

Petri Nets (for Planners)

B. Bonet, P. Haslum

... from various places ...

ICAPS 2011

Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Introduction & Motivation

- Petri Nets (PNs) is formalism for modelling discrete event systems
- Developed by (and named after) C.A. Petri in 1960s
- In general Petri nets, places are **unbounded counters**
 - advantages in expressivity and modelling convenience
 - questions of reachability, coverability, etc. are computationally harder to answer, but still decidable

Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Exchange of ideas between Petri nets and planning holds potential to benefit both areas:

- Analysis methods for Petri nets are often based on ideas & techniques not common in planning:
 - algebraic methods based on the state equation
 - rich literature on the study of classes of nets with special structure
- Yet, some standard planning techniques (e.g., search heuristics) are unknown in the PN community

Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Outline of the Tutorial

- 1 Definitions, notation and modelling
- 2 Decision problems, complexity and expressivity
- 3 Analysis techniques for general Petri nets
 - Coverability
 - The state equation
 - Reachability
- 4 Petri nets with special structure
- 5 Conclusions

Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Definitions, Notation and Modelling

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Terminology and Intuition

- A Petri net has **places**, **transitions**, and **directed arcs**
- Arcs connect places to transitions or vice versa
- Places contain zero or finite number of **tokens**
- A **marking** is disposition of tokens in places
- A transition is **fireable** if there is token at the start place of **each input arc**
- When transition fires:
 - it **consumes** token from start place of each input arc
 - it **puts** token at end place of each output arc
- Execution is **non-deterministic**

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

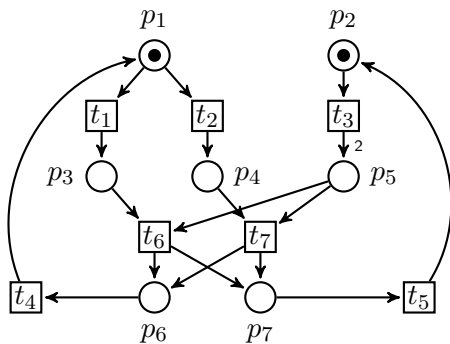
Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Example



Introduction

Basics

- Definitions
- Ordinary Nets
- Types of Nets
- Vector Notation

Complexity & Expressivity

Analysis Techniques

Special Classes of Nets

Conclusion

Formal Definition

Place/Transition (P/T) net is tuple $N = (P, T, W)$ where:

- P is set of places
- T is set of transitions (and $P \cap T = \emptyset$)
- $W \subseteq (P \times T) \cup (T \times P) \rightarrow \mathbb{N}$
(**multiset** of arcs: each (x, y) has **multiplicity** $W(x, y)$)

For transition t :

- **preset** is ${}^\bullet t = \{s : W(s, t) > 0\}$ (input places)
- **postset** is $t^\bullet = \{s : W(t, s) > 0\}$ (output places)

Marking is $\mathbf{m} : P \rightarrow \mathbb{N}$ (zero or more tokens at each place)

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

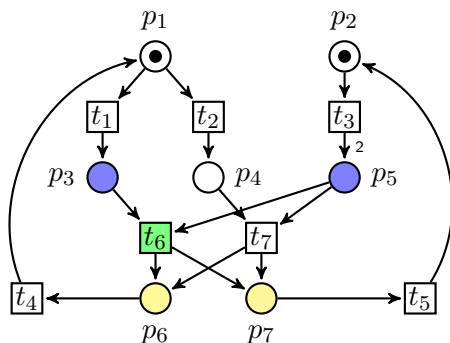
Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Example



- marking $\mathbf{m} = \langle 1, 1, 0, 0, 0, 0, 0 \rangle$
- transition t_6 : $\bullet t_6 = \{p_3, p_5\}$, $t_6 \bullet = \{p_6, p_7\}$

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Execution Semantics

A transition t is **enabled** or **firable** at marking \mathbf{m} if

$$\mathbf{m}(p) \geq W(p, t) \quad \text{for each } p \in \bullet t$$

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Execution Semantics

A transition t is **enabled** or **firable** at marking \mathbf{m} if

$$\mathbf{m}(p) \geq W(p, t) \quad \text{for each } p \in \bullet t$$

Upon firing, t **produces** new marking \mathbf{m}' such that

$$\mathbf{m}'(p) = \begin{cases} \mathbf{m}(p) - \overbrace{W(p, t)}^{\text{consumed}} + \overbrace{W(t, p)}^{\text{added}} & \text{if } p \in \bullet t \cup t\bullet \\ \mathbf{m}(p) & \text{if } p \notin \bullet t \cup t\bullet \end{cases}$$

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Execution Semantics

Transition relations:

- $\mathbf{m} [t \rangle \mathbf{m}'$ if t is enabled at \mathbf{m} and produces \mathbf{m}'
- $\mathbf{m} [\sigma \rangle \mathbf{m}'$, for sequence $\sigma = t_1 t_2 \cdots t_n$, if exists \mathbf{m}'' with
 - $\mathbf{m} [t_1 \rangle \mathbf{m}''$
 - $\mathbf{m}'' [\sigma' \rangle \mathbf{m}'$ for $\sigma' = t_2 \cdots t_n$
- $\mathbf{m} [\rangle \mathbf{m}'$ if $\mathbf{m} [t \rangle \mathbf{m}'$ for **some** t
- $[* \rangle$ is transitive closure of $[\rangle$

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Execution Semantics

Transition relations:

- $\mathbf{m} [t \rangle \mathbf{m}'$ if t is enabled at \mathbf{m} and produces \mathbf{m}'
- $\mathbf{m} [\sigma \rangle \mathbf{m}'$, for sequence $\sigma = t_1 t_2 \cdots t_n$, if exists \mathbf{m}'' with
 - $\mathbf{m} [t_1 \rangle \mathbf{m}''$
 - $\mathbf{m}'' [\sigma' \rangle \mathbf{m}'$ for $\sigma' = t_2 \cdots t_n$
- $\mathbf{m} [\rangle \mathbf{m}'$ if $\mathbf{m} [t \rangle \mathbf{m}'$ for **some** t
- $[* \rangle$ is transitive closure of $[\rangle$

Marked net is $(N = (P, T, W), \mathbf{m}_0)$ where \mathbf{m}_0 is **initial marking**

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Execution Semantics

Transition relations:

- $\mathbf{m} [t \rangle \mathbf{m}'$ if t is enabled at \mathbf{m} and produces \mathbf{m}'
- $\mathbf{m} [\sigma \rangle \mathbf{m}'$, for sequence $\sigma = t_1 t_2 \cdots t_n$, if exists \mathbf{m}'' with
 - $\mathbf{m} [t_1 \rangle \mathbf{m}''$
 - $\mathbf{m}'' [\sigma' \rangle \mathbf{m}'$ for $\sigma' = t_2 \cdots t_n$
- $\mathbf{m} [\rangle \mathbf{m}'$ if $\mathbf{m} [t \rangle \mathbf{m}'$ for **some** t
- $[\ast \rangle$ is transitive closure of $[\rangle$

Marked net is $(N = (P, T, W), \mathbf{m}_0)$ where \mathbf{m}_0 is **initial marking**

Reachable markings $R(N, \mathbf{m}_0) = \{\mathbf{m} \in \mathbb{N}^P : \mathbf{m}_0 [\ast \rangle \mathbf{m}\}$

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Execution Semantics

Transition relations:

- $\mathbf{m} [t \rangle \mathbf{m}'$ if t is enabled at \mathbf{m} and produces \mathbf{m}'
- $\mathbf{m} [\sigma \rangle \mathbf{m}'$, for sequence $\sigma = t_1 t_2 \cdots t_n$, if exists \mathbf{m}'' with
 - $\mathbf{m} [t_1 \rangle \mathbf{m}''$
 - $\mathbf{m}'' [\sigma' \rangle \mathbf{m}'$ for $\sigma' = t_2 \cdots t_n$
- $\mathbf{m} [\rangle \mathbf{m}'$ if $\mathbf{m} [t \rangle \mathbf{m}'$ for **some** t
- $[* \rangle$ is transitive closure of $[\rangle$

Marked net is $(N = (P, T, W), \mathbf{m}_0)$ where \mathbf{m}_0 is **initial marking**

Reachable markings $R(N, \mathbf{m}_0) = \{\mathbf{m} \in \mathbb{N}^P : \mathbf{m}_0 [* \rangle \mathbf{m}\}$

Firing sequences $L(N, \mathbf{m}_0) = \{\sigma \in T^{<\infty} : \exists \mathbf{m}. \mathbf{m}_0 [\sigma \rangle \mathbf{m}\}$

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

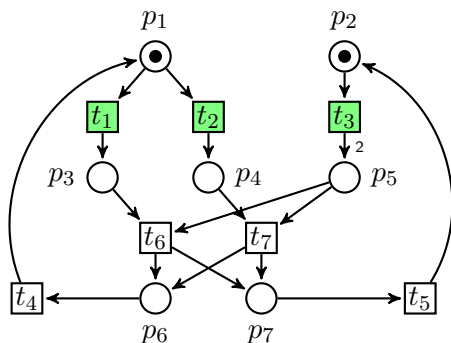
Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Example



- **marking** $\mathbf{m} = \langle 1, 1, 0, 0, 0, 0, 0 \rangle$
- **enabled:** t_1, t_2, t_3

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

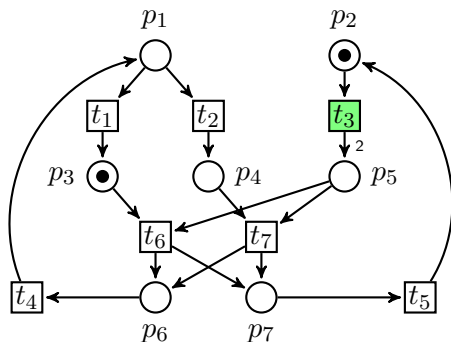
Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Example



- **marking** $\mathbf{m} = \langle 0, 1, 1, 0, 0, 0, 0 \rangle$
- **enabled:** t_3

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

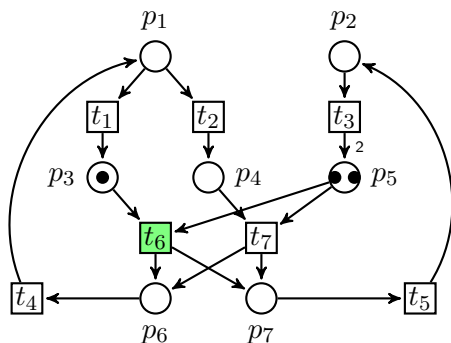
Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Example



- **marking** $\mathbf{m} = \langle 0, 0, 1, 0, 2, 0, 0 \rangle$
- **enabled:** t_6

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

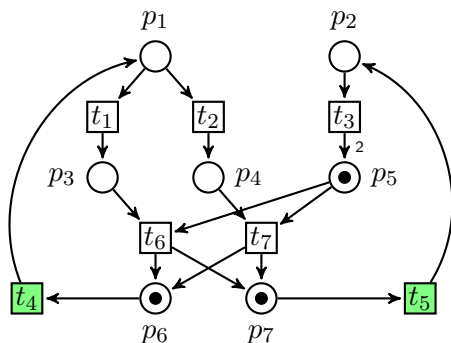
Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Example



- **marking** $\mathbf{m} = \langle 0, 0, 0, 0, 1, 1, 1 \rangle$
- **enabled:** t_4, t_5

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Arithmetic of Functions

For two functions $f, g \in \mathbb{N}^X$:

- $f \geq g$ if $f(x) \geq g(x)$ for each place x
- $f > g$ if $f \geq g$ and there is x such that $f(x) > g(x)$
- $f + g$ defined pointwise as $(f + g)(x) = f(x) + g(x)$

Hence, $\mathbf{m} [t] \mathbf{m}'$ iff

$$\mathbf{m} \geq W(\cdot, t) \quad (\text{enable condition})$$

$$\mathbf{m}' = \mathbf{m} - W(\cdot, t) + W(t, \cdot)$$

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

Complexity &
Expressivity

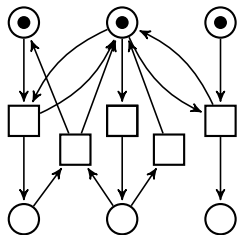
Analysis
Techniques

Special
Classes of
Nets

Conclusion

Ordinary Nets

A P/T net $N = (P, T, W)$ is **ordinary** iff $W(p, t) \leq 1$ for all p, t



Thm: any net can be transformed into **equivalent** ordinary net

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

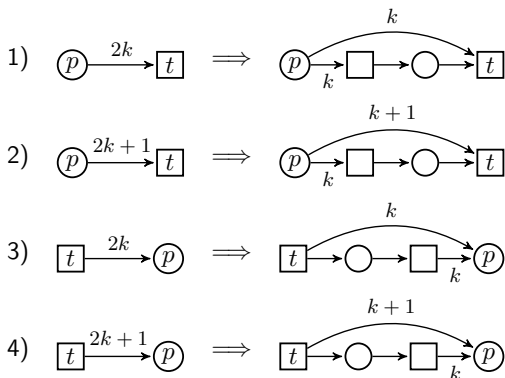
Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Transformation Rules



Each rule **decrease multiplicity** by half and add 2 nodes

Resulting size is $O(\sum_{x,y} W(x,y))$ (exponential)

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

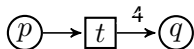
Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Example



Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

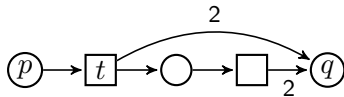
Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Example



Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

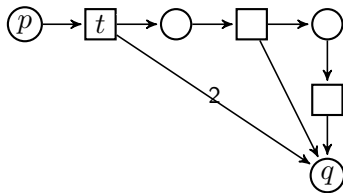
Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Example



Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

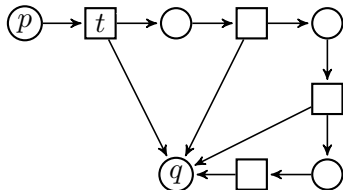
Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Example



Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Types of Nets

- Marking \mathbf{m} is **k -bounded** if $\mathbf{m}(p) \leq k$ for all $p \in P$
- Marked net (N, \mathbf{m}_0) is **k -bounded** if every reachable marking is k -bounded
- It is **bounded** if it is k -bounded for some k
- It is **safe** if it is 1-bounded

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Safe Networks

- Every reachable marking is 1-bounded
- Marking \mathbf{m} can be thought as **state** where places represents fluents:
 - if $\mathbf{m}(p) = 1$ then fluent p is **true** at \mathbf{m}
 - if $\mathbf{m}(p) = 0$ then fluent p is **false** at \mathbf{m}
- Safe networks can be used for STRIPS planning

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Direct STRIPS to PN Translations

- Each atom is a place
- Each (grounded) action is a transition t :
 - input arcs $p \rightarrow t$ for each precondition p
 - output arcs $t \rightarrow p$ for each positive effect p
 - output arcs $t \rightarrow p$ for each precondition p that is not deleted nor added
- Initial state gives initial marking
- Goal state gives **partial** desired marking
- Plan existence becomes "Coverability" problem

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

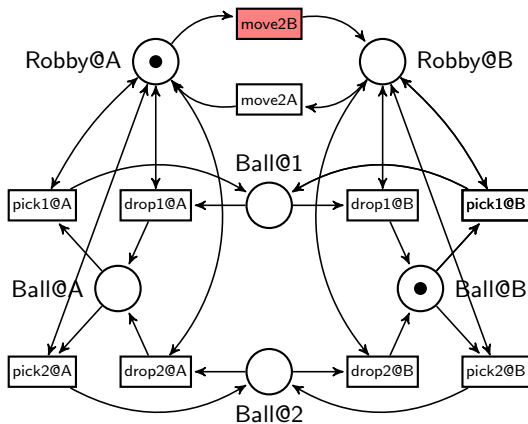
Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Example: Gripper w/ 1 Ball and 2 Arms



move2B, pick1@B, move2A, drop1@A

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

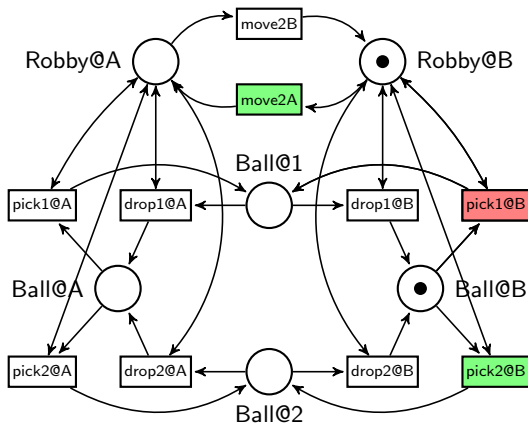
Complexity & Expressivity

Analysis Techniques

Special Classes of Nets

Conclusion

Example: Gripper w/ 1 Ball and 2 Arms



move2B, pick1@B, move2A, drop1@A

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

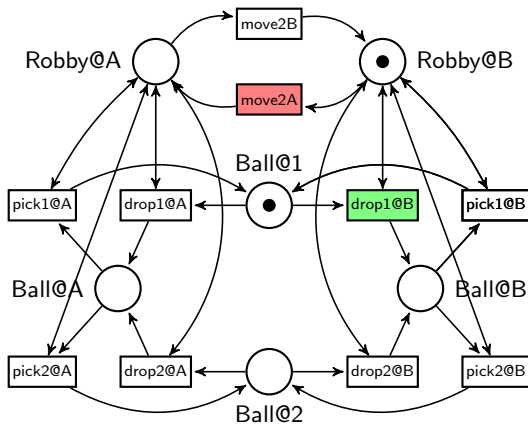
Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Example: Gripper w/ 1 Ball and 2 Arms



move2B, pick1@B, **move2A**, drop1@A

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

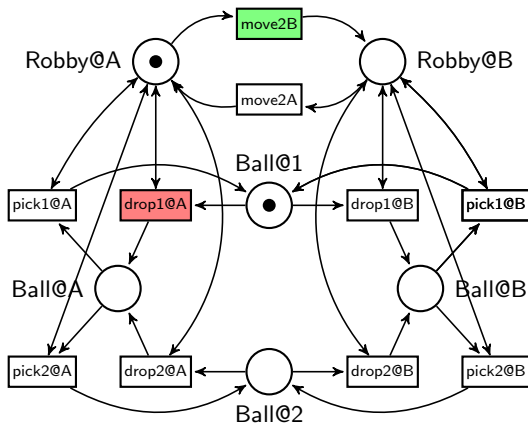
Complexity & Expressivity

Analysis Techniques

Special Classes of Nets

Conclusion

Example: Gripper w/ 1 Ball and 2 Arms



move2B, pick1@B, move2A, **drop1@A**

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

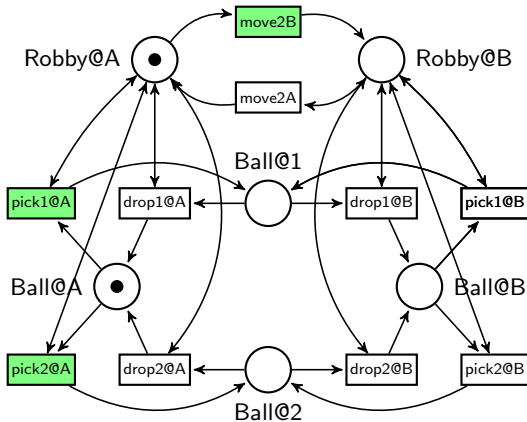
Complexity & Expressivity

Analysis Techniques

Special Classes of Nets

Conclusion

Example: Gripper w/ 1 Ball and 2 Arms



move2B, pick1@B, move2A, drop1@A

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Safe STRIPS Problems

STRIPS problem is **safe** if its direct translation (N, \mathbf{m}_0) is **safe**

Sufficient Condition:

- For each added atom p , there is precondition q that is deleted such that $\{p, q\}$ is mutex

Enforcing the condition:

- Add 'not- p ' atoms for each atom p
- For each action that contains a deleted atom p that is not precondition, generate two similar actions with p and not- p in precondition (respectively)
- Worst-case size of pre-processing is exponential in number of atoms that are deleted and don't appear as preconditions

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Modelling Planning Problems

General nets can “store” multiple tokens at single place

Places can be used to represent:

- **number** of identical objects at location
- **resource quantity**

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

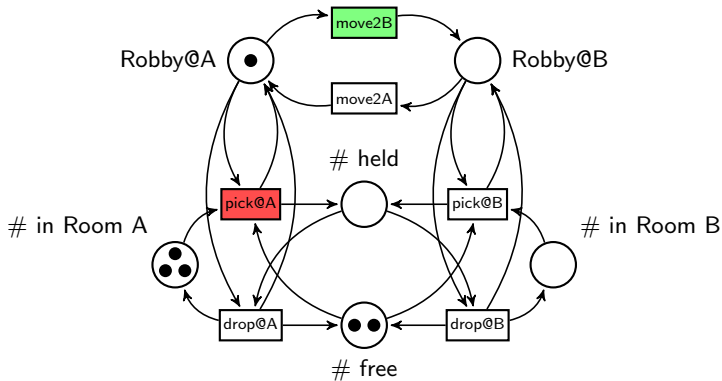
Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Gripper with 3 identical balls and 2 identical arms



pick@A, pick@A, move2B, drop@B, drop@B, move2A, pick@A, move2B, drop@B

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

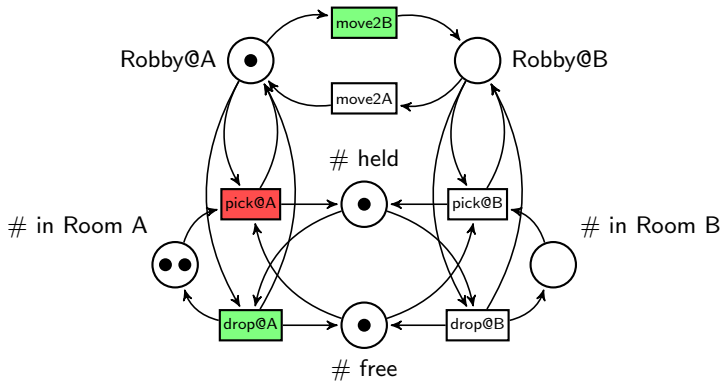
Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Gripper with 3 identical balls and 2 identical arms



pick@A, **pick@A**, move2B, drop@B, drop@B, move2A, pick@A, move2B, drop@B

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

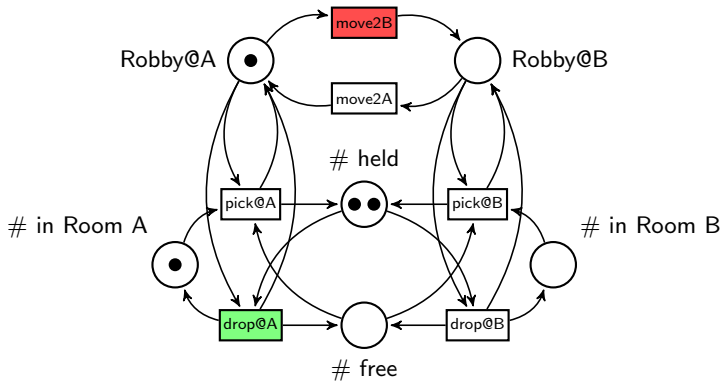
Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Gripper with 3 identical balls and 2 identical arms



pick@A, pick@A, **move2B**, drop@B, drop@B, move2A, pick@A, move2B, drop@B

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

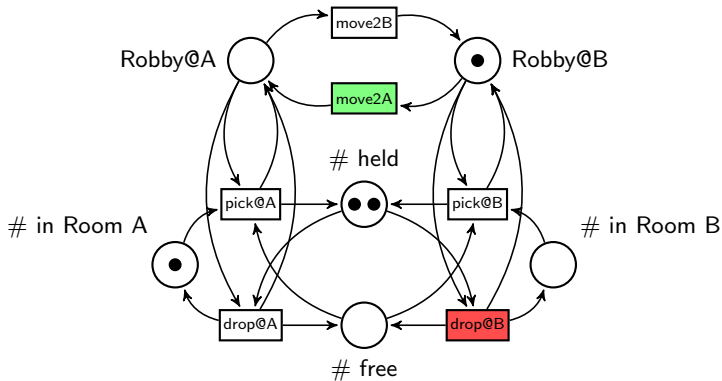
Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Gripper with 3 identical balls and 2 identical arms



pick@A, pick@A, move2B, **drop@B**, drop@B, move2A, pick@A, move2B, drop@B

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

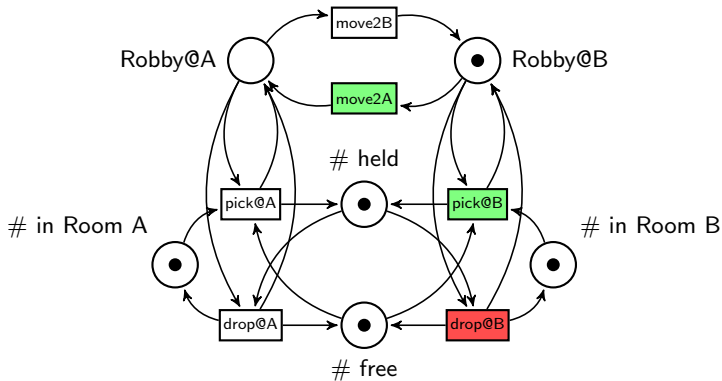
Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Gripper with 3 identical balls and 2 identical arms



pick@A, pick@A, move2B, drop@B, **drop@B**, move2A, pick@A, move2B, drop@B

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

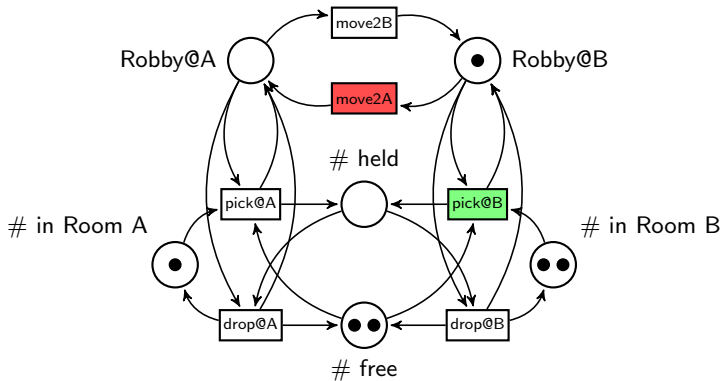
Complexity & Expressivity

Analysis Techniques

Special Classes of Nets

Conclusion

Gripper with 3 identical balls and 2 identical arms



pick@A, pick@A, move2B, drop@B, drop@B, **move2A**, pick@A, move2B, drop@B

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

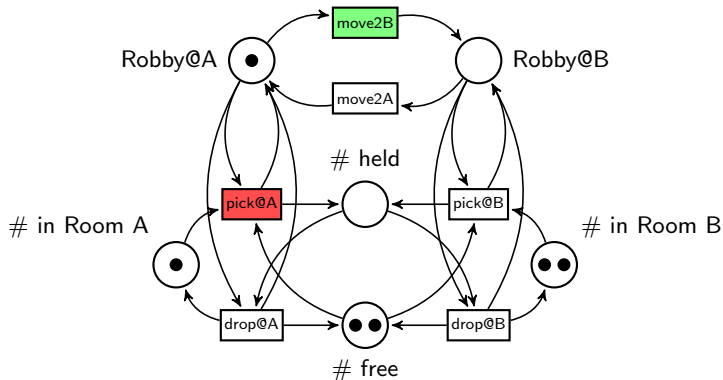
Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Gripper with 3 identical balls and 2 identical arms



pick@A, pick@A, move2B, drop@B, drop@B, move2A, **pick@A**, move2B, drop@B

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

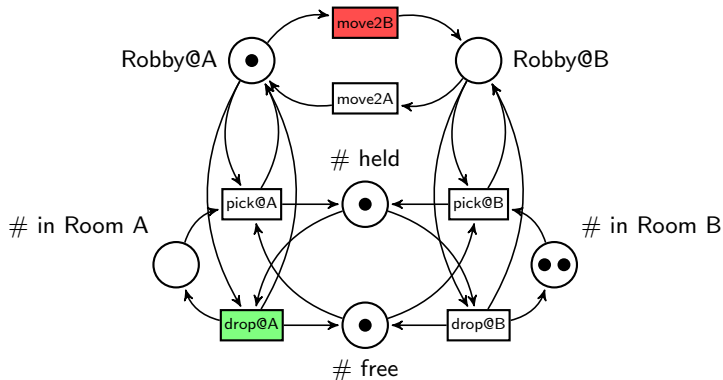
Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Gripper with 3 identical balls and 2 identical arms



pick@A, pick@A, move2B, drop@B, drop@B, move2A, pick@A, **move2B**, drop@B

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

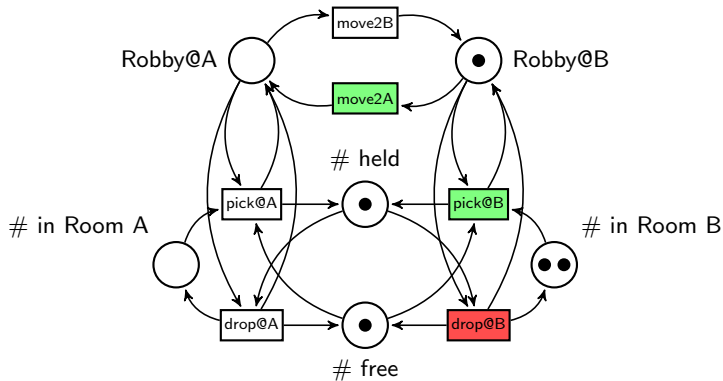
Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Gripper with 3 identical balls and 2 identical arms



pick@A, pick@A, move2B, drop@B, drop@B, move2A, pick@A, move2B, **drop@B**

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

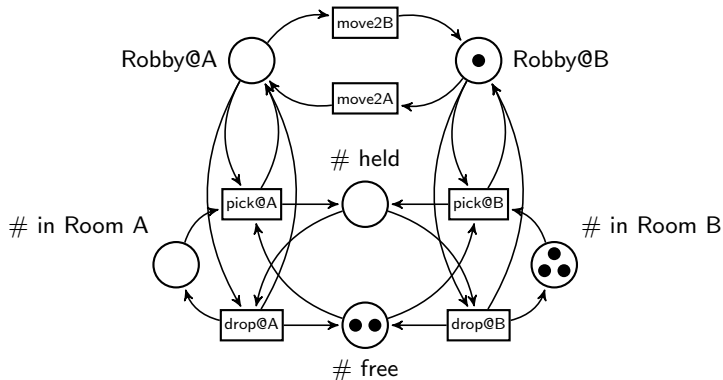
Complexity & Expressivity

Analysis Techniques

Special Classes of Nets

Conclusion

Gripper with 3 identical balls and 2 identical arms



pick@A, pick@A, move2B, drop@B, drop@B, move2A, pick@A, move2B, drop@B

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

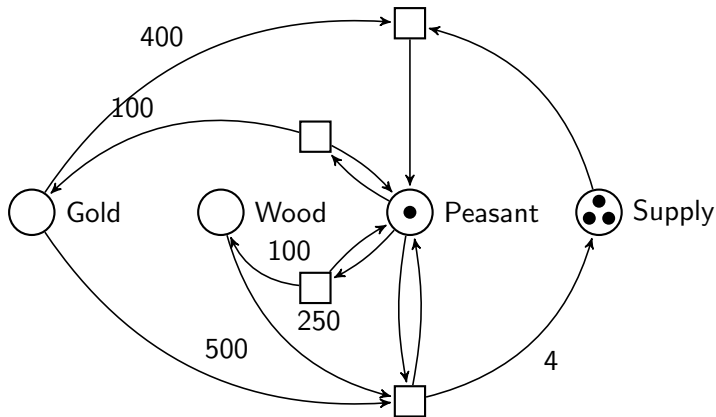
Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Wargus Domain (Chan et al. 2007)



Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Other Types of Nets

State Machines:

- every **transition** has one incoming and one outgoing arc
i.e. $|\bullet t| = |t\bullet| = 1$ for each $t \in T$

Marked Graphs:

- every **place** has one incoming arc, and one outgoing arc
i.e. $|\bullet p| = |p\bullet| = 1$ for each $p \in P$

Free-choice Nets:

- every arc is either the **only arc going from** the place, or **only arc going to** the transition
i.e. $|p\bullet| \leq 1$ or $\bullet(p\bullet) = \{p\}$ for each $p \in P$

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Extensions

Inhibitor arcs (enablers):

- transition enabled when there is **no token at place**

Read arcs (enablers):

- do not **consume** tokens

Reset arcs: erase all tokens at place

Others:

- colored, hierarchical, prioritization, ...

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Vector Notation

Two **vectors** associated with transition t :

$$\mathbf{W}_t^- = \begin{pmatrix} W(p_1, t) \\ \vdots \\ W(p_{|P|}, t) \end{pmatrix} \quad \mathbf{W}_t^+ = \begin{pmatrix} W(t, p_1) \\ \vdots \\ W(t, p_{|P|}) \end{pmatrix}$$

- t enabled at \mathbf{m} iff $\mathbf{m} \geq \mathbf{W}_t^-$
- $\mathbf{W}_t = \mathbf{W}_t^+ - \mathbf{W}_t^-$ is **effect** of t
- firing t leads to $\mathbf{m}' = \mathbf{m} + \mathbf{W}_t$
- $\mathbf{W} = \left(\mathbf{W}_{t_1}, \mathbf{W}_{t_2}, \dots, \mathbf{W}_{t_{|T|}} \right)$ is **incidence matrix**
- \mathbf{r}_p : row of \mathbf{W} corresponding to place p

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

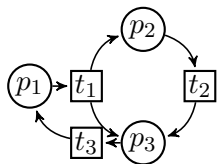
Complexity &
Expressivity

Analysis
Techniques

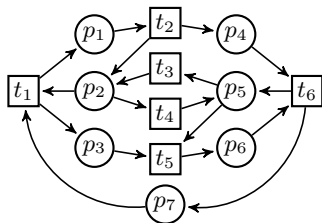
Special
Classes of
Nets

Conclusion

Examples



$$\mathbf{W} = \begin{pmatrix} -1 & 0 & 1 \\ 1 & -1 & 0 \\ 1 & 1 & -1 \end{pmatrix}$$



$$\mathbf{W} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

Complexity &
Expressivity

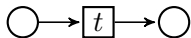
Analysis
Techniques

Special
Classes of
Nets

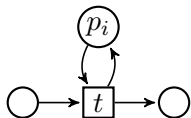
Conclusion

Representation Ambiguity and Pure Nets

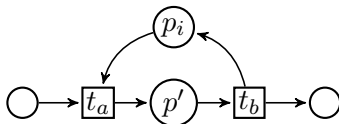
$$\mathbf{W}_t[i] = 0:$$



or



- **Pure nets** have no “self loops”:
• $t \cap t^\bullet = \emptyset$ for every transition t
- For pure nets, incidence matrix \mathbf{W} **unambiguously** defines the net
- Any net can be transformed into a pure net by **splitting loops**:



Transformation is **linear space**

Introduction

Basics

Definitions

Ordinary Nets

Types of Nets

Vector Notation

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Complexity & Expressivity

Introduction

Basics

**Complexity &
Expressivity**

Properties
Equivalence
Structural
Properties
Expressivity
Invariants

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Decision Problems for Marked Nets

Given a marked net (N, \mathbf{m}_0) :

- **Reachability:** Is there a firing sequence that ends with given marking \mathbf{m} ?
- **Coverability:** Is there a firing sequence that ends with marking \mathbf{m}' such that $\mathbf{m}' \geq \mathbf{m}$ for given \mathbf{m} ?
- **Boundedness:** Does there exist a integer k such that every reachable marking is k -bounded? $\mathbf{m} \leq \mathbf{K}$?

Coverability and boundedness are EXPSPACE-complete

Reachability is EXPSPACE-hard, but existing algorithms are non-primitive recursive (i.e., have unbounded complexity)

Introduction

Basics

Complexity &
Expressivity

Properties

Equivalence

Structural

Properties

Expressivity

Invariants

Analysis

Techniques

Special

Classes of

Nets

Conclusion

More Properties

- **Executability:** Is there a firing sequence valid at \mathbf{m}_0 that includes transition t ?
 - Reduces to coverability: t is executable iff \mathbf{W}_t^- is coverable
 - and vice versa: reduction using a “goal transition”
- **Repeated Executability:** Is there a firing sequence in which a given transition (or set of transitions) occurs an infinite number of times?
- **Reachable Deadlock:** Is there a reachable marking \mathbf{m} at which no transition is enabled?
- **Liveness:** Executability of every transition at every reachable marking, i.e.,
$$\forall M : M_0 [* \rangle M \rightarrow \forall t \exists M', M'' : M [* \rangle M' [t \rangle M''.$$

... and many more ...

Introduction

Basics

Complexity &
Expressivity

Properties

Equivalence
Structural
Properties
Expressivity
Invariants

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Equivalence Problems

- **Equivalence:** Given two marked nets, (N_1, \mathbf{m}_1) and (N_2, \mathbf{m}_2) , with equal (or isomorphic) sets of places, do they have the equal sets of reachable markings?
- **Trace Equivalence:** Given two marked nets, (N_1, \mathbf{m}_1) and (N_2, \mathbf{m}_2) , with equal (or isomorphic) sets of transitions, do they have equal sets of valid firing sequences?
- **Language Equivalence:** Trace equivalence under mapping of transitions to a common alphabet
- **Bisimulation:** Equivalence under a bijection between markings

In general, equivalence problems are undecidable

Introduction

Basics

Complexity &
Expressivity

Properties

Equivalence

Structural

Properties

Expressivity

Invariants

Analysis

Techniques

Special

Classes of

Nets

Conclusion

Structural Properties

A **structural property** is **independent** of initial marking \mathbf{m}_0

- **Structural Liveness:** Is there a marking \mathbf{m} such that (N, \mathbf{m}) is live?
- **Structural Boundedness:** Is (N, \mathbf{m}) bounded for every finite initial marking \mathbf{m} ?
- **Repetitiveness:** Is there a marking \mathbf{m} and a firing sequence σ valid at \mathbf{m} such that a given transition (set of transitions) appears infinitely often in σ ?

Deciding structural properties can be easier than corresponding problem for marked net

Structural boundedness and repetitiveness are in NP

Introduction

Basics

Complexity &
Expressivity

Properties
Equivalence
**Structural
Properties**
Expressivity
Invariants

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Complexity: Implications Of and For Expressivity

- Bounded Petri nets are **expressively equivalent** to propositional STRIPS/PDDL
 - Reachability is PSPACE-complete for both
 - **Recall:** direct STRIPS to PN translation may blow up exponentially
- General Petri nets are **strictly more expressive** than propositional STRIPS/PDDL
- General Petri nets are **at least as expressive** as “lifted” (finite 1st order) STRIPS/PDDL
 - probably also strictly more expressive (but no proof yet)

Introduction

Basics

Complexity &
Expressivity

Properties
Equivalence
Structural
Properties
Expressivity
Invariants

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Counter TMs

- A *k-counter machine* (*k*CM) is a **deterministic** finite automaton with *k* (positive) integer counters
 - can increment/decrement (by 1), or reset, counters
 - conditional jumps on $c_i > 0$ or $c_i = 0$
- Note the differences:
 - *k*CMs are **deterministic**: starting configuration determines unique execution; **Petri nets have choice**
 - *k*CMs can **branch on** $c_i > 0/c_i = 0$; Petri nets can only precondition transitions on $\mathbf{m}(p_i) > 0$
- A *k*CM is ***k*-bounded** iff no counter ever exceeds *k*

Introduction

Basics

Complexity &
Expressivity

Properties
Equivalence
Structural
Properties
Expressivity
Invariants

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Counter TMs: Results

- An n -size TM can be simulated by an $O(n)$ -size 2CM (if properly initialised)
 - Halting (i.e., reachability) for unbounded 2CMs is undecidable
 - PNs are strictly **less expressive** than unbounded 2CMs
- An n -size and 2^n space bounded TM can be simulated by $O(n)$ -size 2^{2^n} -bounded 2CM
- A 2^{2^n} -bounded n -size 2CM can be (non-deterministically!) simulated by $O(n^2)$ -size Petri net
 - **Reachability for Petri nets is $\text{DSPACE}(2^{\sqrt{n}})$ -hard**

Introduction

Basics

Complexity & Expressivity

Properties
Equivalence
Structural Properties
Expressivity
Invariants

Analysis Techniques

Special Classes of Nets

Conclusion

Invariants

- A vector $\mathbf{y} \in \mathbb{N}^{|P|}$ is **P-invariant** for N iff for any markings $\mathbf{m} \rightarrow \mathbf{m}'$, $\mathbf{y}^T \mathbf{m} = \mathbf{y}^T \mathbf{m}'$

P-invariant = linear combination of place markings that is invariant under any transition firing

- A vector $\mathbf{x} \in \mathbb{N}^{|T|}$ is a **T-invariant** for N iff for any firing sequence σ such that $\mathbf{n}(\sigma) = \mathbf{x}$ and any marking \mathbf{m} where σ is enabled, $\mathbf{m} \rightarrow \mathbf{m}$

T-invariant = multiset of transitions whose combined effect is zero

Introduction

Basics

Complexity &
Expressivity

Properties
Equivalence
Structural
Properties
Expressivity
Invariants

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Analysis Techniques

Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

The Coverability Tree Construction

- The *coverability tree* of a marked net (N, \mathbf{m}_0) is an explicit representation of reachable markings – but not *exactly* the set of reachable markings.
- Constructed by forwards exploration:
 - Each enabled transition generates a successor marking.
 - If reach \mathbf{m} such that $\mathbf{m} > \mathbf{m}'$ for some ancestor \mathbf{m}' of \mathbf{m} , replace $\mathbf{m}[i]$ by ω for all i s.t. $\mathbf{m}[i] > \mathbf{m}'[i]$.
 - $\mathbf{m}'[s = t_1, \dots, t_l] \mathbf{m}$, and since $\mathbf{m} \geq \mathbf{m}'$, $\mathbf{m}[s] \mathbf{m}''$ such that $\mathbf{m}'' \geq \mathbf{m}$; sequence s can be repeated any number of times.
 - ω means “arbitrarily large”.
 - Also check for regular loops ($\mathbf{m} = \mathbf{m}'$ for some ancestor \mathbf{m}' of \mathbf{m}).
- Every branch has finite depth.

Introduction

Basics

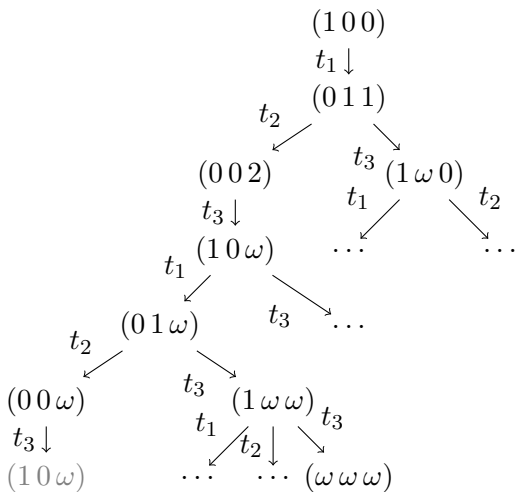
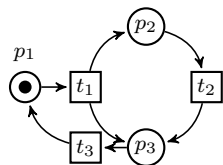
Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Example



Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Uses For The Coverability Tree

- Decides coverability:
 - \mathbf{m} is coverable iff $\mathbf{m} \leq \mathbf{m}'$ for some \mathbf{m}' in the tree (where $n < \omega$ for any $n \in \mathbb{N}$).
 - If \mathbf{m} is coverable, there exists a covering sequence of length at most $O(2^n)$.
- Decides boundedness:
 - (N, \mathbf{m}_0) is unbounded iff there exists a self-covering sequence: $\mathbf{m}_0 [\sigma) \mathbf{m} [\sigma') \mathbf{m}'$ such that $\mathbf{m}' > \mathbf{m}$.
 - I.e., (N, \mathbf{m}_0) is unbounded iff ω appears in some marking in the coverability tree.
 - If (N, \mathbf{m}_0) is unbounded, there exists a self-covering sequence of length at most $O(2^n)$.
- In general, does *not* decide reachability.
 - Except if (N, \mathbf{m}_0) is bounded.

Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

The State Equation

- The *firing count vector* (a.k.a. *Parikh vector*) of a firing sequence $\sigma = t_{i_1}, \dots, t_{i_l}$ is a $|T|$ -dimensional vector $\mathbf{n}(\sigma) = (n_1, \dots, n_{|T|})$ where $n_i \in \mathbb{N}$ is the number of occurrences of transition t_i in σ .
- If $\mathbf{m}_0 \langle \sigma \rangle \mathbf{m}'$, then

$$\mathbf{m}' = \mathbf{m}_0 + \mathbf{w}(t_{i_1}) + \dots + \mathbf{w}(t_{i_l}) = \mathbf{m}_0 + \sum_{j=1 \dots |T|} \mathbf{w}(t_j) \mathbf{n}(\sigma)[j],$$

i.e., $\mathbf{m}' = \mathbf{m}_0 + W\mathbf{n}(\sigma)$.

- \mathbf{m} is reachable from \mathbf{m}_0 only if $W\mathbf{n} = (\mathbf{m} - \mathbf{m}_0)$ has a solution $\mathbf{n} \in \mathbb{N}^{|T|}$.
- This is a **necessary condition** but not sufficient.
 - A solution \mathbf{n} is *realisable* iff, in addition, $\mathbf{n} = \mathbf{n}(\sigma)$ for some valid firing sequence σ .

The State Equation & Invariance

- $\mathbf{y} \in \mathbb{N}^{|P|}$ is a P-invariant iff it is a solution to $\mathbf{y}^T W = \mathbf{0}$.
 - $\mathbf{y}^T \mathbf{m} = \mathbf{y}^T \mathbf{m}_0$ for any \mathbf{m} reachable from \mathbf{m}_0 .
- $\mathbf{x} \in \mathbb{N}^{|T|}$ is a T-invariant iff it is a solution to $W \mathbf{x} = \mathbf{0}$.
 - $\mathbf{m}[\sigma) \mathbf{m}$ whenever $\mathbf{n}(\sigma) = \mathbf{x}$ and σ enabled at \mathbf{m} .
- Any (positive) linear combination of P-/T-invariants is a P-/T-invariant.
- The *reverse dual* of a net N is obtained by swapping places for transitions and vice versa, and reversing all arcs.
 - The incidence matrix of the reverse dual is the transpose of the incidence matrix of N .
 - A P-(T-)invariant of N is a T-(P-)invariant of the reverse dual.

Introduction

Basics

Complexity &
Expressivity

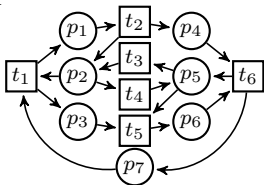
Analysis
Techniques

Special
Classes of
Nets

Conclusion

Example: P-Invariants

$$\begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}^T \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}^T$$



$$\begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 2 \end{pmatrix}^T \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}^T$$

Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

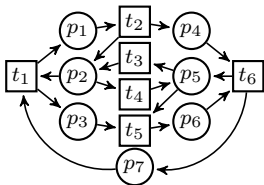
Special
Classes of
Nets

Conclusion

Example: T-Invariants

$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$



Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Minimal Invariants

- The *support* of a P-/T-invariant \mathbf{y} is the set $\{i \mid \mathbf{y}[i] > 0\}$. An invariant has *minimal support* iff no invariants support is a strict subset.
- The number of minimal support P-/T-invariants of a net is finite, but may be exponential.
- All P-/T-invariants are (positive) linear combinations of minimal support P-/T-invariants.
- A P-/T-invariant \mathbf{y} is *minimal* iff no $\mathbf{y}' < \mathbf{y}$ is invariant.
 - A minimal invariant need not have minimal support.
 - For each minimal support, there is a unique minimal invariant.
- Algorithms exist to generate all minimal support P-/T-invariants of a net.

Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

The Fourier-Motzkin Algorithm for P-Invariants

- 1 Initialise $B = [W : I_n]$ ($n = |P|$).
- 2 For $j = 1, \dots, |T|$
 - 1 Append to B all rows resulting from positive linear combinations of pairs of rows in B that eliminate column j .
 - 2 Remove from B all rows with non-zero j th element.
- 3 $B = [\mathbf{0} : D]$, where the rows of D are P-invariants.

Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Example

$$B = \left(\begin{array}{cccccc|cccccccc} 1 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array} \right)$$

Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Example

$$B = \left(\begin{array}{cccccc|cccccc} 1 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \\ 0 & 0 & 1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & -1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{array} \right)$$

Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Example

$$B = \left(\begin{array}{cccccc|cccccc} 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & -1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{array} \right)$$

Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Example

$$B = \left(\begin{array}{cccc|cccc} 0 & 0 & -1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & -1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2 & 2 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{array} \right)$$

Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Example

$$B = \left(\begin{array}{cccc|ccc} 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2 & 2 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 2 & 1 \end{array} \right)$$

Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

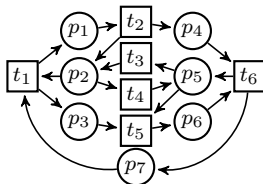
Conclusion

Example

$$B = \left(\begin{array}{cccccc|cccc} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 2 & 1 \end{array} \right)$$

P-invariants:

- $\mathbf{z}_1 = (1001001)$
- $\mathbf{z}_2 = (0010011)$
- $\mathbf{z}_3 = (1100110)$
- $\mathbf{z}_4 = (1110121)$



Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

The State Equation & Structural Properties

- N is structurally bounded iff $\mathbf{y}^T W \leq \mathbf{0}$ has a solution $\mathbf{y} \in \mathbb{N}^{|P|}$ such that $\mathbf{y}[i] \geq 1$ for $i = 1, \dots, |P|$.
 - \mathbf{y} is a linear combination of *all* place markings that is invariant or decreasing under any transition firing.
- N is repetitive w.r.t. transition t iff $W\mathbf{x} \geq \mathbf{0}$ has a solution $\mathbf{x} \in \mathbb{N}^{|T|}$ such that $\mathbf{x}[t] > 0$.
 - \mathbf{x} is a multiset of transitions, including t at least once, whose combined effect is zero or increasing.
 - Can always find some initial marking \mathbf{m}_0 from which \mathbf{x} is realisable.

Introduction

Basics

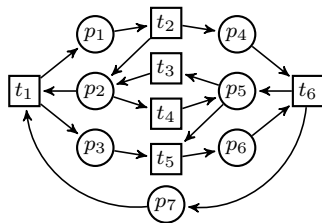
Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Example



- $\mathbf{z}_1 = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$ and $\mathbf{z}_4 = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 2 & 1 \end{pmatrix}$ are P-invariants of the net.
- $\mathbf{y} = \mathbf{z}_1 + \mathbf{z}_4 = \begin{pmatrix} 2 & 1 & 1 & 1 & 1 & 2 & 2 \end{pmatrix}$ is also a P-invariant.
- $\mathbf{y}^T W = \mathbf{0}$ and $\mathbf{y} \geq \mathbf{1}$: The net is **structurally bounded**.

Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Reachability

- Decidability of the (exact) reachability problem for general Petri nets was open for some time.
 - Algorithm proposed by Sacerdote & Tenney in 1977 incorrect (or gaps in correctness proof).
 - Correct algorithm by Mayr in 1981.
 - Simpler correctness proof (for essentially the same algorithm) by Kosaraju in 1982.
- Other algorithms have been presented since.
- All existing algorithms have unbounded complexity.
 - Fun fact: A 2-EXP algorithm was proposed in 1998, but later shown to be incorrect.

Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Reachability: Preliminaries

- \mathbf{m} is *semi-reachable* from \mathbf{m}_0 iff there is a transition sequence $s = t_{i_1}, \dots, t_{i_n}$ such that $\mathbf{m} = \mathbf{m}_0 + \mathbf{w}(t_{i_1}) + \dots + \mathbf{w}(t_{i_n})$.
- s does not have to be valid (firable) at \mathbf{m}_0 .
- \mathbf{m} is semi-reachable from \mathbf{m}_0 iff $W\mathbf{n} = (\mathbf{m} - \mathbf{m}_0)$ has a solution $\mathbf{n} \in \mathbb{N}^{|T|}$.
- If \mathbf{m} is semi-reachable from \mathbf{m}_0 , then $\mathbf{m} + \mathbf{a}$ is reachable from $\mathbf{m}_0 + \mathbf{a}$ for some sufficiently large $\mathbf{a} \geq 0$.

Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

- A *controlled net* is a pair of a marked net $(N = \langle P, T, F \rangle, \mathbf{m}_0)$ and an NFA (A, q_0) over alphabet T .
 - A defines a (regular) subset of (not necessarily firable) transition sequences.
 - Define reachability/coverability/boundedness for (N, \mathbf{m}_0) w.r.t. A in the obvious way.
 - The coverability tree construction is easily modified to consider only sequences accepted by A .
- The *reverse* of N , N_{Rev} (w.r.t. A) is obtained by reversing the flow relation (and arcs in A).
 - $W(N_{\text{Rev}}) = -W(N)$.

Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Reachability: A Sufficient Condition

- In (N, \mathbf{m}_0) w.r.t. (A, q_0) , if
 - (a) (\mathbf{m}_*, q_*) is semi-reachable from (\mathbf{m}_0, q_0) ,
 - (b) $(\mathbf{m}_0 + \mathbf{a}, q_0)$ is reachable from (\mathbf{m}_0, q_0) , for $\mathbf{a} \geq 1$,
 - (c) $(\mathbf{m}_* + \mathbf{b}, q_*)$ is reachable from (\mathbf{m}_*, q_*) in N_{Rev} w.r.t. A , for $\mathbf{b} \geq 1$,
 - (d) $(\mathbf{b} - \mathbf{a}, q_*)$ is semi-reachable from $(\mathbf{0}, q_*)$,
then (\mathbf{m}_*, q_*) is reachable from (\mathbf{m}_0, q_0) .
- The conditions above are effectively checkable:
 - (b) & (c) by coverability tree construction,
 - (a) & (d) through the state equation.

Introduction

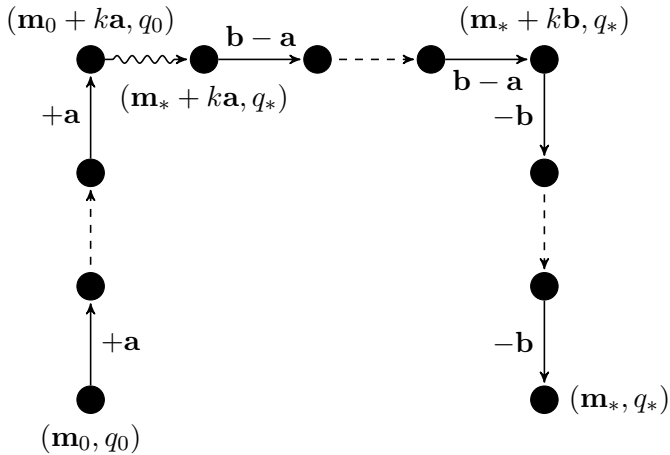
Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion



Introduction

Basics

Complexity &
Expressivity

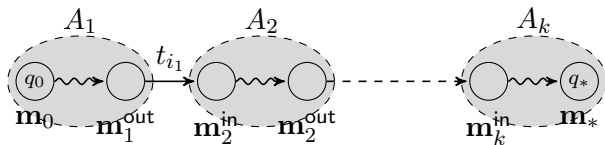
Analysis
Techniques

Special
Classes of
Nets

Conclusion

Reachability: The Mayr/Kosaraju Algorithm

Consider a controlled net (N, A) of the form,



with constraints $\mathbf{m}_i^{\text{in/out}}[j] = x_{i,j}^{\text{i/o}}$ or $\mathbf{m}_i^{\text{in/out}}[j] \geq y_{i,j}^{\text{i/o}} \geq 0$.

- If the sufficient reachability condition holds for each $(\mathbf{m}_i^{\text{in}}, q_i^{\text{in}})$ and $(\mathbf{m}_i^{\text{out}}, q_i^{\text{out}})$ w.r.t A_i , then (\mathbf{m}_*, q_*) is reachable from (\mathbf{m}_0, q_0) .
- Let $\Delta(A_i) = \{\mathbf{m} \mid \mathbf{m} = W\mathbf{n}(s), s \in L(A_i)\}$.
- Let $\Gamma = \{\mathbf{m}_i^{\text{in}}, \mathbf{m}_i^{\text{out}}, \mathbf{n}_i \mid \mathbf{m}_{i+1}^{\text{in}} - \mathbf{m}_i^{\text{out}} = w(t_{i_i}), \mathbf{m}_i^{\text{out}} - \mathbf{m}_i^{\text{in}} \in \Delta(A_i), \text{ and constraints hold}\}$.
- If $(\mathbf{m}_0, q_0) [s] (\mathbf{m}_*, q_*)$, s defines an element in Γ .

Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

- Γ is a *semi-linear set*: consistency (non-emptiness) is decidable via Pressburger arithmetic.
- If Γ is consistent, but the sufficient condition does not hold in some A_i , then A_i can be replaced by a new “chain” of controllers, $A_i^1, \dots, A_i^{l_i}$, each of which is “simpler”:
 - more equality constraints ($\mathbf{m}_{i^l}^{\text{in/out}} = x_{i^l, j}^{\text{i/o}}$), or
 - same equality constraints and smaller automaton.
- There can be several possible replacements (non-deterministic choice).
- If (\mathbf{m}_*, q_*) is not reachable from (\mathbf{m}_0, q_0) , every choice (branch) eventually leads to an inconsistent system.

Special Classes of Nets

Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

**Special
Classes of
Nets**

Conclusion

Special Classes of Nets

- **State Machines:**

- every transition has one incoming and one outgoing arc
i.e. $|\bullet t| = |t\bullet| = 1$ for each $t \in T$.

- **Marked Graphs:**

- every place has one incoming arc, and one outgoing arc
i.e. $|\bullet p| = |p\bullet| = 1$ for each $p \in P$.

- **Free-choice Nets:**

- every arc is either the only arc going from the place, or only arc going to the transition
i.e. $|p\bullet| \leq 1$ or $\bullet(p\bullet) = \{p\}$ for each $p \in P$.

Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Marked Graphs

- An ordinary Petri net with $|\bullet p| = |p\bullet| = 1$ for each place p is a *T-graph*, or *marked graph*.
- Abstracting away places leaves a directed graph:
 - Called the *underlying graph* (usually denoted G).
 - A marking of the net is a marking of the edges of G .
- Marked graphs model “decision-free” concurrent systems.
- Several properties of marked graphs are decidable in polynomial time:
 - Structural liveness and boundedness.
 - Liveness and boundedness for a given initial marking.
- Simple condition for realisability (and thus reachability).

Introduction

Basics

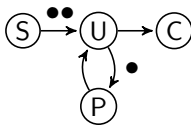
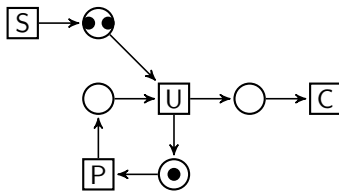
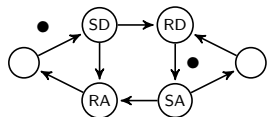
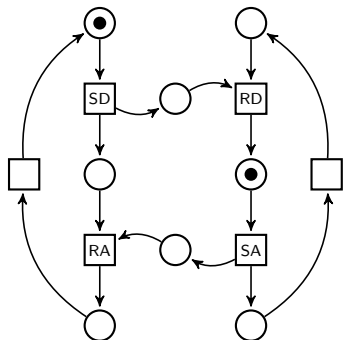
Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Example: Marked and Underlying Graphs



Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Some Properties of Marked Graphs

- **Theorem:** The total number of tokens on every directed circuit in the underlying graph is invariant.
- **Theorem:** The maximum number of tokens an edge $a \rightarrow b$ in (G, \mathbf{m}_0) can ever have is equal to the minimum number of tokens \mathbf{m}_0 places on any directed circuit that contains this edge.
- **Theorem:** A marked graph (G, \mathbf{m}_0) is live iff \mathbf{m}_0 places at least one token on every directed circuit of G .
- **Theorem:** A live marked graph (G, \mathbf{m}_0) is k -bounded iff every place (edge in G) belongs to a directed circuit and \mathbf{m}_0 places at most k tokens on every directed circuit of G .
- **Theorem:** A marked graph net has a live and bounded marking iff G is strongly connected.

Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Free Choice Nets

- An ordinary Petri net such that $|p^\bullet| \leq 1$ or ${}^\bullet(p^\bullet) = \{p\}$ for each place p , is a *free choice* net.
- Equivalently: If $p^\bullet \cap p'^\bullet \neq \emptyset$ then $|p^\bullet| = |p'^\bullet| = 1$, for all $p, p' \in P$.
- Extended free choice net: If $p^\bullet \cap p'^\bullet \neq \emptyset$ then $p^\bullet = p'^\bullet$, for all $p, p' \in P$.
- An extended free choice net can be transformed to a basic free choice net, adding at most a linear number of places and transitions.
- Note: Marked graphs and state machines are also free choice nets.
- A fundamental property of free choice nets: if ${}^\bullet t \cap {}^\bullet t' \neq \emptyset$ then whenever t is enabled, so is t' .

Introduction

Basics

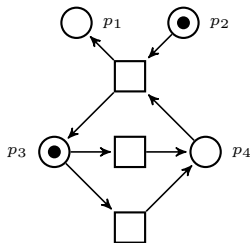
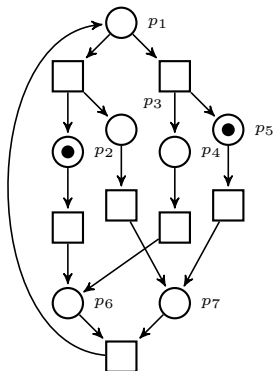
Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Example: Free Choice Nets



Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Decomposition of Free Choice Nets

- A subnet of $N = (P, T, W)$ is a net $N' = (P', T', W')$ with $P' \subseteq P$, $T' \subseteq T$ and $W' = W|_{(P' \cup T')}$.
- The P-subnet induced by $S \subseteq P$ is $(S, \bullet S \cup S \bullet, W')$
 - That is, the subnet consisting of S and all transition incident on a place in S .
- The T-subnet induced by $U \subseteq T$ is $(\bullet U \cup U \bullet, U, W')$
 - That is, the subnet consisting of U and all places incident on a transition in U .
- A *P-component* is a strongly connected P-subnet such that $|\bullet t|, |t \bullet| \leq 1$, for all t .
 - A P-component is a state machine.
- A *T-component* is a strongly connected T-subnet such that $|\bullet p|, |p \bullet| \leq 1$, for all p
 - A T-component is a marked graph.

Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Decomposition of Free Choice Nets

- **Theorem:** A live free choice net (N, \mathbf{m}_0) is 1-bounded (safe) iff it is covered by P-components, each of which has a single token at \mathbf{m}_0 .
- **Theorem:** A live and safe free choice net (N, \mathbf{m}_0) is covered by T-components, and for each T-component, N' , there is a reachable marking \mathbf{m} such that $(N', \mathbf{m}|_{N'})$ is live and safe.

Introduction

Basics

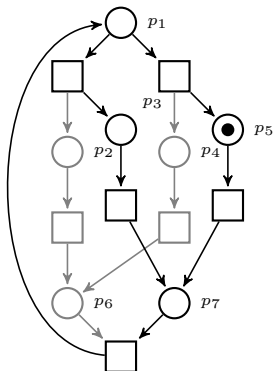
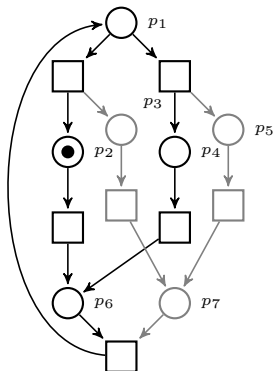
Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Example: Decomposition into State Machines



Introduction

Basics

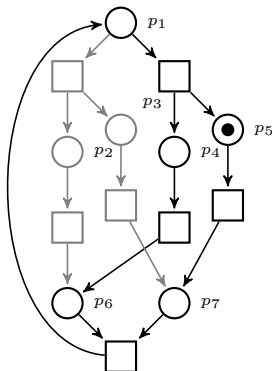
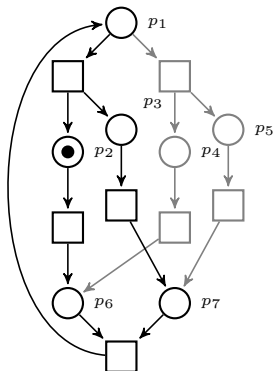
Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Example: Decomposition into Marked Graphs



Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Siphons and Traps

- A *siphon* is a subset S of places such that $\bullet S \subseteq S^\bullet$.
 - Every transition that outputs a token to a place in S also consumes a token from a place in S .
 - If \mathbf{m} places no token in S , no marking reachable from \mathbf{m} does either.
- A *trap* is a subset S of places such that $S^\bullet \subseteq \bullet S$.
 - Every transition that consumes a token from a place in S also outputs a token to a place in S .
 - If \mathbf{m} places at least one token in S , so does every marking reachable from \mathbf{m} .
- **Theorem:** A free choice net (N, \mathbf{m}_0) is live iff every siphon contains a marked trap.

Introduction

Basics

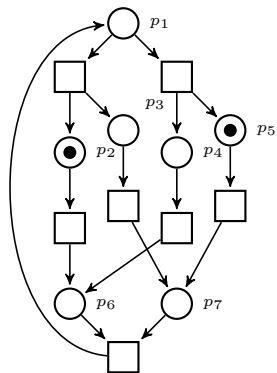
Complexity &
Expressivity

Analysis
Techniques

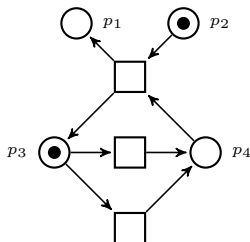
Special
Classes of
Nets

Conclusion

Example: Siphons and Traps



- The only siphon is P .
- P is also a trap.



- $\{p_2\}$ and $\{p_3, p_4\}$ are siphons.
- $\{p_1\}$ and $\{p_3, p_4\}$ are traps.

Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Some Complexity Results for Free Choice Nets

- Liveness for marked free choice nets is decidable in polynomial time.
- Boundedness of *live* free choice nets is decidable in polynomial time.
- A number of properties of *live and bounded* free choice nets are decidable in polynomial time, e.g.,
 - Transition executability and repeated executability.
 - The “home state” property (markings that can always be re-reached).
- Reachability in free choice nets is NP-hard.

Introduction

Basics


Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Characterisation by Derivation Rules

- Initial net: 
- Rule #1: Add a new place p' with $\mathbf{r}(p') = \sum_{p \in P} \lambda_p \mathbf{r}(p)$ and $|p'^{\bullet}| = 1$.
- Rule #2: Replace place p with a connected P-graph N' , and connect each input and output of p to at least one place in N' .
 - Must have $|\bullet p| > 1$ and $|p\bullet| > 1$, except for initial net.
 - Every place $p' \in N'$ must appear on a path in the resulting net that enters and leaves N' .
- **Theorem:** The class of nets obtained by applying the above rules to the initial net is exactly the class of structurally live and structurally bounded free choice nets.

Introduction

Basics

Complexity &
Expressivity

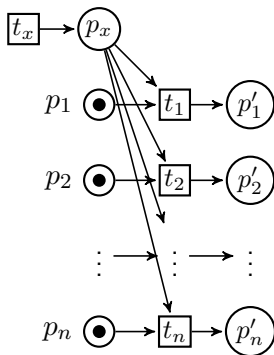
Analysis
Techniques

Special
Classes of
Nets

Conclusion

Reachability: Acyclic Nets

- Recall: $\mathbf{m}_0 [s] \mathbf{m}$ implies $\exists \mathbf{n} \in \mathbb{N}^{|T|} : W\mathbf{n} = (\mathbf{m} - \mathbf{m}_0)$.
- A solution \mathbf{n} is *realisable* iff $\mathbf{n} = \mathbf{n}(s)$ for some valid firing sequence s .
- Theorem:** For an acyclic net, every solution to $W\mathbf{n} = (\mathbf{m} - \mathbf{m}_0)$ is realisable.
- Reachability in acyclic nets is NP-hard.



Introduction

Basics

Complexity &
Expressivity

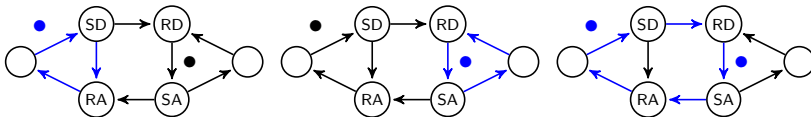
Analysis
Techniques

Special
Classes of
Nets

Conclusion

Reachability: Marked Graphs

- **Theorem:** In a live marked graph, \mathbf{m} is reachable from \mathbf{m}_0 iff \mathbf{m}_0 and \mathbf{m} place the same total number of tokens on every fundamental circuit of the underlying graph.
- A fundamental circuit is obtained by adding one edge to a spanning tree.
- The directed fundamental circuits of a marked graph are a full set of linearly independent P-invariants.



Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Conclusions

Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

Summary & Conclusions

- Petri nets: Intuitive, graphical modelling formalism, closely related to planning.
- Petri net theory offers a different set of tools:
 - Algebraic methods (based on the state equation).
 - Characterisation and study of classes of nets with special structure.
- Planning also has tools potentially applicable to Petri nets.

Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion

The Many Things We Haven't Talked About

- Extensions of basic Place-Transition nets:
 - Read arcs, reset arcs and inhibitor arcs.
 - Colored Petri nets, timed nets, stochastic nets, etc.
- Other properties of Petri nets (and related decision problems):
 - Model checking (tense logics, process calculi).
 - Language (trace) properties.
- Heaps more results concerning different Petri net subclasses.

Introduction

Basics

Complexity &
Expressivity

Analysis
Techniques

Special
Classes of
Nets

Conclusion