Using Decision Diagrams for the Definition and Analysis of Logical Models of Regulatory Networks

Aurélien Naldi,
Claudine Chaouiya, Denis Thieffry

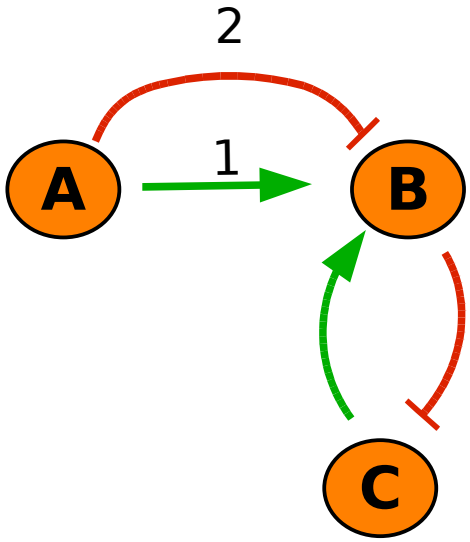
MOCA -- June 25-26 2007

http://tagc.univ-mrs.fr

Outline

- Logical Formalism
- Application: Th Activation and Differentiation
- Decision Diagrams
- Functionality of Regulatory Circuits
- Finding Stable States
- Conclusion

Logical Formalism



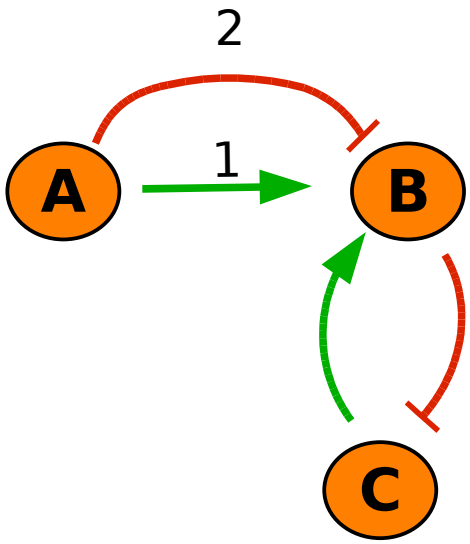
Regulatory graph

- Genes (A, B, C)
  - Discrete expression levels (Boolean or multivalued)
- Interactions
  - Activity threshold
  - Activations, inhibitions

Dynamical rules

- Target expression level depending on regulators

Evolution Rules

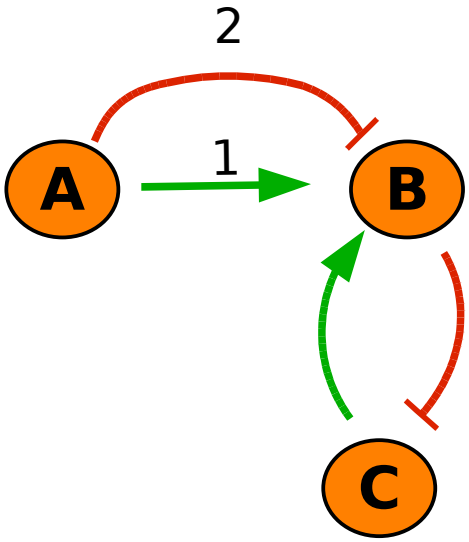


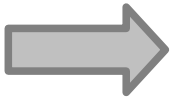
- B expressed in presence of A at low level or of C (or both)
- C expressed in absence of B

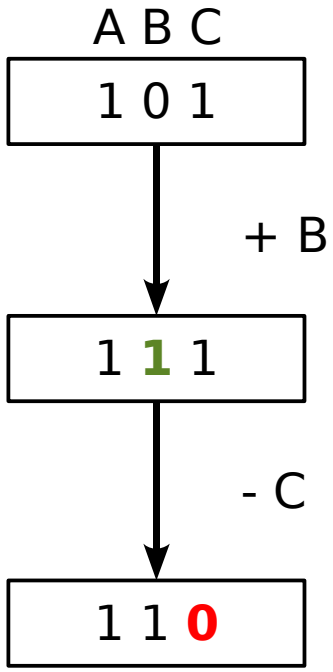
Dynamics of B given by the **logical function K_B**

$$K_B = \begin{cases} 1 & if \quad (A_1 \vee C) \\ 0 & otherwise \end{cases}$$

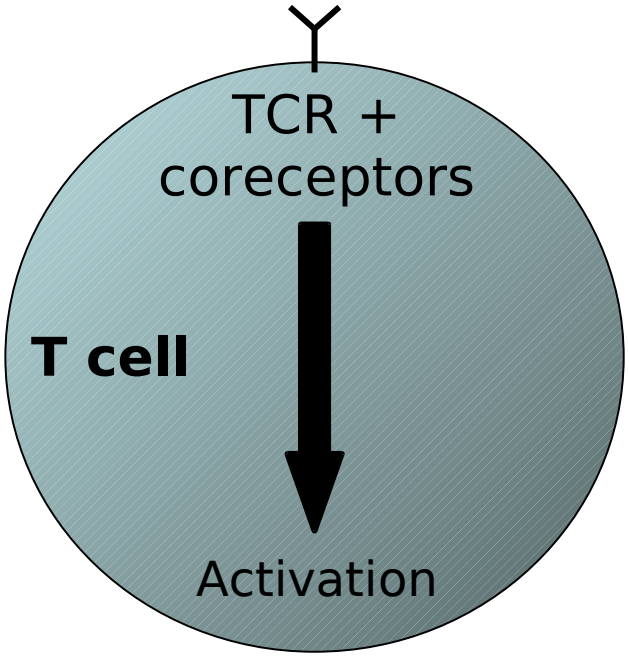
State Transition Graph



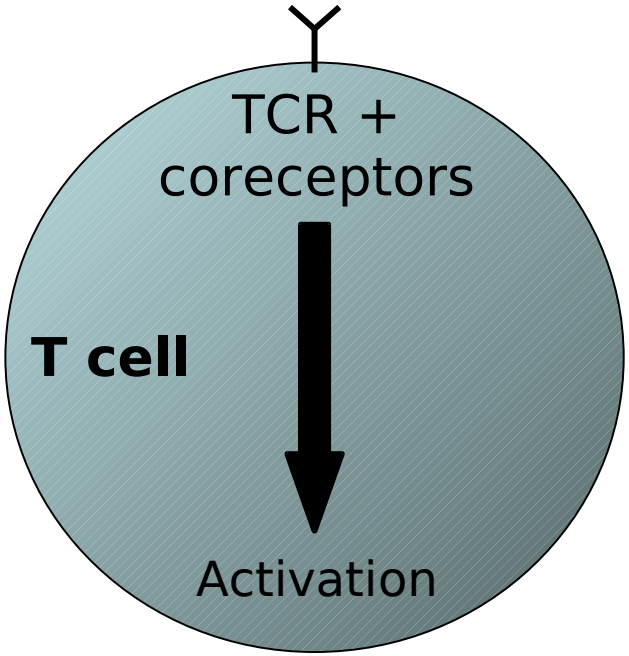


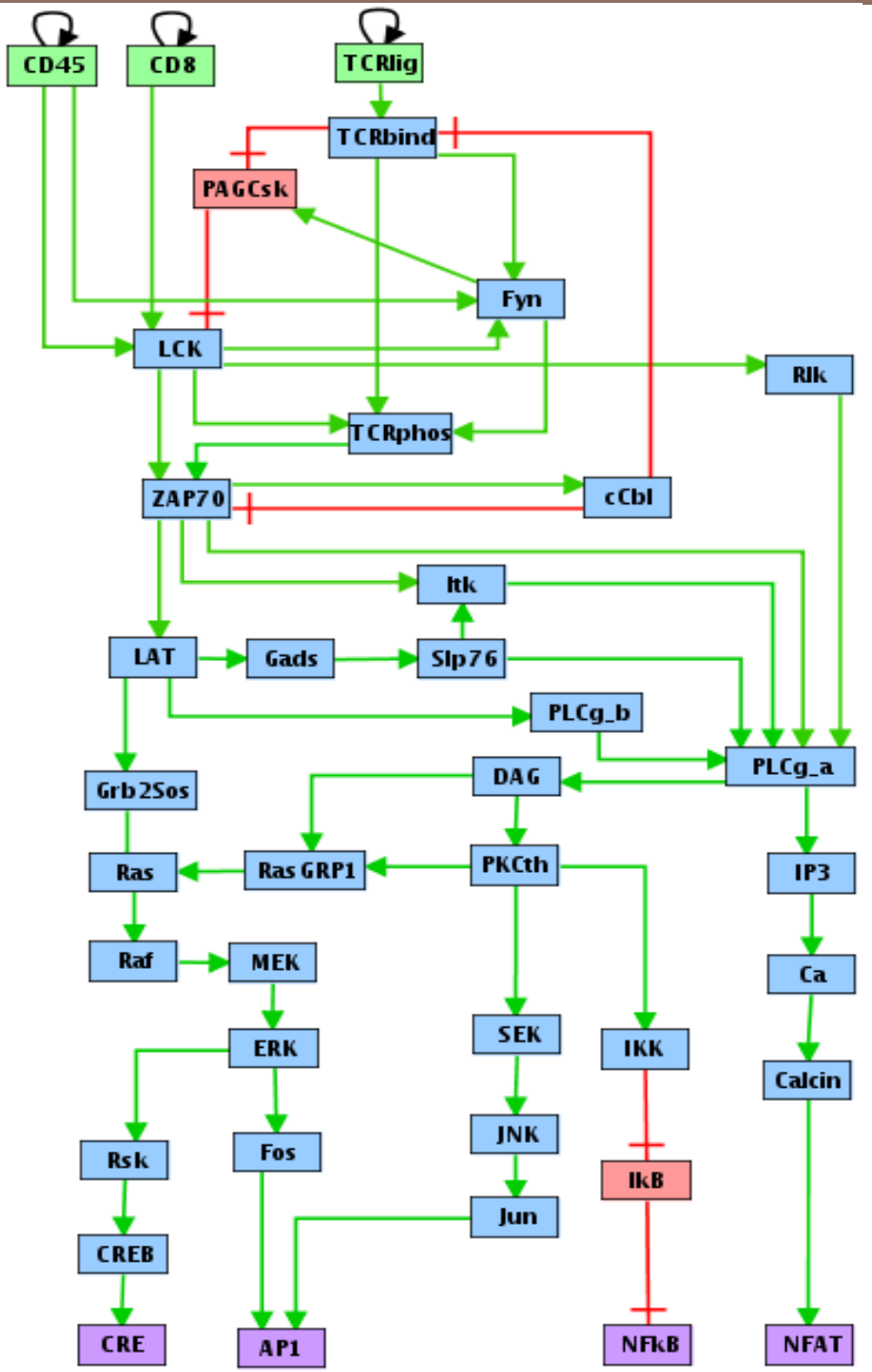


Application: TCR Signalling



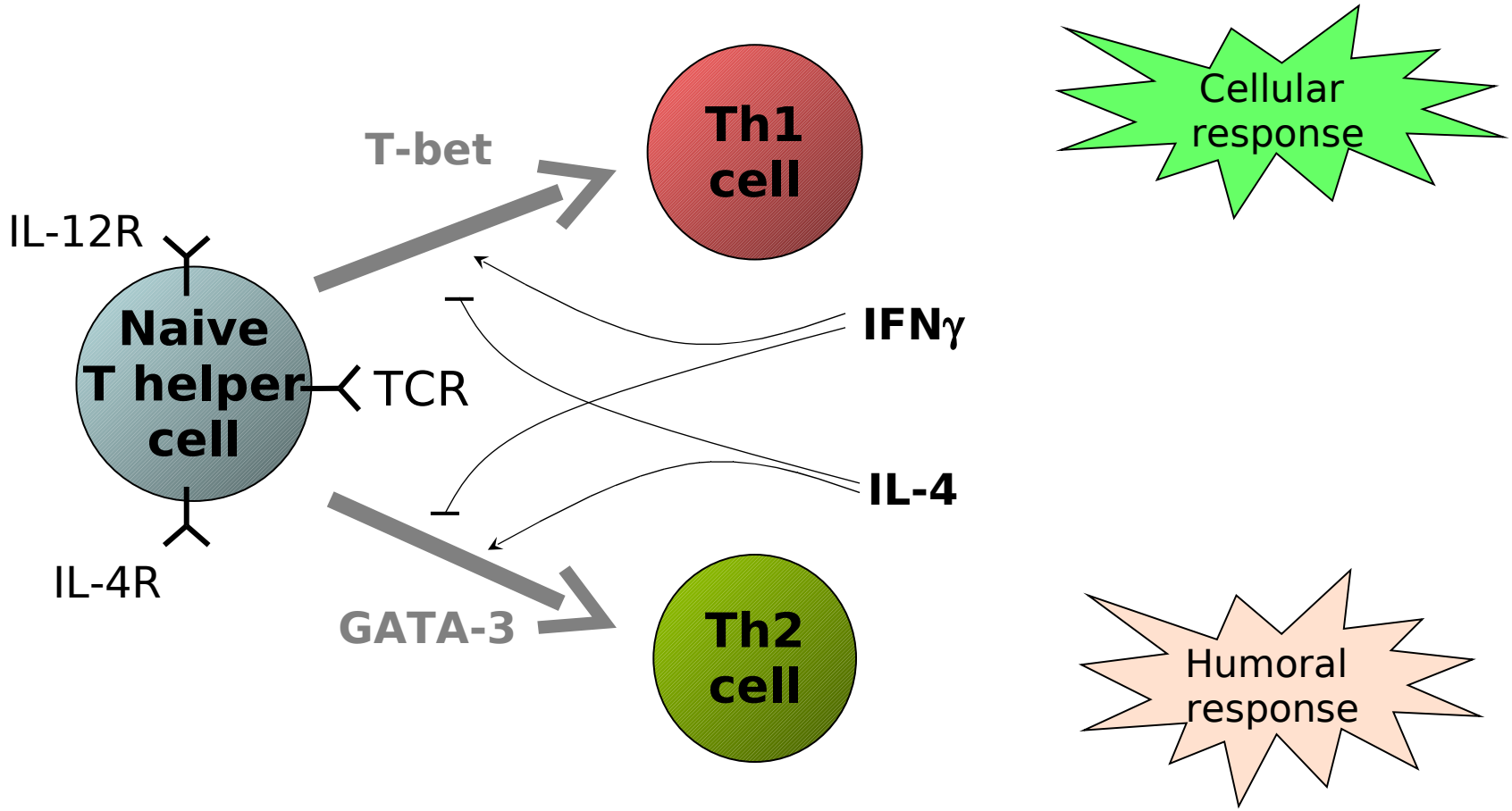
Application: TCR Signalling



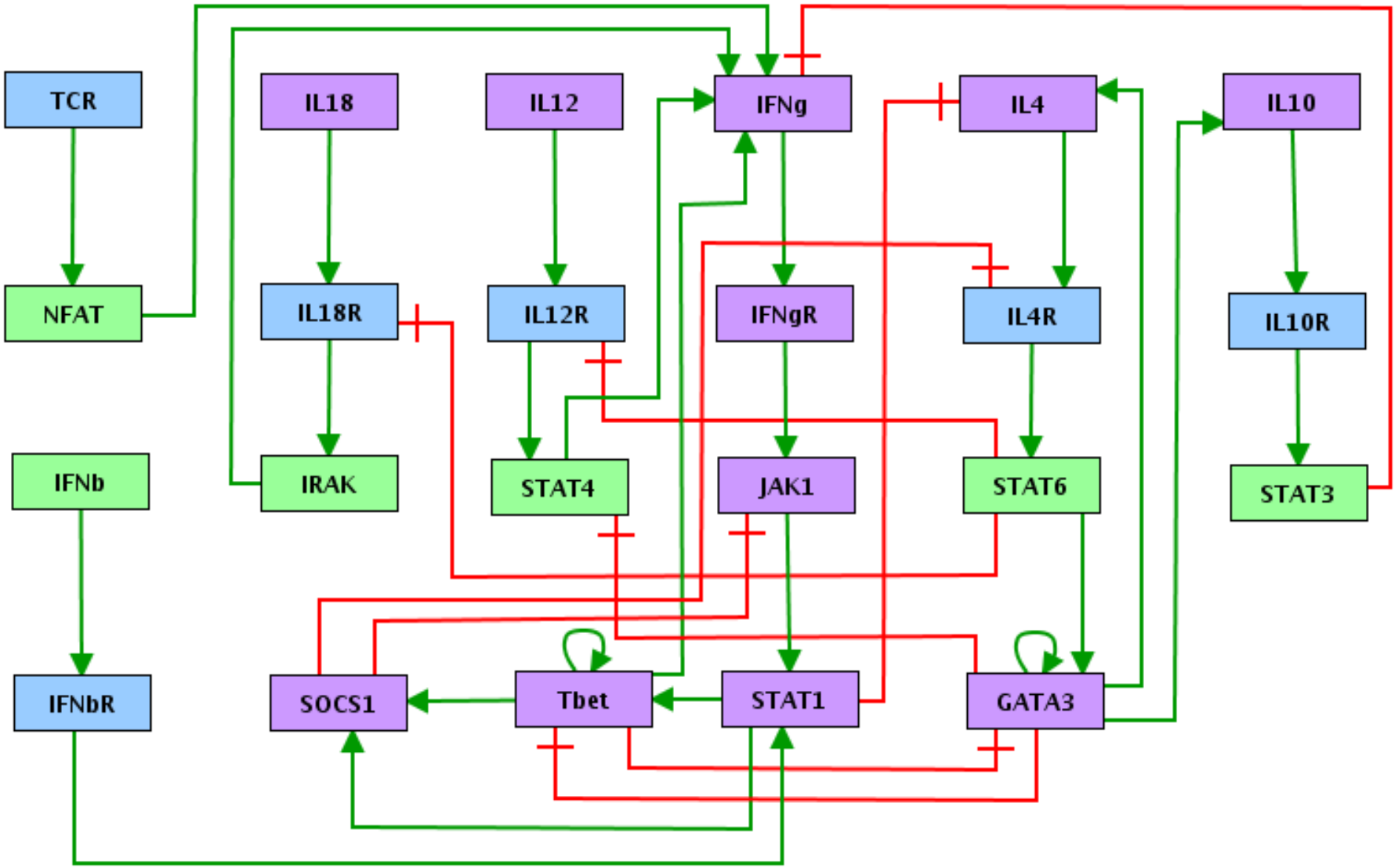


S. Klamt (MPI Magdeburg)

Application: Th Differentiation



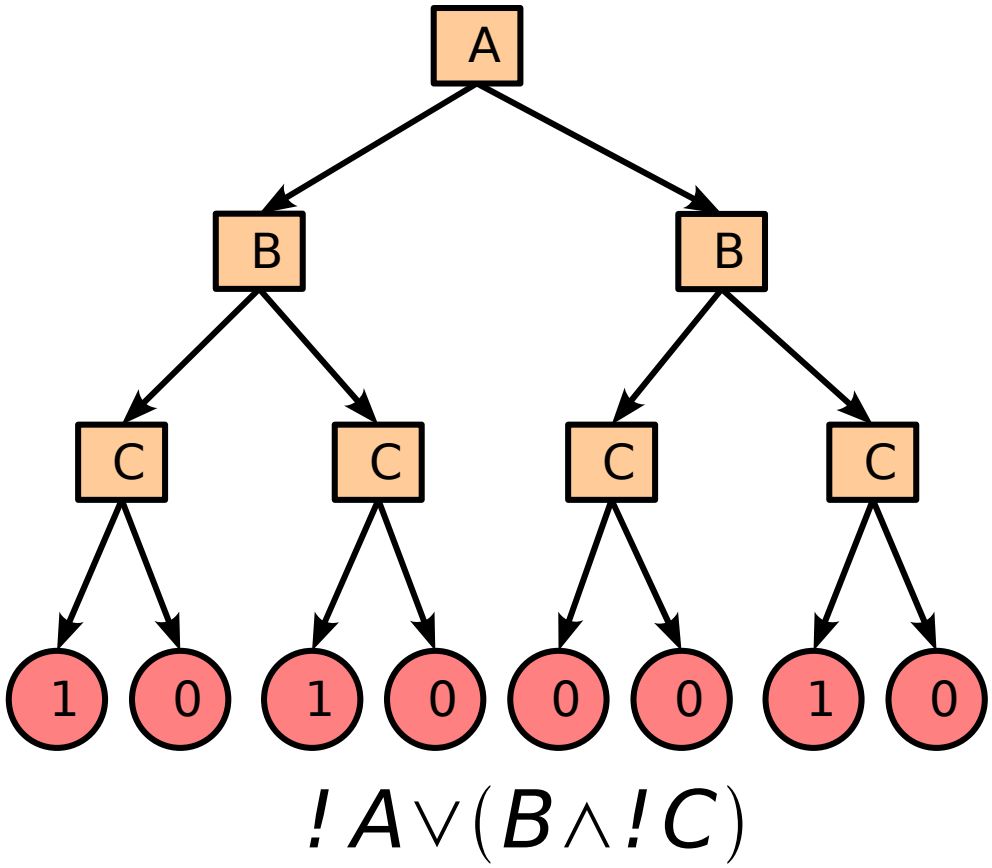
Application: Th Differentiation



L. Mendoza (UNAM, Mexico)

Representation of Logical Functions

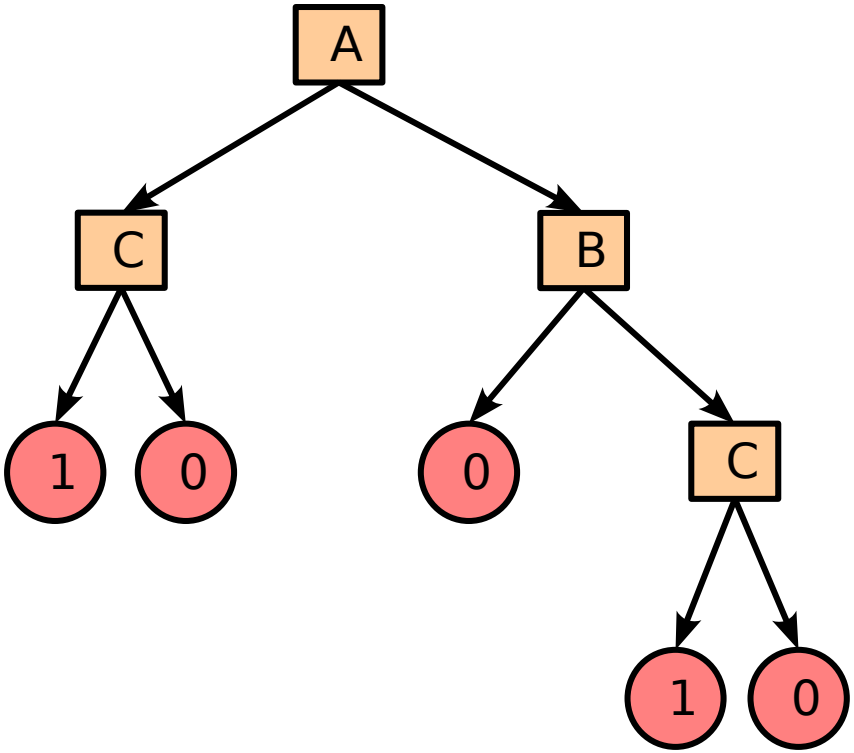
Decision tree



$$!A \vee (B \wedge !C)$$

Representation of Logical Functions

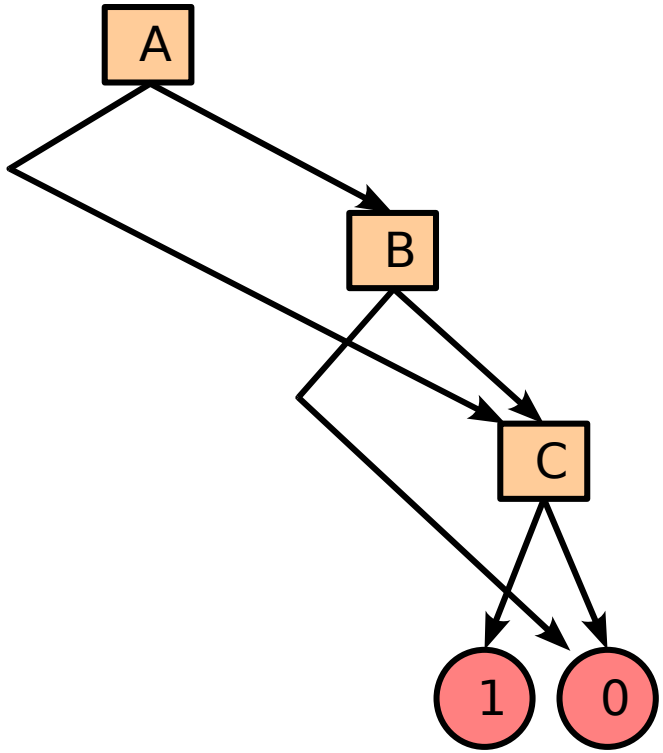
Decision tree



$$!A \vee (B \wedge !C)$$

Representation of Logical Functions

Decision diagram



$$!A \vee (B \wedge !C)$$

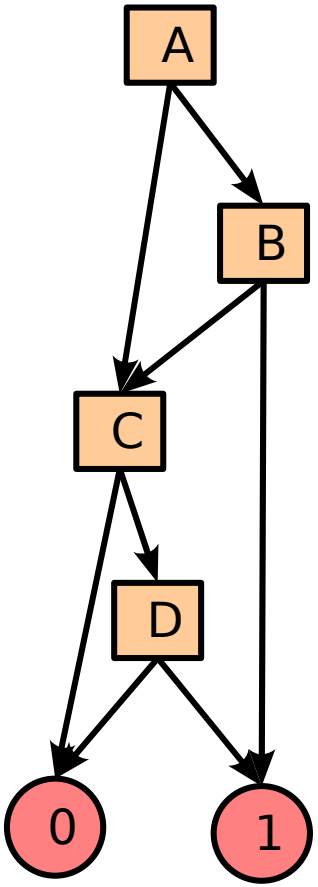
Compact representation

ROMDDs

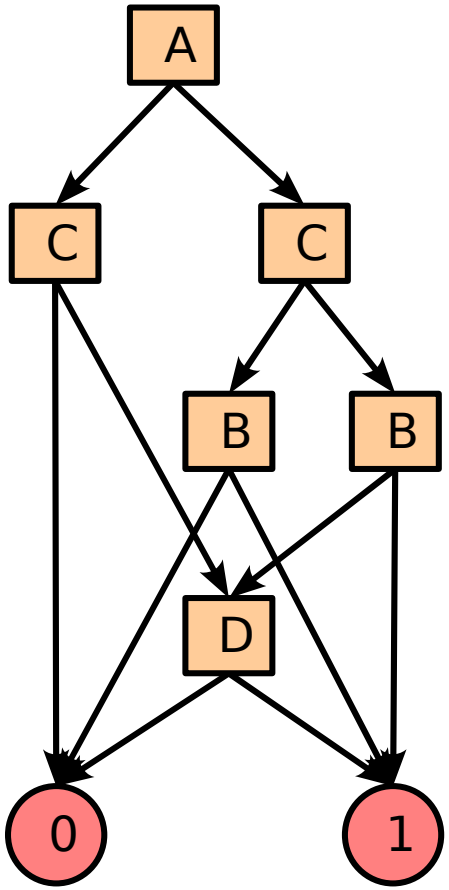
Reduced **O**rdered **M**ultivalued **D**ecision **D**iagrams

- Multivalued
  - Choices on each node: > 2
  - Leaves 0, 1, 2, ...
- Ordered
  - Decision variables are ordered
- Reduced
  - No “useless” node
  - No isomorphic sub-diagrams
    - **Canonical representation**

Impact of the Ordering



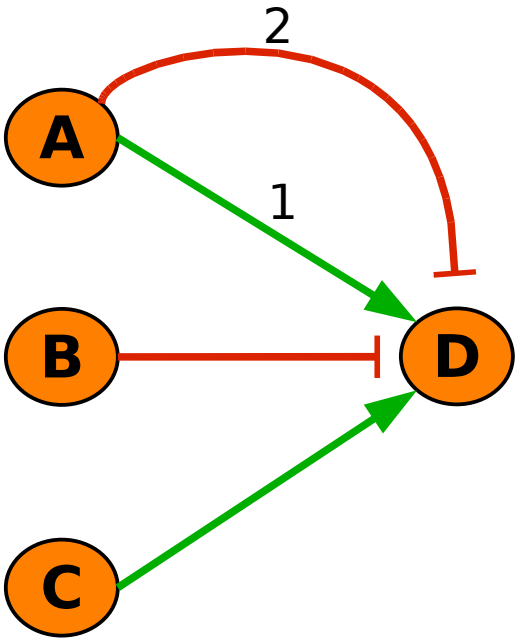
A B C D

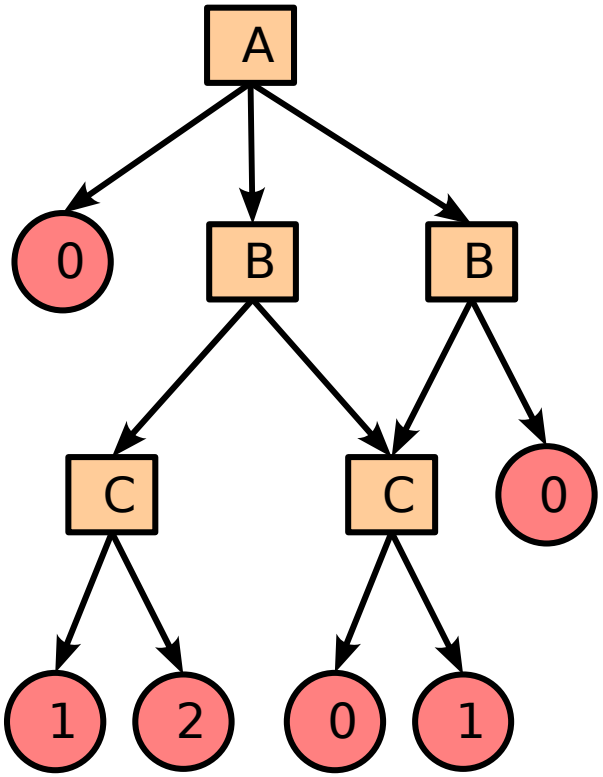


A C B D

$$(A \wedge B) \vee (C \wedge D)$$

Evolution Rules (2)





$$K_D = \begin{cases} 2 \;\; if \;\; (A_1 \wedge !B \wedge C) \\ 1 \;\; if \;\; (A_1 \wedge B \wedge C) \vee (A_1 \wedge !B \wedge !C) \vee (A_2 \wedge !B \wedge C) \\ 0 \;\; otherwise \end{cases}$$

$(i.e.\ if\ A_0 \vee (A_2 \wedge (B \vee !C)) \vee (A_1 \wedge B \wedge !C))$

Use of MDD in GINsim

GINsim: User friendly modelling software

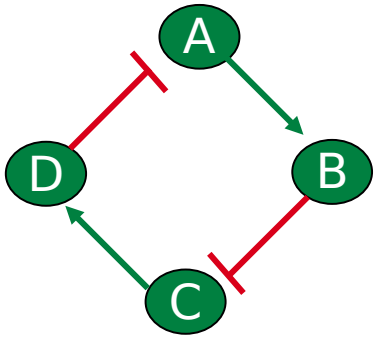
- Regulatory graph definition
- Simulation and analysis
- Freely available: http://gin.univ-mrs.fr/GINsim

Use of MDD leads to several improvements

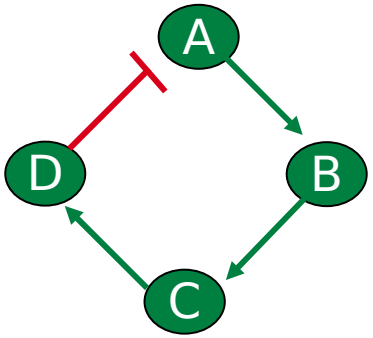
- Faster simulation (finding the next state)
- New exports
  - Petri net
  - Model checker (SMV)
- **Analysis Algorithms**

Role of Regulatory Circuits

Positive circuit

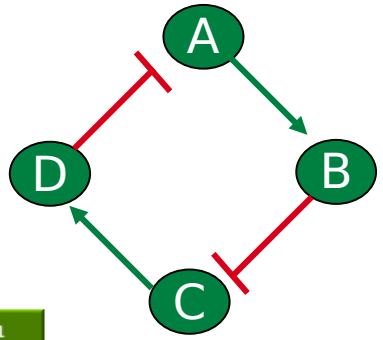


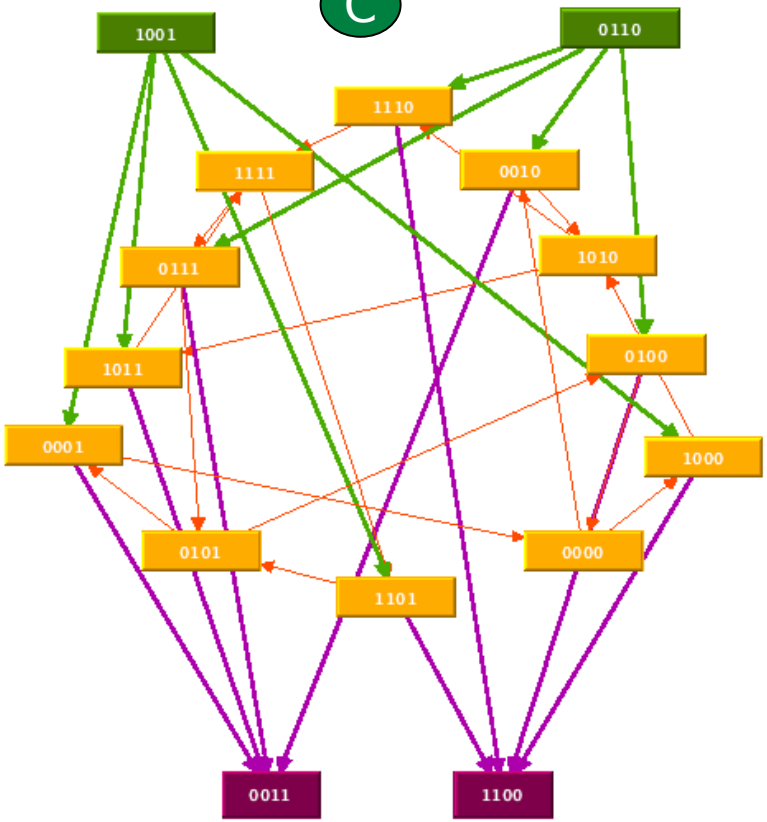
Negative circuit



Role of Regulatory Circuits

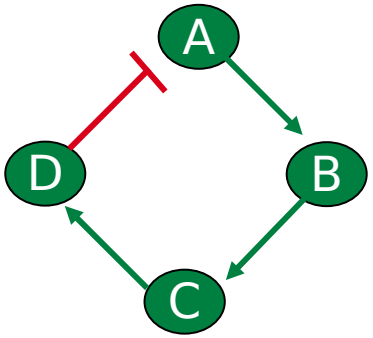
Positive circuit

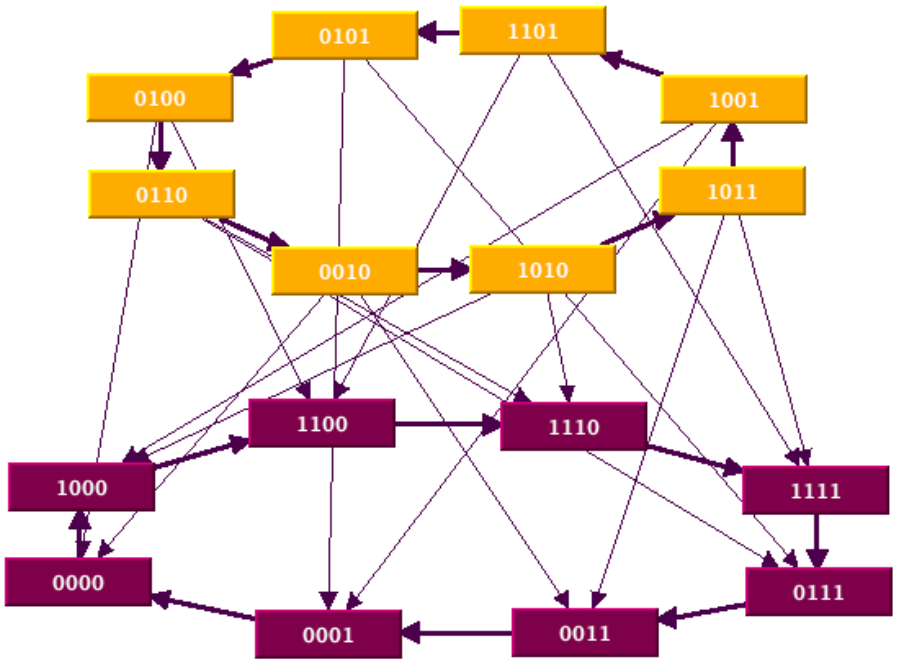




Stable states

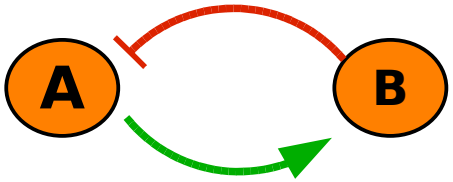
Negative circuit

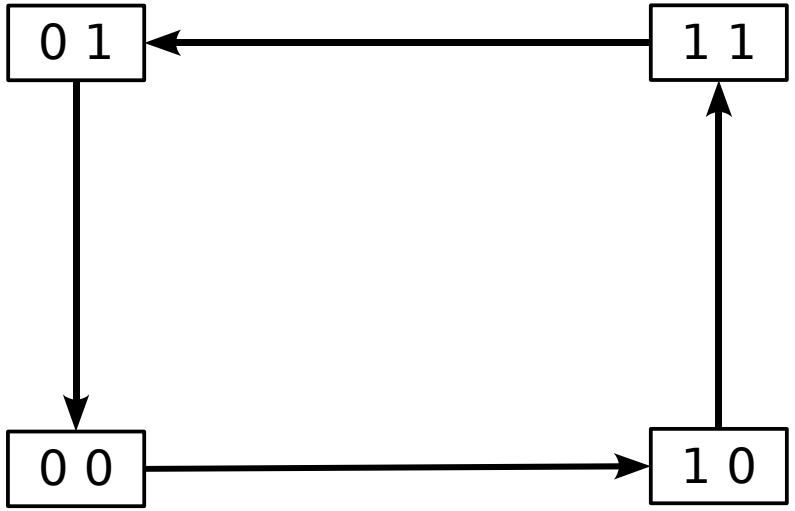




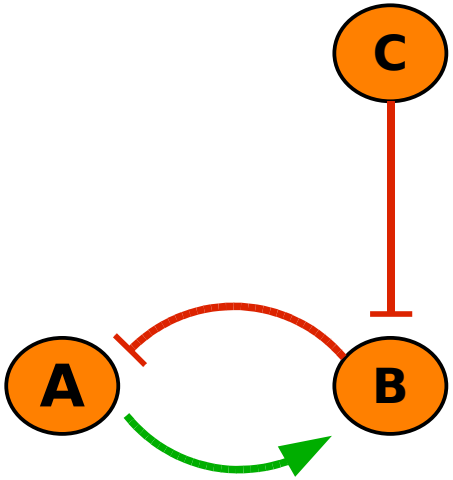
Cyclic attractor

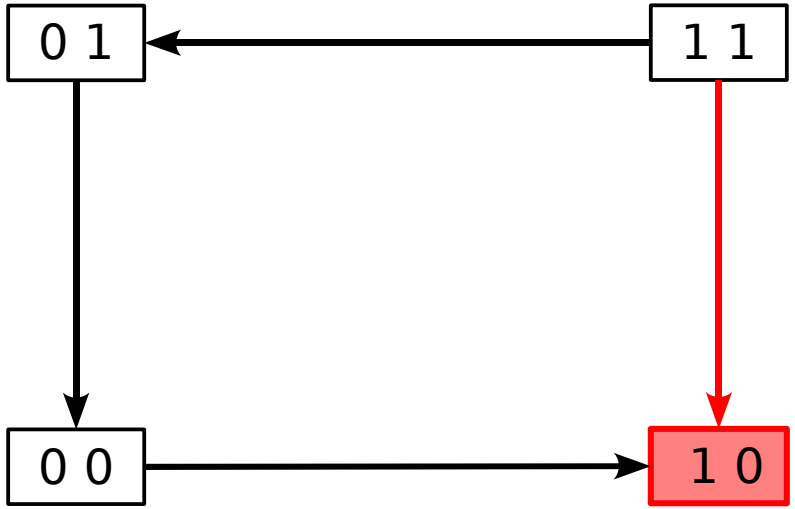
Functionality Context





Functionality Context





C prevents A from activating B

The circuit is **functional** in a given **context**: in absence of C

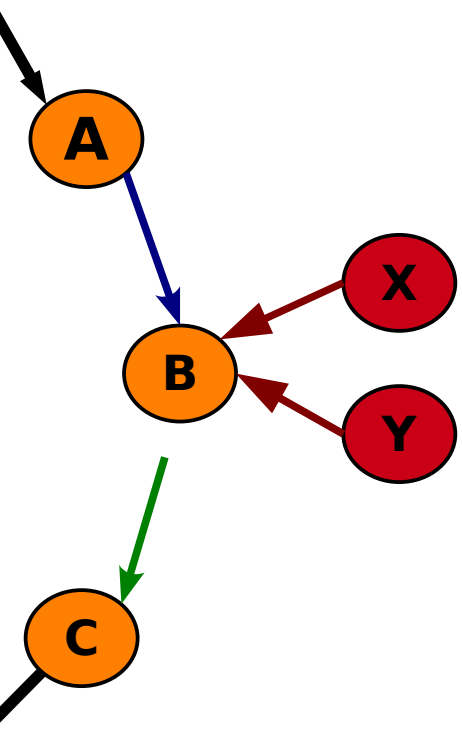
Functionality Context

Context: set of constraints on the expression levels of regulators

Each interaction has its own context

Context of the circuit: intersection of these of its interactions

Functionality of an Interaction

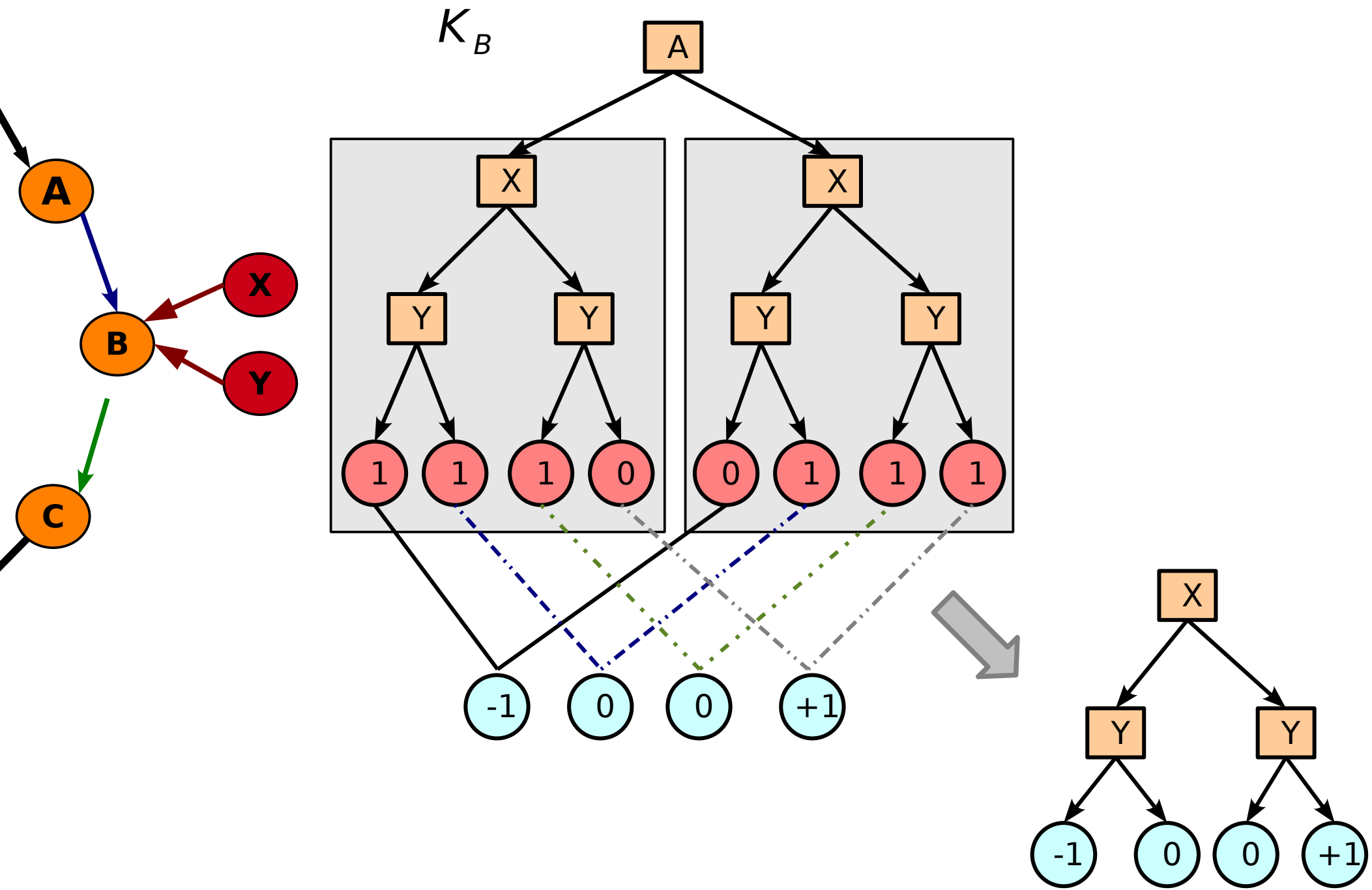


In a circuit *(...,A,B,C,...)*, the functionality of *(A,B)* depends on:

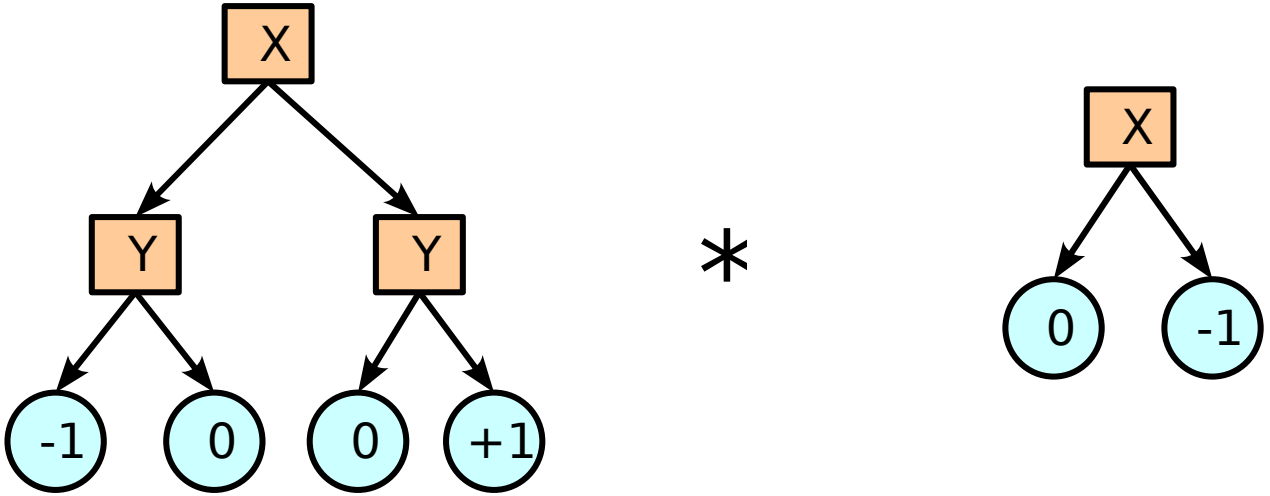
- K_B
- the threshold of *(A,B)*
- the threshold of *(B,C)*

Functionality: logical rule depending on the regulators of B

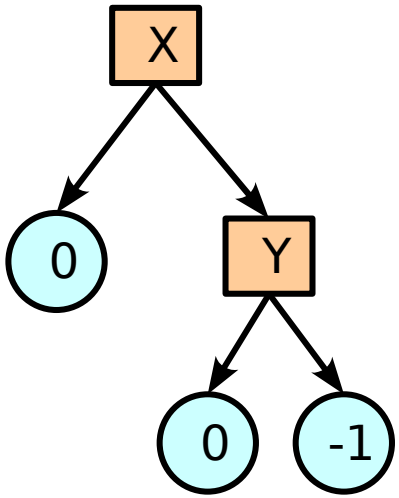
Functionality of an Interaction



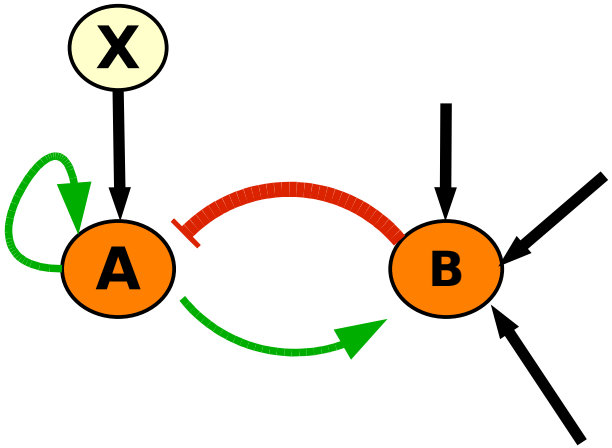
Intersection of Contexts

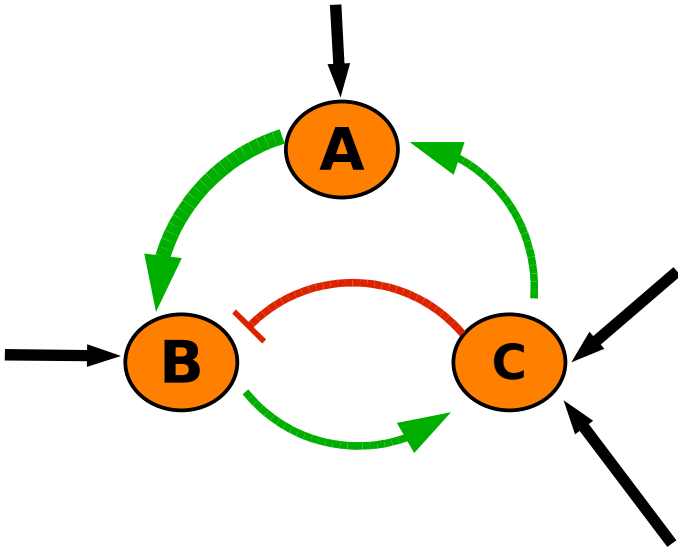


Combination of MDDs



Context Cleanup



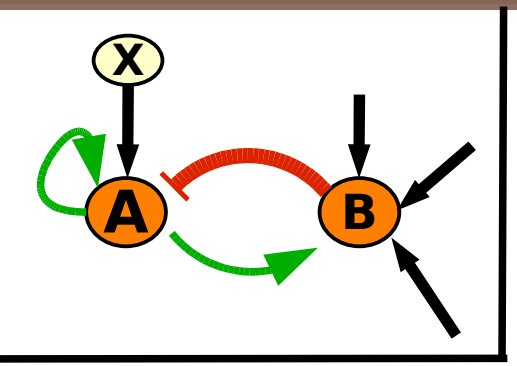


Auto-regulation and “short-circuit”

- Circuit members in functionality context

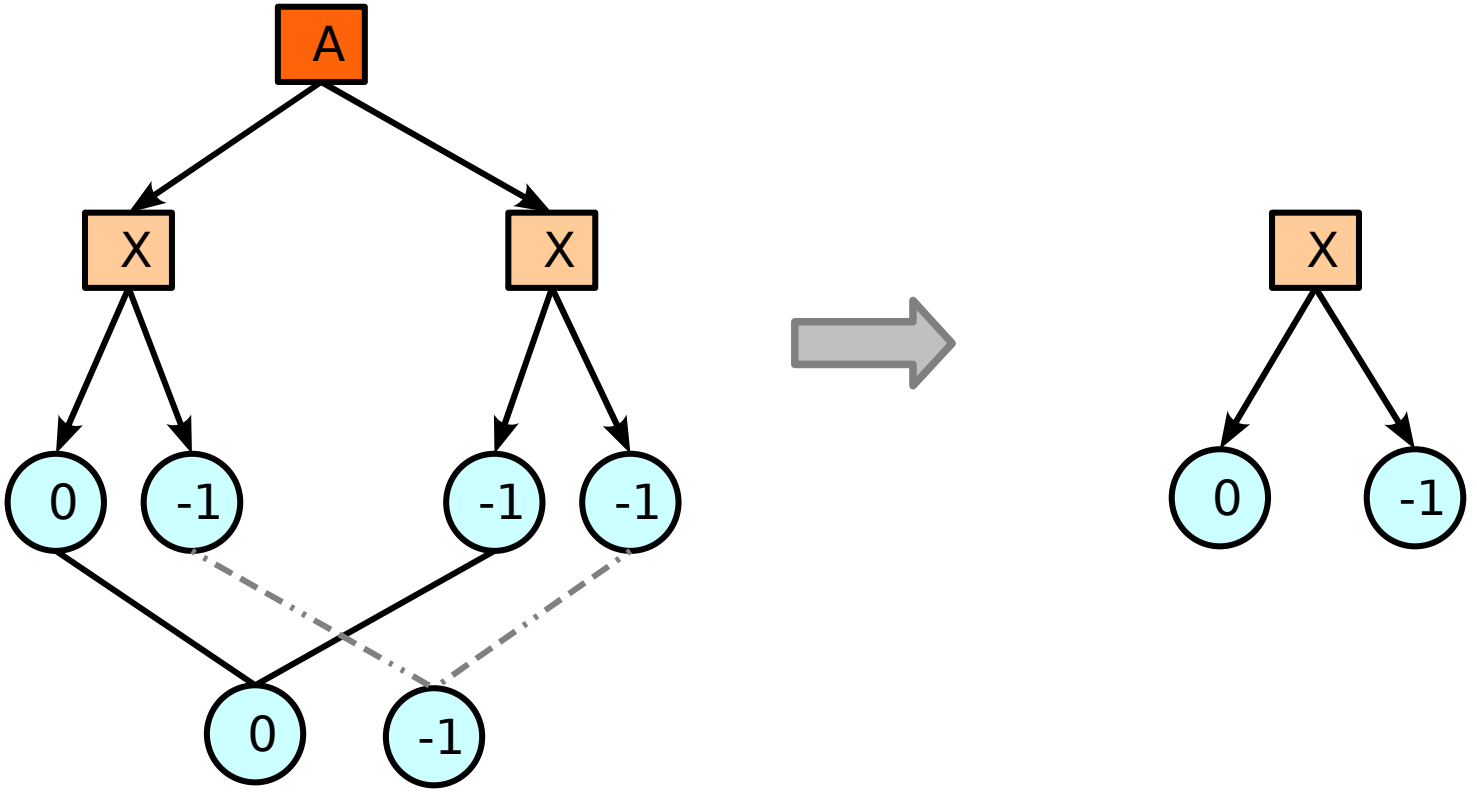
Members of the circuit must be able to cross their threshold

Context Cleanup



Auto-regulation on A

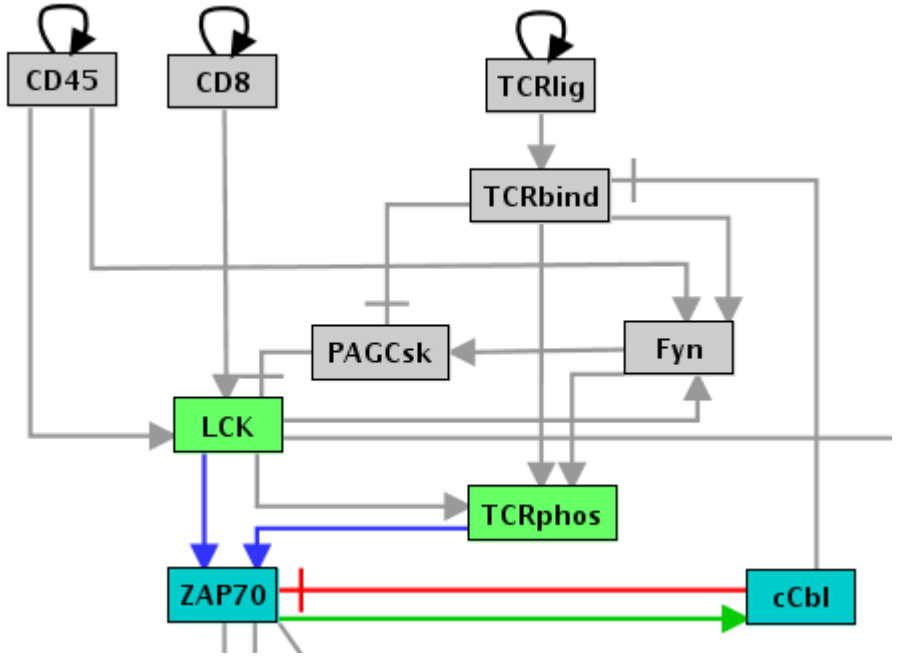
The context of (B,A) introduces a constraint on A



Application: TCR Signalling

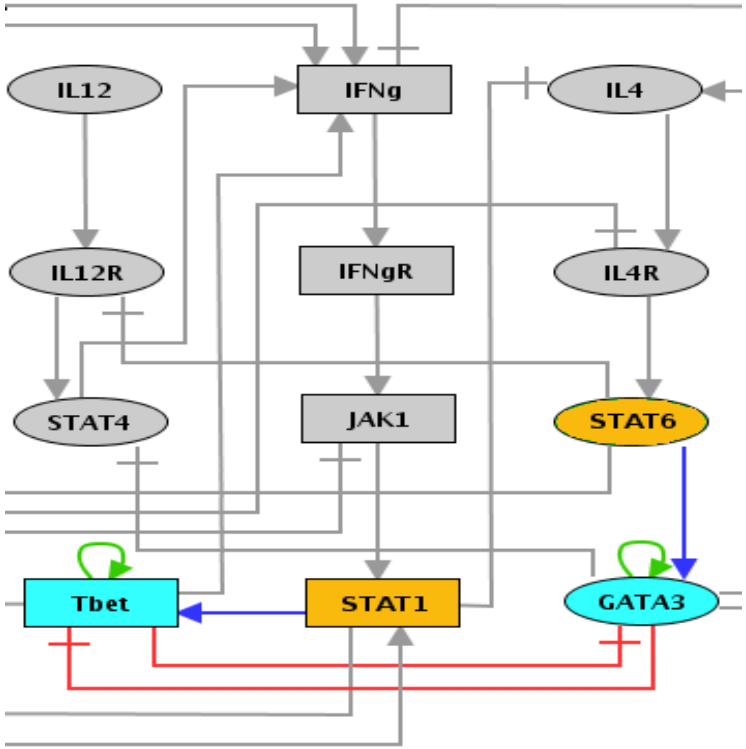
12 circuits – 4 functional:

- 3 positive: auto-regulations on inputs
- 1 negative: ZAP70 cCbl
  LCK TCRphos



Application: Th Differentiation

27 circuits
6 functionals, all of them positives



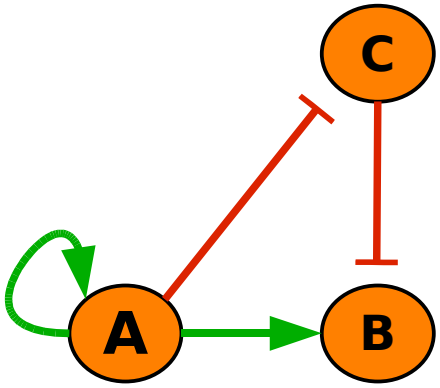
- Tbet
  - GATA3 STAT1
- GATA3
  - Tbet STAT6
- Tbet - GATA3
  - STAT1 STAT6
- 3 circuits including IFNg IFNgR JAK1 STAT1
  - NFAT Tbet IFNbR STAT3

Determination of Stable States

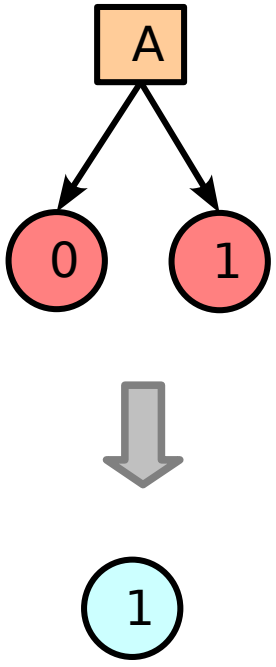
Analytic method to find stable states

- ALL possible stables states, no simulation
- Stable state: all variables are stable
  - Stability condition for each variable
  - Combine these partial conditions

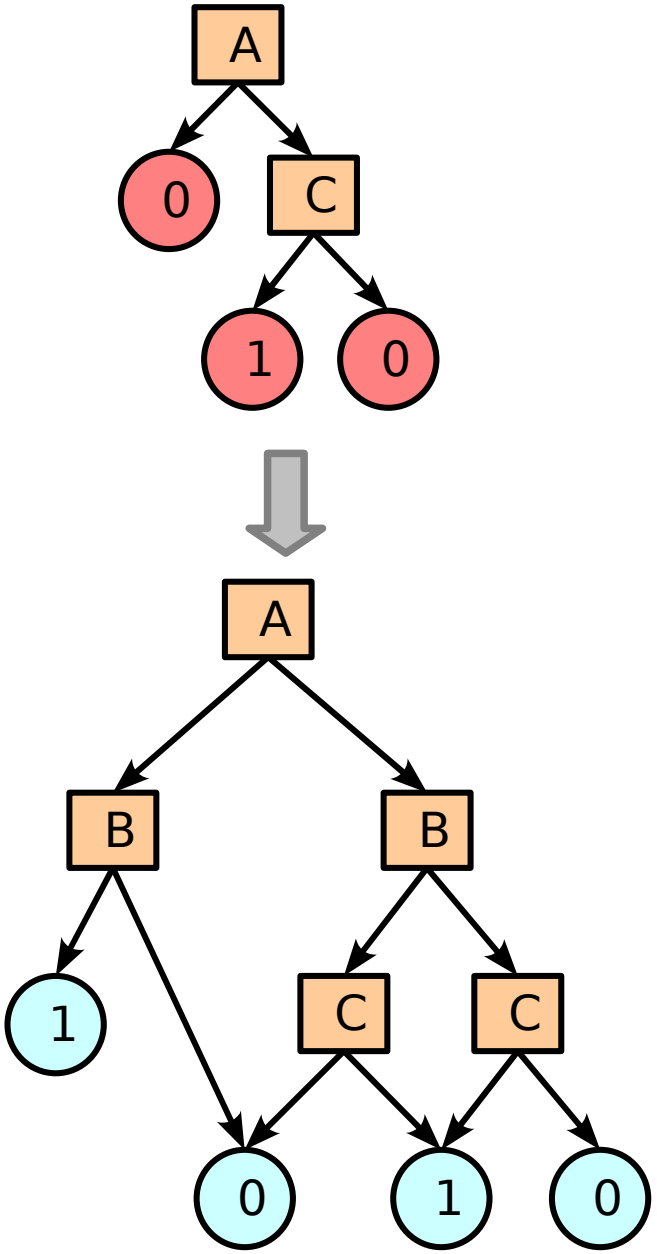
Determination of Stable States



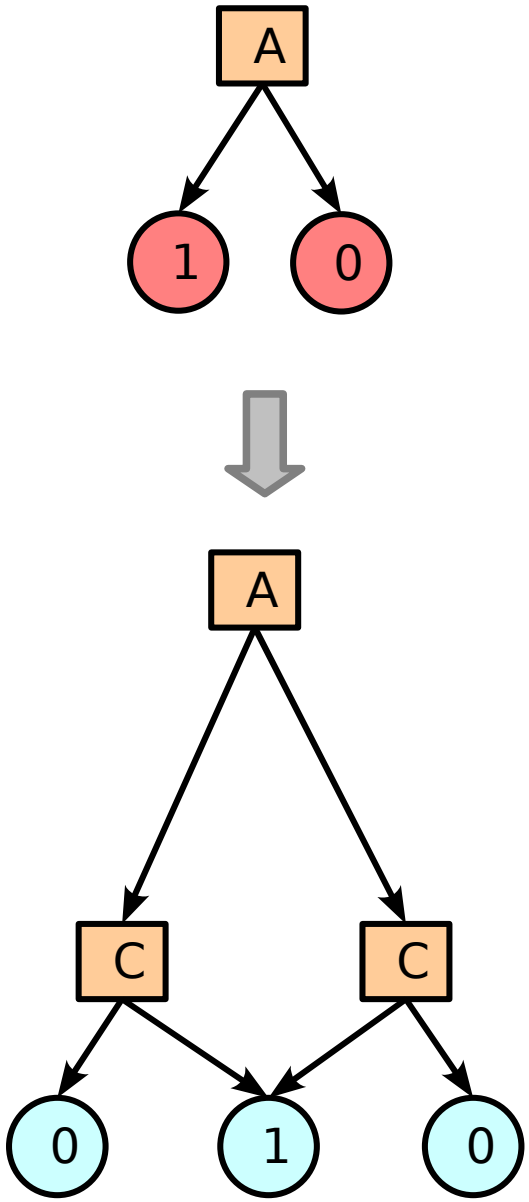
K_A A



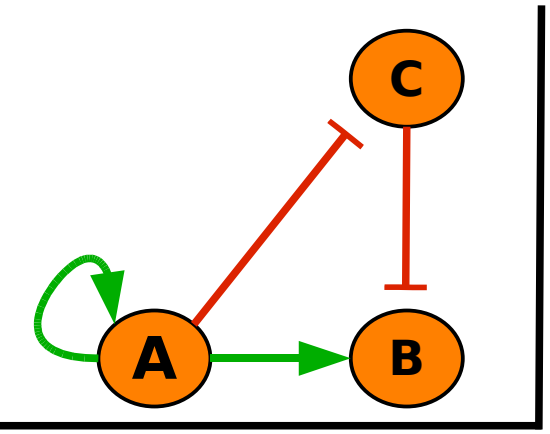
K_B $A \wedge !C$

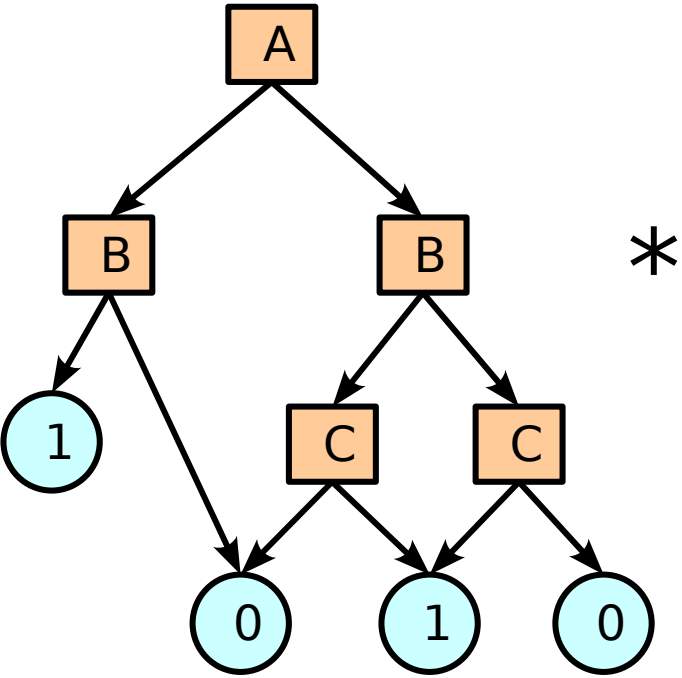


K_C $!A$

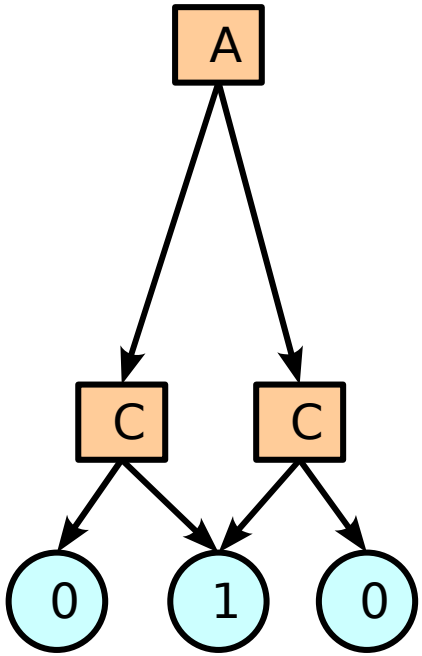


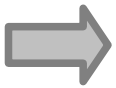
Determination of Stable States

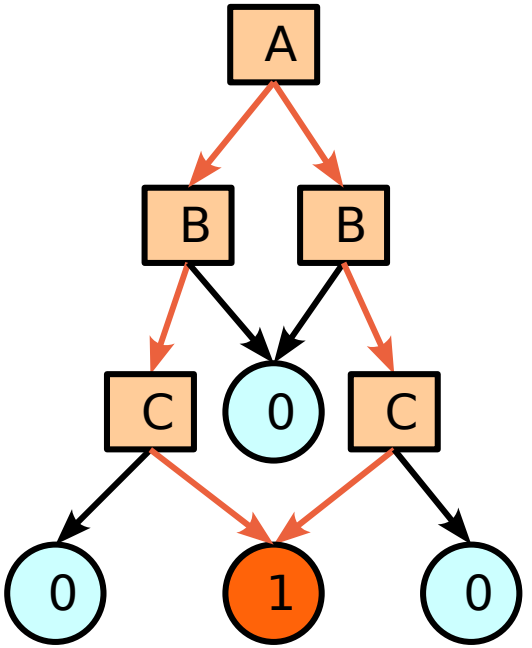




*

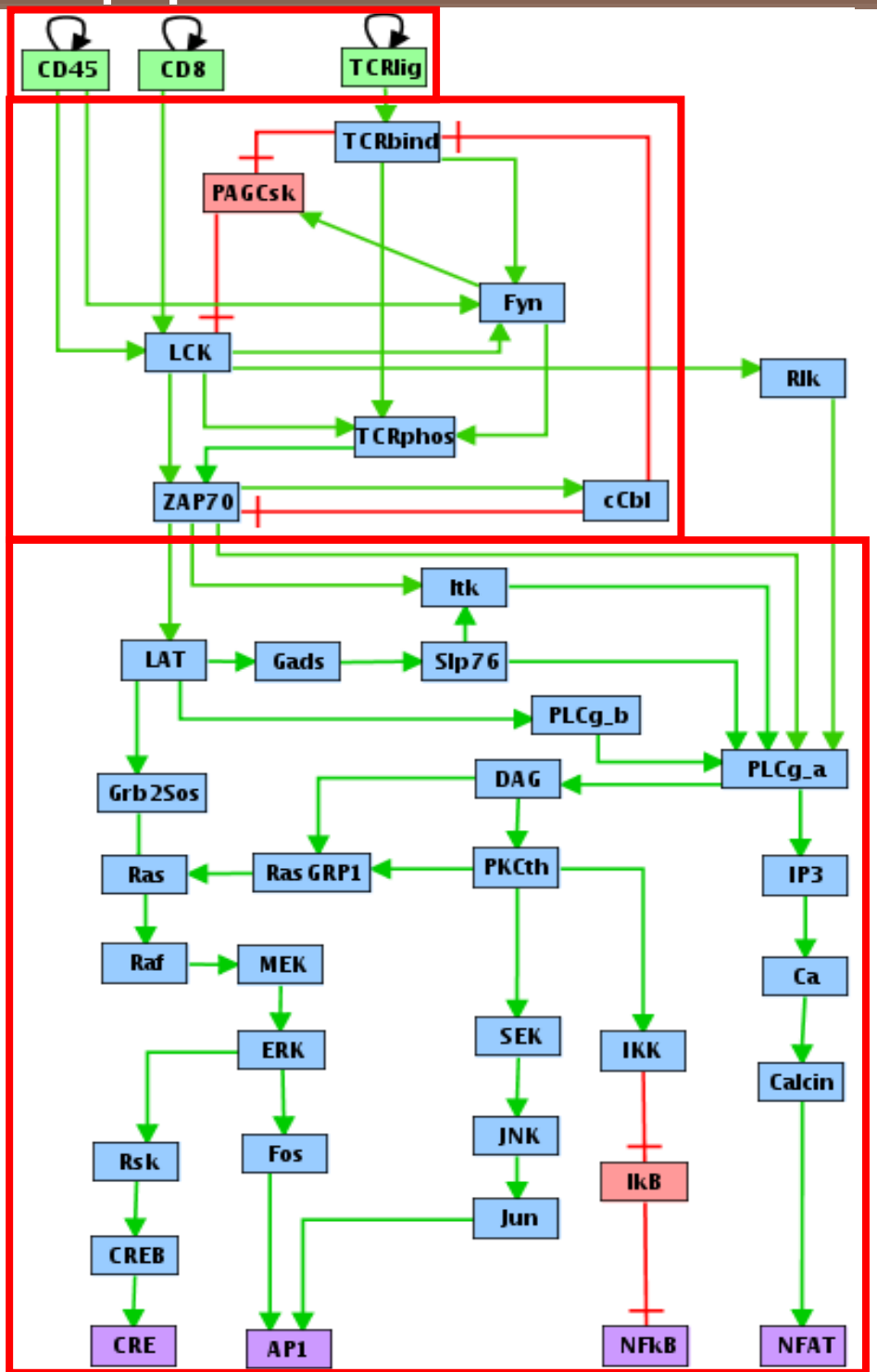






2 stable states : 001 et 110

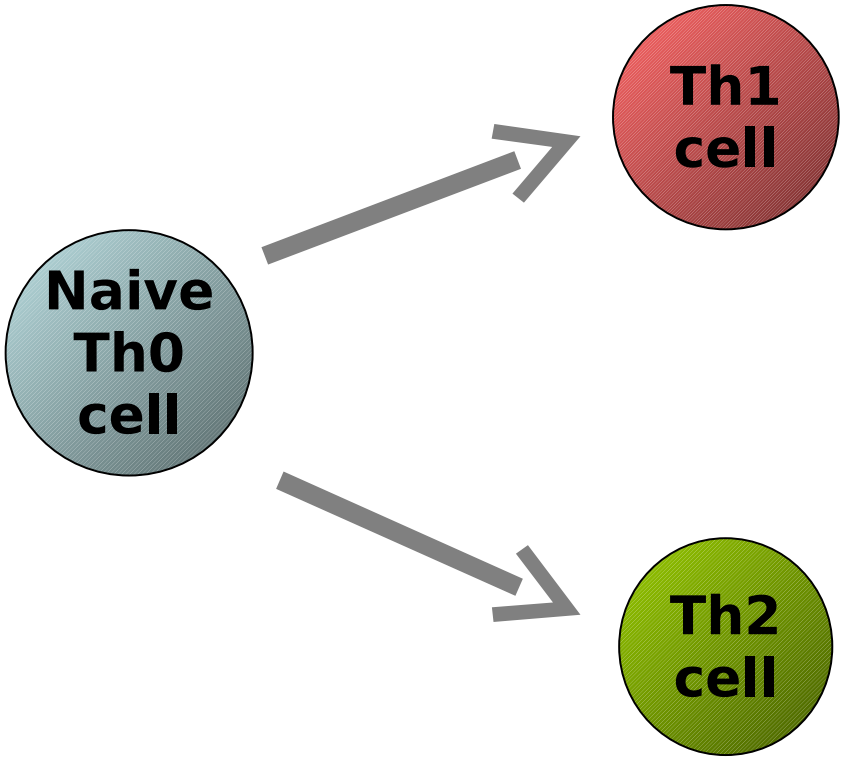
Application: TCR Signalling



TCR signalling:

- 7 stable states: 1 for each input combination except “111”
- For all of them, the signalling cascade is off

Application: Th Differentiation

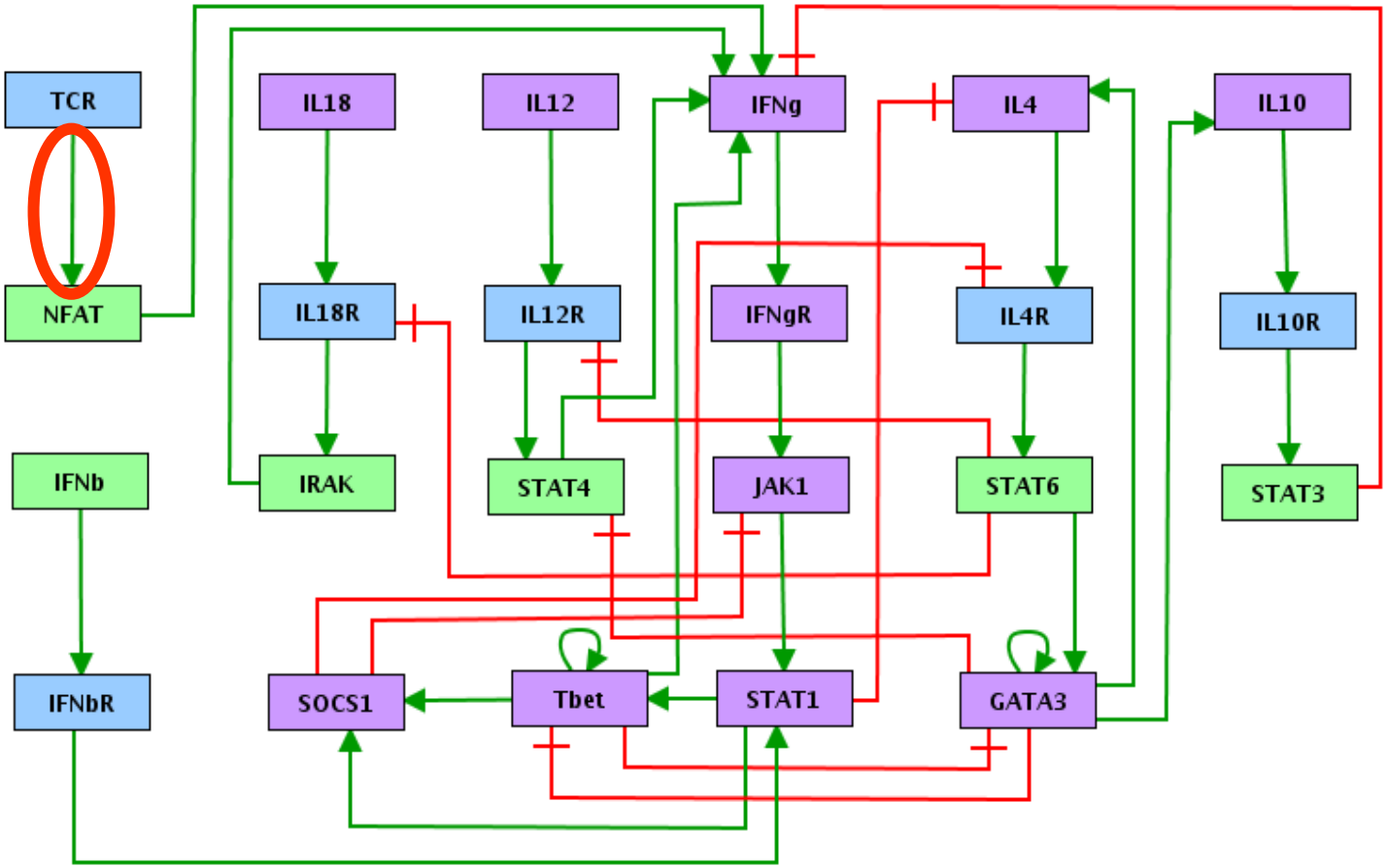


Differentiation: 4 stable states

- Th0: naïve
- Th1 and Th1* (higher expression levels)
- Th2

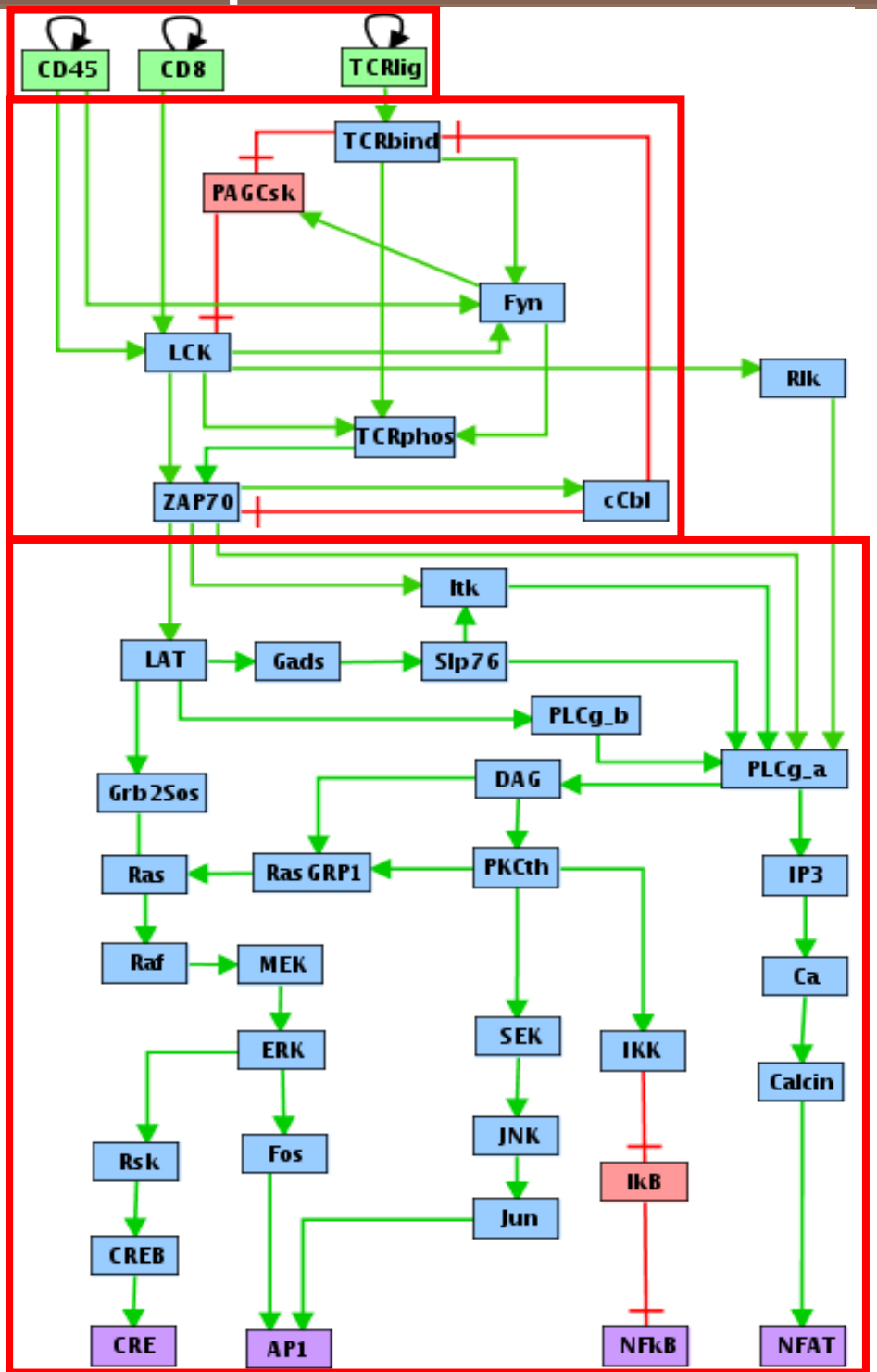
Model Composition

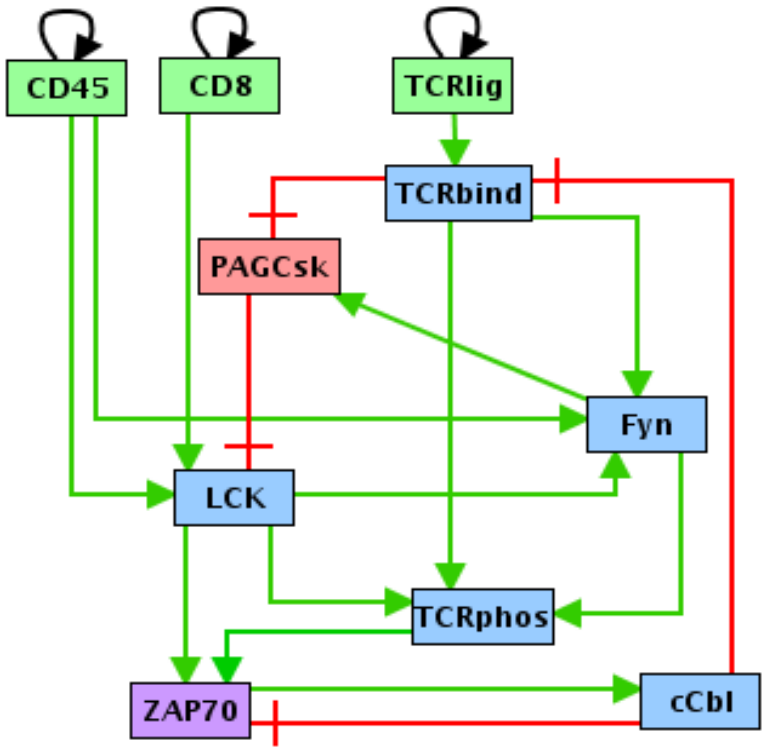
Coupling the resulting models



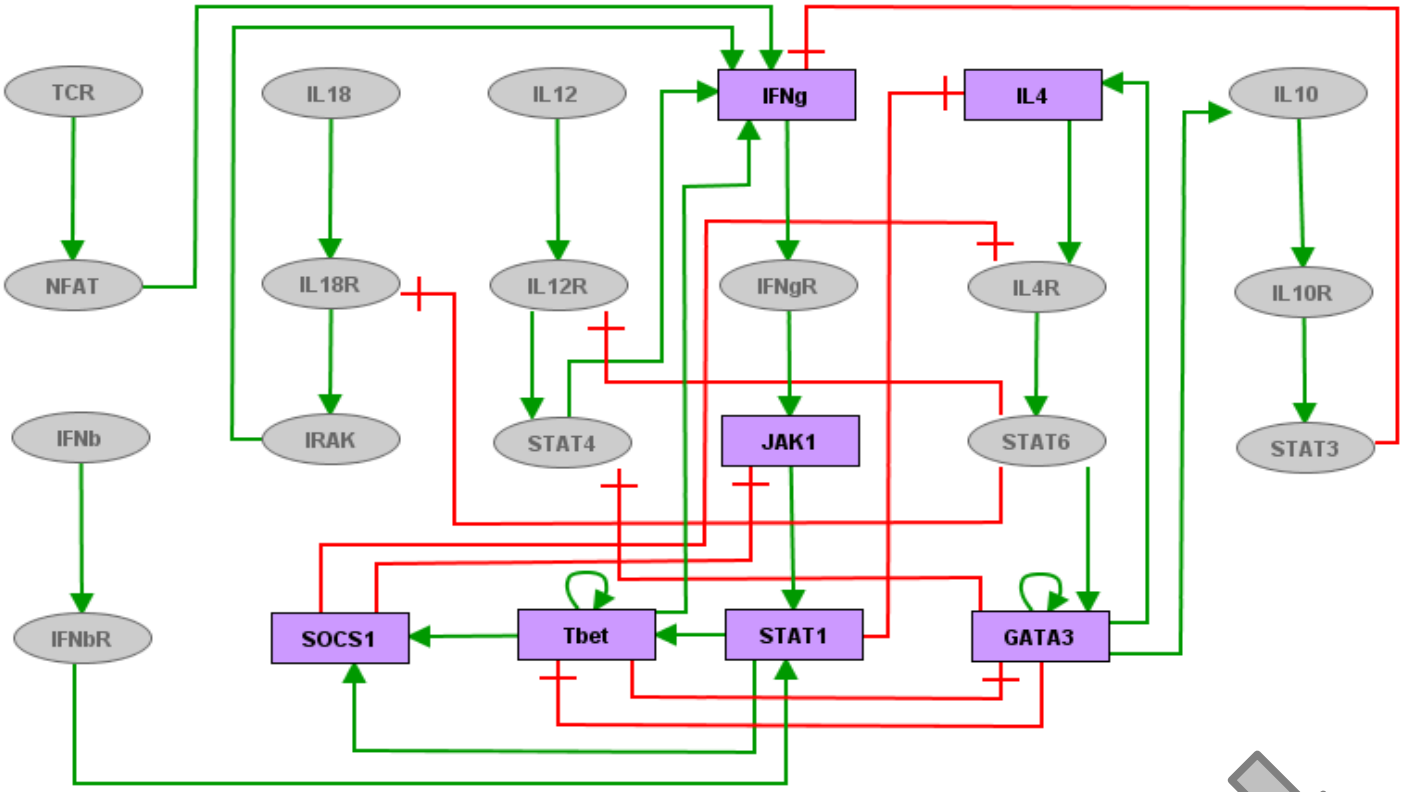
The two models are simplified beforehand

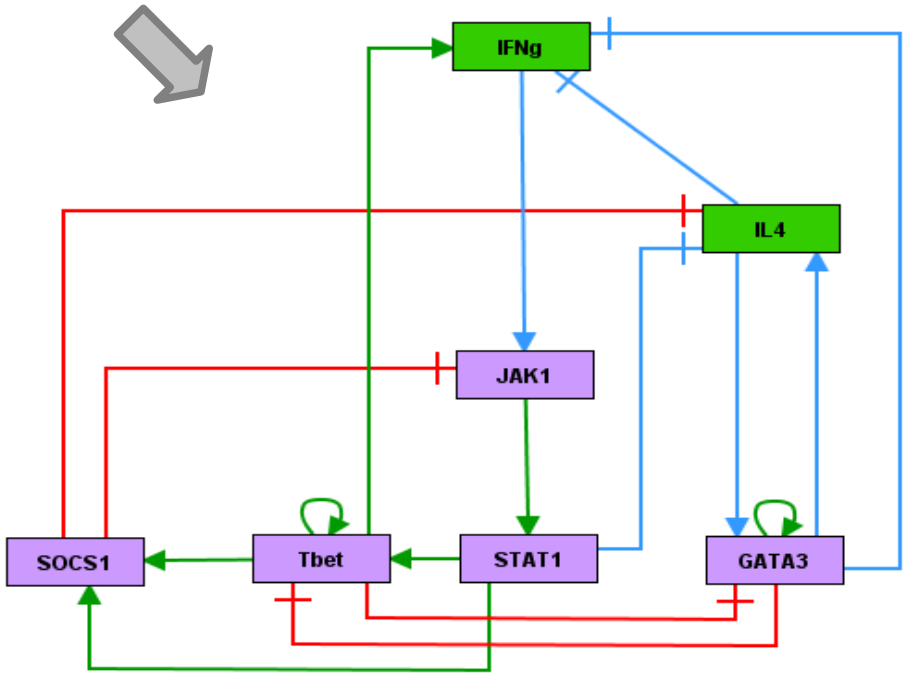
Simplification: TCR Signalling



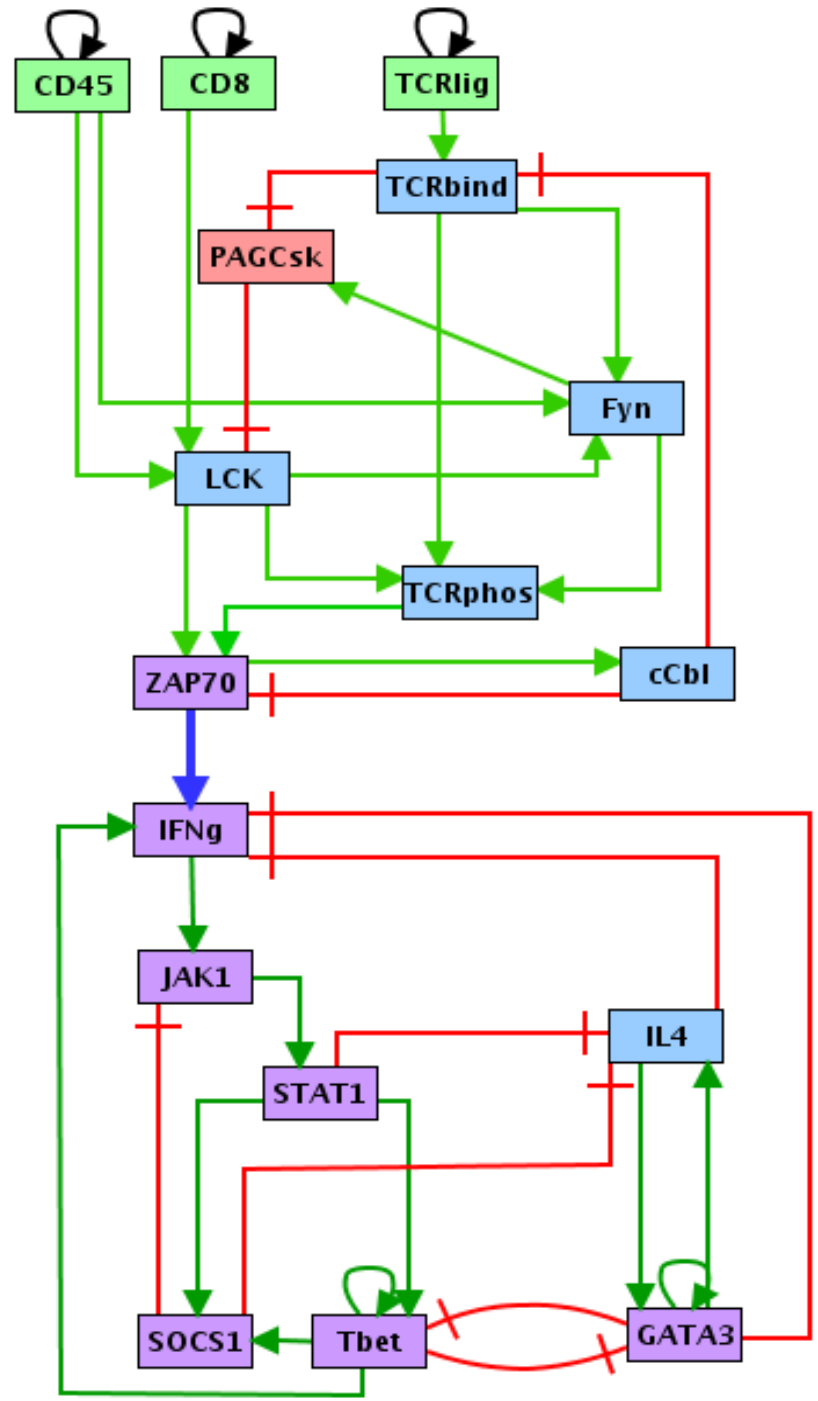


Simplification: Th Differentiation





Coupling the models



Conclusion

- Discrete modelling formalism
  - Dynamics: logical functions
- Decision diagrams (ROMDD)
  - Improve the simulation in GINsim
- Circuit analysis
  - Determine the “core” of a model
- Finding stable states

Prospects

- Decision variables ordering
  - Analysis requires a unique ordering
- Further elaboration of circuit analysis
  - Effect of combinations of circuits
- Extension of the Th model
  - Coupling of activation/differentiation is subtler
  - Extending to Treg (foxp3)
  - Link with cell cycle
- Automated Model composition/simplification