Comparison of the Expressiveness of Timed Automata and Time Petri Nets

B. Bérard[1], F. Cassez[2], S. Haddad[1], D. Lime[3], O.H. Roux[2]

[1]LAMSADE, Paris
France

[2]IRCCyN, Nantes
France

[3]CISS, Aalborg
Denmark

FORMATS'05

Uppsala, Sweden

Outline of the talk

- Context & Motivation

Outline of the talk

- Context & Motivation
- Timed Automata & Time Petri Nets

Outline of the talk

- Context & Motivation
- Timed Automata & Time Petri Nets
- Expressiveness wrt Timed Bisimilarity

Outline of the talk

- Context & Motivation
- Timed Automata & Time Petri Nets
- Expressiveness wrt Timed Bisimilarity
- Expressiveness wrt Timed Language Acceptance

Outline of the talk

- Context & Motivation
- Timed Automata & Time Petri Nets
- Expressiveness wrt Timed Bisimilarity
- Expressiveness wrt Timed Language Acceptance
- Conclusion

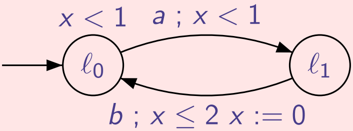
Outline

- Context & Motivation
- Timed Automata & Time Petri Nets
- Expressiveness wrt Timed Bisimilarity
- Expressiveness wrt Timed Language Acceptance
- Conclusion

Timed Automata & Time Petri Nets

Timed Automata [Alur & Dill'94]

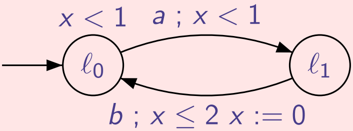
Timed Automata = Finite Automata + timing constraints given by clocks



Timed Automata & Time Petri Nets

Timed Automata [Alur & Dill'94]

Timed Automata = Finite Automata + timing constraints given by clocks

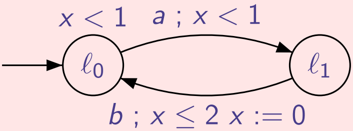


A run: $(\ell_0, 0) \xrightarrow{0.78} (\ell_0, 0.78) \xrightarrow{a} (\ell_1, 0.78) \xrightarrow{1.22} (\ell_1, 2) \xrightarrow{b} (\ell_0, 0) \cdots$

Timed Automata & Time Petri Nets

Timed Automata [Alur & Dill'94]

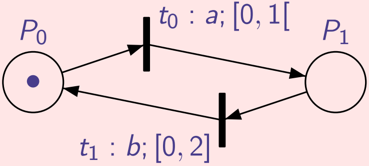
Timed Automata = Finite Automata + timing constraints given by clocks



A run: $(\ell_0, 0) \xrightarrow{0.78} (\ell_0, 0.78) \xrightarrow{a} (\ell_1, 0.78) \xrightarrow{1.22} (\ell_1, 2) \xrightarrow{b} (\ell_0, 0) \cdots$

Time Petri Nets [Merlin'74]

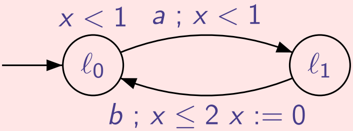
Time Petri Nets = Petri Nets + firing interval relative to markings



Timed Automata & Time Petri Nets

Timed Automata [Alur & Dill'94]

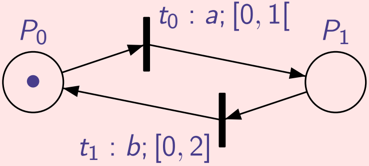
Timed Automata = Finite Automata + timing constraints given by clocks



A run: $(\ell_0, 0) \xrightarrow{0.78} (\ell_0, 0.78) \xrightarrow{a} (\ell_1, 0.78) \xrightarrow{1.22} (\ell_1, 2) \xrightarrow{b} (\ell_0, 0) \cdots$

Time Petri Nets [Merlin'74]

Time Petri Nets = Petri Nets + firing interval relative to markings



A run: $(P_0, 0) \xrightarrow{0.78} (P_0, 0.78) \xrightarrow{a} (P_1, 0) \xrightarrow{2} (P_1, 2) \xrightarrow{b} (P_0, 0) \cdots$

Fundamental Problems for Timed Automata

We consider (Bounded) TPN introduced by [Merlin'74]

Problem	Timed Automata	B-Time Petri Nets
Reachability (RP) Emptyness (EP)		
Universality (UP) Language Inclusion		
Closure Properties		
Effect of ε-transition		
TCTL model checking		

Fundamental Problems for Timed Automata

We consider (Bounded) TPN introduced by [Merlin'74]

Problem	Timed Automata	B-Time Petri Nets
Reachability (RP) Emptyness (EP)	PSPACE-Complete [Alur & Dill'94]	
Universality (UP) Language Inclusion		
Closure Properties		
Effect of ε-transition		
TCTL model checking		

Fundamental Problems for Timed Automata

We consider (Bounded) TPN introduced by [Merlin'74]

Problem	Timed Automata	B-Time Petri Nets
Reachability (RP) Emptyness (EP)	PSPACE-Complete [Alur & Dill'94]	
Universality (UP) Language Inclusion	Undecidable [Alur & Dill'94]	
Closure Properties		
Effect of ε-transition		
TCTL model checking		

Fundamental Problems for Timed Automata

We consider (Bounded) TPN introduced by [Merlin'74]

Problem	Timed Automata	B-Time Petri Nets
Reachability (RP) Emptyness (EP)	PSPACE-Complete [Alur & Dill'94]	
Universality (UP) Language Inclusion	Undecidable [Alur & Dill'94]	
Closure Properties	Closed under $\cap, \cup$ but not under compl. [Alur & Dill'94]	
Effect of ε-transition		
TCTL model checking		

Fundamental Problems for Timed Automata

We consider (Bounded) TPN introduced by [Merlin'74]

Problem	Timed Automata	B-Time Petri Nets
Reachability (RP) Emptyness (EP)	PSPACE-Complete [Alur & Dill'94]	
Universality (UP) Language Inclusion	Undecidable [Alur & Dill'94]	
Closure Properties	Closed under $\cap, \cup$ but not under compl. [Alur & Dill'94]	
Effect of ε-transition	ε-TA $>$ TA [Bérard et al.'96]	
TCTL model checking		

Fundamental Problems for Timed Automata

We consider (Bounded) TPN introduced by [Merlin'74]

Problem	Timed Automata	B-Time Petri Nets
Reachability (RP) Emptyness (EP)	PSPACE-Complete [Alur & Dill'94]	
Universality (UP) Language Inclusion	Undecidable [Alur & Dill'94]	
Closure Properties	Closed under $\cap, \cup$ but not under compl. [Alur & Dill'94]	
Effect of ε-transition	ε-TA $>$ TA [Bérard et al.'96]	
TCTL model checking	Decidable [Alur & Dill'94]	

Fundamental Problems for Timed Automata

We consider (Bounded) TPN introduced by [Merlin'74]

Problem	Timed Automata	B-Time Petri Nets
Reachability (RP) Emptyness (EP)	PSPACE-Complete [Alur & Dill'94]	Decidable [Berthomieu & Diaz'91]
Universality (UP) Language Inclusion	Undecidable [Alur & Dill'94]	
Closure Properties	Closed under $\cap, \cup$ but not under compl. [Alur & Dill'94]	
Effect of ε-transition	ε-TA > TA [Bérard et al.'96]	
TCTL model checking	Decidable [Alur & Dill'94]	

Fundamental Problems for Timed Automata

We consider (Bounded) TPN introduced by [Merlin'74]

Problem	Timed Automata	B-Time Petri Nets
Reachability (RP) Emptyness (EP)	PSPACE-Complete [Alur & Dill'94]	Decidable [Berthomieu & Diaz'91]
Universality (UP) Language Inclusion	Undecidable [Alur & Dill'94]	??
Closure Properties	Closed under $\cap, \cup$ but not under compl. [Alur & Dill'94]	
Effect of ε-transition	ε-TA $>$ TA [Bérard et al.'96]	
TCTL model checking	Decidable [Alur & Dill'94]	

Fundamental Problems for Timed Automata

We consider (Bounded) TPN introduced by [Merlin'74]

Problem	Timed Automata	B-Time Petri Nets
Reachability (RP) Emptyness (EP)	PSPACE-Complete [Alur & Dill'94]	Decidable [Berthomieu & Diaz'91]
Universality (UP) Language Inclusion	Undecidable [Alur & Dill'94]	??
Closure Properties	Closed under $\cap, \cup$ but not under compl. [Alur & Dill'94]	??
Effect of ε-transition	ε-TA $>$ TA [Bérard et al.'96]	
TCTL model checking	Decidable [Alur & Dill'94]	

Fundamental Problems for Timed Automata

We consider (Bounded) TPN introduced by [Merlin'74]

Problem	Timed Automata	B-Time Petri Nets
Reachability (RP) Emptyness (EP)	PSPACE-Complete [Alur & Dill'94]	Decidable [Berthomieu & Diaz'91]
Universality (UP) Language Inclusion	Undecidable [Alur & Dill'94]	??
Closure Properties	Closed under $\cap, \cup$ but not under compl. [Alur & Dill'94]	??
Effect of ε-transition	ε-TA $>$ TA [Bérard et al.'96]	??
TCTL model checking	Decidable [Alur & Dill'94]	

Fundamental Problems for Timed Automata

We consider (Bounded) TPN introduced by [Merlin'74]

Problem	Timed Automata	B-Time Petri Nets
Reachability (RP) Emptyness (EP)	PSPACE-Complete [Alur & Dill'94]	Decidable [Berthomieu & Diaz'91]
Universality (UP) Language Inclusion	Undecidable [Alur & Dill'94]	??
Closure Properties	Closed under $\cap, \cup$ but not under compl. [Alur & Dill'94]	??
Effect of ε-transition	ε-TA $>$ TA [Bérard et al.'96]	??
TCTL model checking	Decidable [Alur & Dill'94]	Decidable [C. & R., AVoCS'04]

Motivation For Studying the Problems for TPNs

- Universality Problem
  checking a TPN against a spec. given by a TPN $\mathcal{L}(A) \subseteq \mathcal{L}(B)$
  if undecidable then Language Inclusion Problem undecidable

Motivation For Studying the Problems for TPNs

- Universality Problem
  checking a TPN against a spec. given by a TPN $\mathcal{L}(A) \subseteq \mathcal{L}(B)$
  if undecidable then Language Inclusion Problem undecidable
- Closure Properties
  TA not closed under complement
  find subclasses of TA enjoying nice closure properties
  TPN is such a class ?

Motivation For Studying the Problems for TPNs

- Universality Problem
  checking a TPN against a spec. given by a TPN $\mathcal{L}(A) \subseteq \mathcal{L}(B)$
  if undecidable then Language Inclusion Problem undecidable
- Closure Properties
  TA not closed under complement
  find subclasses of TA enjoying nice closure properties
  TPN is such a class ?
- Time Petri Nets Properties
  e.g. every Bounded-PN is equivalent to a one-safe PN
  What about TPN ?

Motivation For Studying the Problems for TPNs

- Universality Problem
  checking a TPN against a spec. given by a TPN $\mathcal{L}(A) \subseteq \mathcal{L}(B)$
  if undecidable then Language Inclusion Problem undecidable
- Closure Properties
  TA not closed under complement
  find subclasses of TA enjoying nice closure properties
  TPN is such a class ?
- Time Petri Nets Properties
  e.g. every Bounded-PN is equivalent to a one-safe PN
  What about TPN ?
- TA or TPN as a specification language ?
  precise comparison of expressive power

Our Contribution

- Extended version of (bounded) TPNs: TPN_ε
  open intervals, final and repeated markings, ε-transitions

Our Contribution

- Extended version of (bounded) TPNs: TPN_ε
  open intervals, final and repeated markings, ε-transitions
- TA are strictly more expressive than TPN_ε w.r.t. weak timed bisimilarity

Our Contribution

- Extended version of (bounded) TPNs: TPN_ε
  open intervals, final and repeated markings, ε-transitions
- TA are strictly more expressive than TPN_ε w.r.t. weak timed bisimilarity
- TA_ε and one-safe TPN_ε are equally expressive w.r.t. timed language acceptance

Our Contribution

- Extended version of (bounded) TPNs: TPN_ε
  open intervals, final and repeated markings, ε-transitions
- TA are strictly more expressive than TPN_ε w.r.t. weak timed bisimilarity
- TA_ε and one-safe TPN_ε are equally expressive w.r.t. timed language acceptance
  - Universal Problem is undecidable for (bounded and) one-safe TPN_ε (Language Inclusion is undecidable)

Our Contribution

- Extended version of (bounded) TPNs: TPN_ε
  open intervals, final and repeated markings, ε-transitions
- TA are strictly more expressive than TPN_ε w.r.t. weak timed bisimilarity
- TA_ε and one-safe TPN_ε are equally expressive w.r.t. timed language acceptance
  - Universal Problem is undecidable for (bounded and) one-safe TPN_ε (Language Inclusion is undecidable)
  - Bounded TPN_ε and one-safe TPN_ε are equally expressive (w.r.t. timed language acceptance)

Our Contribution

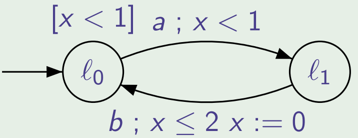
- Extended version of (bounded) TPNs: TPN_ε
  open intervals, final and repeated markings, ε-transitions
- TA are strictly more expressive than TPN_ε w.r.t. weak timed bisimilarity
- TA_ε and one-safe TPN_ε are equally expressive w.r.t. timed language acceptance
  - Universal Problem is undecidable for (bounded and) one-safe TPN_ε (Language Inclusion is undecidable)
  - Bounded TPN_ε and one-safe TPN_ε are equally expressive (w.r.t. timed language acceptance)
- Timed Bisimilarity: B-$TPN_\varepsilon(\leq, \geq)$ (original class defined by Merlin) and $TA_\varepsilon(\leq, \geq)$ are equivalent w.r.t. timed bisimilarity

Outline

Semantics of Timed Automata

Timed Automata

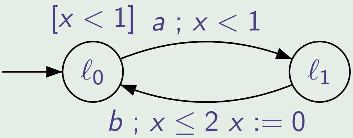
▸ Timed Automata



Semantics of Timed Automata

Timed Automata

▸ Timed Automata

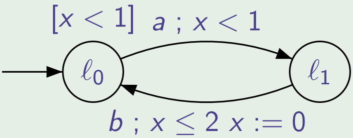


run 1: $(\ell_0, 0) \xrightarrow{0.78} (\ell_0, 0.78) \xrightarrow{a} (\ell_1, 0.78) \xrightarrow{1.22} (\ell_1, 2) \xrightarrow{b} (\ell_0, 0) \cdots$

Semantics of Timed Automata

Timed Automata

▸ Timed Automata



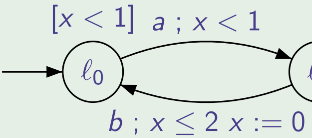
run 1: $(\ell_0, 0) \xrightarrow{0.78} (\ell_0, 0.78) \xrightarrow{a} (\ell_1, 0.78) \xrightarrow{1.22} (\ell_1, 2) \xrightarrow{b} (\ell_0, 0) \cdots$

run 2: $(\ell_0, 0) \xrightarrow{0.78} (\ell_0, 0.78) \xrightarrow{a} (\ell_1, 0.78) \xrightarrow{50} (\ell_1, 50.78) \cdots$

Semantics of Timed Automata

Timed Automata

▸ Timed Automata



run 1: $(\ell_0, 0) \xrightarrow{0.78} (\ell_0, 0.78) \xrightarrow{a} (\ell_1, 0.78) \xrightarrow{1.22} (\ell_1, 2) \xrightarrow{b} (\ell_0, 0) \cdots$

run 2: $(\ell_0, 0) \xrightarrow{0.78} (\ell_0, 0.78) \xrightarrow{a} (\ell_1, 0.78) \xrightarrow{50} (\ell_1, 50.78) \cdots$

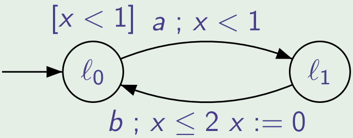
Definition (Semantics of TA)

- States: $(\ell, v) \in Q = L \times (\mathbb{R}_{\geq 0})^X$

Semantics of Timed Automata

Timed Automata

▸ Timed Automata



run 1: $(\ell_0, 0) \xrightarrow{0.78} (\ell_0, 0.78) \xrightarrow{a} (\ell_1, 0.78) \xrightarrow{1.22} (\ell_1, 2) \xrightarrow{b} (\ell_0, 0) \cdots$

run 2: $(\ell_0, 0) \xrightarrow{0.78} (\ell_0, 0.78) \xrightarrow{a} (\ell_1, 0.78) \xrightarrow{50} (\ell_1, 50.78) \cdots$

Definition (Semantics of TA)

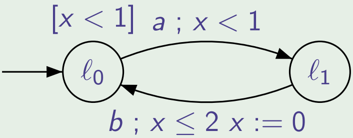
- States: $(\ell, v) \in Q = L \times (\mathbb{R}_{\geq 0})^X$
- Discrete transition: $(l, v) \xrightarrow{a} (l', v')$
  iff there is a transition (l, g, a, R, l') in $\mathcal{A}$ s.t.

$$\begin{cases} \textit{the guard is true in } (l, v) \\ v' \textit{ is } v \textit{ with the clocks in } R \textit{ equal to zero} \\ \textit{The invariant of } l' \textit{ holds for } v' \end{cases}$$

Semantics of Timed Automata

Timed Automata

▸ Timed Automata



run 1: $(\ell_0, 0) \xrightarrow{0.78} (\ell_0, 0.78) \xrightarrow{a} (\ell_1, 0.78) \xrightarrow{1.22} (\ell_1, 2) \xrightarrow{b} (\ell_0, 0) \cdots$

run 2: $(\ell_0, 0) \xrightarrow{0.78} (\ell_0, 0.78) \xrightarrow{a} (\ell_1, 0.78) \xrightarrow{50} (\ell_1, 50.78) \cdots$

Definition (Semantics of TA)

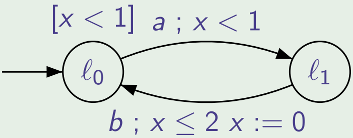
- States: $(\ell, v) \in Q = L \times (\mathbb{R}_{\geq 0})^X$
- Discrete transition: $(l, v) \xrightarrow{a} (l', v')$
- Time transition: $(l, v) \xrightarrow{t} (l', v')$

  iff $\begin{cases} l = l' \text{ and } v' = v + t\text{: the clocks are updated} \\ \text{The invariant of } l \text{ holds for all } v + t', t' \leq t \end{cases}$

Semantics of Timed Automata

Timed Automata

▸ Timed Automata



run 1: $(\ell_0, 0) \xrightarrow{0.78} (\ell_0, 0.78) \xrightarrow{a} (\ell_1, 0.78) \xrightarrow{1.22} (\ell_1, 2) \xrightarrow{b} (\ell_0, 0) \cdots$

run 2: $(\ell_0, 0) \xrightarrow{0.78} (\ell_0, 0.78) \xrightarrow{a} (\ell_1, 0.78) \xrightarrow{50} (\ell_1, 50.78) \cdots$

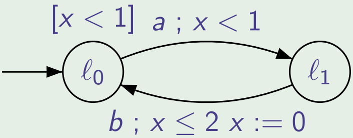
Definition (Semantics of TA)

- States: $(\ell, v) \in Q = L \times (\mathbb{R}_{\geq 0})^X$
- Discrete transition: $(l, v) \xrightarrow{a} (l', v')$
- Time transition: $(l, v) \xrightarrow{t} (l', v')$
- a TA generates a set of runs = alternating discrete and time steps

Semantics of Timed Automata

Timed Automata

▸ Timed Automata



run 1: $(\ell_0, 0) \xrightarrow{0.78} (\ell_0, 0.78) \xrightarrow{a} (\ell_1, 0.78) \xrightarrow{1.22} (\ell_1, 2) \xrightarrow{b} (\ell_0, 0) \cdots$

run 2: $(\ell_0, 0) \xrightarrow{0.78} (\ell_0, 0.78) \xrightarrow{a} (\ell_1, 0.78) \xrightarrow{50} (\ell_1, 50.78) \cdots$

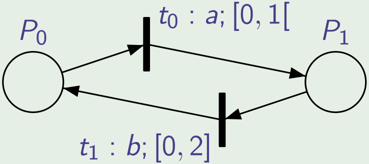
Definition (Semantics of TA)

- States: $(\ell, v) \in Q = L \times (\mathbb{R}_{\geq 0})^X$
- Discrete transition: $(l, v) \xrightarrow{a} (l', v')$
- Time transition: $(l, v) \xrightarrow{t} (l', v')$
- a TA generates a set of runs = alternating discrete and time steps
- semantics of a TA A is a Timed Transition System S_A

Semantics of Time Petri Nets

Time Petri Net

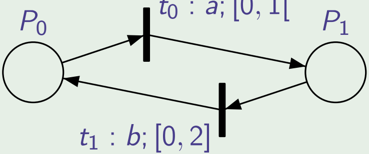
▸ Time Petri Nets



Semantics of Time Petri Nets

Time Petri Net

▸ Time Petri Nets

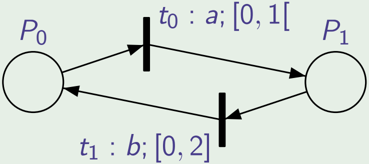


run 1: $(P_0, 0) \xrightarrow{0.78} (P_0, 0.78) \xrightarrow{a} (P_1, 0.78) \xrightarrow{1.5} (P_1, 1.5) \xrightarrow{b} (P_0, 0) \cdots$

Semantics of Time Petri Nets

Time Petri Net

Time Petri Nets



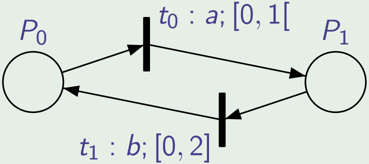
run 1: $(P_0, 0) \xrightarrow{0.78} (P_0, 0.78) \xrightarrow{a} (P_1, 0.78) \xrightarrow{1.5} (P_1, 1.5) \xrightarrow{b} (P_0, 0) \cdots$

run 2: $(P_0, 0) \xrightarrow{0.78} (P_0, 0.78) \xrightarrow{a} (P_1, 0) \xrightarrow{2} (P_0, 0) \cdots$

Semantics of Time Petri Nets

Time Petri Net

▸ Time Petri Nets



run 1: $(P_0, 0) \xrightarrow{0.78} (P_0, 0.78) \xrightarrow{a} (P_1, 0.78) \xrightarrow{1.5} (P_1, 1.5) \xrightarrow{b} (P_0, 0) \cdots$

run 2: $(P_0, 0) \xrightarrow{0.78} (P_0, 0.78) \xrightarrow{a} (P_1, 0) \xrightarrow{2} (P_0, 0) \cdots$

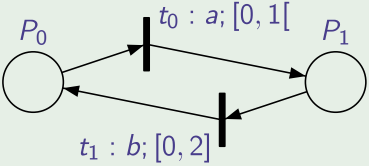
Definition (Semantics of Time Petri Nets)

- States: (M, ν) with M a marking and $\nu : Enabled(M) \mapsto \mathbb{R}_{\geq 0}$

Semantics of Time Petri Nets

Time Petri Net

▸ Time Petri Nets



run 1: $(P_0, 0) \xrightarrow{0.78} (P_0, 0.78) \xrightarrow{a} (P_1, 0.78) \xrightarrow{1.5} (P_1, 1.5) \xrightarrow{b} (P_0, 0) \cdots$

run 2: $(P_0, 0) \xrightarrow{0.78} (P_0, 0.78) \xrightarrow{a} (P_1, 0) \xrightarrow{2} (P_0, 0) \cdots$

Definition (Semantics of Time Petri Nets)

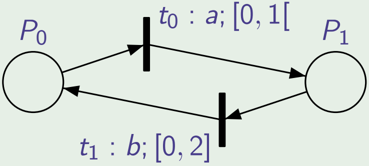
- States: (M, ν) with M a marking and $\nu : Enabled(M) \mapsto \mathbb{R}_{\geq 0}$
- Discrete transition: $(M, \nu) \xrightarrow{a} (M', \nu')$ iff $\exists t : a, I$ in the s.t.

$$\begin{cases} t \text{ is enabled in } M \text{ and } M' = M - {}^\bullet t + t^\bullet \\ \nu(t) \text{ is in } I \text{ (the interval associated with } t) \\ \nu'(t') = 0 \text{ if } t' \text{ is enabled when firing } t, \ \nu'(t') = \nu(t') \text{ otherwise} \end{cases}$$

Semantics of Time Petri Nets

Time Petri Net

▸ Time Petri Nets



run 1: $(P_0, 0) \xrightarrow{0.78} (P_0, 0.78) \xrightarrow{a} (P_1, 0.78) \xrightarrow{1.5} (P_1, 1.5) \xrightarrow{b} (P_0, 0) \cdots$

run 2: $(P_0, 0) \xrightarrow{0.78} (P_0, 0.78) \xrightarrow{a} (P_1, 0) \xrightarrow{2} (P_0, 0) \cdots$

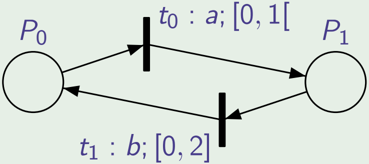
Definition (Semantics of Time Petri Nets)

- States: (M, ν) with M a marking and $\nu : Enabled(M) \mapsto \mathbb{R}_{\geq 0}$
- Discrete transition: $(M, \nu) \xrightarrow{a} (M', \nu')$
- Time transition: $(M, \nu) \xrightarrow{d} (M', \nu')$ iff
$$\begin{cases} M = M' \text{ and } \nu' = \nu + d \text{ (clocks of enabled transitions updated} \\ \text{For all enabled } t\text{, for all } d' \leq d,\ \nu(t) \in I(t) \end{cases}$$

Semantics of Time Petri Nets

Time Petri Net

▸ Time Petri Nets



run 1: $(P_0, 0) \xrightarrow{0.78} (P_0, 0.78) \xrightarrow{a} (P_1, 0.78) \xrightarrow{1.5} (P_1, 1.5) \xrightarrow{b} (P_0, 0) \cdots$

run 2: $(P_0, 0) \xrightarrow{0.78} (P_0, 0.78) \xrightarrow{a} (P_1, 0) \xrightarrow{2} (P_0, 0) \cdots$

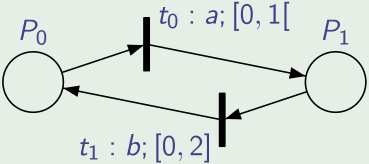
Definition (Semantics of Time Petri Nets)

- States: (M, ν) with M a marking and $\nu : Enabled(M) \mapsto \mathbb{R}_{\geq 0}$
- Discrete transition: $(M, \nu) \xrightarrow{a} (M', \nu')$
- Time transition: $(M, \nu) \xrightarrow{d} (M', \nu')$
- A TPN generates a set of runs = alternating discrete and time steps

Semantics of Time Petri Nets

Time Petri Net

Time Petri Nets



run 1: $(P_0, 0) \xrightarrow{0.78} (P_0, 0.78) \xrightarrow{a} (P_1, 0.78) \xrightarrow{1.5} (P_1, 1.5) \xrightarrow{b} (P_0, 0) \cdots$

run 2: $(P_0, 0) \xrightarrow{0.78} (P_0, 0.78) \xrightarrow{a} (P_1, 0) \xrightarrow{2} (P_0, 0) \cdots$

Definition (Semantics of Time Petri Nets)

- States: (M, ν) with M a marking and $\nu : Enabled(M) \mapsto \mathbb{R}_{\geq 0}$
- Discrete transition: $(M, \nu) \xrightarrow{a} (M', \nu')$
- Time transition: $(M, \nu) \xrightarrow{d} (M', \nu')$
- A TPN generates a set of runs = alternating discrete and time steps
- Semantics of a TPN N = Timed Transition System S_N

Outline

- Context & Motivation
- Timed Automata & Time Petri Nets
- **Expressiveness wrt Timed Bisimilarity**
- Expressiveness wrt Timed Language Acceptance
- Conclusion

Timed Bisimilarity

Definition (Weak Timed Bisimilarity)

Two TTS A and B are timed bisimilar if there is an equivalence relation $\equiv$ on the states of S_A and S_B s.t.: $s_0^A \equiv s_0^B$ and

Timed Bisimilarity

Definition (Weak Timed Bisimilarity)

Two TTS A and B are timed bisimilar if there is an equivalence relation $\equiv$ on the states of S_A and S_B s.t.: $s_0^A \equiv s_0^B$ and

1. for each state s of S_A there is a state q of S_B s.t. $s \equiv q$ and if $s \xrightarrow{\sigma} s'$ then $q \xrightarrow{\sigma} q'$ and $s' \equiv q'$ and s'
2. for each state q of S_B ...

Timed Bisimilarity

Definition (Weak Timed Bisimilarity)

Two TTS A and B are timed bisimilar if there is an equivalence relation $\equiv$ on the states of S_A and S_B s.t.: $s_0^A \equiv s_0^B$ and

1. for each state s of S_A there is a state q of S_B s.t. $s \equiv q$ and if $s \xrightarrow{\sigma} s'$ then $q \xrightarrow{\sigma} q'$ and $s' \equiv q'$ and s'
2. for each state q of S_B ...

Weakly Timed Bisimilar: allows ε-moves

Timed Bisimilarity

Definition (Weak Timed Bisimilarity)

Two TTS A and B are weakly timed bisimilar if there is an equivalence relation $\equiv$ on the states of S_A and S_B s.t.: $s_0^A \equiv s_0^B$ and

1. for each state s of S_A there is a state q of S_B s.t. $s \equiv q$ and if $s \xRightarrow{a} s'$ then $q \xRightarrow{a} q'$ and $s' \equiv q'$ and s'
2. for each state q of S_B ...

Weakly Timed Bisimilar: allows ε-moves

- discrete step: $s \xRightarrow{a} s'$ if $s \xrightarrow{\varepsilon^*}\xrightarrow{a}\xrightarrow{\varepsilon^*} s'$
- time step: $s \xRightarrow{\delta} s'$ if $s \xrightarrow{\varepsilon^*}\xrightarrow{\delta_1}\xrightarrow{\varepsilon^+} \cdots \xrightarrow{\varepsilon^+}\xrightarrow{\delta_n}\xrightarrow{\varepsilon^*} s'$ and $\sum \delta_i = \delta$

Timed Bisimilarity

Definition (Weak Timed Bisimilarity)

Two TTS A and B are weakly timed bisimilar if there is an equivalence relation $\equiv$ on the states of S_A and S_B s.t.: $s_0^A \equiv s_0^B$ and

1. for each state s of S_A there is a state q of S_B s.t. $s \equiv q$ and if $s \xRightarrow{a} s'$ then $q \xRightarrow{a} q'$ and $s' \equiv q'$ and s'
2. for each state q of S_B ...

Weakly Timed Bisimilar: allows ε-moves

Theorem ([C. & R., AVoCS'04])

Each bounded TPN (TPN_ε) is timed bisimilar to a TA (TA_ε).

Timed Bisimilarity

Definition (Weak Timed Bisimilarity)

Two TTS A and B are weakly timed bisimilar if there is an equivalence relation $\equiv$ on the states of S_A and S_B s.t.: $s_0^A \equiv s_0^B$ and

1. for each state s of S_A there is a state q of S_B s.t. $s \equiv q$ and if $s \overset{a}{\Longrightarrow} s'$ then $q \overset{a}{\Longrightarrow} q'$ and $s' \equiv q'$ and s'
2. for each state q of S_B ...

Weakly Timed Bisimilar: allows ε-moves

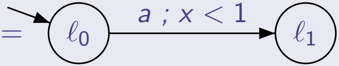
Theorem ([C. & R., AVoCS'04])

Each bounded TPN (TPN_ε) is timed bisimilar to a TA (TA_ε).

Converse: Each TA is timed bisimilar to a TPN ?

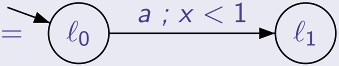
Theorem (TA are strictly more expressive than TPNs)

Let $\mathcal{A}_0 =$



Theorem (TA are strictly more expressive than TPNs)

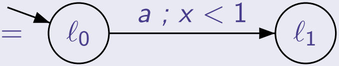
Let $\mathcal{A}_0 =$



There is **no TPN** *weakly timed bisimilar to* $\mathcal{A}_0$.

Theorem (TA are strictly more expressive than TPNs)

Let $\mathcal{A}_0 =$



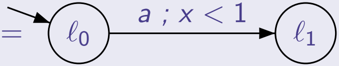
There is no TPN weakly timed bisimilar to $\mathcal{A}_0$*.*

Proof.

□

Theorem (TA are strictly more expressive than TPNs)

Let $\mathcal{A}_0 =$



There is no TPN *weakly timed bisimilar to* $\mathcal{A}_0$.

Proof.

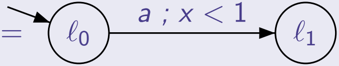
1. Time elapsing cannot disable transitions in a TPN

   $(M, \nu) \xrightarrow{t_1 t_2 \cdots t_k} (M', \nu')$ and $(M, \nu) \xrightarrow{\delta} (M'', \nu'')$

□

Theorem (TA are strictly more expressive than TPNs)

Let $\mathcal{A}_0 =$



There is no TPN weakly timed bisimilar to $\mathcal{A}_0$.

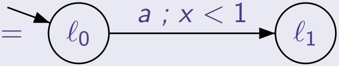
Proof.

1. Time elapsing cannot disable transitions in a TPN
   $(M, \nu) \xrightarrow{t_1 t_2 \cdots t_k} (M', \nu')$ and $(M, \nu) \xrightarrow{\delta} (M'', \nu'')$

□

Theorem (TA are strictly more expressive than TPNs)

Let $\mathcal{A}_0 =$



There is no TPN weakly timed bisimilar to $\mathcal{A}_0$*.*

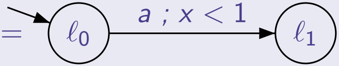
Proof.

1. Time elapsing cannot disable transitions in a TPN
   $(M, \nu) \xrightarrow{t_1 t_2 \cdots t_k} (M', \nu')$ and $(M, \nu) \xrightarrow{\delta} (M'', \nu'') \xrightarrow{t_1 t_2 \cdots t_k}$

□

Theorem (TA are strictly more expressive than TPNs)

Let $\mathcal{A}_0 =$



There is *no TPN* *weakly timed bisimilar to $\mathcal{A}_0$.*

Proof.

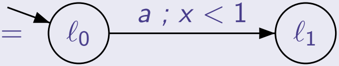
1. Time elapsing cannot disable transitions in a TPN
   $(M,\nu) \xrightarrow{t_1 t_2 \cdots t_k} (M',\nu')$ and $(M,\nu) \xrightarrow{\delta} (M'',\nu'') \xrightarrow{t_1 t_2 \cdots t_k}$
2. Assume there is a TPN $\mathcal{N}$ weakly timed bisimilar to $\mathcal{A}_0$

$$(\ell_0, x = 0) \xrightarrow{\delta = 1} (\ell_0, x = 1)$$

□

Theorem (TA are strictly more expressive than TPNs)

Let $\mathcal{A}_0 =$



There is *no TPN* *weakly timed bisimilar to* $\mathcal{A}_0$*.*

Proof.

1. Time elapsing cannot disable transitions in a TPN
   $(M,\nu) \xrightarrow{t_1 t_2 \cdots t_k} (M',\nu')$ and $(M,\nu) \xrightarrow{\delta} (M'',\nu'') \xrightarrow{t_1 t_2 \cdots t_k}$
2. Assume there is a TPN $\mathcal{N}$ weakly timed bisimilar to $\mathcal{A}_0$

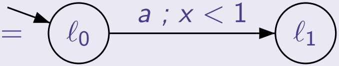
$$(\ell_0, x = 0) \xrightarrow{\delta = 1} (\ell_0, x = 1)$$

$(M_0, 0)$

□

Theorem (TA are strictly more expressive than TPNs)

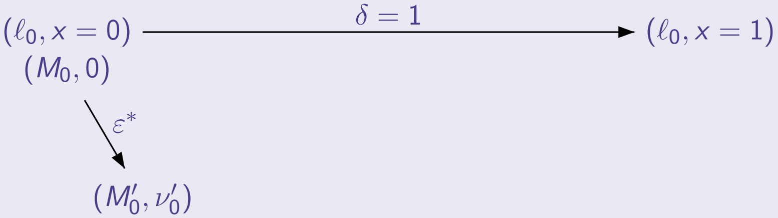
Let $\mathcal{A}_0 =$



There is ***no TPN*** *weakly timed bisimilar to* $\mathcal{A}_0$*.*

Proof.

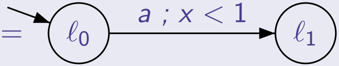
1. **Time elapsing cannot disable transitions in a TPN**
   $(M, \nu) \xrightarrow{t_1 t_2 \cdots t_k} (M', \nu')$ and $(M, \nu) \xrightarrow{\delta} (M'', \nu'') \xrightarrow{t_1 t_2 \cdots t_k}$
2. Assume there is a TPN $\mathcal{N}$ weakly timed bisimilar to $\mathcal{A}_0$



□

Theorem (TA are strictly more expressive than TPNs)

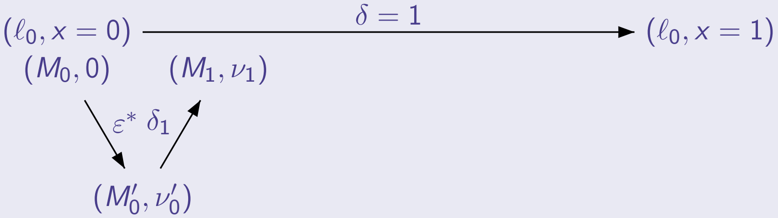
Let $\mathcal{A}_0 =$



There is **no TPN** *weakly timed bisimilar to* $\mathcal{A}_0$.

Proof.

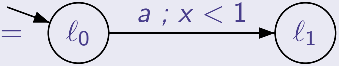
1. Time elapsing cannot disable transitions in a TPN
   $(M,\nu) \xrightarrow{t_1 t_2 \cdots t_k} (M',\nu')$ and $(M,\nu) \xrightarrow{\delta} (M'',\nu'') \xrightarrow{t_1 t_2 \cdots t_k}$
2. Assume there is a TPN $\mathcal{N}$ weakly timed bisimilar to $\mathcal{A}_0$



□

Theorem (TA are strictly more expressive than TPNs)

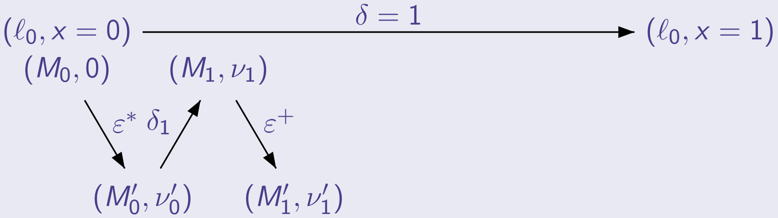
Let $\mathcal{A}_0$ =



There is no TPN weakly timed bisimilar to $\mathcal{A}_0$.

Proof.

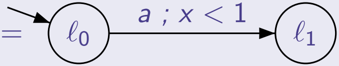
1. Time elapsing cannot disable transitions in a TPN
   $(M,\nu) \xrightarrow{t_1 t_2 \cdots t_k} (M',\nu')$ and $(M,\nu) \xrightarrow{\delta} (M'',\nu'') \xrightarrow{t_1 t_2 \cdots t_k}$
2. Assume there is a TPN $\mathcal{N}$ weakly timed bisimilar to $\mathcal{A}_0$



□

Theorem (TA are strictly more expressive than TPNs)

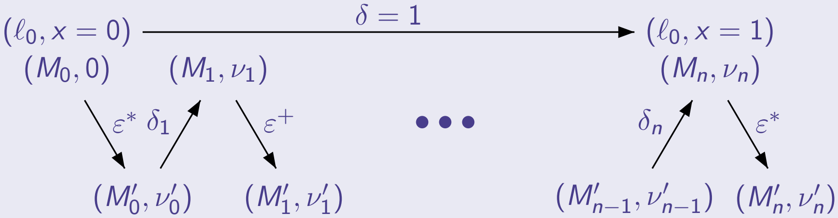
Let $\mathcal{A}_0 =$



There is no TPN *weakly timed bisimilar to* $\mathcal{A}_0$.

Proof.

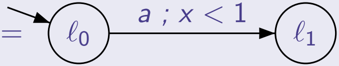
1. Time elapsing cannot disable transitions in a TPN
   $(M,\nu) \xrightarrow{t_1 t_2 \cdots t_k} (M',\nu')$ and $(M,\nu) \xrightarrow{\delta} (M'',\nu'') \xrightarrow{t_1 t_2 \cdots t_k}$
2. Assume there is a TPN $\mathcal{N}$ weakly timed bisimilar to $\mathcal{A}_0$



□

Theorem (TA are strictly more expressive than TPNs)

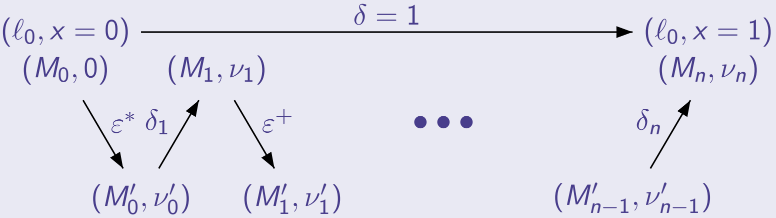
Let $\mathcal{A}_0$ =



There is no TPN weakly timed bisimilar to $\mathcal{A}_0$.

Proof.

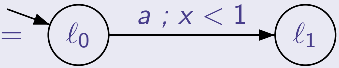
1. Time elapsing cannot disable transitions in a TPN
   $(M, \nu) \xrightarrow{t_1 t_2 \cdots t_k} (M', \nu')$ and $(M, \nu) \xrightarrow{\delta} (M'', \nu'') \xrightarrow{t_1 t_2 \cdots t_k}$
2. Assume there is a TPN $\mathcal{N}$ weakly timed bisimilar to $\mathcal{A}_0$



□

Theorem (TA are strictly more expressive than TPNs)

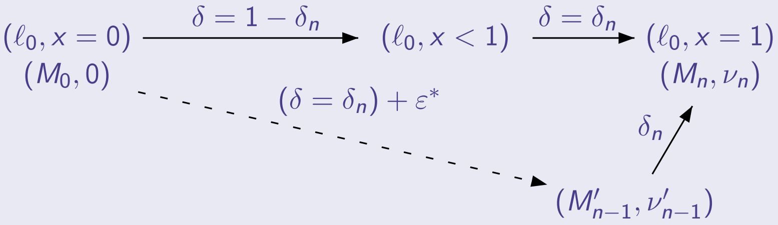
Let $\mathcal{A}_0 =$



There is *no TPN* *weakly timed bisimilar to* $\mathcal{A}_0$*.*

Proof.

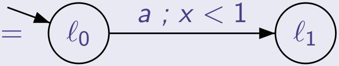
1. Time elapsing cannot disable transitions in a TPN
   $(M,\nu) \xrightarrow{t_1 t_2 \cdots t_k} (M',\nu')$ and $(M,\nu) \xrightarrow{\delta} (M'',\nu'') \xrightarrow{t_1 t_2 \cdots t_k}$
2. Assume there is a TPN $\mathcal{N}$ weakly timed bisimilar to $\mathcal{A}_0$



□

Theorem (TA are strictly more expressive than TPNs)

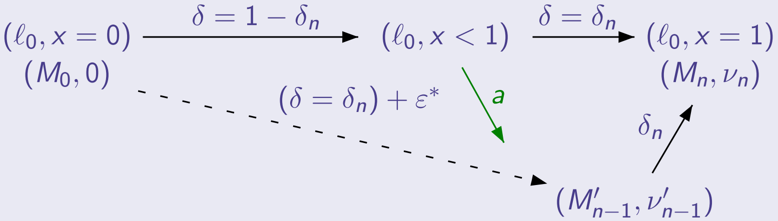
Let $\mathcal{A}_0 =$



There is no TPN *weakly timed bisimilar to* $\mathcal{A}_0$.

Proof.

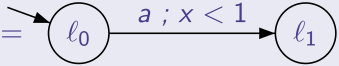
1. Time elapsing cannot disable transitions in a TPN
   $(M,\nu) \xrightarrow{t_1 t_2 \cdots t_k} (M',\nu')$ and $(M,\nu) \xrightarrow{\delta} (M'',\nu'') \xrightarrow{t_1 t_2 \cdots t_k}$
2. Assume there is a TPN $\mathcal{N}$ weakly timed bisimilar to $\mathcal{A}_0$



□

Theorem (TA are strictly more expressive than TPNs)

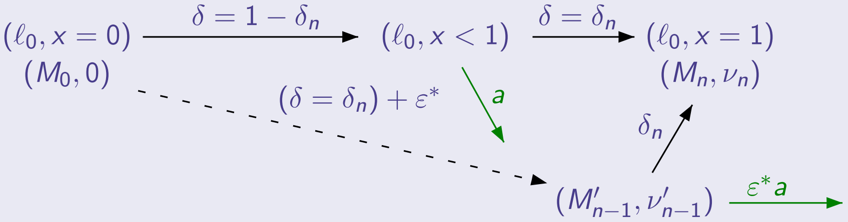
Let $\mathcal{A}_0 =$



There is no TPN *weakly timed bisimilar to $\mathcal{A}_0$.*

Proof.

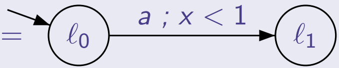
1. Time elapsing cannot disable transitions in a TPN
   $(M,\nu) \xrightarrow{t_1 t_2 \cdots t_k} (M',\nu')$ and $(M,\nu) \xrightarrow{\delta} (M'',\nu'') \xrightarrow{t_1 t_2 \cdots t_k}$
2. Assume there is a TPN $\mathcal{N}$ weakly timed bisimilar to $\mathcal{A}_0$



□

Theorem (TA are strictly more expressive than TPNs)

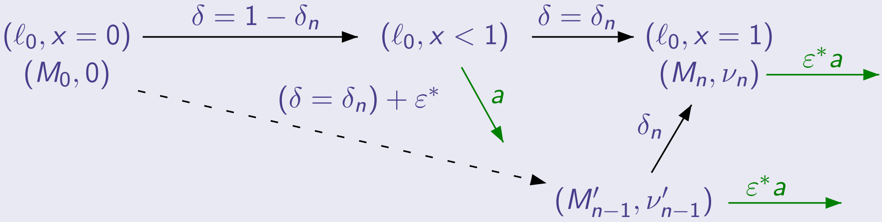
Let $\mathcal{A}_0 =$



There is no TPN weakly timed bisimilar to $\mathcal{A}_0$*.*

Proof.

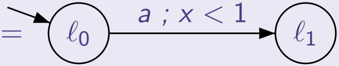
1. Time elapsing cannot disable transitions in a TPN
   $(M, \nu) \xrightarrow{t_1 t_2 \cdots t_k} (M', \nu')$ and $(M, \nu) \xrightarrow{\delta} (M'', \nu'') \xrightarrow{t_1 t_2 \cdots t_k}$
2. Assume there is a TPN $\mathcal{N}$ weakly timed bisimilar to $\mathcal{A}_0$



□

Theorem (TA are strictly more expressive than TPNs)

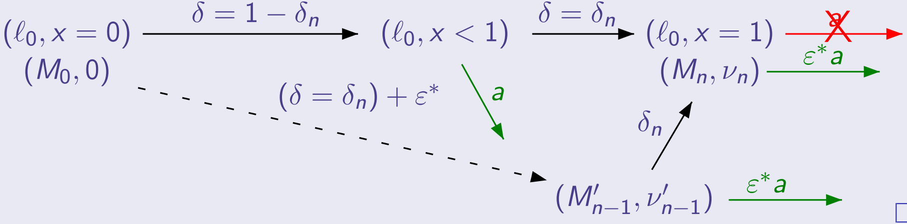
Let $\mathcal{A}_0 =$



There is no TPN *weakly timed bisimilar to* $\mathcal{A}_0$*.*

Proof.

1. Time elapsing cannot disable transitions in a TPN
   $(M,\nu) \xrightarrow{t_1 t_2 \cdots t_k} (M',\nu')$ and $(M,\nu) \xrightarrow{\delta} (M'',\nu'') \xrightarrow{t_1 t_2 \cdots t_k}$
2. Assume there is a TPN $\mathcal{N}$ weakly timed bisimilar to $\mathcal{A}_0$



□

Outline

- Context & Motivation
- Timed Automata & Time Petri Nets
- Expressiveness wrt Timed Bisimilarity
- **Expressiveness wrt Timed Language Acceptance**
- Conclusion

Language Equivalence

Definition (Timed Word & Timed Language)

A timed word over Σ is a sequence $w = (a_0, \delta_0)(a_1, \delta_1) \cdots (a_n, \delta_n) \cdots$ with $a_i \in \Sigma$; $\delta_i \in \mathbb{R}_{\geq 0}$.

A timed language is a set of timed words.

A TTS A with final and repeated states accepts a timed language $\mathcal{L}(A)$.

Language Equivalence

Definition (Timed Word & Timed Language)

A timed word over Σ is a sequence $w = (a_0, \delta_0)(a_1, \delta_1) \cdots (a_n, \delta_n) \cdots$ with $a_i \in \Sigma$; $\delta_i \in \mathbb{R}_{\geq 0}$.
A timed language is a set of timed words.
A TTS A with final and repeated states accepts a timed language $\mathcal{L}(A)$.

Definition (Language Equivalence)

Two TTS A and B are equivalent w.r.t. Timed Language Acceptance if

$$\mathcal{L}(A) = \mathcal{L}(B).$$

Language Equivalence

Definition (Timed Word & Timed Language)

A timed word over Σ is a sequence $w = (a_0, \delta_0)(a_1, \delta_1) \cdots (a_n, \delta_n) \cdots$ with $a_i \in \Sigma$; $\delta_i \in \mathbb{R}_{\geq 0}$.
A timed language is a set of timed words.
A TTS A with final and repeated states accepts a timed language $\mathcal{L}(A)$.

Definition (Language Equivalence)

Two TTS A and B are equivalent w.r.t. Timed Language Acceptance if

$$\mathcal{L}(A) = \mathcal{L}(B).$$

Theorem ([C. & R., AVoCS'04])

A timed language accepted by a bounded TPN is accepted by a TA.

Language Equivalence

Definition (Timed Word & Timed Language)

A timed word over Σ is a sequence $w = (a_0, \delta_0)(a_1, \delta_1) \cdots (a_n, \delta_n) \cdots$ with $a_i \in \Sigma$; $\delta_i \in \mathbb{R}_{\geq 0}$.
A timed language is a set of timed words.
A TTS A with final and repeated states accepts a timed language $\mathcal{L}(A)$.

Definition (Language Equivalence)

Two TTS A and B are equivalent w.r.t. Timed Language Acceptance if

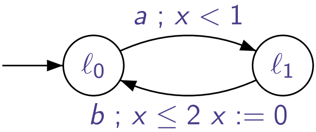
$$\mathcal{L}(A) = \mathcal{L}(B).$$

Theorem ([C. & R., AVoCS'04])

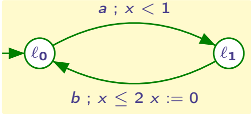
A timed language accepted by a bounded TPN is accepted by a TA.

Each language accepted by a TA is accepted by a TPN ?

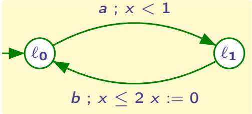
Encoding a TA into a Bounded TPN



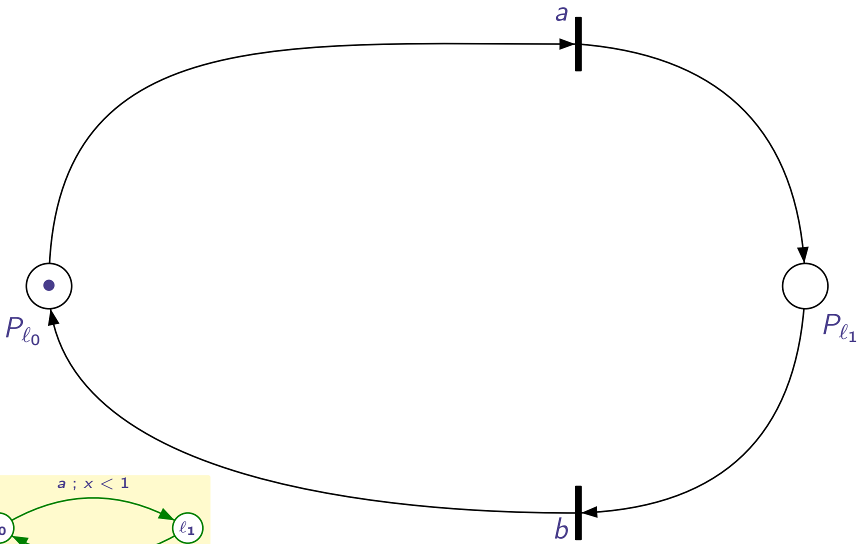
Encoding a TA into a Bounded TPN

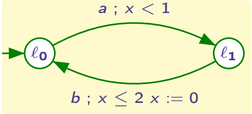


Encoding a TA into a Bounded TPN

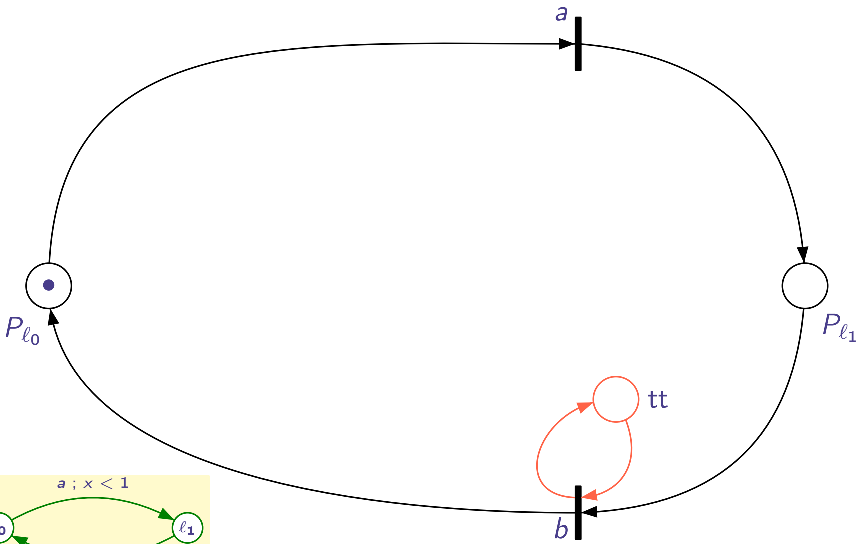


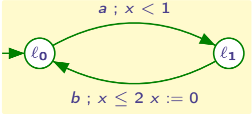
Encoding a TA into a Bounded TPN



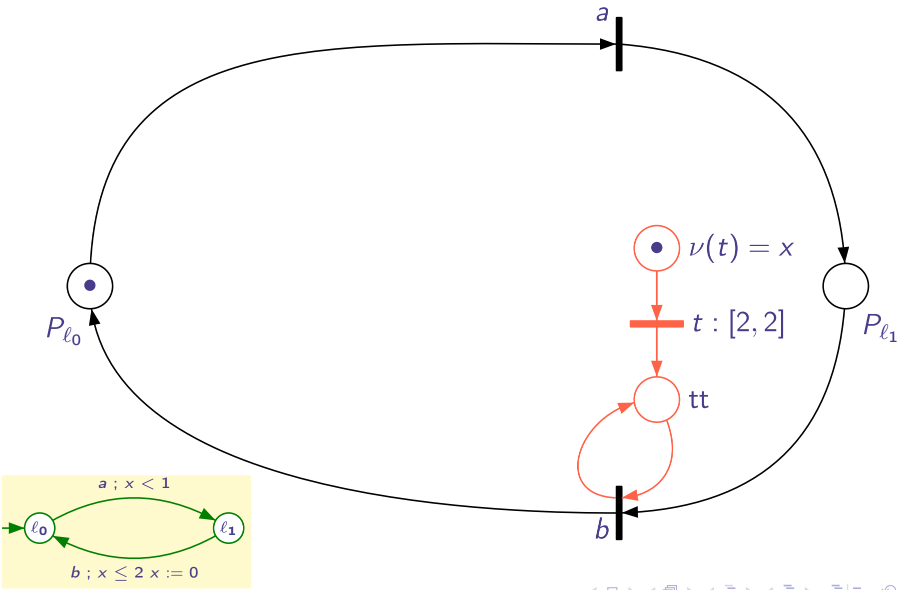


Encoding a TA into a Bounded TPN

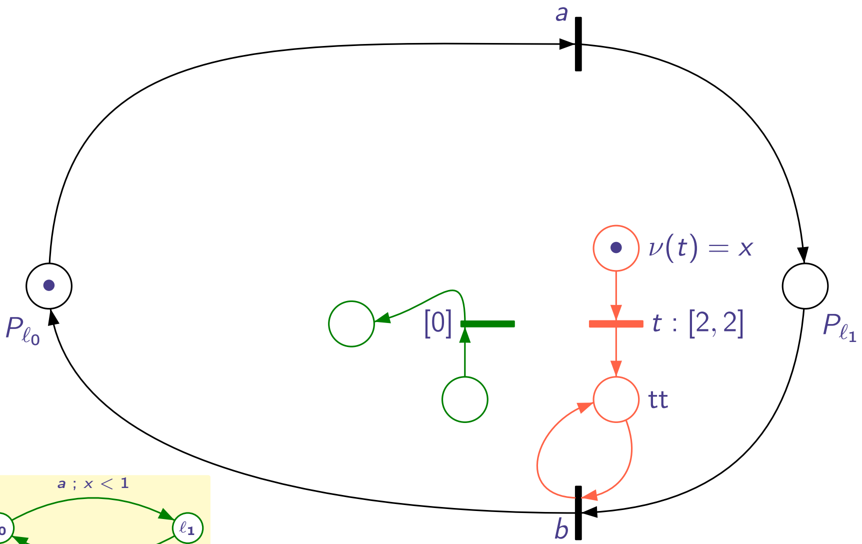


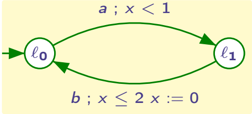


Encoding a TA into a Bounded TPN

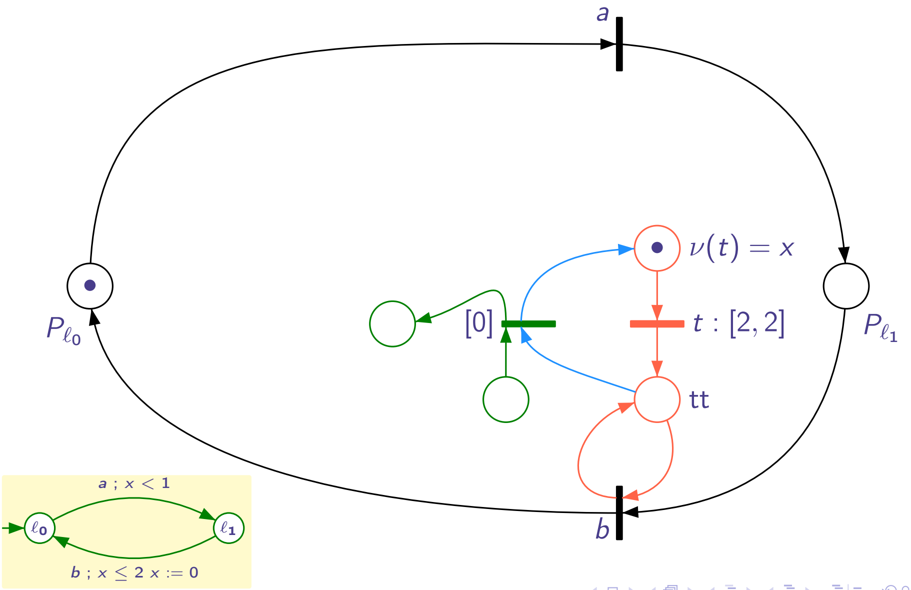


Encoding a TA into a Bounded TPN

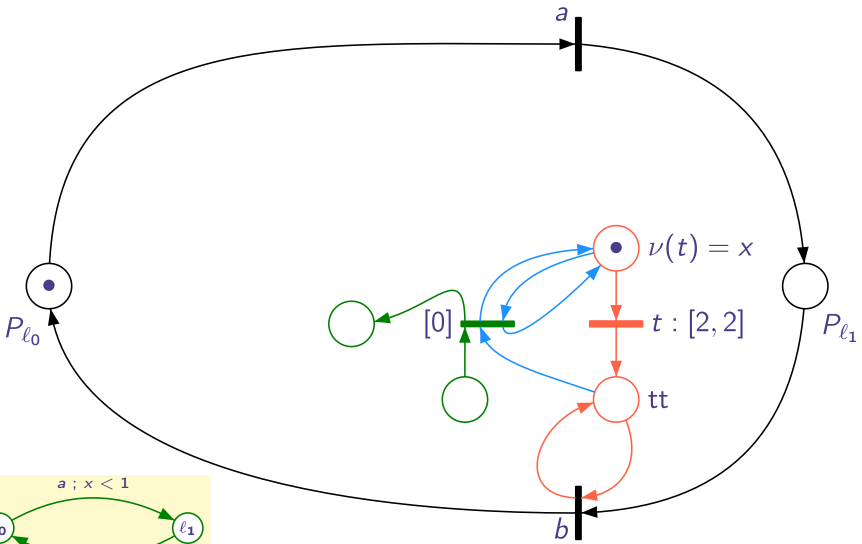


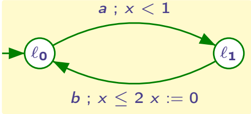


Encoding a TA into a Bounded TPN

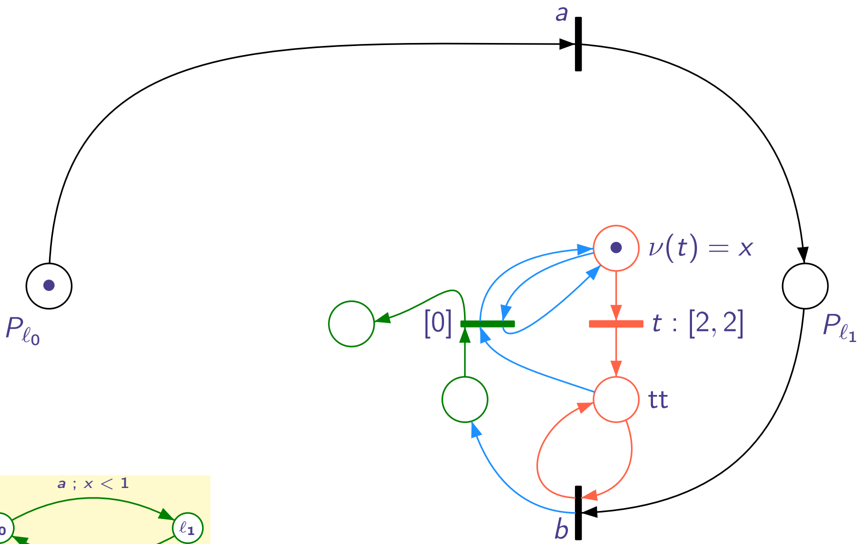


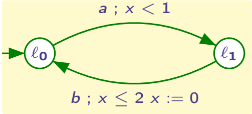
Encoding a TA into a Bounded TPN



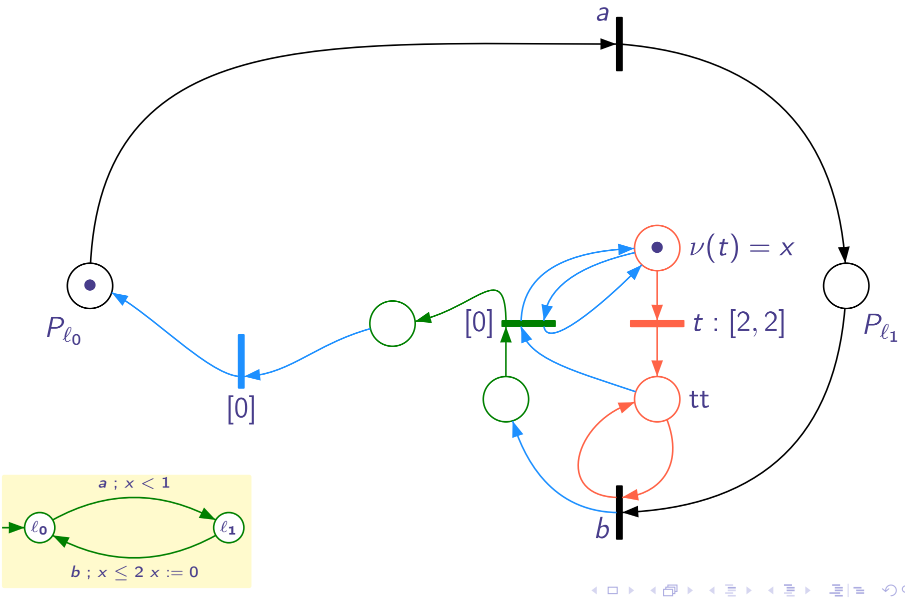


Encoding a TA into a Bounded TPN

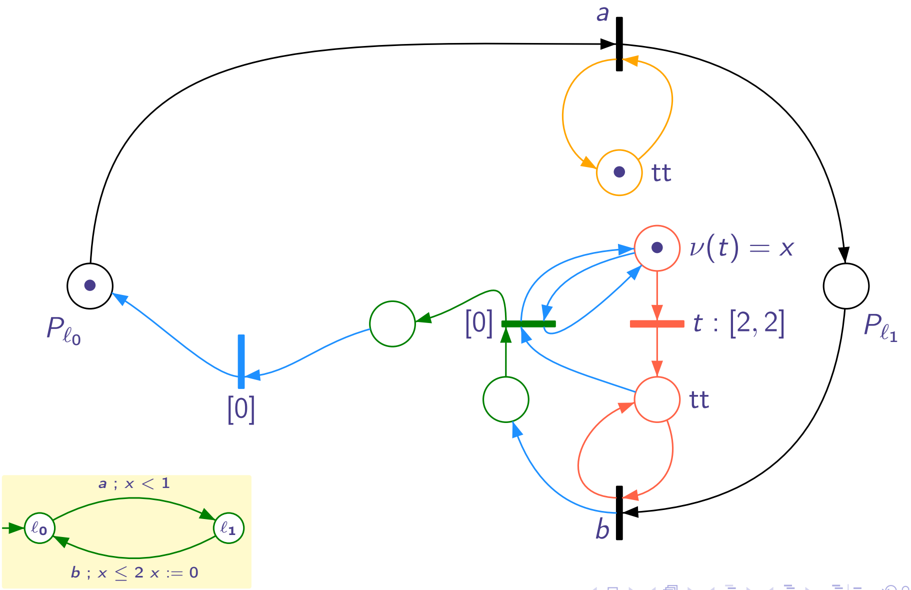




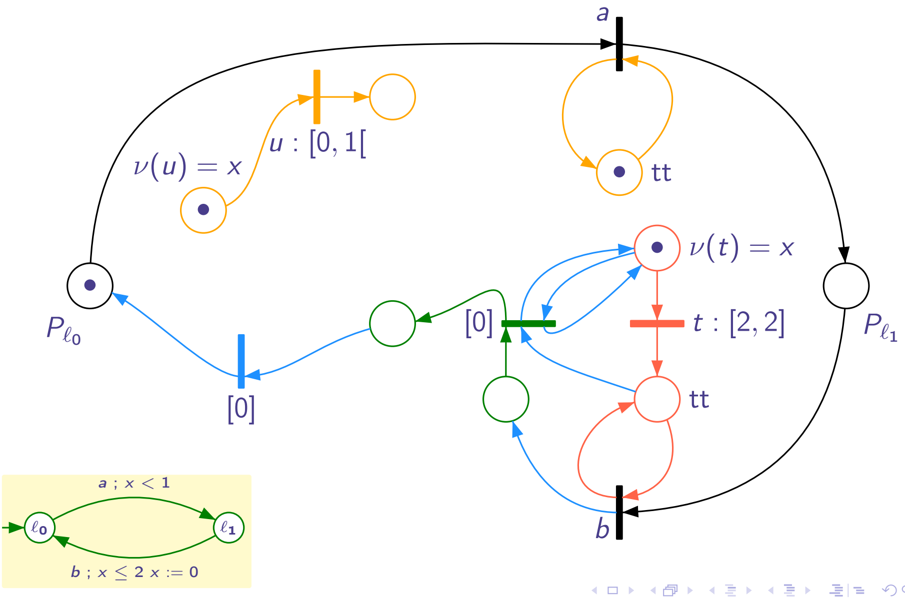
Encoding a TA into a Bounded TPN



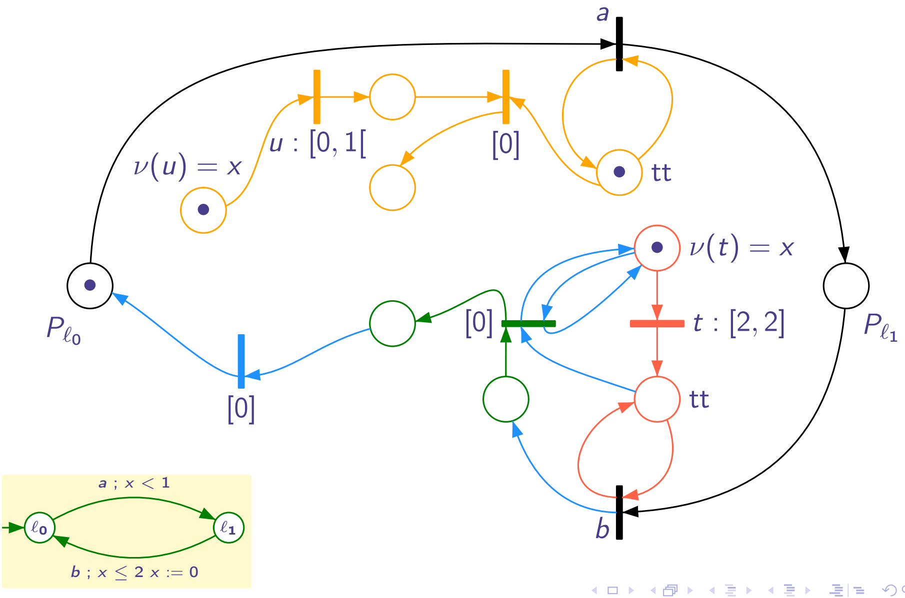
Encoding a TA into a Bounded TPN



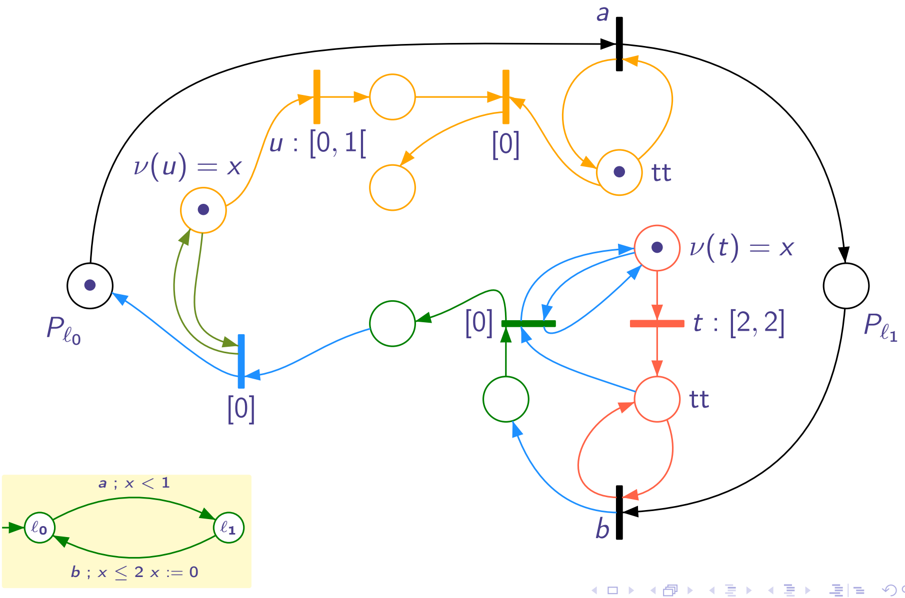
Encoding a TA into a Bounded TPN



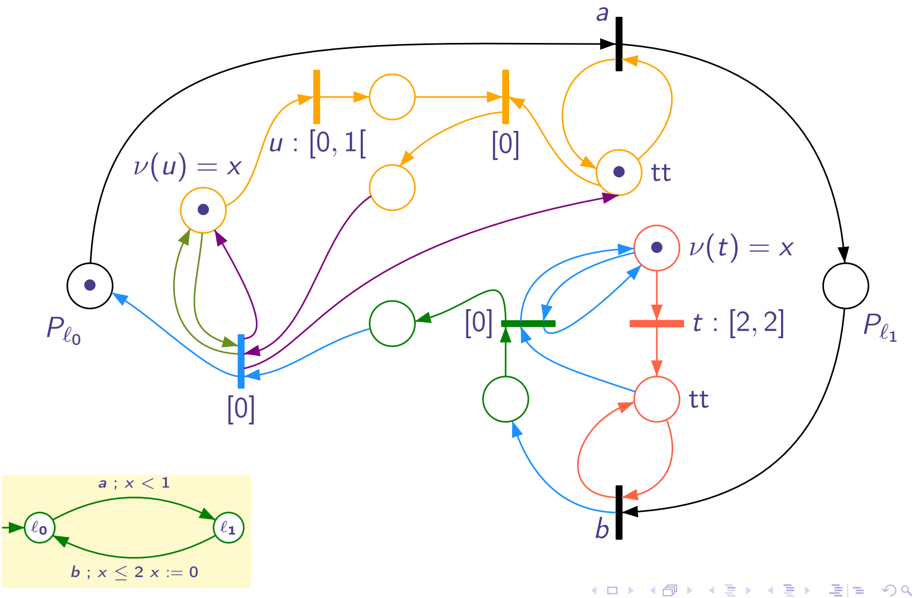
Encoding a TA into a Bounded TPN



Encoding a TA into a Bounded TPN



Encoding a TA into a Bounded TPN



TA and TPN are equally Expressive

Theorem

One-safe TPN_ε and TA_ε are equally expressive w.r.t. Timed Language Acceptance.

TA and TPN are equally Expressive

Theorem

One-safe TPN_ε and TA_ε are equally expressive w.r.t. Timed Language Acceptance.

Sketch.

$\mathcal{A}$ a TA; $\mathcal{N}$ the TPN as described previously.
To prove $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{N})$ we prove

□

TA and TPN are equally Expressive

Theorem

One-safe TPN_ε and TA_ε are equally expressive w.r.t. Timed Language Acceptance.

Sketch.

$\mathcal{A}$ a TA; $\mathcal{N}$ the TPN as described previously.
To prove $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{N})$ we prove

1. $\mathcal{N}$ simulates $\mathcal{A}$ which entails $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{N})$
   Define a proper simulation relation

□

TA and TPN are equally Expressive

Theorem

One-safe TPN_ε and TA_ε are equally expressive w.r.t. Timed Language Acceptance.

Sketch.

$\mathcal{A}$ a TA; $\mathcal{N}$ the TPN as described previously.
To prove $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{N})$ we prove

1. $\mathcal{N}$ simulates $\mathcal{A}$ which entails $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{N})$
   Define a proper simulation relation
2. for $\mathcal{L}(\mathcal{N}) \subseteq \mathcal{L}(\mathcal{A})$:
   Design $\mathcal{A}'$ s.t. $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$ and prove that $\mathcal{A}'$ simulates $\mathcal{N}$

□

Back to Timed Bisimilarity

Definition (The class $TA_\varepsilon(\leq, \geq)$)

Back to Timed Bisimilarity

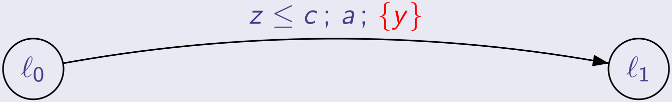
Definition (The class $TA_{\varepsilon}(\leq, \geq)$)

- Guards are of the form $x \geq c$ or $x \leq c$

Back to Timed Bisimilarity

Definition (The class $TA_\varepsilon(\leq, \geq)$)

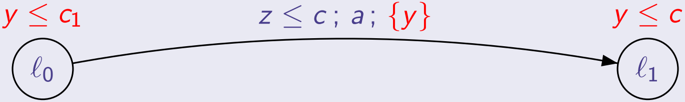
- Guards are of the form $x \geq c$ or $x \leq c$
- Invariants are increasing: between two resets of a clock x, the sequence of invariants encountered from any location is of the form $x \leq c_1$ and later on $x \leq c_2$ with $c_1 \leq c_2$ etc



Back to Timed Bisimilarity

Definition (The class $TA_\varepsilon(\leq, \geq)$)

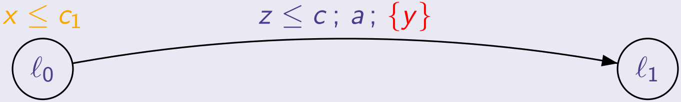
- Guards are of the form $x \geq c$ or $x \leq c$
- Invariants are increasing: between two resets of a clock x, the sequence of invariants encountered from any location is of the form $x \leq c_1$ and later on $x \leq c_2$ with $c_1 \leq c_2$ etc



Back to Timed Bisimilarity

Definition (The class $TA_{\varepsilon}(\leq, \geq)$)

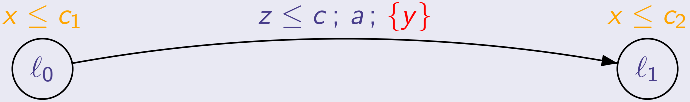
- Guards are of the form $x \geq c$ or $x \leq c$
- Invariants are increasing: between two resets of a clock x, the sequence of invariants encountered from any location is of the form $x \leq c_1$ and later on $x \leq c_2$ with $c_1 \leq c_2$ etc



Back to Timed Bisimilarity

Definition (The class $TA_\varepsilon(\leq, \geq)$)

- Guards are of the form $x \geq c$ or $x \leq c$
- Invariants are increasing: between two resets of a clock x, the sequence of invariants encountered from any location is of the form $x \leq c_1$ and later on $x \leq c_2$ with $c_1 \leq c_2$ etc



Back to Timed Bisimilarity

Definition (The class $TA_\varepsilon(\leq, \geq)$)

- Guards are of the form $x \geq c$ or $x \leq c$
- Invariants are increasing: between two resets of a clock x, the sequence of invariants encountered from any location is of the form $x \leq c_1$ and later on $x \leq c_2$ with $c_1 \leq c_2$ etc

Theorem (For Timed Bisimilarity)

B-$TPN_\varepsilon(\leq, \geq)$ and $TA_\varepsilon(\leq, \geq)$ are equally expressive.

Proof.

- From B-$TPN_\varepsilon(\leq, \geq)$ to $TA_\varepsilon(\leq, \geq)$: [C. & R., AVoCS'04]

□

Back to Timed Bisimilarity

Definition (The class $TA_\varepsilon(\leq, \geq)$)

- Guards are of the form $x \geq c$ or $x \leq c$
- Invariants are increasing: between two resets of a clock x, the sequence of invariants encountered from any location is of the form $x \leq c_1$ and later on $x \leq c_2$ with $c_1 \leq c_2$ etc

Theorem (For Timed Bisimilarity)

B-$TPN_\varepsilon(\leq, \geq)$ and $TA_\varepsilon(\leq, \geq)$ are equally expressive.

Proof.

- From B-$TPN_\varepsilon(\leq, \geq)$ to $TA_\varepsilon(\leq, \geq)$: [C. & R., AVoCS'04]
- From $TA_\varepsilon(\leq, \geq)$ to B-$TPN_\varepsilon(\leq, \geq)$
  Extension of the previous construction for Timed Language

□

New Results for TPNs

Theorem (One is enough)

One-safe B-TPN$_\varepsilon$ and B-TPN$_\varepsilon$ are equally expressive w.r.t. timed language acceptance.

New Results for TPNs

Theorem (One is enough)

One-safe B-TPN$_\varepsilon$ and B-TPN$_\varepsilon$ are equally expressive w.r.t. timed language acceptance.

Theorem (Universal Problem Undecidable)

The universal problem is undecidable for B-TPN$_\varepsilon$.

New Results for TPNs

Theorem (One is enough)

One-safe B-TPN$_\varepsilon$ and B-TPN$_\varepsilon$ are equally expressive w.r.t. timed language acceptance.

Theorem (Universal Problem Undecidable)

The universal problem is undecidable for B-TPN$_\varepsilon$.

Corollary (Language Inclusion Undecidable)

Language Inclusion in undecidable for B-TPN$_\varepsilon$.

New Results for TPNs

Theorem (One is enough)

One-safe B-TPN$_\varepsilon$ and B-TPN$_\varepsilon$ are equally expressive w.r.t. timed language acceptance.

Theorem (Universal Problem Undecidable)

The universal problem is undecidable for B-TPN$_\varepsilon$.

Corollary (Language Inclusion Undecidable)

Language Inclusion in undecidable for B-TPN$_\varepsilon$.

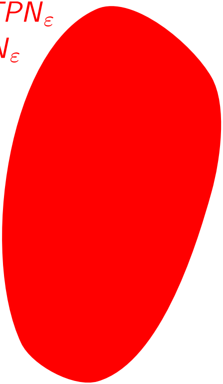
Theorem

One-safe B-TPN$_\varepsilon(\leq, \geq)$ and B-TPN$_\varepsilon(\leq, \geq)$ are equally expressive w.r.t. weak timed bisimilarity.

Current Picture

Timed Language Acceptance

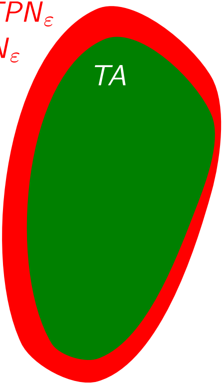
$TA_\varepsilon = \text{B-}TPN_\varepsilon$
$= \text{1-B-}TPN_\varepsilon$



Current Picture

Timed Language Acceptance

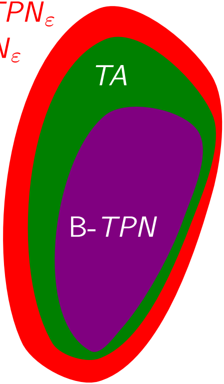
TA_ε = B-TPN_ε
= 1-B-TPN_ε



Current Picture

Timed Language Acceptance

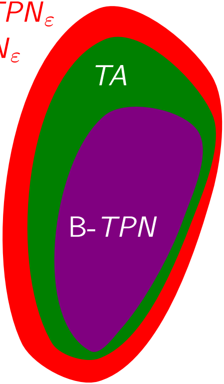
$TA_\varepsilon = \text{B-}TPN_\varepsilon$
$= \text{1-B-}TPN_\varepsilon$



Current Picture

Timed Language Acceptance

$TA_\varepsilon = \text{B-}TPN_\varepsilon$
$= \text{1-B-}TPN_\varepsilon$



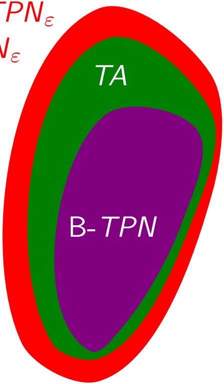
Q:B-$TPN = TA$?

Current Picture

Timed Language Acceptance

Timed Bisimilarity

TA_ε = B-TPN_ε
= 1-B-TPN_ε

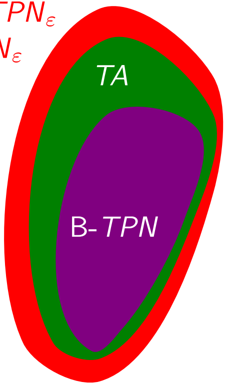


Q:B-TPN = TA ?

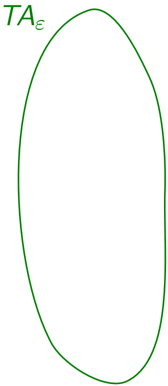
Current Picture

Timed Language Acceptance

$TA_\varepsilon = \text{B-}TPN_\varepsilon$
$= \text{1-B-}TPN_\varepsilon$



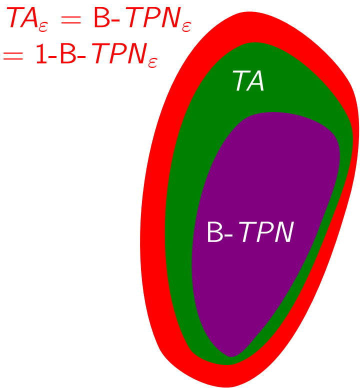
Timed Bisimilarity



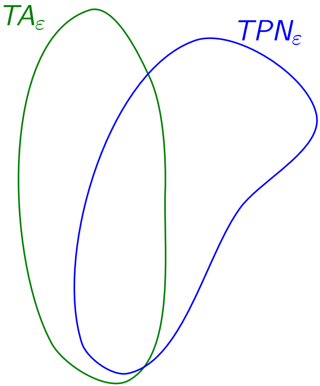
Q:B-$TPN = TA$?

Current Picture

Timed Language Acceptance



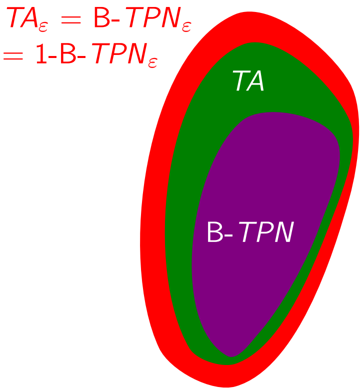
Timed Bisimilarity



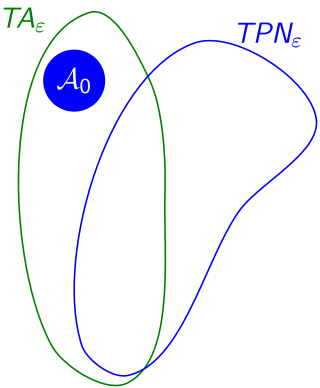
Q:B-$TPN = TA$?

Current Picture

Timed Language Acceptance



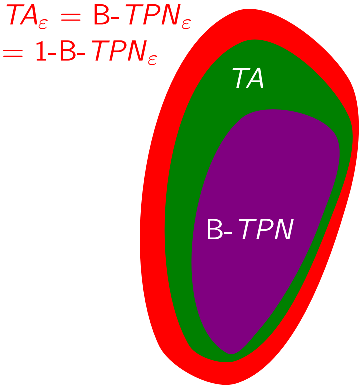
Timed Bisimilarity



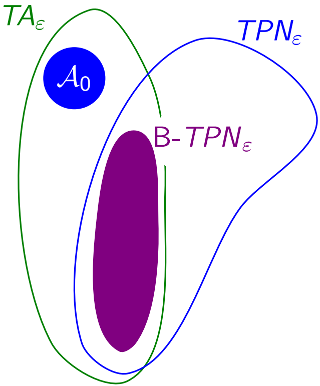
Q:B-TPN = TA ?

Current Picture

Timed Language Acceptance



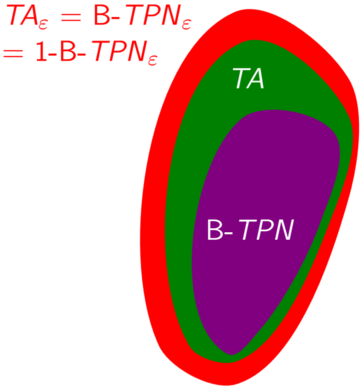
Timed Bisimilarity



Q:B-TPN = TA ?

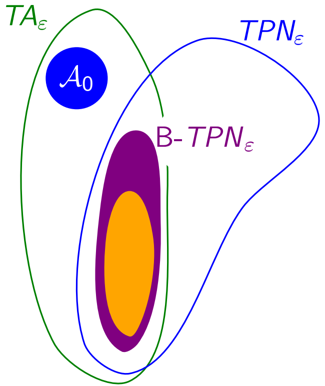
Current Picture

Timed Language Acceptance



Q:B-$TPN = TA$?

Timed Bisimilarity



$TA(\leq,\geq) = \text{B-}TPN_\varepsilon(\leq,\geq)$
$= \text{1-B-}TPN_\varepsilon(\leq,\geq)$

Outline

- Context & Motivation
- Timed Automata & Time Petri Nets
- Expressiveness wrt Timed Bisimilarity
- Expressiveness wrt Timed Language Acceptance
- Conclusion

Conclusion & Recent Work

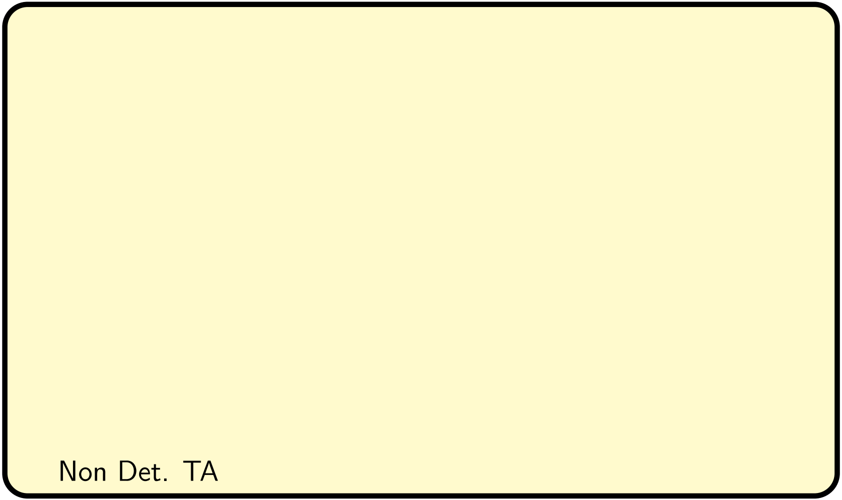
Results:

- Expressive Power of TA vs. TPNs
  Timed Language Acceptance and Timed Bisimilarity
- Undecidability of the Universal Problem
- Equivalence between one-safe TPN and TPN

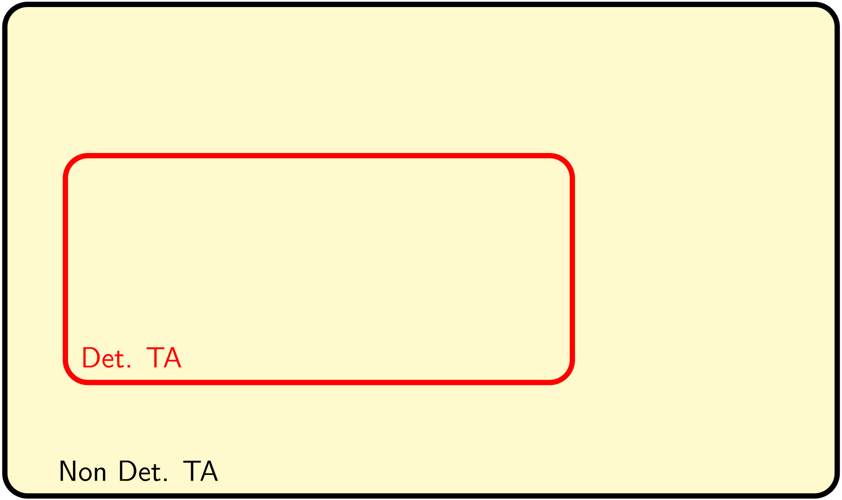
Recent Results:

- more than one semantics for TPN
  [Bérard et al., ATVA'05]
- semantic definition of the class of TA that are timed bisimilar to TPN
  [Bérard et al., FSTTCS'05]

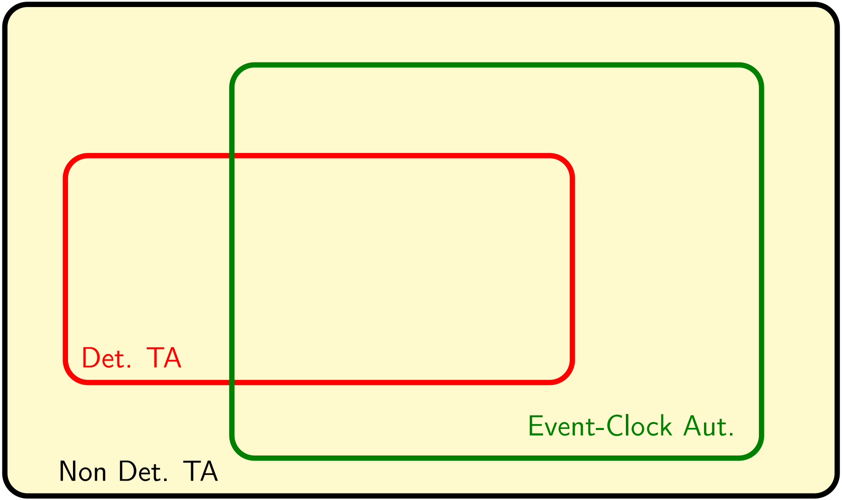
TPN and Event-Clock Automata [Alur et al.'99]



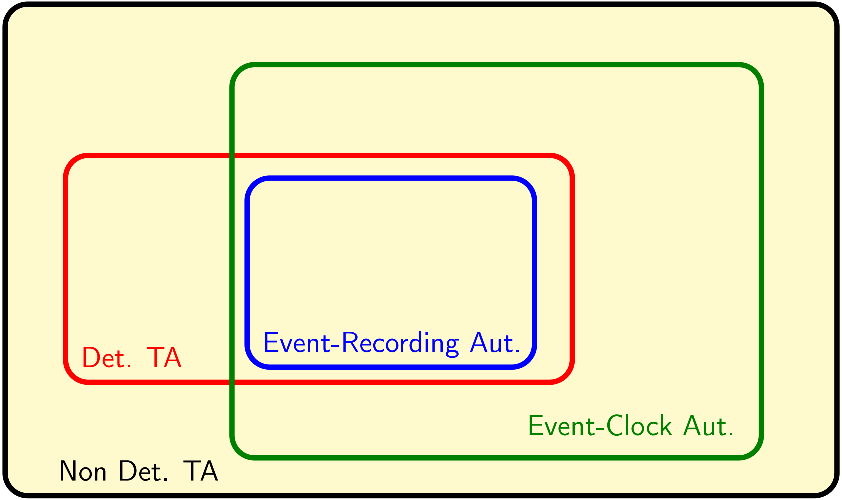
TPN and Event-Clock Automata [Alur et al.'99]



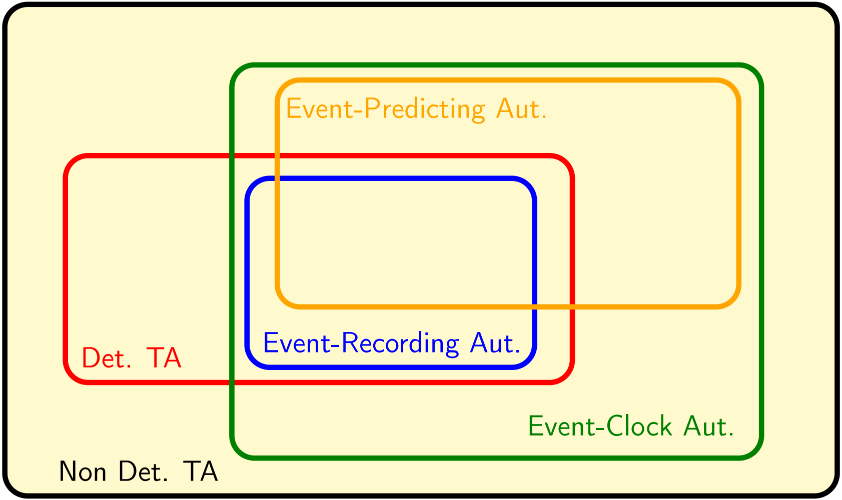
TPN and Event-Clock Automata [Alur et al.'99]



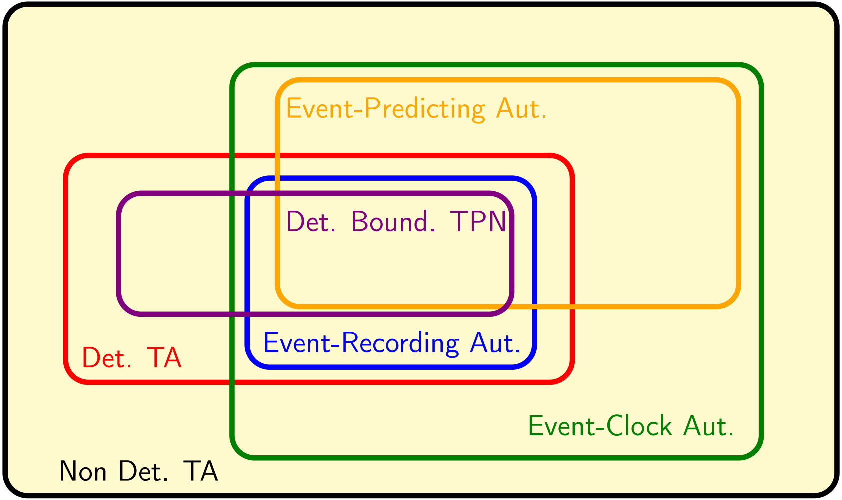
TPN and Event-Clock Automata [Alur et al.'99]



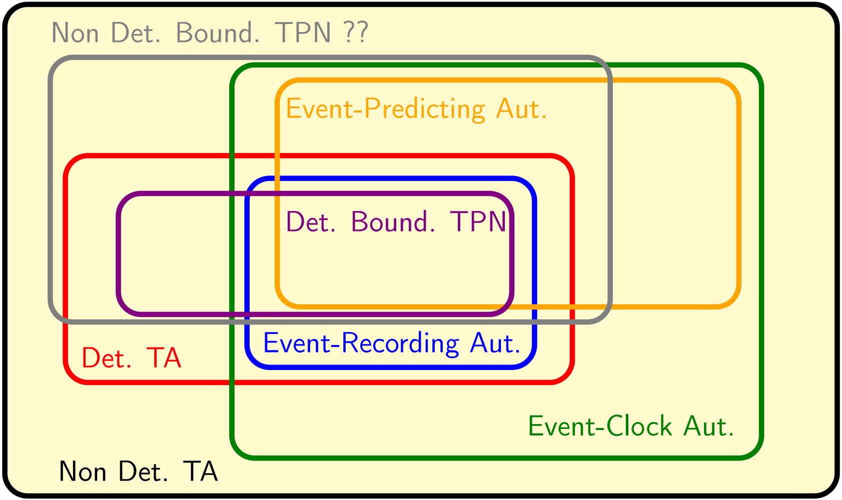
TPN and Event-Clock Automata [Alur et al.'99]



TPN and Event-Clock Automata [Alur et al.'99]

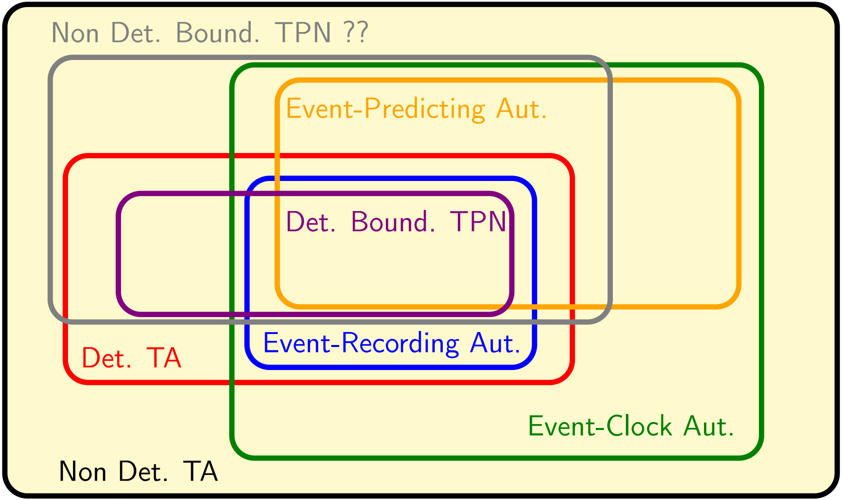


TPN and Event-Clock Automata [Alur et al.'99]



TPN and Event-Clock Automata [Alur et al.'99]

Waterloo Station •



References

R. Alur and D. Dill.
A theory of timed automata.
Theoretical Computer Science B, 126:183–235, 1994.

R. Alur, L. Fix, and T. A. Henzinger.
Event-Clock Automata: A Determinizable Class of Timed Automata.
Theoretical Computer Science B, 211:253-273, 1999.

B. Bérard, F. Cassez, S. Haddad, O. H. Roux, and D. Lime.
Comparison of Different Semantics for Time Petri Nets.
In *Proceedings of the Third International Symposium on Automated Technology for Verification and Analysis (ATVA'2005)*, volume 3707 of *Lecture Notes in Computer Science*, Taipei, Taiwan, October 2005. Springer.
To appear.

References (cont.)

B. Bérard, F. Cassez, S. Haddad, O. H. Roux, and D. Lime.
When are timed automata weakly timed bisimilar to timed Patri nets ?.
In *FSTTCS'05* Hyberadad, India, December 2005. Springer.
To appear.

B. Bérard, P. Gastin and A. Petit.
On the Power of Non-Observable Actions in Timed Automata.
Proceedings of the 13th Annual Symposium on Theoretical Aspects of Computer Science (STACS'96), LNCS 1046, pages 257–268, 1996

B. Berthomieu and M. Diaz.
Modeling and verification of time dependent systems using time Petri nets.
IEEE Transactions on Software Engineering, 17(3):259–273, March 1991.

References (cont.)

F. Cassez and O.H. Roux.
Structural Translation of Time Petri Nets into Timed Automata.
In Michael Huth, editor, *Workshop on Automated Verification of Critical Systems (AVoCS'04)*, Electronic Notes in Computer Science. Elsevier, August 2004.

P.M. Merlin.
A study of the recoverability of computing systems.
PhD thesis, University of California, Irvine, CA, 1974.

Timed Automata [Alur & Dill'94]

Back

A Timed Automaton $\mathcal{A}$ is a tuple $(L, \ell_0, \mathrm{Act}, X, \mathrm{inv}, \longrightarrow)$ where:

- L is a finite set of locations
- ℓ_0 is the initial location
- X is a finite set of clocks
- Act is a finite set of actions
- $\longrightarrow$ is a set of transitions of the form $\ell \xrightarrow{g, a, R} \ell'$ with:
  - $\ell, \ell' \in L$,
  - $a \in \mathrm{Act}$
  - a guard g which is a clock constraint over X
  - a reset set R which is the set of clocks to be reset to 0

Clock constraints are boolean combinations of $x \sim k$ with $x \in C$ and $k \in \mathbb{Z}$ and $\sim \in \{\leq, <\}$.

Semantics of Timed Automata

Back

Let $\mathcal{A} = (L, \ell_0, \mathrm{Act}, X, \mathrm{inv}, \longrightarrow)$ be a Timed Automaton.

A state (ℓ, v) of $\mathcal{A}$ is in $L \times \mathbb{R}_{\geq 0}^X$

The semantics of $\mathcal{A}$ is a Timed Transition System $S_{\mathcal{A}} = (Q, q_0, \mathrm{Act} \cup \mathbb{R}_{\geq 0}, \longrightarrow)$ with:

- $Q = L \times \mathbb{R}_{\geq 0}^X$
- $q_0 = (\ell_0, \overline{0})$
- $\longrightarrow$ consists in:

discrete transition: $(\ell, v) \xrightarrow{a} (\ell', v') \iff \begin{cases} \exists\, \ell \xrightarrow{g,a,r} \ell' \in \mathcal{A} \\ v \models g \\ v' = v[r \leftarrow 0] \\ v' \models \mathrm{inv}(\ell') \end{cases}$

delay transition: $(\ell, v) \xrightarrow{d} (\ell, v + d) \iff d \in \mathbb{R}_{\geq 0} \wedge v + d \models \mathrm{inv}(\ell)$

Time Petri Nets

Back

A *Time Petri Net* $\mathcal{N}$ is a tuple $(P, T, \Sigma_\varepsilon, {}^\bullet(.), (.)^\bullet, M_0, \Lambda, I)$ where:

- $P = \{p_1, p_2, \cdots, p_m\}$ is a finite set of places
- $T = \{t_1, t_2, \cdots, t_n\}$ is a finite set of transitions and $P \cap T = \emptyset$;
- Σ is a finite set of actions
- ${}^\bullet(.) \in (\mathbb{N}^P)^T$ is the backward incidence mapping; $(.)^\bullet \in (\mathbb{N}^P)^T$ is the forward incidence mapping;
- $M_0 \in \mathbb{N}^P$ is the initial marking;
- $\Lambda : T \to \Sigma_\varepsilon$ is the labeling function;
- $I : T \to \mathcal{I}(\mathbb{Q}_{\geq 0})$ associates with each transition a firing interval;

Semantics of Time Petri Nets

◂ Back

Let $\mathcal{N} = (P, T, \Sigma_\varepsilon, {}^\bullet(.), (.)^\bullet, M_0, \Lambda, I)$ be a Time Petri Net.

A state (M, ν) of $\mathcal{N}$ is a pair with in $\mathbb{N}^P$ and $\nu \in \mathbb{R}_{\geq 0}^{Enabled(M)}$.
$ADM(\mathcal{N})$ is the set of states of $\mathcal{N}$.

The semantics of $\mathcal{N}$ is a Timed Transition System $S_\mathcal{N} = (Q, \{q_0\}, T, \rightarrow)$:

- $Q = ADM(\mathcal{N})$,
- $q_0 = (M_0, \mathbf{0})$, $F' = \{(M, \nu) \mid \; M \in F\}$ and
- $\longrightarrow \in Q \times (T \cup \mathbb{R}_{\geq 0}) \times Q$ is the transition relations:
  - the discrete transition relation is defined $\forall t \in T$ by:

$$(M,\nu) \xrightarrow{\Lambda(t)} (M',\nu') \textit{ iff } \begin{cases} t \in Enabled(M) \wedge M' = M - {}^\bullet t + t^\bullet \\ \nu(t) \in I(t), \\ \forall t \in \mathbb{R}_{\geq 0}^{Enabled(M')}, \nu'(t) = \begin{cases} 0 \textit{ if } \uparrow enabled(t', M, t), \\ \nu(t) \textit{ otherwise.} \end{cases} \end{cases}$$

Semantics of Time Petri Nets

◂ Back

Let $\mathcal{N} = (P, T, \Sigma_\varepsilon, {}^\bullet(.), (.)^\bullet, M_0, \Lambda, I)$ be a Time Petri Net.

A state (M, ν) of $\mathcal{N}$ is a pair with in $\mathbb{N}^P$ and $\nu \in \mathbb{R}_{\geq 0}^{Enabled(M)}$.
$ADM(\mathcal{N})$ is the set of states of $\mathcal{N}$.

The semantics of $\mathcal{N}$ is a Timed Transition System $S_{\mathcal{N}} = (Q, \{q_0\}, T, \rightarrow)$:

- $Q = ADM(\mathcal{N})$,
- $q_0 = (M_0, \mathbf{0})$, $F' = \{(M, \nu) \mid M \in F\}$ and
- $\longrightarrow \in Q \times (T \cup \mathbb{R}_{\geq 0}) \times Q$ is the transition relations:
  - the discrete transition relation is defined $\forall t \in T$ by:
  - continuous transition relation is defined $\forall d \in \mathbb{R}_{\geq 0}$ by:

$$(M, \nu) \xrightarrow{d} (M, \nu') \text{ iff } \begin{cases} \nu' = \nu + d \\ \forall t \in Enabled(M), \nu'(t) \in I(t)^{\downarrow} \end{cases}$$