ERROR - CORRECTING PETRI NETS

Monika Heiner

**BTU Cottbus,
Dep. of Computer Science**

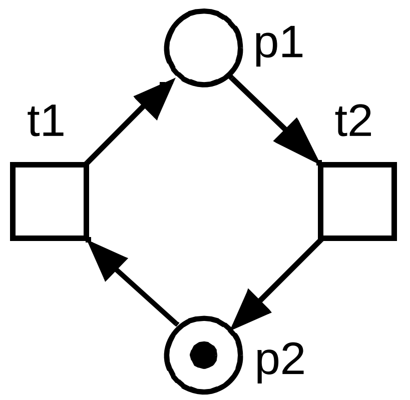
OBJECTIVE

- **fault-tolerant system design**
  - -> *error detection*
  - -> *error correction*
- **fault model**
  - -> *local changes in system state*
  - -> *(occasional) observations of a running system*
- **disturbances**
  - -> *measurement errors, wrong read-out*
  - -> *external influences, unexpected changes in the actual system*
  - -> *deterioration, unavoidable changes*
  - -> *. . . not-designed system states*
- **localize and correct detected error(s) in observed system state**
  - -> *general procedure*

Modulo
Petri nets

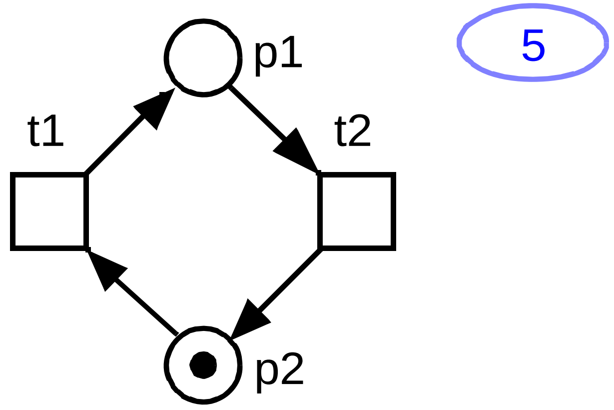
MODULO PETRI NETS, EXAMPLE

- **standard place/transition nets**



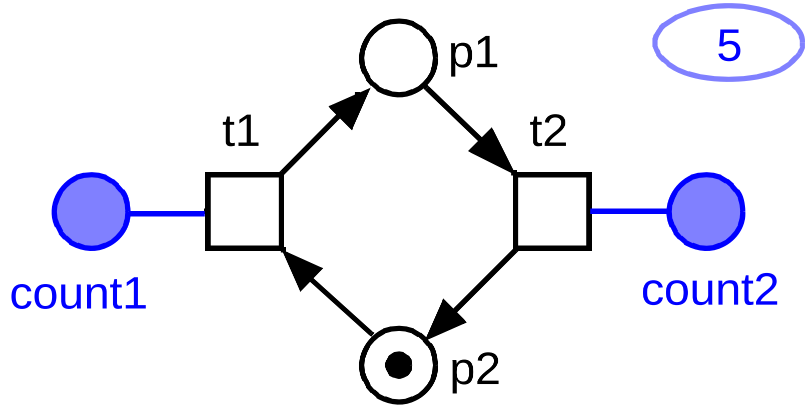
MODULO PETRI NETS, EXAMPLE

- **standard place/transition nets**
- **net-global modulo number p**
  -> prime



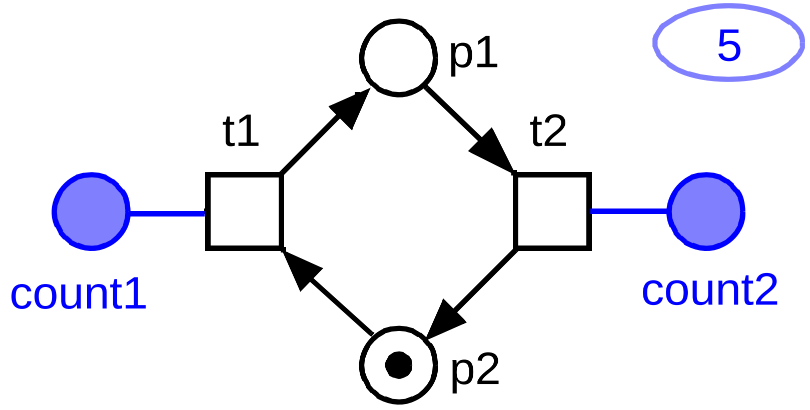
MODULO PETRI NETS, EXAMPLE

- **standard place/transition nets**
- **net-global modulo number p**
  - -> *prime*
- **undirected arcs**
  - -> *no influence on a transition's enabledness*
  - -> *add token number modulo p*

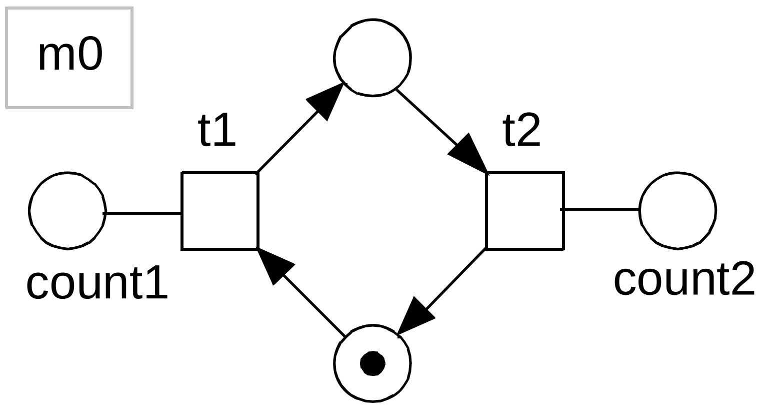


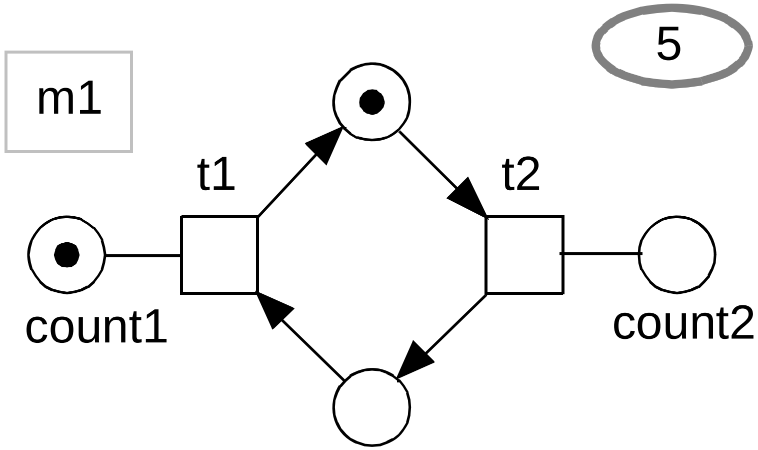
MODULO PETRI NETS, EXAMPLE

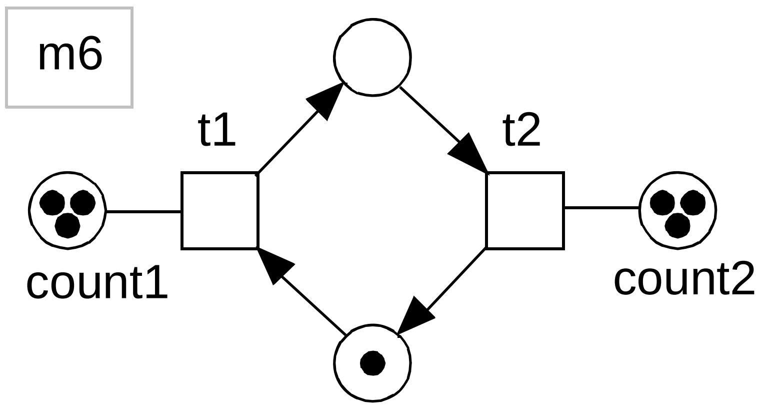
- **standard place/transition nets**
- **net-global modulo number p**
  - -> *prime*
- **undirected arcs**
  - -> *no influence on a transition's enabledness*
  - -> *add token number modulo p*
- **each place is connected with arcs of one type only**
  - -> *standard places (directed arcs)*
  - -> *control places (undirected arcs)*

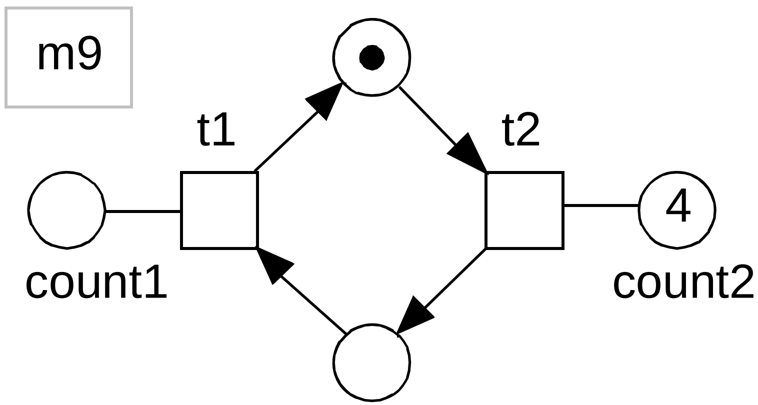


MODULO PETRI NETS, EXAMPLE









Modulo Petri Nets, Example

	t1	t2	m0
p1	1	-1	0
p2	-1	1	1
count1	1	0	0
count2	0	1	0

MODULO PETRI NETS, EXAMPLE

	t1	t2	m0
p1	1	-1	0
p2	-1	1	1
count1	1	0	0
count2	0	1	0

MODULO PETRI NETS, EXAMPLE

	t1	t2	m0	m1
p1	1	-1	0	1
p2	-1	1	1	0
count1	1	0	0	1
count2	0	1	0	0

Modulo Petri Nets, Example

	t1	t2	m0	m1
p1	1	-1	0	1
p2	-1	1	1	0
count1	1	0	0	1
count2	0	1	0	0

MODULO PETRI NETS, EXAMPLE

	t1	t2	m0	m1	m2
p1	1	-1	0	1	0
p2	-1	1	1	0	1
count1	1	0	0	1	1
count2	0	1	0	0	1

MODULO PETRI NETS, EXAMPLE

	t1	t2	m0	m1	m2	m3	m4	m5	m6	m7	m8	m9
p1	1	-1	0	1	0	1	0	1	0	1	0	1
p2	-1	1	1	0	1	0	1	0	1	0	1	0
count1	1	0	0	1	1	2	2	3	3	4	4	0
count2	0	1	0	0	1	1	2	2	3	3	4	4

MODULO PETRI NETS, EXAMPLE

	t1	t2	m0	m1	m2	m3	m4	m5	m6	m7	m8	m9
p1	1	-1	0	1	0	1	0	1	0	1	0	1
p2	-1	1	1	0	1	0	1	0	1	0	1	0
count1	1	0	0	1	1	2	2	3	3	4	4	0
count2	0	1	0	0	1	1	2	2	3	3	4	4

let’s count modulo . . .

MODULO PETRI NETS, EXAMPLE

	t1	t2	m0	m1	m2	m3	m4	m5	m6	m7	m8	m9
p1	1	-1	0	1	0	1	0	1	0	1	0	1
p2	-1	1	1	0	1	0	1	0	1	0	1	0
count1	1	0	0	1	1	2	2	3	3	4	4	0
count2	0	1	0	0	1	1	2	2	3	3	4	4

let's count modulo . . . 2 5

	t1	t2
p1	1	1
p2	1	1
count1	1	0
count2	0	1

MODULO PETRI NETS, EXAMPLE

	t1	t2	m0	m1	m2	m3	m4	m5	m6	m7	m8	m9
p1	1	-1	0	1	0	1	0	1	0	1	0	1
p2	-1	1	1	0	1	0	1	0	1	0	1	0
count1	1	0	0	1	1	2	2	3	3	4	4	0
count2	0	1	0	0	1	1	2	2	3	3	4	4

let's count modulo . . . 2 5

	t1	t2	m0	m1
p1	1	1	0	1
p2	1	1	1	0
count1	1	0	0	1
count2	0	1	0	0

Modulo Petri Nets, Example

	t1	t2	m0	m1	m2	m3	m4	m5	m6	m7	m8	m9
p1	1	-1	0	1	0	1	0	1	0	1	0	1
p2	-1	1	1	0	1	0	1	0	1	0	1	0
count1	1	0	0	1	1	2	2	3	3	4	4	0
count2	0	1	0	0	1	1	2	2	3	3	4	4

let's count modulo . . . 2 5

	t1	t2	m0	m1
p1	1	1	0	1
p2	1	1	1	0
count1	1	0	0	1
count2	0	1	0	0

Modulo Petri Nets, Example

	t1	t2	m0	m1	m2	m3	m4	m5	m6	m7	m8	m9
p1	1	-1	0	1	0	1	0	1	0	1	0	1
p2	-1	1	1	0	1	0	1	0	1	0	1	0
count1	1	0	0	1	1	2	2	3	3	4	4	0
count2	0	1	0	0	1	1	2	2	3	3	4	4

let's count modulo . . . 2 5

	t1	t2	m0	m1	m2
p1	1	1	0	1	0
p2	1	1	1	0	1
count1	1	0	0	1	1
count2	0	1	0	0	1

MODULO PETRI NETS, EXAMPLE

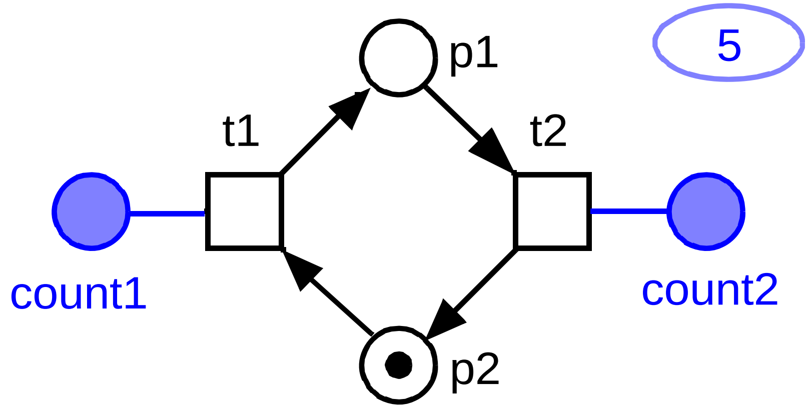
	t1	t2	m0	m1	m2	m3	m4	m5	m6	m7	m8	m9
p1	1	-1	0	1	0	1	0	1	0	1	0	1
p2	-1	1	1	0	1	0	1	0	1	0	1	0
count1	1	0	0	1	1	2	2	3	3	4	4	0
count2	0	1	0	0	1	1	2	2	3	3	4	4

let's count modulo . . . 2 5

	t1	t2	m0	m1	m2	m3	m4	m5	m6	m7	m8	m9
p1	1	1	0	1	0	1	0	1	0	1	0	1
p2	1	1	1	0	1	0	1	0	1	0	1	0
count1	1	0	0	1	1	2	2	3	3	4	4	0
count2	0	1	0	0	1	1	2	2	3	3	4	4

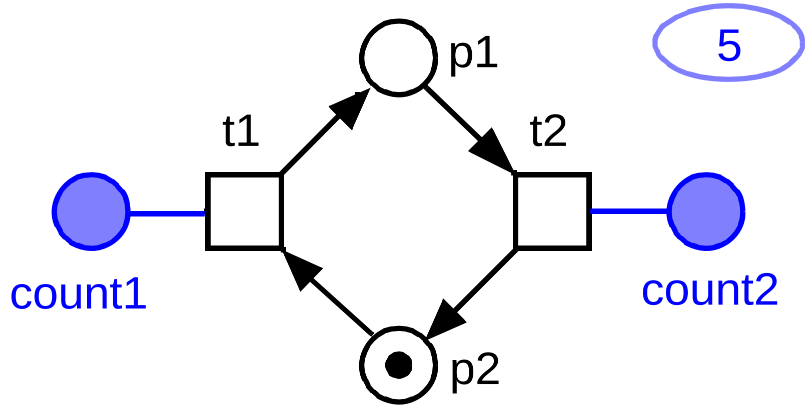
MODULO PETRI NETS, EXAMPLE - SUMMARY

- **any reachable marking is a linear combination of**
  - -> *m0*
  - -> *columns of incidence matrix*



MODULO PETRI NETS, EXAMPLE - SUMMARY

- **any reachable marking is a linear combination of**
  - -> *m0*
  - -> *columns of incidence matrix*
- **control places count (modulo) the transitions' occurrences**



The two places count1, count2 count modulo 5

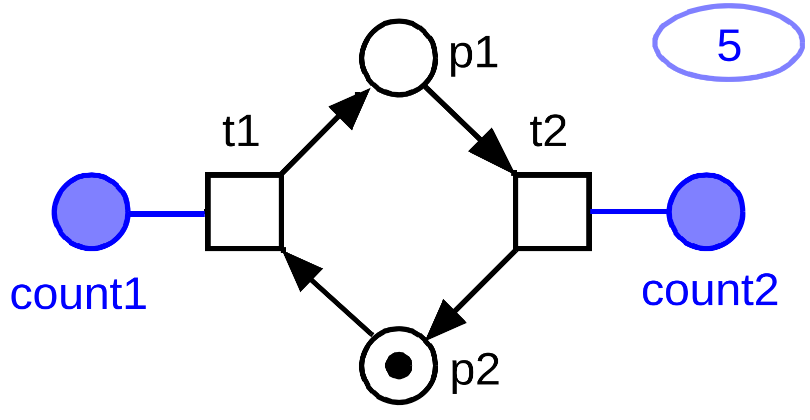
the number of the transitions' occurrences.

Obviously, they can differ by 1 (modulo 5) in any reachable marking only.

- **any reachable marking is a linear combination of**
  - -> *m0*
  - -> *columns of incidence matrix*
- **control places count (modulo) the transitions' occurrences**
- **How to extend the net structure by control places**

  - to detect -
  - to correct -

  erroneous markings ?



The two places count1, count2 count modulo 5

the number of the transitions' occurrences.

Obviously, they can differ by 1 (modulo 5) in any reachable marking only.

- **any reachable marking is a linear combination of**
  - -> *m0*
  - -> *columns of incidence matrix*

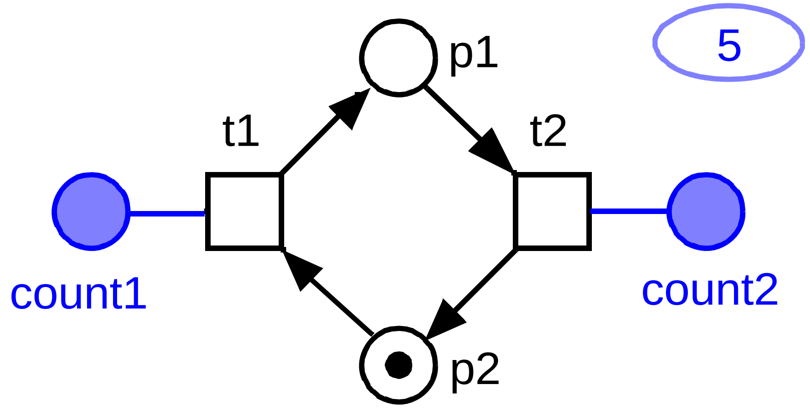
- **control places count (modulo) the transitions' occurrences**

- **How to extend the net structure by control places**

  - to detect -
  - to correct -

erroneous markings ?

????????

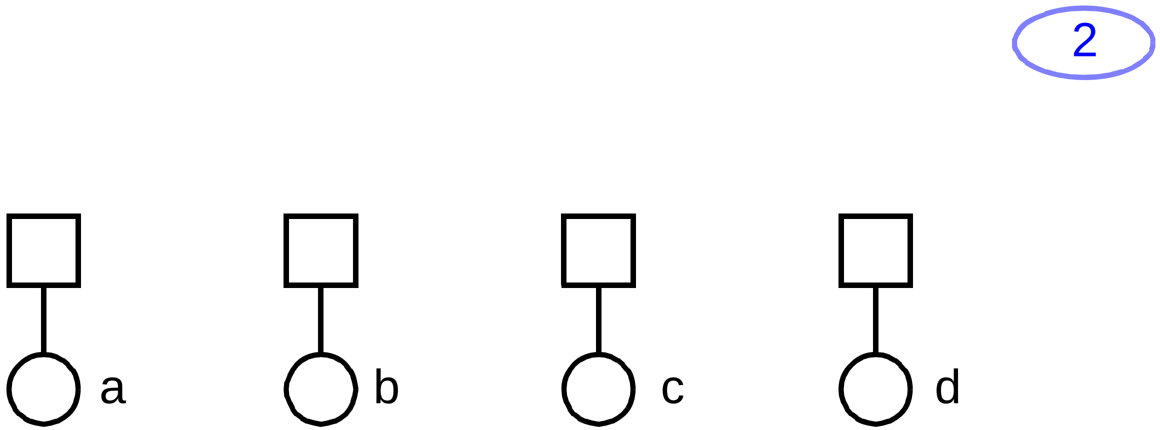


The two places count1, count2 count modulo 5

the number of the transitions' occurrences.

Obviously, they can differ by 1 (modulo 5) in any reachable marking only.

Hamming Code



DATA BITS

a b c d

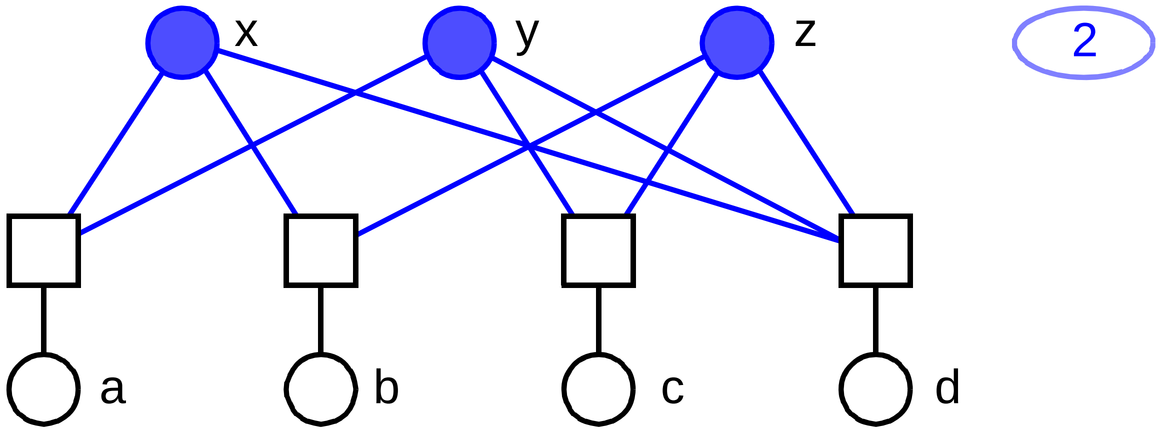
HAMMING CODE (7,4)

CONTROL BITS

x y a z b c d

DATA BITS

a b c d



x = a + b + d, y = a + c + d, z = b + c + d

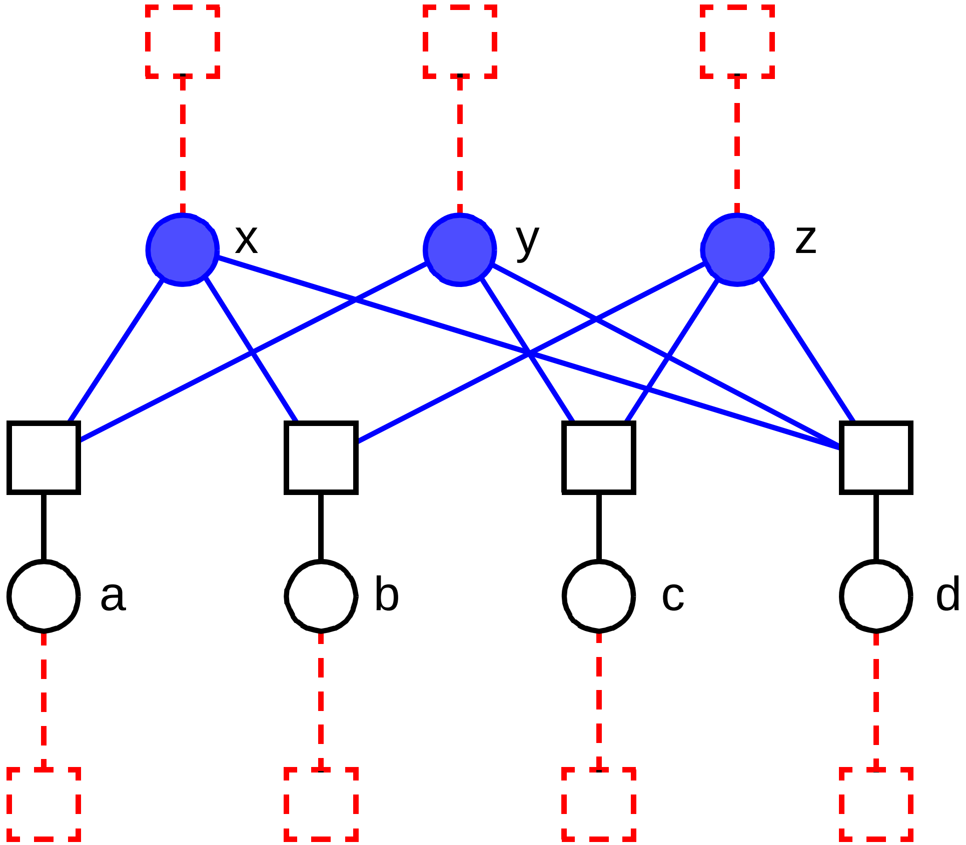
FAULT MODEL

CONTROL BITS

x y a z b c d

DATA BITS

a b c d



2

x = a + b + d, y = a + c + d, z = b + c + d

HAMMING CODE (7,4)

	t1	t2	t3	t4	m0
x	1	1	0	1	0
y	1	0	1	1	0
a	1	0	0	0	0
z	0	1	1	1	0
b	0	1	0	0	0
c	0	0	1	0	0
d	0	0	0	1	0

t1

	t1	t2	t3	t4	m0	m1
x	1	1	0	1	0	
y	1	0	1	1	0	
a	1	0	0	0	0	
z	0	1	1	1	0	
b	0	1	0	0	0	
c	0	0	1	0	0	
d	0	0	0	1	0	

t1

	t1	t2	t3	t4	m0	m1
x	1	1	0	1	0	1
y	1	0	1	1	0	1
a	1	0	0	0	0	1
z	0	1	1	1	0	0
b	0	1	0	0	0	0
c	0	0	1	0	0	0
d	0	0	0	1	0	0

Hamming Code (7,4)

t1 t2

	t1	t2	t3	t4	m0	m1	m2
x	1	1	0	1	0	1	
y	1	0	1	1	0	1	
a	1	0	0	0	0	1	
z	0	1	1	1	0	0	
b	0	1	0	0	0	0	
c	0	0	1	0	0	0	
d	0	0	0	1	0	0	

HAMMING CODE (7,4)

					t1	t2	
	t1	t2	t3	t4	m0	m1	m2
x	1	1	0	1	0	1	0
y	1	0	1	1	0	1	1
a	1	0	0	0	0	1	1
z	0	1	1	1	0	0	1
b	0	1	0	0	0	0	1
c	0	0	1	0	0	0	0
d	0	0	0	1	0	0	0

t1 t2 t3

	t1	t2	t3	t4	m0	m1	m2	m3
x	1	1	0	1	0	1	0	0
y	1	0	1	1	0	1	1	0
a	1	0	0	0	0	1	1	1
z	0	1	1	1	0	0	1	0
b	0	1	0	0	0	0	1	1
c	0	0	1	0	0	0	0	1
d	0	0	0	1	0	0	0	0

						t1	t2	t3	t4
	t1	t2	t3	t4	m0	m1	m2	m3	m4
x	1	1	0	1	0	1	0	0	1
y	1	0	1	1	0	1	1	0	1
a	1	0	0	0	0	1	1	1	1
z	0	1	1	1	0	0	1	0	1
b	0	1	0	0	0	0	1	1	1
c	0	0	1	0	0	0	0	1	1
d	0	0	0	1	0	0	0	0	1

						t1	t2	t3	t4	t1
	t1	t2	t3	t4	m0	m1	m2	m3	m4	m5
x	1	1	0	1	0	1	0	0	1	0
y	1	0	1	1	0	1	1	0	1	0
a	1	0	0	0	0	1	1	1	1	0
z	0	1	1	1	0	0	1	0	1	1
b	0	1	0	0	0	0	1	1	1	1
c	0	0	1	0	0	0	0	1	1	1
d	0	0	0	1	0	0	0	0	1	1

t1 t2 t3 t4 t1

	t1	t2	t3	t4	m0	m1	m2	m3	m4	m5
x	1	1	0	1	0	1	0	0	1	0
y	1	0	1	1	0	1	1	0	1	0
a	1	0	0	0	0	1	1	1	1	0
z	0	1	1	1	0	0	1	0	1	1
b	0	1	0	0	0	0	1	1	1	1
c	0	0	1	0	0	0	0	1	1	1
d	0	0	0	1	0	0	0	0	1	1

t1 t2 t3 t4 t1

	t1	t2	t3	t4	m0	m1	m2	m3	m4	m5	m15
x	1	1	0	1	0	1	0	0	1	0	1
y	1	0	1	1	0	1	1	0	1	0	1
a	1	0	0	0	0	1	1	1	1	0	1
z	0	1	1	1	0	0	1	0	1	1	1
b	0	1	0	0	0	0	1	1	1	1	1
c	0	0	1	0	0	0	0	1	1	1	1
d	0	0	0	1	0	0	0	0	1	1	1

$$16 < 2^7$$

t1 t2 t3 t4 t1

	t1	t2	t3	t4	m0	m1	m2	m3	m4	m5	m15
x	**1**	**1**	**0**	**1**	0	1	0	0	1	0	1
y	1	0	1	1	0	1	1	0	1	0	1
a	1	0	0	0	0	1	1	1	1	0	1
z	0	1	1	1	0	0	1	0	1	1	1
b	0	1	0	0	0	0	1	1	1	1	1
c	0	0	1	0	0	0	0	1	1	1	1
d	0	0	0	1	0	0	0	0	1	1	1

x = a + b + d

y = a + c + d

z = b + c + d

$16 < 2^7$

t1 t2 t3 t4 t1

	t1	t2	t3	t4	m0	m1	m2	m3	m4	m5	m15
x	1	1	0	1	0	1	0	0	1	0	1
y	**1**	**0**	**1**	**1**	0	1	1	0	1	0	1
a	1	0	0	0	0	1	1	1	1	0	1
z	0	1	1	1	0	0	1	0	1	1	1
b	0	1	0	0	0	0	1	1	1	1	1
c	0	0	1	0	0	0	0	1	1	1	1
d	0	0	0	1	0	0	0	0	1	1	1

$x = a + b + d$

$y = a + c + d$

$z = b + c + d$

$$16 < 2^7$$

t1 t2 t3 t4 t1

	t1	t2	t3	t4	m0	m1	m2	m3	m4	m5	m15
x	1	1	0	1	0	1	0	0	1	0	1
y	1	0	1	1	0	1	1	0	1	0	1
a	1	0	0	0	0	1	1	1	1	0	1
z	**0**	**1**	**1**	**1**	0	0	1	0	1	1	1
b	0	1	0	0	0	0	1	1	1	1	1
c	0	0	1	0	0	0	0	1	1	1	1
d	0	0	0	1	0	0	0	0	1	1	1

$x = a + b + d$

$y = a + c + d$

$z = b + c + d$

$$16 < 2^7$$

ERROR DETECTION

m0	m1	m2	m3	m4	m5	m15
0	1	0	0	1	0	1
0	1	1	0	1	0	1
0	1	1	1	1	0	1
0	0	1	0	1	1	1
0	0	1	1	1	1	1
0	0	0	1	1	1	1
0	0	0	0	1	1	1

$$H\,m = 0$$

0	0	0	1	1	1	1
0	1	1	0	0	1	1
1	0	1	0	1	0	1

ERROR DETECTION

H m = 0

m0	m1	m2	m3	m4	m5	m15
0	1	0	0	1	0	1
0	1	1	0	1	0	1
0	1	1	1	1	0	1
0	0	1	0	1	1	1
0	0	1	1	1	1	1
0	0	0	1	1	1	1
0	0	0	0	1	1	1

0	0	0	1	1	1	1	0
0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0

ERROR DETECTION

	m0	m1	m2	m3	m4	m5	m15
	0	1	0	0	1	0	1
	0	1	1	0	1	0	1
	0	1	1	1	1	0	1
H m = 0	0	0	1	0	1	1	1
	0	0	1	1	1	1	1
	0	0	0	1	1	1	1
	0	0	0	0	1	1	1

0	0	0	1	1	1	1
0	1	1	0	0	1	1
1	0	1	0	1	0	1

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

ERROR CORRECTION (single bit)

	m0	m1	m2	m3	m4	m5	m15
(1)	0	1	0	0	1	0	1
(2)	0	1	1	0	1	0	1
(3)	0	1	1	1	1	0	1
(4)	0	0	1	0	1	1	1
(5)	0	0	1	1	1	1	1
(6)	0	0	0	1	1	1	1
(7)	0	0	0	0	1	1	1

$H\,m = 0$

0	0	0	1	1	1	1
0	1	1	0	0	1	1
1	0	1	0	1	0	1

ERROR CORRECTION (single bit)

	m0	m1	m2	m3	m4	m5	m15
(1)	0	1	0	0	1	1	1
(2)	1	1	1	0	1	0	1
(3)	0	1	0	1	1	0	1
(4)	0	0	1	0	0	1	1
(5)	0	1	1	1	1	1	1
(6)	0	0	0	1	1	1	0
(7)	0	0	0	1	1	1	1

H m = 0

0	0	0	1	1	1	1
0	1	1	0	0	1	1
1	0	1	0	1	0	1

ERROR CORRECTION (single bit)

	m0	m1	m2	m3	m4	m5	m15
(1)	0	1	0	0	1	1	1
(2)	1	1	1	0	1	0	1
(3)	0	1	0	1	1	0	1
(4)	0	0	1	0	0	1	1
(5)	0	1	1	1	1	1	1
(6)	0	0	0	1	1	1	0
(7)	0	0	0	1	1	1	1

H m = 0

0	0	0	1	1	1	1	0
0	1	1	0	0	1	1	1
1	0	1	0	1	0	1	0

ERROR CORRECTION (single bit)

$$H\,m = 0$$

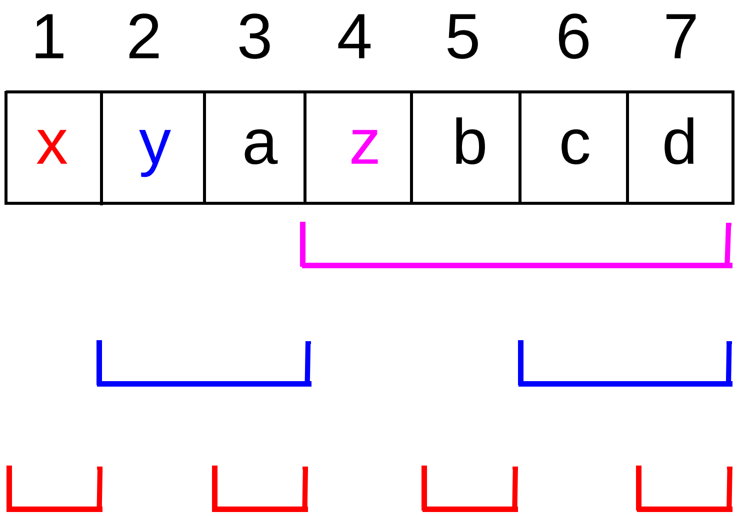
	m0	m1	m2	m3	m4	m5	m15
(1)	0	1	0	0	1	1	1
(2)	1	1	1	0	1	0	1
(3)	0	1	0	1	1	0	1
(4)	0	0	1	0	0	1	1
(5)	0	1	1	1	1	1	1
(6)	0	0	0	1	1	1	0
(7)	0	0	0	1	1	1	1

0	0	0	1	1	1	1	0	1
0	1	1	0	0	1	1	1	0
1	0	1	0	1	0	1	0	1

ERROR CORRECTION (single bit)

		m0	m1	m2	m3	m4	m5	m15
(1)		0	1	0	0	1	1	1
(2)		1	1	1	0	1	0	1
(3)		0	1	0	1	1	0	1
(4)	H m = 0	0	0	1	0	0	1	1
(5)		0	1	1	1	1	1	1
(6)		0	0	0	1	1	1	0
(7)		0	0	0	1	1	1	1

0	0	0	1	1	1	1	0	1	0	1	1	0	1
0	1	1	0	0	1	1	1	0	1	1	0	0	1
1	0	1	0	1	0	1	0	1	1	1	0	1	0



$z = b + c + d$

$y = a + c + d$

$x = a + b + d$

$H\,m = 0$ -> correct code word

$H\,m = e \neq 0$ -> defect code word

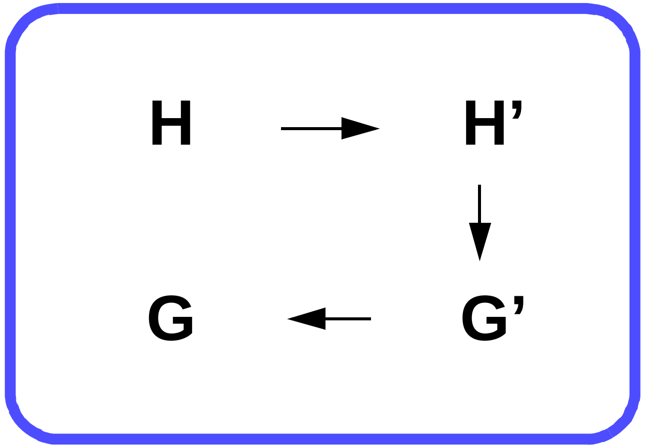
$[e]_2$ -> defect position

PARITY MATRIX H ⟷ **GENERATOR MATRIX G**

$$\mathbf{H}\, m = 0$$

$$\mathbf{G}^T\, \mathbf{C} = \overline{\mathbf{C}}$$

↑ incidence matrix



Hamming Code, General Principle

$$H = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \\ \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \end{array}$$

$$H' = \begin{array}{c} \begin{matrix} 3 & 5 & 6 & 7 & 4 & 2 & 1 \end{matrix} \\ \left(\begin{array}{cccc|ccc} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right) \end{array}$$

A binary linear code corrects single errors, iff the columns of its parity matrix H are

- non-null
- two-by-two different

[- linearly independent]

HAMMING CODE, GENERAL PRINCIPLE

	1	2	3	4	5	6	7
H =	0	0	0	1	1	1	1
	0	1	1	0	0	1	1
	1	0	1	0	1	0	1

PARITY MATRIX H

	3	5	6	7	4	2	1
H' =	0	1	1	1	1	0	0
	1	0	1	1	0	1	0
	1	1	0	1	0	0	1

$H' = \left[\, A \mid I_3 \,\right]$

HAMMING CODE, GENERAL PRINCIPLE

$$H = \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{array}$$

$$H' = \begin{array}{cccc|ccc} 3 & 5 & 6 & 7 & 4 & 2 & 1 \\ \hline 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}$$

$$H' = \left[\; A \;\middle|\; I_3 \;\right]$$

$$G' = \left[\; I_4 \;\middle|\; -A^T \;\right]$$

H =

1	2	3	4	5	6	7
0	0	0	1	1	1	1
0	1	1	0	0	1	1
1	0	1	0	1	0	1

H' =

3	5	6	7	4	2	1
0	1	1	1	1	0	0
1	0	1	1	0	1	0
1	1	0	1	0	0	1

$$H' = [\,A \mid I_3\,]$$

$$G' = [\,I_4 \mid -A^T\,]$$

G' =

3	5	6	7	4	2	1
1	0	0	0	0	1	1
0	1	0	0	1	0	1
0	0	1	0	1	1	0
0	0	0	1	1	1	1

$$H = \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \left[\begin{array}{ccccccc} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{array}\right] \end{array}$$

$$H' = \begin{array}{c} \begin{array}{ccccccc} 3 & 5 & 6 & 7 & 4 & 2 & 1 \end{array} \\ \left[\begin{array}{cccc|ccc} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right] \end{array}$$

$$H' = \left[\begin{array}{c|c} A & I_3 \end{array}\right]$$

$$G' = \left[\begin{array}{c|c} I_4 & -A^T \end{array}\right]$$

$$G = \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \left[\begin{array}{cccc|ccc} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right] \end{array}$$

$$G' = \begin{array}{c} \begin{array}{ccccccc} 3 & 5 & 6 & 7 & 4 & 2 & 1 \end{array} \\ \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right] \end{array}$$

HAMMING CODE, EXAMPLE

$$G = \begin{array}{cccc|ccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}$$

$$G^T = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Hamming Code, Example

G =

1	2	3	4	5	6	7
1	1	1	0	0	0	0
1	0	0	1	1	0	0
0	1	0	1	0	1	0
1	1	0	1	0	0	1

C

	t1	t2	t3	t4
a	1	0	0	0
b	0	1	0	0
c	0	0	1	0
d	0	0	0	1

G^T =

1	1	0	1
1	0	1	1
1	0	0	0
0	1	1	1
0	1	0	0
0	0	1	0
0	0	0	1

Hamming Code, Example

G =

	1	2	3	4	5	6	7
	1	1	1	0	0	0	0
	1	0	0	1	1	0	0
	0	1	0	1	0	1	0
	1	1	0	1	0	0	1

C

	t1	t2	t3	t4
a	1	0	0	0
b	0	1	0	0
c	0	0	1	0
d	0	0	0	1

G^T =

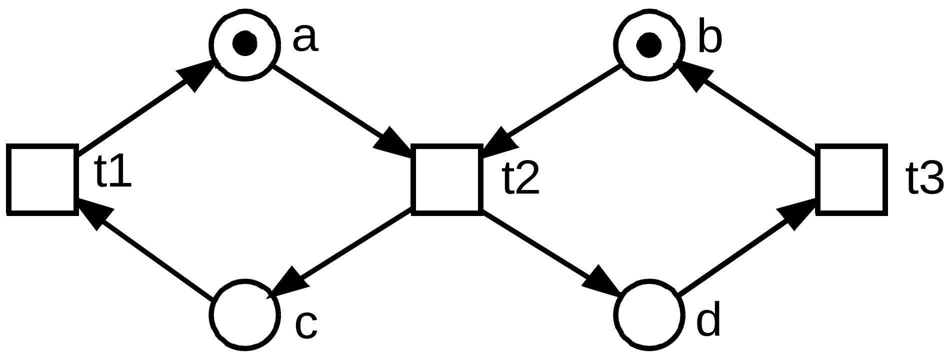
1	1	0	1
1	0	1	1
1	0	0	0
0	1	1	1
0	1	0	0
0	0	1	0
0	0	0	1

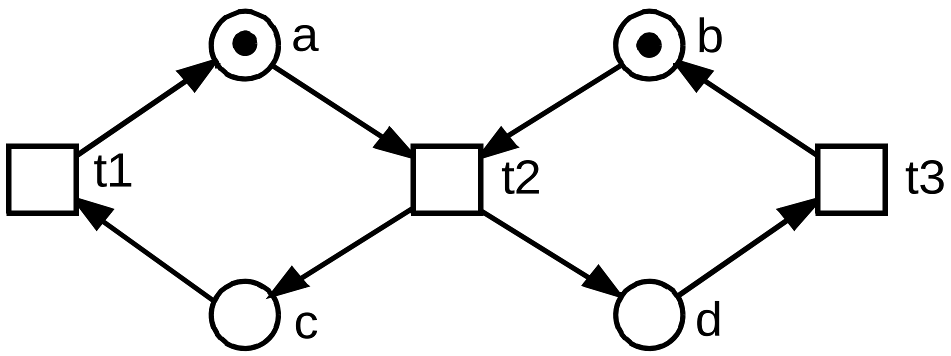
$\overline{C}$

x	1	1	0	1
y	1	0	1	1
a	1	0	0	0
z	0	1	1	1
b	0	1	0	0
c	0	0	1	0
d	0	0	0	1

Error-correcting Petri nets

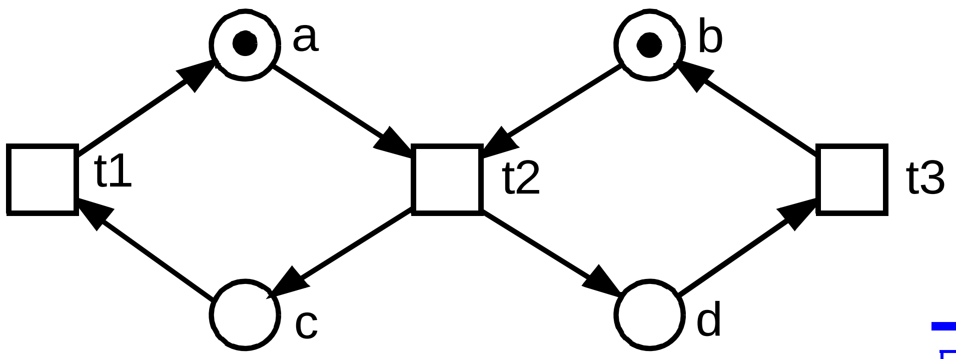
ERROR-CORRECTING PETRI NETS, EXAMPLE





	t1	t2	t3
a	1	-1	0
b	0	-1	1
c	-1	1	0
d	0	1	-1

Error-correcting Petri Nets, Example



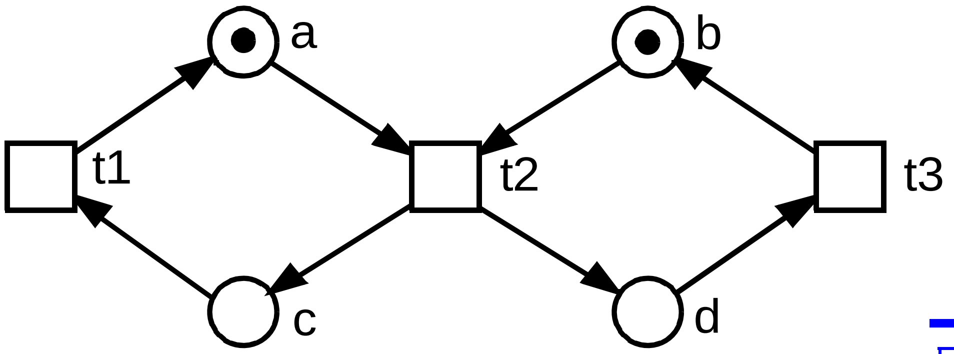
$$\overline{C} = G^T C$$

	t1	t2	t3
a	1	-1	0
b	0	-1	1
c	-1	1	0
d	0	1	-1

$G^T =$

1	1	0	1
1	0	1	1
1	0	0	0
0	1	1	1
0	1	0	0
0	0	1	0
0	0	0	1

Error-correcting Petri Nets, Example



$\overline{C} = G^T C$

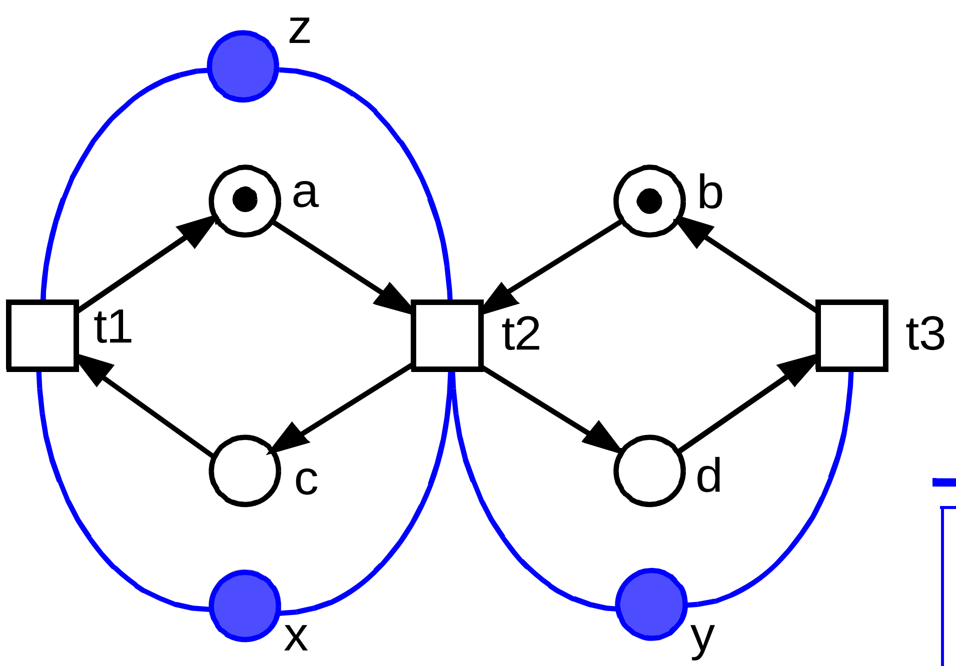
	t1	t2	t3
a	1	-1	0
b	0	-1	1
c	-1	1	0
d	0	1	-1

$G^T =$

1	1	0	1
1	0	1	1
1	0	0	0
0	1	1	1
0	1	0	0
0	0	1	0
0	0	0	1

1	1	0	x
0	1	1	y
1	1	0	a
1	1	0	z
0	1	1	b
1	1	0	c
0	1	1	d

Error-correcting Petri Nets, Example



$\overline{C} = G^T C$

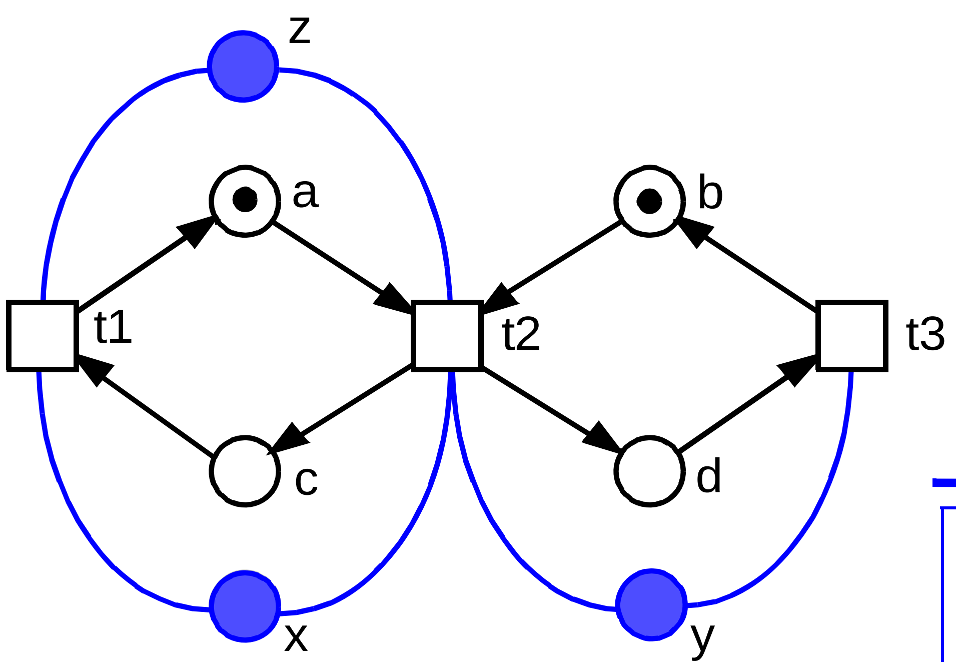
	t1	t2	t3
a	1	-1	0
b	0	-1	1
c	-1	1	0
d	0	1	-1

$G^T =$

1	1	0	1
1	0	1	1
1	0	0	0
0	1	1	1
0	1	0	0
0	0	1	0
0	0	0	1

t1	t2	t3	
1	1	0	x
0	1	1	y
1	1	0	a
1	1	0	z
0	1	1	b
1	1	0	c
0	1	1	d

ERROR-CORRECTING PETRI NETS, EXAMPLE



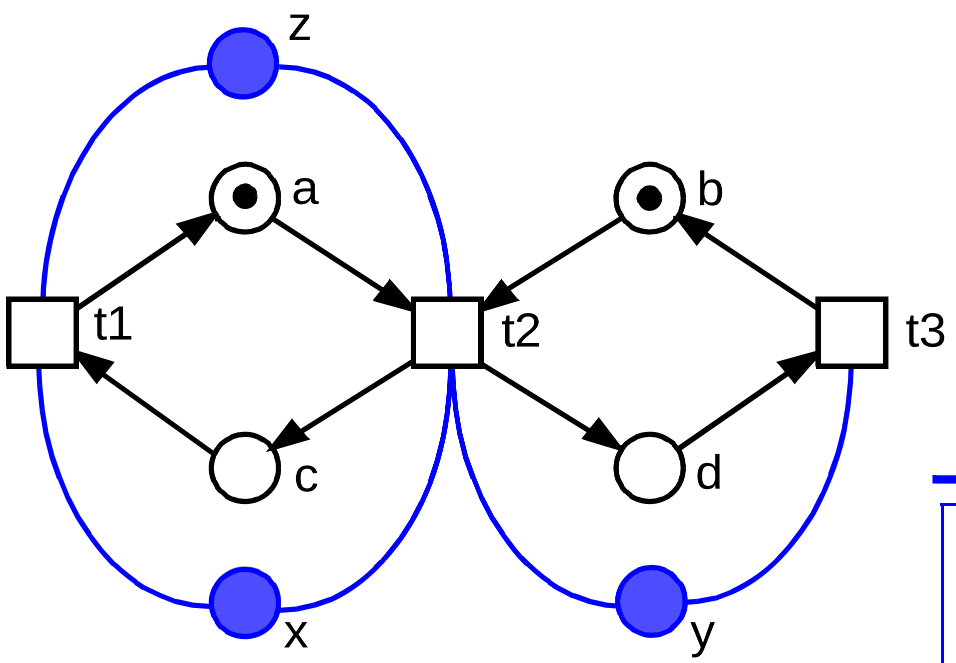
$\overline{m_0} = G^T m_0$

	t1	t2	t3	m0
a	1	-1	0	1
b	0	-1	1	1
c	-1	1	0	0
d	0	1	-1	0

$G^T =$

1	1	0	1	1	1	0	
1	0	1	1	0	1	1	
1	0	0	0	1	1	0	
0	1	1	1	1	1	0	
0	1	0	0	0	1	1	
0	0	1	0	1	1	0	
0	0	0	1	0	1	1	

INITIAL MARKING OF CONTROL PLACES



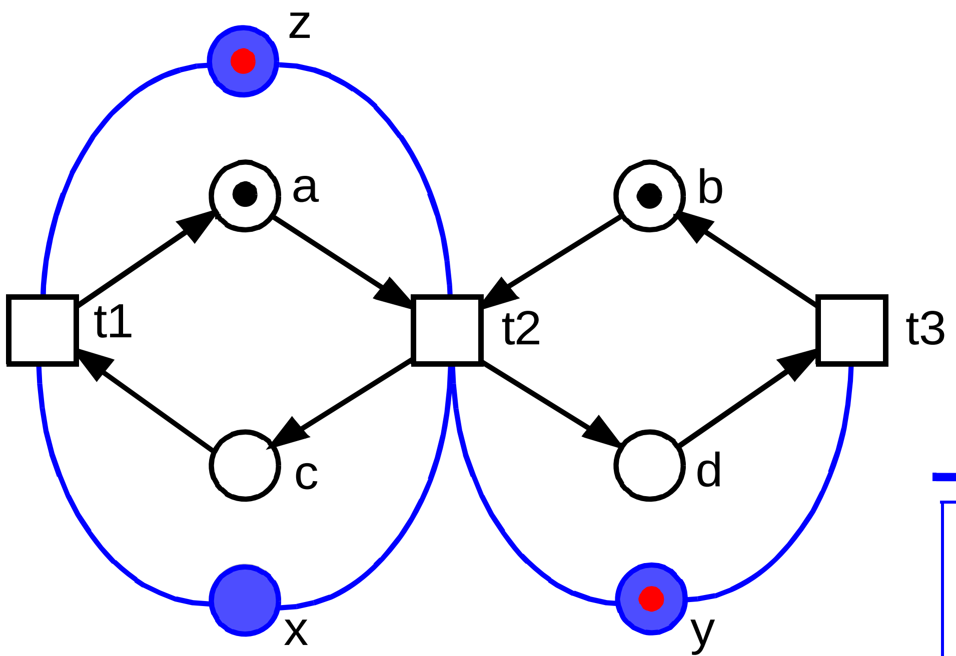
$$\overline{m_0} = G^T m_0$$

	t1	t2	t3	m0
a	1	-1	0	1
b	0	-1	1	1
c	-1	1	0	0
d	0	1	-1	0

$G^T =$

1	1	0	1	1	1	0	0
1	0	1	1	0	1	1	1
1	0	0	0	1	1	0	1
0	1	1	1	1	1	0	1
0	1	0	0	0	1	1	1
0	0	1	0	1	1	0	0
0	0	0	1	0	1	1	0

INITIAL MARKING OF CONTROL PLACES



$\overline{m_0} = G^T m_0$

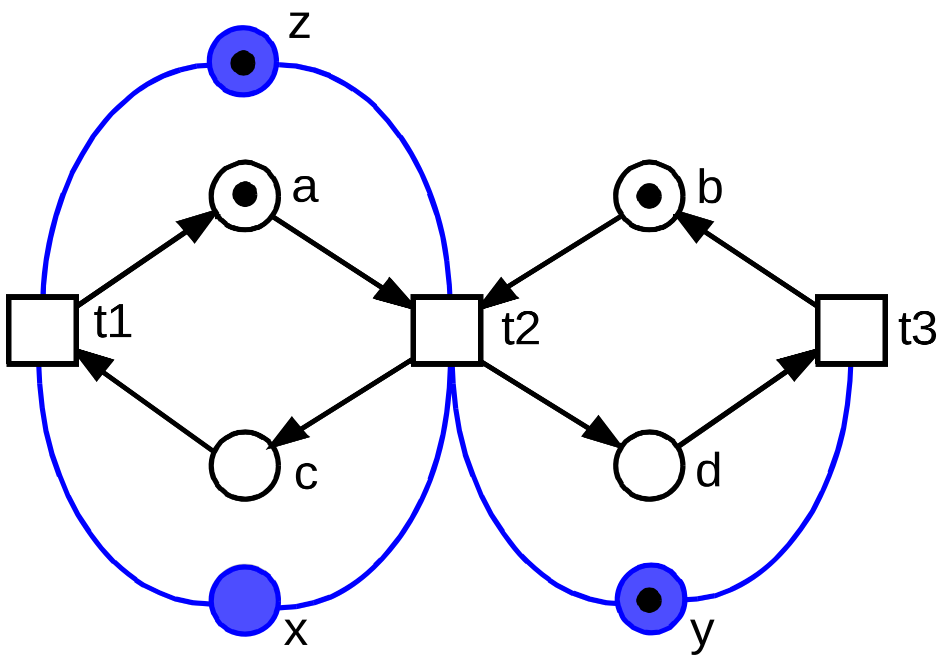
	t1	t2	t3	m0
a	1	-1	0	1
b	0	-1	1	1
c	-1	1	0	0
d	0	1	-1	0

$G^T =$

1	1	0	1	1	1	0	0
1	0	1	1	0	1	1	1
1	0	0	0	1	1	0	1
0	1	1	1	1	1	0	1
0	1	0	0	0	1	1	1
0	0	1	0	1	1	0	0
0	0	0	1	0	1	1	0

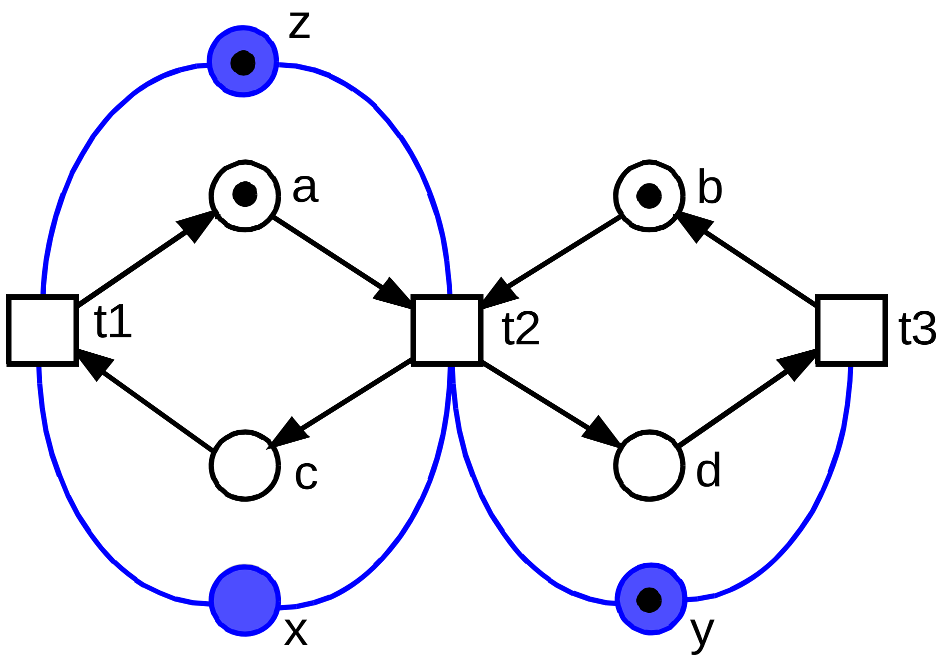
INITIAL MARKING OF CONTROL PLACES

Error-correcting Petri Nets, Example



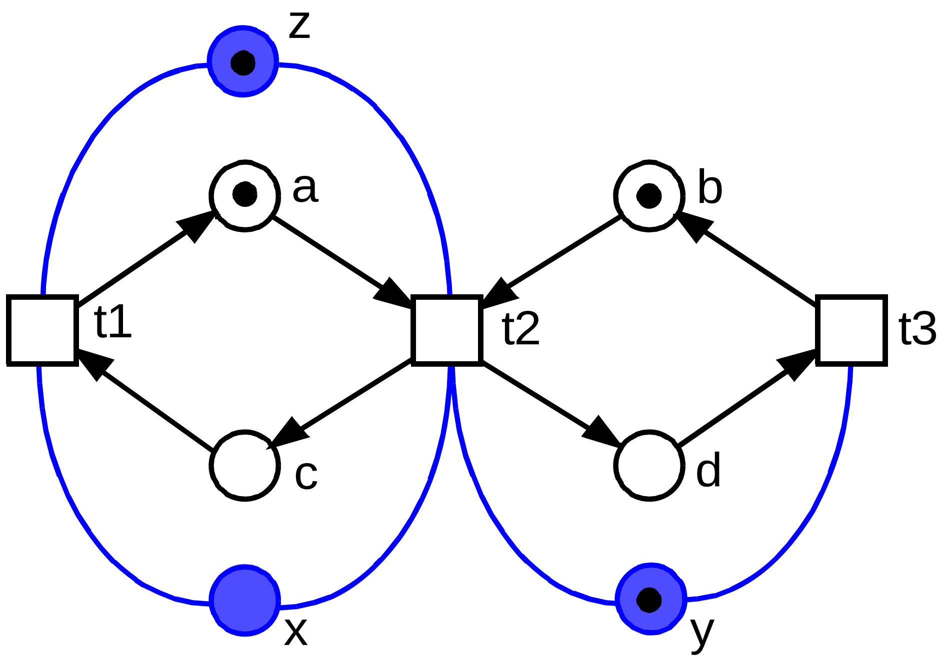
	m0	m1	m2	m3	$m3^1$	$m3^2$
x	0	1	0	1	0	1
y	1	0	0	1	1	0
a	1	0	1	0	0	0
z	1	0	1	0	0	0
b	1	0	0	1	1	1
c	0	1	0	1	1	1
d	0	1	1	0	0	0

ERROR-CORRECTING PETRI NETS, EXAMPLE



	m0	m1	m2	m3	$m3^1$	$m3^2$
x	0	1	0	1	0	1
y	1	0	0	1	1	0
a	1	0	1	0	0	0
z	1	0	1	0	0	0
b	1	0	0	1	1	1
c	0	1	0	1	1	1
d	0	1	1	0	0	0

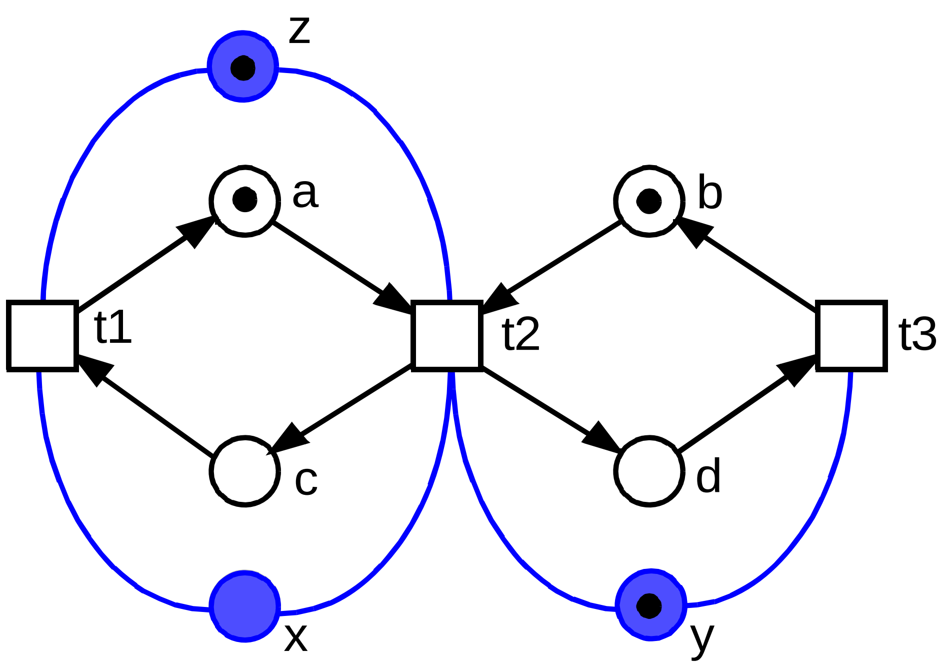
0	0	0	1	1	1	1	0
0	1	1	0	0	1	1	0
1	0	1	0	1	0	1	0



	m0	m1	m2	m3	$m3^1$	$m3^2$
x	0	1	0	1	0	1
y	1	0	0	1	1	0
a	1	0	1	0	0	0
z	1	0	1	0	0	0
b	1	0	0	1	1	1
c	0	1	0	1	1	1
d	0	1	1	0	0	0

							m0	m1
0	0	0	1	1	1	1	0	0
0	1	1	0	0	1	1	0	0
1	0	1	0	1	0	1	0	0

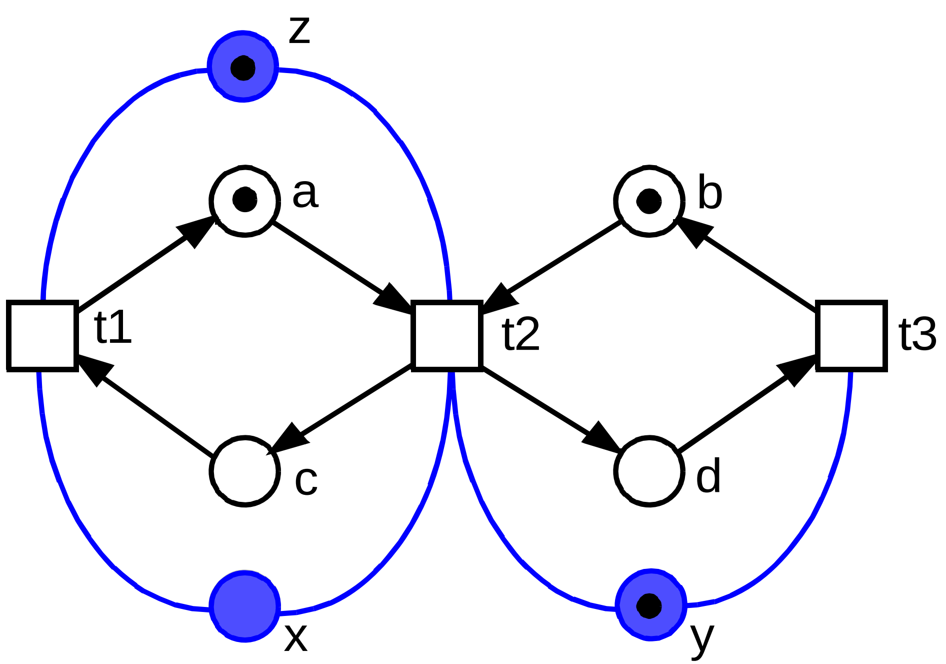
Error-correcting Petri Nets, Example



	m0	m1	m2	m3	$m3^1$	$m3^2$
x	0	1	0	1	0	1
y	1	0	0	1	1	0
a	1	0	1	0	0	0
z	1	0	1	0	0	0
b	1	0	0	1	1	1
c	0	1	0	1	1	1
d	0	1	1	0	0	0

0	0	0	1	1	1	1	0	0	0
0	1	1	0	0	1	1	0	0	0
1	0	1	0	1	0	1	0	0	0

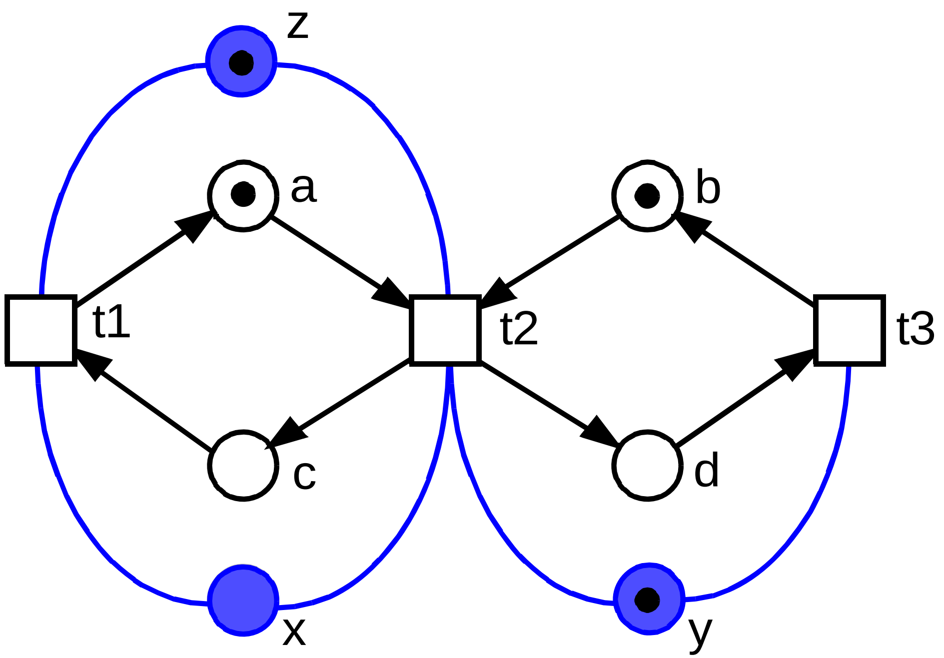
Error-correcting Petri Nets, Example



	m0	m1	m2	m3	$m3^1$	$m3^2$
x	0	1	0	1	0	1
y	1	0	0	1	1	0
a	1	0	1	0	0	0
z	1	0	1	0	0	0
b	1	0	0	1	1	1
c	0	1	0	1	1	1
d	0	1	1	0	0	0

0	0	0	1	1	1	1		0	0	0	0
0	1	1	0	0	1	1		0	0	0	0
1	0	1	0	1	0	1		0	0	0	0

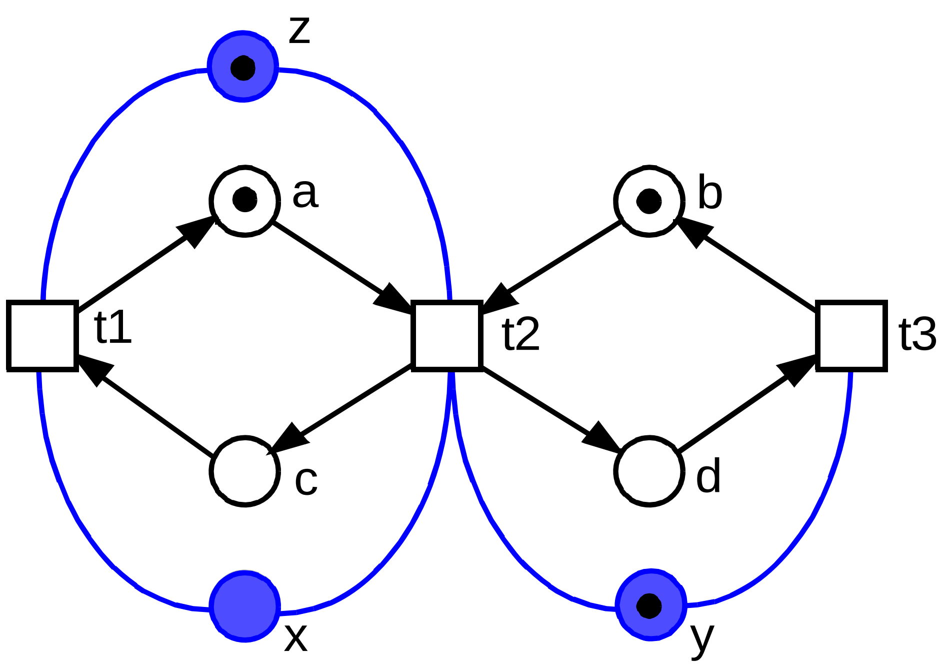
Error-correcting Petri Nets, Example



	m0	m1	m2	m3	$m3^1$	$m3^2$
x	0	1	0	1	0	1
y	1	0	0	1	1	0
a	1	0	1	0	0	0
z	1	0	1	0	0	0
b	1	0	0	1	1	1
c	0	1	0	1	1	1
d	0	1	1	0	0	0

0	0	0	1	1	1	1	0	0	0	0	0
0	1	1	0	0	1	1	0	0	0	0	0
1	0	1	0	1	0	1	0	0	0	0	1

ERROR-CORRECTING PETRI NETS, EXAMPLE



	m0	m1	m2	m3	$m3^1$	$m3^2$
x	0	1	0	1	0	1
y	1	0	0	1	1	0
a	1	0	1	0	0	0
z	1	0	1	0	0	0
b	1	0	0	1	1	1
c	0	1	0	1	1	1
d	0	1	1	0	0	0

0	0	0	1	1	1	1	0	0	0	0	0	0
0	1	1	0	0	1	1	0	0	0	0	0	1
1	0	1	0	1	0	1	0	0	0	0	1	0

ERROR-CORRECTING PETRI NETS, GENERAL PRINCIPLE

- **reachable markings are linear combinations of**
  - -> *initial marking*
  - -> *columns of incidence matrix*
- **each linear combination corresponds to a possible marking, which has not to be reachable**

ERROR-CORRECTING PETRI NETS, GENERAL PRINCIPLE

- **reachable markings are linear combinations of**
  - -> *initial marking*
  - -> *columns of incidence matrix*
- **each linear combination corresponds to a possible marking, which has not to be reachable**
- **set of reachable markings $\subseteq$ set of possible markings**

ERROR-CORRECTING PETRI NETS, GENERAL PRINCIPLE

- **reachable markings are linear combinations of**
  - -> *initial marking*
  - -> *columns of incidence matrix*
- **each linear combination corresponds to a possible marking, which has not to be reachable**
- **set of reachable markings ⊆ set of possible markings**
- **possible markings**
  - -> *code words of error-correcting code*

WE WERE EVEN ABLE TO CORRECT THE POSSIBLE MARKINGS, WHICH WILL NEVER BE REACHED.

ERROR-CORRECTING PETRI NETS, GENERAL PROCEDERE

- **determine**
  - -> *number of places h*
  - -> *number of required control places k*
  - -> *parity matrix H, generator matrix G*

$$2^k \geq h + k + 1$$

- **determine incidence matrix** *C* **of the Petri net**

ERROR-CORRECTING PETRI NETS, GENERAL PROCEDERE

- **determine**
  - -> *number of places h*
  - -> *number of required control places k*
  - -> *parity matrix H, generator matrix G*

$$2^k \geq h + k + 1$$

- **determine incidence matrix** *C* **of the Petri net**
- **compute incidence matrix of the error-correcting Petri net**
  - -> *extend net structure by control places*

$$\overline{C} = G^T C$$

- **compute initial marking of the error- correcting Petri net**
  - -> *initial marking of control places*

$$\overline{m_0} = G^T m_0$$

Error-correcting Petri Nets, General Procedere

- **determine**
  - -> *number of places h*
  - -> *number of required control places k*
  - -> *parity matrix H, generator matrix G*

$$2^k \geq h + k + 1$$

- **determine incidence matrix** *C* **of the Petri net**
- **compute incidence matrix of the error-correcting Petri net**
  - -> *extend net structure by control places*

$$\overline{C} = G^T C$$

- **compute initial marking of the error- correcting Petri net**
  - -> *initial marking of control places*

$$\overline{m}_0 = G^T m_0$$

- **check observed markings** $\overline{m}$

-> $H\,\overline{m} = 0$	-> *correct state*
$H\,\overline{m} = e \neq 0$	-> *defect state,* $[e]_2$ *- error position*

$$2^k \geq h + k + 1$$

h - number of places

k - number of control places

k control places	h info places
3	4
4	11
5	26
6	57
7	120

works for arbitrary prime p

-> Z_p

ERROR-CORRECTING PETRI NETS, SUMMARY

- **error-correcting Petri nets = (modulo) Petri nets + Hamming code**

ERROR-CORRECTING PETRI NETS, SUMMARY

- **error-correcting Petri nets = (modulo) Petri nets + Hamming code**

- **words of Hamming code over Z_p**

  -> reachable Petri net markings $\subseteq$ possible Petri net markings

  -> linear combinations of

  - initial marking

  - columns of the incidence matrix

ERROR-CORRECTING PETRI NETS, SUMMARY

- **error-correcting Petri nets = (modulo) Petri nets + Hamming code**
- **words of Hamming code over Z_p**
  - -> *reachable Petri net markings* $\subseteq$ *possible Petri net markings*
  - -> *linear combinations of*
    - *- initial marking*
    - *- columns of the incidence matrix*
- **parity bits**
  - -> *redundant control places*
- **general procedure**
  - -> *generator matrix & parity matrix*
  - -> *works for arbitrary primes p*

ERROR-CORRECTING PETRI NETS, SUMMARY

- **error-correcting Petri nets = (modulo) Petri nets + Hamming code**
- **words of Hamming code over Z_p**
  - -> *reachable Petri net markings* $\subseteq$ *possible Petri net markings*
  - -> *linear combinations of*
    - *- initial marking*
    - *- columns of the incidence matrix*
- **parity bits**
  - -> *redundant control places*
- **general procedure**
  - -> *generator matrix & parity matrix*
  - -> *works for arbitrary primes p*
- **THE STATE SPACE IS NEVER CONSTRUCTED !**

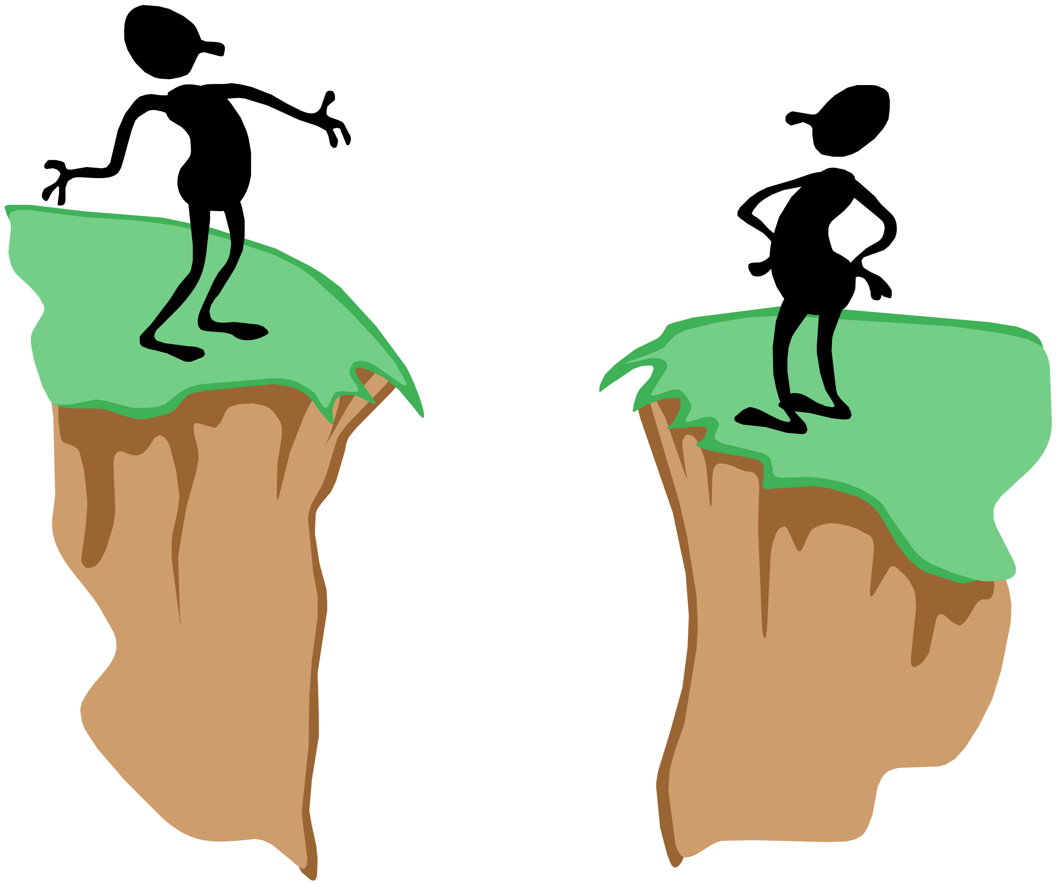
Credits

REFERENCES

- Anastasia Pagnoni
  Detecting and correcting operation errors in distributed systems;
  Bulletin of the European Association of Theoretical CS 58, 1996.

- Anastasia Pagnoni, Andrea Visconti
  Detection and analysis of unexpected state components in biological systems;
  LNCS 2602, Springer 2003

- Anastasia Pagnoni
  Error-Correcting Petri nets;
  Journal NACO, to appear.



Thanks !

http://www-dssz.informatik.tu-cottbus.de