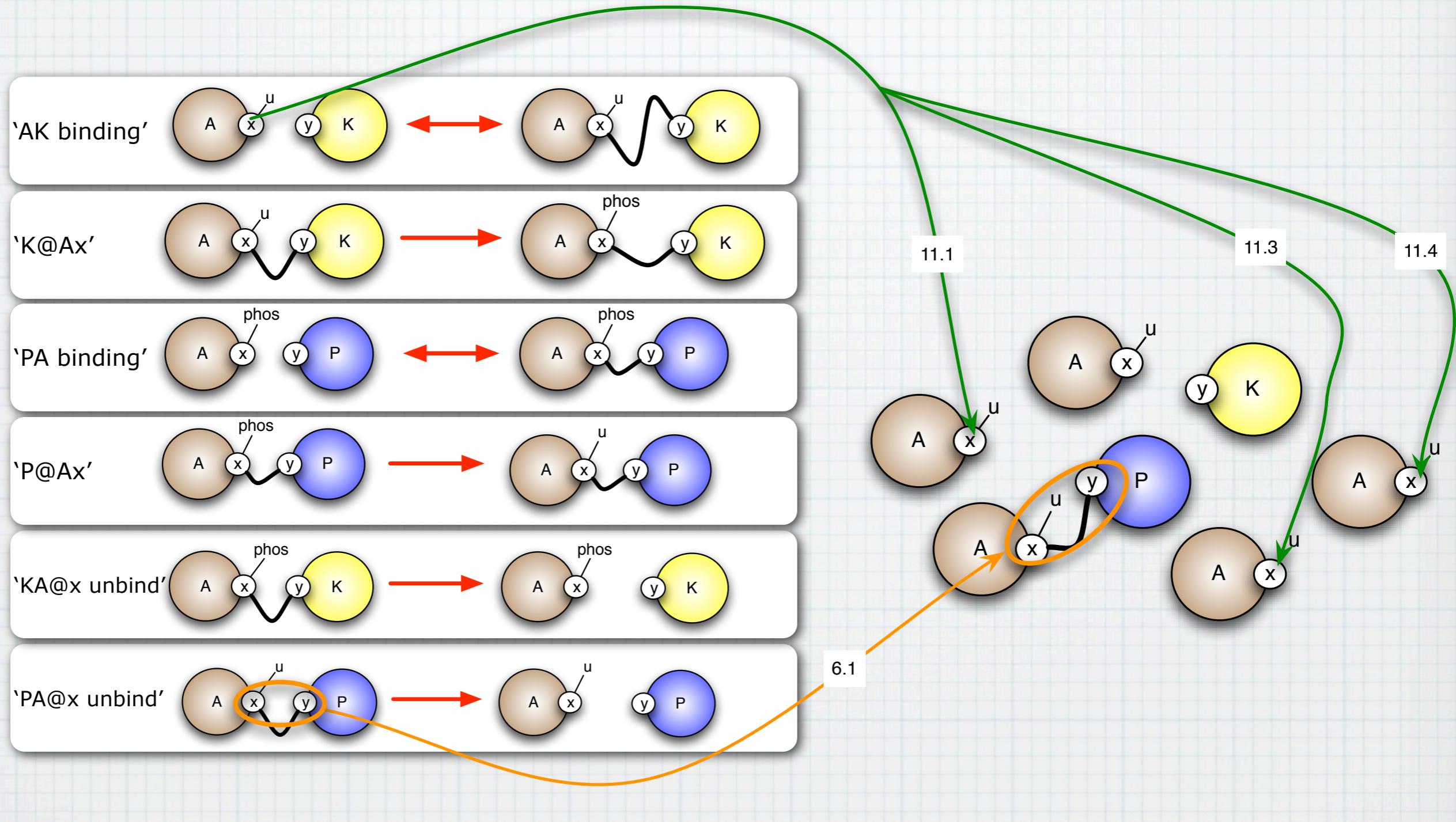
Influence map



Influence map

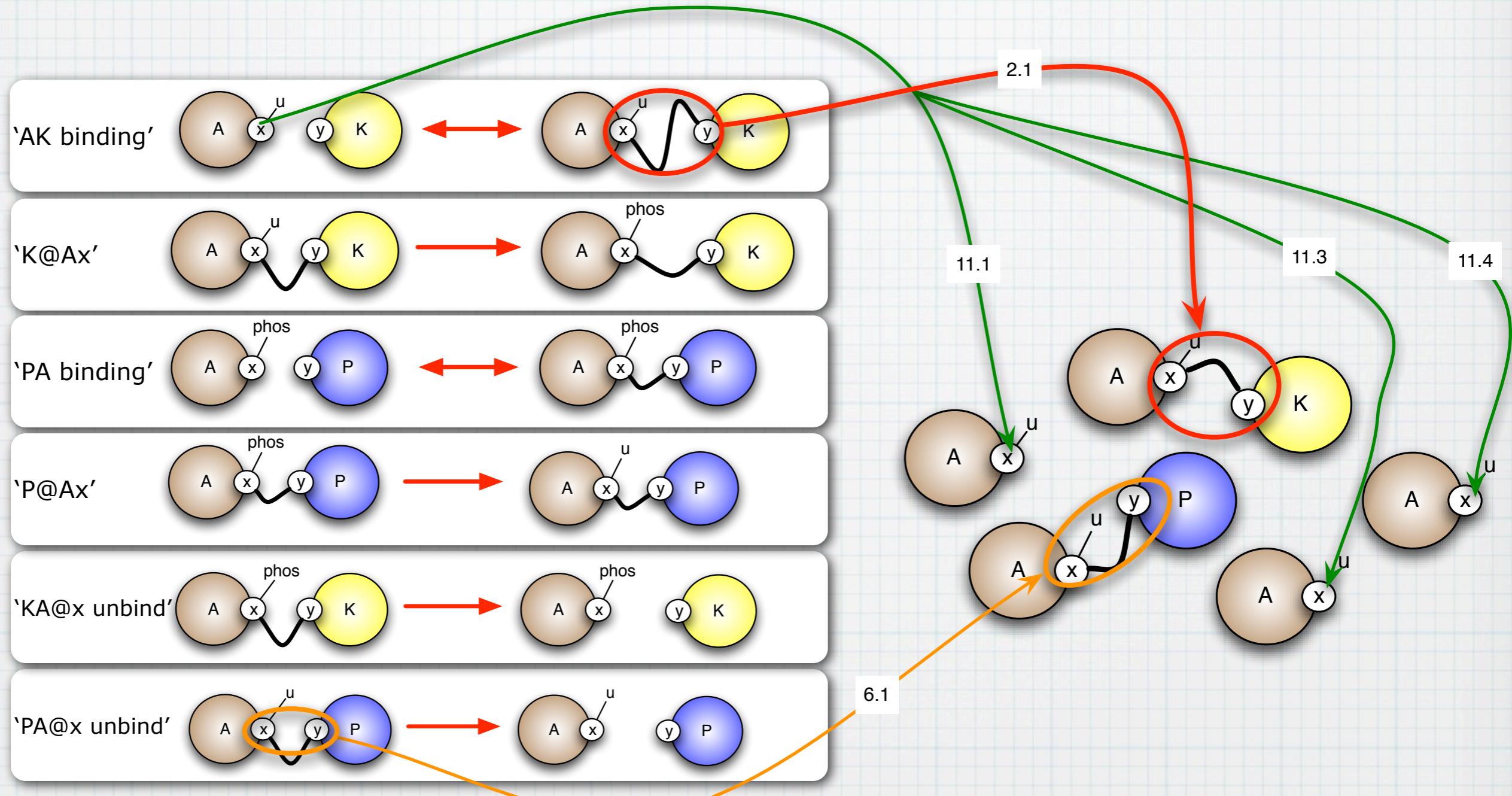


Back to simulation



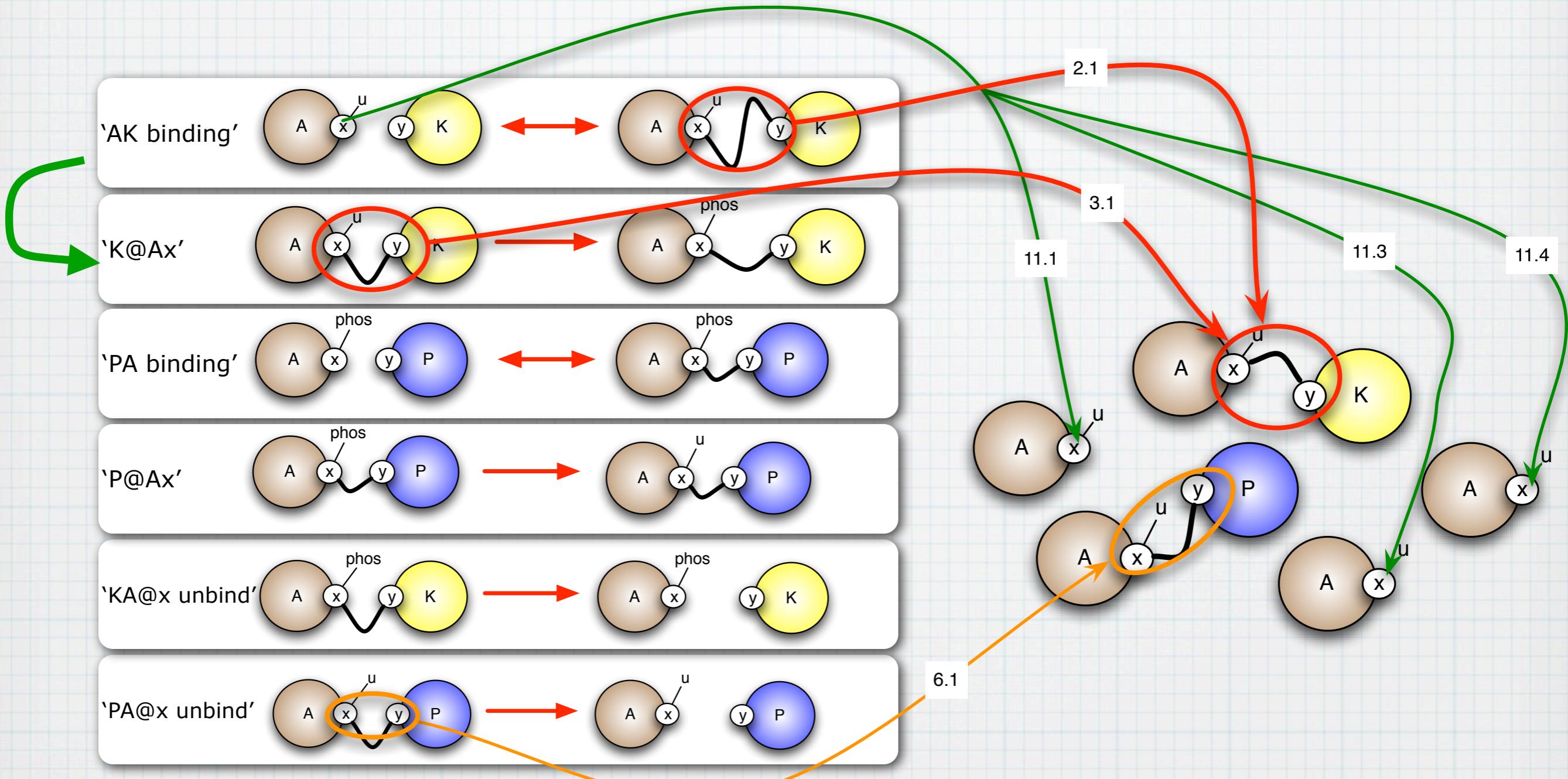
Negative update is purely local: just erase injections using the modified sites...

Back to simulation



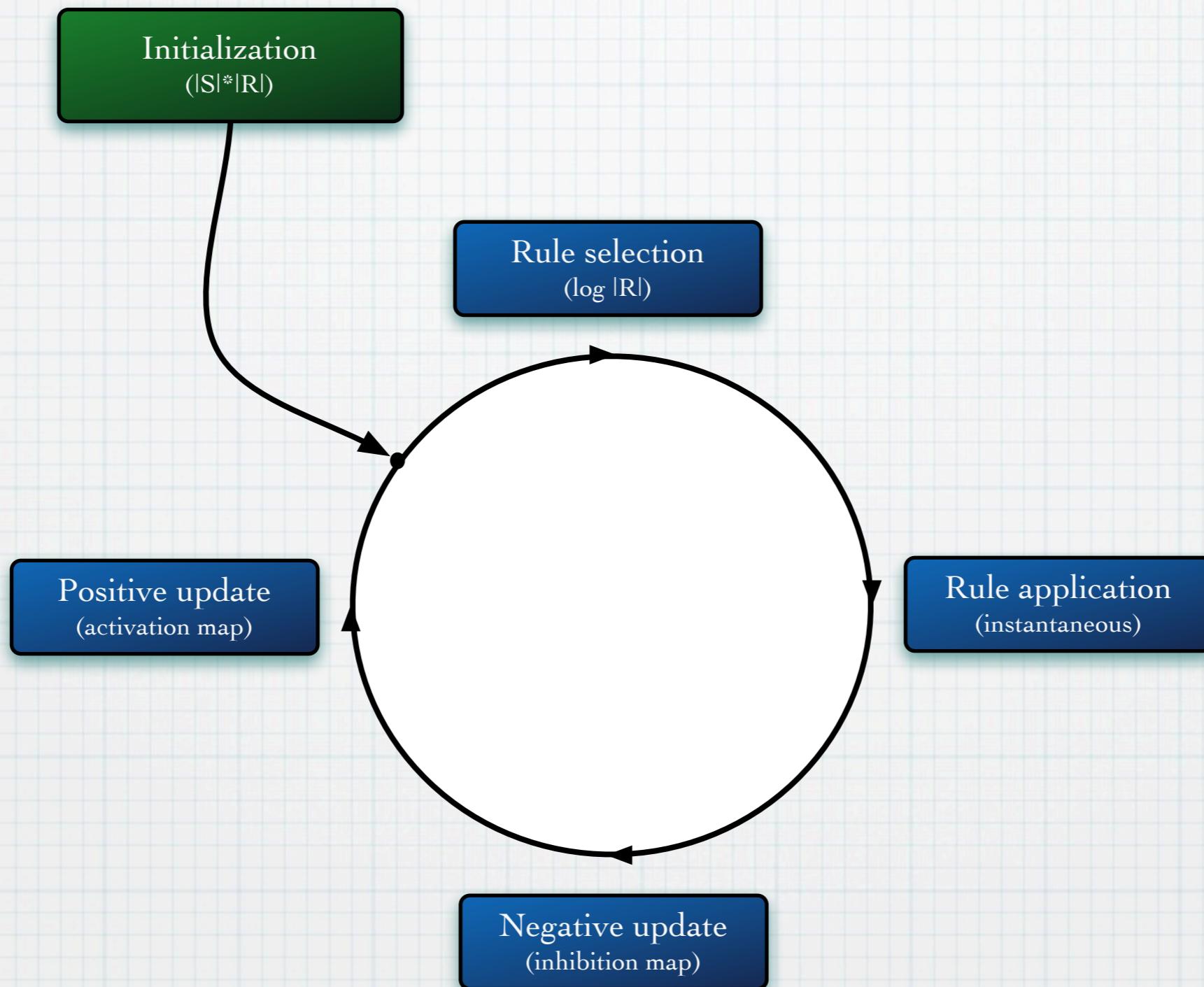
Negative update is purely local: just erase injections using the modified sites...

Back to simulation



Positive update: one needs only to try matching activated rules!

Event loop



Event loop

