Refinements and Abstractions of Signal-Event (Timed) Languages

Paul Gastin

LSV
ENS de Cachan & CNRS
Paul.Gastin@lsv.ens-cachan.fr

Joint work with Béatrice Bérard and Antoine Petit

FSTTCS Workshop, Dec. 12th, 2006

Outline

1. Introduction

Signal-Event (Timed) Words and Automata

Signal-Event (Timed) Substitutions

Recognizable substitutions

Intersection

Conclusion

Refinements and Abstractions

Abstract level | Concrete level

ConnectToServer $\xrightarrow{\text{refinement}}$ Details used to establish the connection

ConnectToServer $\xleftarrow{\text{abstraction}}$ Details used to establish the connection

Formalisation of refinement

Let $\sigma : A \to \mathcal{P}(B^*)$ be a substitution.

Abstract level		Concrete level
Action $a \in A$	$\xrightarrow{\text{refinement}}$	$\sigma(a) \subseteq B^*$
Behavior $w = abaac \in A^*$	$\xrightarrow{\text{refinement}}$	$\sigma(w) = \sigma(a)\sigma(b)\sigma(a)\sigma(a)\sigma(c) \subseteq B^*$
Language $K \subseteq A^*$	$\xrightarrow{\text{refinement}}$	$\sigma(K) = \bigcup_{w \in K} \sigma(w) \subseteq B^*$

Refinements and Abstractions

Abstract level		Concrete level
ConnectToServer	$\xrightarrow{\text{refinement}}$ $\xleftarrow{\text{abstraction}}$	Details used to establish the connection

Formalisation of refinement

Let $\sigma : A \to \mathcal{P}(B^*)$ be a substitution.

Abstract level		Concrete level
Action $a \in A$	$\xrightarrow{\text{refinement}}$	$\sigma(a) \subseteq B^*$
Behavior $w = abaac \in A^*$	$\xrightarrow{\text{refinement}}$	$\sigma(w) = \sigma(a)\sigma(b)\sigma(a)\sigma(a)\sigma(c) \subseteq B^*$
Language $K \subseteq A^*$	$\xrightarrow{\text{refinement}}$	$\sigma(K) = \bigcup_{w \in K} \sigma(w) \subseteq B^*$

Refinements and Abstractions

Abstract level		Concrete level
ConnectToServer	$\xrightarrow{\text{refinement}}$ $\xleftarrow{\text{abstraction}}$	Details used to establish the connection

Formalisation of abstraction

Let $\sigma : A \to \mathcal{P}(B^*)$ be a substitution.

Abstract level		Concrete level
$\sigma^{-1}(L) = \{w \in A^* \mid \sigma(w) \cap L \neq \emptyset\}$	$\xleftarrow{\text{abstraction}}$	$L \subseteq B^*$

Adding time to the picture

Timed refinement

Abstract level	$\xrightarrow{\text{refinement}}$ $\xleftarrow{\text{abstraction}}$	Concrete level
$\mathsf{ConnectToServer}^2$		$\mathsf{Req} \cdot \mathsf{Wait}^2 \cdot \mathsf{Ack}$
$\mathsf{ConnectToServer}^{4.5}$		$\mathsf{Req} \cdot \mathsf{Wait}^1 \cdot \mathsf{Nack} \cdot \mathsf{Wait}^{0.5} \cdot \mathsf{Retry} \cdot \mathsf{Wait}^3 \cdot \mathsf{Ack}$

An abstract action a with duration d should be replaced by a concrete execution (word) w with the same duration $\|w\| = d$.

Adding time to the picture

Timed refinement

Abstract level $\xrightarrow{\text{refinement}}$ Concrete level

Abstract level $\xleftarrow{\text{abstraction}}$ Concrete level

$\mathsf{ConnectToServer}^{2}$	$\mathsf{Req} \cdot \mathsf{Wait}^{2} \cdot \mathsf{Ack}$
$\mathsf{ConnectToServer}^{4.5}$	$\mathsf{Req} \cdot \mathsf{Wait}^{1} \cdot \mathsf{Nack} \cdot \mathsf{Wait}^{0.5} \cdot \mathsf{Retry} \cdot \mathsf{Wait}^{3} \cdot \mathsf{Ack}$

An abstract action a with duration d should be replaced by a concrete execution (word) w with the same duration $\|w\| = d$.

Outline

Signal-Event (Timed) Words

Asarin - Caspi - Maler 2002

- 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 signal-event (timed) words
  Example: $a^3 f f g b^{1.5} a^2 f$
- Signal stuttering: $a^2 a^3 \approx a^5$, $\quad a^\infty = a^2 a^2 a^2 \cdots$,
  $a^1 = a^{\frac{1}{2}} + a^{\frac{1}{4}} + a^{\frac{1}{8}} + \dots$

Signal-Event (Timed) Words

Asarin - Caspi - Maler 2002

- 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 signal-event (timed) words
  Example: $a^3 f f g b^{1.5} a^2 f$
- Signal stuttering: $a^2 a^3 \approx a^5$, $a^\infty = a^2 a^2 a^2 \cdots$,
  $a^1 = a^{\frac{1}{2}} + a^{\frac{1}{4}} + a^{\frac{1}{8}} + \ldots$

Signal-Event (Timed) Words

Asarin - Caspi - Maler 2002

- 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 signal-event (timed) words
  Example: $a^3 f f g b^{1.5} a^2 f$
- Signal stuttering: $a^2 a^3 \approx a^5$, $\qquad a^\infty = a^2 a^2 a^2 \cdots$,
  $a^1 = a^{\frac{1}{2}} + a^{\frac{1}{4}} + a^{\frac{1}{8}} + \ldots$

Signal-Event (Timed) Words

Unobservable signal τ

- Useful to hide signals:

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

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

- $\tau^0 \approx \varepsilon$: an 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 : $a^2 \tau^0 a^1 f \tau^0 g \tau^1 f b^2 b^2 b^2 \cdots \approx a^3 f g \tau^1 f b^\infty$
- Signal-event words $SE(\Sigma) = \Sigma^\infty / \approx$

Signal-Event (Timed) Words

Unobservable signal τ

- Useful to hide signals:

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

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

- $\tau^0 \approx \varepsilon$: an 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 : $a^2\tau^0 a^1 f \tau^0 g \tau^1 f b^2 b^2 b^2 \cdots \approx a^3 f g \tau^1 f b^\infty$
- Signal-event words $SE(\Sigma) = \Sigma^\infty / \approx$

Signal-Event (Timed) Words

Unobservable signal τ

- Useful to hide signals:

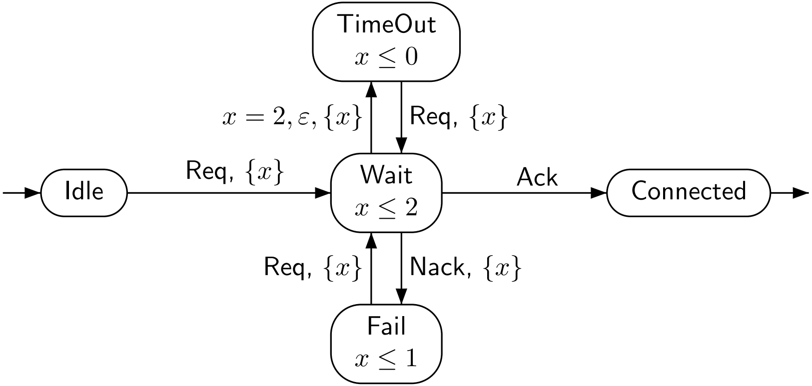
  Signal-event word $\xrightarrow{\text{hiding signals}}$ Classical time-event words

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

- $\tau^0 \approx \varepsilon$: an 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 : $a^2\tau^0 a^1 f \tau^0 g \tau^1 f b^2 b^2 b^2 \cdots \approx a^3 f g \tau^1 f b^\infty$
- Signal-event words $SE(\Sigma) = \Sigma^\infty / \approx$

Signal-Event (Timed) automata

- States emit signals
- Transitions emit (instantaneous) events



- Run : Idle^3 · Req · Wait^2 · TimeOut^0 · Req · Wait^1 · Ack · Connected^8
- SEL : languages accepted by SE-automata without ε-transitions.
- SEL_ε : languages accepted by SE-automata with ε-transitions.

Outline

Signal-Event (Timed) Substitutions

Definition

- Abstract alphabet : Σ_e and Σ_s
- Concrete alphabet : Σ'_e and Σ'_s
- Substitution σ from $SE(\Sigma)$ to $SE(\Sigma')$ defined by:

$$a \in \Sigma_e : \ L_a \subseteq (\Sigma'_e \cup \Sigma'_s \times \{0\})^*$$

$$\sigma(a) = L_a$$

$a \in \Sigma_s \setminus \{\tau\} : \ L_a \subseteq SE(\Sigma')$ not containing Zeno words.

$$\sigma(a^d) = \{w \in L_a \mid \|w\| = d\}$$

$$a = \tau : \ L_\tau = \{\tau\} \times \overline{\mathbb{T}}$$

$$\sigma(\tau^d) = \{\tau^d\}$$

Remark

If we allow Zeno words in L_a then we may get transfinite words as refinements. Example: if $b^1 f b^{1/2} f b^{1/4} f \cdots \in L_a$ and $L_g = \{g\}$ then $\sigma(a^2 g)$ is transfinite.

Signal-Event (Timed) Substitutions

Definition

- Abstract alphabet : Σ_e and Σ_s
- Concrete alphabet : Σ'_e and Σ'_s
- Substitution σ from $SE(\Sigma)$ to $SE(\Sigma')$ defined by:

$$
\begin{aligned}
a \in \Sigma_e \; : \;& L_a \subseteq (\Sigma'_e \cup \Sigma'_s \times \{0\})^* \\
& \sigma(a) = L_a \\
a \in \Sigma_s \setminus \{\tau\} \; : \;& L_a \subseteq SE(\Sigma') \text{ not containing Zeno words.} \\
& \sigma(a^d) = \{w \in L_a \mid \|w\| = d\} \\
a = \tau \; : \;& L_\tau = \{\tau\} \times \overline{\mathbb{T}} \\
& \sigma(\tau^d) = \{\tau^d\}
\end{aligned}
$$

Remark

If we allow Zeno words in L_a then we may get transfinite words as refinements.
Example: if $b^1 f b^{1/2} f b^{1/4} f \cdots \in L_a$ and $L_g = \{g\}$ then $\sigma(a^2 g)$ is transfinite.

Signal-Event (Timed) Substitutions

Remark

In general, SE-substitutions are not morphisms

Example: if $L_a = \{b^2\}$ then $\sigma(a^1) = \emptyset$ and $\sigma(a^2) \neq \sigma(a^1)\sigma(a^1)$

Substitutions are applied to SE-words in normal form:

$\sigma(a^2\tau^0 a^1 f\tau^0 g\tau^1 f b^2 b^2 b^2 \cdots) = \sigma(a^3)\sigma(f)\sigma(g)\tau^1\sigma(f)\sigma(b^\infty)$

Proposition

Let σ be a timed substitution, given by a family $(L_a)_{a\in\Sigma_e\cup\Sigma_s}$.
Then, σ is a morphism if and only if for each signal $a \in \Sigma_s$ we have

1. L_a is closed under concatenation:
   for all $u, v \in L_a$ with $\|u\| < \infty$, we have $uv \in L_a$,
2. L_a is closed under decomposition:
   for each $v \in L_a$ with $\|v\| = d$, for all $d_1 \in \mathbb{T}$, $d_2 \in \overline{\mathbb{T}}$ such that $d = d_1 + d_2$, there exist $v_i \in L_a$ with $\|v_i\| = d_i$ such that $v = v_1 v_2$.

Signal-Event (Timed) Substitutions

Remark

In general, SE-substitutions are not morphisms

Example: if $L_a = \{b^2\}$ then $\sigma(a^1) = \emptyset$ and $\sigma(a^2) \neq \sigma(a^1)\sigma(a^1)$

Substitutions are applied to SE-words in normal form:

$\sigma(a^2\tau^0 a^1 f \tau^0 g \tau^1 f b^2 b^2 b^2 \cdots) = \sigma(a^3)\sigma(f)\sigma(g)\tau^1\sigma(f)\sigma(b^\infty)$

Proposition

Let σ be a timed substitution, given by a family $(L_a)_{a\in\Sigma_e\cup\Sigma_s}$.
Then, σ is a morphism if and only if for each signal $a \in \Sigma_s$ we have

1. L_a is closed under concatenation:
   for all $u, v \in L_a$ with $\|u\| < \infty$, we have $uv \in L_a$,
2. L_a is closed under decomposition:
   for each $v \in L_a$ with $\|v\| = d$, for all $d_1 \in \mathbb{T}$, $d_2 \in \overline{\mathbb{T}}$ such that $d = d_1 + d_2$, there exist $v_i \in L_a$ with $\|v_i\| = d_i$ such that $v = v_1 v_2$.

Outline

Recognizable substitutions

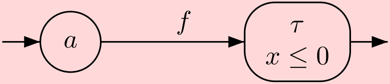
Definition

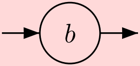
Let σ be a substitution defined by $(L_a)_{a \in \Sigma_e \cup \Sigma_s}$. Then,

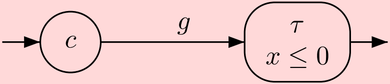
- σ is a SEL-substitution if each L_a is in SEL
- σ is a SEL_ε-substitution if each L_a is in SEL_ε

Closure under SEL-substitutions

SEL is not closed under SEL-substitutions

- $L = \{a^0 f\}$ is recognized by 

- $L_a = \{b\} \times \overline{\mathbb{T}}$ is recognized by 

- $L_f = \{c^0 g\}$ is recognized by 

- $\sigma(L) = \{b^0 c^0 g\}$ cannot be accepted without ε-transitions.

Theorem

The class SEL is closed under SEL-substitutions satisfying for each $f \in \Sigma_e$

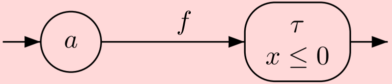
$$L_f \subseteq \Sigma_e'((\Sigma_s' \times \{0\})\Sigma_e')^*$$

i.e., each word in L_f must start and end with an instantaneous event.

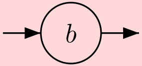
Closure under SEL-substitutions

SEL is not closed under SEL-substitutions

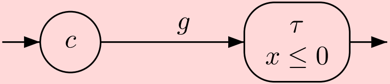
- $L = \{a^0 f\}$ is recognized by



- $L_a = \{b\} \times \overline{\mathbb{T}}$ is recognized by



- $L_f = \{c^0 g\}$ is recognized by



- $\sigma(L) = \{b^0 c^0 g\}$ cannot be accepted without ε-transitions.

Theorem

The class SEL is closed under SEL-substitutions satisfying for each $f \in \Sigma_e$

$$L_f \subseteq \Sigma'_e((\Sigma'_s \times \{0\})\Sigma'_e)^*$$

i.e., each word in L_f must start and end with an instantaneous event.

Closure under SEL-substitutions

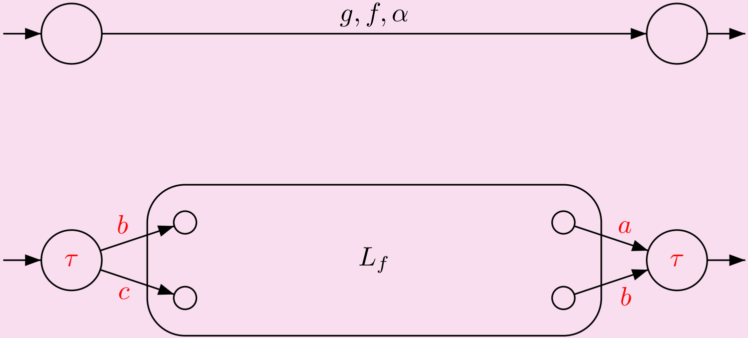
Theorem

The class SEL is closed under SEL-substitutions satisfying for each $f \in \Sigma_e$

$$L_f \subseteq \Sigma'_e((\Sigma'_s \times \{0\})\Sigma'_e)^*$$

i.e., each word in L_f must start and end with an instantaneous event.

Handling events is easy for SEL-substitutions.



Closure under SEL-substitutions

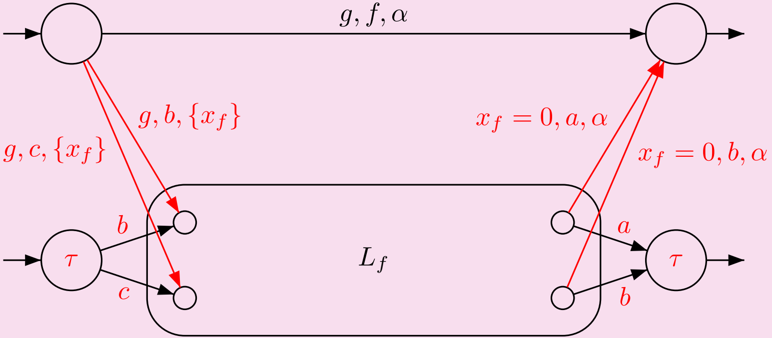
Theorem

The class SEL is closed under SEL-substitutions satisfying for each $f \in \Sigma_e$

$$L_f \subseteq \Sigma'_e((\Sigma'_s \times \{0\})\Sigma'_e)^*$$

i.e., each word in L_f must start and end with an instantaneous event.

Handling events is easy for SEL-substitutions.



Closure under SEL-substitutions

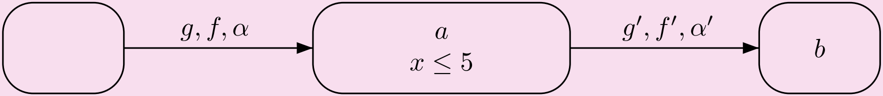
Theorem

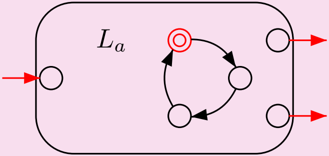
The class SEL is closed under SEL-substitutions satisfying for each $f \in \Sigma_e$

$$L_f \subseteq \Sigma'_e((\Sigma'_s \times \{0\})\Sigma'_e)^*$$

i.e., each word in L_f must start and end with an instantaneous event.

Handling signals is easy for SEL-substitutions.





Closure under SEL-substitutions

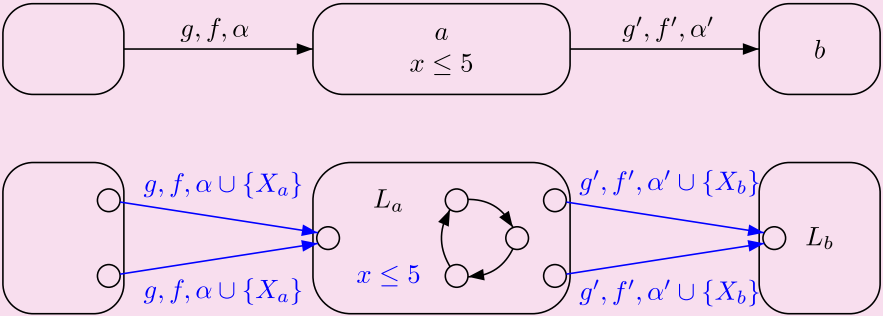
Theorem

The class SEL is closed under SEL-substitutions satisfying for each $f \in \Sigma_e$

$$L_f \subseteq \Sigma'_e((\Sigma'_s \times \{0\})\Sigma'_e)^*$$

i.e., each word in L_f must start and end with an instantaneous event.

Handling signals is easy for SEL-substitutions.



Closure under SEL-substitutions

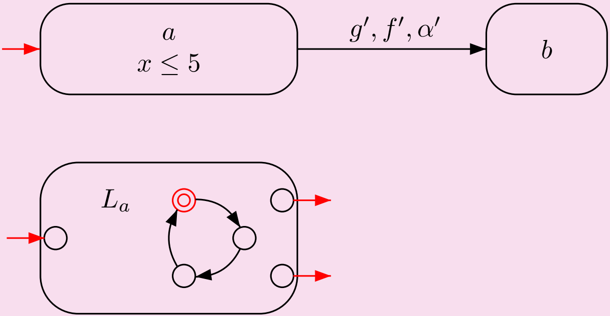
Theorem

The class SEL is closed under SEL-substitutions satisfying for each $f \in \Sigma_e$

$$L_f \subseteq \Sigma'_e((\Sigma'_s \times \{0\})\Sigma'_e)^*$$

i.e., each word in L_f must start and end with an instantaneous event.

Handling signals is easy for SEL-substitutions.



Closure under SEL-substitutions

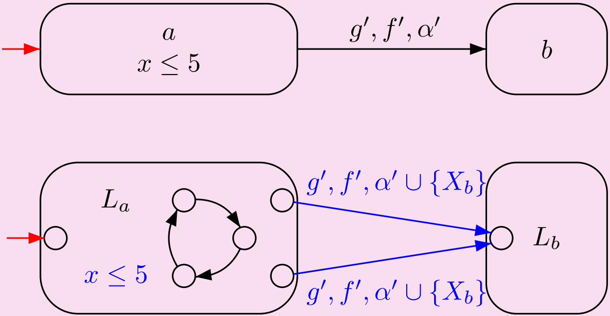
Theorem

The class SEL is closed under SEL-substitutions satisfying for each $f \in \Sigma_e$

$$L_f \subseteq \Sigma'_e((\Sigma'_s \times \{0\})\Sigma'_e)^*$$

i.e., each word in L_f must start and end with an instantaneous event.

Handling signals is easy for SEL-substitutions.



Closure under SEL-substitutions

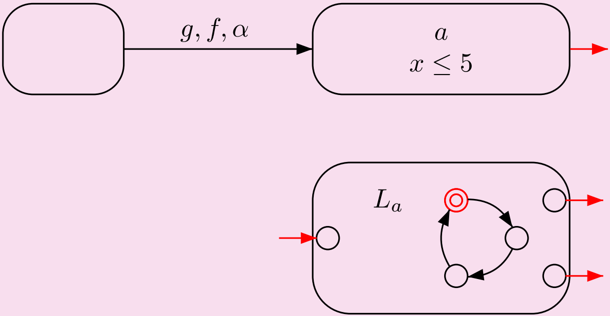
Theorem

The class SEL is closed under SEL-substitutions satisfying for each $f \in \Sigma_e$

$$L_f \subseteq \Sigma'_e((\Sigma'_s \times \{0\})\Sigma'_e)^*$$

i.e., each word in L_f must start and end with an instantaneous event.

Handling signals is easy for SEL-substitutions.



Closure under SEL-substitutions

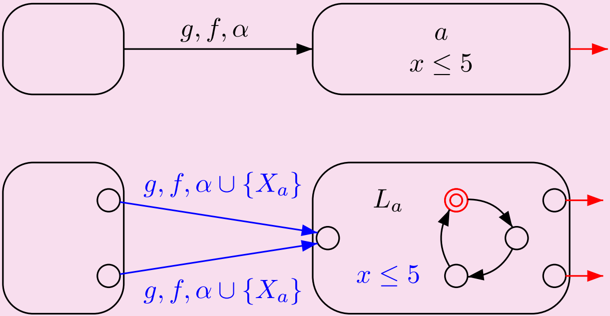
Theorem

The class SEL is closed under SEL-substitutions satisfying for each $f \in \Sigma_e$

$$L_f \subseteq \Sigma'_e((\Sigma'_s \times \{0\})\Sigma'_e)^*$$

i.e., each word in L_f must start and end with an instantaneous event.

Handling signals is easy for SEL-substitutions.



Closure under SEL-substitutions

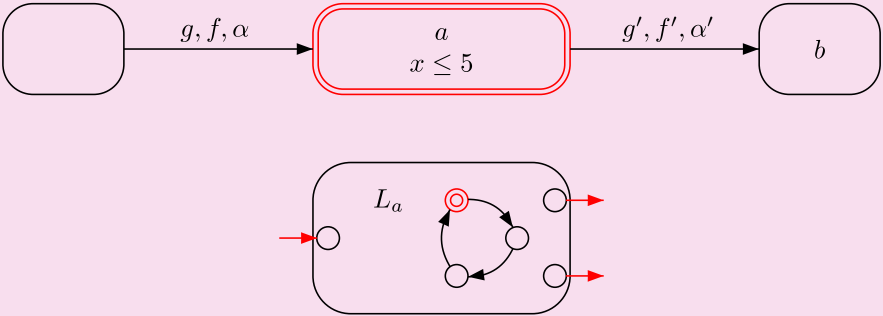
Theorem

The class SEL is closed under SEL-substitutions satisfying for each $f \in \Sigma_e$

$$L_f \subseteq \Sigma'_e((\Sigma'_s \times \{0\})\Sigma'_e)^*$$

i.e., each word in L_f must start and end with an instantaneous event.

Handling signals is easy for SEL-substitutions.



Closure under SEL-substitutions

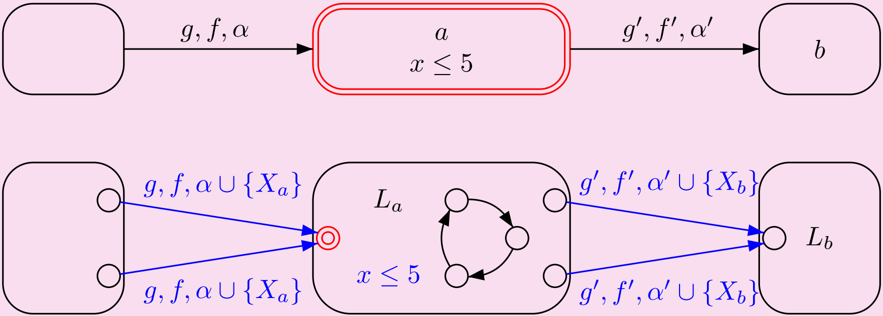
Theorem

The class SEL is closed under SEL-substitutions satisfying for each $f \in \Sigma_e$

$$L_f \subseteq \Sigma'_e((\Sigma'_s \times \{0\})\Sigma'_e)^*$$

i.e., each word in L_f must start and end with an instantaneous event.

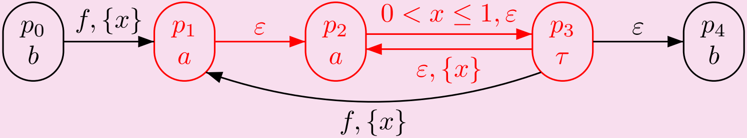
Handling signals is easy for SEL-substitutions.



Closure under SEL_ε-substitutions

Handling signals for SEL_ε-substitutions is harder.

Remember that substitutions are applied to SE-words in normal form.



A possible run gives : $f a^{0.3} a^{0.6} \tau^0 a^{0.5} \tau^1 a^{0.6} \tau^0 a^{0.5} \tau^0 b^3 \quad \approx \quad f a^{1.4} \tau^1 a^{1.1} b^3$

We cannot simply replace each a-labelled state by a copy of $\mathcal{A}_a$.

Closure under substitutions

Proof technique inspired from the word case

- Let $\sigma : A \to \mathcal{P}(B^*)$ be a rational substitution
- Let $\Pi_A : (A \uplus B)^* \to A^*$ and $\Pi_B : (A \uplus B)^* \to B^*$ be the projections
- Let $M = \left(\bigcup_{a \in A} a\sigma(a) \right)^* \subseteq (A \uplus B)^*$ is rational.
- Then, $\sigma(L) = \Pi_B(\Pi_A^{-1}(L) \cap M)$.

$$\begin{tikzcd} {} \arrow[r, "\cap M"] & {} \arrow[d, "\Pi_B"] \\ L \arrow[u, "\Pi_A^{-1}"] \arrow[r, "\sigma"] & \sigma(L) \end{tikzcd}$$

- This proof technique also applies to inverse substitutions: $\sigma^{-1}(L) = \Pi_A(\Pi_B^{-1}(L) \cap M)$.

Closure under substitutions

Proof technique inspired from the word case

- Let $\sigma : A \to \mathcal{P}(B^*)$ be a rational substitution
- Let $\Pi_A : (A \uplus B)^* \to A^*$ and $\Pi_B : (A \uplus B)^* \to B^*$ be the projections
- Let $M = \left(\bigcup_{a \in A} a\sigma(a) \right)^* \subseteq (A \uplus B)^*$ is rational.
- Then, $\sigma(L) = \Pi_B(\Pi_A^{-1}(L) \cap M)$.

$$\begin{tikzcd}
{} \arrow[r, "\cap M"] & {} \arrow[d, "\Pi_B"] \\
L \arrow[u, "\Pi_A^{-1}"] \arrow[r, "\sigma"] & \sigma(L)
\end{tikzcd}$$

- This proof technique also applies to inverse substitutions: $\sigma^{-1}(L) = \Pi_A(\Pi_B^{-1}(L) \cap M)$.

Closure under SEL_ε-substitutions

Theorem

The class SEL_ε is closed under SEL_ε-substitutions and inverse SEL_ε-substitutions.

Proof: Signal-event words

- Let $\hat{\Sigma}_e = \Sigma_e \uplus \Sigma'_e$ and $\hat{\Sigma}_s = \Sigma_s \times \Sigma'_s$.
- Let $\pi_1 : SE(\hat{\Sigma}) \to SE(\Sigma)$ and $\pi_2 : SE(\hat{\Sigma}) \to SE(\Sigma')$ be the natural projections defined by

  $\pi_1(f) = f$ and $\pi_2(f) = \varepsilon$ if $f \in \Sigma_e$,
  $\pi_1(f) = \varepsilon$ and $\pi_2(f) = f$ if $f \in \Sigma'_e$,
  $\pi_1((a,b)^d) = a^d$ and $\pi_2((a,b)^d) = b^d$ if $(a,b)^d \in \Sigma_s \times \Sigma'_s \times \overline{\mathbb{T}}$.

- We will show that for a suitable SEL_ε-language M we have

$$\sigma(L) = \pi_2(\pi_1^{-1}(L) \cap M)$$
$$\sigma^{-1}(L) = \pi_1(\pi_2^{-1}(L) \cap M)$$

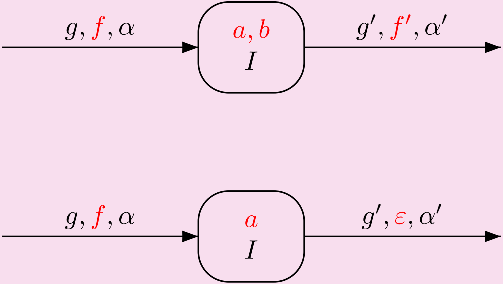
- The class SEL_ε is closed under projection, inverse projection and intersection.

Closure under SEL_ε-substitutions

Lemma

If L is in the class SEL_ε, then so is $\pi_1(L)$.

Proof

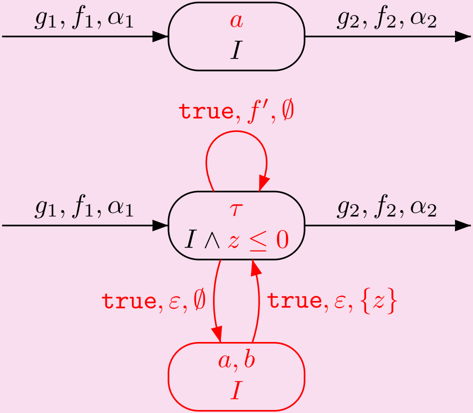


Closure under SEL_ε-substitutions

Lemma

If L is in the class SEL_ε, then so is $\pi_1^{-1}(L)$.

Proof



Closure under SEL_ε-substitutions

Lemma Words: $M = \left(\bigcup_{a\in A} a\sigma(a)\right)^*$

If $M \subseteq SE(\hat{\Sigma})$ satisfies

1. $\pi_2(w) \in \sigma(\pi_1(w))$ for each $w \in M$,
2. $\forall u \in SE(\Sigma)$, $\forall v \in \sigma(u)$, $\exists w \in M$ such that $u = \pi_1(w)$ and $v = \pi_2(w)$.

Then,

- for $L \subseteq SE(\Sigma)$, we have $\sigma(L) = \pi_2(\pi_1^{-1}(L) \cap M)$,
- for $L \subseteq SE(\Sigma')$, we have $\sigma^{-1}(L) = \pi_1(\pi_2^{-1}(L) \cap M)$.

Proof

- $\sigma(L) \subseteq \pi_2(\pi_1^{-1}(L) \cap M)$:
  Let $v \in \sigma(L)$ and let $u \in L$ with $v \in \sigma(u)$.
  From 2, $\exists w \in M$ with $\pi_1(w) = u$ and $\pi_2(w) = v$.
  Then, $w \in \pi_1^{-1}(L) \cap M$ and $v \in \pi_2(\pi_1^{-1}(L) \cap M)$.
- $\pi_2(\pi_1^{-1}(L) \cap M) \subseteq \sigma(L)$:
  Let $v \in \pi_2(\pi_1^{-1}(L) \cap M)$ and let $w \in \pi_1^{-1}(L) \cap M$ with $\pi_2(w) = v$.
  We have $u = \pi_1(w) \in L$ and from 1 we get $v \in \sigma(u) \subseteq \sigma(L)$.

Closure under SEL_ε-substitutions

Lemma Words: $M = \left(\bigcup_{a \in A} a\sigma(a)\right)^*$

If $M \subseteq SE(\hat{\Sigma})$ satisfies

1. $\pi_2(w) \in \sigma(\pi_1(w))$ for each $w \in M$,
2. $\forall u \in SE(\Sigma)$, $\forall v \in \sigma(u)$, $\exists w \in M$ such that $u = \pi_1(w)$ and $v = \pi_2(w)$.

Then,

- for $L \subseteq SE(\Sigma)$, we have $\sigma(L) = \pi_2(\pi_1^{-1}(L) \cap M)$,
- for $L \subseteq SE(\Sigma')$, we have $\sigma^{-1}(L) = \pi_1(\pi_2^{-1}(L) \cap M)$.

Proof

- $\sigma(L) \subseteq \pi_2(\pi_1^{-1}(L) \cap M)$:
  Let $v \in \sigma(L)$ and let $u \in L$ with $v \in \sigma(u)$.
  From 2, $\exists w \in M$ with $\pi_1(w) = u$ and $\pi_2(w) = v$.
  Then, $w \in \pi_1^{-1}(L) \cap M$ and $v \in \pi_2(\pi_1^{-1}(L) \cap M)$.
- $\pi_2(\pi_1^{-1}(L) \cap M) \subseteq \sigma(L)$:
  Let $v \in \pi_2(\pi_1^{-1}(L) \cap M)$ and let $w \in \pi_1^{-1}(L) \cap M$ with $\pi_2(w) = v$.
  We have $u = \pi_1(w) \in L$ and from 1 we get $v \in \sigma(u) \subseteq \sigma(L)$.

Closure under SEL_ε-substitutions

Definition of M — Words: $M = \left(\bigcup_{a\in A} a\sigma(a)\right)^*$

For $f \in \Sigma_e$ and $a \in \Sigma_s \setminus \{\tau\}$, we define

$$\begin{aligned} M_f &= \{w \in SE(\hat{\Sigma}) \mid w = (\tau, b_0)^0 f_1 (\tau, b_1)^0 f_2 \cdots (\tau, b_n)^0 \\ &\qquad \text{with } b_0^0 f_1 b_1^0 f_2 \cdots b_n^0 \in \sigma(f)\} \cdot f \\ M_a &= \{w \in SE(\hat{\Sigma}) \mid w = (a, b_0)^{d_0} f_1 (a, b_1)^{d_1} f_2 \cdots \\ &\qquad \text{with } b_0^{d_0} f_1 b_1^{d_1} f_2 \cdots \in \sigma(a^{d_0+d_1+\cdots})\} \\ M_\tau &= \{(\tau,\tau)^d \mid d \in \overline{\mathbb{T}} \setminus \{0\}\} \end{aligned}$$

Note that each set M_f and M_a satisfies properties 1 and 2.

$M = \{w_1 w_2 \cdots \mid \exists a_1, a_2, \ldots \in \Sigma_e \cup \Sigma_s$ with $w_i \in M_{a_i}$ and $a_i \in \Sigma_s \Rightarrow a_{i+1} \neq a_i\}$.

Lemma

1. $\pi_2(w) \in \sigma(\pi_1(w))$ for each $w \in M$,
2. $\forall u \in SE(\Sigma)$, $\forall v \in \sigma(u)$, $\exists w \in M$ such that $u = \pi_1(w)$ and $v = \pi_2(w)$,
3. the language M is in the class SEL_ε.

Closure under inverse SEL-substitutions

The class SEL is not closed under arbitrary inverse SEL-substitutions

- Let $\Sigma_s = \Sigma'_s = \{a, b\}$ and $\Sigma_e = \Sigma'_e = \{f\}$.
- Let σ be the SEL-substitution defined by $L_a = \{a^1 f\}$, $L_b = \{b^0\}$ and $L_f = \{f\}$.
- $L = \{a^1 f b^0\}$ is a SEL.
- $\sigma^{-1}(L) = \{a^1 b^0\}$ is not a SEL.

Theorem

The class SEL is closed under inverse SEL-substitution acting only on events: $L_a = \{a\} \times \overline{\mathbb{T}}$ for all $a \in \Sigma_s$.

Closure under inverse SEL-substitutions

The class SEL is not closed under arbitrary inverse SEL-substitutions

- Let $\Sigma_s = \Sigma'_s = \{a, b\}$ and $\Sigma_e = \Sigma'_e = \{f\}$.
- Let σ be the SEL-substitution defined by $L_a = \{a^1 f\}$, $L_b = \{b^0\}$ and $L_f = \{f\}$.
- $L = \{a^1 f b^0\}$ is a SEL.
- $\sigma^{-1}(L) = \{a^1 b^0\}$ is not a SEL.

Theorem

The class SEL is closed under inverse SEL-substitution acting only on events: $L_a = \{a\} \times \overline{\mathbb{T}}$ for all $a \in \Sigma_s$.

Outline

Closure under intersection

Theorem

Classes SEL and SEL_ε are closed under intersection

Remarks

- Easy for the class SEL (no ε-transitions) or for time-event languages.
- More difficult with signals and ε-transitions due to signal stuttering and unobservability of τ^0.
- In LICS'97, Asarin, Caspi and Maler do not handle signal stuttering and consider finite runs only.
  In JACM'02, Asarin, Caspi and Maler deal with the intersection of time-event automata only.
- In STACS'00, Dima gives a construction to remove stuttering for automata with a single clock.
- In IPL'04 Durand-Lose 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 (no ε-transitions) or for time-event languages.
- More difficult with signals and ε-transitions due to signal stuttering and unobservability of τ^0.
- In LICS'97, Asarin, Caspi and Maler do not handle signal stuttering and consider finite runs only.
  In JACM'02, Asarin, Caspi and Maler deal with the intersection of time-event automata only.
- In STACS'00, Dima gives a construction to remove stuttering for automata with a single clock.
- In IPL'04 Durand-Lose 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 (no ε-transitions) or for time-event languages.
- More difficult with signals and ε-transitions due to signal stuttering and unobservability of τ^0.
- In LICS'97, Asarin, Caspi and Maler do not handle signal stuttering and consider finite runs only.
  In JACM'02, Asarin, Caspi and Maler deal with the intersection of time-event automata only.
- In STACS'00, Dima gives a construction to remove stuttering for automata with a single clock.
- In IPL'04 Durand-Lose 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 (no ε-transitions) or for time-event languages.
- More difficult with signals and ε-transitions due to signal stuttering and unobservability of τ^0.
- In LICS'97, Asarin, Caspi and Maler do not handle signal stuttering and consider finite runs only.
  In JACM'02, Asarin, Caspi and Maler deal with the intersection of time-event automata only.
- In STACS'00, Dima gives a construction to remove stuttering for automata with a single clock.
- In IPL'04 Durand-Lose 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 (no ε-transitions) or for time-event languages.
- More difficult with signals and ε-transitions due to signal stuttering and unobservability of τ^0.
- In LICS'97, Asarin, Caspi and Maler do not handle signal stuttering and consider finite runs only.
  In JACM'02, Asarin, Caspi and Maler deal with the intersection of time-event automata only.
- In STACS'00, Dima gives a construction to remove stuttering for automata with a single clock.
- In IPL'04 Durand-Lose 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 (no ε-transitions) or for time-event languages.
- More difficult with signals and ε-transitions due to signal stuttering and unobservability of τ^0.
- In LICS'97, Asarin, Caspi and Maler do not handle signal stuttering and consider finite runs only.
  In JACM'02, Asarin, Caspi and Maler deal with the intersection of time-event automata only.
- In STACS'00, Dima gives a construction to remove stuttering for automata with a single clock.
- In IPL'04 Durand-Lose 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.

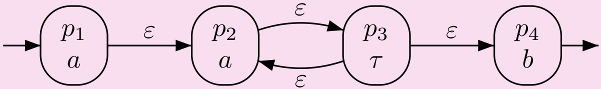
Closure under intersection

Theorem

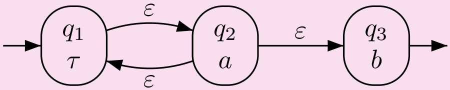
SEL_ε is closed under intersection

Problem 1 : stuttering with unobservability of τ^0

$\mathcal{B}_1$:



$\mathcal{B}_2$:



$p_1 \xrightarrow{1} p_1 \xrightarrow{\varepsilon} p_2 \xrightarrow{2} p_2 \xrightarrow{\varepsilon} p_3 \xrightarrow{0} p_3 \xrightarrow{\varepsilon} p_2 \xrightarrow{1} p_2 \xrightarrow{\varepsilon} p_3 \xrightarrow{0} p_3 \xrightarrow{\varepsilon} p_4$

$q_1 \xrightarrow{0} q_1 \xrightarrow{\varepsilon} q_2 \xrightarrow{2} q_2 \xrightarrow{\varepsilon} q_1 \xrightarrow{0} q_1 \xrightarrow{\varepsilon} q_2 \xrightarrow{2} q_2 \xrightarrow{\varepsilon} q_3$

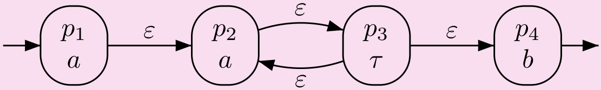
Closure under intersection

Theorem

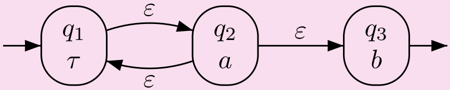
SEL_ε is closed under intersection

Problem 1 : stuttering with unobservability of τ^0

$\mathcal{B}_1$:



$\mathcal{B}_2$:



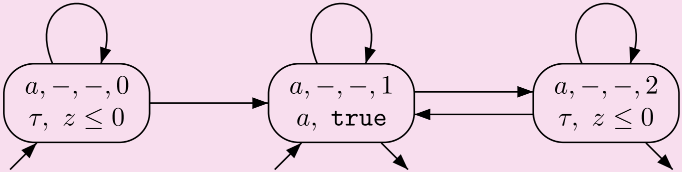
$$p_1 \xrightarrow{1} p_1 \xrightarrow{\varepsilon} p_2 \xrightarrow{2} p_2 \xrightarrow{\varepsilon} p_3 \xrightarrow{0} p_3 \xrightarrow{\varepsilon} p_2 \xrightarrow{1} p_2 \xrightarrow{\varepsilon} p_3 \xrightarrow{0} p_3 \xrightarrow{\varepsilon} p_4$$

$$q_1 \xrightarrow{0} q_1 \xrightarrow{\varepsilon} q_2 \xrightarrow{2} q_2 \xrightarrow{\varepsilon} q_1 \xrightarrow{0} q_1 \xrightarrow{\varepsilon} q_2 \xrightarrow{2} q_2 \xrightarrow{\varepsilon} q_3$$

Stuttering with unobservability of τ^0

Building maximal a-blocks

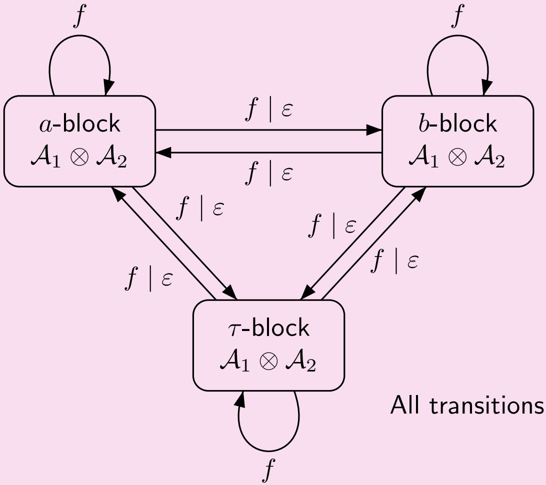
States : (a, p, q, i), where i is the synchronization mode.



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

Stuttering with unobservability of τ^0

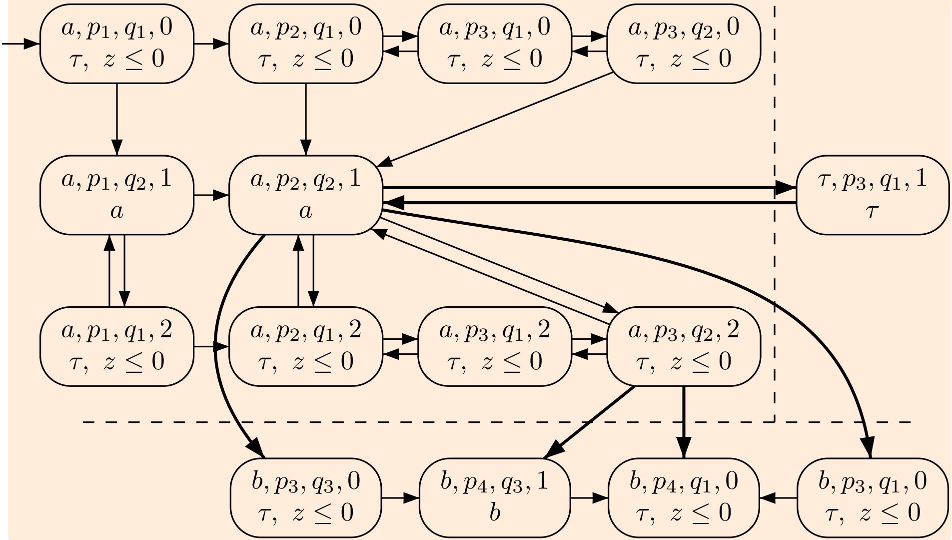
Connecting modules for a-blocks with synchronous transitions



All transitions reset a new clock z

Solution to problem 1

asynchronous/asynchronous product

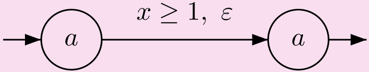


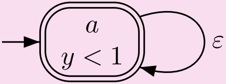
Closure under intersection

Theorem

SEL_ε is closed under intersection

Problem 2 : finite and infinite runs

$\mathcal{A}_1$: 

$\mathcal{A}_2$: 

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

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

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

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.

Outline

Conclusion

- Signal-event words are the natural objects for studying refinements, abstractions and other problems.
- 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 properties (refinement, abstraction) for the general case of SE-automata.
- We have proved closure under intersection for the general case of languages accepted by SE-automata.