CONCUR 2023 – Antwerp Belgium

Safety and Liveness of Quantitative Automata

Udi Boker [†]

Thomas A. Henzinger [‡]

Nicolas Mazzocchi [‡]

N. Ege Saraç [‡]

[†] Reichman University, Israel

[‡] Institute of Science and Technology, Austria

Boolean Properties

Definition

A Boolean property $\Phi \subseteq \Sigma^\omega$ or equivalently $\Phi \colon \Sigma^\omega \to \{0, 1\}$, is a language

Safety

Requests Not Duplicated

Liveness

All Requests Granted

Boolean Properties

Definition

A Boolean property $\Phi \subseteq \Sigma^\omega$ or equivalently $\Phi\colon \Sigma^\omega \to \{0,1\}$, is a language

Safety

Requests Not Duplicated

Liveness

All Requests Granted

Theorem: Decomposition[1]

All Boolean property Φ can be expressed by $\Phi = \Phi_{safe} \cap \Phi_{live}$

- Φ_{safe} *is safe*
- Φ_{live} *is live*

[1] Alpern, Schneider. *Defining liveness*. 1985

Quantitative Properties

Definition

A quantitative property[2] $\Phi\colon \Sigma^\omega \to \mathbb{D}$ is a quantitative language where $\mathbb{D}$ is a complete lattice

[2] Chatterjee, Doyen, Henzinger. *Quantitative Languages*. 2010

Quantitative Properties

Definition

A quantitative property $\Phi \colon \Sigma^\omega \to \mathbb{D}$ is a quantitative language where $\mathbb{D}$ is a complete lattice

Safety[3]

Minimal Response Time

Liveness[3]

Average Response Time

Theorem: Decomposition[3]

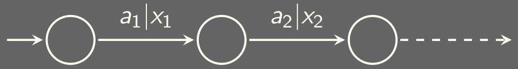
All quantitative property Φ can be expressed by $\Phi(w) = \min\{\Phi_{safe}(w), \Phi_{live}(w)\}$ for all $w \in \Sigma^\omega$

- *Φ_{safe} is quantitative safe*
- *Φ_{live} is quantitative live*

[3] Henzinger, Mazzocchi, Saraç. *Quantitative Safety and Liveness*. 2023

Quantitative Automata

Runs



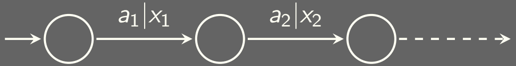
Word: $w = a_1 a_2 \ldots$ Value: $x = f(x_1 x_2 \ldots)$

Value functions

Inf, Sup, LimInf, LimSup
LimInfAvg, LimSupAvg, DSum

Quantitative Automata

Runs

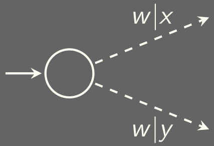


Word: $w = a_1 a_2 \dots$ Value: $x = f(x_1 x_2 \dots)$

Value functions

Inf, Sup, LimInf, LimSup
LimInfAvg, LimSupAvg, DSum

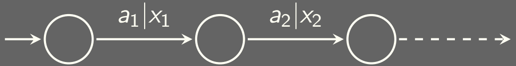
Non-determinism



$A(w) = \sup\{\text{values of } w\text{'s runs}\}$

Quantitative Automata

Runs



Word: $w = a_1 a_2 \ldots$ Value: $x = f(x_1 x_2 \ldots)$

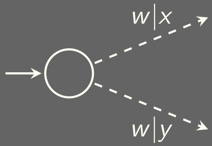
Subset of quantitative properties

- totally ordered domain

Value functions

Inf, Sup, LimInf, LimSup
LimInfAvg, LimSupAvg, DSum

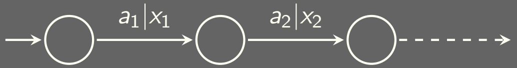
Non-determinism



$A(w) = \sup\{\text{values of } w\text{'s runs}\}$

Quantitative Automata

Runs



Word: $w = a_1 a_2 \dots$ Value: $x = f(x_1 x_2 \dots)$

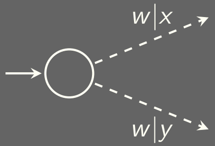
Subset of quantitative properties

- totally ordered domain
- finitely many weights

Value functions

Inf, Sup, LimInf, LimSup
LimInfAvg, LimSupAvg, DSum

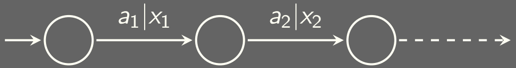
Non-determinism



$A(w) = \sup\{\text{values of } w\text{'s runs}\}$

Quantitative Automata

Runs



Word: $w = a_1 a_2 \dots$ Value: $x = f(x_1 x_2 \dots)$

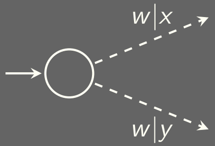
Subset of quantitative properties

- totally ordered domain
- finitely many weights
- supremum-closed
  $\forall u \in \Sigma^* : \sup_{v \in \Sigma^\omega} A(uv) \in \{A(uv') : v' \in \Sigma^\omega\}$

Value functions

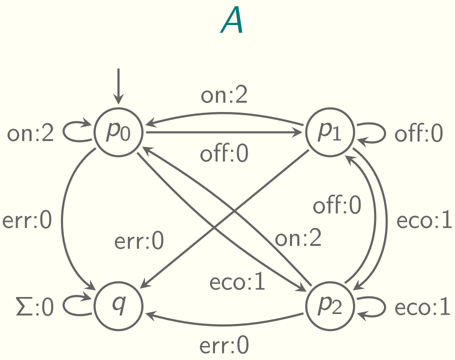
Inf, Sup, LimInf, LimSup
LimInfAvg, LimSupAvg, DSum

Non-determinism



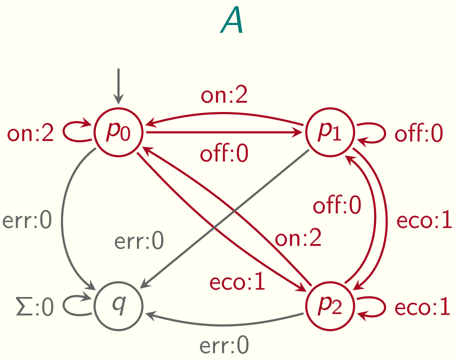
$A(w) = \sup\{\text{values of } w\text{'s runs}\}$

Example of LimSup Automaton



w = off on eco off eco off eco . . . off eco . . . $\quad A(w) = \text{LimSup } 0210101\ldots01\cdots = 1$

Example of LimSup Automaton

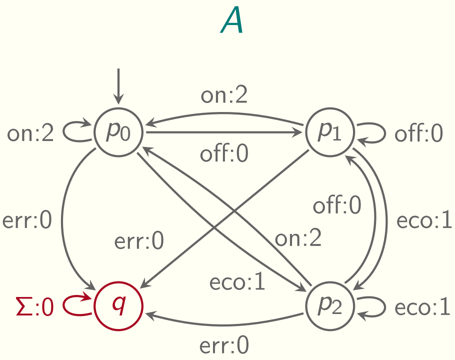


No Error

$$\forall u \in (\Sigma \setminus \{\text{err}\})^* : A(u\ \text{on}^\omega) = 2$$
$$\forall u \in (\Sigma \setminus \{\text{err}\})^* : A(u\ \text{eco}^\omega) = 1$$
$$\forall u \in (\Sigma \setminus \{\text{err}\})^* : A(u\ \text{off}^\omega) = 0$$

$w = \text{off on eco off eco off eco} \ldots \text{off eco} \ldots$ $\quad A(w) = \text{LimSup } 0210101 \ldots 01 \cdots = 1$

Example of LimSup Automaton



No Error

$$\forall u \in (\Sigma \setminus \{\text{err}\})^* : A(u\ \text{on}^\omega) = 2$$
$$\forall u \in (\Sigma \setminus \{\text{err}\})^* : A(u\ \text{eco}^\omega) = 1$$
$$\forall u \in (\Sigma \setminus \{\text{err}\})^* : A(u\ \text{off}^\omega) = 0$$

After Error

$$\forall v \in \Sigma^\omega : A(\text{err}\ v) = 0$$

$w = \text{off on eco off eco off eco} \ldots \text{off eco} \ldots$ $\quad A(w) = \text{LimSup } 0210101 \ldots 01 \cdots = 1$

Boolean Safety

Intuition

Every **wrong** hypothesis $w \in \Phi$ can always be rejected after a finite number of observations

Boolean Safety

Intuition

Every **wrong** hypothesis $w \in \Phi$ can always be rejected after a finite number of observations

Example: Requests Not Duplicated

- $\Sigma = \{\mathtt{r}, \mathtt{g}, \mathtt{t}, \mathtt{o}\}$ $\quad$ `r`: request, `g`: grant, `t`: clock-tick, `o`: other
- $\Phi =$ no `r` is followed by another `r` without some `g` in between

$$w = \mathtt{t\,r\,t\,o\,t\,t\,o\,g\,t\,o\,o\,r\,t\,t\,o\,r\,t\,t\,o\,g\,t\,r} \cdots$$

$$w \in \Phi: \quad \mathtt{T\,.\,.\,.\,.\,.\,.\,.\,.\,.\,.\,.\,.\,.\,.\,F\,.\,.\,.\,.\,.\,.} \cdots$$

Boolean Safety

Intuition

Every **wrong** hypothesis $w \in \Phi$ can always be rejected after a finite number of observations

Example: Requests Not Duplicated

- $\Sigma = \{\mathtt{r}, \mathtt{g}, \mathtt{t}, \mathtt{o}\}$ $\mathtt{r}$: request, $\mathtt{g}$: grant, $\mathtt{t}$: clock-tick, $\mathtt{o}$: other
- $\Phi =$ no $\mathtt{r}$ is followed by another $\mathtt{r}$ without some $\mathtt{g}$ in between

Definition

A boolean property $\Phi \subseteq \Sigma^\omega$ is safe when

$$\forall w \in \Sigma^\omega : w \notin \Phi \implies \exists u \sqsubseteq w : \forall v \in \Sigma^\omega : uv \notin \Phi$$

Quantitative Safety

Intuition

Every **wrong** hypothesis $\Phi(w) \geq x$, can always be rejected after a finite number of observations

Quantitative Safety

Intuition

Every **wrong** hypothesis $\Phi(w) \geq x$, can always be rejected after a finite number of observations

Example: Minimal Response Time

- $\Sigma = \{\mathtt{r}, \mathtt{g}, \mathtt{t}, \mathtt{o}\}$ $\mathtt{r}$: request, $\mathtt{g}$: grant, $\mathtt{t}$: clock-tick, $\mathtt{o}$: other
- $\Phi_{\mathsf{min}}(w)$ = greatest lower bound on the occurrences of $\mathtt{t}$ between all matching $\mathtt{r}/\mathtt{g}$ in w

$$\begin{array}{rl} w = & \mathtt{t\,r\,t\,o\,t\,t\,o\,g\,t\,o\,o\,r\,t\,t\,o\,r\,t\,t\,o\,g\,t\,r} \cdots \\ \Phi(w) \geq 3: & \mathtt{T\,.\,.\,.\,.\,.\,.\,.\,.\,.\,.\,.\,.\,.\,.\,.\,.\,.\,.\,F\,.\,.} \cdots \end{array}$$

Quantitative Safety

Intuition

Every **wrong** hypothesis $\Phi(w) \geq x$, can always be rejected after a finite number of observations

Example: Minimal Response Time

- $\Sigma = \{\mathtt{r}, \mathtt{g}, \mathtt{t}, \mathtt{o}\}$ $\mathtt{r}$: request, $\mathtt{g}$: grant, $\mathtt{t}$: clock-tick, $\mathtt{o}$: other
- $\Phi_{\mathsf{min}}(w)$ = greatest lower bound on the occurrences of $\mathtt{t}$ between all matching $\mathtt{r}/\mathtt{g}$ in w

Definition[4]

A quantitative property $\Phi : \Sigma^\omega \to \mathbb{D}$ is safe when

$$\forall x \in \mathbb{D} : \forall w \in \Sigma^\omega : \Phi(w) \not\geq x \implies \exists u \sqsubseteq w : \sup_{v \in \Sigma^\omega} \Phi(uv) \not\geq x$$

[4] Henzinger, Mazzocchi, Saraç. *Quantitative Safety and Liveness*. 2023

Safety of Quantitative Automata

Boolean Safety

$$\forall w \in \Sigma^\omega : w \notin \Phi \implies \exists u \sqsubseteq w : \forall v \in \Sigma^\omega : uv \notin \Phi$$

Quantitative Safety

$$\forall x \in \mathbb{D} : \forall w \in \Sigma^\omega : \Phi(w) \ngtr x \implies \exists u \sqsubseteq w : \sup_{v \in \Sigma^\omega} \Phi(uv) \ngtr x$$

Safety of Quantitative Automata

Boolean Safety

$$\forall w \in \Sigma^\omega : w \notin \Phi \implies \exists u \sqsubseteq w : \forall v \in \Sigma^\omega : uv \notin \Phi$$

Quantitative Safety

$$\forall x \in \mathbb{D} : \forall w \in \Sigma^\omega : \Phi(w) \not\geq x \implies \exists u \sqsubseteq w : \sup_{v \in \Sigma^\omega} \Phi(uv) \not\geq x$$

Threshold safety

A quantitative property $\Phi : \Sigma^\omega \to \mathbb{D}$ is threshold-safe when

$$\forall x \in \mathbb{D} : \Phi_{\geq x} = \{w \in \Sigma^\omega \mid \Phi(w) \geq x\} \text{ is safe}$$

Safety of Quantitative Automata

Boolean Safety

$$\forall w \in \Sigma^\omega : w \notin \Phi \implies \exists u \sqsubseteq w : \forall v \in \Sigma^\omega : uv \notin \Phi$$

Quantitative Safety

$$\forall x \in \mathbb{D} : \forall w \in \Sigma^\omega : \Phi(w) \not\geq x \implies \exists u \sqsubseteq w : \sup_{v \in \Sigma^\omega} \Phi(uv) \not\geq x$$

Threshold safety

A quantitative property $\Phi : \Sigma^\omega \to \mathbb{D}$ is threshold-safe when

$$\forall x \in \mathbb{D} : \Phi_{\geq x} = \{w \in \Sigma^\omega \mid \Phi(w) \geq x\} \text{ is safe}$$

Theorem: For totally ordered domain, threshold-safety = quantitative safety

Quantitative Safety Closure

Intuition

The safety closure $\Phi^\star$ is the least safety property that bound Φ from above

Quantitative Safety Closure

Intuition

The safety closure $\Phi^\star$ is the least safety property that bound Φ from above

Example: Minimal Response Time

- $\Sigma = \{\mathtt{r}, \mathtt{g}, \mathtt{t}, \mathtt{o}\}$
- $\Phi_{\mathsf{min}}(w) =$ greatest lower bound on the occurrences of $\mathtt{t}$ between all matching $\mathtt{r}/\mathtt{g}$ in w

$$
\begin{array}{rl}
w = & \mathtt{t\,r\,t\,o\,t\,t\,o\,g\,t\,o\,o\,r\,t\,t\,o\,r\,t\,t\,o\,g\,t\,r} \cdots \\
\text{least upper bound}: & \infty\,.\,.\,.\,.\,.\,3\,.\,.\,.\,.\,.\,.\,.\,.\,.\,.\,.\,2\,.\,.\,\cdots
\end{array}
$$

Quantitative Safety Closure

Intuition

The safety closure $\Phi^\star$ is the least safety property that bound Φ from above

Example: Minimal Response Time

- $\Sigma = \{\mathtt{r}, \mathtt{g}, \mathtt{t}, \mathtt{o}\}$
- $\Phi_{\mathsf{min}}(w)$ = greatest lower bound on the occurrences of $\mathtt{t}$ between all matching $\mathtt{r}/\mathtt{g}$ in w

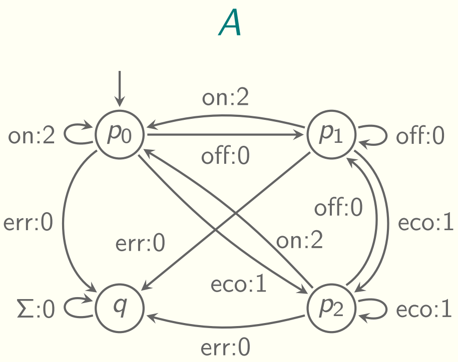
Definition[5]

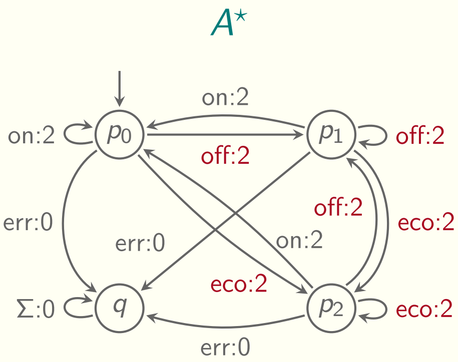
Given $\Phi : \Sigma^\omega \to \mathbb{D}$, its safety closure is $\Phi^\star(w) := \inf_{u \sqsubseteq w} \sup_{v \in \Sigma^\omega} \Phi(uv)$ for all $w \in \Sigma^\omega$

Theorem[5]: Φ is safe $\iff \Phi = \Phi^\star$

[5] Henzinger, Mazzocchi, Saraç. *Quantitative Safety and Liveness*. 2023

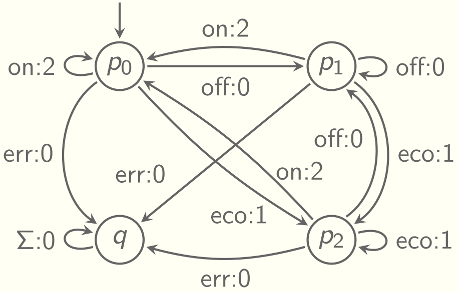
Example of Safety Closure



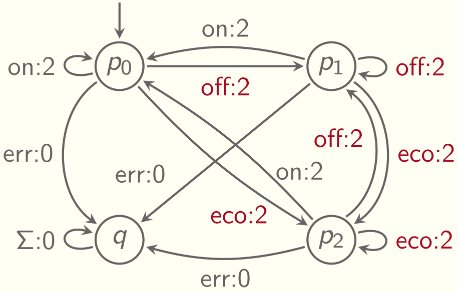


Example of Safety Closure

A



$A^\star$



A is not safe since $A \neq A^\star$ as witnessed by $A(\mathrm{eco}^\omega) = 1$, $A^\star(\mathrm{eco}^\omega) = 2$

Deciding Safety

Reduction to language equivalence problem

Classes of Sup, LimInf and LimSup are decidable for equivalence: determine whether $A = A^\star$

Deciding Safety

Reduction to language equivalence problem

Classes of Sup, LimInf and LimSup are decidable for equivalence: determine whether $A = A^\star$

Safe value function

Classes of Inf and DSum automata contain only safe automata: safety is trivial

Deciding Safety

Reduction to language equivalence problem

Classes of Sup, LimInf and LimSup are decidable for equivalence: determine whether $A = A^\star$

Safe value function

Classes of Inf and DSum automata contain only safe automata: safety is trivial

About LimInfAvg and LimSupAvg

- $\mathsf{Avg}(x_1 x_2 \ldots) - \mathsf{Avg}(y_1 y_2 \ldots) \neq \mathsf{Avg}((x_1 - y_1)(x_2 - y_2) \ldots)$

Deciding Safety

Reduction to language equivalence problem

Classes of Sup, LimInf and LimSup are decidable for equivalence: determine whether $A = A^\star$

Safe value function

Classes of Inf and DSum automata contain only safe automata: safety is trivial

About LimInfAvg and LimSupAvg

- $\mathsf{Avg}(x_1x_2\ldots) - \mathsf{Avg}(y_1y_2\ldots) \neq \mathsf{Avg}((x_1 - y_1)(x_2 - y_2)\ldots)$
- Equals if $y_1y_2\ldots$ is eventually constant

Deciding Safety

Reduction to language equivalence problem

Classes of Sup, LimInf and LimSup are decidable for equivalence: determine whether $A = A^\star$

Safe value function

Classes of Inf and DSum automata contain only safe automata: safety is trivial

About LimInfAvg and LimSupAvg

- $\mathsf{Avg}(x_1 x_2 \dots) - \mathsf{Avg}(y_1 y_2 \dots) \neq \mathsf{Avg}((x_1 - y_1)(x_2 - y_2) \dots)$
- Equals if $y_1 y_2 \dots$ is eventually constant
- $A = A^\star \iff A - A^\star = 0$ because all runs of $A^\star$ is eventually constant

Deciding Safety

Reduction to language equivalence problem

Classes of Sup, LimInf and LimSup are decidable for equivalence: determine whether $A = A^\star$

Safe value function

Classes of Inf and DSum automata contain only safe automata: safety is trivial

About LimInfAvg and LimSupAvg

- $\mathrm{Avg}(x_1 x_2 \dots) - \mathrm{Avg}(y_1 y_2 \dots) \neq \mathrm{Avg}((x_1 - y_1)(x_2 - y_2) \dots)$
- Equals if $y_1 y_2 \dots$ is eventually constant
- $A = A^\star \iff A - A^\star = 0$ because all runs of $A^\star$ is eventually constant
- Determine whether $A - A^\star = 0$, by reducing the limitedness of distance automata

Deciding Safety

Reduction to language equivalence problem

Classes of Sup, LimInf and LimSup are decidable for equivalence: determine whether $A = A^\star$

Safe value function

Classes of Inf and DSum automata contain only safe automata: safety is trivial

About LimInfAvg and LimSupAvg

- $\mathsf{Avg}(x_1 x_2 \dots) - \mathsf{Avg}(y_1 y_2 \dots) \neq \mathsf{Avg}((x_1 - y_1)(x_2 - y_2) \dots)$
- Equals if $y_1 y_2 \dots$ is eventually constant
- $A = A^\star \iff A - A^\star = 0$ because all runs of $A^\star$ is eventually constant
- Determine whether $A - A^\star = 0$, by reducing the limitedness of distance automata

Theorem: Safety is decidable for Inf, Sup, LimInf, LimSup, Avg, and DSum automata

Boolean Liveness

Intuition

Some **wrong** hypothesis $w \in \Phi$ can never be rejected after any finite number of observations

Boolean Liveness

Intuition

Some **wrong** hypothesis $w \in \Phi$ can never be rejected after any finite number of observations

Example: All Requests Granted

- $\Sigma = \{\mathtt{r}, \mathtt{g}, \mathtt{t}, \mathtt{o}\}$
- Φ = every $\mathtt{r}$ is eventually followed by some $\mathtt{g}$

$$
\begin{aligned}
w = \quad & \mathtt{t\,r\,t\,o\,t\,t\,o\,g\,t\,o\,o\,r\,t\,t\,o\,r\,t\,t\,o\,g\,t\,r} \cdots \\
w \in \Phi: \quad & \mathtt{T\,.\,?} \cdots
\end{aligned}
$$

Boolean Liveness

Intuition

Some **wrong** hypothesis $w \in \Phi$ can never be rejected after any finite number of observations

Example: All Requests Granted

- $\Sigma = \{\mathtt{r}, \mathtt{g}, \mathtt{t}, \mathtt{o}\}$
- $\Phi =$ every $\mathtt{r}$ is eventually followed by some $\mathtt{g}$

Definition

A boolean property $\Phi \subseteq \Sigma^\omega$ is live when

$$\forall u \in \Sigma^* : \exists v \in \Sigma^\omega : uv \in \Phi$$

Quantitative Liveness

Intuition

Some **wrong** hypothesis $\Phi(w) \geq x$ can never be rejected after any finite number of observations

Quantitative Liveness

Intuition

Some **wrong** hypothesis $\Phi(w) \geq x$ can never be rejected after any finite number of observations

Example: Average Response Time

- $\Sigma = \{\mathtt{r}, \mathtt{g}, \mathtt{t}, \mathtt{o}\}$
- $\Phi_{\text{avg}}(w) =$ average on the occurrences of $\mathtt{t}$ between all matching $\mathtt{r}/\mathtt{g}$ in w

$$
\begin{array}{rl}
w = & \mathtt{t\,r\,t\,o\,t\,t\,o\,g\,t\,o\,o\,r\,t\,t\,o\,r\,t\,t\,o\,g\,t\,r} \cdots \\
\Phi(w) \geq 3: & \mathtt{T\,.\,?} \cdots
\end{array}
$$

Quantitative Liveness

Intuition

Some **wrong** hypothesis $\Phi(w) \geq x$ can never be rejected after any finite number of observations

Example: Average Response Time

- $\Sigma = \{\mathtt{r}, \mathtt{g}, \mathtt{t}, \mathtt{o}\}$
- $\Phi_{\text{avg}}(w) =$ average on the occurrences of $\mathtt{t}$ between all matching $\mathtt{r}/\mathtt{g}$ in w

Definition[6]

A quantitative property $\Phi : \Sigma^\omega \to \mathbb{D}$ is live when

$$\forall w \in \Sigma^\omega : \Phi(w) < \top \implies \exists x \in \mathbb{D} : \Phi(w) \not\geq x \wedge \forall u \sqsubseteq w : \sup_{v \in \Sigma^\omega} \Phi(uv) \geq x$$

[6] Henzinger, Mazzocchi, Saraç. *Quantitative Safety and Liveness*. 2023

Liveness of Quantitative Automata

Threshold Liveness

A quantitative property $\Phi : \Sigma^\omega \to \mathbb{D}$ *is threshold-live when*

$$\forall x \in \mathbb{D} : \Phi_{\geq x} = \{w \in \Sigma^\omega \mid \Phi(w) \geq x\} \text{ is live}$$

Liveness of Quantitative Automata

Threshold Liveness

A quantitative property $\Phi : \Sigma^\omega \to \mathbb{D}$ *is threshold-live when*

$$\forall x \in \mathbb{D} : \Phi_{\geq x} = \{w \in \Sigma^\omega \mid \Phi(w) \geq x\} \textit{ is live}$$

Theorem: A property Φ is threshold live iff the set $\{w \in \Sigma^\omega \mid \Phi(w) = \top\}$ is dense

Liveness of Quantitative Automata

Threshold Liveness

A quantitative property $\Phi : \Sigma^\omega \to \mathbb{D}$ *is threshold-live when*

$$\forall x \in \mathbb{D} : \Phi_{\geq x} = \{w \in \Sigma^\omega \mid \Phi(w) \geq x\} \text{ is live}$$

Theorem: A property Φ is threshold live iff the set $\{w \in \Sigma^\omega \mid \Phi(w) = \top\}$ is dense

Top Liveness

A quantitative property $\Phi : \Sigma^\omega \to \mathbb{D}$ *is top-live when* $\Phi^\star(w) = \top$ *for all* $w \in \Sigma^\omega$

Liveness of Quantitative Automata

Threshold Liveness

A quantitative property $\Phi : \Sigma^\omega \to \mathbb{D}$ *is threshold-live when*

$$\forall x \in \mathbb{D} : \Phi_{\geq x} = \{w \in \Sigma^\omega \mid \Phi(w) \geq x\} \textit{ is live}$$

Theorem: A property Φ is threshold live iff the set $\{w \in \Sigma^\omega \mid \Phi(w) = \top\}$ is dense

Top Liveness

A quantitative property $\Phi : \Sigma^\omega \to \mathbb{D}$ *is top-live when* $\Phi^\star(w) = \top$ *for all* $w \in \Sigma^\omega$

Theorem: For supremum-closed properties, top-liveness = threshold-liveness = liveness

Deciding Liveness

Reduction to constant function problem

All classes are decidable for the constant function problem: determine whether $A^\star = \top$

Deciding Liveness

Reduction to constant function problem

All classes are decidable for the constant function problem: determine whether $A^\star = \top$

About DSum

- Every DSum automaton equals its safety closure: determine whether $A = \top$

Deciding Liveness

Reduction to constant function problem

All classes are decidable for the constant function problem: determine whether $A^\star = \top$

About DSum

- Every DSum automaton equals its safety closure: determine whether $A = \top$
- Determine the highest achievable value of each state

Deciding Liveness

Reduction to constant function problem

All classes are decidable for the constant function problem: determine whether $A^\star = \top$

About DSum

- Every DSum automaton equals its safety closure: determine whether $A = \top$
- Determine the highest achievable value of each state
- Trim transitions that do not lead to the highest value of the source state

Deciding Liveness

Reduction to constant function problem

All classes are decidable for the constant function problem: determine whether $A^\star = \top$

About DSum

- Every DSum automaton equals its safety closure: determine whether $A = \top$
- Determine the highest achievable value of each state
- Trim transitions that do not lead to the highest value of the source state
- Decide universality of underlying finite state automaton (ignoring weights)

Deciding Liveness

Reduction to constant function problem

All classes are decidable for the constant function problem: determine whether $A^\star = \top$

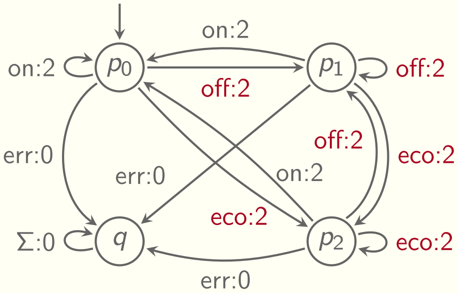
About DSum

- Every DSum automaton equals its safety closure: determine whether $A = \top$
- Determine the highest achievable value of each state
- Trim transitions that do not lead to the highest value of the source state
- Decide universality of underlying finite state automaton (ignoring weights)

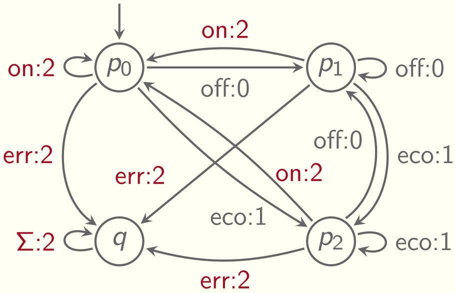
Theorem: Liveness is decidable for Inf, Sup, LimInf, LimSup, Avg, and DSum automata

Example of Safety-Liveness Decomposition

$A_{\text{safe}} = A^{\star}$



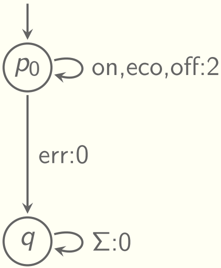
A_{live}



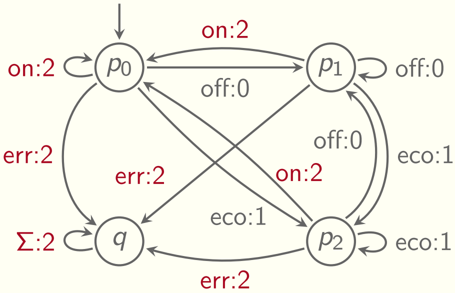
construction for **deterministic** for Sup, LimInf, and LimSup automata

Example of Safety-Liveness Decomposition

$A_{\text{safe}} = A^{\star}$



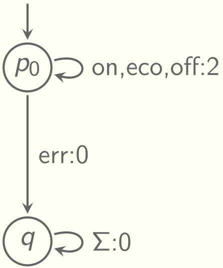
A_{live}



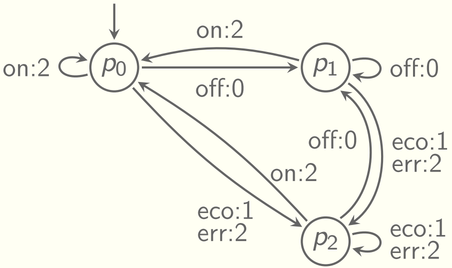
$$A(w) = \min\{A_{\text{safe}}(w), A_{\text{live}}(w)\}$$

Example of Safety-Liveness Decomposition

$A_{\text{safe}} = A^{\star}$



A_{live}



$$A(w) = \min\{A_{\text{safe}}(w), A_{\text{live}}(w)\}$$

In a nutshell

	Inf	Sup, LimInf, LimSup	LimInfAvg, LimSupAvg	DSum
Safety Closure construct $A^\star$	$O(1)$	PTime		$O(1)$
Is A constant? i.e., $A = \top$	PSpace-complete			
Is A safe? i.e., $A^\star = A$	$O(1)$	PSpace-complete	ExpSpace PSpace-hard	$O(1)$
Is A live? i.e., $A^\star = \top$	PSpace-complete			
Decomposition construct A_{safe} A_{live}	$O(1)$	PTime if **deterministic**	Open	$O(1)$

In a nutshell

	Inf	Sup, LimInf, LimSup	LimInfAvg, LimSupAvg	DSum
Safety Closure construct $A^\star$	$O(1)$	PTime		$O(1)$
Is A constant? i.e., $A = \top$	PSpace-complete			
Is A safe? i.e., $A^\star = A$	$O(1)$	PSpace-complete	ExpSpace PSpace-hard	$O(1)$
Is A live? i.e., $A^\star = \top$	PSpace-complete			
Decomposition construct A_{safe} A_{live}	$O(1)$	PTime if **deterministic**	Open	$O(1)$

Thank you