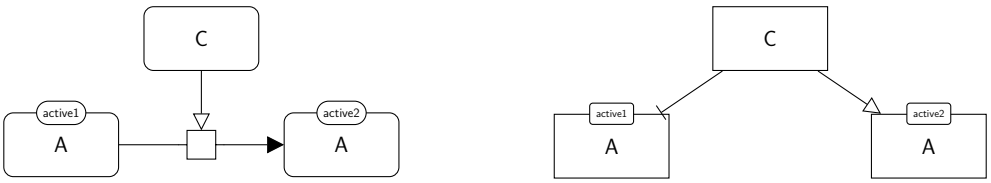
1 Comments

Below I give a few comments on the rules of the specification. I do not know how the active EPNs are obtained, so I am sorry in adance if some of my comments are completely irrelevant.

Comment 1 When multiple forms of a same macromolecule are active, the distinction between the resulting activities could be made using some units of information rather than the labels. For example, in place of 1.16 a-stim-a, we could have:



That would for example leave the possibility to merge activities that have the same labels and are targeted by the same activities.

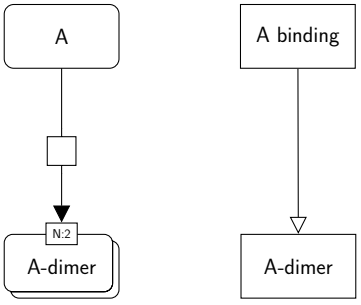
This concerns rules 1.16 a-sti-a, 1.17 a-cat-a, 1.18 a-nsti-a, 1.19 a-mod-a, and 1.20 a-inh-a.

Comment 2 I think that a macromolecule that is not (found as) active should not result in an activity. So for example in 1.1 i-sti-i, A is not active, and thus should not result in an activity A. This concerns rules 1.1 i-sti-i, 1.2 i-cat-i, 1.3 i-nst-i, 1.4 i-mod-i, 1.5 i-inh-i and 2.1 ss-sti-i.

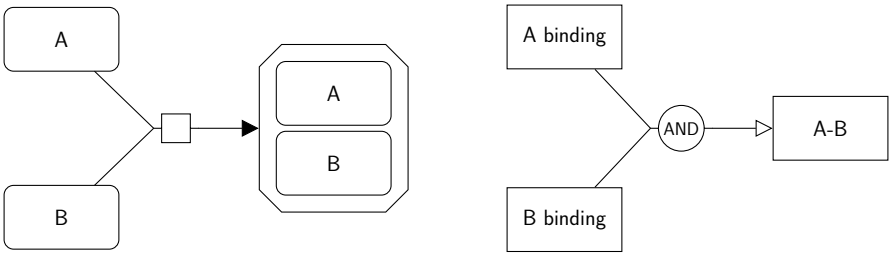
At worst we could make the hypothesis that every macromolecule has at least one active form that would result in an activity; but then, in 1.1 i-sti-i, 1.2 i-cat-i, 1.3 i-nst-i, and 1.5 i-inh-i, we would still not know how activity C modulates activity A, so we would need to have an unknown influence from C to A, and not a stimulation or an inhibition. I believe this hypothesis would however make 2.1 ss-sti-i correct.

Comment 3 In 1.27 a-catr-a, the catalysis results in two stimulations. The "sign" (stimulation or inhibition) of a resulting AF modulation depends on the "sign" of the PD modulation and the direction of the modulated reaction. Since here we do not know which direction is the favoured reaction, I believe it should rather result in two unknown influences.

Comment 4 When the activity is a binding activity, it could be specified in the label. For example:



Or:

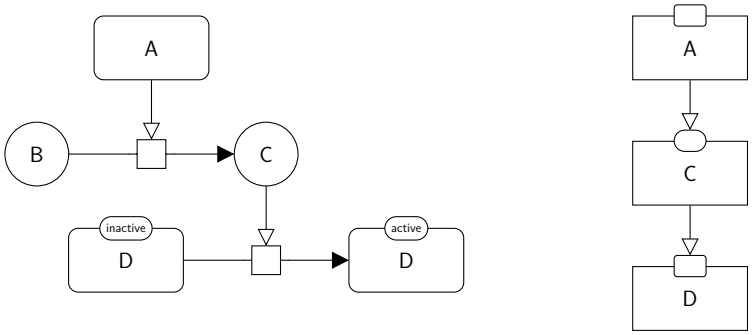


This concerns 3.1 Dimerization x2, 3.N Oligomerization xN, 3.10 Multimerization 1, 3.11 Multimerization 2, 3.12 Oligomerization x6 r, 5.1 Complex association with complex name, 5.2 Complex association, 5.3 Complex association regulated.

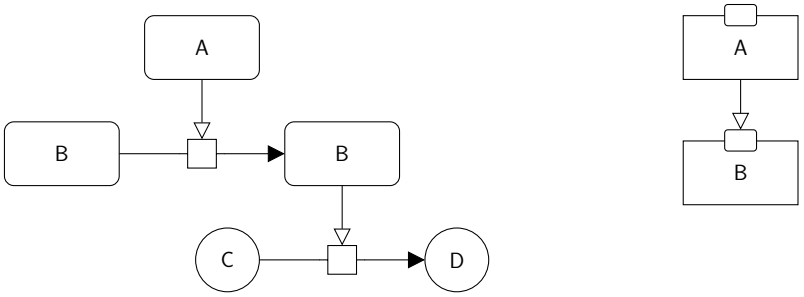
Comment 5 When transforming PD maps to AF maps, only EPNs having an activity (i.e. realizing a function) may result in an AF activity. Hence, in general, PD processes cannot be represented in AF, because the participants (reactants and products) do not have any activity. As a consequence, rules such as 1.21 m-sti-m, that transform metabolic processes (involving simple chemicals) are not correct.

I believe this might not be a problem because I do not think such metabolic reactions have to be transformed as such:

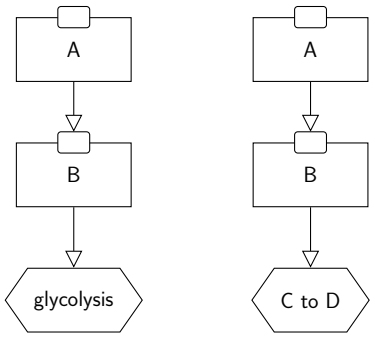
- either the metabolic reaction (or the chain of metabolic reactions) is in the middle of a signalling cascade, and then some product of the reaction/chain must be active, so that it can be transformed the same way as reactions activating macromolecules. For example:



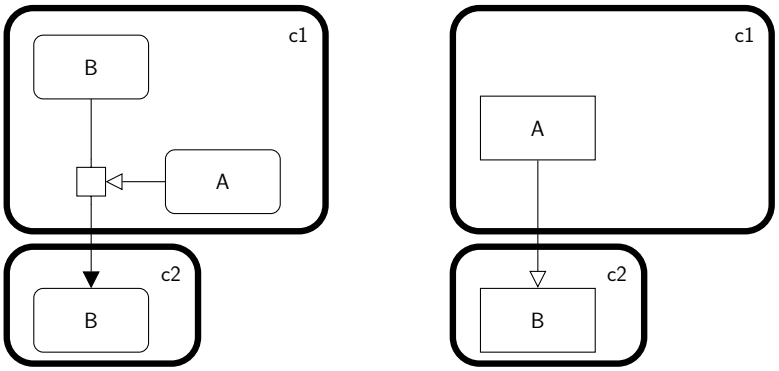
- or the metabolic reaction is at the end of the signalling cascade, and it may not be represented at all:



or it may be represented as a phenotype (i.e. a biological process that is an outcome of the pathway). The ideal would be to link the oucome reaction to a "meta" process (e.g. glycolysis, gluconeogenesis) but that would require knowledge that would be outside the input PD map. The less ideal way (but still better than representing processes in AF, in my opinion) would be to name the outcome phenotype after the metabolic reaction (e.g. having a phenotype with a label "β-D-fructose-6P to β-D-fructose-2,6P2"). For example, provided that the reaction transforming C into D is a forward reaction in glycolysis, the above PD map could be transformed in either of the following ways (ideal and not ideal):



As for tanslocations, they could only be represented in AF when there is an active transporter:



This comment concerns rules 1.21 m-sti-m, 1.22 m-cat-m, 1.23 m-nsti-m, 1.24 m-mod-m, 1.25 m-inh-m, 1.26 aiaimm-cat-aiiamm, 1.28 Logic gates, 4.1 Translocation 1, 4.2 Translocation 2.

2 Some material that could be useful

I have been developing a logical framework based on the concepts of SBGN, and to illustrate the use of this framework, I have built a small logic-based method to transform PD maps into AF maps (for signalling networks). Maybe some ideas/patterms used in this method can be of any help. This work is not yet published so below I give material from my PhD thesis (sorry for the French language) and from a presentation I gave at COMBINE 2017.

The method relies on first finding wich EPNs are active, then transforming these EPNs into activities, and finally transforming PD paths between those EPNs into AF modulations.

Figure 3.11 gives the conditions for an EPN to be active: an EPN E is active if and only if it belongs to one of the 5 patterns (where gray EPNs are already found active EPNs). I used unspecified entities to imply that the nature of the EPN is not important. Each active EPN is transformed into an AF activity. Pattern 3 might be new: an EPN E that can bind to an EPN E' that is active might inhibit the latter (and hence be active).

Figure 3.12 gives the paths in the PD maps that will be transformed in AF modulations. Patterns in 6 gives the definition of a chain of reaction (that can be found by transitivity: a reaction is a chain, and a reaction plus a chain gives a new chain). Any of the three paths in 7 is transformed into a stimulations from activity A to activity B, and any of the three paths in 8 is transformed into an inhibition. The use of chains of reactions might be new here.

Below are some slides that explain rapidly the method (you can ignore the logical formulas). There is a real life example on the last slide, that was obtained completely automatically (transformation of PD to AF + layout).

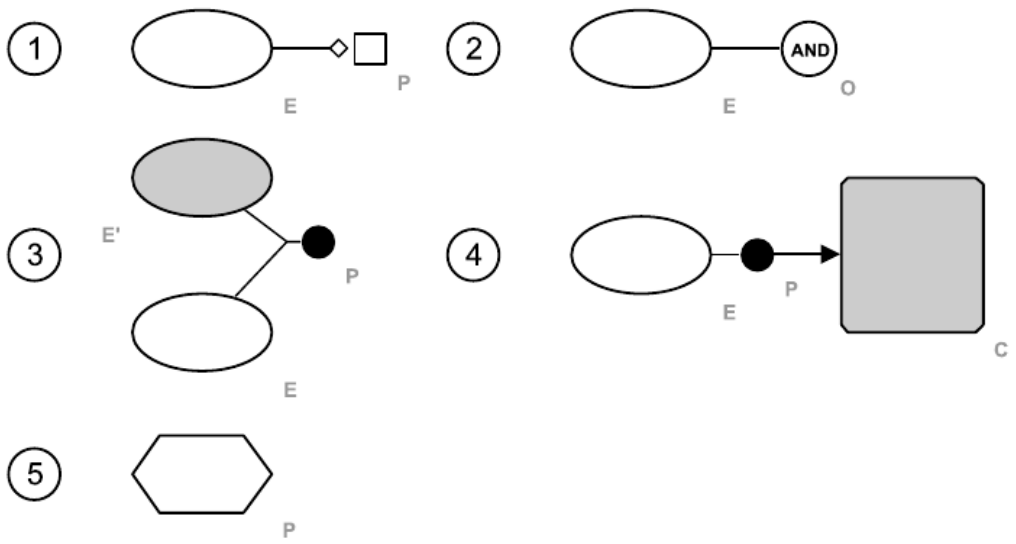


FIGURE 3.11 – **Motifs définissant les EPNs actifs.** Pour chacun des cinq motifs, si l'EPN E appartient à ce motif, alors il opère une activité. Les EPNs colorés en gris sont des EPNs qui ont préalablement été définis comme actifs, i.e. qui appartiennent également à l'un des six motifs représentés.

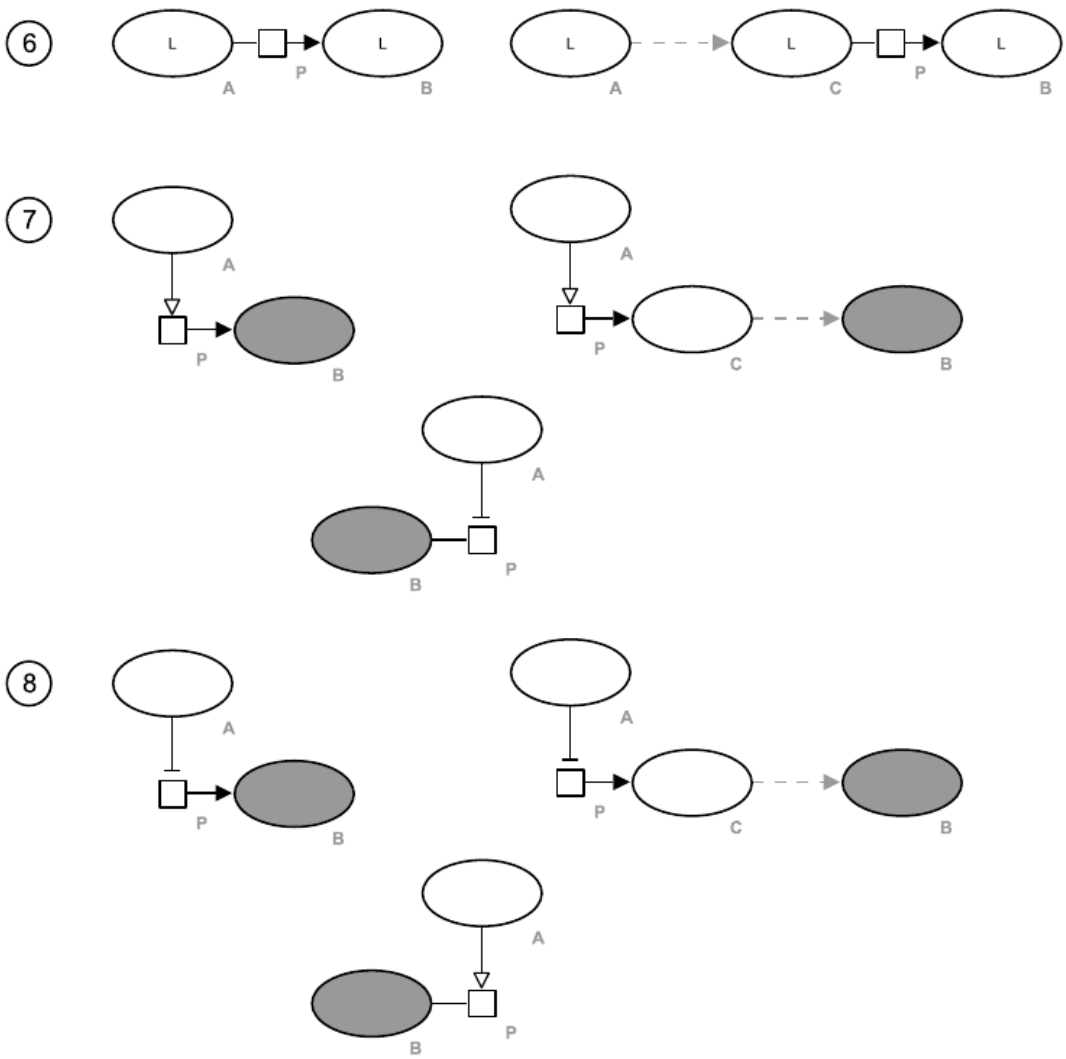
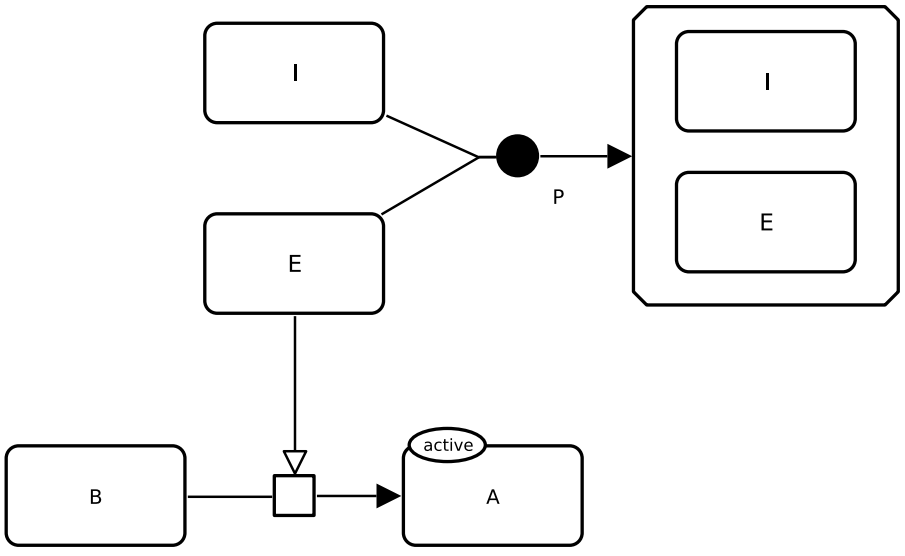


FIGURE 3.12 – **Motifs définissant les chemins.** ⑥ : motifs définissant les chemins réactionnels. ⑦ : motifs définissant les chemins positifs. ⑧ : motifs définissant les chemins négatifs. Les EPNs colorés en gris sont des EPNs actifs. Les flèches en pointillés et colorées en gris sont des chemins réactionnels. Pour les motifs ⑦ et ⑧, la source de la modulation, dénotée par A, est représentée sous la forme d'un EPN, mais peut également être un opérateur logique.

Example: reaction networks into influence graphs

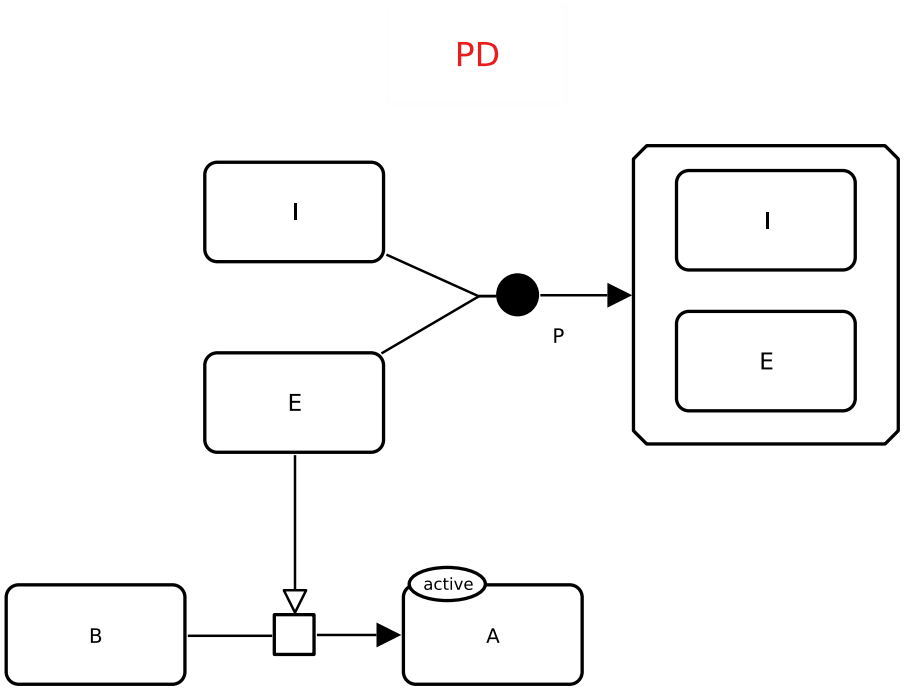
PD



AF

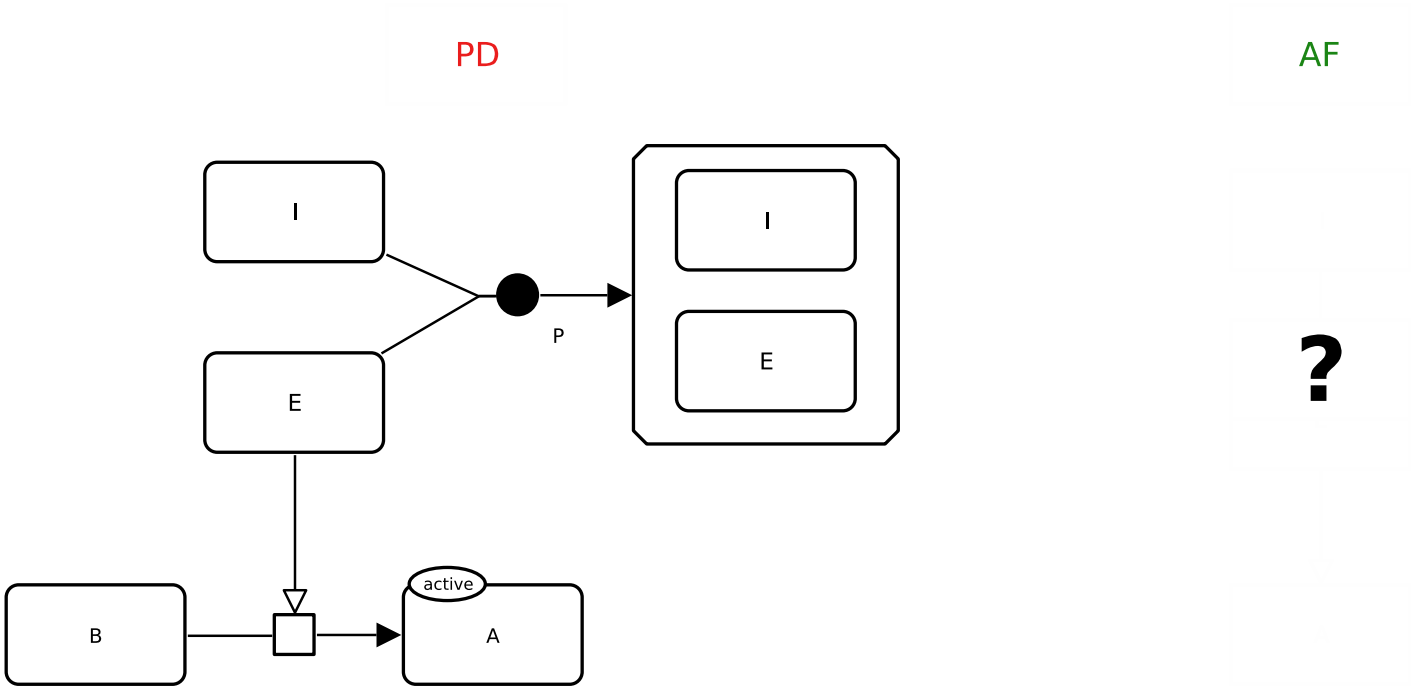
?

Example: reaction networks into influence graphs



- Find which entity pools are active
- PD active entity pools $\rightarrow$ AF activities
- Relations between PD active entity pools $\rightarrow$ AF modulations

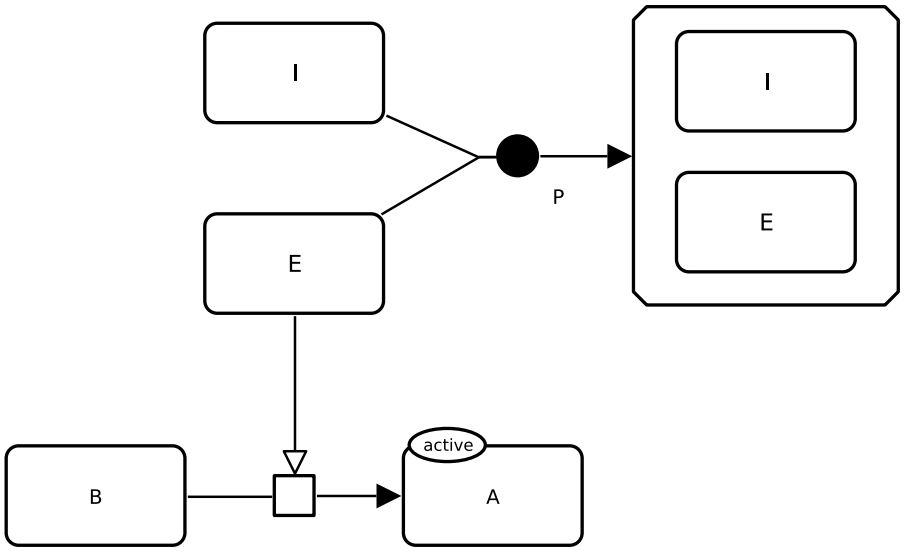
Example: reaction networks into influence graphs



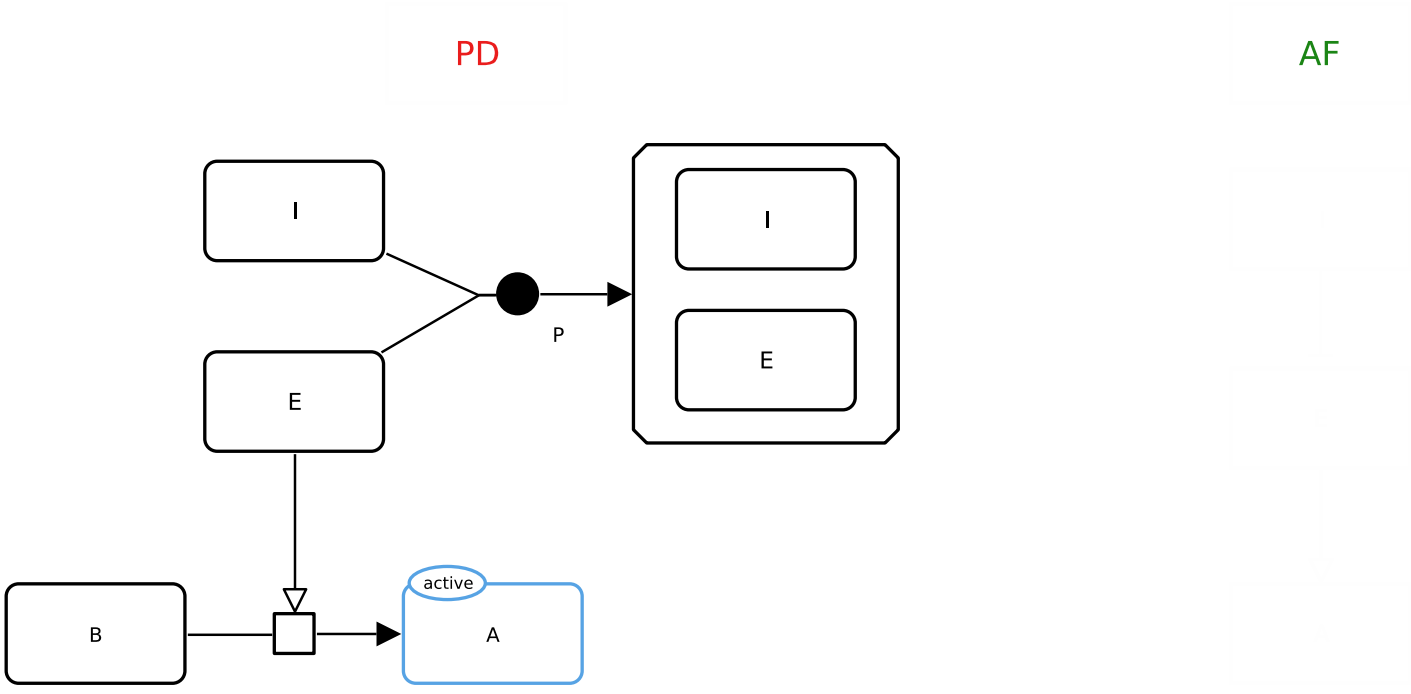
- Find which entity pools are active
- PD active entity pools $\rightarrow$ AF activities
- Relations between PD active entity pools $\rightarrow$ AF modulations

Finding and transforming patterns!

Example: reaction networks into influence graphs



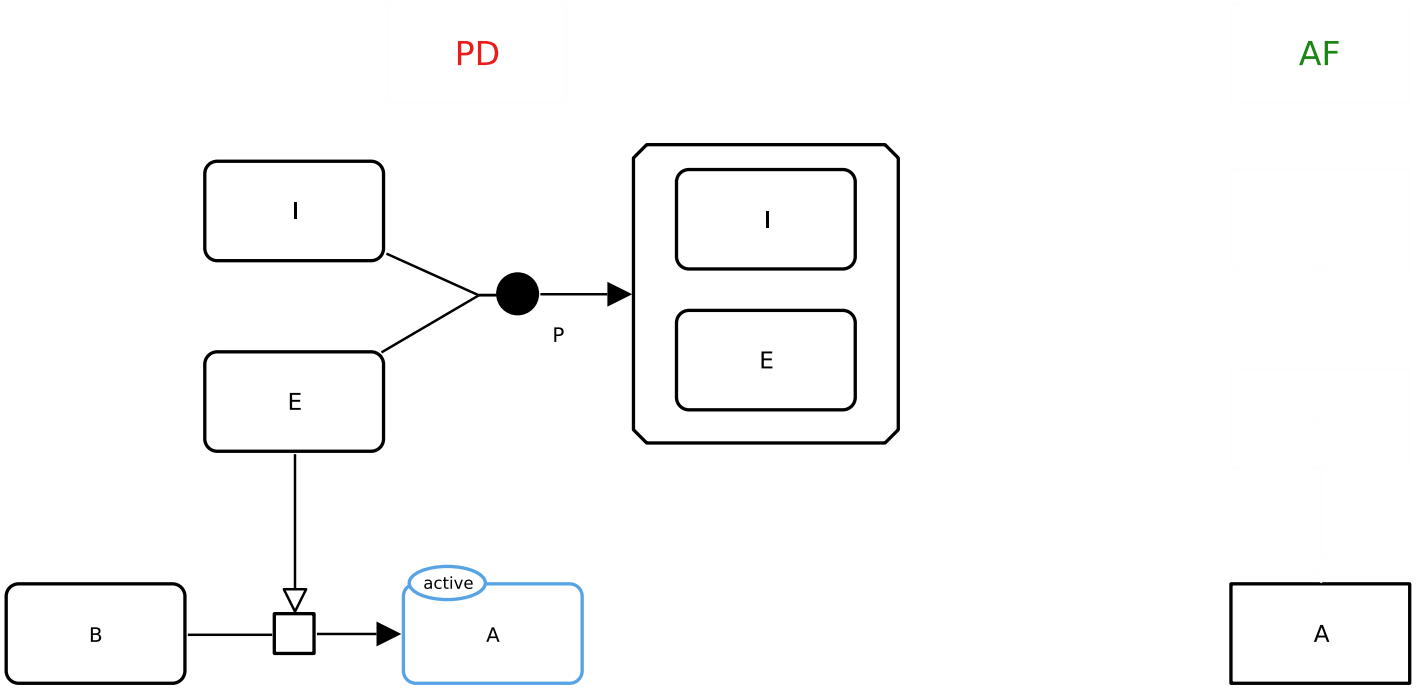
Example: reaction networks into influence graphs



If A is active, then A has an activity

$$epn_{PD}(A) \wedge stateVariable_{PD}(A, \text{"}active\text{"}, Var) \Rightarrow hasActivity(A)$$

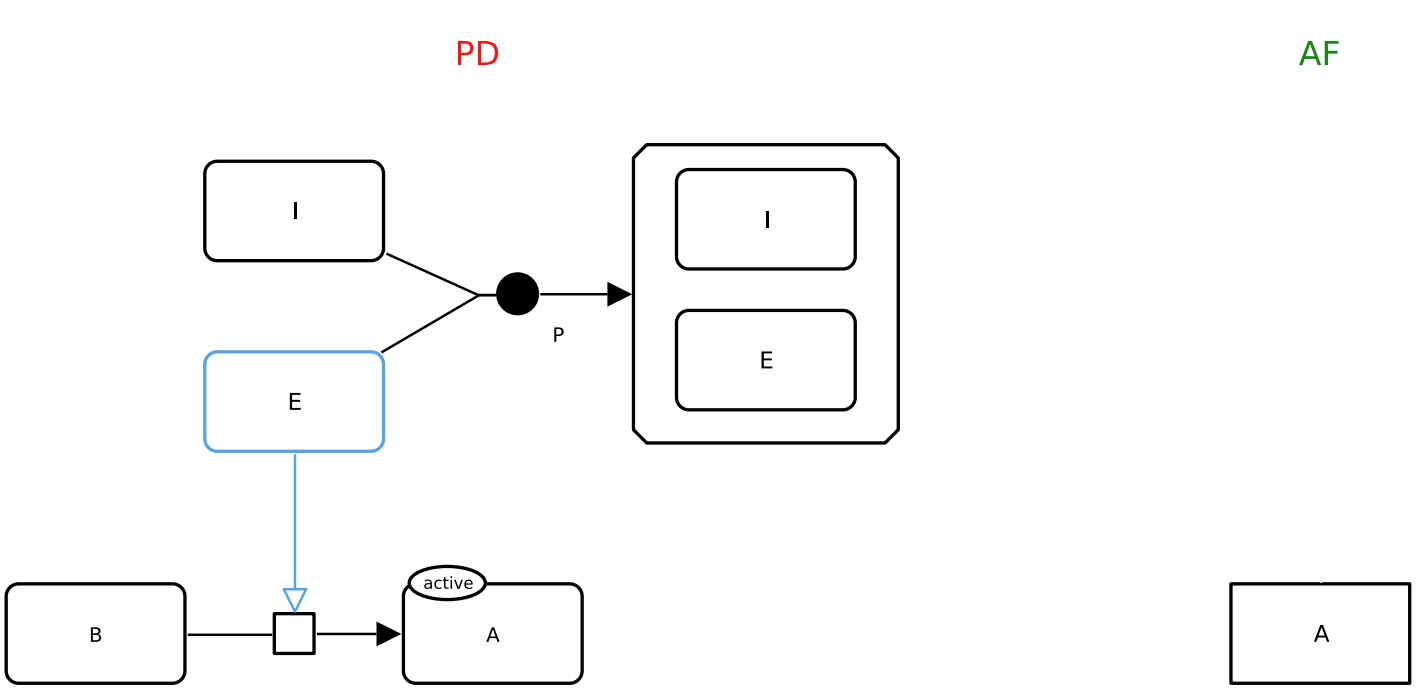
Example: reaction networks into influence graphs



We build a new activity

$$hasActivity(A, R) \Rightarrow biologicalActivity_{AF}(a(A))$$
$$hasActivity(A, R) \wedge labeled_{PD}(A, L) \Rightarrow labeled_{AF}(a(A), L)$$

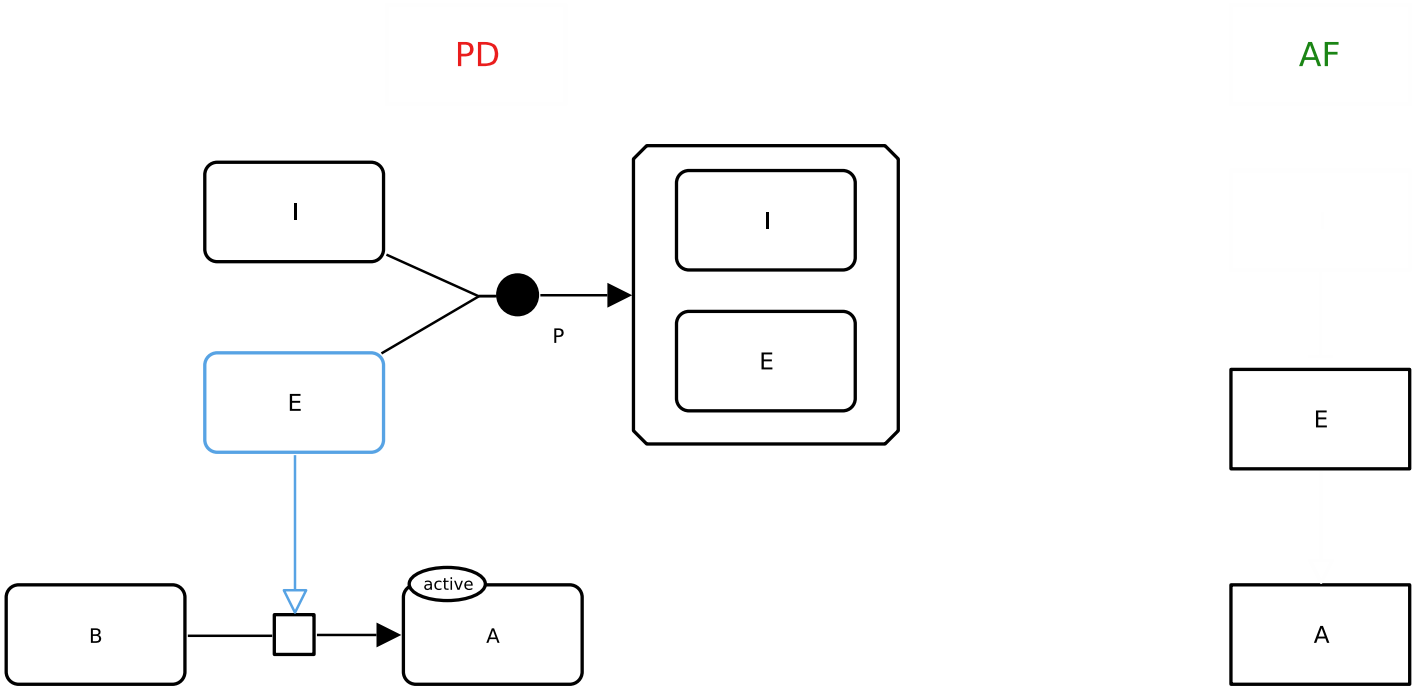
Example: reaction networks into influence graphs



If E stimulates a process, then E has an activity

$epn_{PD}(E) \wedge process_{PD}(P) \wedge stimulates_{PD}(E, P) \Rightarrow hasActivity(E, mod(P))$

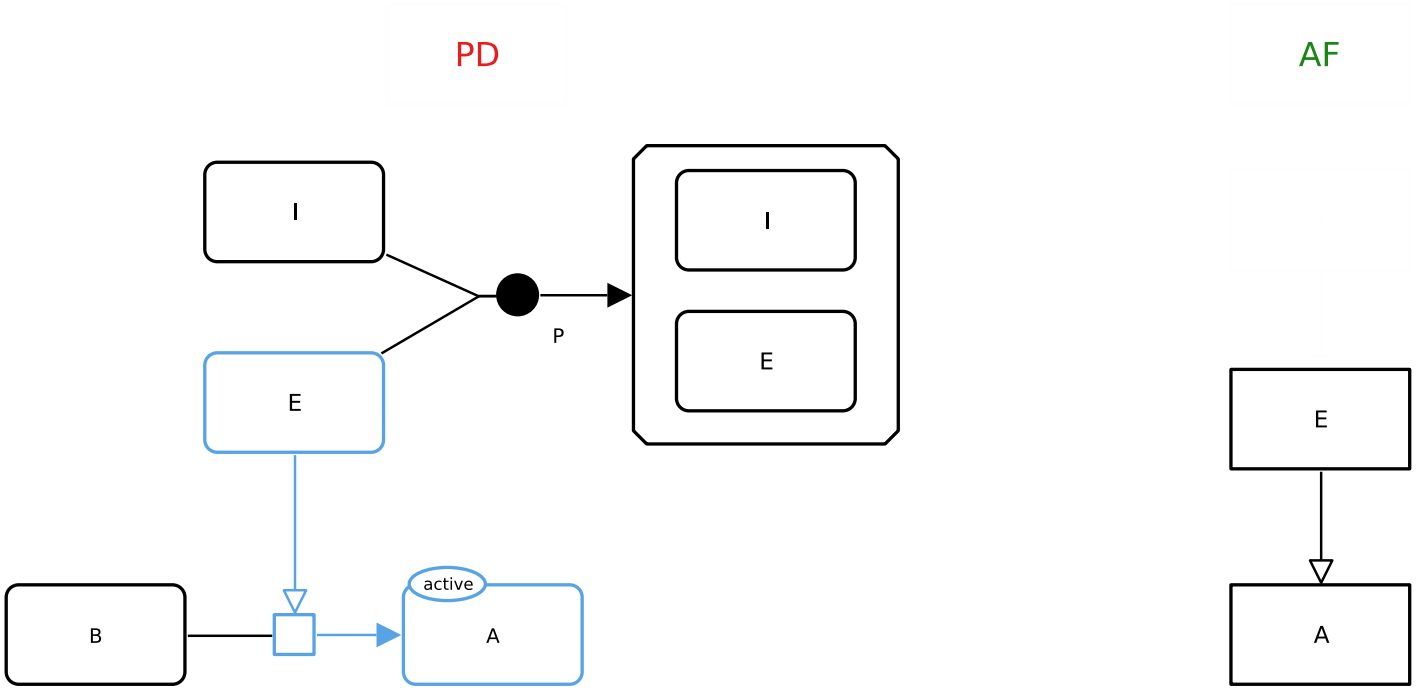
Example: reaction networks into influence graphs



We build a new activity

$$hasActivity(E, R) \Rightarrow biologicalActivity_{AF}(a(E))$$
$$hasActivity(E, R) \wedge labeled_{PD}(E, L) \Rightarrow labeled_{AF}(a(E), L)$$

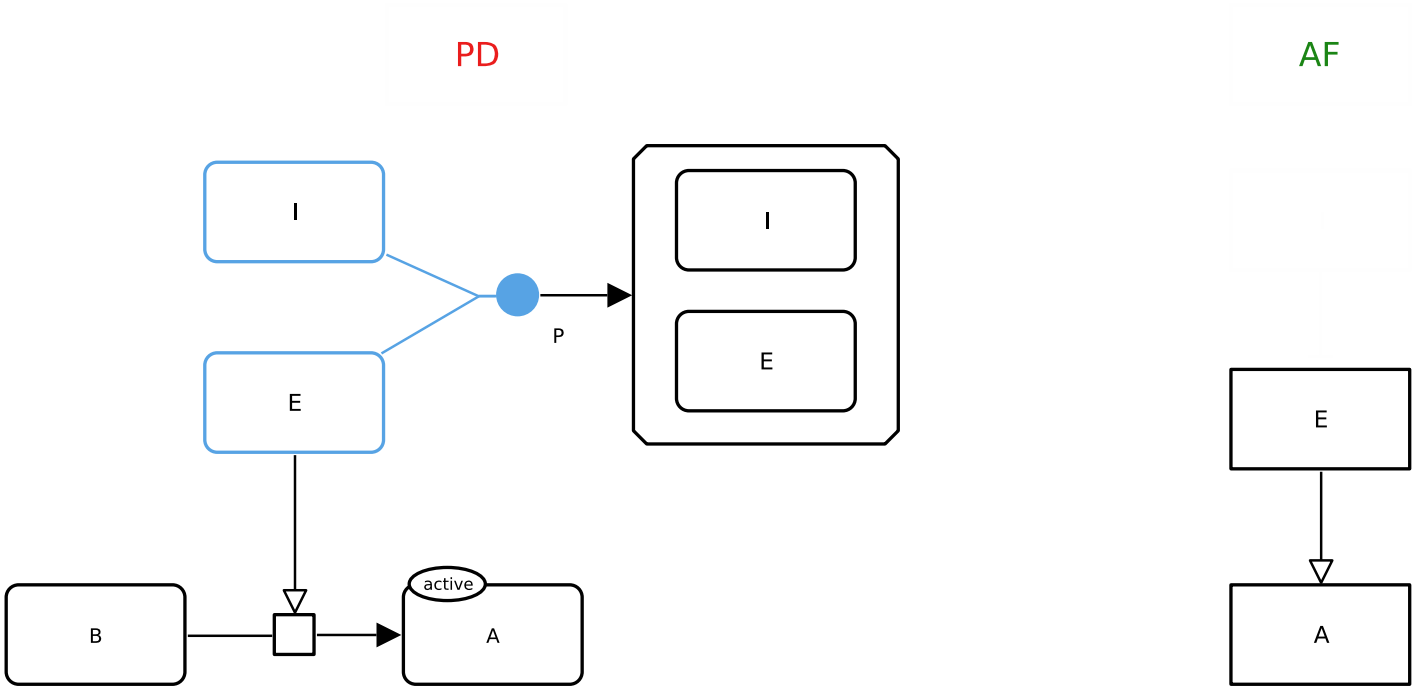
Example: reaction networks into influence graphs



If E stimulates a process that produces A that has an activity, then the activity of E stimulates the activity of A

$$hasActivity(E, mod(P)) \wedge produces_{PD}(P, A, N) \wedge$$
$$hasActivity(A, R) \Rightarrow stimulates_{AF}(a(E), a(A))$$

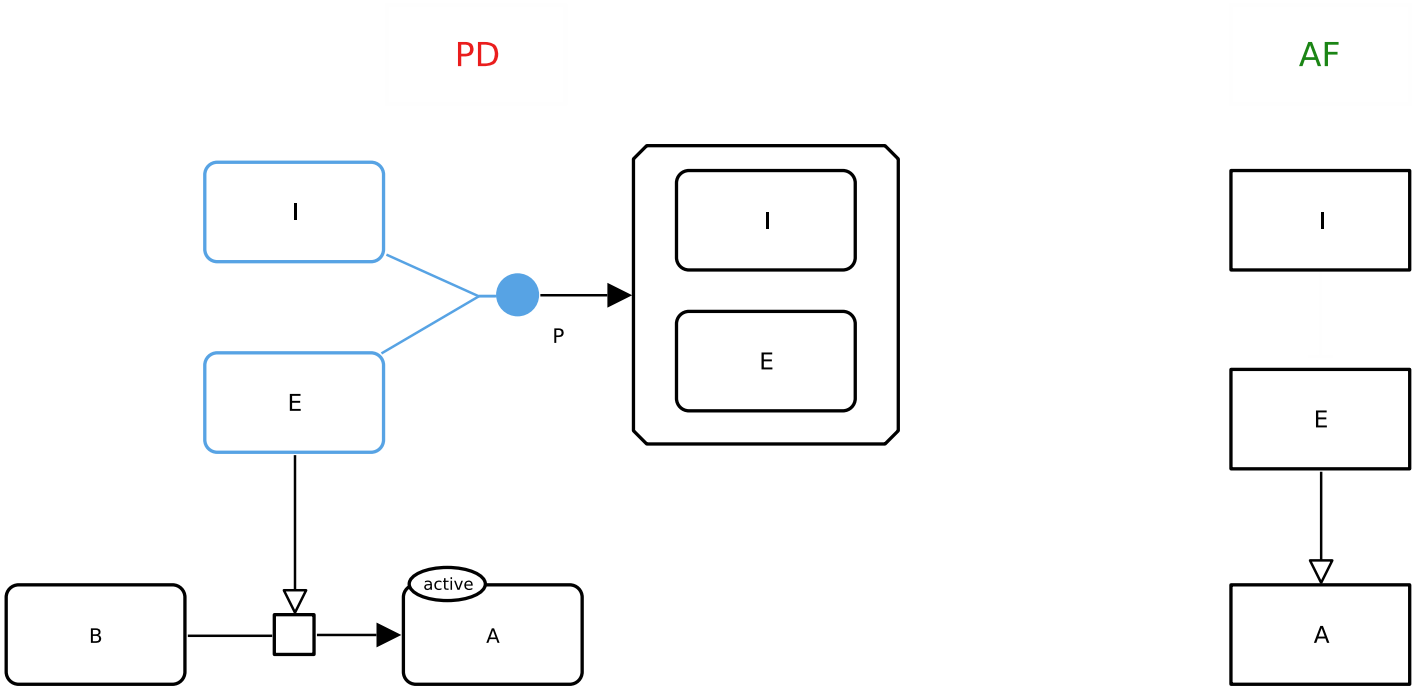
Example: reaction networks into influence graphs



If I can form a complex with E that has an activity, then I has an activity (inhibition)

$$epn_{PD}(I) \wedge assocation_{PD}(P) \wedge consumes_{PD}(P, I, Ni) \wedge consumes_{PD}(P, E, Ne) \wedge hasActivity(E, R) \Rightarrow hasActivity(I, inh(E))$$

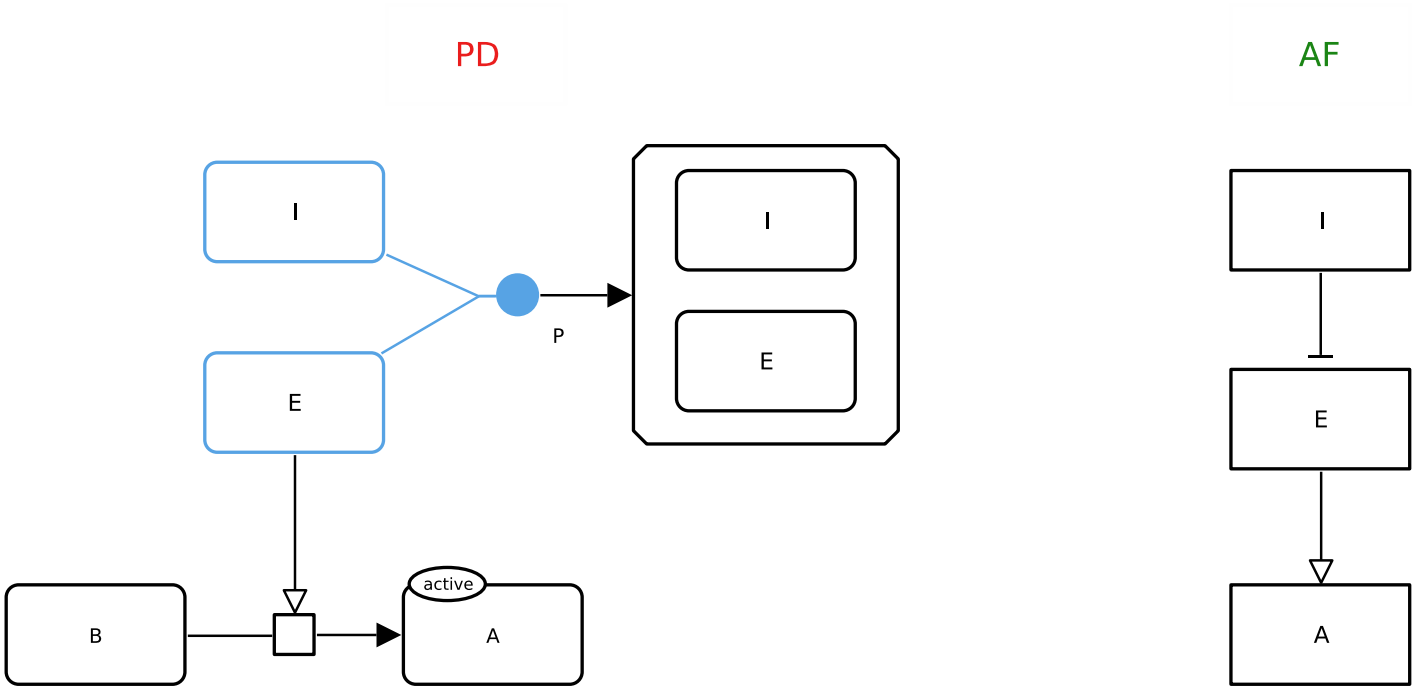
Example: reaction networks into influence graphs



We build a new activity

$$hasActivity(I, R) \Rightarrow biologicalActivity_{AF}(a(I))$$
$$hasActivity(I, R) \wedge labeled_{PD}(I, L) \Rightarrow labeled_{AF}(a(I), L)$$

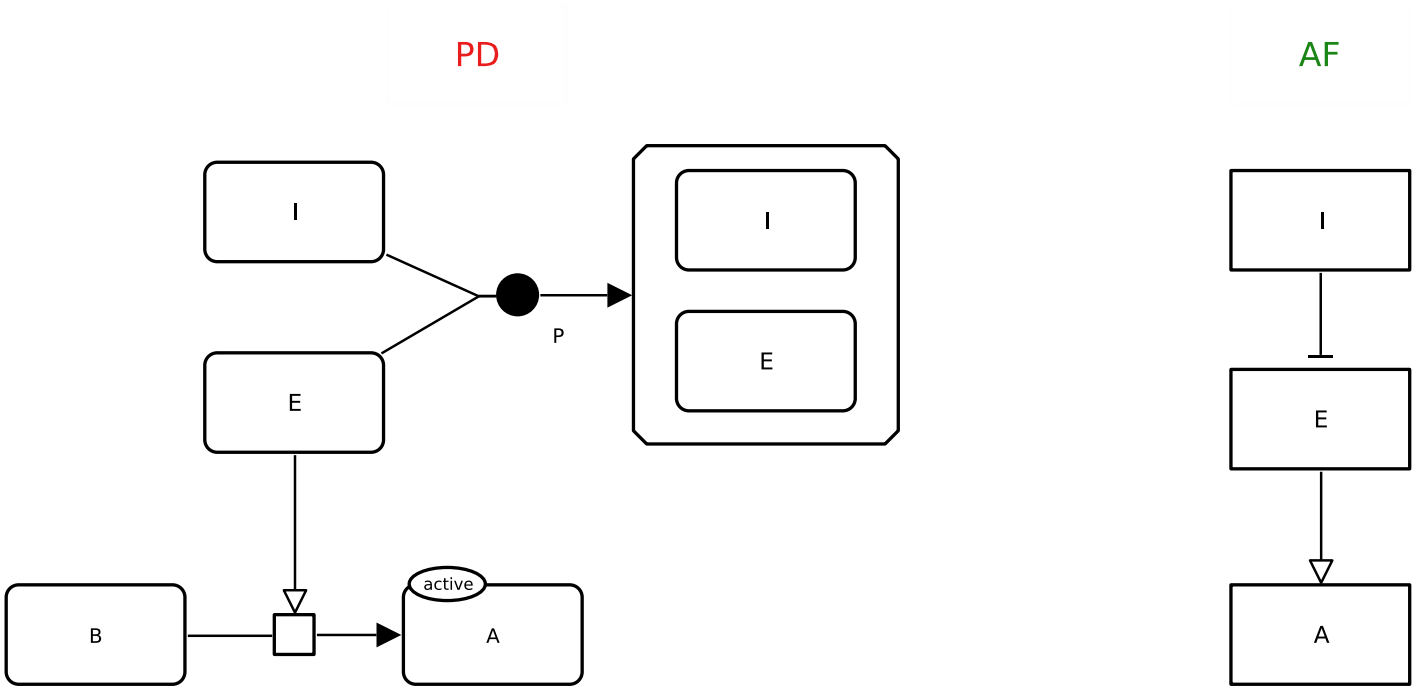
Example: reaction networks into influence graphs



If I can form a complex with E that has an activity, then the activity of I inhibits the activity of E

$$hasActivity(I, inh(E)) \wedge hasActivity(E, R) \Rightarrow inhibits_{AF}(a(I), a(E))$$

Example: reaction networks into influence graphs



$epn_{PD}(E) \wedge stateVariable_{PD}(A, \text{"}active\text{"}, Var) \Rightarrow hasActivity(A, active)$

$epn_{PD}(E) \wedge process_{PD}(P) \wedge stimulates_{PD}(E, P) \Rightarrow hasActivity(E, mod(P))$

$epn_{PD}(E) \wedge assocation_{PD}(P) \wedge consumes_{PD}(P, E, Ne) \wedge consumes_{PD}(P, E', Ne') \wedge hasActivity(E', R) \Rightarrow hasActivity(E, inh(E'))$

$hasActivity(E, R) \Rightarrow biologicalActivity_{AF}(a(E))$

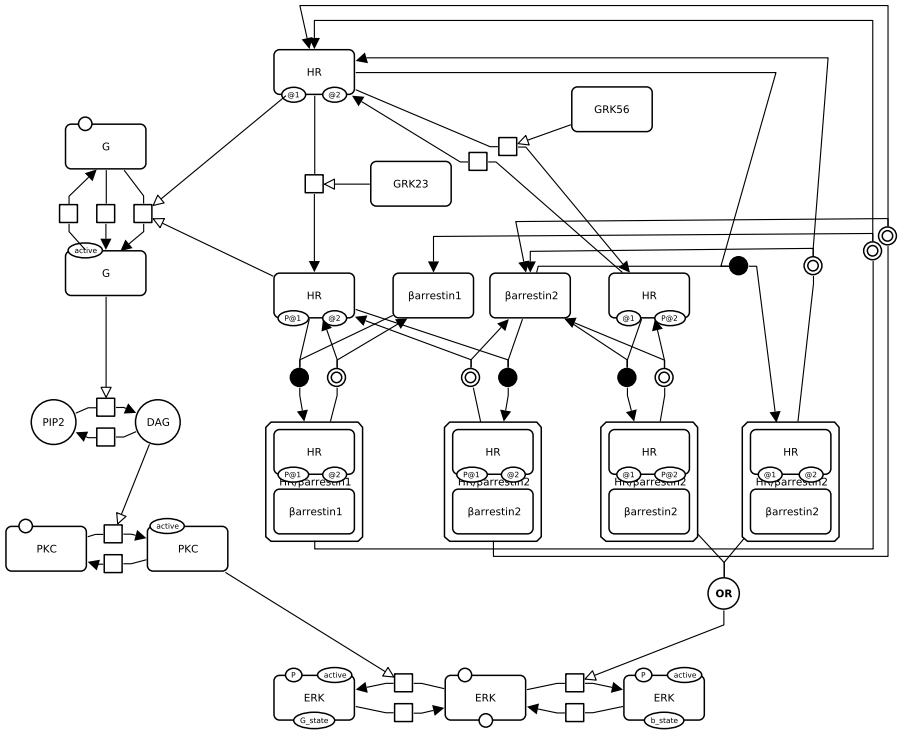
$hasActivity(E, R) \wedge labeled_{PD}(E, L) \Rightarrow labeled_{AF}(a(E), L)$

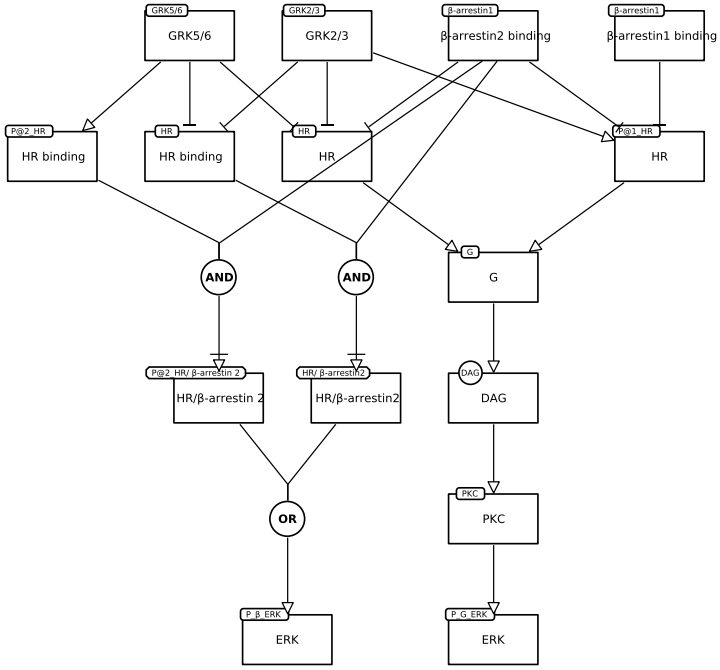
$hasActivity(E, mod(P)) \wedge produces_{PD}(P, E', N) \wedge hasActivity(E', R) \Rightarrow stimulates_{AF}(a(E), a(E'))$

$hasActivity(E, inh(E')) \wedge hasActivity(E', R) \Rightarrow inhibits_{AF}(a(E), a(E'))$

Example: reaction networks into influence graphs

- Logic program: ≈ 100 rules to identifty and transform patterns
- Input: SBGNlog PD map
- Output: SBGNlog AF map





Adapted from [Heitzler *et al.*, 2012]