Applications of

Polynomial Invariants

Enric Rodríguez-Carbonell
Universitat Politècnica de Catalunya (UPC)
Barcelona

Joint work with **Robert Clarisó, Jordi Cortadella** (UPC)
Ashish Tiwari (SRI)

Overview of the Talk

- **Petri Nets**
  1. Introduction
  2. Modelling with Petri Nets
  3. Generating Invariants
  4. Related Work
  5. Conclusions

- **Hybrid Systems**

Introduction

- **Petri nets:** mathematical model for studying systems
  - concurrency
  - parallelism
  - non-determinism
- **Applications:**
  - Manufacturing and Task Planning
  - Communication Networks
  - Hardware Design

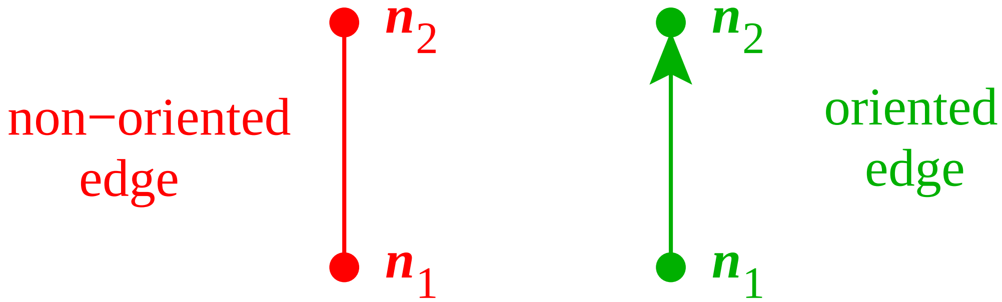
Overview of the Talk

- **Petri Nets**
  1. Introduction
  2. Modelling with Petri Nets
  3. Generating Invariants
  4. Related Work
  5. Conclusions

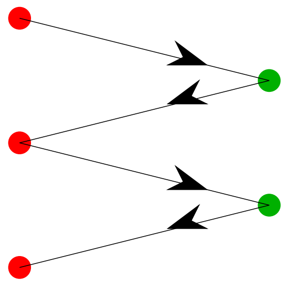
- **Hybrid Systems**

Modelling with Petri Nets
Preliminaries

- A directed graph is a graph where all edges are oriented

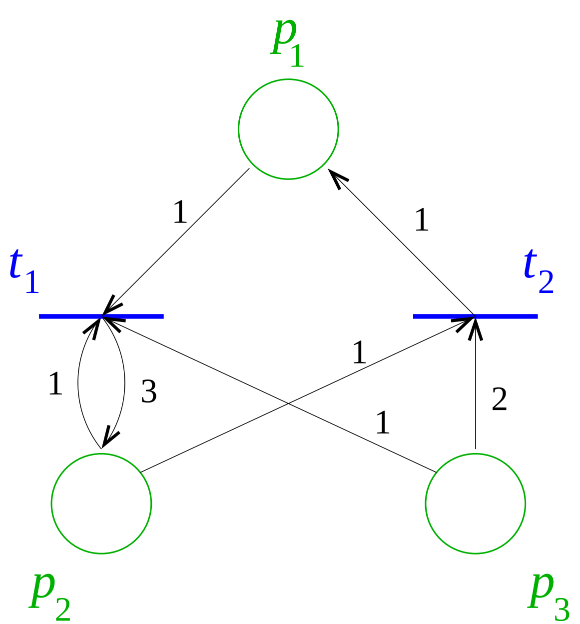


- A bipartite graph is a graph where
  1. there are two kinds of nodes
  2. edges connect only nodes that belong to different kinds



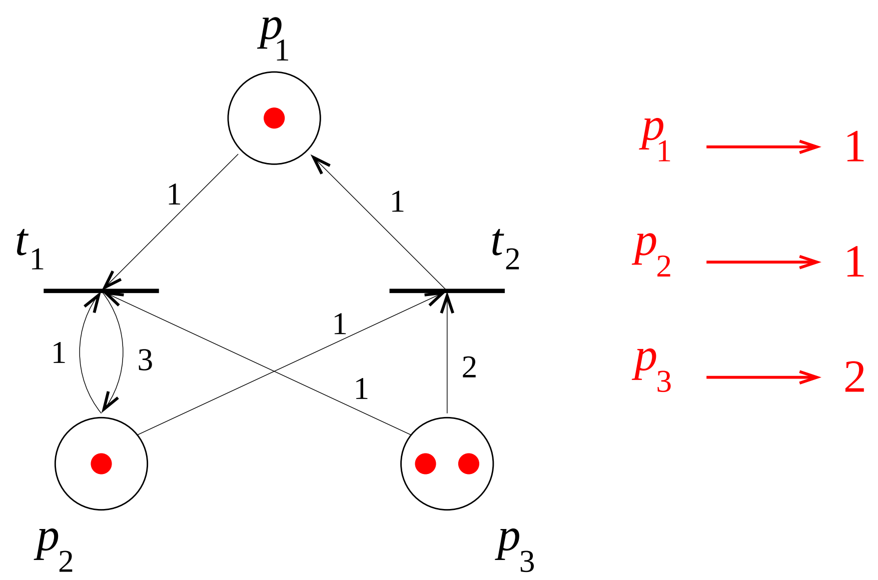
Modelling with Petri Nets
Definitions (1)

- A **Petri net** is a *bipartite directed* graph where:
  - Nodes partitioned into places (◯) and transitions (|)
  - Arcs (edges) are labelled with a natural number



Modelling with Petri Nets
Definitions (2)

- Petri nets can be *executed*: the execution shows the dynamics of the modelled system
- Tokens (•) are non-distinguishable objects located in places
- A marking maps a (natural) number of tokens to each place of the net



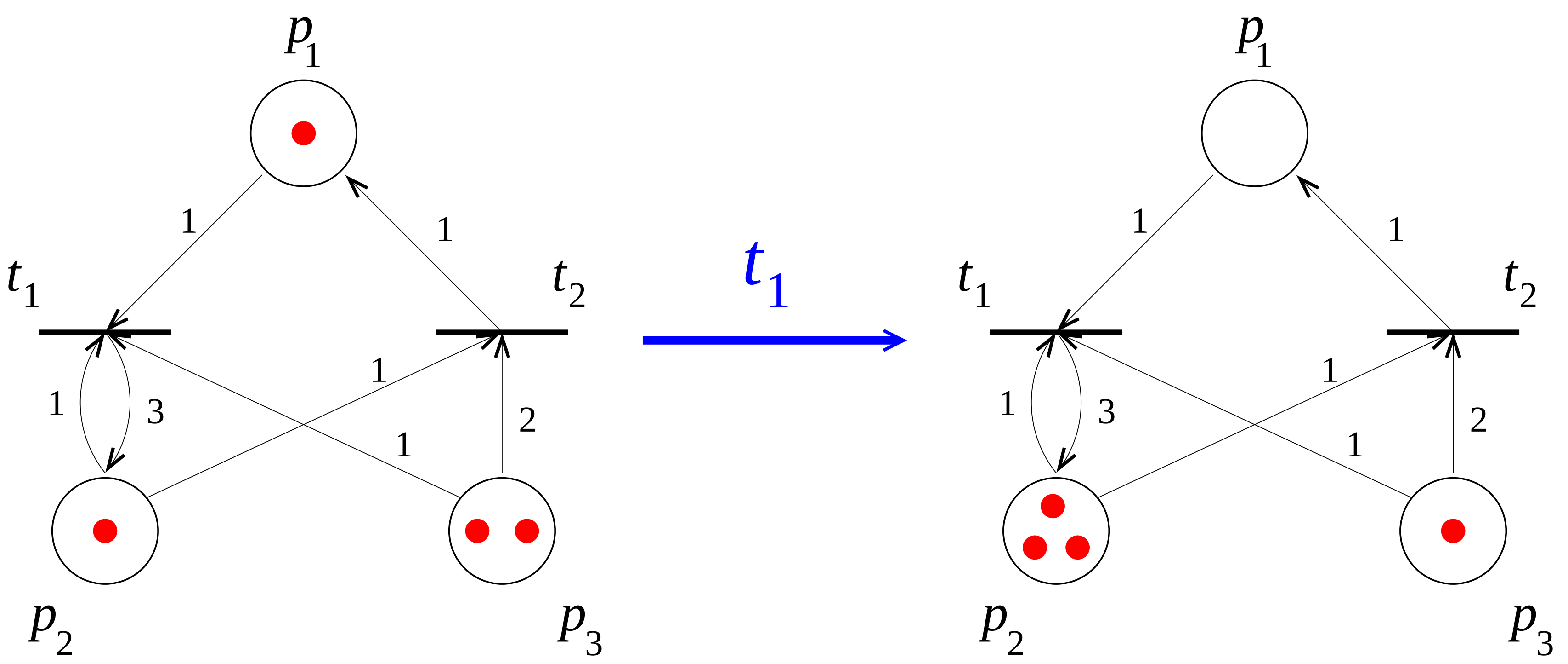
Modelling with Petri Nets

Dynamics (1)

- Dynamics of a Petri net described by
  - initial marking
  - firing of transitions

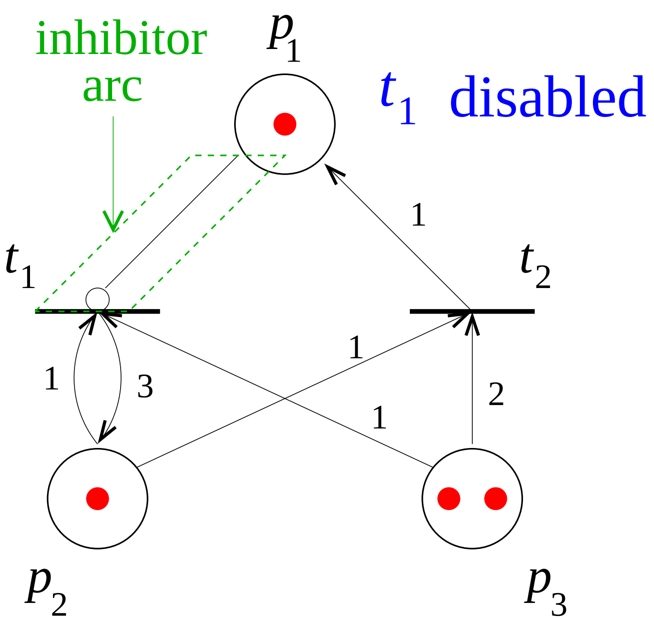
- A transition is enabled if there are $\geq$ tokens in each input place than indicated in the arcs

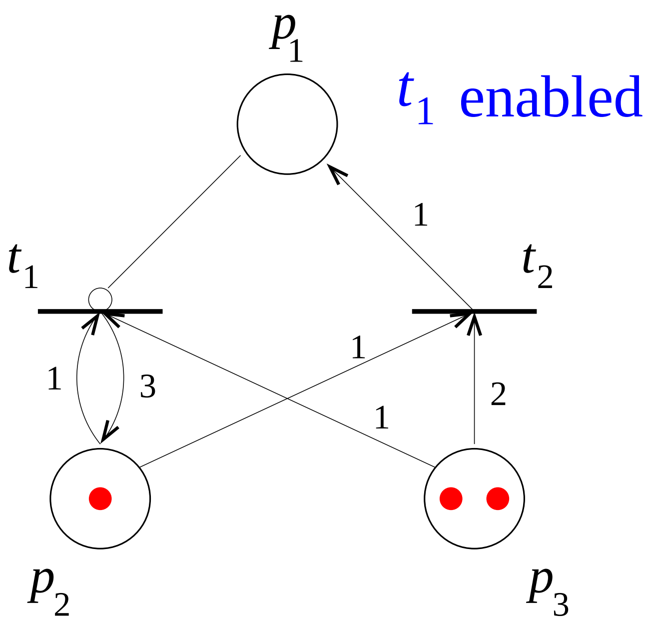
- When a transition is enabled, it can fire:
  1. the number of tokens indicated in the arcs is removed from input places
  2. tokens are generated in output places according to arcs


p_1
t_1
t_2
p_2
p_3
1
1
1
3
1
1
2
t_1
p_1
t_1
t_2
p_2
p_3
1
1
1
3
1
1
2

Modelling with Petri Nets
Dynamics (2)

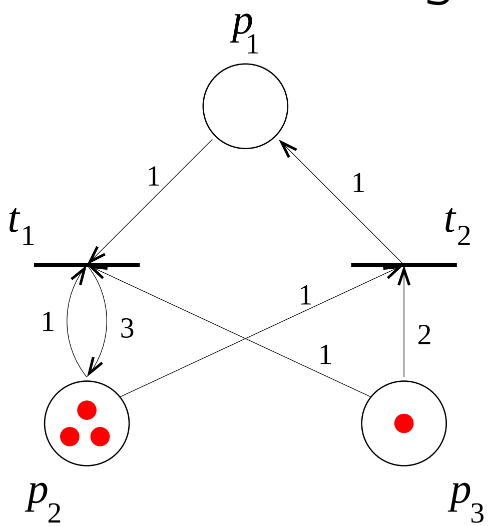
- Enabling of transitions may also depend on inhibitor arcs
- An inhibitor arc is an arc connecting place p to transition t so that there cannot be tokens in p for t to be enabled





Modelling with Petri Nets
Dynamics (3)

- The reachability set are all markings reachable by successive firings of transitions from initial marking
- Deadlocks are markings for which all transitions are disabled



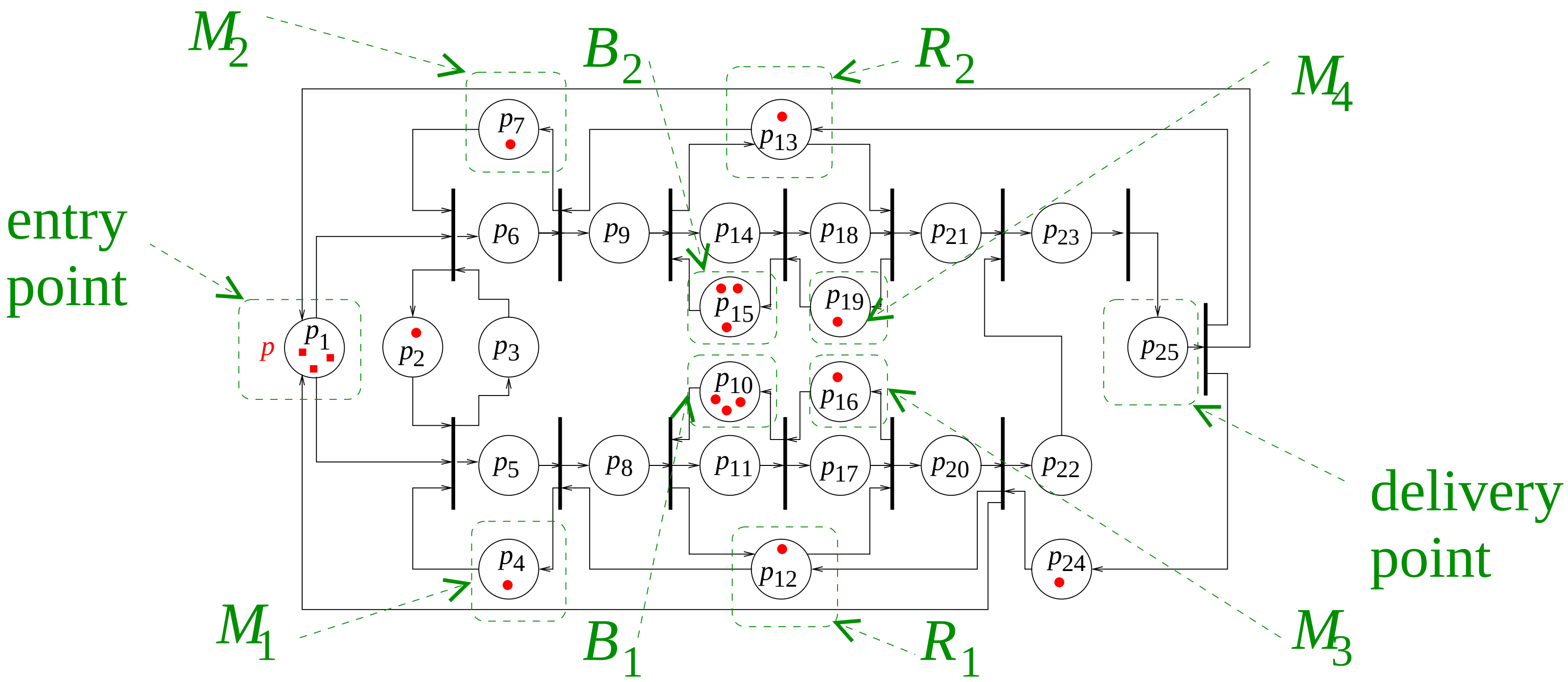
t_1 disabled

t_2 disabled

→ DEADLOCK !!

- Given a Petri net with an initial marking:
  - Invariant properties of reachable states ?
  - Any deadlocks ?

Modelling with Petri Nets
Example: Automated Manufacturing System



- Four machines M_1, M_2, M_3, M_4
- Two robots R_1, R_2
- Two buffers B_1, B_2 with capacity 3

Overview of the Talk

- **Petri Nets**
  1. Introduction
  2. Modelling with Petri Nets
  3. Generating Invariants
  4. Related Work
  5. Conclusions

- **Hybrid Systems**

Generating Invariants

Translation into Loop Programs (1)

Given a Petri net with n places p_i and m transitions t_j:

- Define variable x_i meaning number of tokens at place p_i
- Initial marking μ_1, ..., μ_n transformed into assignments

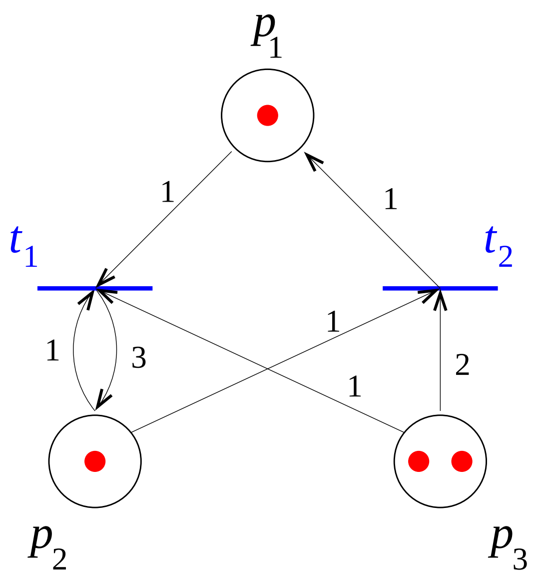
$$x_1 := \mu_1; \cdots ; x_m := \mu_m;$$

- Enabling of transition t_j with input place p_i and label c_i:

$$\cdots (x_i \neq 0) \wedge (x_i \neq 1) \wedge \cdots \wedge (x_i \neq c_i - 1) \cdots$$

- Enabling of transition t_j with inhibitor place p_i: $x_i = 0$
- Firing of transition t_j
  - with input place p_i and label c_i: $x_i := x_i - c_i;$
  - with output place p_i and label c_i: $x_i := x_i + c_i;$

Generating Invariants
Translation into Loop Programs (2)



$x_1 := 1; x_2 := 1; x_3 := 2;$

while ? **do**

t_1 : **if** $x_1 \neq 0 \wedge x_2 \neq 0 \wedge x_3 \neq 0 \rightarrow$

$$x_1 := x_1 - 1;$$
$$x_2 := x_2 + 2;$$
$$x_3 := x_3 - 1;$$

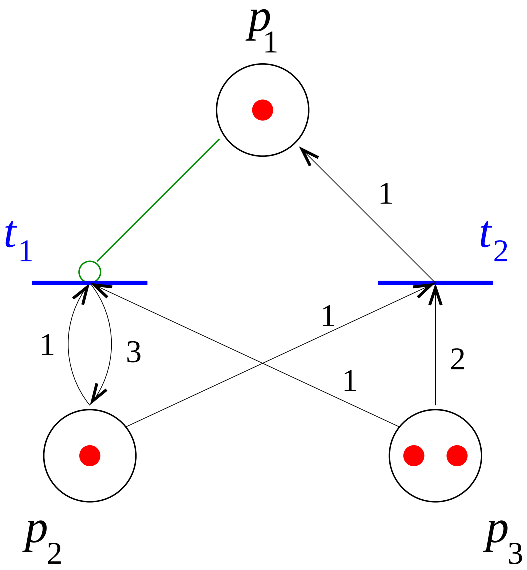
t_2 : $[]\ x_2 \neq 0 \wedge x_3 \neq 0 \wedge x_3 \neq 1 \rightarrow$

$$x_1 := x_1 + 1;$$
$$x_2 := x_2 - 1;$$
$$x_3 := x_3 - 2;$$

end if

end while

Generating Invariants
Translation into Loop Programs (3)



$x_1 := 1; x_2 := 1; x_3 := 2;$

while ? **do**

t_1 : **if** $x_1 = 0 \wedge x_2 \neq 0 \wedge x_3 \neq 0 \rightarrow$

$x_1 := x_1 - 1;$

$x_2 := x_2 + 2;$

$x_3 := x_3 - 1;$

t_2 : [] $x_2 \neq 0 \wedge x_3 \neq 0 \wedge x_3 \neq 1 \rightarrow$

$x_1 := x_1 + 1;$

$x_2 := x_2 - 1;$

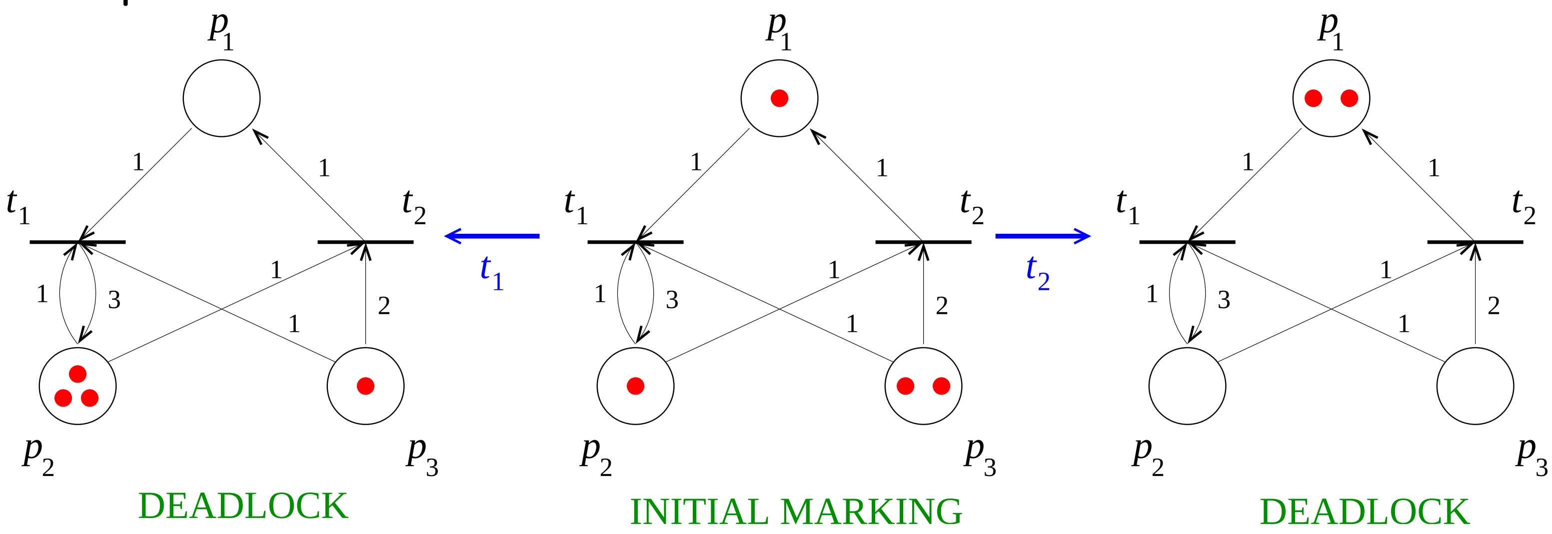
$x_3 := x_3 - 2;$

end if

end while

Generating Invariants
Applying Abstract Interpretation

- Abstract interpretation is applied to the loop program to obtain polynomial invariants of the Petri net
- Example:



- Polynomial invariants obtained:

$$Inv = \begin{cases} 5x_1 + 3x_2 + x_3 - 10 & = 0 \\ 5x_3^2 + 2x_2 - 11x_3 & = 0 \\ x_2x_3 + 2x_3^2 - 5x_3 & = 0 \\ 5x_2^2 - 17x_2 + 6x_3 & = 0 \end{cases}$$

- In this example invariants characterize reachability set

$$Inv \Leftrightarrow \begin{cases} (x_1, x_2, x_3) & = (0, 3, 1) \\ (x_1, x_2, x_3) & = (1, 1, 2) \\ (x_1, x_2, x_3) & = (2, 0, 0) \end{cases}$$

- In general overapproximation of reachability set is obtained

Generating Invariants
Deadlock Analysis (1)

- Assume no inhibitor arcs
- Generate polynomial invariants Inv of the Petri net
- Codify disabling conditions as polynomial equations Dis

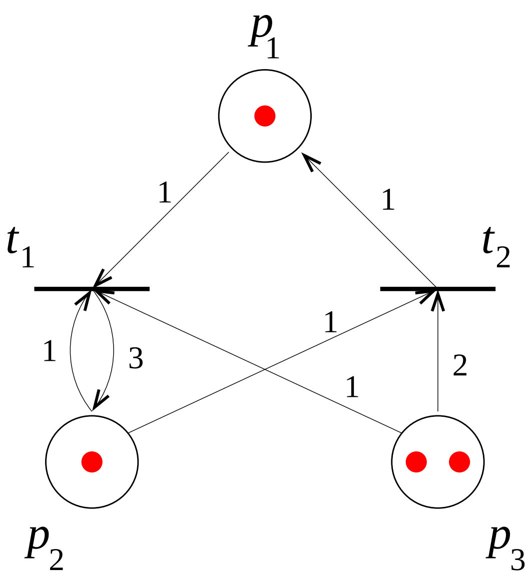
$$\neg\Big((x_i \neq 0) \wedge (x_i \neq 1) \wedge \cdots \wedge (x_i \neq c_i - 1)\Big) \equiv$$

$$\equiv (x_i = 0) \vee (x_i - 1 = 0) \vee \cdots \vee (x_i - c_i + 1 = 0)$$

$$\equiv x_i(x_i - 1) \cdots (x_i - c_i + 1) = 0$$

- If there is a deadlock, there is a solution to $Inv \cup Dis$
  $\Longrightarrow$ If the system $Inv \cup Dis$ is unfeasible, no deadlocks

Generating Invariants
Deadlock Analysis (2)

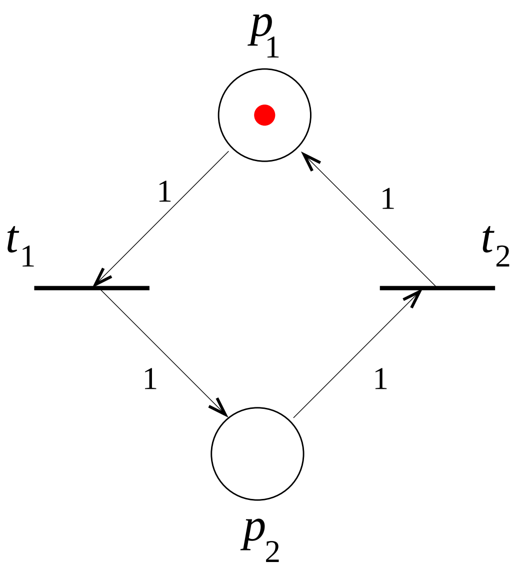


$$Inv = \begin{cases} 5x_1 + 3x_2 + x_3 - 10 & = 0 \\ 5x_3^2 + 2x_2 - 11x_3 & = 0 \\ x_2x_3 + 2x_3^2 - 5x_3 & = 0 \\ 5x_2^2 - 17x_2 + 6x_3 & = 0 \end{cases}$$

$$Dis = \begin{cases} x_1x_2x_3 & = 0 \\ x_3(x_3 - 1)x_2 & = 0 \end{cases}$$

$$Inv \cup Dis = \begin{cases} (x_1, x_2, x_3) & = (0, 3, 1) \\ & \vee \\ (x_1, x_2, x_3) & = (2, 0, 0) \end{cases}$$

Generating Invariants
Deadlock Analysis (3)



$$Inv = \begin{cases} x_1 + x_2 & = 1 \\ x_1^2 & = x_1 \end{cases}$$

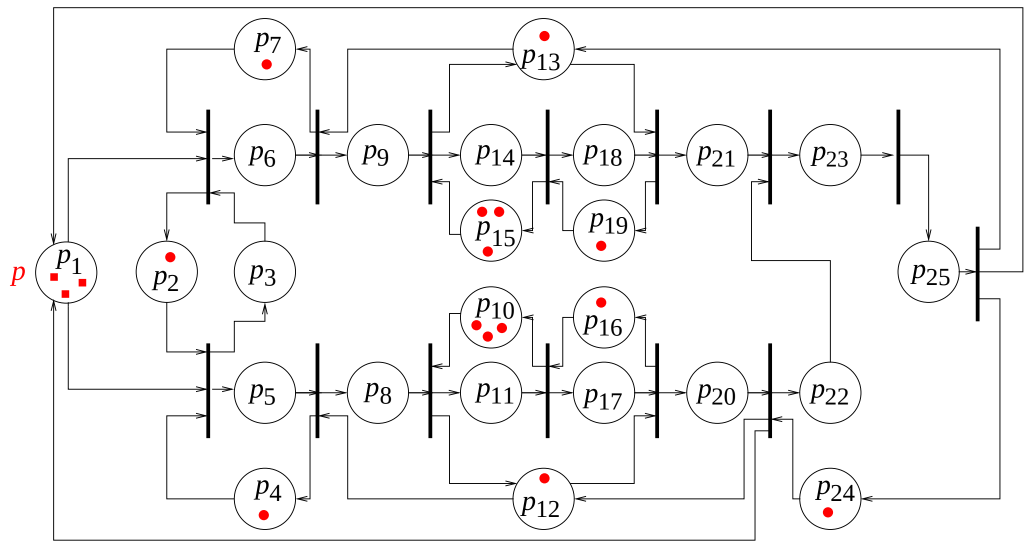
$$Dis = \begin{cases} x_1 & = 0 \\ x_2 & = 0 \end{cases}$$

$$Inv \cup Dis = \begin{cases} 1 & = 0 \end{cases}$$

UNFEASIBLE !!

Generating Invariants
Deadlock Analysis (4)

Automated Manufacturing System Revisited



- For $1 \leq p \leq 8$ Petri net is shown to be deadlock-free using polynomial invariants
- For $p \geq 9$ there are deadlocks

Overview of the Talk

- **Petri Nets**
  1. **Introduction**
  2. **Modelling with Petri Nets**
  3. **Generating Invariants**
  4. **Related Work**
  5. **Conclusions**

- **Hybrid Systems**

Related Work (1)

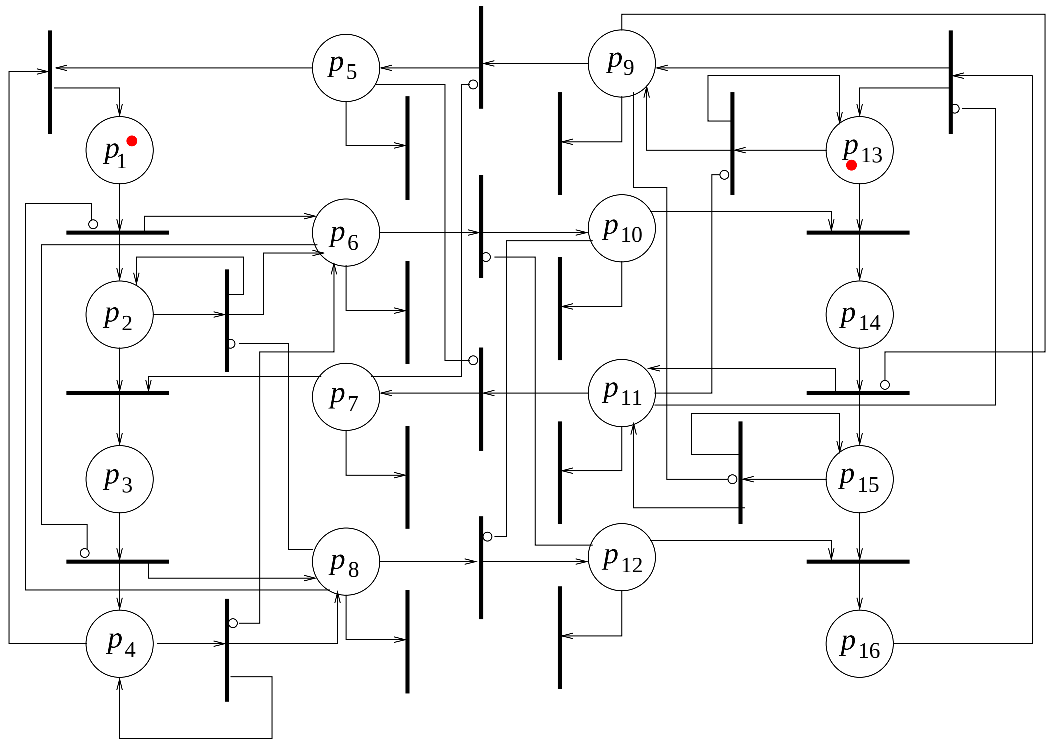
- (Sankaranarayanan et al., 2003): **linear inequality** invariants for Petri nets
  - Advantages: good to express boundedness
  - Disadvantages: bad at expressing disjunctions; but with **polynomial equalities**:

$$x_1 = 0 \vee x_2 = 1 \Leftrightarrow x_1(x_2 - 1) = 0$$

- (Müller-Olm & Seidl, 2004): polynomial equality invariants in programs with **just disequality conditions**
  - Disadvantages: inhibitor arcs cannot be considered

Related Work (2)

Alternating Bit Protocol



- Linear inequality analysis is too coarse
- There are inhibitor arcs
- Polynomial invariants prove the protocol correct

Overview of the Talk

- **Petri Nets**
  1. Introduction
  2. Modelling with Petri Nets
  3. Generating Invariants
  4. Related Work
  5. Conclusions

- **Hybrid Systems**

Conclusions

- Generated invariants for Petri nets using polynomial invariant inference
- Applied polynomial invariants to show absence of deadlocks
- Shown several non-trivial examples that can be analyzed

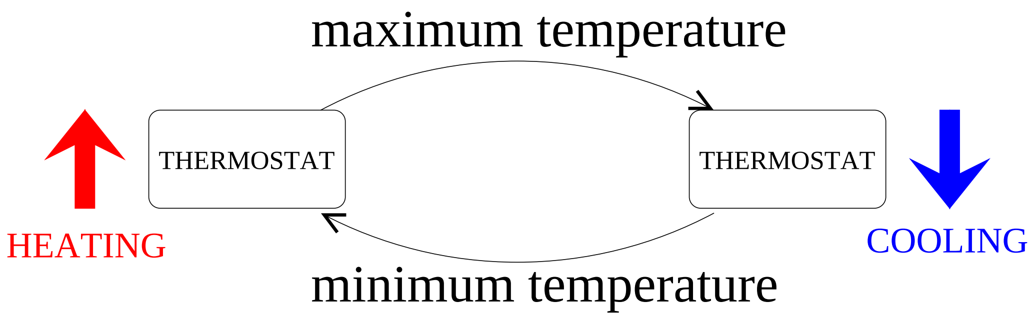
Overview of the Talk

- **Petri Nets**
- **Hybrid Systems**

1. **Introduction**
2. **Preliminaries**
3. **Invariants of Linear Systems**
4. **Invariants of Hybrid Systems**
5. **Related Work**
6. **Conclusions**

Introduction (1)

- **Hybrid Systems:** discrete systems embedded in analog environments
- **Examples:**
  - A **thermostat** that heats/cools depending on the temperature in the room



  - A **robot controller** that changes the direction of movement if the robot is too close to a wall.
  - A **biochemical reaction** whose behaviour depends on the concentration of the substances in the environment

Introduction (2)

- **Applications:**
  - Automotive Control
  - Avionics
  - Transportation Networks
  - Manufacturing
  - Robotics
  - Analysis of Biological Processes

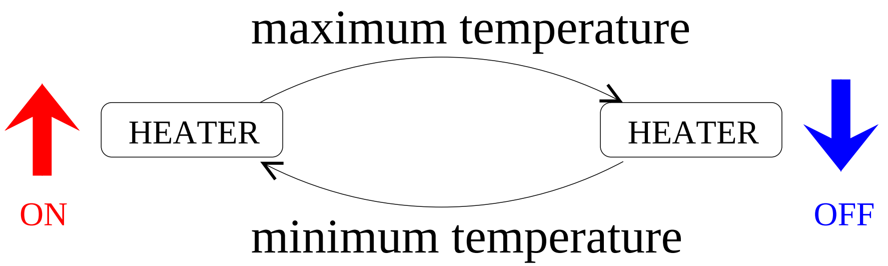
Need for verification of safety properties !

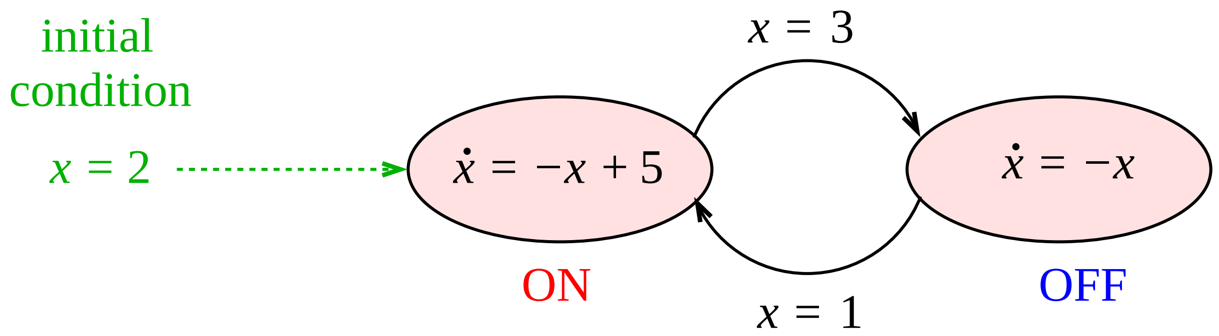
Overview of the Talk

- **Petri Nets**
- **Hybrid Systems**
  1. **Introduction**
  2. **Preliminaries**
  3. **Invariants of Linear Systems**
  4. **Invariants of Hybrid Systems**
  5. **Related Work**
  6. **Conclusions**

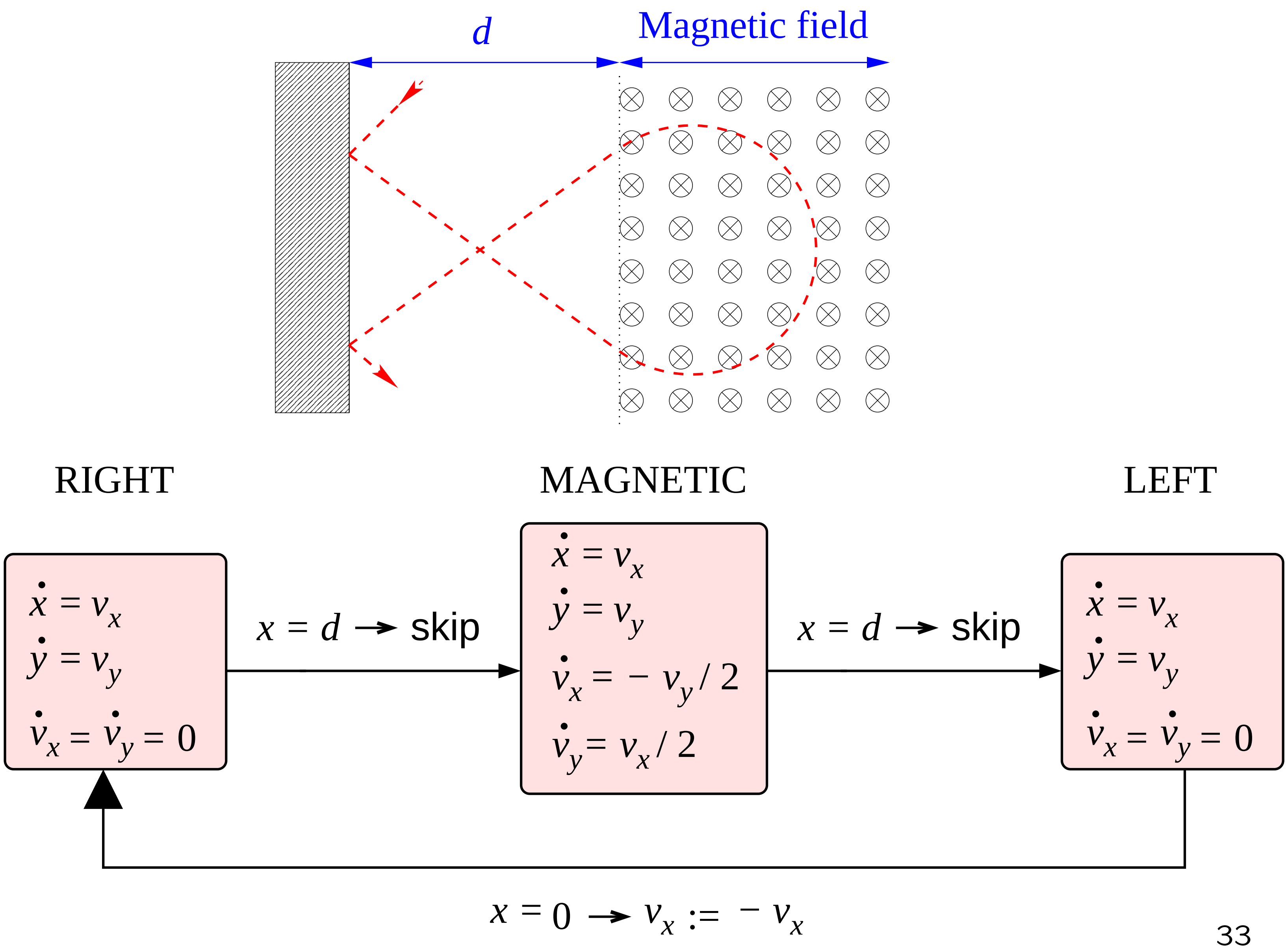
Preliminaries (1)

- A **hybrid system** is a finite automaton with real-valued variables that change continuously according to a system of differential equations at each location of the automaton





- Restrict to linear differential equations at locations


d
Magnetic field
RIGHT
MAGNETIC
LEFT
$\dot{x} = v_x$
$\dot{y} = v_y$
$\dot{v}_x = \dot{v}_y = 0$
$x = d \rightarrow$ skip
$\dot{x} = v_x$
$\dot{y} = v_y$
$\dot{v}_x = -v_y / 2$
$\dot{v}_y = v_x / 2$
$x = d \rightarrow$ skip
$\dot{x} = v_x$
$\dot{y} = v_y$
$\dot{v}_x = \dot{v}_y = 0$
$x = 0 \rightarrow v_x := -v_x$

Preliminaries (2)

- A **computation** is a sequence of states (discrete location,valuation of variables)

  $$(l_0, x_0), (l_1, x_1), (l_2, x_2), \ldots$$

  such that

  1. Initial state (l_0, x_0) satisfies the initial condition
  2. For each consecutive pair of states $(l_i, x_i), (l_{i+1}, x_{i+1})$:
     - **Discrete transition:** there is a transition of the automaton (l_i, l_{i+1}, ρ) such that $(x_i, x_{i+1}) \models \rho$

       or

     - **Continuous evolution:** there is a trajectory going from x_i to x_{i+1} along the flow determined by the differential equation $\dot{x} = Ax + B$ at location $l_i = l_{i+1}$

Preliminaries (3)

- A state is reachable if there exists a computation where it appears
- **Goal:** generate invariant polynomial equalities
  - We know how to deal with discrete systems
  - How to handle continuous evolution ?

$\longrightarrow$ computing polynomial invariants of
linear systems of differential equations

Overview of the Talk

- Petri Nets
- Hybrid Systems

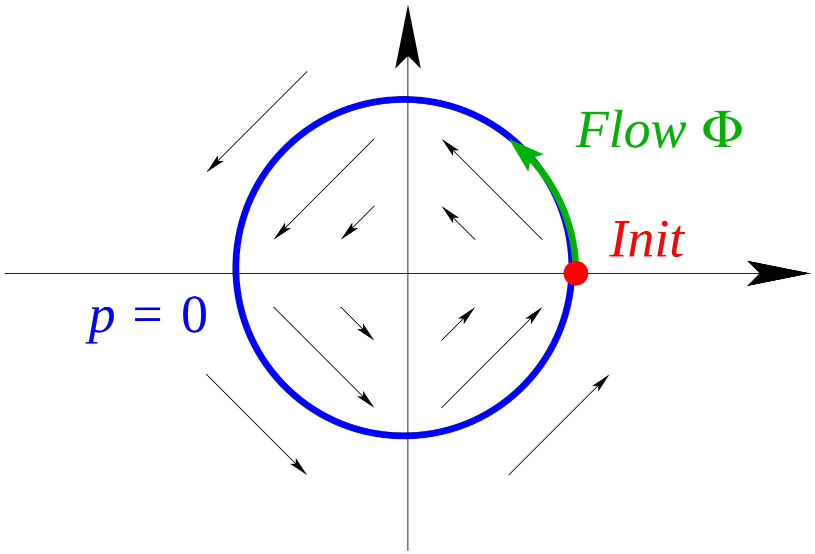
1. Introduction
2. Preliminaries
3. Invariants of Linear Systems
4. Invariants of Hybrid Systems
5. Related Work
6. Conclusions

Invariants of Linear Systems
Problem

- Given a system $\dot{x} = Ax + B$ and a set of initial values $Init$, find polynomials p evaluating to 0 at reachable points:

$$\forall x^* \in Init, \quad \forall t \geq 0 \qquad p(\Phi(x^*, t)) = 0$$

where $\Phi(x^*, t)$ is the flow $\equiv$ solution to $\dot{x} = Ax + B$ with initial condition x^*



Invariants of Linear Systems

Form of the Flow

- Solution to $\dot{x} = Ax + B$ with initial condition x^*

$$\Phi(x^*, t) = e^{At}x^* + e^{At}(\int_0^t e^{-A\tau}d\tau)\ B$$

- Can be expressed as polynomials in t, $e^{\pm at}$, $\cos(bt)$, $\sin(bt)$, where $\lambda = a + bi$ are eigenvalues of matrix A.

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{v_x} \\ \dot{v_y} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1/2 \\ 0 & 0 & 1/2 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ v_x \\ v_y \end{pmatrix}$$

$$\begin{cases} x &= x^* + 2\sin(t/2)\, v_x^* + (2\cos(t/2) - 2)\, v_y^* \\ y &= y^* + (-2\cos(t/2) + 2)\, v_x^* + 2\sin(t/2)\, v_y^* \\ v_x &= \cos(t/2)\, v_x^* - \sin(t/2)\, v_y^* \\ v_y &= \sin(t/2)\, v_x^* + \cos(t/2)\, v_y^* \end{cases}$$

Invariants of Linear Systems
Elimination of Time (1)

- **Idea:** eliminate terms depending on t from the flow
- **Simple case:**
  eigenvalues of matrix A have real and imaginary parts in $\mathbb{Q}$

  - $\exists p \in \mathbb{Q}$ such that for all exponential terms e^{at}:

    $$e^{at} = (e^{pt})^c \text{ for a certain } c \in \mathbb{Z}$$

    If we introduce new variables $u = e^{pt}$, $v = e^{-pt}$,
    then either $e^{at} = u^{|c|}$ or $e^{at} = v^{|c|}$

  - For trigonometric terms similarly for $q \in \mathbb{Q}$ and new variables $w = \cos(qt)$, $z = \sin(qt)$

Invariants of Linear Systems
Elimination of Time (2)

- Eliminate auxiliary variables using $uv = 1$ and $w^2 + z^2 = 1$ by means of Gröbner bases
- Use elimination term ordering with the auxiliary variables the biggest ones

INITIAL CONDITIONS

$$\begin{cases} v_x^* &= 2 \\ v_y^* &= -2 \end{cases}$$

FLOW

$$\begin{cases} x &= x^* + 2zv_x^* + (2w-2)v_y^* \\ y &= y^* + (-2w+2)\,v_x^* + 2zv_y^* \\ v_x &= wv_x^* - zv_y^* \\ v_y &= zv_x^* + wv_y^* \end{cases}$$

AUXILIARY EQUATIONS

$$\{\ w^2 + z^2 = 1$$

$$\Downarrow$$

$$v_x^2 + v_y^2 = 8$$

(conservation of energy)

Invariants of Linear Systems
Elimination of Time (3)

- **General case:** similarly by computing $\mathbb{Q}$-bases of the real and imaginary parts of eigenvalues of matrix A
  - Exponential terms: new variables x_1, y_1, ..., x_k, y_k satisfying $x_i y_i = 1$
  - Trigonometric terms: new variables w_1, z_1, ..., w_l, z_l satisfying $w_j^2 + z_j^2 = 1$
- **All** polynomial invariants of linear system are generated

Overview of the Talk

- **Petri Nets**
- **Hybrid Systems**

1. **Introduction**
2. **Preliminaries**
3. **Invariants of Linear Systems**
4. **Invariants of Hybrid Systems**
5. **Related Work**
6. **Conclusions**

Invariants of Hybrid Systems

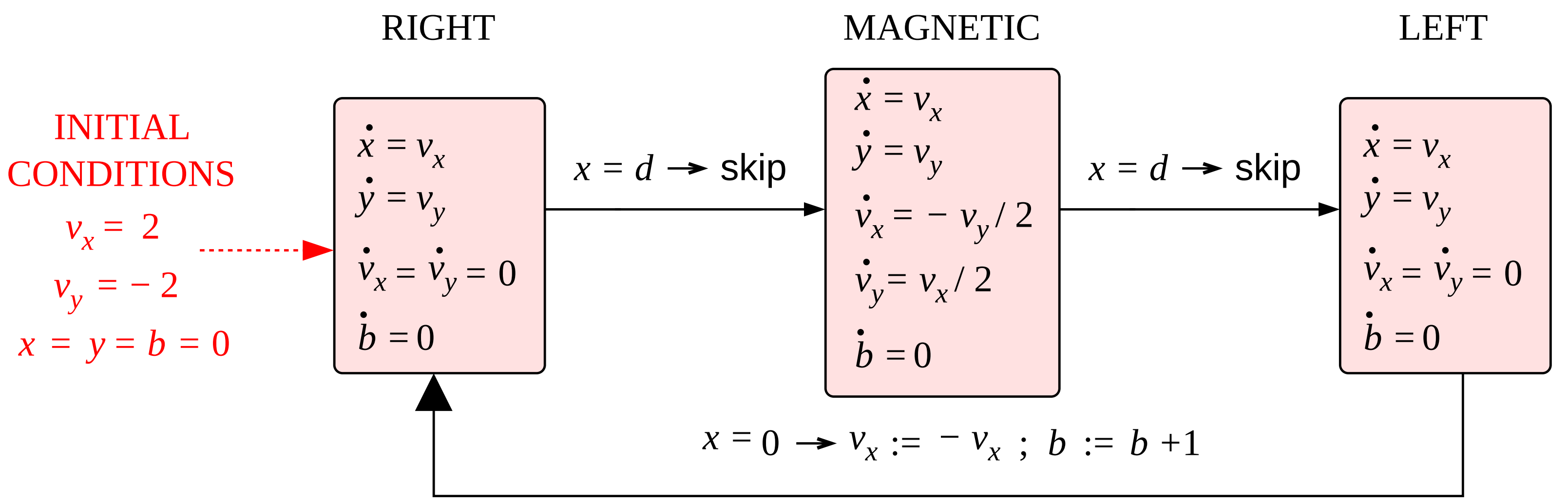
Abstract Semantics of Continuous Evolution

- System of linear differential equations $\dot{x} := Ax + B$ with flow equations Φ_1, ..., Φ_n in variables x, x^*, u_i, v_i, w_j, z_j
- Input ideal: I
- Output ideal:

$$\langle I(x \leftarrow x^*), \Phi_1, ..., \Phi_n, u_i v_i - 1, w_j^2 + z_j^2 - 1 \rangle \cap \mathbb{R}[x]$$

Invariants of Hybrid Systems
Examples (1)

Variable b counts the number of bounces against wall



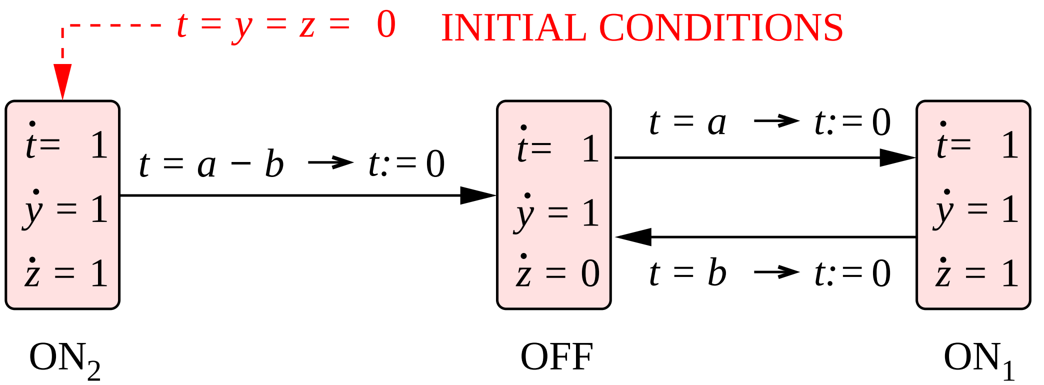
$$\text{RIGHT} \rightarrow v_y = -2 \wedge v_x = 2 \wedge 2db - 8b + y + x = 0$$
$$\text{MAGNETIC} \rightarrow x - 2v_y - d = 4 \wedge v_x^2 + v_y^2 = 8 \wedge 2v_x + y + 2db - 8b + d = 4$$
$$\text{LEFT} \rightarrow v_y = -2 \wedge v_x = -2 \wedge 2db - 8b + y - x = 8$$

Invariants of Hybrid Systems
Examples (2)

Variable t counts the time at current location

Variable y counts the total time elapsed

Variable z counts the time the heater has been on



Safety requirement: heater on $<$ 40 % of the first 60 seconds

$\longrightarrow$ proved using polynomial invariants

$$
\begin{array}{lcl}
ON_2 & \rightarrow & y = t \wedge z = t \\
OFF & \rightarrow & -a^2 + ab + az + bz - by + bt = 0 \\
ON_1 & \rightarrow & a^2 - 2ab - az - bz + by + at = 0
\end{array}
$$

Overview of the Talk

- **Petri Nets**
- **Hybrid Systems**
  1. **Introduction**
  2. **Preliminaries**
  3. **Invariants of Linear Systems**
  4. **Invariants of Hybrid Systems**
  5. **Related Work**
  6. **Conclusions**

Related Work (1)

- (Sankaranarayanan & Sipma & Manna, 2004):
  discovery of polynomial equality invariants using constrained-based invariant generation and heuristics
- Advantages:
  - Polynomial vector fields allowed in differential equations
- Disadvantages:
  - No completeness result

Related Work (2)

- (Laferriere & Pappas & Yovine, 1999):
  computation of **exact** reachability set using
  polynomial inequalities and quantifier elimination

- Advantages:
  - Polynomial inequalities more expressive than equalities:
    exact characterization of reachability set

- Disadvantages:
  - More restricted linear systems: eigenvalues in $\mathbb{Q}$ or $i \cdot \mathbb{Q}$
  - No extension to hybrid systems
  - Quantifier elimination more costly than Gröbner bases

Overview of the Talk

- **Petri Nets**
- **Hybrid Systems**

1. **Introduction**
2. **Preliminaries**
3. **Invariants of Linear Systems**
4. **Invariants of Hybrid Systems**
5. **Related Work**
6. **Conclusions**

Conclusions

- Method for finding **all** polynomial equality invariants of general linear systems:

  1. Solve differential equations

  2. Eliminate time with Gröbner bases

     - Auxiliary variables

$$\begin{array}{lcl} u_i & \leftrightarrow & e^{pt} \\ v_i & \leftrightarrow & e^{-pt} \end{array} \qquad \begin{array}{lcl} w_i & \leftrightarrow & \cos(qt) \\ z_i & \leftrightarrow & \sin(qt) \end{array}$$

     - Auxiliary equations:

$$u_i v_i = 1, \qquad w_i^2 + z_i^2 = 1$$

- Extension to hybrid systems using the abstract interpretation framework