Intersection of regular signal-event (timed) languages

Béatrice Bérard

LAMSADE

Université Paris-Dauphine & CNRS

Beatrice.Berard@dauphine.fr

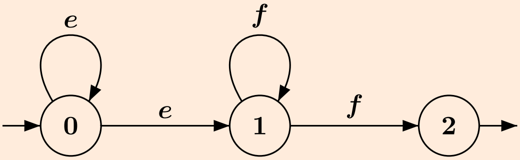
Joint work with Paul Gastin and Antoine Petit

FORMATS, September 26th, 2006

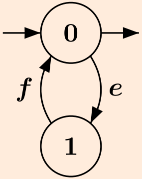
Outline

Closure under intersection

is well known for regular languages



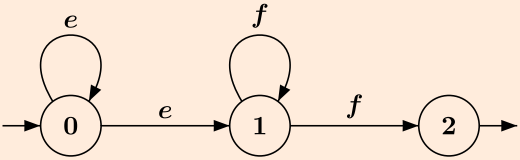
accepts $L_1 = e^+ f^+$



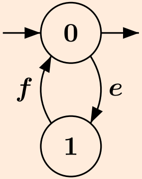
accepts $L_2 = (ef)^*$

Closure under intersection

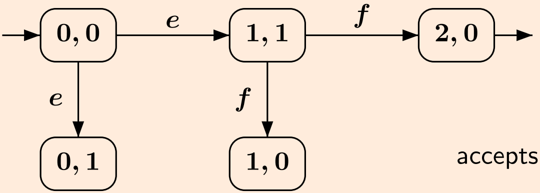
is well known for regular languages



accepts $L_1 = e^+ f^+$



accepts $L_2 = (ef)^*$



accepts $L_1 \cap L_2 = \{ef\}$

Closure under intersection

is a nice property

An implementation $\mathcal{I} = \mathcal{L}(\mathcal{M})$ cannot behave badly as specified by $\mathcal{B} = \mathcal{L}(\mathcal{P})$:

$$\mathcal{L}(\mathcal{M}) \cap \mathcal{L}(\mathcal{P}) = \emptyset$$

Build a machine $\mathcal{A}$ in the same class as $\mathcal{M}$ and $\mathcal{P}$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{M}) \cap \mathcal{L}(\mathcal{P})$ and test emptiness in this class.
Many other applications (see next talk).

The construction has been extended to

- automata for infinite words: Büchi
- automata for timed words: Alur - Dill 1990

automata for signal-event words

?

Closure under intersection

is a nice property

An implementation $\mathcal{I} = \mathcal{L}(\mathcal{M})$ cannot behave badly as specified by $\mathcal{B} = \mathcal{L}(\mathcal{P})$:

$$\mathcal{L}(\mathcal{M}) \cap \mathcal{L}(\mathcal{P}) = \emptyset$$

Build a machine $\mathcal{A}$ in the same class as $\mathcal{M}$ and $\mathcal{P}$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{M}) \cap \mathcal{L}(\mathcal{P})$ and test emptiness in this class.
Many other applications (see next talk).

The construction has been extended to

- automata for infinite words: Büchi
- automata for timed words: Alur - Dill 1990

automata for signal-event words

?

Closure under intersection

is a nice property

An implementation $\mathcal{I} = \mathcal{L}(\mathcal{M})$ cannot behave badly as specified by $\mathcal{B} = \mathcal{L}(\mathcal{P})$:

$$\mathcal{L}(\mathcal{M}) \cap \mathcal{L}(\mathcal{P}) = \emptyset$$

Build a machine $\mathcal{A}$ in the same class as $\mathcal{M}$ and $\mathcal{P}$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{M}) \cap \mathcal{L}(\mathcal{P})$ and test emptiness in this class.
Many other applications (see next talk).

The construction has been extended to

- automata for infinite words: Büchi
- automata for timed words: Alur - Dill 1990

automata for signal-event words

?

Closure under intersection

is a nice property

An implementation $\mathcal{I} = \mathcal{L}(\mathcal{M})$ cannot behave badly as specified by $\mathcal{B} = \mathcal{L}(\mathcal{P})$:

$$\mathcal{L}(\mathcal{M}) \cap \mathcal{L}(\mathcal{P}) = \emptyset$$

Build a machine $\mathcal{A}$ in the same class as $\mathcal{M}$ and $\mathcal{P}$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{M}) \cap \mathcal{L}(\mathcal{P})$ and test emptiness in this class.
Many other applications (see next talk).

The construction has been extended to

- automata for infinite words: Büchi
- automata for timed words: Alur - Dill 1990

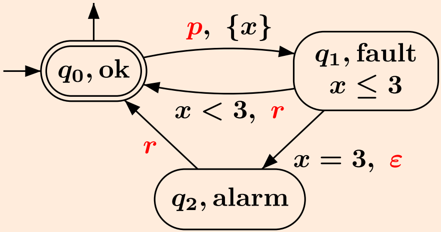
automata for signal-event words

?

Outline

Signal-Event (Timed) Automata

Asarin - Caspi - Maler 2002



x: clock

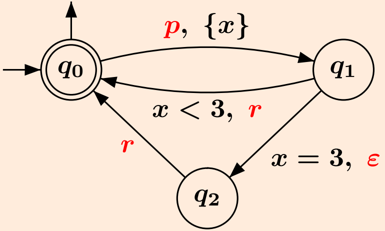
$$\begin{bmatrix} q_0 \\ 0 \end{bmatrix} \xrightarrow{8.3} \begin{bmatrix} q_0 \\ 8.2 \end{bmatrix} \xrightarrow{p} \begin{bmatrix} q_1 \\ 0 \end{bmatrix} \dots$$

Signal-event word : $\mathbf{ok}^{8.2}\ p\ \mathbf{fault}^{3}\ \mathbf{alarm}^{1.5}\ r \dots$

- States emit (possibly hidden) signals
- Transitions emit (instantaneous, possibly silent) events
- Clocks are used for time constraints

Signal-Event (Timed) Automata

Alur - Dill 1990



x: clock

$$\begin{bmatrix} q_0 \\ 0 \end{bmatrix} \xrightarrow{8.3} \begin{bmatrix} q_0 \\ 8.2 \end{bmatrix} \xrightarrow{p} \begin{bmatrix} q_1 \\ 0 \end{bmatrix} \ldots$$

time-event automata and time-event words:
$8.2\ p\ 4.5\ r \ldots$ or equivalently $(p, 8.3)(r, 12.7)\ldots$

- States emit (possibly hidden) signals
- Transitions emit (instantaneous, possibly silent) events
- Clocks are used for time constraints

Signal-Event (Timed) Words

- Σ_e finite set of (instantaneous) events
- Σ_s finite set of signals
- $\mathbb{T}$ time domain, $\overline{\mathbb{T}} = \mathbb{T} \cup \{\infty\}$
- $\Sigma = \Sigma_e \cup (\Sigma_s \times \mathbb{T})$
- Notation: a^d for $(a, d) \in \Sigma_s \times \overline{\mathbb{T}}$
- Σ^∞ : set of finite and infinite words over Σ

Signal-Event (Timed) Words

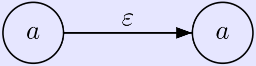
- Σ_e finite set of (instantaneous) events
- Σ_s finite set of signals
- $\mathbb{T}$ time domain, $\overline{\mathbb{T}} = \mathbb{T} \cup \{\infty\}$
- $\Sigma = \Sigma_e \cup (\Sigma_s \times \mathbb{T})$
- Notation: a^d for $(a, d) \in \Sigma_s \times \overline{\mathbb{T}}$
- Σ^∞ : set of finite and infinite words over Σ

Signal-Event (Timed) Words

- Σ_e finite set of (instantaneous) events
- Σ_s finite set of signals
- $\mathbb{T}$ time domain, $\overline{\mathbb{T}} = \mathbb{T} \cup \{\infty\}$
- $\Sigma = \Sigma_e \cup (\Sigma_s \times \mathbb{T})$
- Notation: a^d for $(a, d) \in \Sigma_s \times \overline{\mathbb{T}}$
- Σ^∞ : set of finite and infinite words over Σ

Signal stuttering

$a^2 a^3 \approx a^5$, $\qquad a^1 \approx a^{\frac{1}{2}} a^{\frac{1}{4}} a^{\frac{1}{8}} \ldots$, $\qquad a^\infty = a^2 a^2 a^2 \ldots$



Observation of signal a is not interrupted by an internal (instantaneous) action ε

Signal-Event (Timed) Words

Unobservable signal τ

- Useful to hide signals:

  Signal-event word $\xrightarrow{\text{hiding signals}}$ time-event word

  $a^3 f b^1 g f a^2 f$ $\qquad$ $\tau^3 f \tau^1 g f \tau^2 f = (f,3)(g,4)(f,4)(f,6)$

- $\tau^0 \approx \varepsilon$: a hidden signal with zero duration is not observable.
  $a^0 \not\approx \varepsilon$: a signal, even of zero duration, is observable.
  $\tau^2 \not\approx \varepsilon$: we still observe a time delay but the actual signal has been hidden.
  Example : $a^2\tau^0a^1f\tau^0g\tau^1fb^2b^2b^2\cdots \approx a^3fg\tau^1fb^\infty$

- $SE(\Sigma) = \Sigma^\infty/\approx$: signal-event words
- SEL_ε : languages accepted by SE-automata
- SEL : languages accepted by SE-automata without ε-transitions

Signal-Event (Timed) Words

Unobservable signal τ

- Useful to hide signals:

  Signal-event word $\xrightarrow{\text{hiding signals}}$ time-event word

  $a^3 f b^1 g f a^2 f$ $\tau^3 f \tau^1 g f \tau^2 f = (f,3)(g,4)(f,4)(f,6)$

- $\tau^0 \approx \varepsilon$: a hidden signal with zero duration is not observable.
  $a^0 \not\approx \varepsilon$: a signal, even of zero duration, is observable.
  $\tau^2 \not\approx \varepsilon$: we still observe a time delay but the actual signal has been hidden.
  Example : $a^2\tau^0 a^1 f \tau^0 g \tau^1 f b^2 b^2 b^2 \cdots \approx a^3 f g \tau^1 f b^\infty$

- $SE(\Sigma) = \Sigma^\infty / \approx$: signal-event words
- SEL_ε : languages accepted by SE-automata
- SEL : languages accepted by SE-automata without ε-transitions

Signal-Event (Timed) Words

Unobservable signal τ

- Useful to hide signals:

  Signal-event word $\xrightarrow{\text{hiding signals}}$ time-event word

  $a^3 f b^1 g f a^2 f$ $\qquad$ $\tau^3 f \tau^1 g f \tau^2 f = (f,3)(g,4)(f,4)(f,6)$

- $\tau^0 \approx \varepsilon$: a hidden signal with zero duration is not observable.
  $a^0 \not\approx \varepsilon$: a signal, even of zero duration, is observable.
  $\tau^2 \not\approx \varepsilon$: we still observe a time delay but the actual signal has been hidden.
  Example : $a^2\tau^0 a^1 f \tau^0 g \tau^1 f b^2 b^2 b^2 \cdots \approx a^3 f g \tau^1 f b^\infty$

- $SE(\Sigma) = \Sigma^\infty / \approx$: signal-event words
- SEL_ε : languages accepted by SE-automata
- SEL : languages accepted by SE-automata without ε-transitions

Outline

Closure under intersection

Theorem

Classes SEL and SEL_ε are closed under intersection

Remarks

- Easy for the class SEL or for time-event languages.
- More difficult with signals and ε-transitions due to stuttering and unobservability of τ^0.
- Asarin, Caspi and Maler 97 does not handle signal stuttering and considers finite runs only.
  Asarin, Caspi and Maler 02 deals with the intersection of time-event automata only.
- Dima 00 gives a construction to remove stuttering for automata with a single clock.
- Durand-Lose 04 gives a construction for intersection taking stuttering into account but restricted to finite runs and without zero-duration signals.
  His approach does not extend to infinite runs since it would introduce Zeno runs leading to transfinite problems.

Closure under intersection

Theorem

Classes SEL and SEL_ε are closed under intersection

Remarks

- Easy for the class SEL or for time-event languages.
- More difficult with signals and ε-transitions due to stuttering and unobservability of τ^0.
- Asarin, Caspi and Maler 97 does not handle signal stuttering and considers finite runs only.
  Asarin, Caspi and Maler 02 deals with the intersection of time-event automata only.
- Dima 00 gives a construction to remove stuttering for automata with a single clock.
- Durand-Lose 04 gives a construction for intersection taking stuttering into account but restricted to finite runs and without zero-duration signals.
  His approach does not extend to infinite runs since it would introduce Zeno runs leading to transfinite problems.

Closure under intersection

Theorem

Classes SEL and SEL_ε are closed under intersection

Remarks

- Easy for the class SEL or for time-event languages.
- More difficult with signals and ε-transitions due to stuttering and unobservability of τ^0.
- Asarin, Caspi and Maler 97 does not handle signal stuttering and considers finite runs only.
  Asarin, Caspi and Maler 02 deals with the intersection of time-event automata only.
- Dima 00 gives a construction to remove stuttering for automata with a single clock.
- Durand-Lose 04 gives a construction for intersection taking stuttering into account but restricted to finite runs and without zero-duration signals.
  His approach does not extend to infinite runs since it would introduce Zeno runs leading to transfinite problems.

Closure under intersection

Theorem

Classes SEL and SEL_ε are closed under intersection

Remarks

- Easy for the class SEL or for time-event languages.
- More difficult with signals and ε-transitions due to stuttering and unobservability of τ^0.
- Asarin, Caspi and Maler 97 does not handle signal stuttering and considers finite runs only.
  Asarin, Caspi and Maler 02 deals with the intersection of time-event automata only.
- Dima 00 gives a construction to remove stuttering for automata with a single clock.
- Durand-Lose 04 gives a construction for intersection taking stuttering into account but restricted to finite runs and without zero-duration signals.
  His approach does not extend to infinite runs since it would introduce Zeno runs leading to transfinite problems.

Closure under intersection

Theorem

Classes SEL and SEL_ε are closed under intersection

Remarks

- Easy for the class SEL or for time-event languages.
- More difficult with signals and ε-transitions due to stuttering and unobservability of τ^0.
- Asarin, Caspi and Maler 97 does not handle signal stuttering and considers finite runs only.
  Asarin, Caspi and Maler 02 deals with the intersection of time-event automata only.
- Dima 00 gives a construction to remove stuttering for automata with a single clock.
- Durand-Lose 04 gives a construction for intersection taking stuttering into account but restricted to finite runs and without zero-duration signals.
  His approach does not extend to infinite runs since it would introduce Zeno runs leading to transfinite problems.

Closure under intersection

Theorem

Classes SEL and SEL_ε are closed under intersection

Remarks

- Easy for the class SEL or for time-event languages.
- More difficult with signals and ε-transitions due to stuttering and unobservability of τ^0.
- Asarin, Caspi and Maler 97 does not handle signal stuttering and considers finite runs only.
  Asarin, Caspi and Maler 02 deals with the intersection of time-event automata only.
- Dima 00 gives a construction to remove stuttering for automata with a single clock.
- Durand-Lose 04 gives a construction for intersection taking stuttering into account but restricted to finite runs and without zero-duration signals.
  His approach does not extend to infinite runs since it would introduce Zeno runs leading to transfinite problems.

Closure under intersection

Theorem

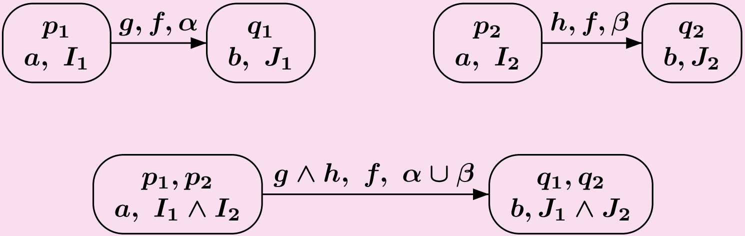
Classes SEL and SEL_ε are closed under intersection

Closure under intersection

Theorem

Classes SEL and SEL_ε are closed under intersection

Basic technique for SEL or time-event words

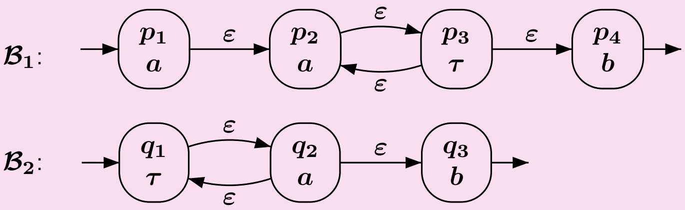


Closure under intersection

Theorem

SEL_ε is closed under intersection

Problem 1 : stuttering with unobservability of τ^0

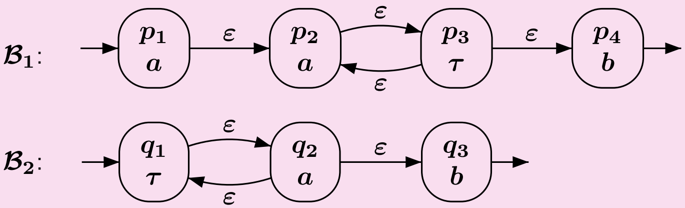


Closure under intersection

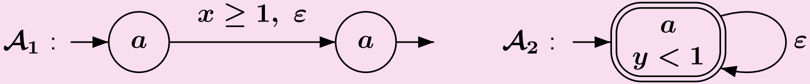
Theorem

SEL_ε is closed under intersection

Problem 1 : stuttering with unobservability of τ^0



Problem 2 : finite and infinite runs

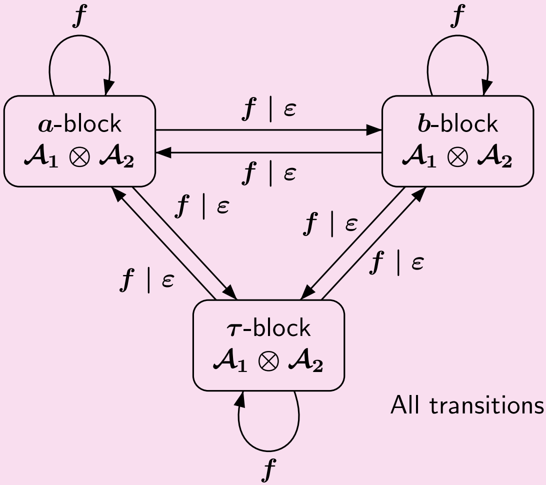


$\mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2) = \{a^1\}$

$a^1 \approx a^{\frac{1}{2}} a^{\frac{1}{4}} a^{\frac{1}{8}} \dots$

Stuttering with unobservability of τ^0

Connecting modules for a-blocks with synchronous transitions

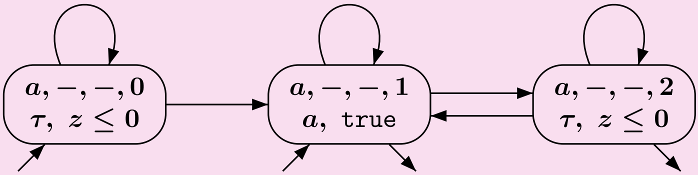


All transitions reset a new clock z

Stuttering with unobservability of τ^0

Building maximal a-blocks

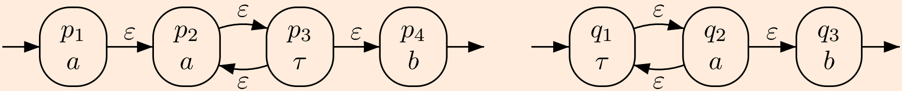
States : (a, p, q, i), where i is the synchronization mode.

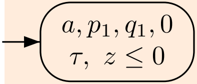


with $a \neq \tau$ and asynchronous ε-transitions that reset clock z.

Example

The a-block for $\mathcal{B}_1$ and $\mathcal{B}_2$

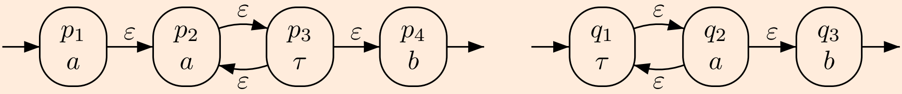


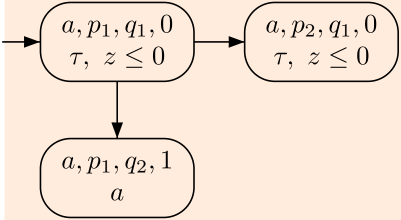


To be completed with a τ-block and a b-block.

Example

The a-block for $\mathcal{B}_1$ and $\mathcal{B}_2$

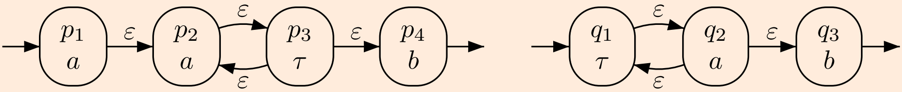


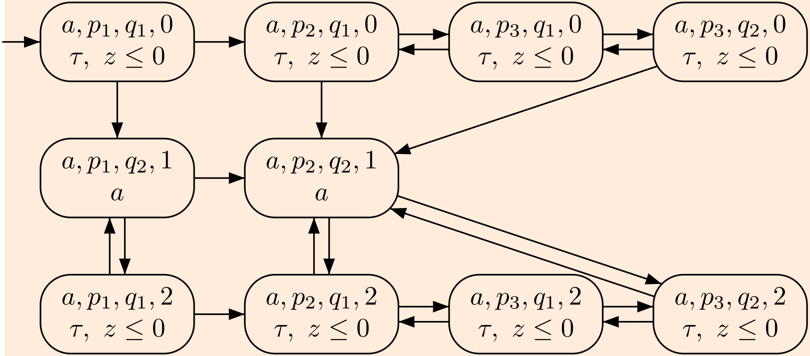


To be completed with a τ-block and a b-block.

Example

The a-block for $\mathcal{B}_1$ and $\mathcal{B}_2$

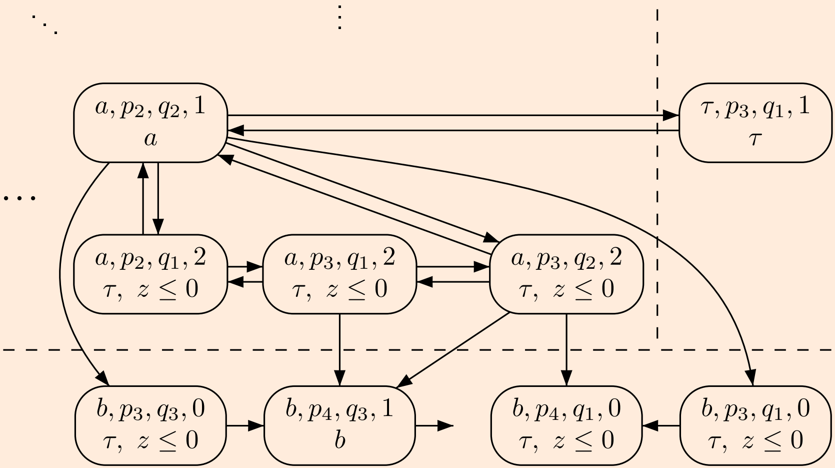




To be completed with a τ-block and a b-block.

Example (cont.)

Connecting the blocks



Finite and infinite runs

Theorem : a normal form for SE-automata

Let $\mathcal{A}$ be a SE-automaton. We can effectively construct an equivalent SE-automaton $\mathcal{A}'$ such that:

1. no infinite run of $\mathcal{A}'$ accepts a finite word with finite duration, and
2. no finite run of $\mathcal{A}'$ accepts a word with infinite duration.

Remarks

- The construction removes Zeno runs accepting finite runs with finite duration: replacing for instance an infinite ε-loop producing $a^{\frac{1}{2}}$, $a^{\frac{1}{4}}$, $a^{\frac{1}{8}}$ ... by a finite run producing a^1.
- Easy if Zeno runs or ε-transitions are forbidden.
- The result is interesting in itself to obtain a more realistic implementation of an arbitrary SE-automaton.

Finite and infinite runs

Theorem : a normal form for SE-automata

Let $\mathcal{A}$ be a SE-automaton. We can effectively construct an equivalent SE-automaton $\mathcal{A}'$ such that:

1. no infinite run of $\mathcal{A}'$ accepts a finite word with finite duration, and
2. no finite run of $\mathcal{A}'$ accepts a word with infinite duration.

Remarks

- The construction removes Zeno runs accepting finite runs with finite duration: replacing for instance an infinite ε-loop producing $a^{\frac{1}{2}}, a^{\frac{1}{4}}, a^{\frac{1}{8}} \ldots$ by a finite run producing a^1.
- Easy if Zeno runs or ε-transitions are forbidden.
- The result is interesting in itself to obtain a more realistic implementation of an arbitrary SE-automaton.

Finite and infinite runs

Theorem : a normal form for SE-automata

Let $\mathcal{A}$ be a SE-automaton. We can effectively construct an equivalent SE-automaton $\mathcal{A}'$ such that:

1. no infinite run of $\mathcal{A}'$ accepts a finite word with finite duration, and
2. no finite run of $\mathcal{A}'$ accepts a word with infinite duration.

Remarks

- The construction removes Zeno runs accepting finite runs with finite duration: replacing for instance an infinite ε-loop producing $a^{\frac{1}{2}}$, $a^{\frac{1}{4}}$, $a^{\frac{1}{8}}$ ... by a finite run producing a^1.
- Easy if Zeno runs or ε-transitions are forbidden.
- The result is interesting in itself to obtain a more realistic implementation of an arbitrary SE-automaton.

Finite and infinite runs

Theorem : a normal form for SE-automata

Let $\mathcal{A}$ be a SE-automaton. We can effectively construct an equivalent SE-automaton $\mathcal{A}'$ such that:

1. no infinite run of $\mathcal{A}'$ accepts a finite word with finite duration, and
2. no finite run of $\mathcal{A}'$ accepts a word with infinite duration.

Remarks

- The construction removes Zeno runs accepting finite runs with finite duration: replacing for instance an infinite ε-loop producing $a^{\frac{1}{2}}$, $a^{\frac{1}{4}}$, $a^{\frac{1}{8}}$ ... by a finite run producing a^1.
- Easy if Zeno runs or ε-transitions are forbidden.
- The result is interesting in itself to obtain a more realistic implementation of an arbitrary SE-automaton.

Finite and infinite runs

Theorem: a normal form for SE-automata

Let $\mathcal{A}$ be a SE-automaton. We can effectively construct an equivalent SE-automaton $\mathcal{A}'$ such that:

1. no infinite run of $\mathcal{A}'$ accepts a finite word with finite duration, and
2. no finite run of $\mathcal{A}'$ accepts a word with infinite duration.

Main problem

We have to replace infinite accepting ε-loops

by finite accepting runs.

Finite and infinite runs

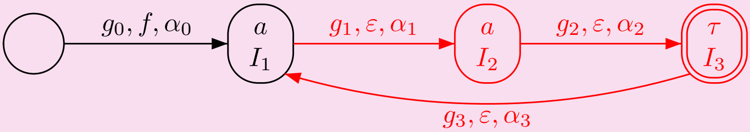
Theorem: a normal form for SE-automata

Let $\mathcal{A}$ be a SE-automaton. We can effectively construct an equivalent SE-automaton $\mathcal{A}'$ such that:

1. no infinite run of $\mathcal{A}'$ accepts a finite word with finite duration, and
2. no finite run of $\mathcal{A}'$ accepts a word with infinite duration.

Main problem

We have to replace infinite accepting ε-loops



by finite accepting runs.

Finite and infinite runs

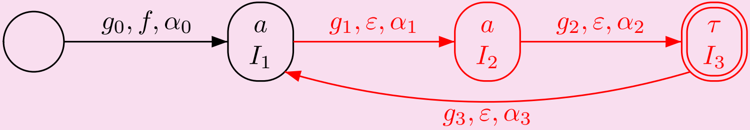
Theorem: a normal form for SE-automata

Let $\mathcal{A}$ be a SE-automaton. We can effectively construct an equivalent SE-automaton $\mathcal{A}'$ such that:

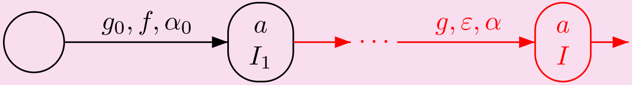
1. no infinite run of $\mathcal{A}'$ accepts a finite word with finite duration, and
2. no finite run of $\mathcal{A}'$ accepts a word with infinite duration.

Main problem

We have to replace infinite accepting ε-loops

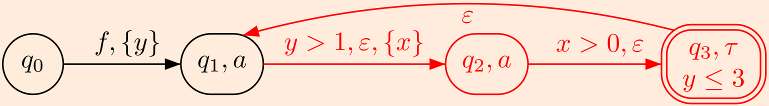


by finite accepting runs.



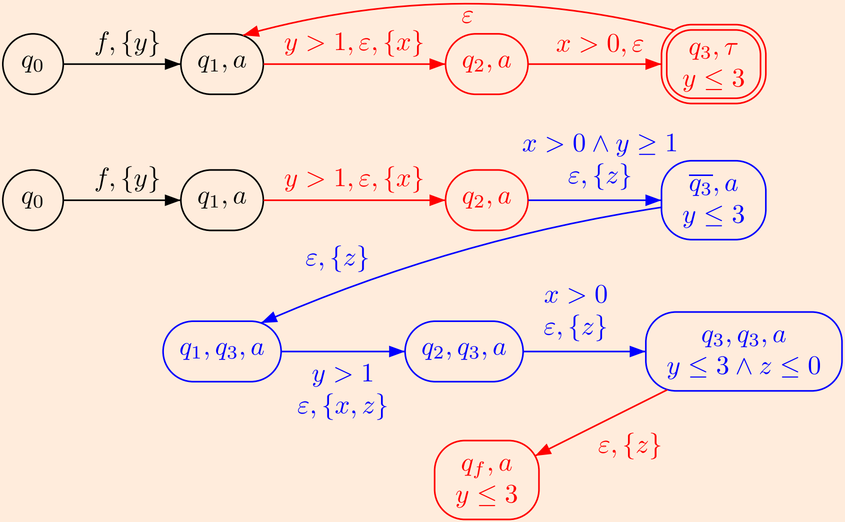
Example

Simulating the loop



Example

Simulating the loop



Outline

Conclusion

- Extending classical results to SE-automata is not always easy due to ε-transitions, signal stuttering, unobservability of τ^0, Zeno runs, ...
- We have proved closure under intersection for the general case of languages accepted by SE-automata.
- Signal-event words are natural objects for studying refinements and abstractions, see next talk.