CMSB 2018

KaSa: a Static Analyzer for Kappa

Jerome Feret

DI - ÉNS

http://www.di.ens.fr/~feret

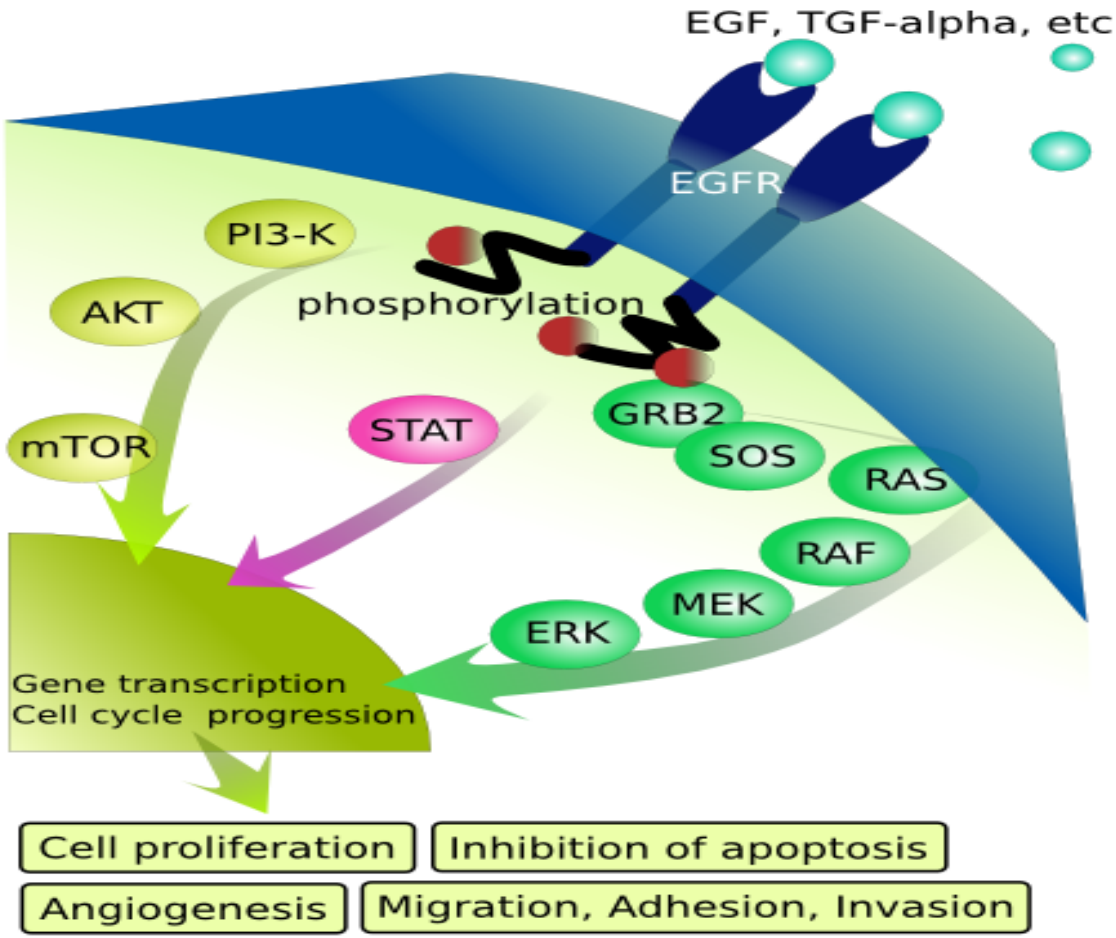
Joint work with
Pierre Boutillier, Ferdinanda Camporesi, Jean Cocquet,
Kim Quyên Lý, Nathalie Theret, and Pierre Vignet

Brno, 2018 September 14th

On the menu today

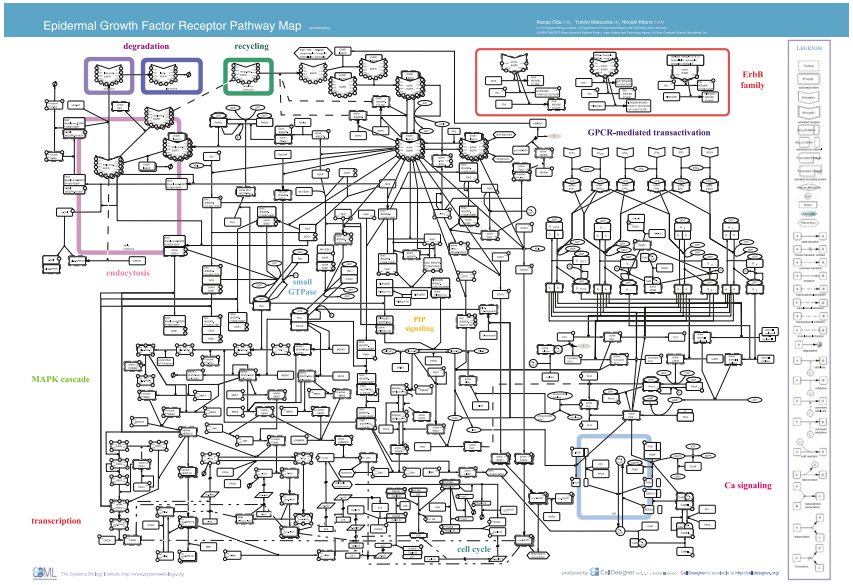
1. Context and motivation
2. Kappa language
3. Main functionalities
4. Conclusion

Signalling pathways



Eikuch, 2007

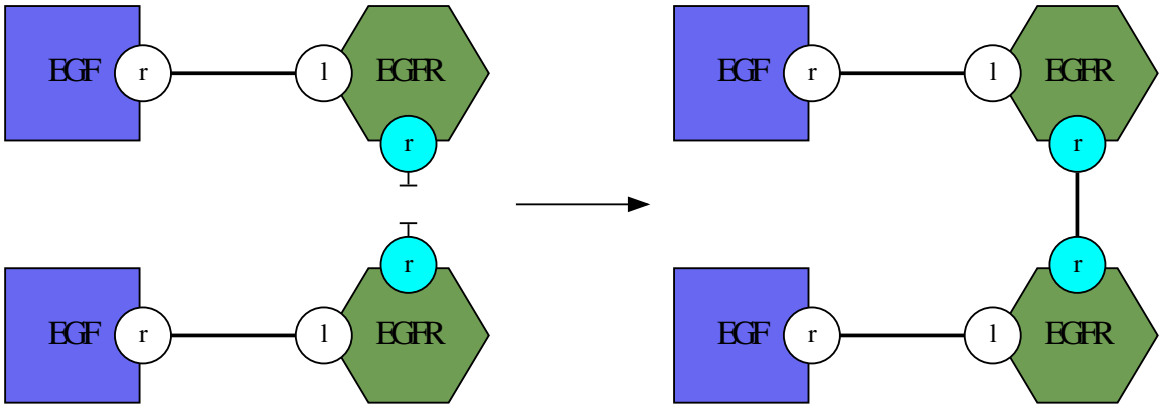
Bridge the gap between...



$$
\begin{cases}
\frac{dx_1}{dt} = -k_1 \cdot x_1 \cdot x_2 + k_{-1} \cdot x_3 \\
\frac{dx_2}{dt} = -k_1 \cdot x_1 \cdot x_2 + k_{-1} \cdot x_3 \\
\frac{dx_3}{dt} = k_1 \cdot x_1 \cdot x_2 - k_{-1} \cdot x_3 + 2 \cdot k_2 \cdot x_3 \cdot x_3 - k_{-2} \cdot x_4 \\
\frac{dx_4}{dt} = k_2 \cdot x_3^2 - k_2 \cdot x_4 + \frac{v_4 \cdot x_5}{p_4 + x_5} - k_3 \cdot x_4 - k_{-3} \cdot x_5 \\
\frac{dx_5}{dt} = \cdots \\
\quad \vdots \\
\frac{dx_n}{dt} = -k_1 \cdot x_1 \cdot c_2 + k_{-1} \cdot x_3
\end{cases}
$$

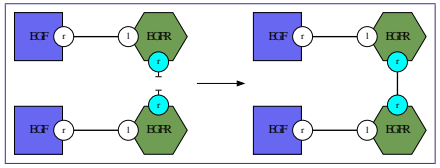
knowledge representation **and** models of the behaviour of systems

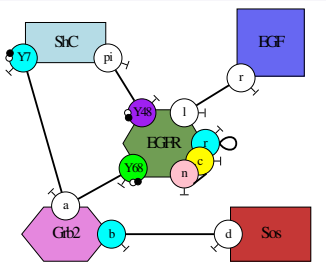
Site-graphs rewriting



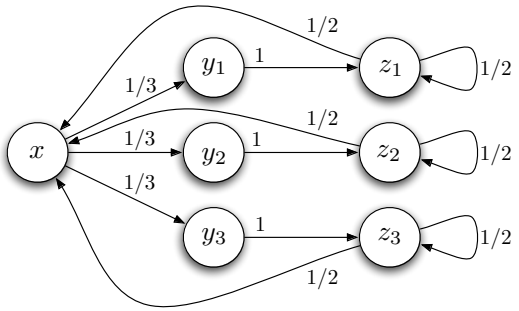
- a language close to knowledge representation;
- rules are easy to update;
- a compact description of models.

Choices of semantics





interaction map



Markov chain

$$\begin{cases} \frac{dx_1}{dt} = -k_1 \cdot x_1 \cdot x_2 + k_{-1} \cdot x_3 \\ \frac{dx_2}{dt} = -k_1 \cdot x_1 \cdot x_2 + k_{-1} \cdot x_3 \\ \frac{dx_3}{dt} = k_1 \cdot x_1 \cdot x_2 - k_{-1} \cdot x_3 + 2 \cdot k_2 \cdot x_3 \cdot x_3 - k_{-2} \cdot x_4 \\ \frac{dx_4}{dt} = k_2 \cdot x_3^2 - k_2 \cdot x_4 + \frac{v_4 \cdot x_5}{p_4 + x_5} - k_3 \cdot x_4 - k_{-3} \cdot x_5 \\ \frac{dx_5}{dt} = \cdots \\ \quad \vdots \\ \frac{dx_n}{dt} = -k_1 \cdot x_1 \cdot c_2 + k_{-1} \cdot x_3 \end{cases}$$

ordinary differential equations

Static analysis

Kappa:

1. solves the problem of knowledge representation;
2. provides formal definition for the collective behaviour of proteins.

But how to ensure that Kappa rules match with what the modelers have in mind?

We use static analysis:

1. to increase the level of confidence.
2. to get a quick snapshop of what is going on in a model.
3. *to initiate a virtuous loop between model refinement and static analysis.*

KaSa

- Goal: Provide accurate static analysis, fast enough to be integrated in the graphical user interface.
- Theory: Abstract interpretation, category theory, group actions, linear algebra simplicial complexes, graphs.
- Algorithms and Data-structures: binary decision diagrams, hash-consing, memoization, patrician trees, row echelon forms, Tarjan's SCC algorithm.
- Authors: Jérôme Feret (2006-) and Kim Quyên Lý (2015-2017)
- 68,000 lines of OCaml

"AbstractCell" (2009-2013)

"Big Mechanism" (2014-2017)

"TGFβSysBio" (2015-2018)

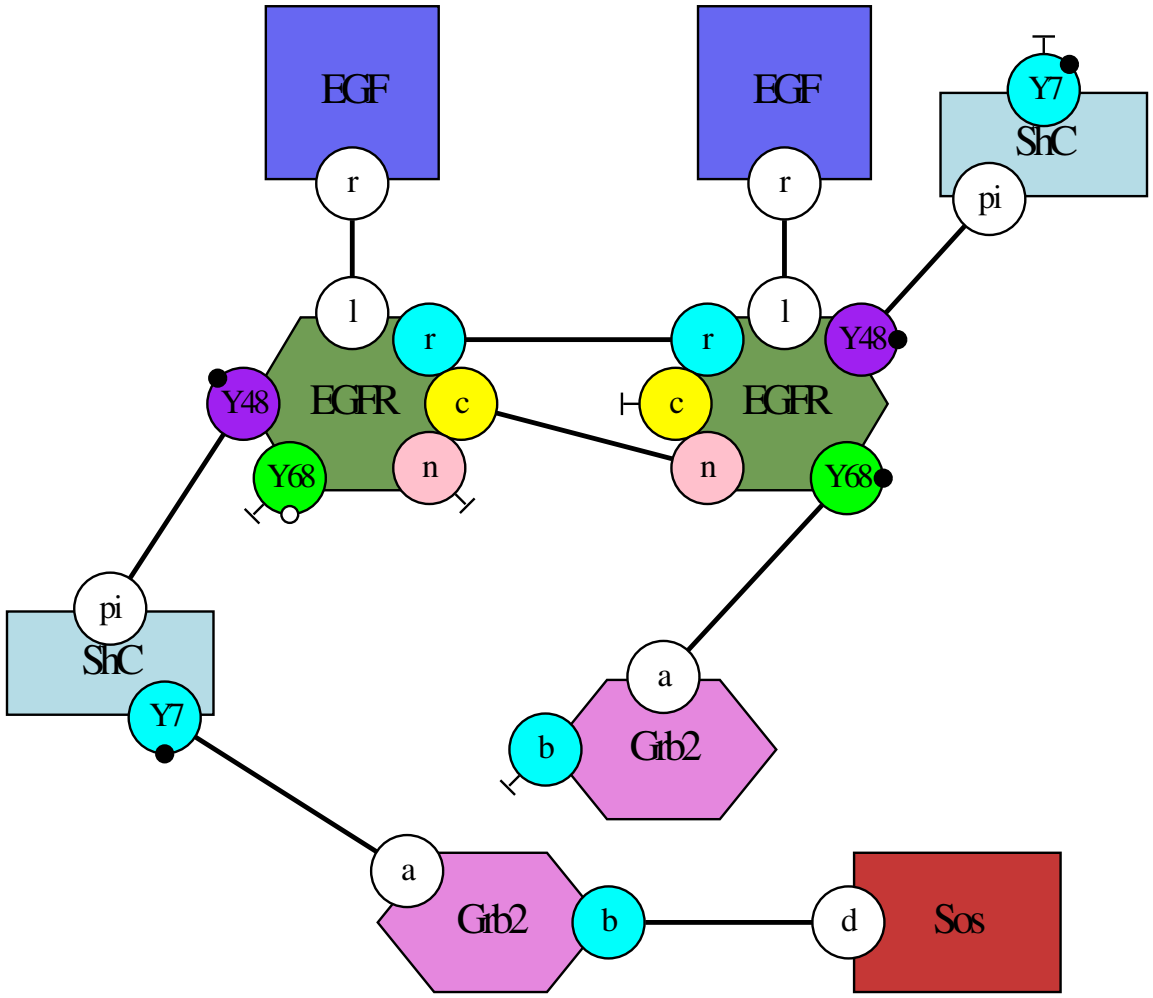
Case study: TGF-b

- Model of the extracellular activation of the transforming growth factor TGF-b:
  - TGF-b controls cell homeostasis in normal tissue,
  - TGF-b promotes the development of fibrosis and cancer.
- Developped by Nathalie Theret, Jean Cocquet, and Pierre Vignet.
- About 300 interaction rules.
- Point of interest:
  - Competition for shared resources;
  - Very long time-scale;
  - Polymers formation with counterfactual causality;
- Nice interplay:
  - KaSa has been helpful to curate the model;
  - KaSa has been extended to cope with new properties of interest identified during the modelling process.

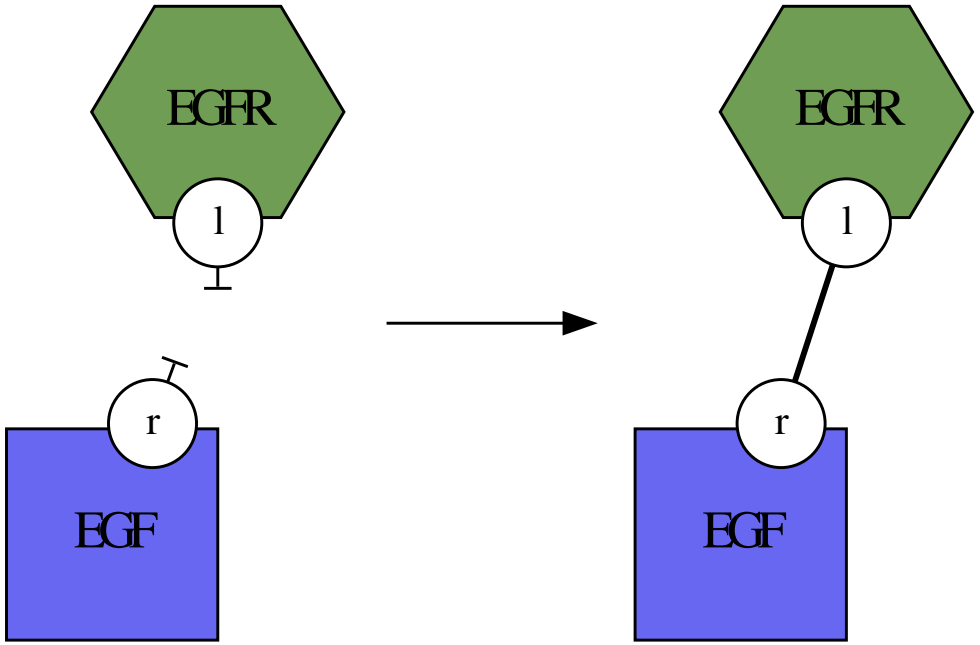
On the menu today

1. Context and motivation
2. Kappa language
3. Main functionalities
4. Conclusion

Bio-molecular compound

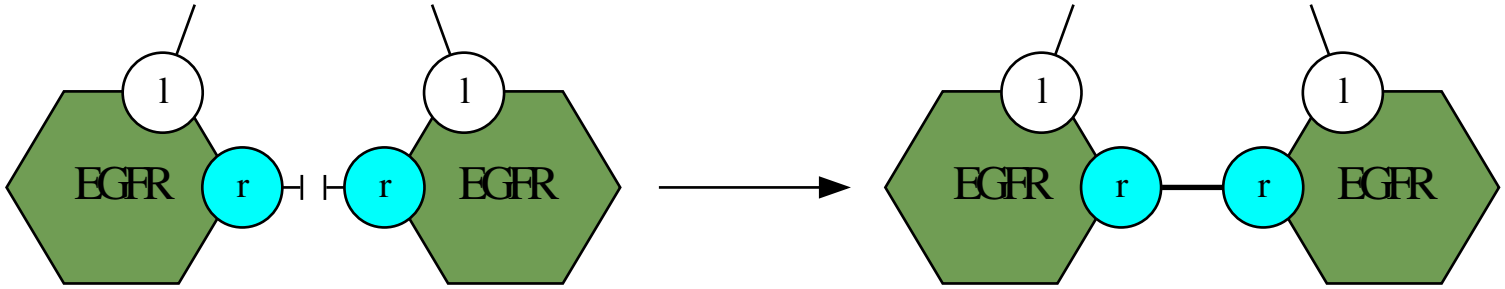


Receptor activation

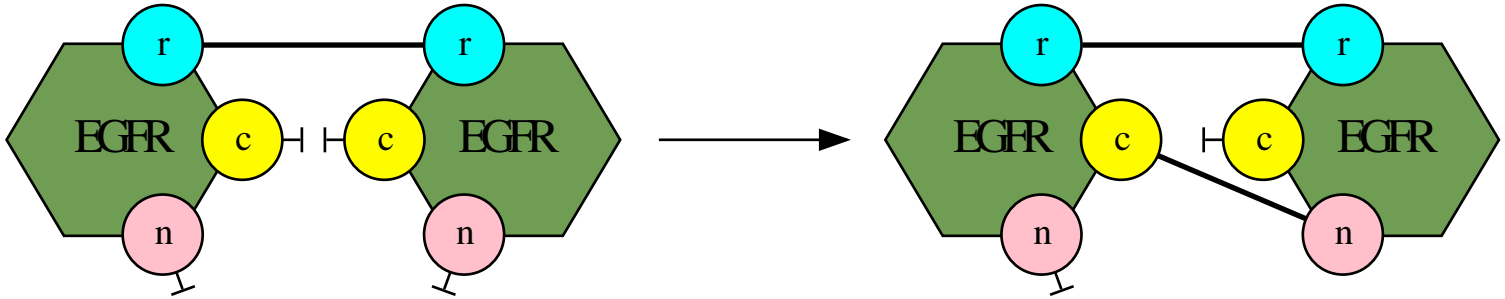


$$\texttt{EGF(r[.])}\,,\,\texttt{EGFR(l[.])} \longrightarrow \texttt{EGF(r[1])}\,,\,\texttt{EGFR(l[1])}$$

Asymmetric dimerisation

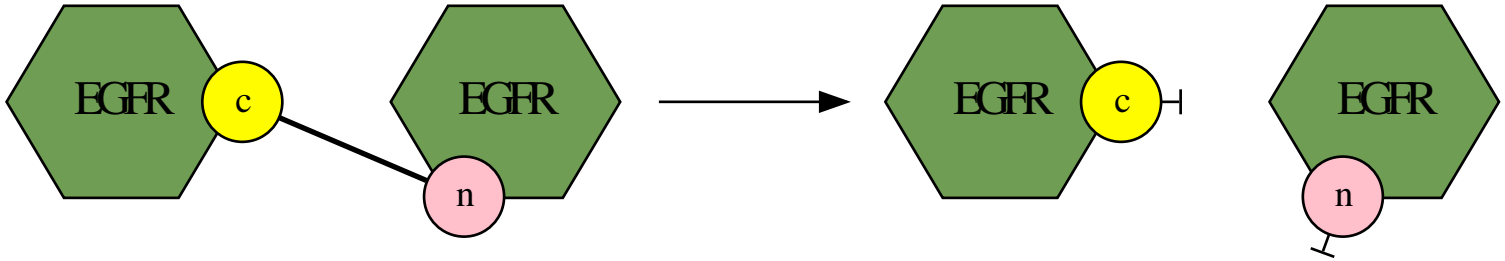


$\texttt{EGFR(l[\_],r[.])}\ ,\ \texttt{EGFR(l[\_],r[.])} \longrightarrow \texttt{EGFR(l[\_],r[1])}\ ,\ \texttt{EGFR(l[\_],r[1])}$

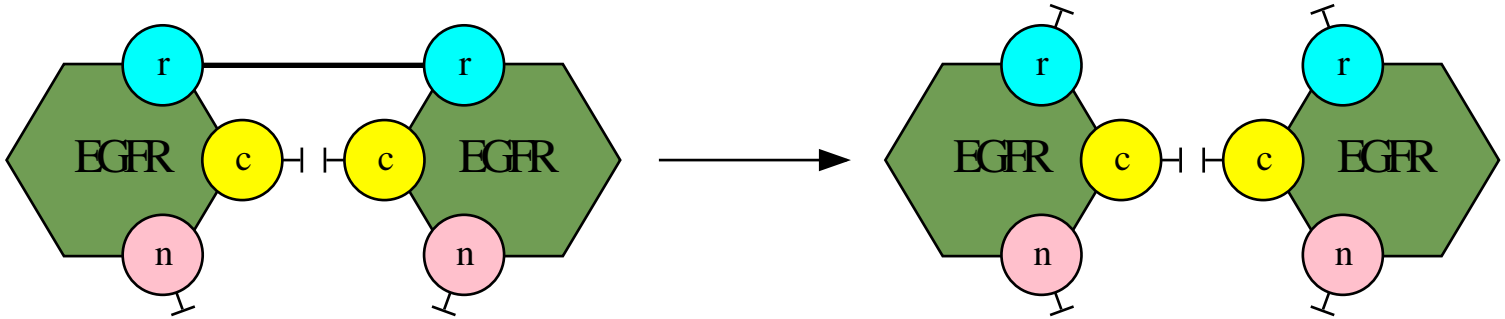


$\texttt{EGFR(r[1],c[.],n[.])}\ ,\ \texttt{EGFR(r[1],c[.],n[.])} \longrightarrow \texttt{EGFR(r[1],c[2],n[.])}\ ,\ \texttt{EGFR(r[1],c[.],n[2])}$

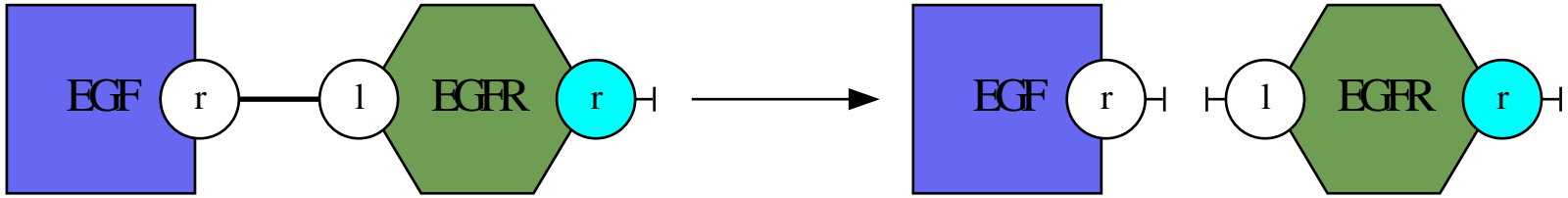
Sequential unbinding



$$\texttt{EGFR(c[1])}\,,\,\texttt{EGFR(n[1])} \longrightarrow \texttt{EGFR(c[.])}\,,\,\texttt{EGFR(n[.])}$$



$$\texttt{EGFR(r[1],c[.],n[.])}\,,\,\texttt{EGFR(r[1],c[.],n[.])} \longrightarrow \texttt{EGFR(r[.],c[.],n[.])}\,,\,\texttt{EGFR(r[.],c[.],n[.])}$$



$$\texttt{EGF(r[1])}\,,\,\texttt{EGFR(l[1],r[.])} \longrightarrow \texttt{EGF(r[.])}\,,\,\texttt{EGFR(l[.],r[.])}$$

On the menu today

1. Context and motivation
2. Kappa language
3. Main functionalities
4. Conclusion

Main functionnalities

- Invariant inference;
- Dead rule identification;
- Non weakly reversible transition deterction;
- Detection of potentially unbounded polymers.

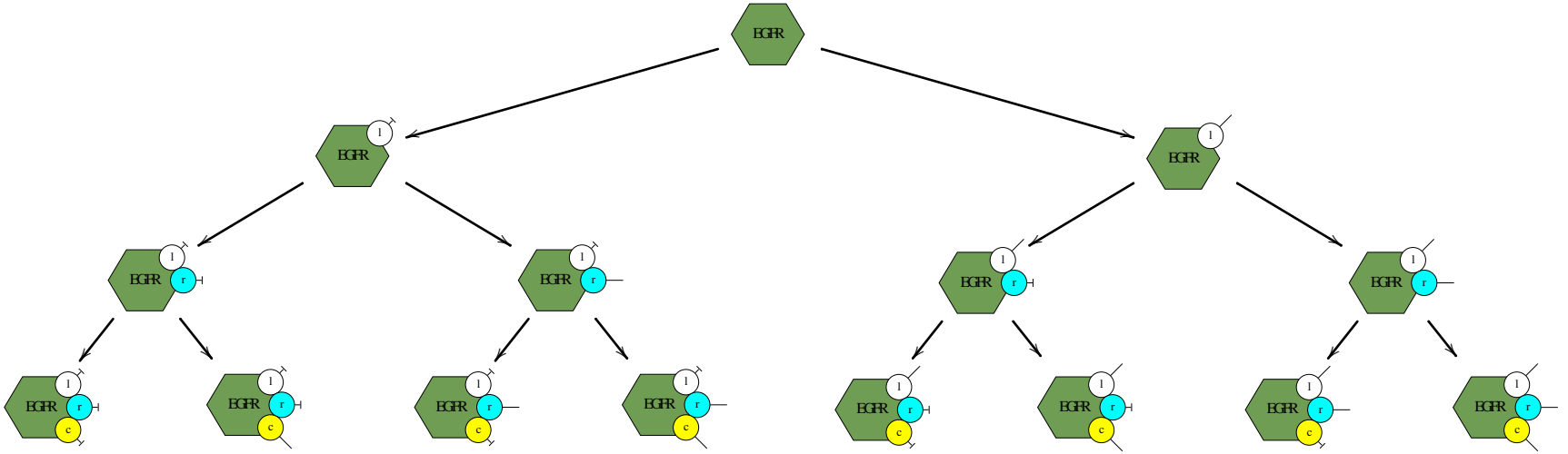
Invariant inference: Motivations

Detect local invariants about bio-molecular compounds:

- Are there relationships between the states of sites?
- Are they transmembrane molecular compounds?
- May a given receptor simultaneously bound to two receptors?

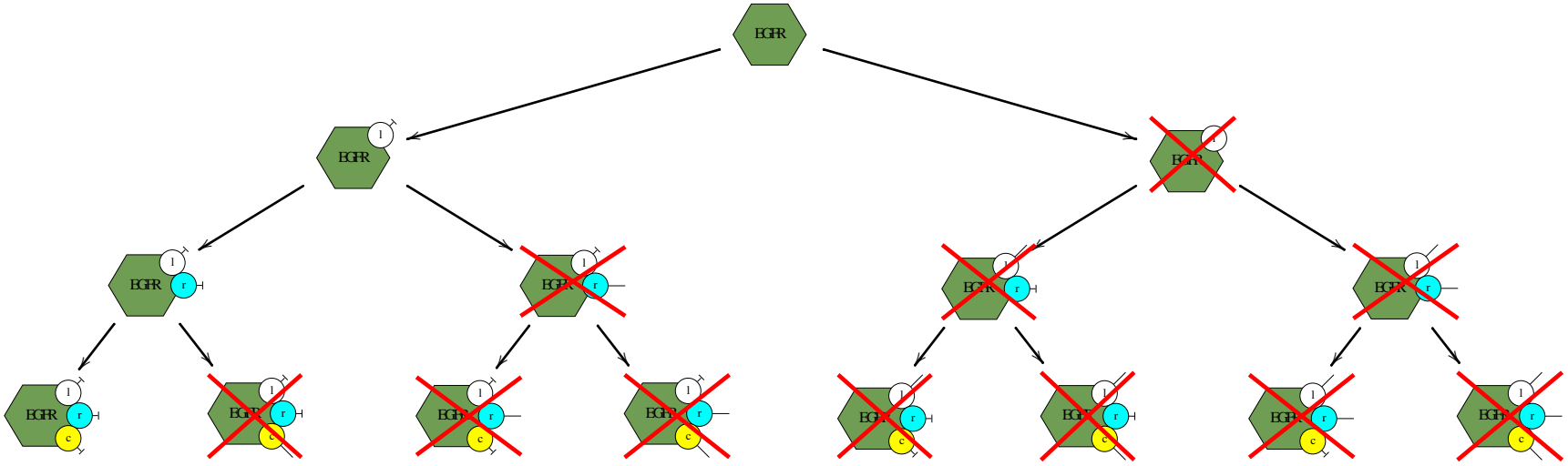
Core properties to be used in every further analyses.

Invariant inference: Orthogonal refinements [SASB'16]



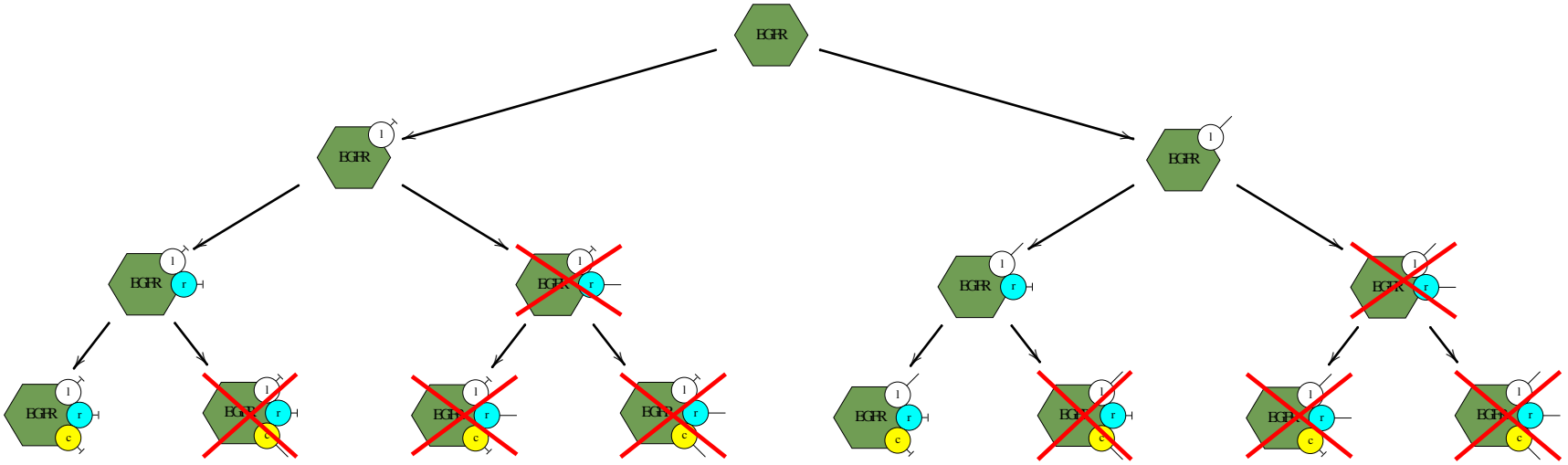
Invariant inference: Initialization

We assume that every pattern not in the initial state does not occur.



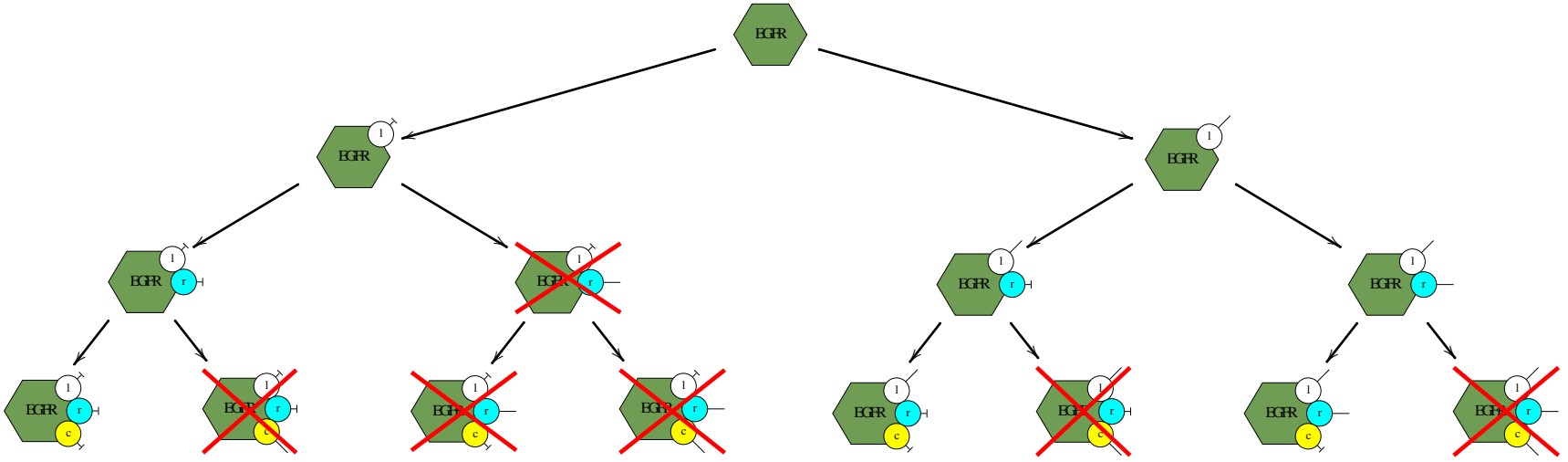
Invariant inference: Induction (step 1)

The ones that can be obtained in one rule application may be reachable.



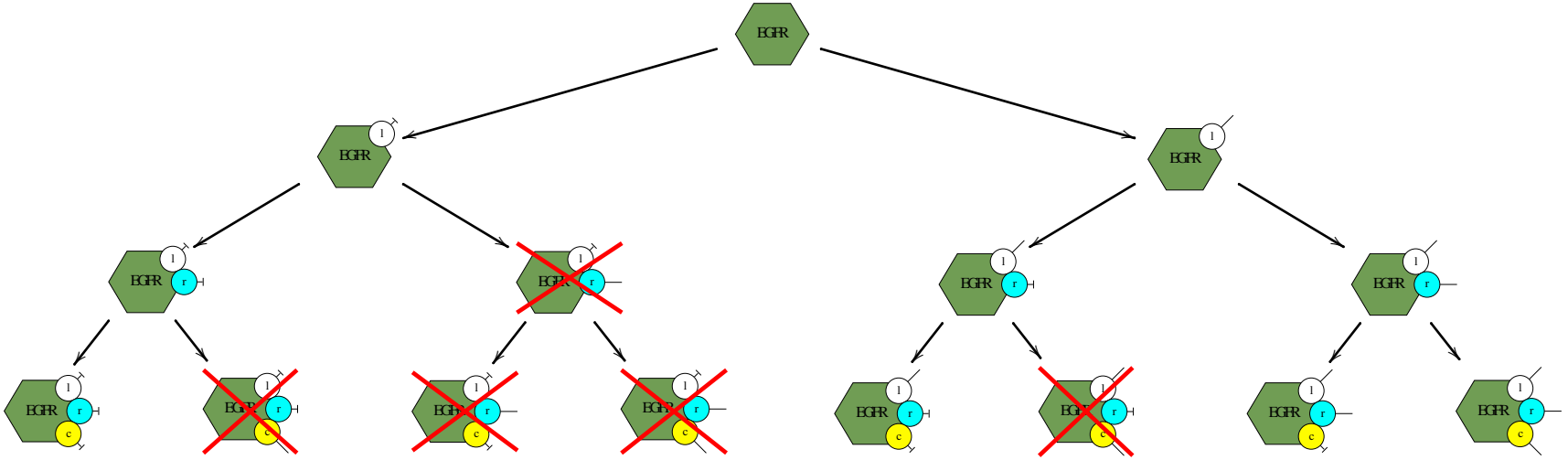
Invariant inference: Induction (step 2)

The ones that can be obtained in one more rule application may be reachable.



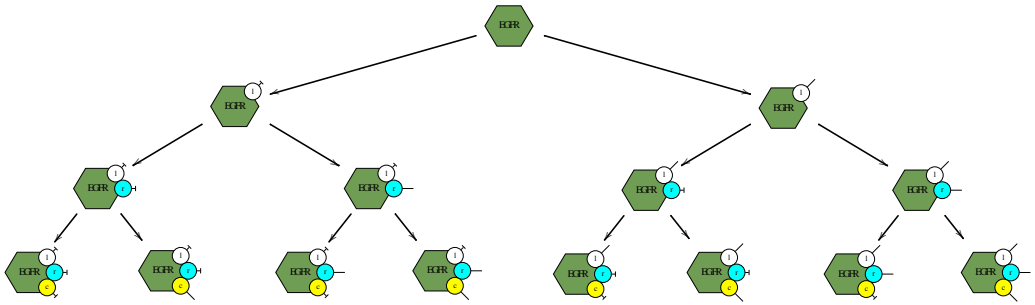
Invariant inference: Fix-point

Until we reach a fix-point.

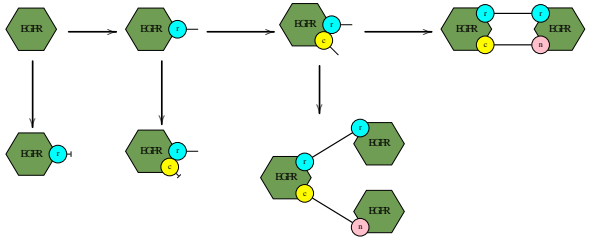


Invariant inference:

1. Accuracy:
   Mutual induction over several orthogonal refinement trees
   (selected automatically by syntactic inspection of the model)
2. Scalability:
   Each tree is:
   - either complete



     (efficiency relies on the use of BDDs)
   - or a comb.



Dead rule/pattern detection
Motivations

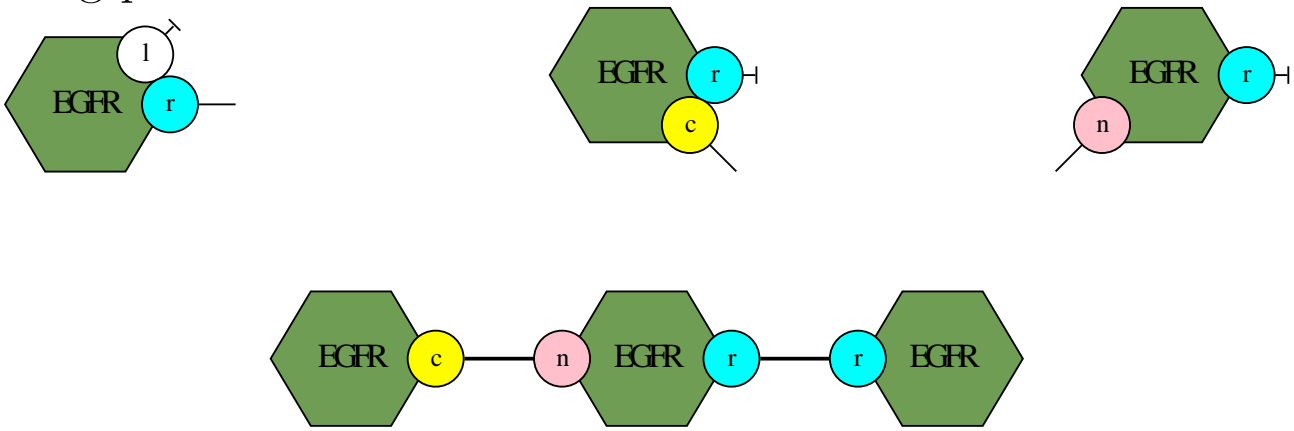
Some rules are unreachable possibly because of:

1. typos (or ambiguate ontology) on protein and/or site names;
2. missing parts in models;
3. unrealisable causality constraints.

KaSa uses the properties it has infered to detect some patterns that are unreachable. Due to abstraction, the other rules are not necessarily reachable (maybe).

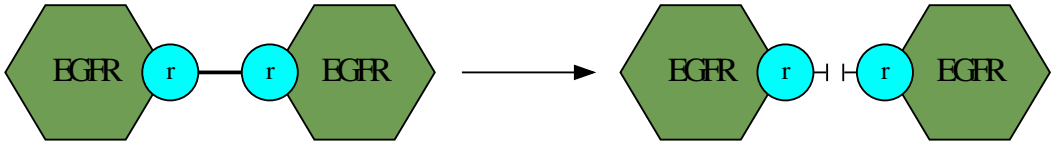
Dead rule/pattern detection
Demo

Are the following patterns:



reachable?

What if we consider the following additional rule:



?

Non-weakly reversible transitions: Motivations

$\Rightarrow$ when after a transition there is no way for a protein to return to its previous configuration (in one or several steps).

This may be due to:

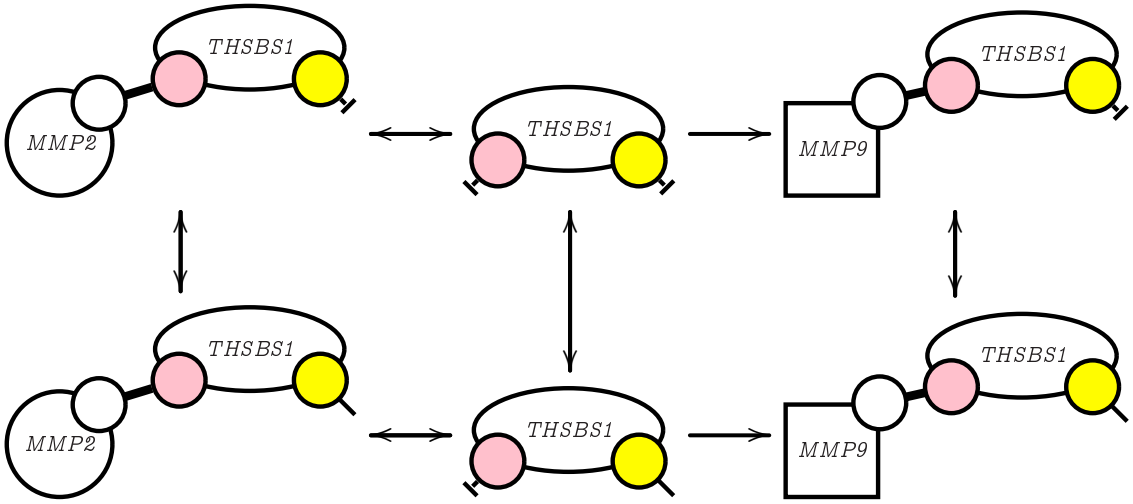
1. definitive degradation of a protein
   (the model must be documented accordingly);

2. some reverse mechanisms may be missing
   (the model must be completed accordingly).

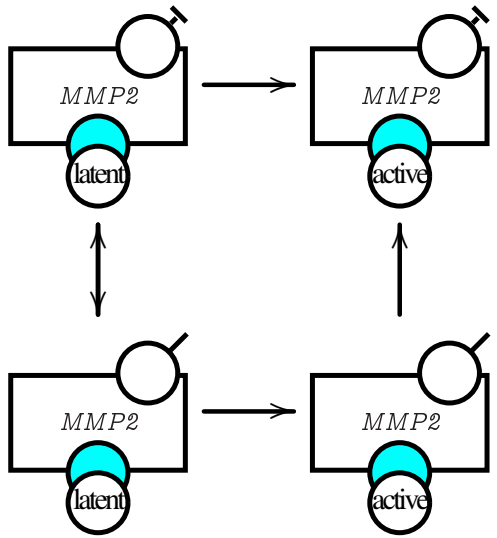
Demo

Non-weakly reversible transitions: Local traces [CMSB'16]

A local trace describes the potential transitions between the configurations of a given single protein.



Unbinding rules are missing.



Definitive degradation.

Polymers: Motivations

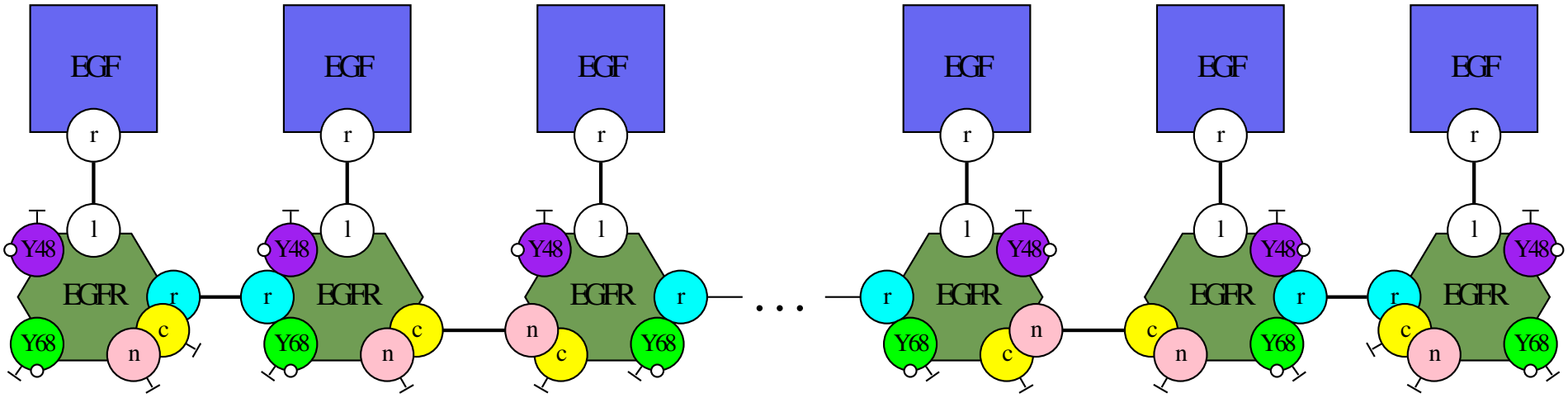
Potentially unbounded polymers:

1. may naturally emerge in models with self-assembling of giant macro-molecules
   - DNA (Jean Krivine),
   - filaments (Nathalie Theret),
   - signalosome (Hector Medica Abarca);

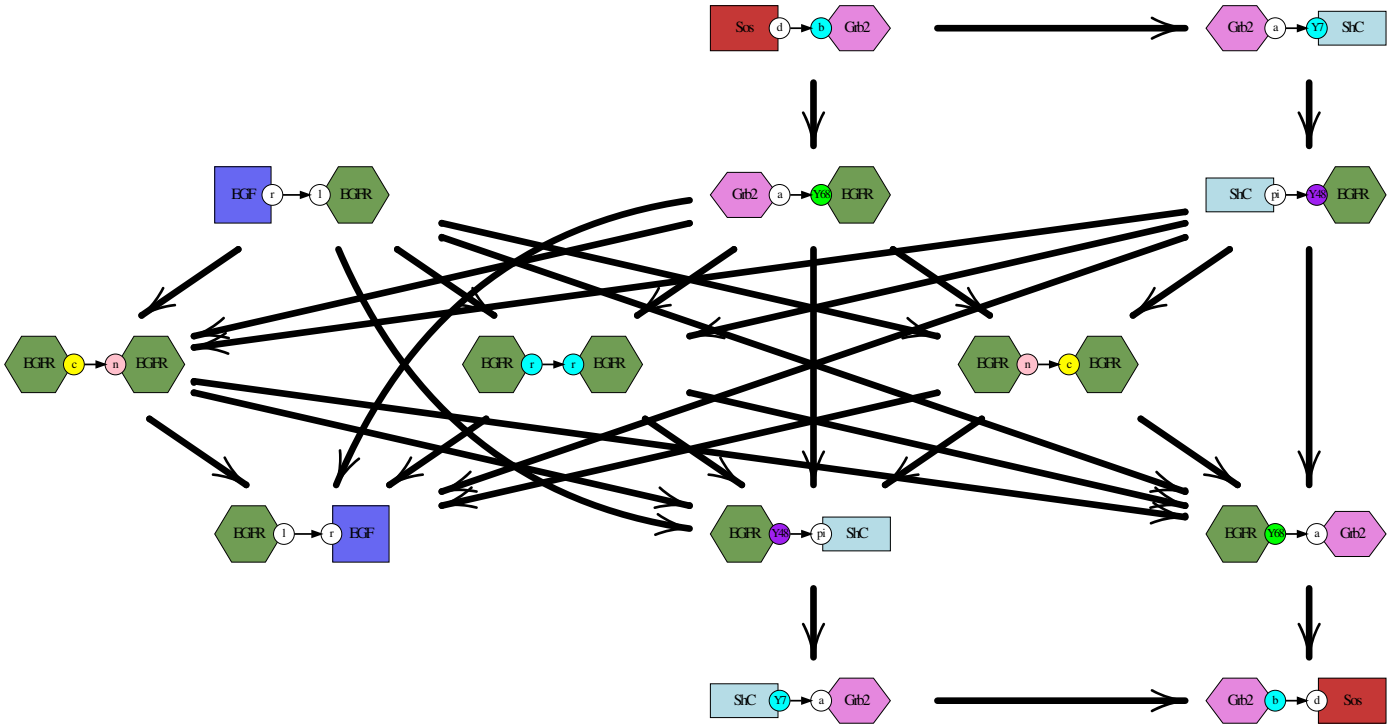
2. may result of missing conflicts between bonds;
   often the case when the model is extracted from a more abstract description
   - Cell-Designer (Luca Grieco),
   - natural language processing (Sorger Lab)

Polymers:
Our goal

We want to prove the absence of unbounded polymers:



Polymers:
Graph of the links[SASB'18]



Demo

On the menu today

1. Context and motivation
2. Kappa language
3. Main functionalities
4. Conclusion

Benchmarks

model	Number of												Analysis time (second)								
	rules	constraints	dead rules	non weakly rev. trans.	rules with non weakly rev. trans.	indirect influences	direct influences	realisable influences	SCC (Syntactic)	SCC (Realisable)	average size of SCC (Syntactic)	average size of SCC (Realisable)	reachability analysis	influence map (Indirect)	influence map (Direct)	influence map (Realisable)	polymer detection (Syntactic)	polymer detection (Realisable)	non weakly reversible transitions	local traces	all-in-one (at maxiaml accuracy)
egfr_net	39	17	0	0	0	764	280	280	0	0	N/A	N/A	0.02	0.04	0.02	0.04	0.02	0.02	0.03	0.02	0.05
fceri_fyn	46	21	0	8	8	758	304	304	1	0	8	N/A	0.04	0.05	0.03	0.10	0.04	0.04	0.06	0.04	0.10
fceri_fyn_lig	48	21	0	8	8	760	306	306	1	0	8	N/A	0.04	0.05	0.03	0.09	0.04	0.05	0.06	0.05	0.10
fceri_fyn_trimer	362	22	36	96	96	59971	7405	6763	1	0	10	N/A	0.53	4.63	0.66	2.15	0.54	0.54	0.58	0.54	2.24
fceri_fyn_gamma2	59	21	0	0	0	1464	518	518	1	0	8	N/A	0.06	0.10	0.04	0.15	0.06	0.06	0.08	0.06	0.16
fceri_fyn_ji	36	16	0	0	0	536	231	231	1	0	8	N/A	0.03	0.02	0.01	0.06	0.03	0.03	0.05	0.03	0.07
fceri_fyn_lyn_745	40	18	2	2	2	620	255	243	1	0	8	N/A	0.04	0.04	0.02	0.07	0.04	0.04	0.05	0.04	0.08
fceri_fyn_trimer	192	19	0	0	0	21557	2536	2536	1	0	10	N/A	0.24	1.60	0.24	0.81	0.24	0.24	0.27	0.24	0.86
sos	20	28	0	0	0	95	69	69	1	0	3	N/A	0.02	0.01	0.01	0.02	0.02	0.02	0.03	0.02	0.03
machine	220	72	7	17	10	5319	2873	2735	0	0	N/A	N/A	0.77	0.13	0.10	1.05	0.76	0.77	0.97	0.77	1.22
ensemble	233	86	0	1	1	4841	2936	2936	0	0	N/A	N/A	0.62	0.15	0.13	0.82	0.61	0.63	0.91	0.62	1.14
korkut (2017/01/13)	3916	1289	1610	2016	2016	75563	75563	39280	1	1	131	131	14	2.49	2.71	16	14	14	14	14	18
korkut (2017/02/06)	5750	2571	884	1397	1397	81412	75472	55101	1	1	2693	2687	94	4.01	4.16	94	96	115	99	114	119
TGF (V19)	97	107	10	153	53	3471	3009	2631	1	1	78	74	0.24	0.09	0.09	0.47	0.23	0.25	0.37	0.25	0.63
TGF (2018/04/19)	292	112	0	314	28	6040	5504	5504	1	1	108	108	0.89	0.18	0.19	1.36	0.86	0.90	1.25	0.92	1.73
BigWnt (2015/12/28)	356	134	1	833	14	5974	5271	5264	1	1	49	49	3.99	0.16	0.16	4.47	3.98	3.96	125	4.00	127
BigWnt (2017/03/22)	1486	182	12	61	16	1091187	38110	37958	1	1	84	80	15	26	5.15	25	15	15	260	15	286

On a MacBook Pro, 3.3 Ghz intel Core.

Kappa ecosystem

- Meta-modelling:
  - KaMi, Harmer (2014-), Basso-Blandin (2014-15) Le Cornec (2015-), Légaré (2015-), Oshurko (2015-)
  - Robin and Iota, Krivine (2014-) Adrien Husson (2014-)
- Static analysis: KaSa, Feret (2006-), Lỳ (2015-17)
- Simulator: KaSim, Krivine (2006-), Boutillier (2014-)
- Causality analysis: KaStor, Feret (2006-)
- Trace query language: KaTQL, Laurent (2018-)
- Network generation and model reduction: KaDe, Feret (2006-)
- User interface: KaUI, Boutillier (2015-)

Tutorialis and online tools available at:

http://www.kappalanguage.org

Future developments

1. Domains for the detection of unbounded bio-molecular compounds.
   (Aurélie Faure De Pebeyre, AIV M1 internship)
2. Non local properties (based on Floyd-Warshall closure).
   Reasoning on polymers and molecular ambiguity.
3. Solver for algebraic equations over patterns.