PART I - TALK 2

PN-BASED ANALYSIS OF BIOCHEMICAL NETWORKS

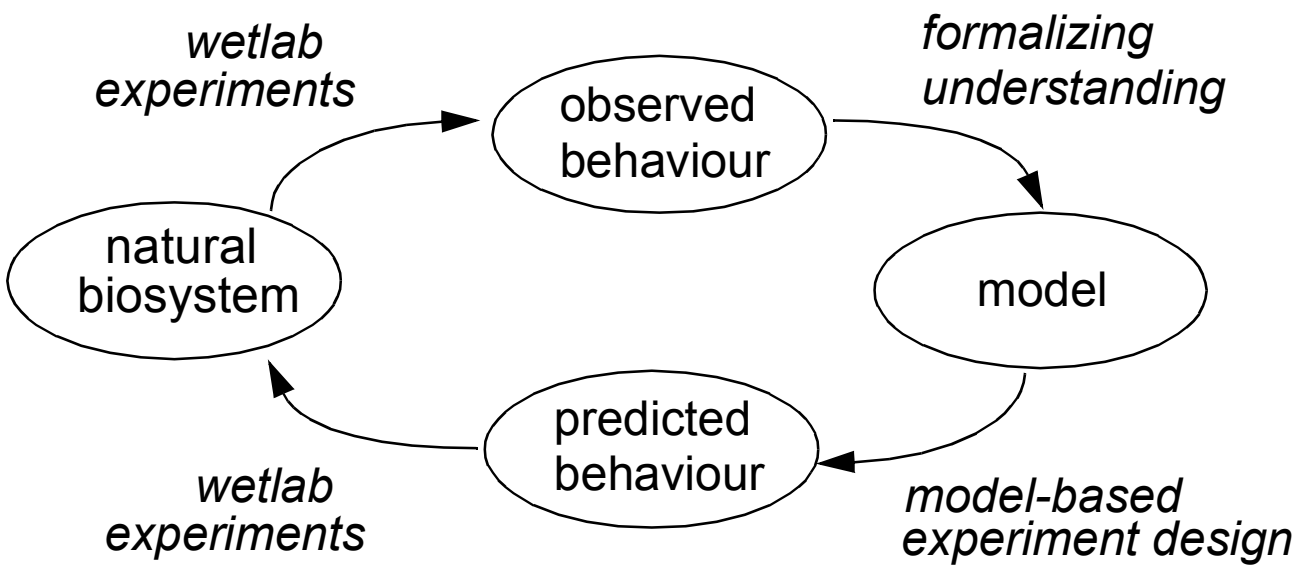
Monika Heiner

Brunel University, on sabbatical leave from

Brandenburg University of Technology Cottbus

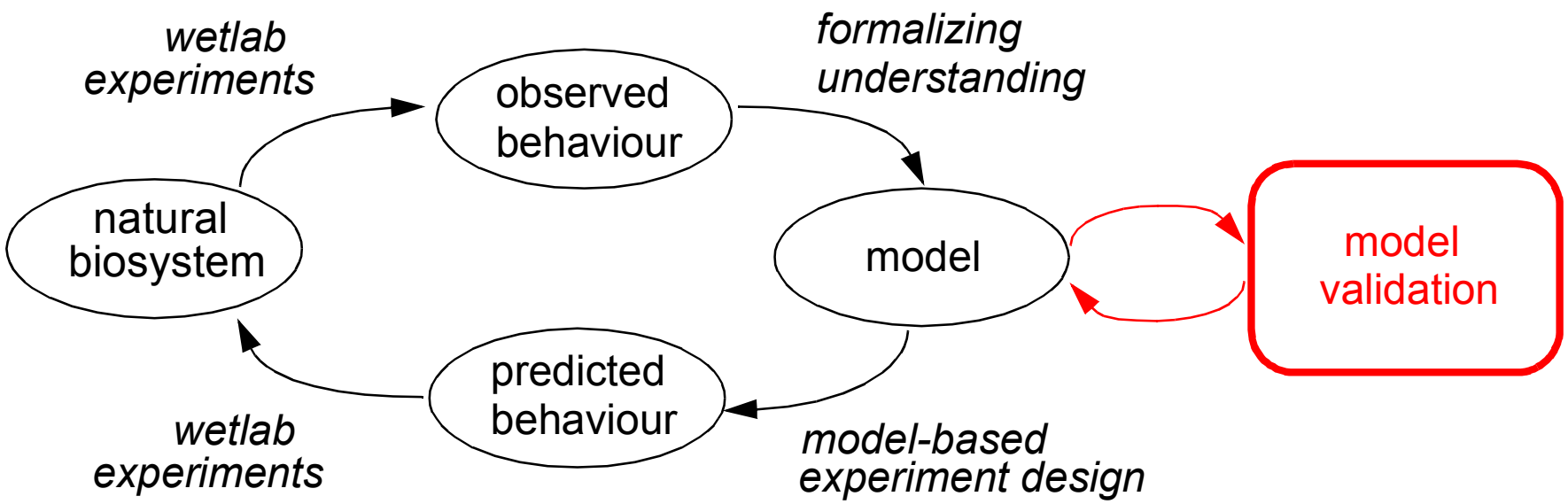
FRAMEWORK: SYSTEMS BIOLOGY

MODELLING = FORMAL KNOWLEDGE REPRESENTATION



FRAMEWORK: SYSTEMS BIOLOGY

MODELLING = FORMAL KNOWLEDGE REPRESENTATION

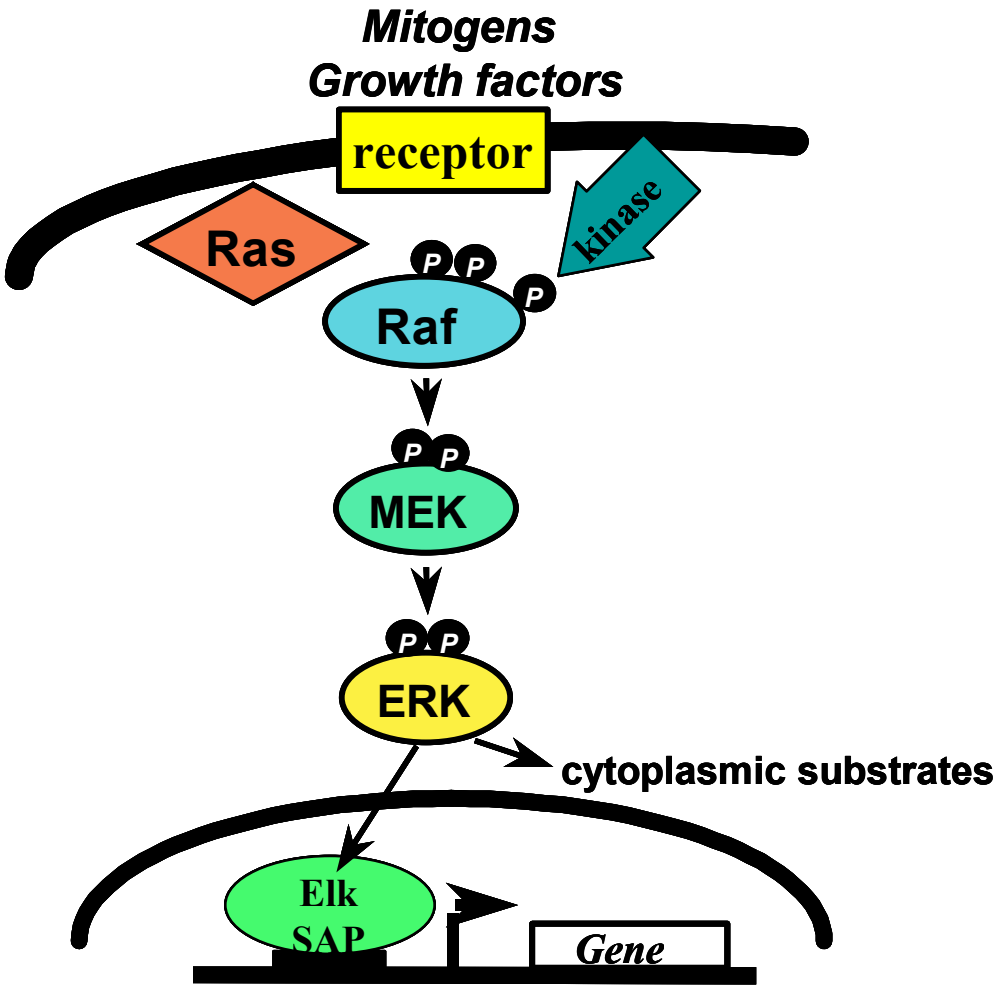


MODEL VALIDATION = CONFIDENCE INCREASE

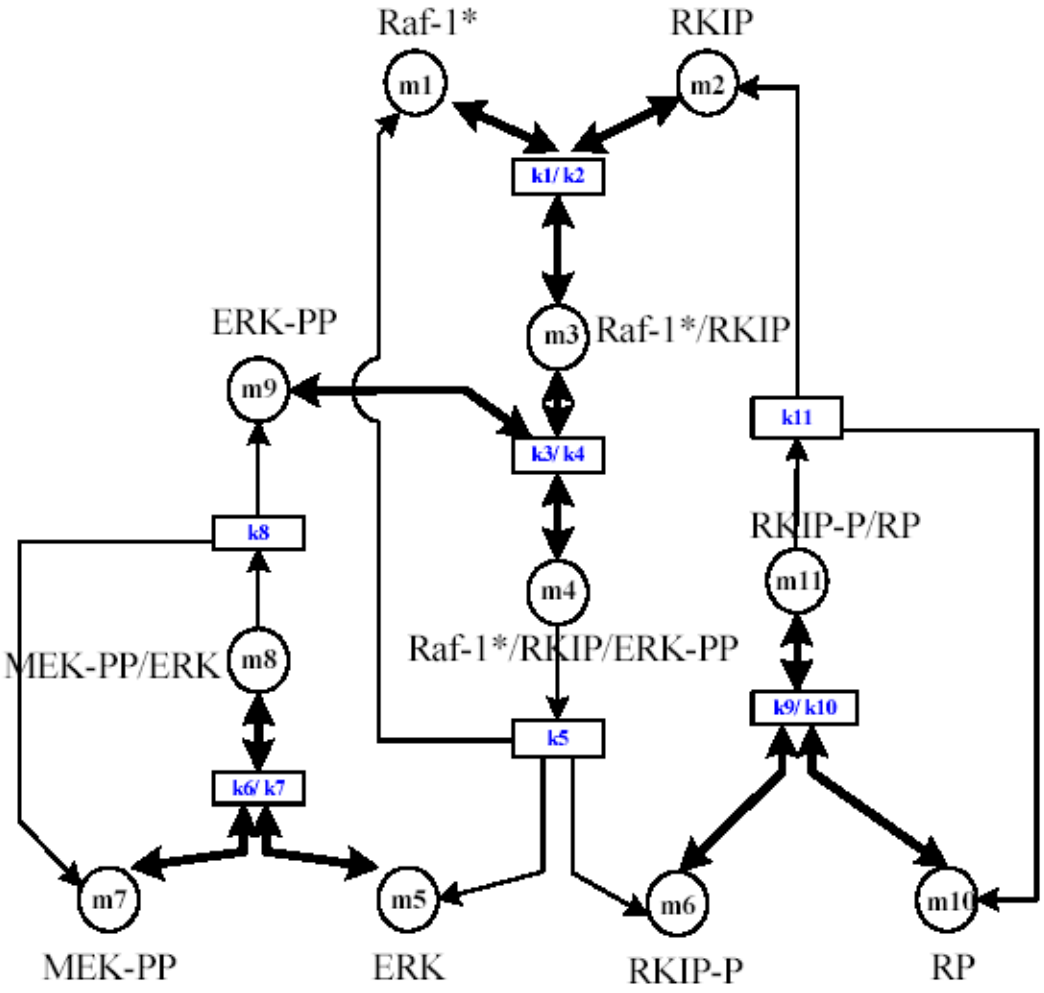
A CASE STUDY

RKIP PATHWAY

…one pathway…

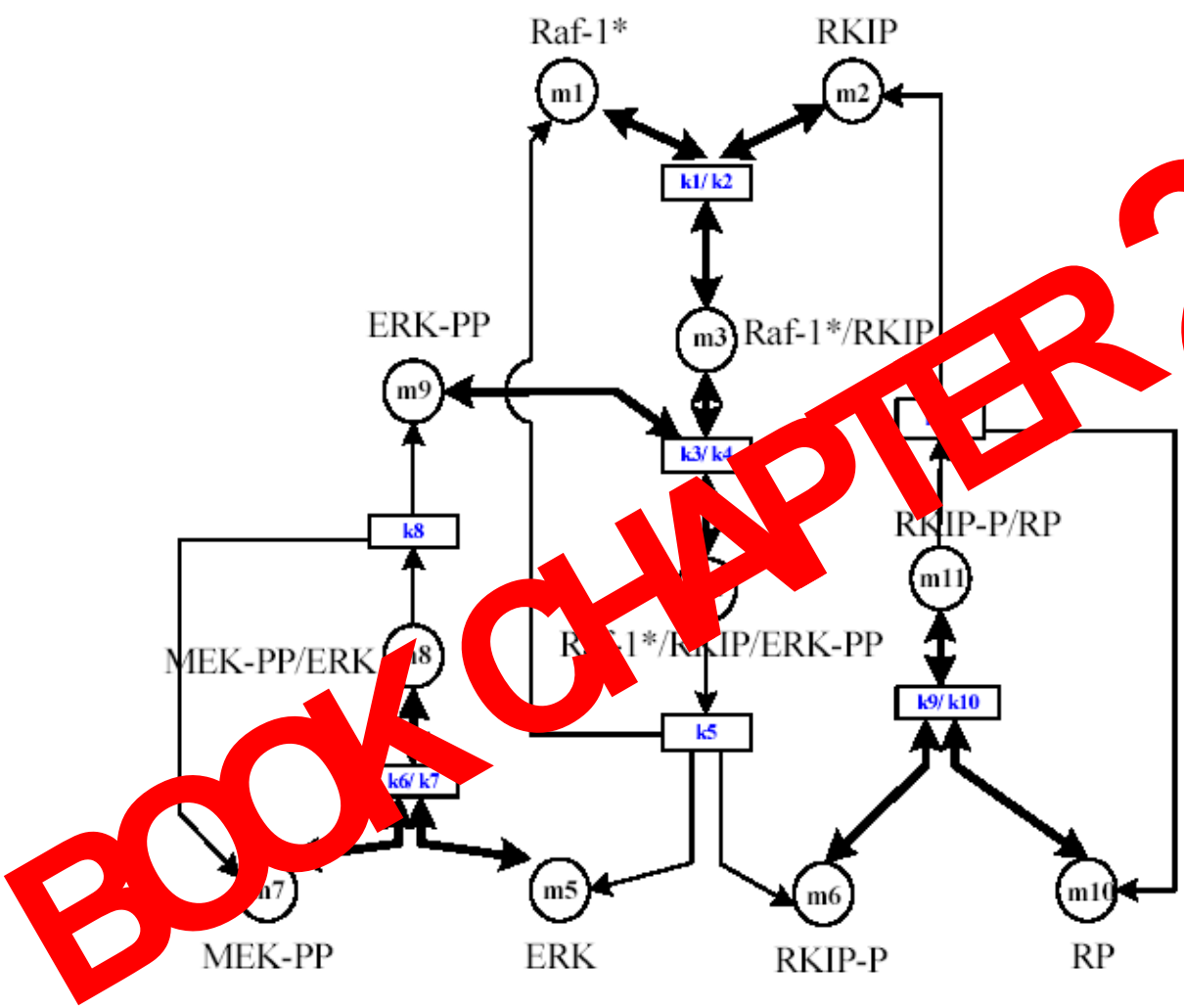


RKIP PATHWAY



[Cho et al., CMSB 2003]

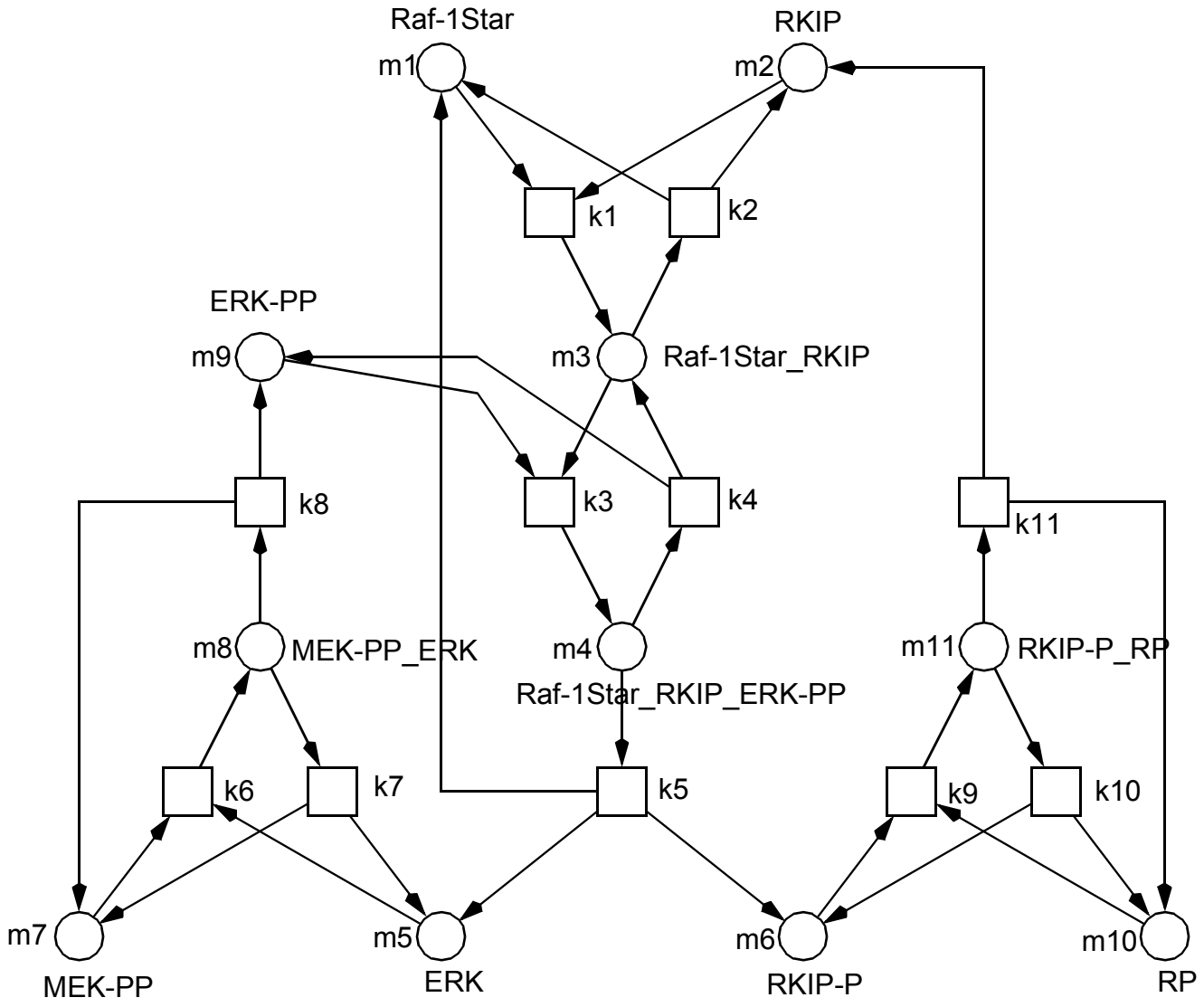
RKIP PATHWAY



BOOK CHAPTER 2010

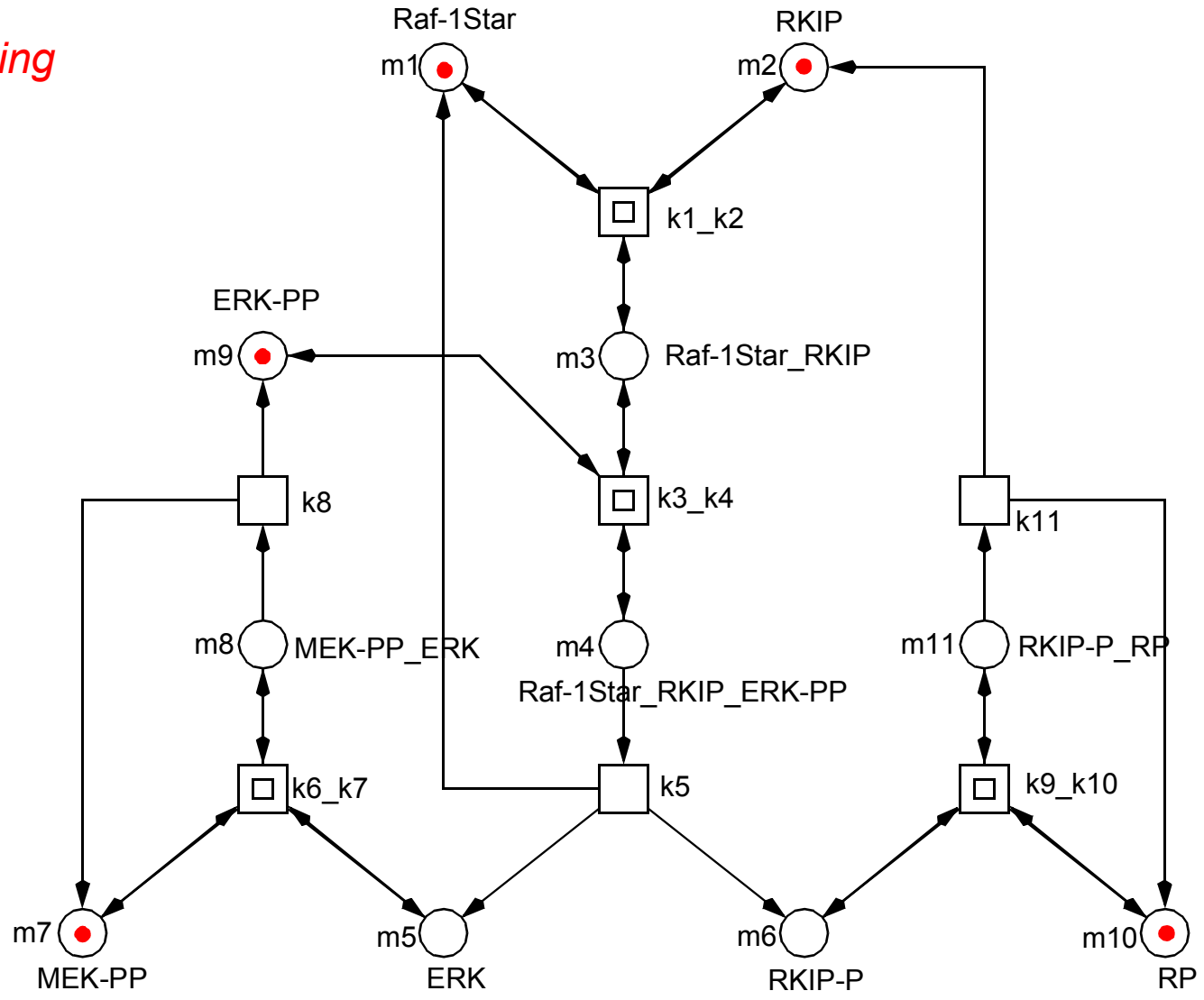
[Cho et al., CMSB 2003]

RKIP Pathway, Petri Net



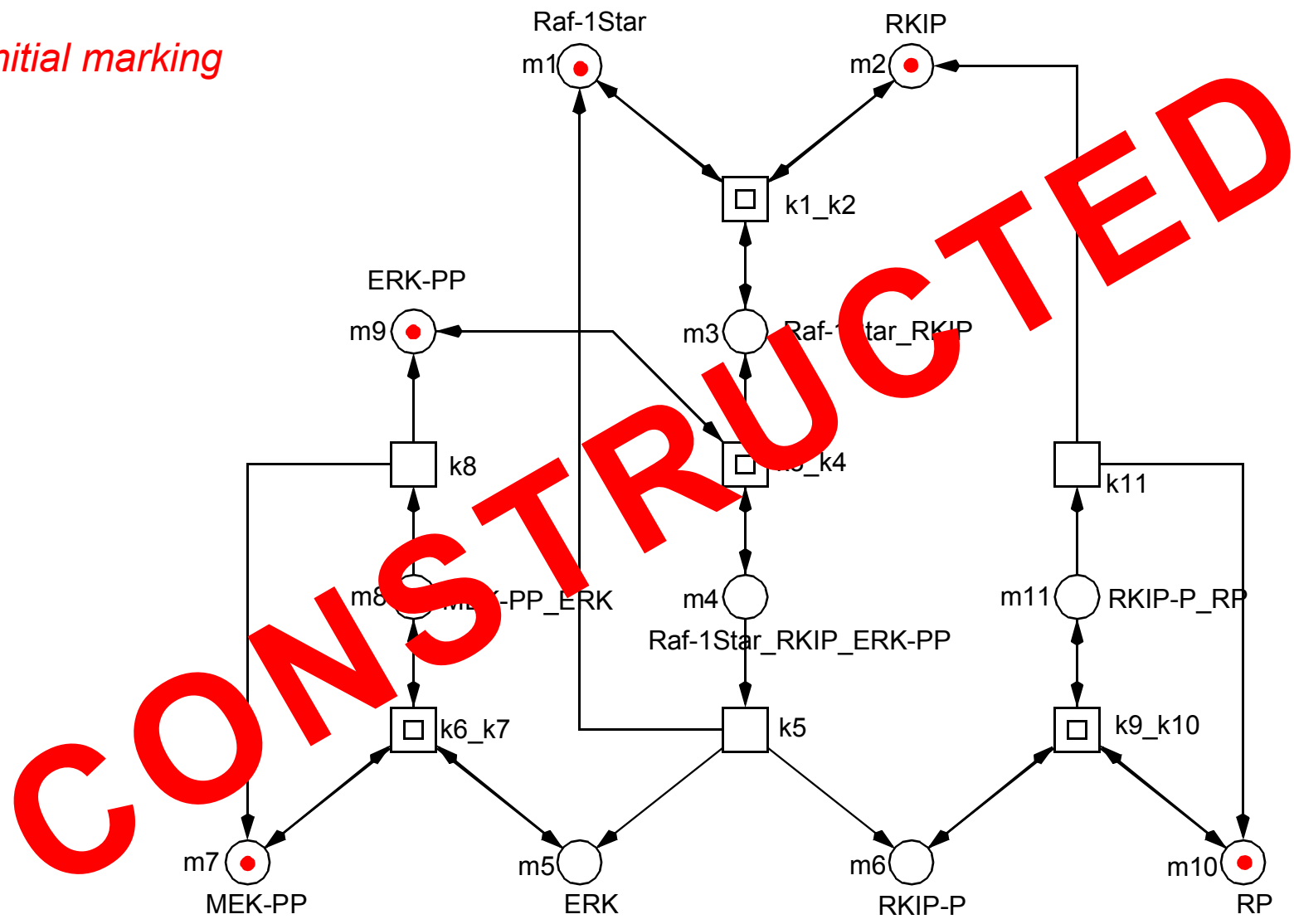
RKIP Pathway, Hierarchical Petri Net

initial marking



RKIP Pathway, Hierarchical Petri Net

initial marking



Petri nets,
typical properties

TYPICAL PETRI NET PROPERTIES

- **How many tokens can reside at most in a given place ?**

  -> *(0, 1, k, oo)* -> ***BOUNDEDNESS***

TYPICAL PETRI NET PROPERTIES

- **How many tokens can reside at most in a given place ?**

  -> *(0, 1, k, oo)* -> ***BOUNDEDNESS***

- **How often can a transition fire ?**

  -> *(0-times, n-times, oo-times)* -> ***LIVENESS***

TYPICAL PETRI NET PROPERTIES

- **How many tokens can reside at most in a given place ?**

  -> *(0, 1, k, oo)* -> ***BOUNDEDNESS***

- **How often can a transition fire ?**

  -> *(0-times, n-times, oo-times)* -> ***LIVENESS***

- **How often can a system state be reached ?**

  -> *never* -> ***UNREACHABLE -> SAFETY PROPERTIES***

  -> *n-times* -> ***REPRODUCIBLE***

  -> *always reachable again* -> ***REVERSIBLE (HOME STATE)***

  -> *reversible initial state* -> ***REVERSIBILITY***

TYPICAL PETRI NET PROPERTIES

- **How many tokens can reside at most in a given place ?**
  - -> *(0, 1, k, oo)* -> ***BOUNDEDNESS***
- **How often can a transition fire ?**
  - -> *(0-times, n-times, oo-times)* -> ***LIVENESS***
- **How often can a system state be reached ?**
  - -> *never* -> ***UNREACHABLE -> SAFETY PROPERTIES***
  - -> *n-times* -> ***REPRODUCIBLE***
  - -> *always reachable again* -> ***REVERSIBLE (HOME STATE)***
  - -> *reversible initial state* -> ***REVERSIBILITY***
- **Are there behaviourally invariant subnet structures ?**
  - -> *token conservation* -> *P* - ***INVARIANTS***
  - -> *token distribution reproduction* -> *T* - ***INVARIANTS***
- **. . . and many more -> temporal logics (CTL, LTL)**

TYPICAL PETRI NET PROPERTIES

- **How many tokens can reside at most in a given place ?**
  - -> *(0, 1, k, oo)* -> ***BOUNDEDNESS***
- **How often can a transition fire ?**
  - -> *(0-times, n-times, oo-times)* -> ***LIVENESS***
- **How often can a system state be reached ?**
  - -> *never* -> ***UNREACHABLE -> SAFETY PROPERTIES***
  - -> *n-times* -> ***REPRODUCIBLE***
  - -> *always reachable again* -> ***REVERSIBLE (HOME STATE)***
  - -> *reversible initial state* -> ***REVERSIBILITY***
- **Are there behaviourally invariant subnet structures ?**
  - -> *token conservation* -> *P* - ***INVARIANTS***
  - -> *token distribution reproduction* -> *T* - ***INVARIANTS***
- **. . . and many more -> temporal logics (CTL, LTL)**

GENERAL BEHAVIOURAL PROPERTIES

SPECIAL BEHAVIOURAL PROPERTIES

Petri nets, typical analysis techniques

MODEL ANIMATION (?)

monika.heiner@informatik.tu-cottbus.de

ANALYSIS TECHNIQUES, OVERVIEW

- **static analyses** -> **no state space construction**

- **dynamic analyses** -> **total/ partial state space construction**

ANALYSIS TECHNIQUES, OVERVIEW

- **static analyses** **-> no state space construction**
  - -> *structural properties (graph theory)*
  - -> *P / T - invariants (linear algebra)*
- **dynamic analyses** **-> total/ partial state space construction**

ANALYSIS TECHNIQUES, OVERVIEW

- **static analyses** **-> no state space construction**

  -> *structural properties (graph theory)*

  -> *P / T - invariants (linear algebra)*

- **dynamic analyses** **-> total/ partial state space construction**

  -> *analysis of general behavioural system properties,*

  e.g. boundedness, liveness, reversibility, . . .

  -> *model checking of special behavioural system properties,*

  e.g. reachability of a given (sub-) system state (with constaints),
  reproducability of a given (sub-) system state (with constraints)

  expressed in temporal logics (CTL / LTL),

  -> *very flexible, powerful query language*

  -> *behaviour checking*

Dynamic analyses

EX1 - RKIP, REACHABILITY GRAPH (STS)

- **simple algorithm**

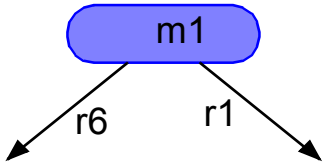
- **nodes : system states**

EX1 - RKIP, REACHABILITY GRAPH (STS)

- simple algorithm
- nodes : system states

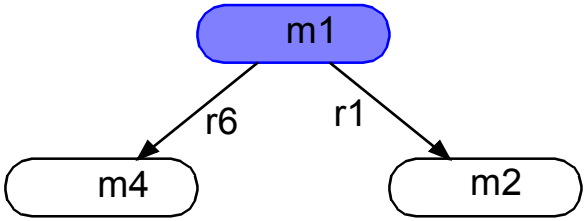
Ex1 - Rkip, Reachability Graph (STS)

- simple algorithm
- nodes : system states
- single step firing rule
- arcs : the (single) firing transition



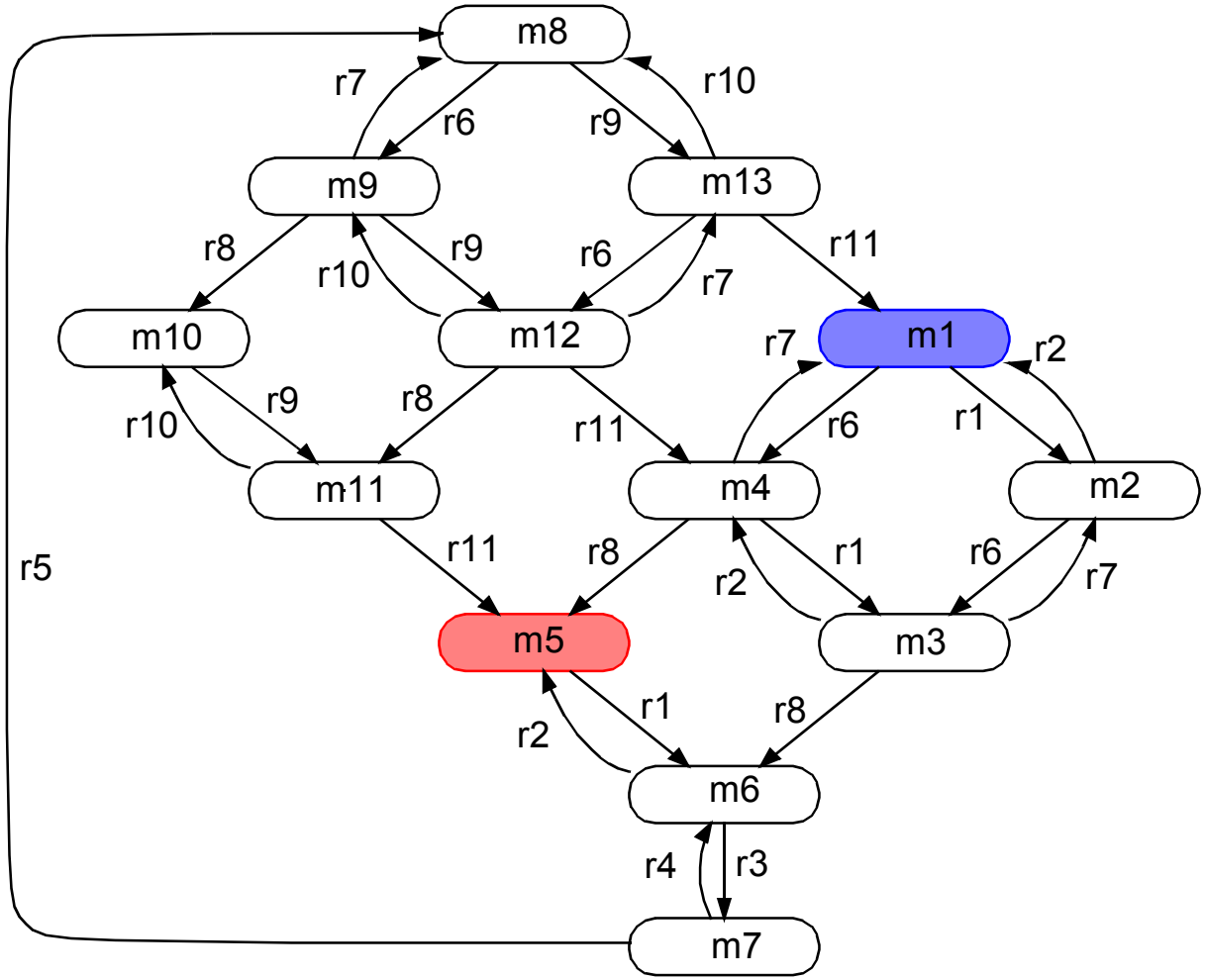
Ex1 - Rkip, Reachability Graph (STS)

- simple algorithm
- nodes : system states
- single step firing rule
- arcs : the (single) firing transition



Ex1 - Rkip, Reachability Graph (STS)

- simple algorithm
- nodes : system states
- single step firing rule
- arcs : the (single) firing transition



DYNAMIC ANALYSES

- **reachability graph *(occurrence / marking graph)***
  - -> *(labelled) state transition system (-> graph)*
  - -> *Kripke structure, CTMC, . . .*
- **interleaving semantics**
  - -> *(sequential) finite automaton*
  - -> *concurrency == enumerating all interleaving sequences*
- **reachability graph construction - simple algorithm**
  - -> *depth-first construction*
  - -> *breath-first construction*
- **efficiency depends on efficiency of data structures**

REACHABILITY GRAPH EVALUATION

- **boundedness**

  -> *finite graph*

- **reversibility**

  -> *one Strongly Connected Component (SCC)*

- **dead states (deadlock)**

  -> *terminal nodes*

- **liveness**

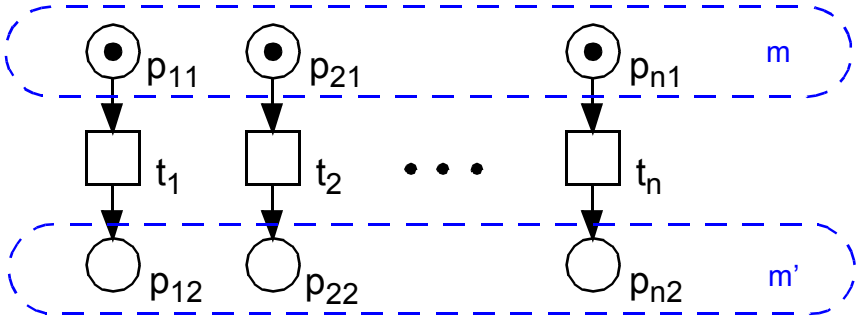
  -> *every transition contained in all terminal SCC*

REACHABILITY GRAPH EVALUATION

- **boundedness**
  -> *finite graph*
- **reversibility**
  -> *one Strongly Connected Component (SCC)*
- **liveness**
  -> *every transition contained in all terminal SCC*
- **dead states (deadlock)**
  -> *terminal nodes*

-> reachability graphs tend to be huge <-

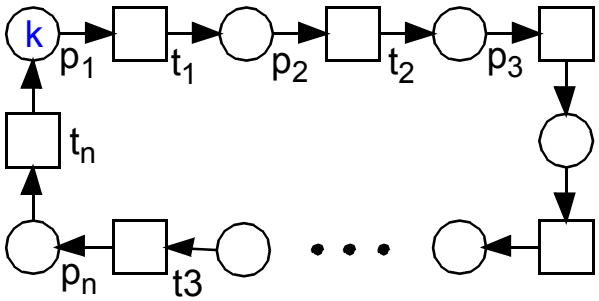
STATE SPACE COMPLEXITY, CAUSES



n! interleaving sequences

m -> m'

2^n - 2 intermediate states



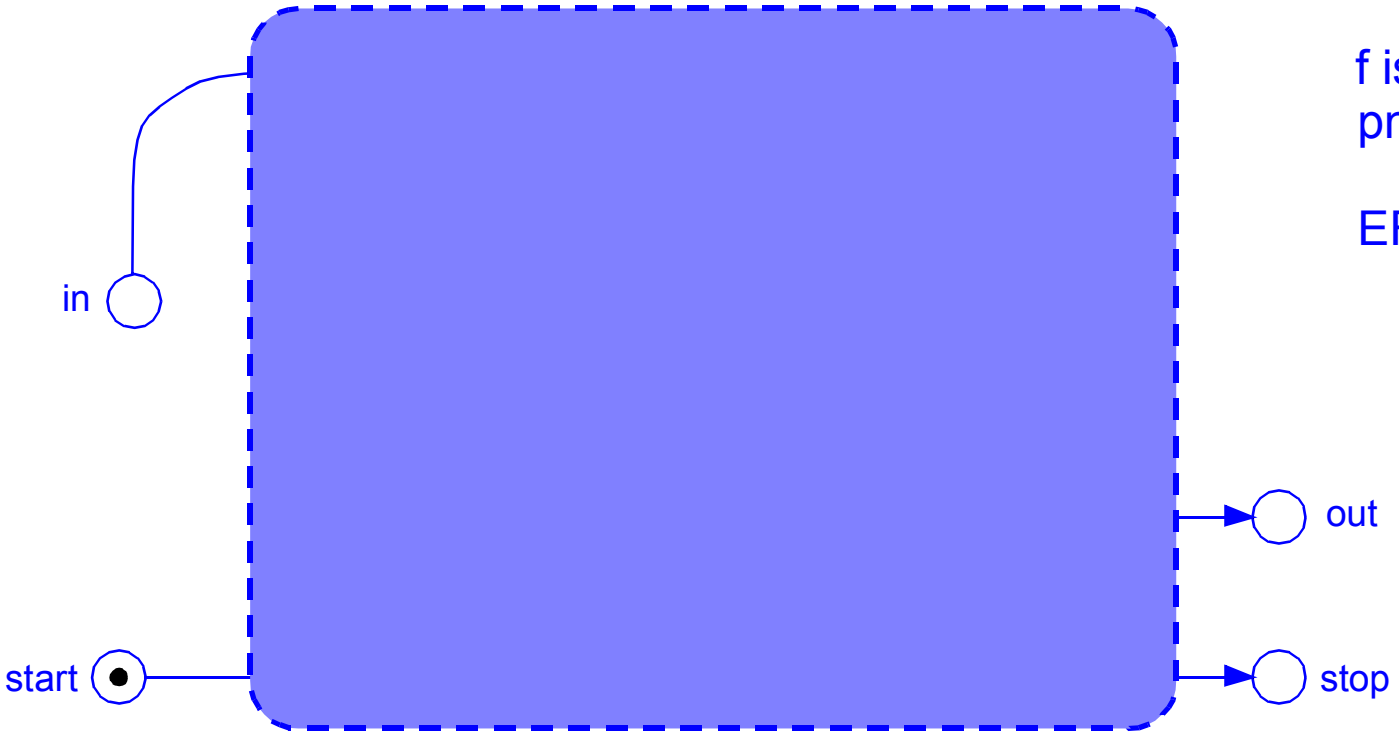
$$\frac{(n + k - 1)!}{(n - 1)!\ k!}$$ states

(combination with repetition)

STATE SPACE COMPLEXITY, FORMAL BACKGROUND

- **infinite for unbounded nets**
- **worst-case for finite state spaces** **[Priese, Wimmel 2003]**

  ... cannot be bounded by a primitive recursive function ...
- **proof** -> Petri net computer for a function **f: N_0^m -> N_0**



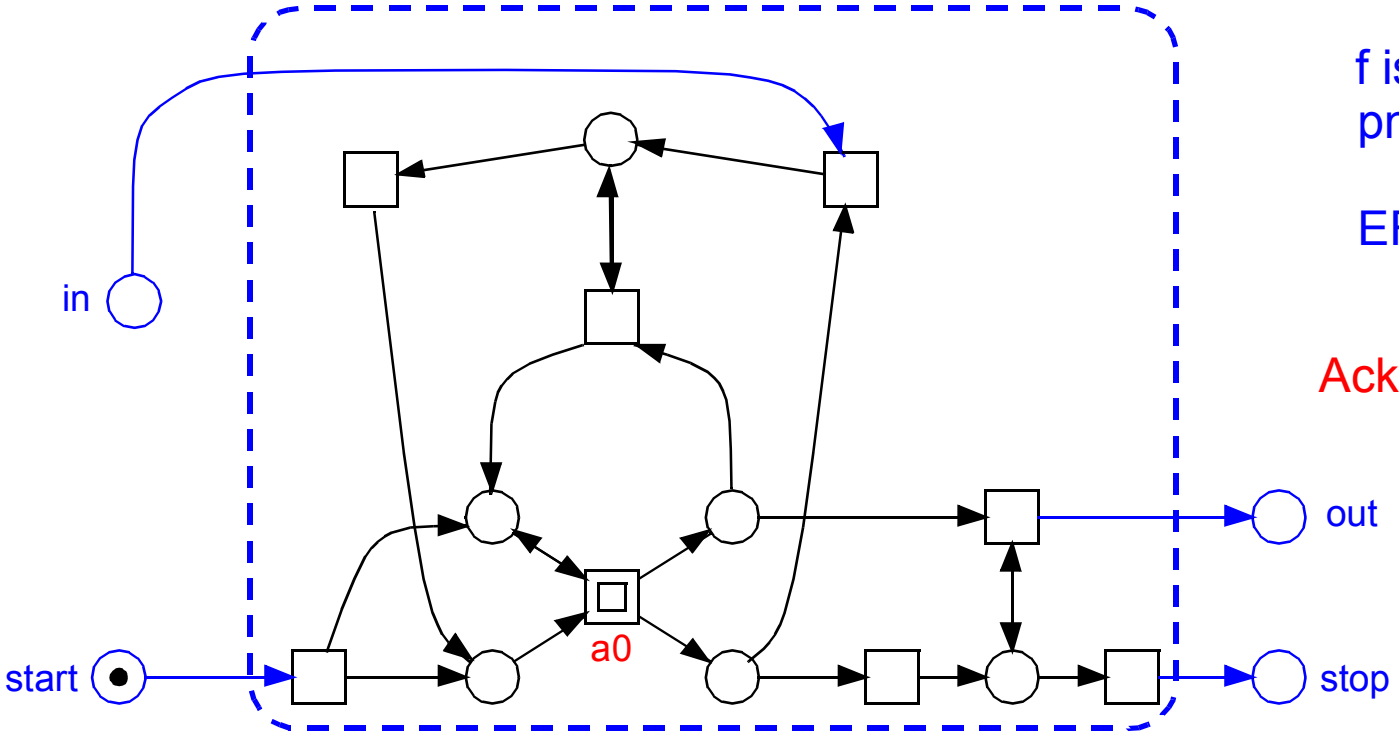
f is weakly pn-computable:

EF(out = f(in) &
stop = 1)

STATE SPACE COMPLEXITY, FORMAL BACKGROUND

- **infinite for unbounded nets**
- **worst-case for finite state spaces** **[Priese, Wimmel 2003]**

  ... cannot be bounded by a primitive recursive function ...
- **proof** -> Petri net computer for a function **f: N_0^m -> N_0**



f is weakly pn-computable:

EF(out = f(in) &
stop = 1)

Ackermann function a1

STATIC ANALYSES

STATIC ANALYSIS, OVERVIEW

- **no (full / partial) state space construction**

  -> *works also for unbounded models = infinite state spaces*

- **structural analysis**

  -> *boundary nodes, . . .*

  -> *Siphon Trap Property*

- **net reduction**

-> *to decide liveness (hopefully)*

STATIC ANALYSIS, OVERVIEW

- **no (full / partial) state space construction**

  -> works also for unbounded models = infinite state spaces

- **structural analysis**

  -> boundary nodes, . . .

  -> Siphon Trap Property

- **net reduction**

-> to decide liveness (hopefully)

- **invariants**

  -> P-invariants - mass-preserving modules

  -> T-invariants - state-repeating modules

-> to decide boundedness

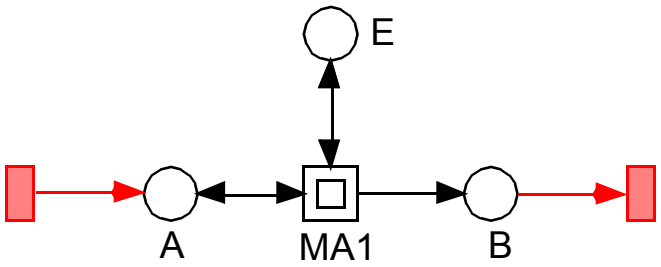
- **modularization by T-invariants**

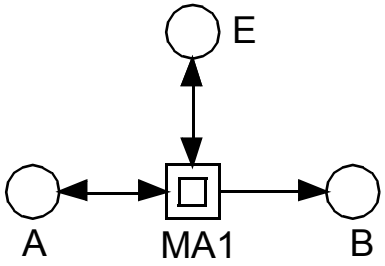
  -> abstract dependent transition sets (ADT-sets) define building blocks

- **core network identification**

STRUCTURAL ANALYSIS, SOME EXAMPLES

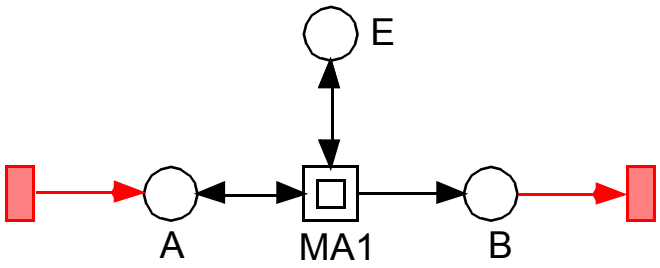
- **boundary nodes**
  - -> *input transitions -> not BND*
  - -> *input places -> not LIVE*
  - -> *LIVE & BND -> no boundary nodes*



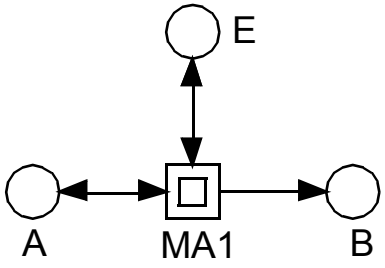


STRUCTURAL ANALYSIS, SOME EXAMPLES

- **boundary nodes**
  - -> *input transitions -> not BND*
  - -> *input places -> not LIVE*
  - -> *LIVE & BND -> no boundary nodes*

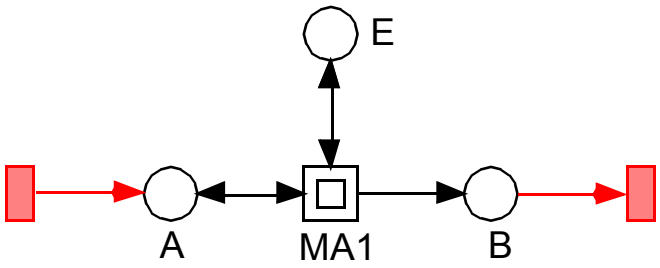


- **conservative -> BND**
  - -> *all transitions preserve token number*

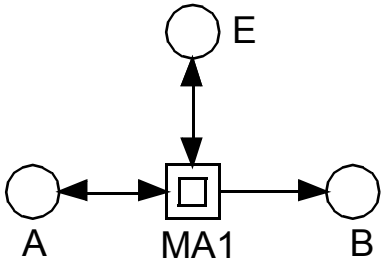


STRUCTURAL ANALYSIS, SOME EXAMPLES

- **boundary nodes**
  - -> *input transitions -> not BND*
  - -> *input places -> not LIVE*
  - -> *LIVE & BND -> no boundary nodes*



- **conservative -> BND**
  - -> *all transitions preserve token number*

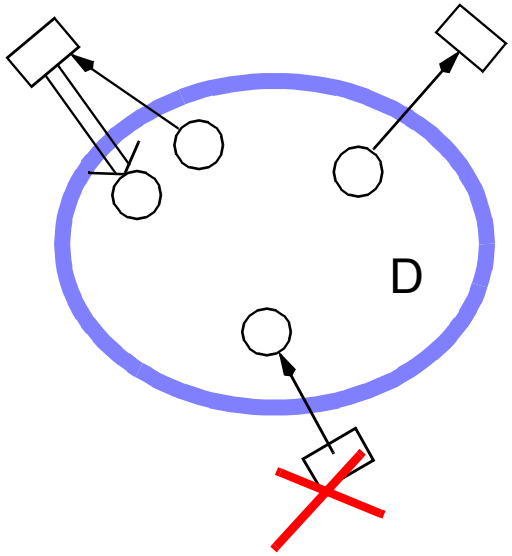


- **Siphon Trap Property (STP)**

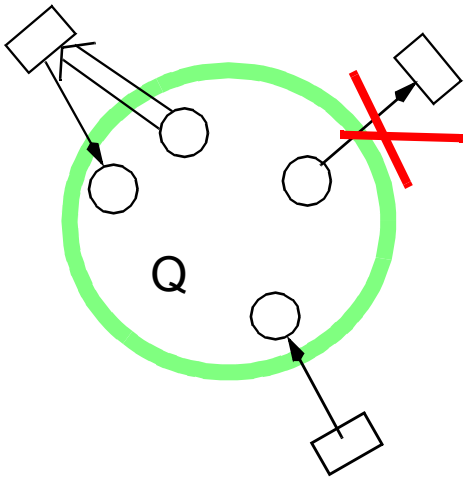
SIPHON TRAP PROPERTY (STP)

Siphon D

FD DF



any transition putting token into the set
also takes token from it:
an empty siphon will never get marked



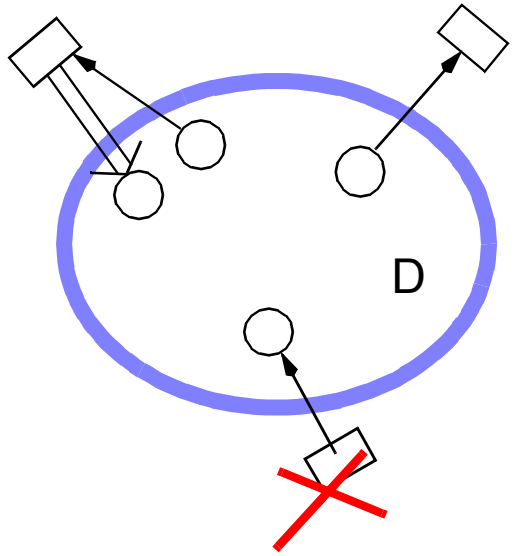
Trap Q

QF FQ

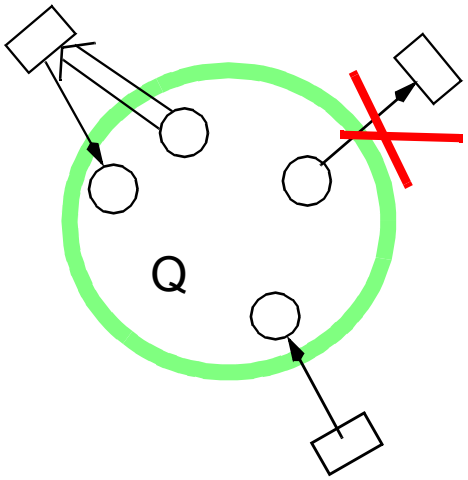
any transition taking tokens from the set
also puts token into it:
a marked trap will never get empty

SIPHON TRAP PROPERTY (STP)

Siphon D
FD DF

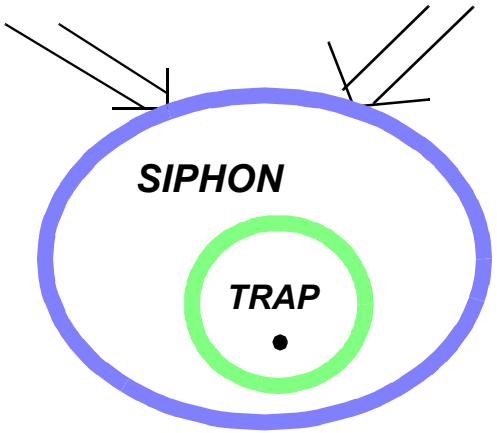


any transition putting token into the set
also takes token from it:
an empty siphon will never get marked



Trap Q
QF FQ

any transition taking tokens from the set
also puts token into it:
a marked trap will never get empty



STP: each siphon contains a
(sufficiently) marked trap (at m_0)

SIPHON TRAP PROPERTY (STP)

- **allows to decide liveness, sometimes**

EFC -> STP (& HOM & NBM)	*<->*	*live*
ES & STP (& HOM & NBM)	*->*	*live*
STP (& HOM & NBM)	*->*	*no DSt*
no siphon	*->*	*live*

SIPHON TRAP PROPERTY (STP)

- **allow to decide liveness, sometimes**

EFC -> STP (& HOM & NBM)	*<->*	*live*
ES & STP (& HOM & NBM)	*->*	*live*
STP (& HOM & NBM)	*->*	*no DSt*
no siphon	*->*	*live*

- **some examples, where it helps**

-> RKIP pathway (BND): ES & STP -> live

-> MAPK cascade (BND): STP & nES -> no DSt

-> biosensor (not BND): ES & STP -> live

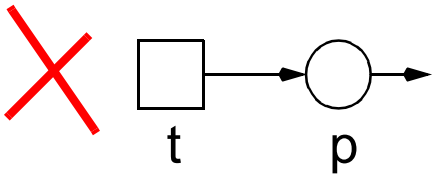
-> apoptosis (not BND): no siphon -> live

-> lac operon (not BND): STP & nES -> no DSt

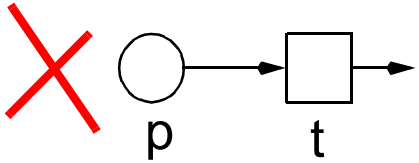
NET REDUCTION

- **downsizing the net structure while preserving some properties**
  -> *liveness, boundedness*

- **example of two simple reduction rules**



t live
p unbounded



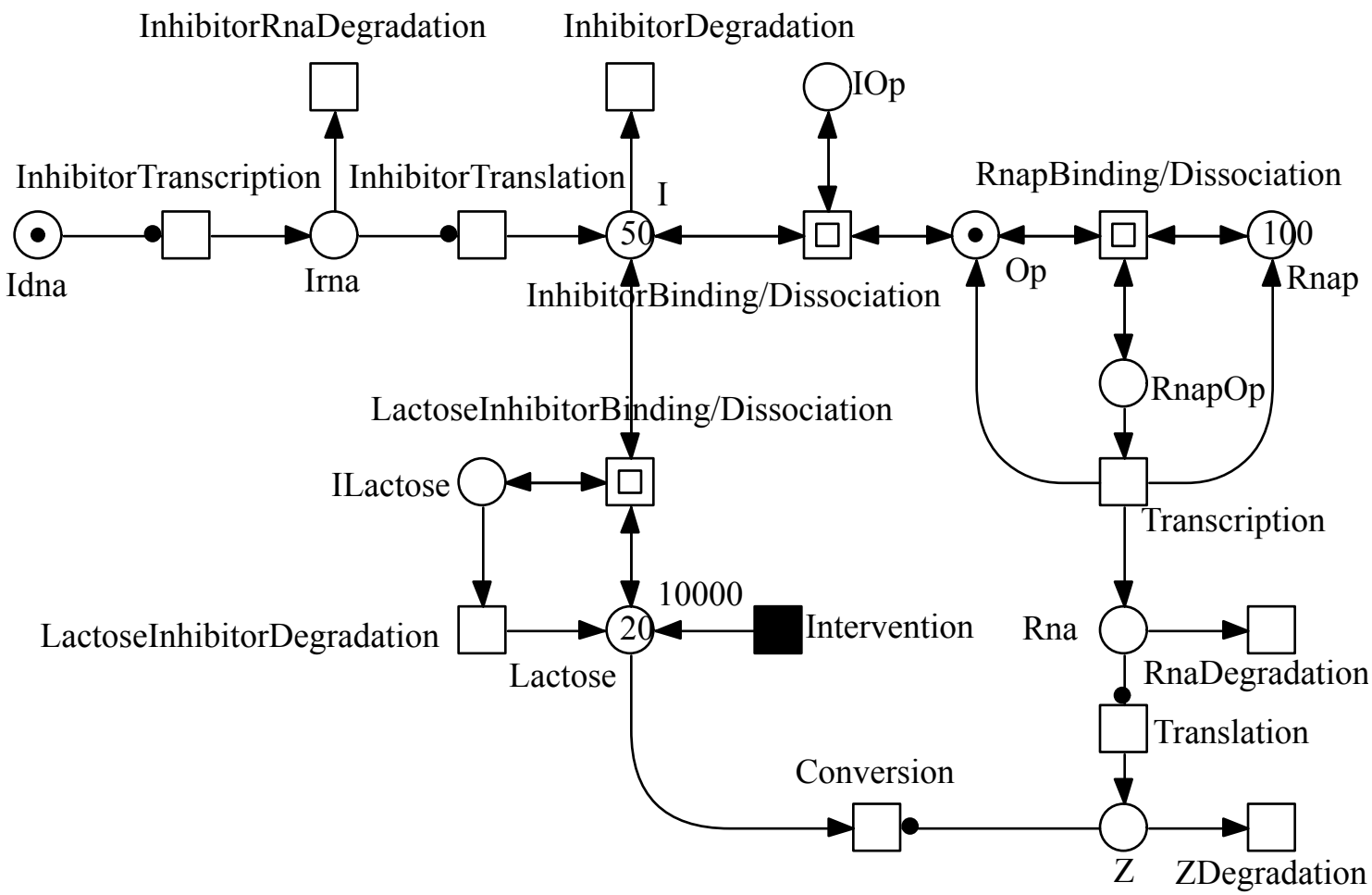
t not live
p bounded

-> *input nodes allow net reduction:*
conclude properties -> delete nodes -> conclude properties -> . . .

- **several sets of reduction rules**
  -> *relatively weak, in general*
  -> *sensitive to the order they are applied*
  -> *however, sometimes they help*

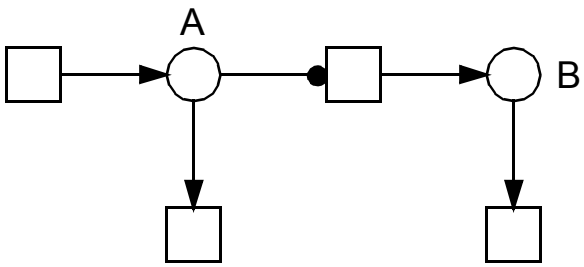
Net Reduction, Example

- lac operon (Wilkinson 2006)



NET REDUCTION, EXAMPLE

- lac operon, reduced by INA/HUB



- liveness becomes obvious

PUR	ORD	HOM	NBM	CSV	SCF	CON	SC	FT0	TF0	FP0	PF0	NC
N	N	Y	Y	N	N	N	N	Y	Y	Y	N	nES
DTP	CPI	CTI	SCTI	SB	k-B	1-B	DCF	DSt	DTr	LIV	RV	
Y	N	Y	-	N	N	N	-	N	-	-	-	

NET INVARIANTS,
A CRASH COURSE

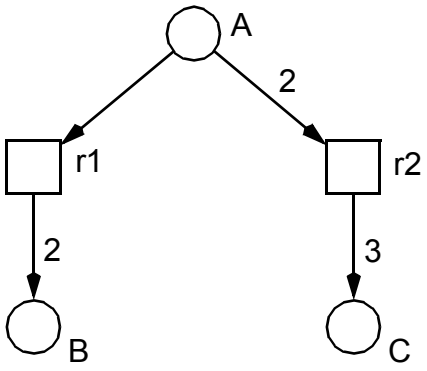
monika.heiner@informatik.tu-cottbus.de

T-INVARIANTS, EX1

r1: A -> 2 B
r2: 2 A -> 3 C

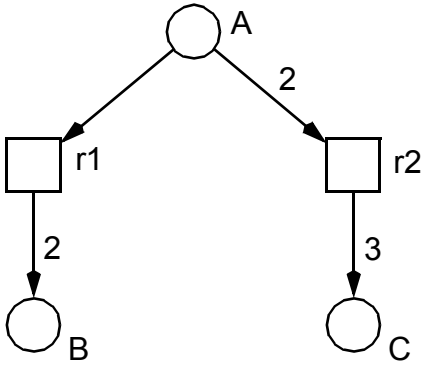
T-INVARIANTS, EX1

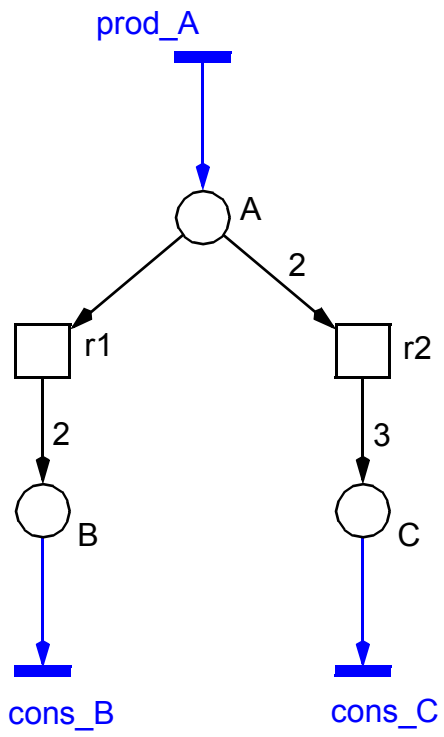
r1: A -> 2 B
r2: 2 A -> 3 C



T-INVARIANTS, EX1

r1: A -> 2 B
r2: 2 A -> 3 C





INCIDENCE MATRIX C

- **a representation of the net structure** **=> stoichiometric matrix**

$C =$

P \ T	t1 . . .	tj	. . . tm
p1			
pi		cij	
: pn		vtj	

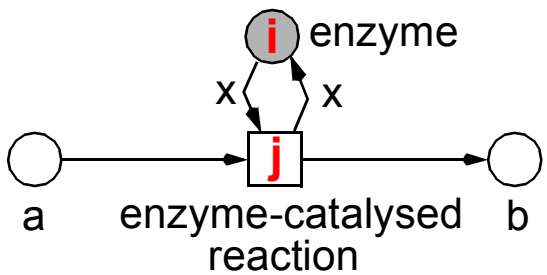
cij = C(pi, tj) = delta tj(pi)

vtj = C(,tj)*

- **matrix entry cij:**
  token change in place pi by firing of transition tj

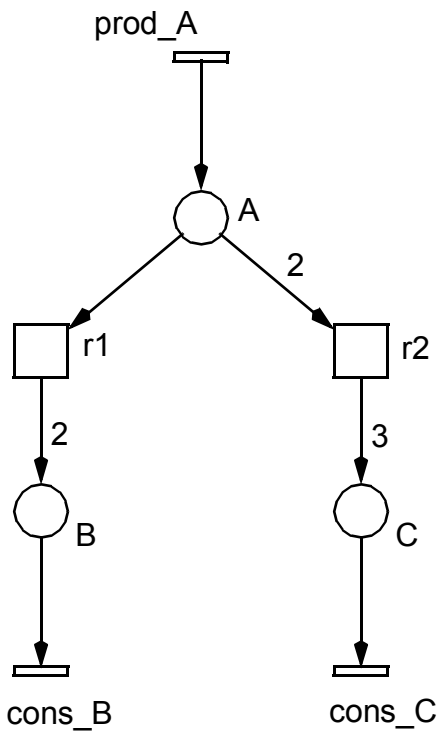
- **matrix column vtj:**
  vector describing the change of the whole marking by firing of tj

- **side-conditions are neglected**

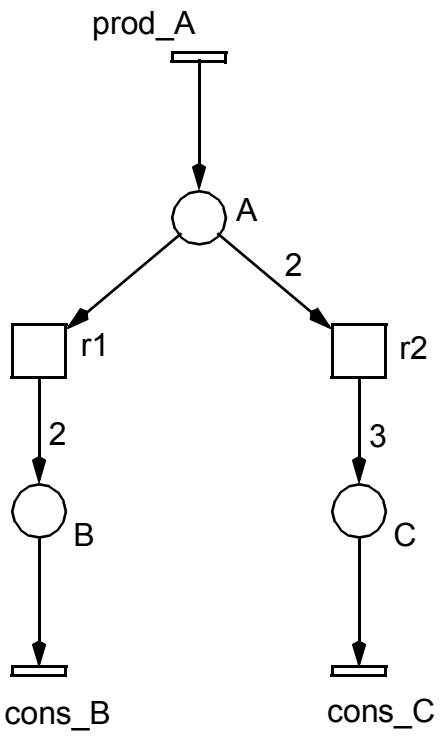


cij = 0

INCIDENCE MATRIX C, EX1

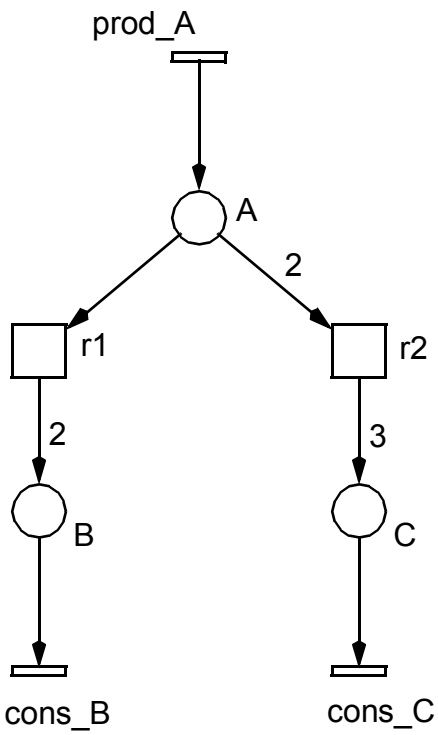


INCIDENCE MATRIX C, EX1



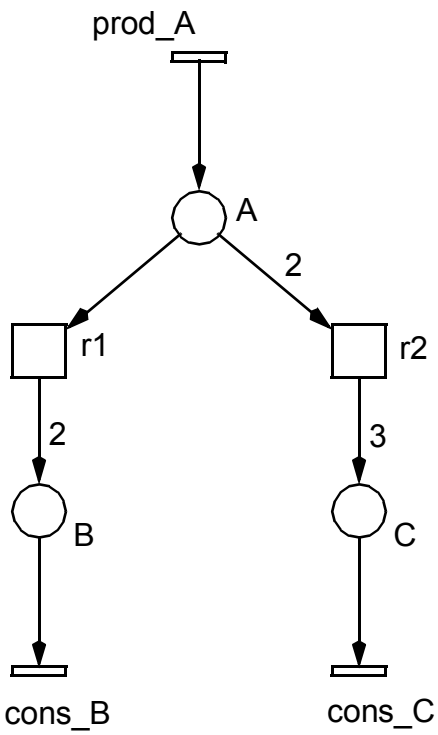
P \ T	r1	r2	prod_A	cons_B	cons_C
A					
B					
C					

INCIDENCE MATRIX C, EX1



P \ T	r1	r2	prod_A	cons_B	cons_C
A	-1	-2	1		
B	2			-1	
C		3			-1

INCIDENCE MATRIX C, EX1



P \ T	r1	r2	prod_A	cons_B	cons_C
A	-1	-2	1		
B	2			-1	
C		3			-1
	1		**1**	**2**	

T-INVARIANTS, BASICS

- **Lautenbach, 1973**
- **T-invariants**
  -> *integer solutions x*

$$Cx = 0,\ x \neq 0,\ x \geq 0$$

-> Schuster, 1993

-> multisets of transitions

T-INVARIANTS, BASICS

- **Lautenbach, 1973** **-> Schuster, 1993**
- **T-invariants** ***-> multisets of transitions***
  - -> *integer solutions x*

$$Cx = 0, x \neq 0, x \geq 0$$

- **minimal T-invariants**
  - -> *there is no T-invariant with a smaller support* -> *sets of transitions*
  - -> *gcD of all non-zero entries is 1*
- **any T-invariant is a non-negative linear combination of minimal ones**
  - -> *multiplication with a positive integer*
  - -> *addition*
  - -> *division by a common divisor*

$$kx = \sum_i a_i x_i$$

T-INVARIANTS, BASICS

- **Lautenbach, 1973** -> **Schuster, 1993**
- **T-invariants** -> ***multisets of transitions***
  - -> *integer solutions x*

$$Cx = 0,\ x \neq 0,\ x \geq 0$$

- **minimal T-invariants**
  - -> *there is no T-invariant with a smaller support* -> *sets of transitions*
  - -> *gcD of all non-zero entries is 1*
- **any T-invariant is a non-negative linear combination of minimal ones**
  - -> *multiplication with a positive integer*
  - -> *addition*
  - -> *division by a common divisor*

$$kx = \sum_i a_i x_i$$

- **Covered by T-Invariants (CTI)**
  - -> *each transition belongs to a T-invariant*

-> BND & LIVE => CTI (necessary condition)

T-INVARIANTS, INTERPRETATIONS

- **T-invariants = (multi-) sets of transitions = Parikh vector**
  - -> *zero effect on marking*
  - -> *reproducing a marking / system state*

- **two interpretations**
  1. *partially ordered transition sequence* **-> behaviour understanding**
     of transitions occuring one after the other
     - -> *substance / signal flow*
  2. *relative transition firing rates* **-> steady state behaviour**
     of transitions occuring permanently & concurrently
     - -> *steady state behaviour*

- **a minimal T-invariant defines a connected subnet**
  - -> *the T-invariant's transitions (the support),*
    + all their pre- and post-places
    + the arcs in between
  - -> *pre-set of support = post-set of support* **-> self-contained**

T-INVARIANTS, INTERPRETATIONS

- **T-invariants = (multi-) sets of transitions = Parikh vector**
  - -> *zero effect on marking*
  - -> *reproducing a marking / system state*

- **two interpretations**
  1. *partially ordered transition sequence* **-> behaviour understanding**
     of transitions occuring one after the other
     - -> *substance / signal flow*
  2. *relative transition firing rates* **-> steady state behaviour**
     of transitions occuring permanently & concurrently
     - -> *steady state behaviour*

- **a minimal T-invariant defines a connected subnet**
  - -> *the T-invariant's transitions (the support),*
    + all their pre- and post-places
    + the arcs in between
  - -> *pre-set of support = post-set of support* **-> self-contained**

T-INVARIANTS, INTERPRETATIONS

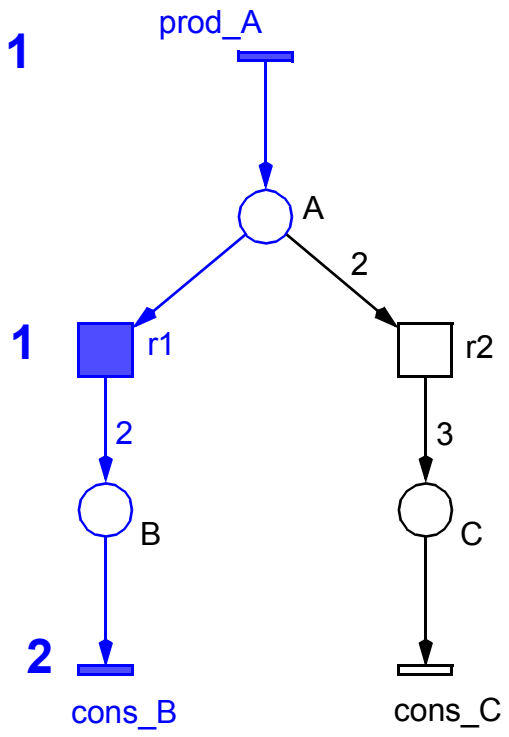
- **T-invariants = (multi-) sets of transitions = Parikh vector**
  - -> *zero effect on marking*
  - -> *reproducing a marking / system state*

- **two interpretations**
  1. *partially ordered transition sequence*
     ->behaviour understanding
     of transitions occuring one after the other
     - -> *substance / signal flow*
  2. *relative transition firing rates* **->steady state behaviour**
     of transitions occuring permanently & concurrently
     - -> *steady state behaviour*

- **a minimal T-invariant defines a connected subnet**
  - -> *the T-invariant's transitions (the support),*
    + all their pre- and post-places
    + the arcs in between
  - -> *pre-set of support = post-set of support*
    -> **self-contained**

T-INVARIANTS, EX1

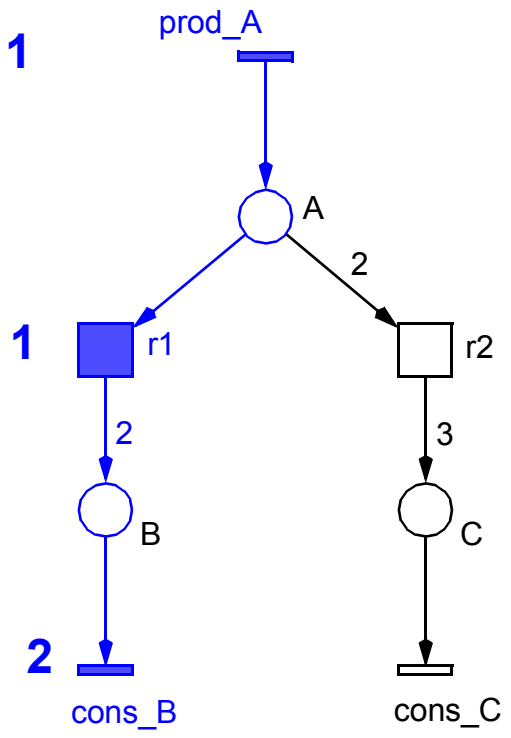
r1: A -> 2 B
r2: 2 A -> 3 C



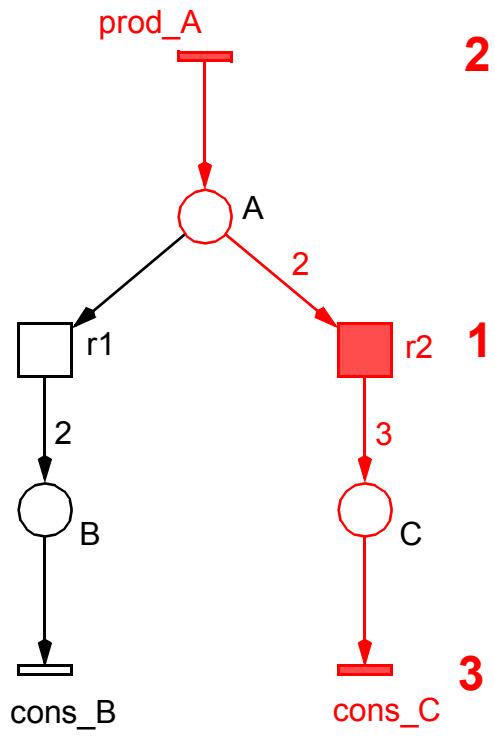
T-INVARIANT 1

T-INVARIANTS, EX1

r1: A -> 2 B
r2: 2 A -> 3 C



T-INVARIANT 1

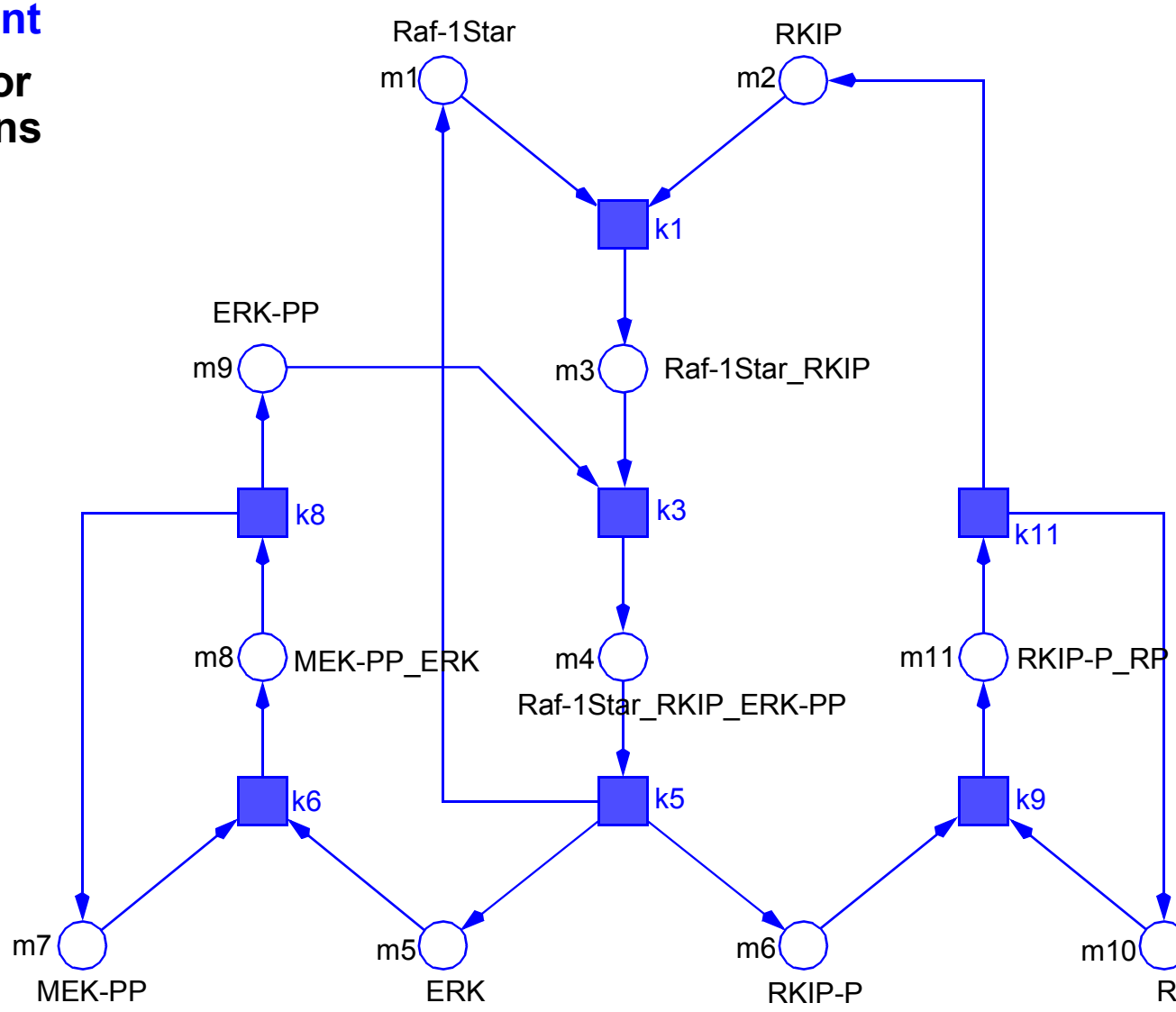


T-INVARIANT 2

T-INVARIANT, EX2 - RKIP PATHWAY

-> non-trivial T-invariant

+ four trivial ones for reversible reactions



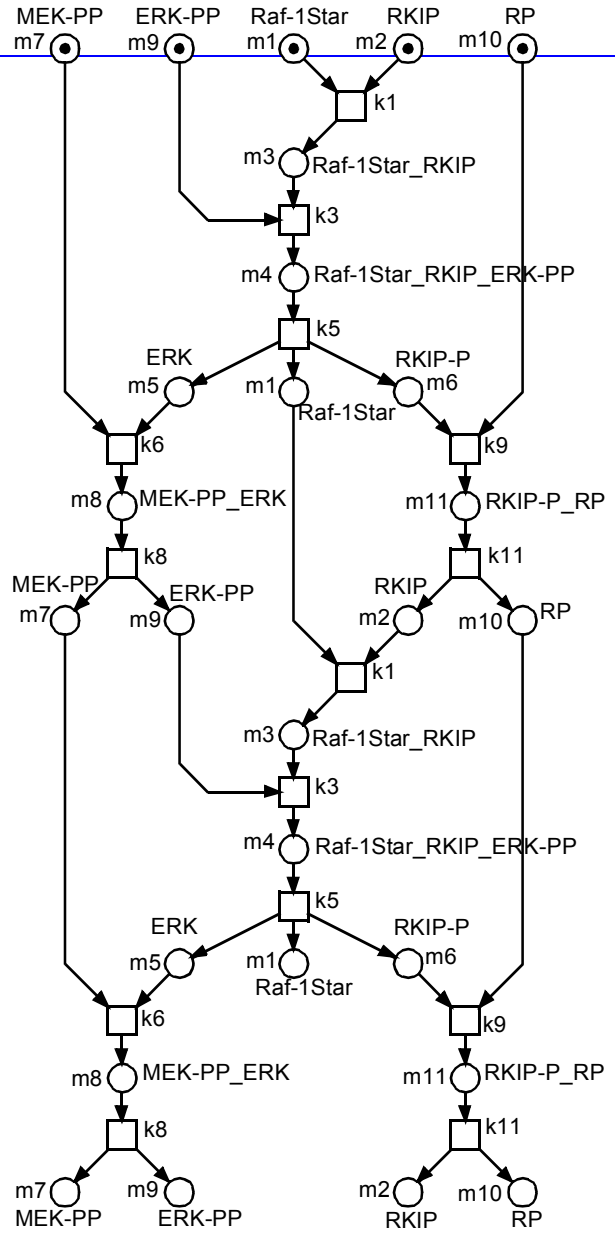
NON-TRIVIAL T-INVARIANT, RUN

- **feasibility check under the constructed marking**

- **T-invariant's unfolding to describe its behaviour**
  - -> **partial order structure**

- **labelled condition / event net**
  - -> *events (boxes)*
    - *transition occurences*
  - -> *conditions (circles)*
    - *involved compounds*

- **occurrence net**
  - -> *acyclic*
  - -> *no backward branching conditions*
  - -> *infinite*


MEK-PP
m7
ERK-PP
m9
Raf-1Star
m1
RKIP
m2
RP
m10
k1
m3
Raf-1Star_RKIP
k3
m4
Raf-1Star_RKIP_ERK-PP
k5
ERK
m5
m1
Raf-1Star
RKIP-P
m6
k6
k9
m8
MEK-PP_ERK
m11
RKIP-P_RP
k8
k11
MEK-PP
m7
ERK-PP
m9
RKIP
m2
m10
RP
k1
m3
Raf-1Star_RKIP
k3
m4
Raf-1Star_RKIP_ERK-PP
k5
ERK
m5
m1
Raf-1Star
RKIP-P
m6
k6
k9
m8
MEK-PP_ERK
m11
RKIP-P_RP
k8
k11
m7
MEK-PP
m9
ERK-PP
m2
RKIP
m10
RP

P-INVARIANTS, BASICS

- **Lautenbach, 1973**

- **P-invariants** -> *multisets of places*
  - -> *integer solutions y*

$$yC = 0,\ y \neq 0,\ y \geq 0$$

- **minimal P-invariants**
  - -> *there is no P-invariant with a smaller support* -> *sets of places*
  - -> *gcD of all entries is 1*

- **any P-invariant is a non-negative linear combination of minimal ones**
  - -> *multiplication with a positive integer*
  - -> *addition*
  - -> *Division by gcD*

$$ky = \sum_i a_i y_i$$

- **Covered by P-Invariants (CPI)**
  - -> *each place belongs to a P-invariant*
  - -> *CPI => BND (sufficient condition)*

P-INVARIANTS, INTERPRETATION

- **the firing of any transition has no influence on the weighted sum of tokens on the P-invariant's places**

  -> *for all t: the effect of the arcs, removing tokens from a P-invariant's places is equal to the effect of the arcs, adding tokens to a P-invariant's places*

P-INVARIANTS, INTERPRETATION

- **the firing of any transition has no influence on the weighted sum of tokens on the P-invariant's places**

  -> *for all t: the effect of the arcs, removing tokens from a P-invariant's places is equal to the effect of the arcs, adding tokens to a P-invariant's places*

- **P-invariants = set of places with**

  -> *a constant weighted sum of tokens for all markings m reachable from m_0*

  $$ym = ym_0$$

  -> *token / compound preservation,*

  -> *moieties*

  -> *a place belonging to a P-invariant is bounded*

P-INVARIANTS, INTERPRETATION

- **the firing of any transition has no influence on the weighted sum of tokens on the P-invariant's places**

  -> *for all t: the effect of the arcs, removing tokens from a P-invariant's places is equal to the effect of the arcs, adding tokens to a P-invariant's places*

- **P-invariants = set of places with**

  -> *a constant weighted sum of tokens for all markings m reachable from m_0*

  $ym = ym_0$

  -> *token / compound preservation,*

  -> *moieties*

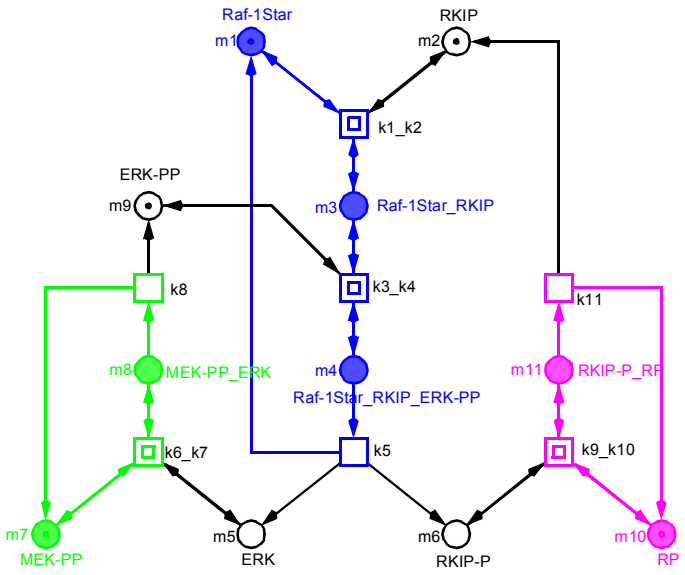
  -> *a place belonging to a P-invariant is bounded*

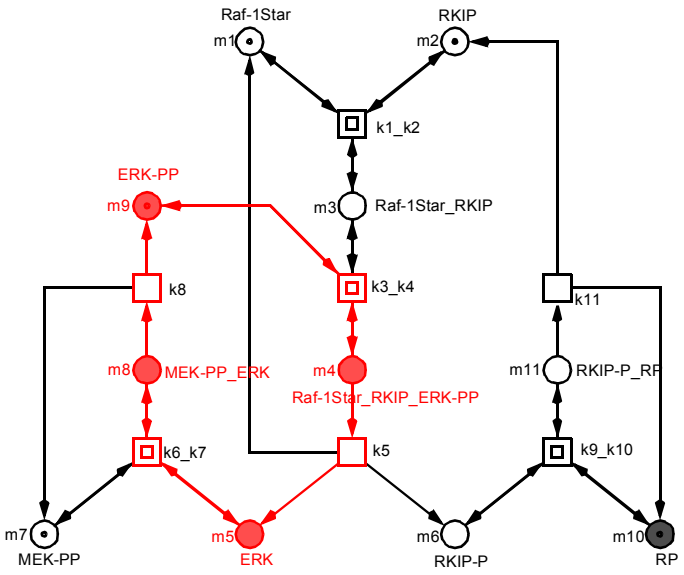
- **a P-invariant defines a subnet**

  -> *the P-invariant's places (the support),*
  + all their pre- and post-transitions
  + the arcs in between

  -> *pre-sets of supports = post-sets of supports* -> **self-contained**

P-INVARIANTS, EX - RKIP PATHWAY





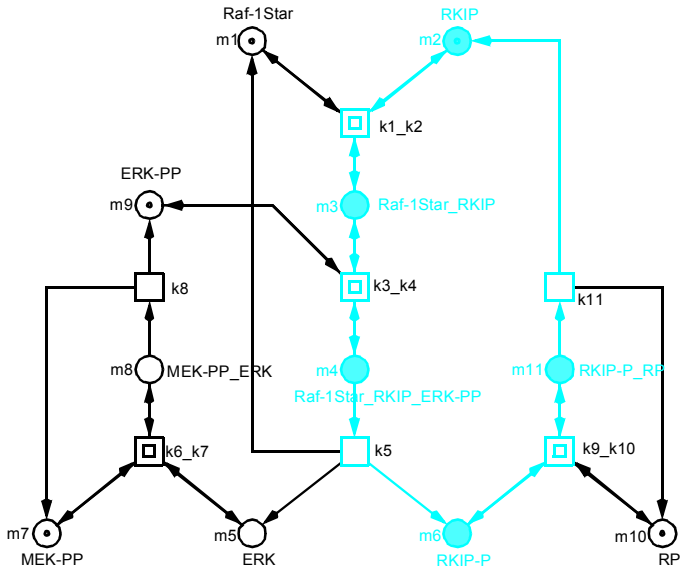
P-INV1: MEK

P-INV2: RAF-1STAR

P-INV3: RP

P-INV4: ERK

P-INV5: RKIP



CONSTRUCTION OF THE INITIAL MARKING

- **each P-invariant gets at least one token**

  -> *P-invariants are structural deadlocks and traps*

CONSTRUCTION OF THE INITIAL MARKING

- **each P-invariant gets at least one token**
  - -> *P-invariants are structural deadlocks and traps*
- **in signal transduction networks**
  - -> *exactly one token, corresponding to species conservation*
  - -> *token in least active state*

CONSTRUCTION OF THE INITIAL MARKING

- **each P-invariant gets at least one token**
  - -> *P-invariants are structural deadlocks and traps*
- **in signal transduction networks**
  - -> *exactly one token, corresponding to species conservation*
  - -> *token in least active state*
- **all (non-trivial) T-invariants get feasible**
  - -> *to make the net live*

CONSTRUCTION OF THE INITIAL MARKING

- **each P-invariant gets at least one token**
  - -> *P-invariants are structural deadlocks and traps*
- **in signal transduction networks**
  - -> *exactly one token, corresponding to species conservation*
  - -> *token in least active state*
- **all (non-trivial) T-invariants get feasible**
  - -> *to make the net live*
- **minimal marking**
  - -> *minimization of the state space*

Construction of the Initial Marking

- **each P-invariant gets at least one token**
  - -> *P-invariants are structural deadlocks and traps*
- **in signal transduction networks**
  - -> *exactly one token, corresponding to species conservation*
  - -> *token in least active state*
- **all (non-trivial) T-invariants get feasible**
  - -> *to make the net live*
- **minimal marking**
  - -> *minimization of the state space*

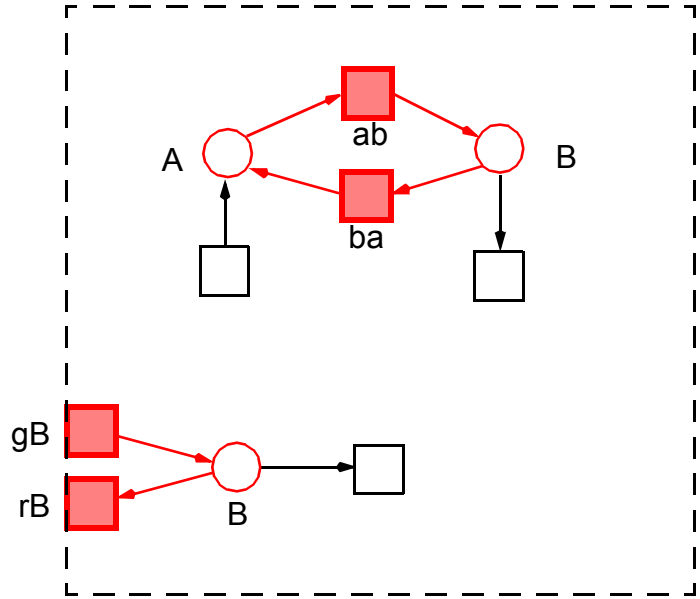
-> UNIQUE INITIAL MARKING <-

T-INVARIANTS, BASIC TYPES IN BIO NETWORKS

- **trivial minimal T-invariants**
  - -> *reversible reactions*
  - -> *boundary transitions of auxiliary compounds*

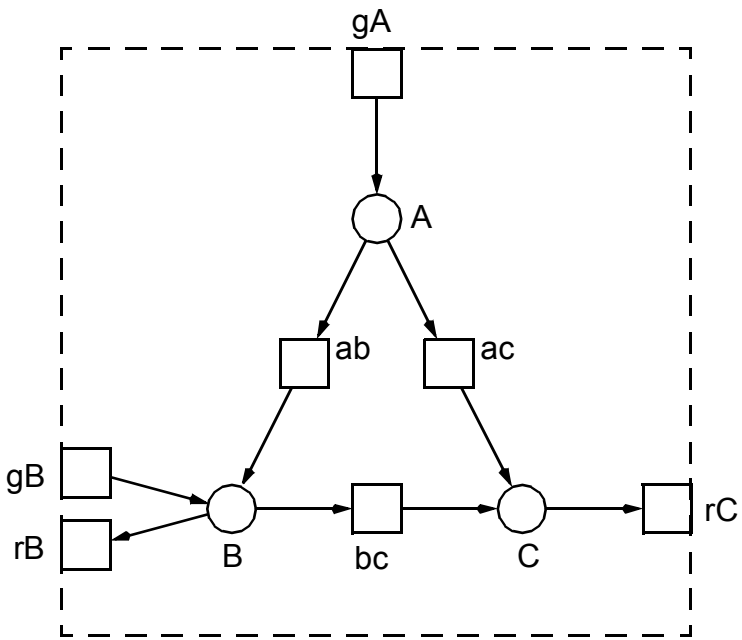
- **non-trivial minimal T-invariants**
  - -> *i/o-T-invariants*

    covering boundary transitions of input / output compounds
  - -> *inner cycles*



T-INVARIANTS, EX3

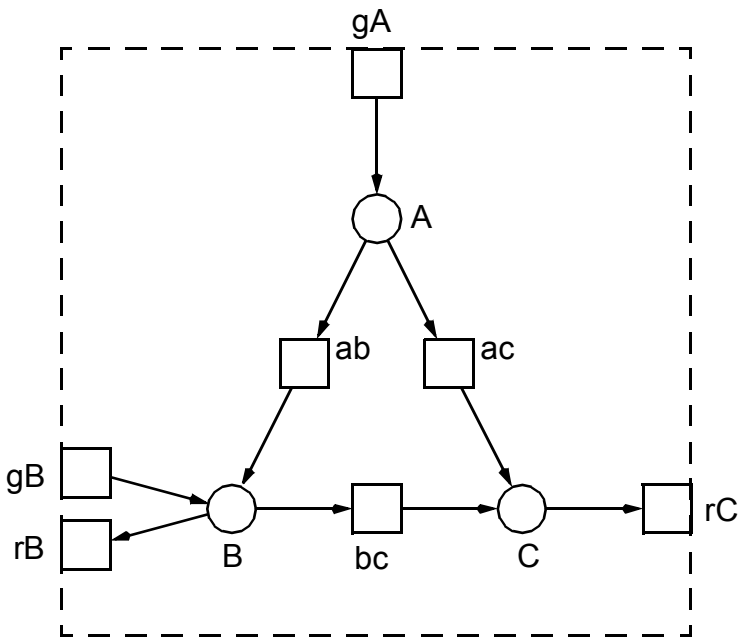
- **substances involved**
  - -> *input substance A*
  - -> *output substance C*
  - -> *auxiliary substance B*



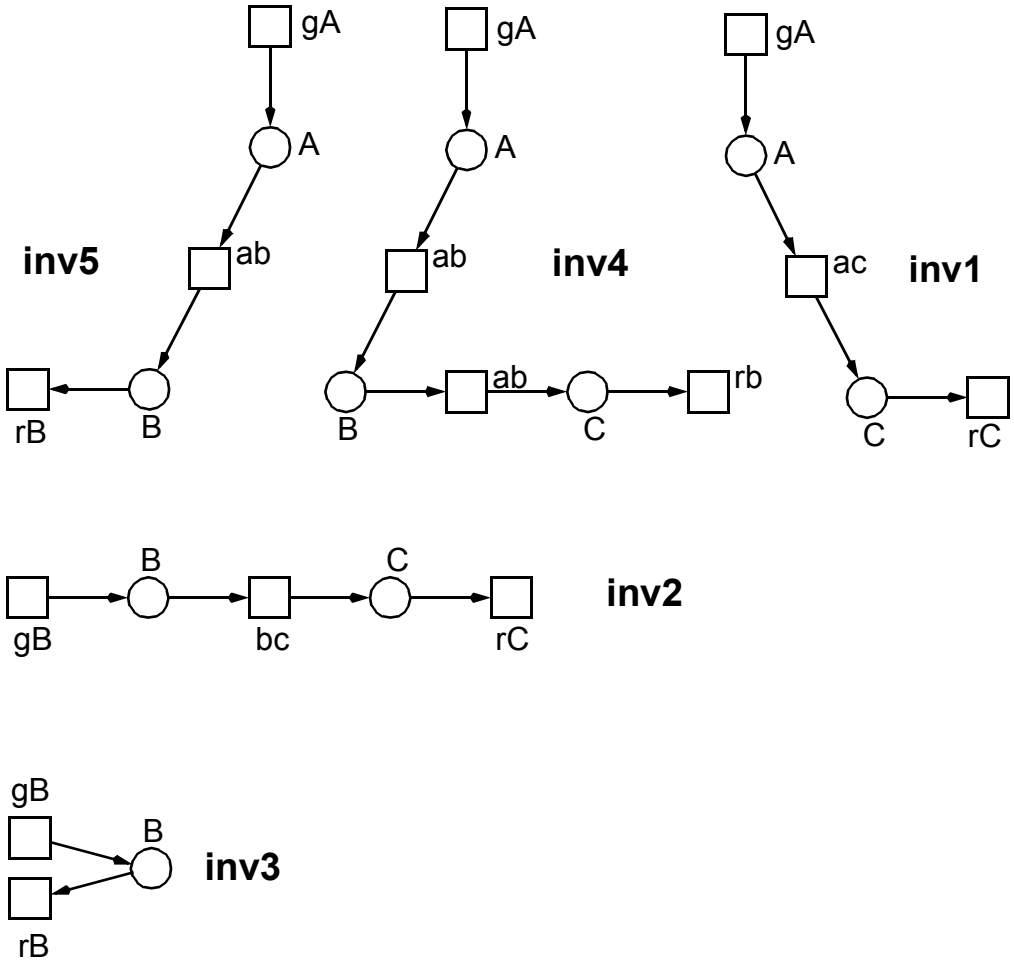
papin2003.spped

T-INVARIANTS, EX3

- **substances involved**
  - -> *input substance A*
  - -> *output substance C*
  - -> *auxiliary substance B*

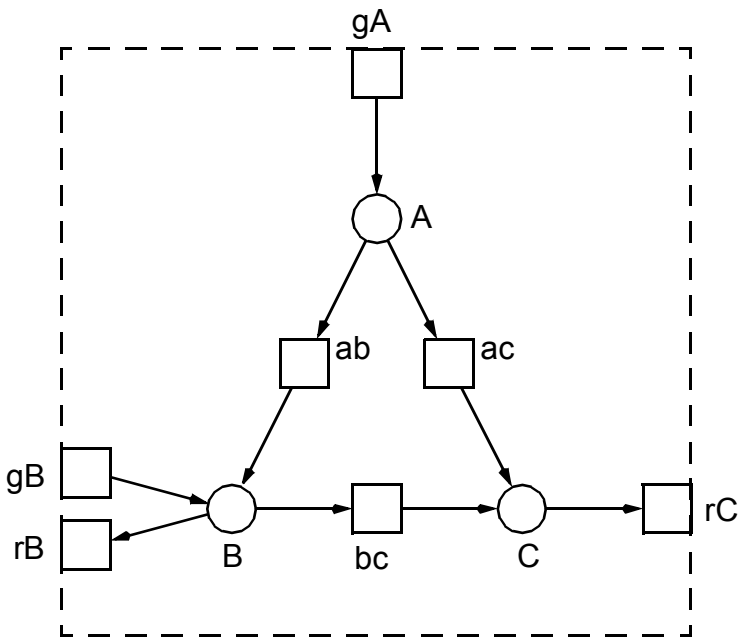


papin2003.spped

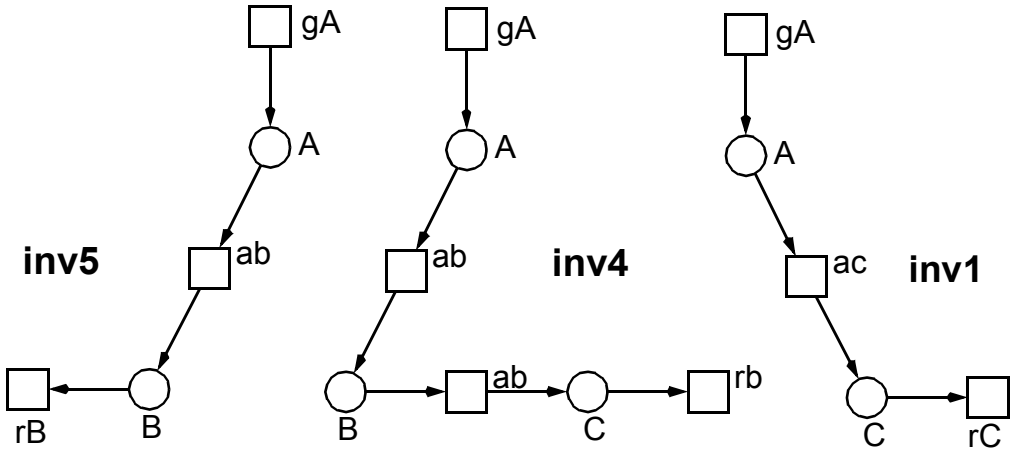


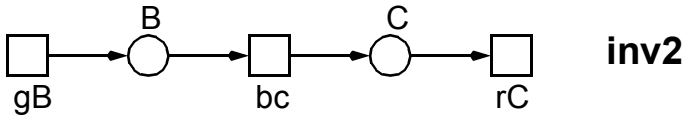
T-INVARIANTS, EX3

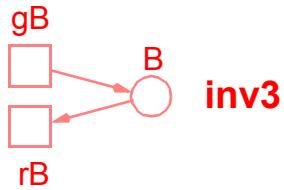
- **substances involved**
  - -> *input substance A*
  - -> *output substance C*
  - -> *auxiliary substance B*



papin2003.spped



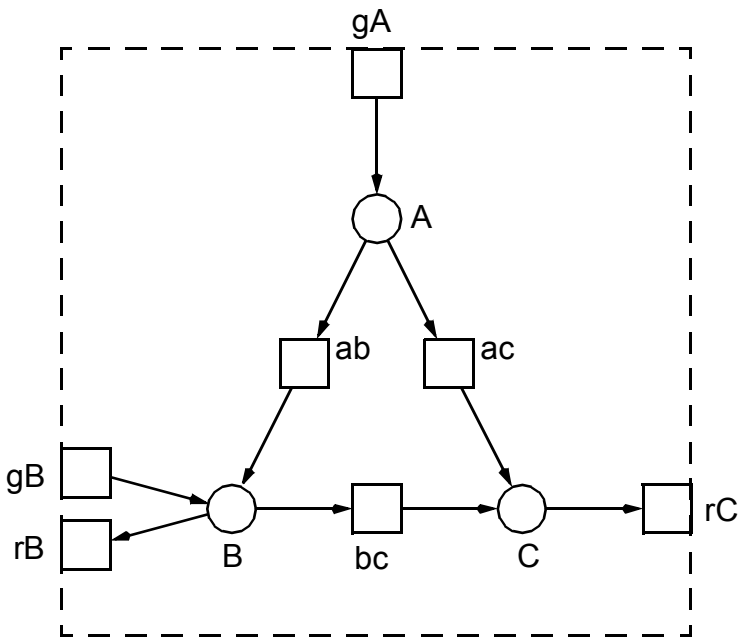




no elementary mode

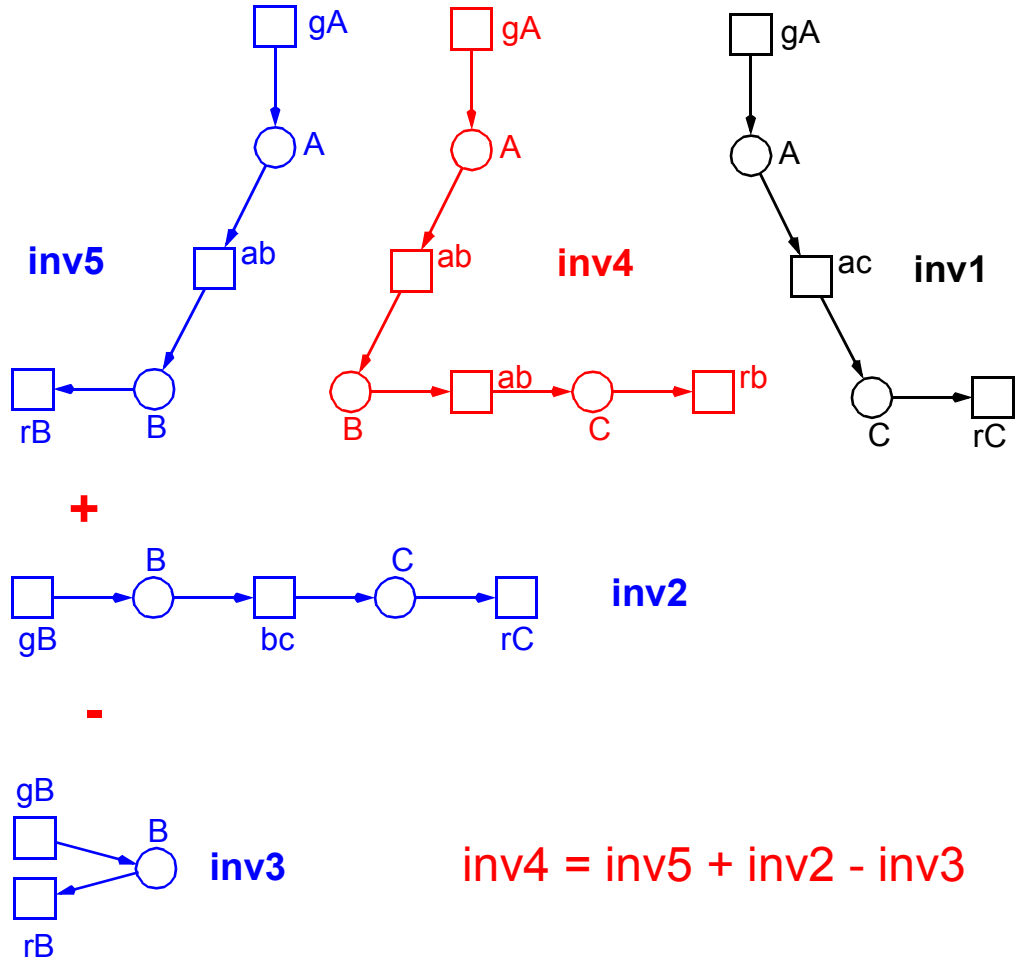
T-INVARIANTS, EX3

- **substances involved**
  - -> *input substance A*
  - -> *output substance C*
  - -> *auxiliary substance B*



papin2003.spped

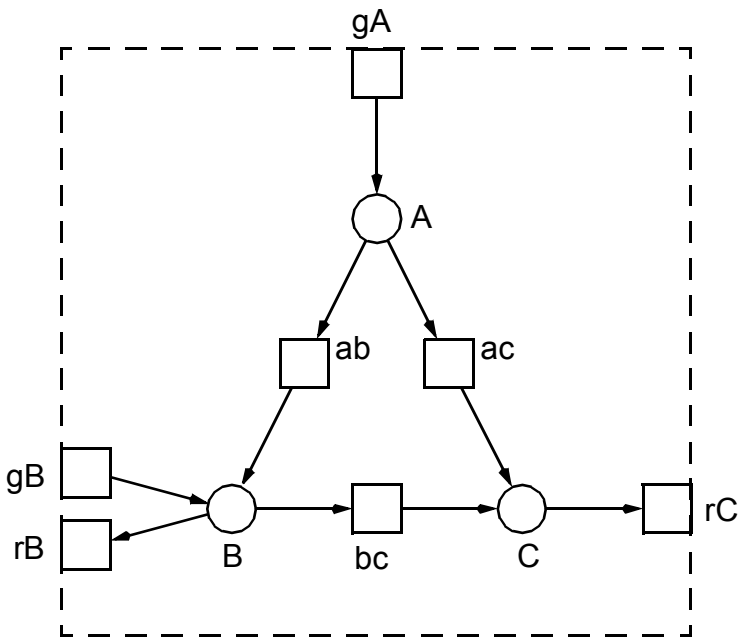
no extreme pathway



inv4 = inv5 + inv2 - inv3

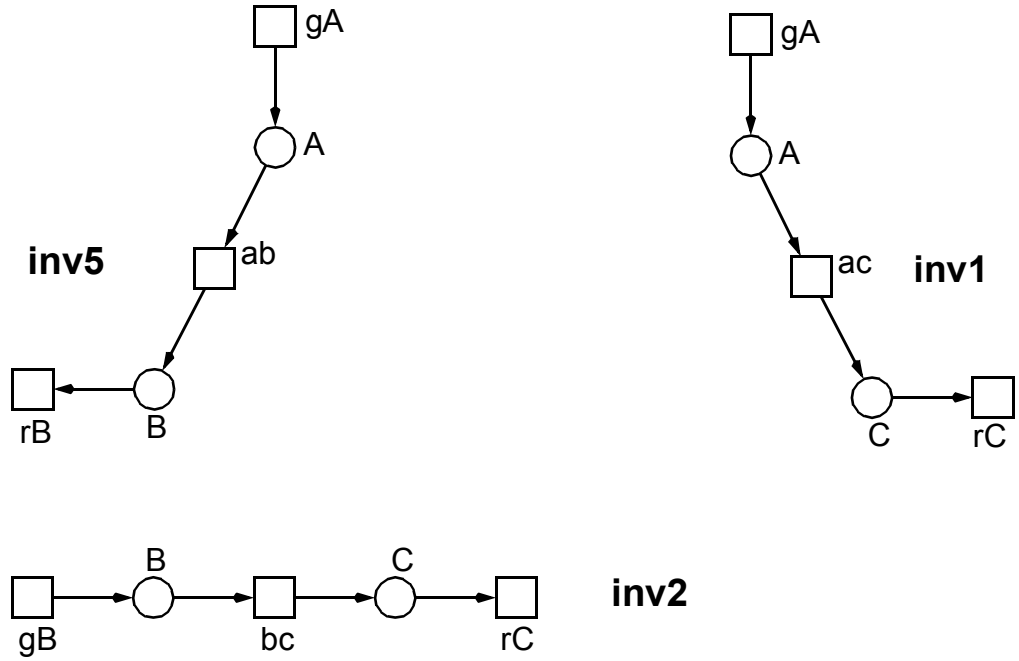
T-INVARIANTS, EX3, SUMMARY

- **substances involved**
  - -> *input substance A*
  - -> *output substance C*
  - -> *auxiliary substance B*



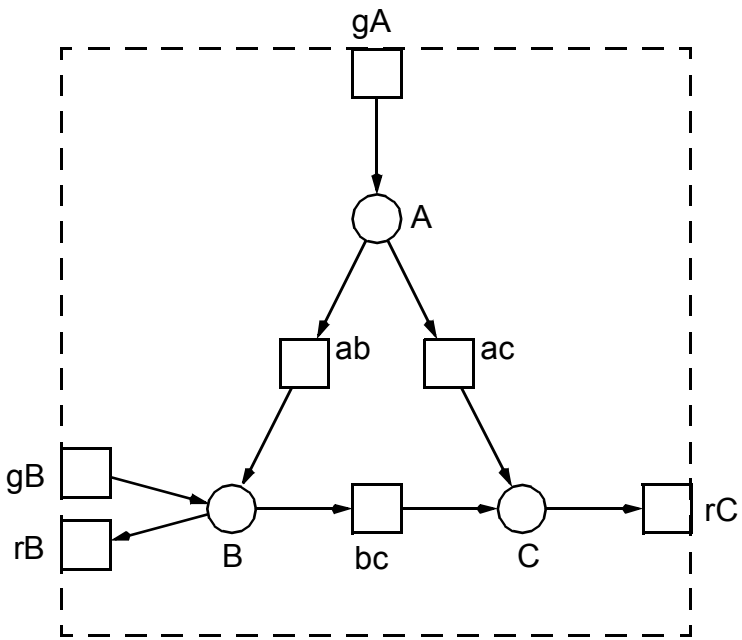
papin2003.spped

EXTREME PATHWAYS



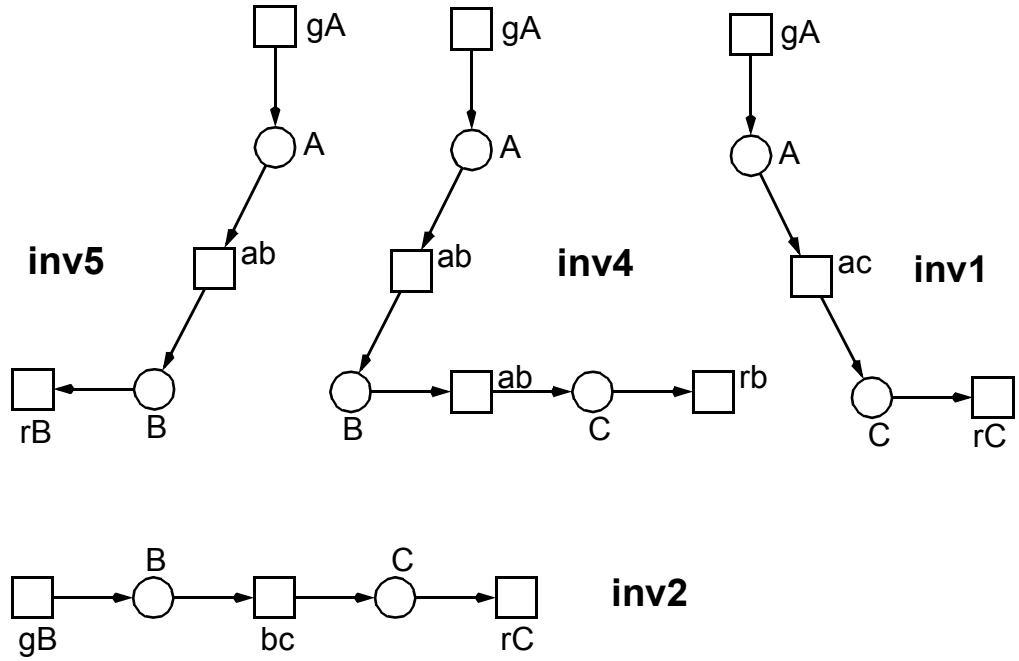
T-INVARIANTS, EX3, SUMMARY

- **substances involved**
  - -> *input substance A*
  - -> *output substance C*
  - -> *auxiliary substance B*



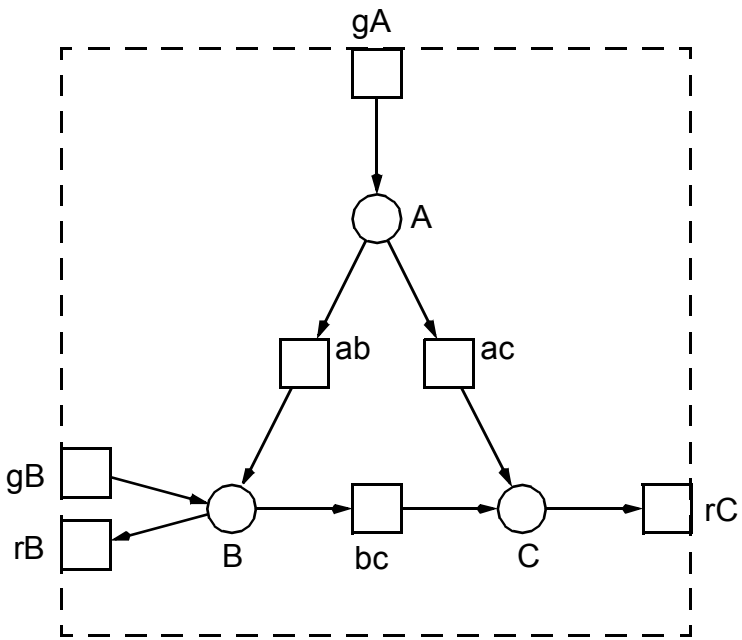
papin2003.spped

ELEMENTARY MODES



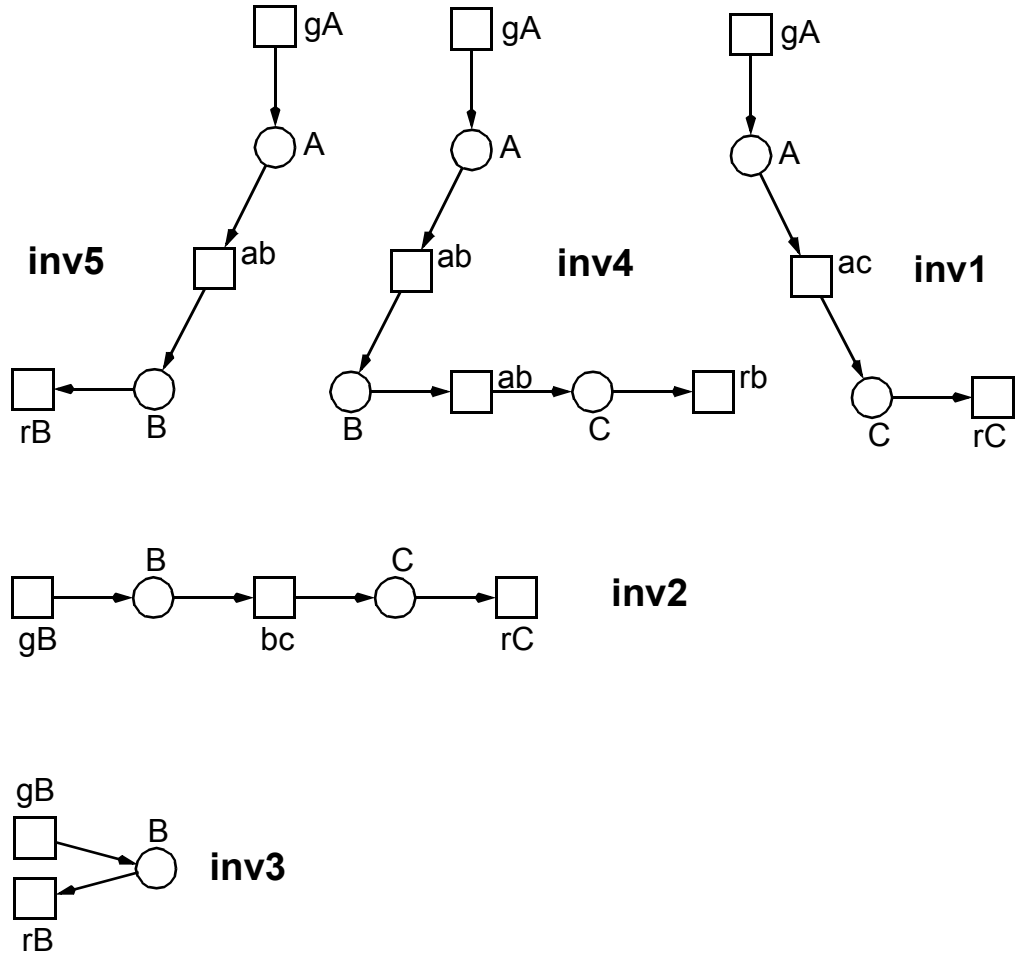
T-INVARIANTS, EX3, SUMMARY

- **substances involved**
  - -> *input substance A*
  - -> *output substance C*
  - -> *auxiliary substance B*



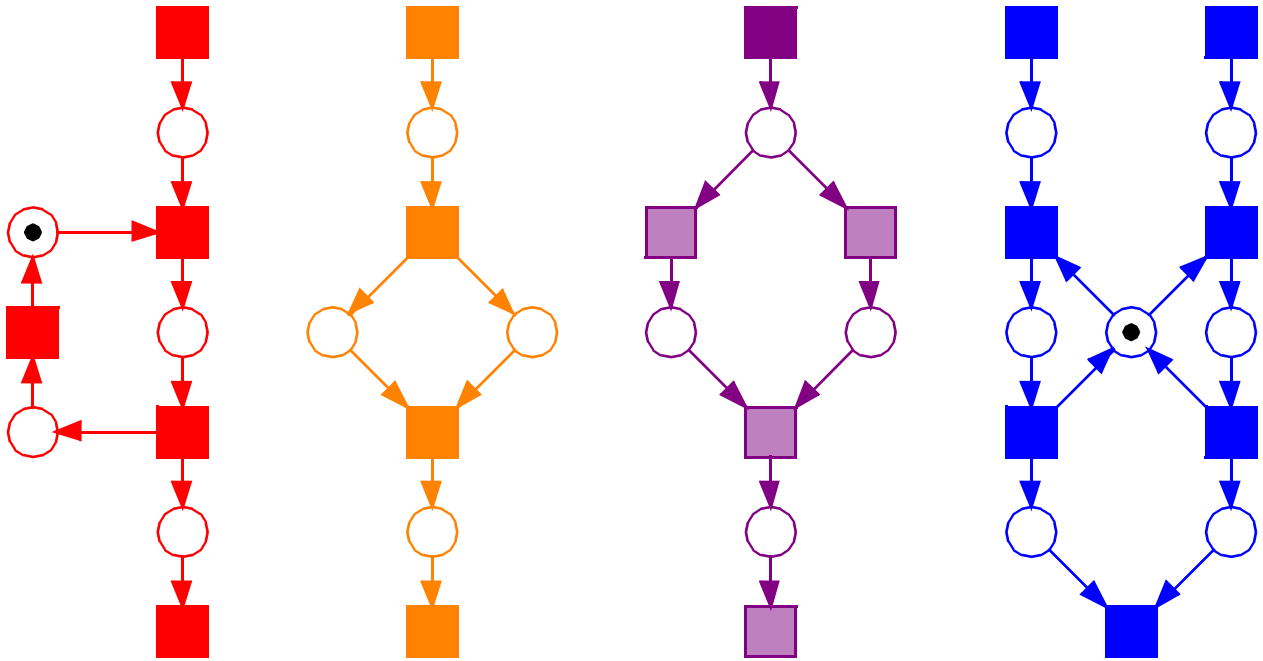
papin2003.spped

MINIMAL T-INVARIANTS



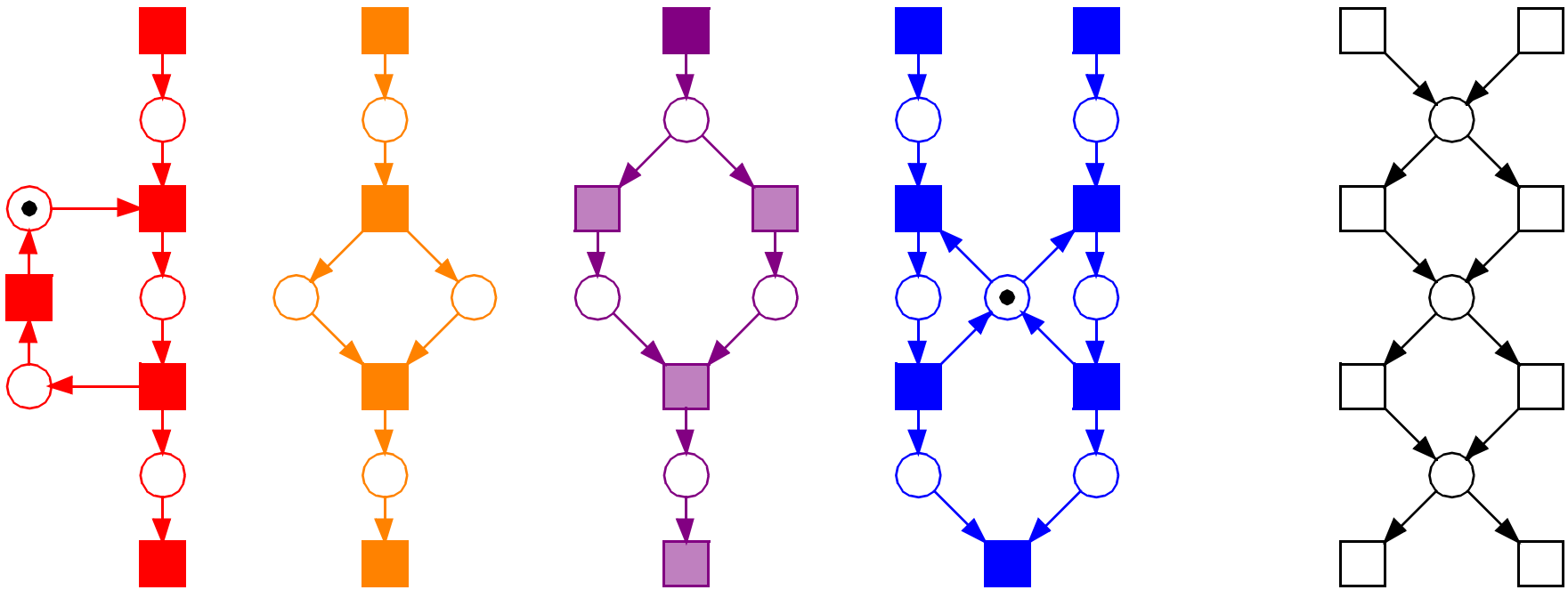
T-INVARIANTS, OBSERVATIONS

- **T-invariants may contain any structure**



T-INVARIANTS, OBSERVATIONS

- **T-invariants may contain any structure**



- **T-invariants generally overlap**

  -> combinatorial effect brings explosion in the number of min. T-invariants **(2^4)**

- **likewise for P-invariants**

STATIC ANALYSIS, SUMMARY

- **subnetwork identification**
  - -> *P-invariants: token-preserving modules (mass conservation)*
  - -> *T-invariants: state-repeating modules (elementary modes)*

STATIC ANALYSIS, SUMMARY

- **subnetwork identification**
  - -> *P-invariants: token-preserving modules (mass conservation)*
  - -> *T-invariants: state-repeating modules (elementary modes)*

- **network validation**
  - -> *CPI (if closed model), CTI*
  - -> *no minimal P/T-invariant without biological interpretation*
  - -> *no known mass conservation without corresponding P-invariant*
  - -> *no known biological behaviour without corresponding T-invariant*

STATIC ANALYSIS, SUMMARY

- **subnetwork identification**
  - -> *P-invariants: token-preserving modules (mass conservation)*
  - -> *T-invariants: state-repeating modules (elementary modes)*
- **network validation**
  - -> *CPI (if closed model), CTI*
  - -> *no minimal P/T-invariant without biological interpretation*
  - -> *no known mass conservation without corresponding P-invariant*
  - -> *no known biological behaviour without corresponding T-invariant*
- **construction of initial marking**

STATIC ANALYSIS, SUMMARY

- **subnetwork identification**
  - -> *P-invariants: token-preserving modules (mass conservation)*
  - -> *T-invariants: state-repeating modules (elementary modes)*
- **network validation**
  - -> *CPI (if closed model), CTI*
  - -> *no minimal P/T-invariant without biological interpretation*
  - -> *no known mass conservation without corresponding P-invariant*
  - -> *no known biological behaviour without corresponding T-invariant*
- **construction of initial marking**
- **sometimes decision of liveness**

STATIC ANALYSIS, SUMMARY

- **subnetwork identification**
  - -> *P-invariants: token-preserving modules (mass conservation)*
  - -> *T-invariants: state-repeating modules (elementary modes)*
- **network validation**
  - -> *CPI (if closed model), CTI*
  - -> *no minimal P/T-invariant without biological interpretation*
  - -> *no known mass conservation without corresponding P-invariant*
  - -> *no known biological behaviour without corresponding T-invariant*
- **construction of initial marking**
- **sometimes decision of liveness**
- **choice of stochastic analysis techniques**
  - -> *bounded - analysis techniques, esp. analytic model checking*
  - -> *unbounded - simulation techniques, esp. simulative model checking*

REFERENCES

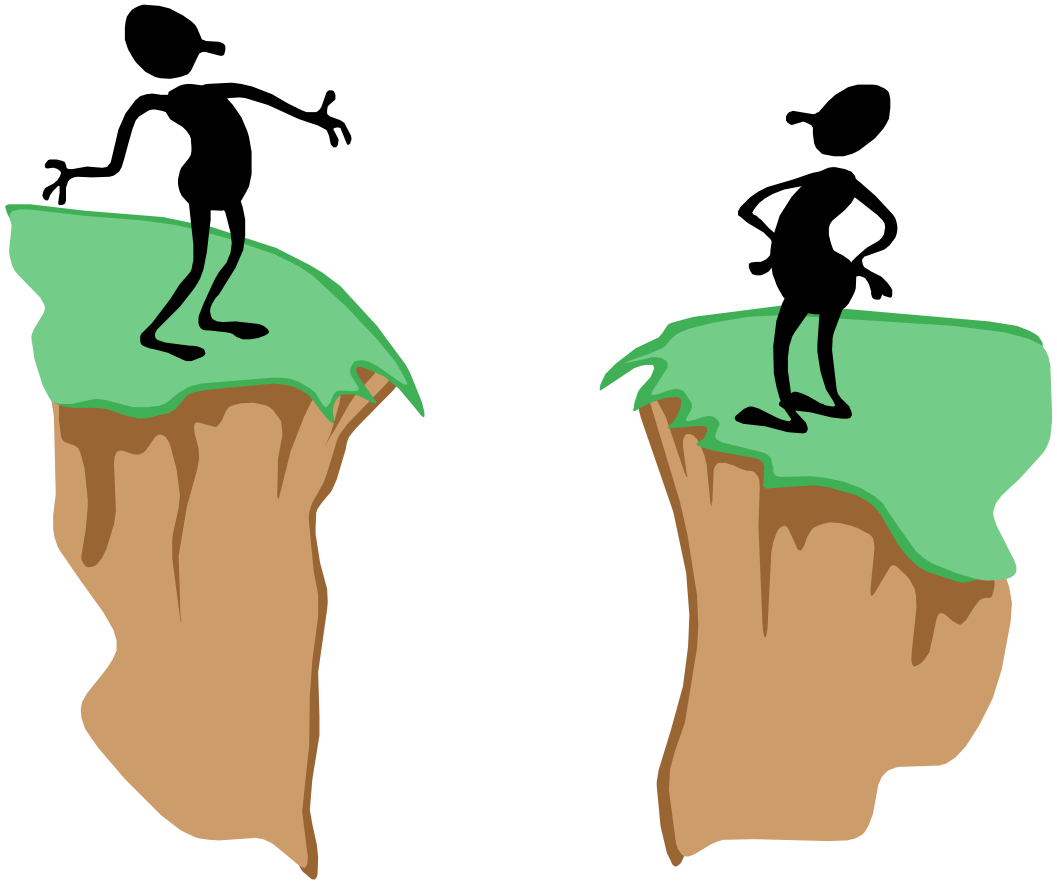
- M Heiner, D Gilbert, R Donaldson:
  Petri Nets for Systems and Synthetic Biology;
  in SFM 2008, Springer LNCS 5016, pp. 215-264, 2008.

- M Heiner:
  Understanding Network Behaviour by Structured Representations of Transition Invariants - A Petri Net Perspective on Systems and Synthetic Biology;
  in Algorithmic Bioprocesses; Chapter 19, Springer, July 2009.

- M Heiner, K Sriram:
  Structural Analysis to Determine the Core of Hypoxia Response Network;
  PLoS ONE 5(1): e8600, doi:10.1371/journal.pone.0008600, January 2010.

- M Heiner; R Donaldson; D Gilbert:
  Petri Nets for Systems Biology;
  in MS Iyengar (ed.): Symbolic Systems Biology: Theory and Methods,
  Jones & Bartlett Publishers, LLC, 2010.

THANKS !



HTTP://WWW-DSSZ.INFORMATIK.TU-COTTBUS.DE