THE MUTEX PATTERN - DIFFERENT ANALYSIS APPROACHES

QUALITATIVE ANALYSIS METHODS, OVERVIEW

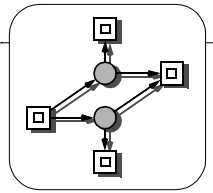
- NET REDUCTION
- STRUCTURAL PROPERTIES
- LINEAR PROGRAMMING
  - place / transition invariants
  - state equation
  - trap equation

static analysis

- REACHABILITY ANALYSIS
  - (complete) reachability graph
  - compressed state spaces
    - BDDs, NDDs, ..., xDDs
    - Kronecker products
  - reduced state spaces
    - coverability graph
    - symmetry
    - stubborn sets
  - branching process

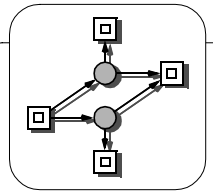
dynamic analysis

(model checking)



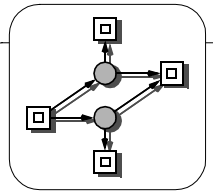
SEMAPHORE AS ADT

```
public class Semaphore {

    private int value;

    public Semaphore (int initial) {
        if (initial<0) {
            System.exit(1); // out of range error
        } else {
            value = initial;
        } // if
    } // Semaphore

    synchronized public void up() {
        ++value;
        notify();
    } // up

    synchronized public void down() {
        try {
            while (value==0) wait();
                --value;
        } catch(InterruptedException e){}
    } // down

} // class Semaphore
```

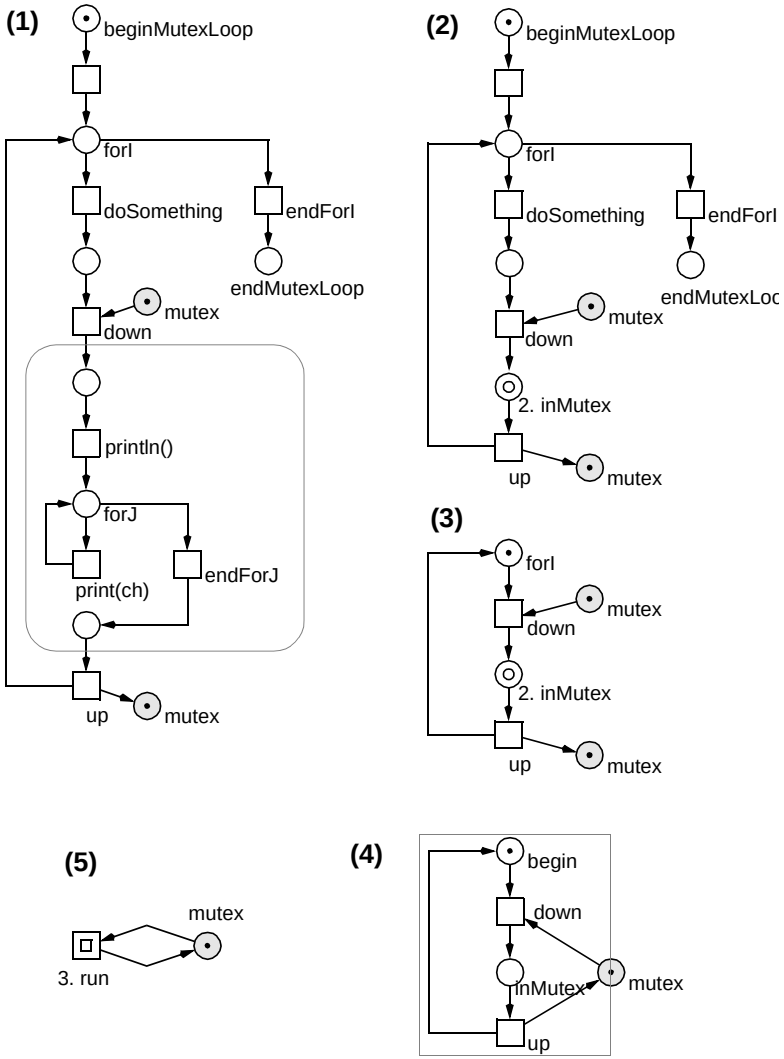



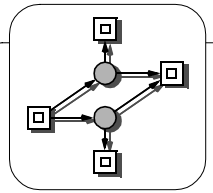
PATTERN OF BEHAVIOUR

```
class MutexLoop extends Thread {

    private Semaphore mutex;
    private char ch;

    MutexLoop (Semaphore sema, char toPrint) {
        mutex=sema;
        ch = toPrint;
    } // MutexLoop

    public void run() {
        for (int i=0;i<MutexDemo.LENGTH;i++)  {
            Simulate.doSomething();// pre action

            mutex.down(); // get mutual exclusion
                System.out.println();
                for (int j=0;j<MutexDemo.MAX;j++) {
                    System.out.print(ch);
                } // for
            mutex.up();   //release mutual exclusion

            Simulate.doSomething();// post action
        } // for

    } // run

} // class MutexLoop
```



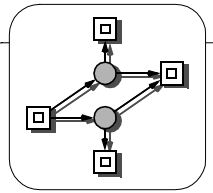
PETRI NET MODEL, PATTERN OF BEHAVIOUR





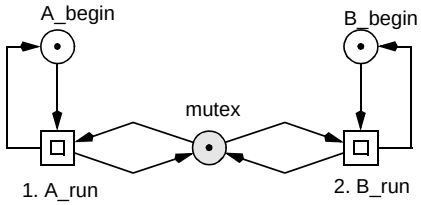
APPLICATION MUTEXDEMO

```
public class MutexDemo {

   static final int MAX = 30;// line width to weave
   static final int LENGTH = 50;// number of lines

   public static void main (String[] argv) {

      Semaphore mutex = new Semaphore(1);

      Thread a = new MutexLoop(mutex,'a');
      Thread b = new MutexLoop(mutex,'b');
      // Thread c = new MutexLoop(mutex,'c');

      System.out.println(" -- begin of MutexDemo -- ");

      a.start();
      b.start();
      // c.start();

   } // main

} // class MutexDemo
```

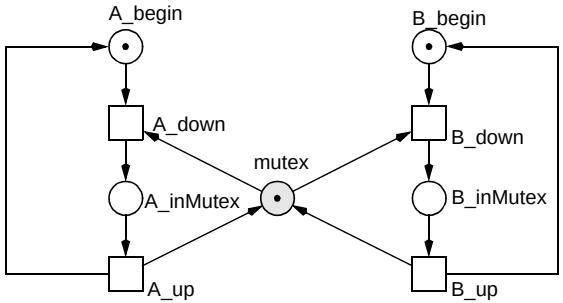


PETRI NET MODEL, MUTEXDEMO

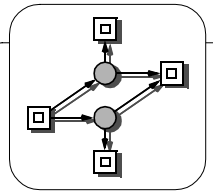
SYNCHRONIZATION SKELETON



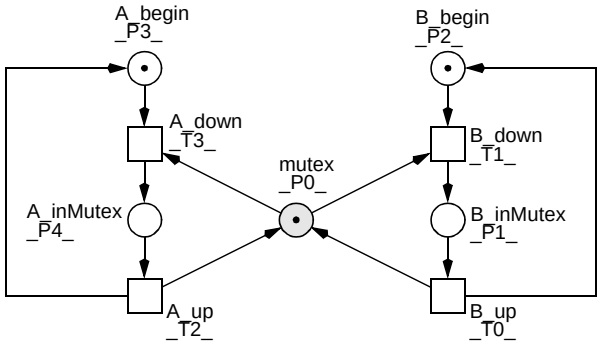
FLATTEN



ORD	HOM	NBM	PUR	CSV	SCF	CON	SC	Ft0	tF0	Fp0	pF0	MG	SM	FC	EFC	ES
Y	Y	Y	Y	N	N	Y	Y	N	N	N	N	N	N	N	N	Y
DTP	SMC	SMD	SMA	CPI	CTI	B	SB	REV	DSt	BSt	DTr	DCF	L	LV	L&S	
Y	Y	Y	Y	Y	Y	Y	Y	Y	N	?	N	N	Y	Y	Y	



MUTEXDEMO, ANALYSIS



REQUIRED SAFETY PROPERTY

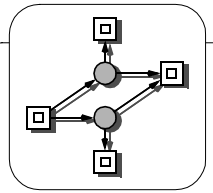
forever, there is at most one process in the mutex section (critical section)

HOW TO ANALYZE IT ?

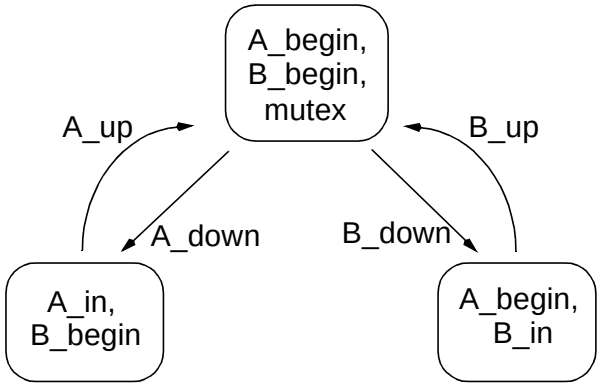
(1) reachability graph
(2) state equation
(3) p-invariants, non-reachability check
(4) p-invariants, reasoning

- - - - - - - - -

(5) model checking of temporal formulae



ANALYSIS OF MUTEXDEMO (1), REACHABILITY GRAPH

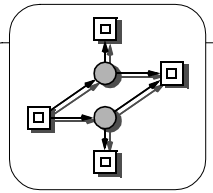


-> RG is finite
   -> BND

-> 1 SC component, containg all transitions
   -> LIVE & REV

-> no state with (A_in = 1 and B_in = 1) reachable
   -> safety property is valid

in temporal logics:
not EF (A_in and B_in)
AG (not(A_in and B_in))



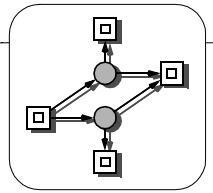
INCIDENCE MATRIX C- A REPRESENTATION OF THE NET STRUCTURE

P+T \ P+T	p1 card(P)	t1 card(T)
p1 card(P)	ϕ	- PRE
t1 card(T)	+ POST	ϕ

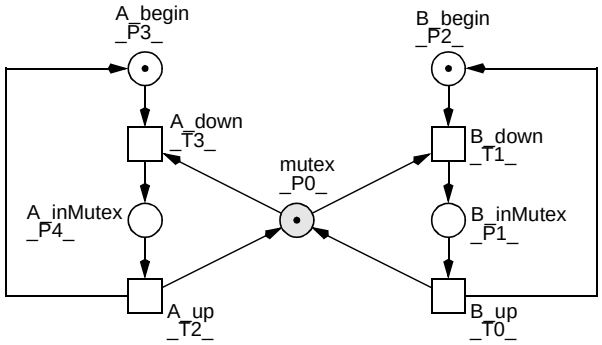
POST - PRE

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

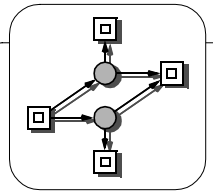
cij = (pi, tj) = F(tj,pi) - F(pi, tj) = Δ tj(pi)
-> token change
in place pi by firing of transition tj



MUTEXDEMO, INCIDENCE MATRIX

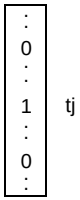


P \ T	B_up	B_down	A_up	A_down
mutex	+1	-1	+1	-1
B_inMutex	-1	+1	0	0
B_begin	+1	-1	0	0
A_begin	0	0	+1	-1
A_inMutex	0	0	-1	+1



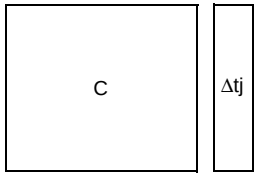
STATE EQUATION 1

PARIKH VECTOR
parikh(tj)



incidence matrix C

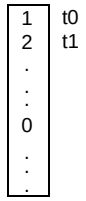
P \ T	t1	**tj**	tm
p1			
pi		**Δtj**	
pn			



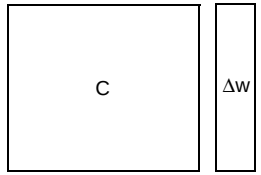
Δtj - vector describing the change of the whole marking by firing of tj

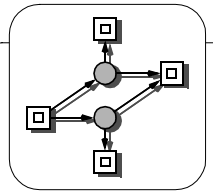
Let the word w = t1-t0-t1-. . .
be a sequence of firing transitions;

PARIKH VECTOR
parikh(w)



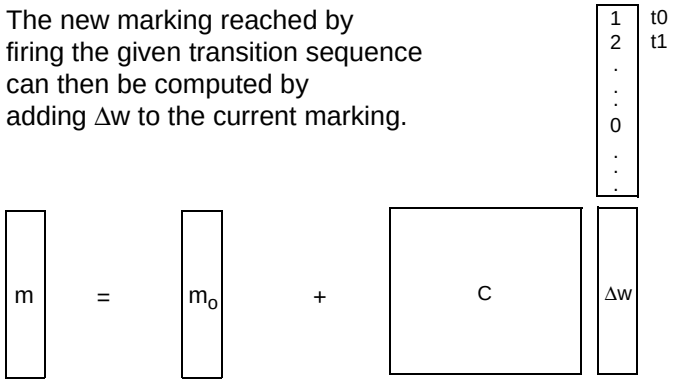
The change of the
marking Δw
by firing that sequence
can be computed
by multiplying
the incidence matrix C
with the
Parikh vector parikh(w)
of that transition sequence.





STATE EQUATION 2

The new marking reached by
firing the given transition sequence
can then be computed by
adding Δw to the current marking.



m_0 - initial marking
m - new marking reached by firing of w

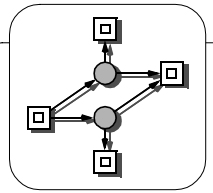
STATE EQUATION -> linear programming problem

$m = m_0 + C\,x$, x - T-vector
$x >= 0$

There exists an integer solution for every reachable marking m
(the Parikh vector of the transition sequence going to m).

-> the integer solvability is a necessary condition
for the reachability of a marking;

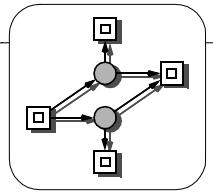
-> **NON-REACHABILITY CHECK**
if there is no integer solution,
then the marking is not reachable.



ANALYSIS OF MUTEXDEMO (2), STATE EQUATION

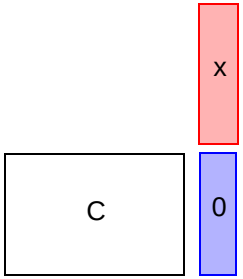
m		m_0		C * x				x
0		1		+1	-1	+1	-1	x0
1		0		-1	+1	0	0	x1
0	=	1	+	+1	-1	0	0	x2
0		1		0	0	+1	-1	x3
1		0		0	0	-1	+1	

-1	=	+ x0	- x1	+ x2	- x3
1	=	- x0	+ x1		
-1	=	+ x0	- x1		
-1	=			+ x2	- x3
1	=			- x2	+ x3



T-INVARIANTS

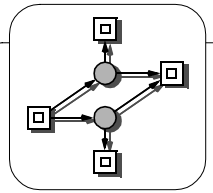
- Lautenbach, 1973
- T-invariants
  - -> *integer solutions x of* $Cx = 0, x \neq 0, x \geq 0$
  - -> *Parikh vector*
  - -> *exponential complexity*



- **minimal** T-invariants
  - -> *there is no T-invariant with a smaller support*
  - -> *greatest common divisor (gcd) of all entries is 1*
- support
  - -> *set of transitions belonging to the T-invariant*
- any T-invariant is a non-negative linear combination of minimal ones
  - -> *multiplication with a positive integer*
  - -> *addition*
  - -> *division by gcd*

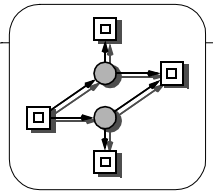
$$kx = \sum_i a_i x_i$$

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



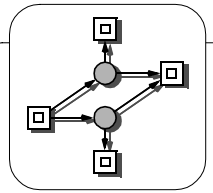
T-INVARIANTS, INTERPRETATION

- T-invariants = (multi-) sets of transitions
  - -> *zero effect on marking*
  - -> *reproducing a marking / system state*
  - -> *steady state substance flows*
  - -> *elementary modes, Schuster 1993*
- the T-invariant corresponds to cycles in the RG, if the T-invariant is realizable
- in the RG, concurrency of transitions is described by all transitions' interleaving sequences
- if there are concurrent transitions in a realizable T-invariant, then there is a RG cycle for each interleaving sequence
  - -> *T-inv3*, *T-inv4*
- pre-sets of supports = post-sets of supports
- a T-invariant defines a connected subnet
  - -> *the T-invariant's transitions (the support),*
    + all their pre- and post-places
    + the arcs in between



T-INVARIANTS, THEOREMS

- **If a bounded net is live, then it is CTI.**
  -> NECESSARY CONDITION FOR well-formedness



T-INVARIANTS, MUTEXDEMO

- INA, session protocol

```
1  | 0.B_up       :  1,
   | 1.B_down     :  1

2  | 2.A_up       :  1,
   | 3.A_down     :  1
```

- interpretation:

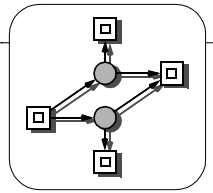
  T-invariant1
  reproduces m_0 by a cyclic run of process B

  T-invariant2
  reproduces m_0 by a cyclic run of process A

  -> *both T-invariants are cycles in the reachability graph*

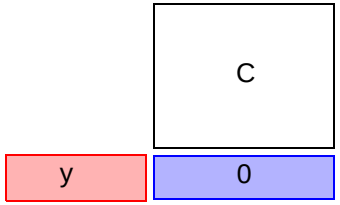
  -> *both T-invariants are realizable*

- CTI



P-INVARIANTS

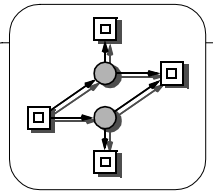
- Lautenbach, 1973
- P-invariants
  - -> *integer solutions y of* $yC = 0, y \neq 0, y \geq 0$
  - -> *exponential complexity*



- **minimal** P-invariants
  - -> *there is no P-invariant with a smaller support*
  - -> *gcd of all entries is 1*
- support
  - -> *set of places belonging to the P-invariant*
- any P-invariant is a non-negative linear combination of minimal ones
  - -> *multiplication with a positive integer*
  - -> *addition*
  - -> *division by gcd*

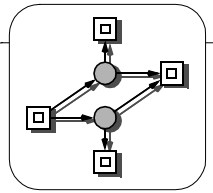
$$ky = \sum_i a_i y_i$$

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



P-INVARIANTS, INTERPRETATION

- set of places with
  - -> *a constant weighted sum of tokens*
    ym = ym_0
    for all from m_0 reachable markings m
  - -> *token / compound preservation*
- a place belonging to a P-invariant is bounded
  - -> *CPI - sufficient condition for BND*
- the firing of any transition has no influence on the weighted sum of tokens on the P-invariant's places
  - -> *for all transition t:*
    the effect of the arcs,
    removing tokens from a P-invariant's place
    is equal to the effect of the arcs,
    adding tokens to a P-invariant's place
- pre-sets of supports = post-sets of supports
- a P-invariant defines a connected subnet
  - -> *the P-invariant's places (the support),*
    + all their pre- and post-transitions
    + the arcs in between



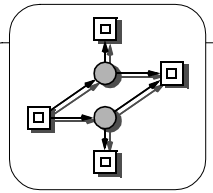
P-INVARIANTS, THEOREMS

- a place belonging to a P-invariant is bounded

- a net is **covered** by P-invariants (CPI)
  -> each place belongs to a P-invariant
  -> SUFFICIENT CONDITION FOR BND

- non-reachability check of m
  -> if there is a P-invariant y with
  $ym_0 \neq ym$,
  then m is not reachable from m_0.
  -> SUFFICIENT CONDITION FOR NON-REACHABILITY OF M

- sub-P-invariants
  -> $yC \leq 0, y \neq 0, y \geq 0$
  -> *covered by sub-P-invariants*
  -> NECESSARY AND SUFFICIENT CONDITION FOR STRUCTRURALLY BND



ANALYSIS OF MUTEXDEMO (3), NON-REACHABILITY CHECK

- P-Invariants/ Charlie (INA) session protocol

```
1  | 1.B_inMutex     : 1,
   | 2.B_begin       : 1

2  | 3.A_begin       : 1,
   | 4.A_inMutex     : 1

3  | 0.mutex         : 1,
   | 1.B_inMutex     : 1,
   | 4.A_inMutex     : 1
```

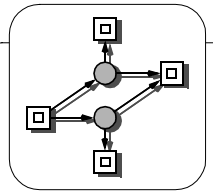
- m_0 = (A_begin, B_begin, mutex),
  m = (A_inMutex, B_inMutex)

  m_0 * y3 = m * y3 ?

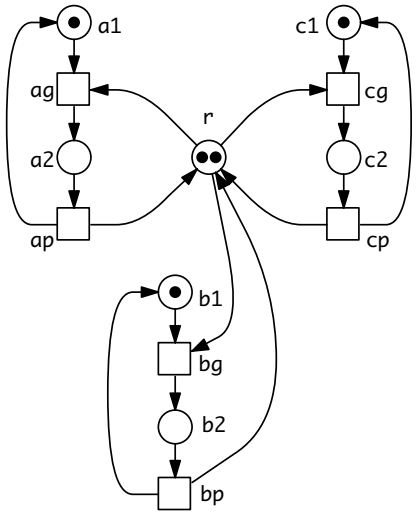
  m_0 * y3 = 1
  m * y3 = 2

  -> the marking m is not reachable

- NOTE:
  If the equation is fullfilled for all P-invariants,
  we know **N O T H I N G**
  concerning the reachability of the marking!



ALL MINIMAL (POSITIVE) P/ T- INVARIANTS



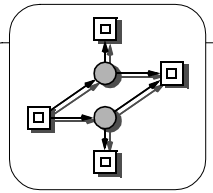
- T - invariants

  (ag, ap), (bg, bp), (cg, bp)
  -> local process cycles reproducing m_o

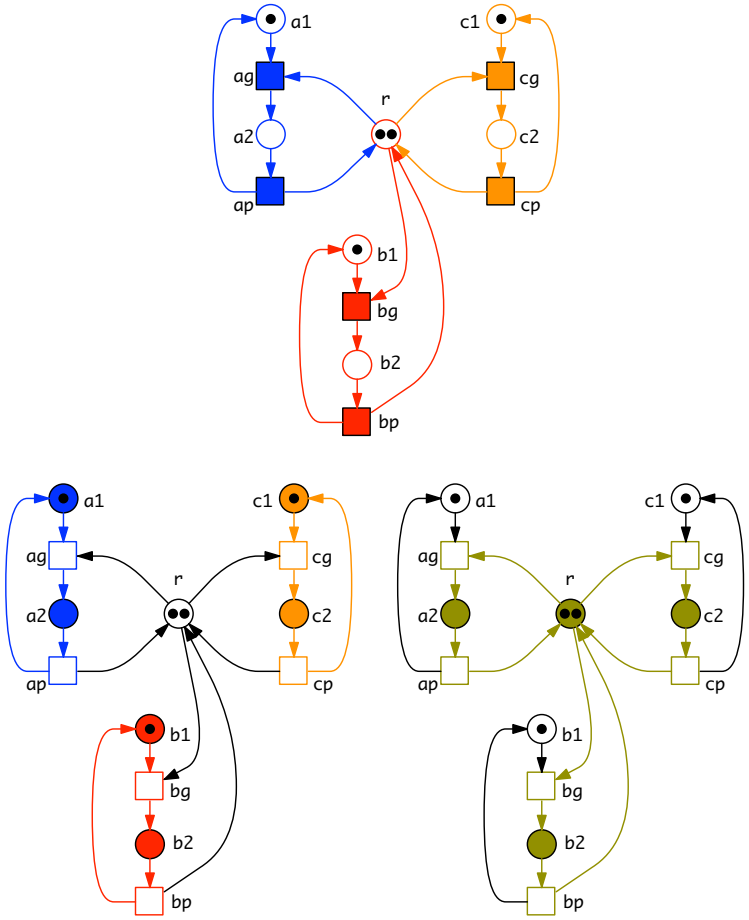
- P - invariants

  (a1, a2), (b1, b2), (c1, c2)
  -> possible local process states

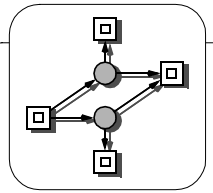
  (a2, b2, c2, r)
  -> a resource is either free or occupied



ALL MINIMAL (POSITIVE) P/ T- INVARIANTS



remark:
invariants computed with Charlie can be visualised with Snoopy



ZANALYSIS OF MUTEXDEMO (4), REASONING WITH P-INVARIANTS

- sometimes it is possible to use
  P-invariants for logical reasoning
  to prove a property in question;

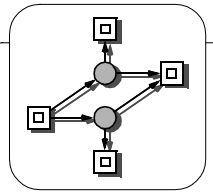
- example:
  y3 is a so-called 1-P-invariant
  (the constant token sum is 1)

  *because y3 * m_o = 1,*
  y3 = (mutex, B_inMutex, A_inMutex)
  m_o = (A_begin, B_begin, mutex)

- consequently,
  either mutex **xor** A_inMutex **xor** B_inMutex
  can carry a token at any time;

  -> they can never carry a token at the same time

  -> the mutex property is fullfilled



UNREACHABILITY ANALYSIS, COMPARISON

- reachability graph

  pros: necessary & sufficient condition,
  rg allows also the decision of other properties,
  also sub-markings may be checked;

  cons: uncontrolled growth
  (potentially beyond exponential growth)

- state equation

  pros: static analysis technique
  (size of the state space does not matter);

  cons: only sufficient condition,
  no sub-markings can be checked;

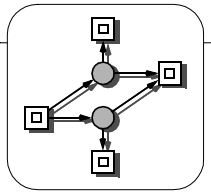
- p-invariants, token conservation

  pro/cons: see above

- p-invariants, reasoning

  pros: logical faults/hints for the reachability
  may be found

  cons: limited by human skills in reasoning,
  not applicable for larger nets with larger invariants



REFERENCES

[Desel 1998]
Desel, J.:
Petrinetze, lineare Algebra und lineare Programmierung;
B. G. Teubner 1998.
-> additional material, not discussed here

[Heiner 2008]
M Heiner, D Gilbert and R Donaldson:
Petri Nets for Systems and Synthetic Biology;
SFM 2008, Bertinoro, Springer, LNCS 5016, pages 215–264.

[Heiner 2009]
M Heiner:
Understanding Network Behaviour by Structured Representations of Transition Invariants –
A Petri Net Perspective on Systems and Synthetic Biology;
Algorithmic Bioprocesses, Springer, pages 367–389, 2009.

[Starke 1990]
Starke, P. H.:
Analyse von Petri-Netz-Modellen;
B. G. Teubner 1990.