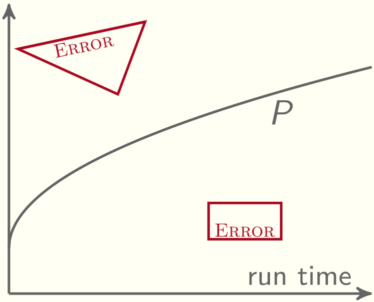
Weighted Transducers for Robustness Verification

Emmanuel Filiot[*]
Jean-François Raskin[*]
Nicolas Mazzocchi[*]
Sriram Sankaranarayanan[†]
Ashutosh Trivedi[†]

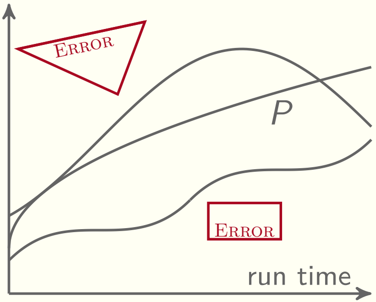
[*] Université libre de Bruxelles
[†] University Colorado Boulder

CONCUR 2020

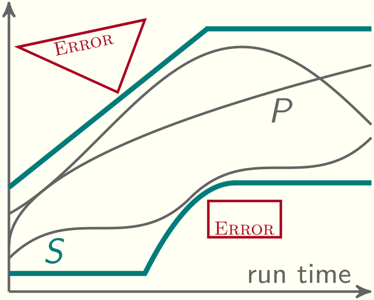
Model-checking



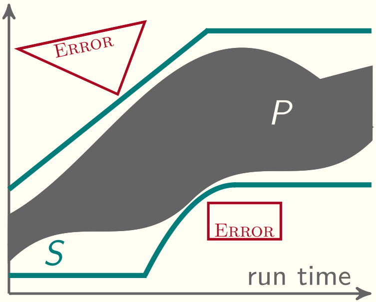
Model-checking



Model-checking



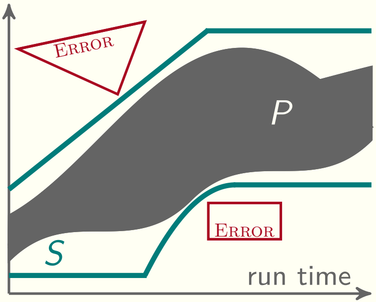
Model-checking



Model-checking

Program traces | **S**pecified traces

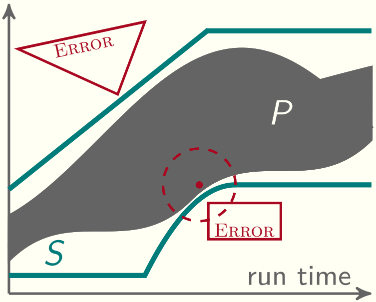
$$P \subseteq S$$



Model-checking

Program traces **S**pecified traces

$$P \subseteq S$$

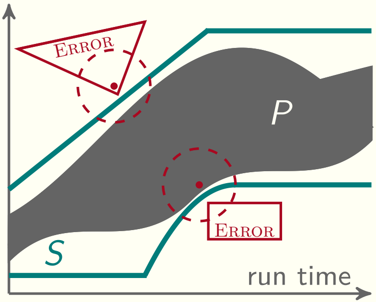


Model-checking

Program traces

Specified traces

$$P \subseteq S$$



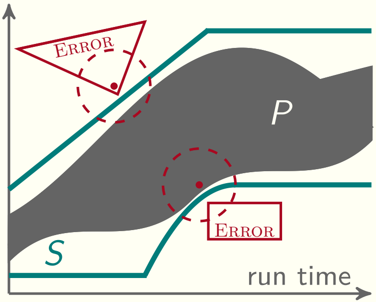
Model-checking

Program traces | **S**pecified traces

$$P \subseteq S$$
$$P \subseteq_{N,\nu} S$$

Noise threshold ν

$\underbrace{\qquad\qquad\qquad}_{\nu\text{-Neighborhood}}$



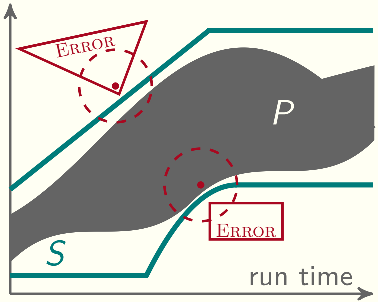
Model-checking

Program traces | **S**pecified traces

$$P \subseteq S$$

$$P \subseteq_{N,\nu} S$$

$\underbrace{\textbf{N}\text{oise} \quad \text{threshold } \nu}_{\nu\text{-}\mathbf{N}\text{eighborhood}}$



Language inclusion problem

- **Classical** $\forall t \quad t \in P \implies t \in S$

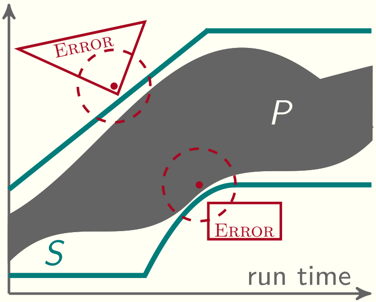
Model-checking

Program traces — **S**pecified traces

$$P \subseteq S$$

$$P \subseteq_{N,\nu} S$$

$\underbrace{\textbf{N}\text{oise} \quad \text{threshold } \boldsymbol{\nu}}_{\nu\text{-Neighborhood}}$



Language inclusion problem

- **Classical** $\forall t \quad t \in P \implies t \in S$
- **ν-Robust** $\forall t_1, t_2 \quad (N(t_1, t_2) \leq \nu \wedge t_1 \in P) \implies t_2 \in S$

Quantitative formalism

Accepting run

Quantitative formalism

Accepting run

Quantitative formalism

Accepting run

Noise model

$$u_{in} = a_0 \dots a_n \overset{noise}{\rightsquigarrow} u_{out} = w_0 \dots w_n$$

$$N(u_{in}, u_{out}) = \mathrm{cost}(x_0, \dots, x_n)$$

$$\mathrm{cost}\colon \mathbb{N}, \dots, \mathbb{N} \to \mathbb{Q}_{\geq 0}$$

Quantitative formalism

Accepting run

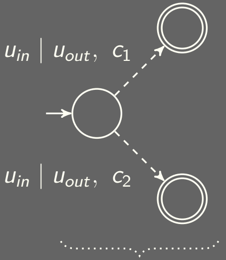
Noise model

$$u_{in} = a_0 \ldots a_n \overset{noise}{\rightsquigarrow} u_{out} = w_0 \ldots w_n$$

$$N(u_{in}, u_{out}) = \mathrm{cost}(x_0, \ldots, x_n)$$

$$\mathrm{cost} : \mathbb{N}, \ldots, \mathbb{N} \to \mathbb{Q}_{\geq 0}$$

Non-determinism



$$N(u_{in}, u_{out}) = \min\{c_1, c_2\}$$

Quantitative formalism

Accepting run

Noise model

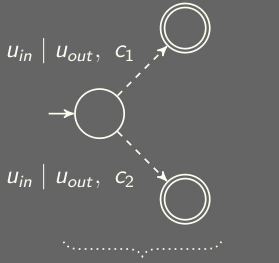
$$u_{in} = a_0 \ldots a_n \overset{noise}{\rightsquigarrow} u_{out} = w_0 \ldots w_n$$

$$N(u_{in}, u_{out}) = \mathrm{cost}(x_0, \ldots, x_n)$$

$$\mathrm{cost} : \mathbb{N}, \ldots, \mathbb{N} \to \mathbb{Q}_{\geq 0}$$

$$N(u, u) = 0$$

Non-determinism



$$N(u_{in}, u_{out}) = \min\{c_1, c_2\}$$

Neighborhood functions

Edit distance

$$\begin{array}{l} \texttt{R E L E } \square \texttt{ V A N T} \\ \texttt{\ \ E L E P H A N T} \end{array} \quad \text{least} \begin{cases} \text{inserts} \\ \text{deletes} \\ \text{changes} \end{cases}$$

Sum $= x_0 + \cdots + x_n$

Mohri, CIAA'02

Neighborhood functions

Edit distance

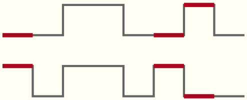
R E L E □ V A N T
E L E P H A N T

least $\left\{\begin{array}{l}\text{inserts}\\ \text{deletes}\\ \text{changes}\end{array}\right.$

Sum $= x_0 + \cdots + x_n$

Mohri, CIAA'02

Average mismatch



3 mismatch
5 match

Avg $= \frac{x_0 + \cdots + x_n}{n+1}$

Neighborhood functions

Edit distance

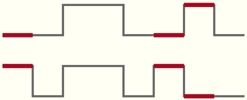
`R E L E □ V A N T`
`  E L E P H A N T`

$$\text{least} \begin{cases} \text{inserts} \\ \text{deletes} \\ \text{changes} \end{cases}$$

$$\textbf{Sum} = x_0 + \cdots + x_n$$

Mohri, CIAA'02

Average mismatch



3 mismatch
5 match

$$\textbf{Avg} = \frac{x_0 + \cdots + x_n}{n+1}$$

Discounted-sum distance

1	0	0	1	0	0	0	$\equiv 72$
0	1	1	1	1	1	0	$\equiv 66$

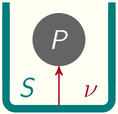
$$\textbf{Disc} = \lambda^0 x_0 + \cdots + \lambda^n x_n$$

$$\lambda \in \mathbb{Q} \cap (0,1)$$

Contributions

Robust inclusion

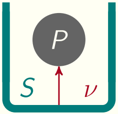
Input: P, S, N, ν
Output: Decide whether $P \subseteq_{N,\nu} S$



Contributions

Robust inclusion

Input: P, S, N, ν
Output: Decide whether $P \subseteq_{N,\nu} S$



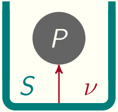
Monotonicity

$$\nu_2 > \nu_1 \implies P \subseteq_{N,\nu_2} S \implies P \subseteq_{N,\nu_1} S$$

Contributions

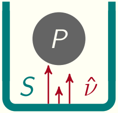
Robust inclusion

Input: P, S, N, ν
Output: Decide whether $P \subseteq_{N,\nu} S$



Threshold synthesis

Input: P, S, N
Output: Compute $\hat{\nu}$ such that $P \subseteq_{N,\hat{\nu}} S$



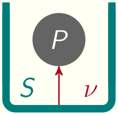
Monotonicity

$$\nu_2 > \nu_1 \implies P \subseteq_{N,\nu_2} S \implies P \subseteq_{N,\nu_1} S$$

Contributions

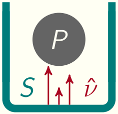
Robust inclusion

Input: P, S, N, ν
Output: Decide whether $P \subseteq_{N,\nu} S$



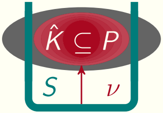
Threshold synthesis

Input: P, S, N
Output: Compute $\hat{\nu}$ such that $P \subseteq_{N,\hat{\nu}} S$



Kernel synthesis

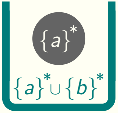
Input: P, S, N, ν
Output: Compute $\hat{K} \subseteq P$ such that $\hat{K} \subseteq_{N,\nu} S$



Robust inclusion

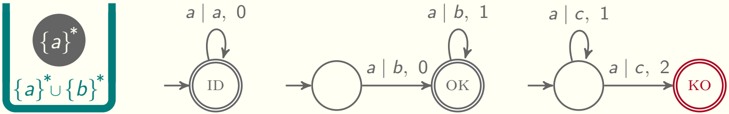
Towards an algorithm

Example



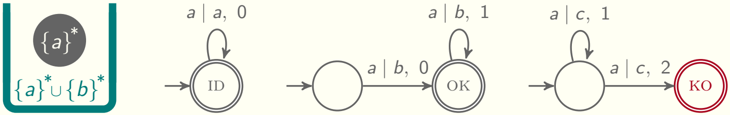
Towards an algorithm

Example



Towards an algorithm

Example

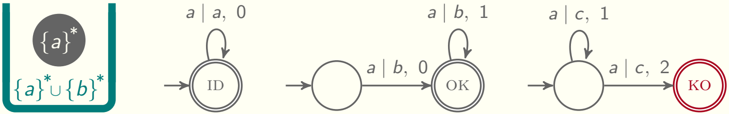


$\longrightarrow \quad 2$

	$P \subseteq_{N,\nu} S$
Sum	
Avg	
Disc	

Towards an algorithm

Example



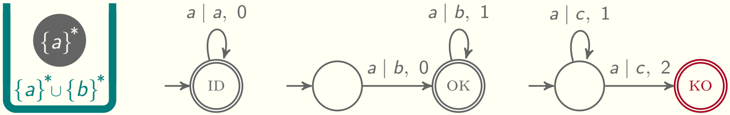
$\longrightarrow$ 2

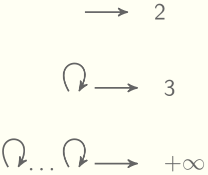
$\circlearrowright \longrightarrow$ 3

	$P \subseteq_{N,\nu} S$
Sum	
Avg	
Disc	

Towards an algorithm

Example

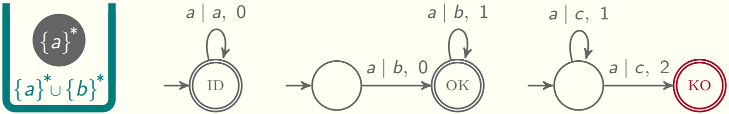


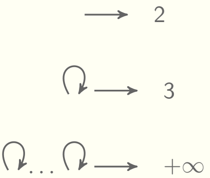


	$P \subseteq_{N,\nu} S$
Sum	
Avg	
Disc	

Towards an algorithm

Example

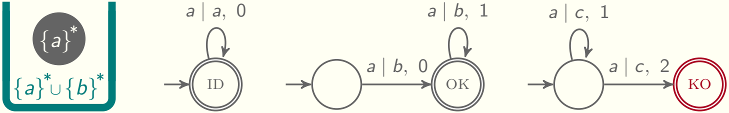




	$P \subseteq_{N,\nu} S$
Sum	$\nu \in [0, 2)$
Avg	
Disc	

Towards an algorithm

Example

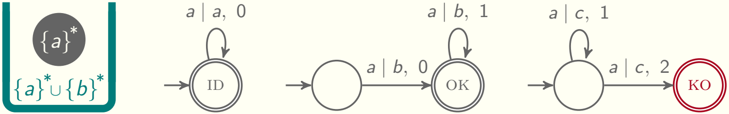


$\longrightarrow \quad 2$

	$P \subseteq_{N,\nu} S$
Sum	$\nu \in [0, 2)$
Avg	
Disc	

Towards an algorithm

Example



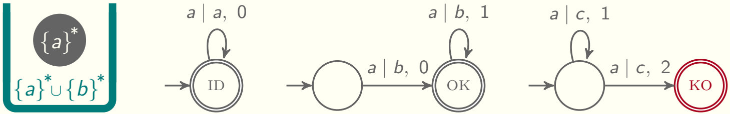
$\longrightarrow$ 2

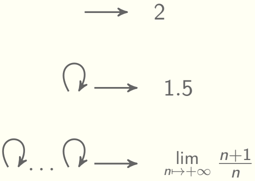
$\longrightarrow$ 1.5

	$P \subseteq_{N,\nu} S$
Sum	$\nu \in \big[0, 2\big)$
Avg	
Disc	

Towards an algorithm

Example

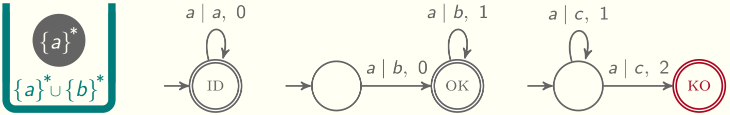


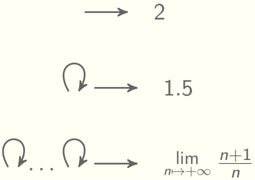


	$P \subseteq_{N,\nu} S$
Sum	$\nu \in [0, 2)$
Avg	
Disc	

Towards an algorithm

Example

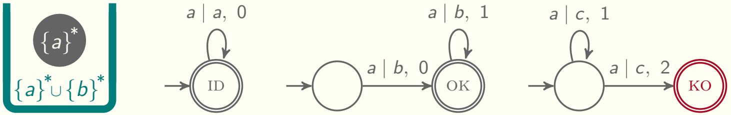




	$P \subseteq_{N,\nu} S$
Sum	$\nu \in [0, 2)$
Avg	$\nu \in [0, 1]$
Disc	

Towards an algorithm

Example

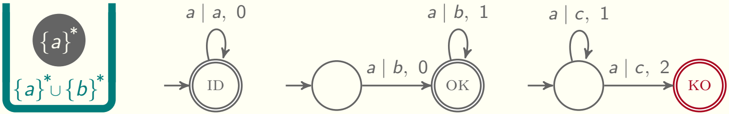


$\longrightarrow\ 2$

	$P \subseteq_{N,\nu} S$
Sum	$\nu \in [0, 2)$
Avg	$\nu \in [0, 1]$
Disc	

Towards an algorithm

Example



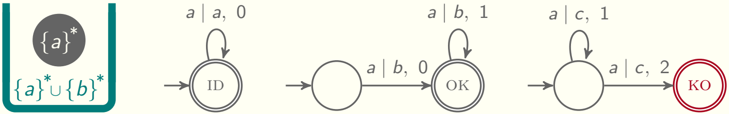
$\longrightarrow \quad 2$

$\circlearrowright \longrightarrow \quad 1 + 2\lambda$

	$P \subseteq_{N,\nu} S$
Sum	$\nu \in \left[0, 2\right)$
Avg	$\nu \in \left[0, 1\right]$
Disc	

Towards an algorithm

Example



$\longrightarrow \quad 2$

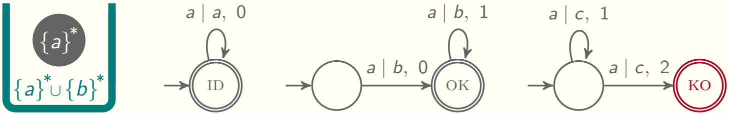
$\longrightarrow \quad 1 + 2\lambda$

$\ldots \longrightarrow \quad \lim_{n \mapsto +\infty} \frac{1-\lambda^{n+1}}{1-\lambda} + 2\lambda^{n+1}$

	$P \subseteq_{N,\nu} S$
Sum	$\nu \in [0, 2)$
Avg	$\nu \in [0, 1]$
Disc	

Towards an algorithm

Example



$\longrightarrow \quad 2$

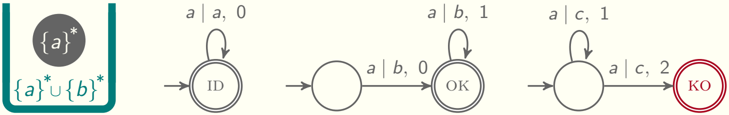
$\longrightarrow \quad 1 + 2\lambda$

$\ldots \longrightarrow \quad \lim_{n \mapsto +\infty} \frac{1-\lambda^{n+1}}{1-\lambda} + 2\lambda^{n+1}$

	$P \subseteq_{N,\nu} S$
Sum	$\nu \in \left[0, 2\right)$
Avg	$\nu \in \left[0, 1\right]$
Disc	$\nu \in \left[0, \frac{1}{1-\lambda}\right]$

Towards an algorithm

Example

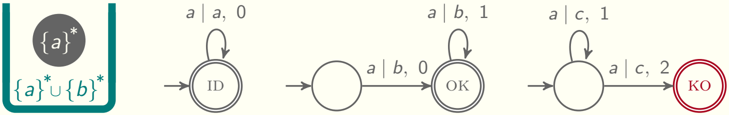


1. identify the erroneous outputs
   automata product
2. compute its smallest value
   Dijkstra, Karp, Alur et al. LATA'13
3. determine its feasibility

	$P \subseteq_{N,\nu} S$
Sum	$\nu \in \left[0, 2\right)$
Avg	$\nu \in \left[0, 1\right]$
Disc	$\nu \in \left[0, \frac{1}{1-\lambda}\right]$

Towards an algorithm

Example



1. identify the erroneous outputs
   automata product
2. compute its smallest value
   Dijkstra, Karp, Alur et al. LATA'13
3. determine its feasibility

	$P \subseteq_{N,\nu} S$
Sum	$\nu \in \left[0, 2\right)$
Avg	$\nu \in \left[0, 1\right]$
Disc	$\nu \in \left[0, \frac{1}{1-\lambda}\right]$

Robust inclusion: PTIME with P, S given as NFA, DFA

Application

Type-1 Diabetes

Hypoglycemia	$<$70 mg/dl glucose levels for $>$3h
Hyperglycemia	$>$300 mg/dl glucose levels for $>$3h

Application

Type-1 Diabetes

Hypoglycemia	$<$70 mg/dl glucose levels for $>$3h
Hyperglycemia	$>$300 mg/dl glucose levels for $>$3h

Noise: Glucose sensor & Symptoms

Application

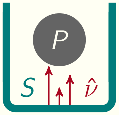
Type-1 Diabetes

Hypoglycemia	$<$70 mg/dl glucose levels for $>$3h
Hyperglycemia	$>$300 mg/dl glucose levels for $>$3h

Noise: Glucose sensor & Symptoms

Threshold Synthesis

- **P**atient glucose record
- **S**ymptoms



Robust kernel

Sum operation

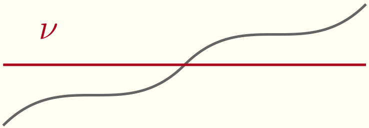
Robust kernel is regular

$$\hat{K} = \{u_{in} \in P \mid \forall u_{out}\ N(u_{in}, u_{out}) \leq \nu \Rightarrow u_{out} \in S\}$$
$$\overline{\hat{K}} = \{u_{in} \in P \mid \exists u_{out}\ N(u_{in}, u_{out}) \leq \nu \wedge u_{out} \notin S\}$$
$$\overline{\hat{K}} = N^{\leq \nu}_{P, \not S}$$

Sum operation

Robust kernel is regular

$$\hat{K} = \{u_{in} \in P \mid \forall u_{out}\ N(u_{in}, u_{out}) \leq \nu \Rightarrow u_{out} \in S\}$$
$$\overline{\hat{K}} = \{u_{in} \in P \mid \exists u_{out}\ N(u_{in}, u_{out}) \leq \nu \wedge u_{out} \notin S\}$$
$$\overline{\hat{K}} = N^{\leq \nu}_{P,\not{S}}$$



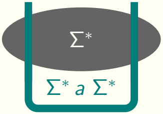
Cost at runtime

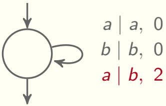
Weights in $\mathbb{N}$
non-decreasing cost

Kernel emptiness is PSPACE-C

Avg operation

Robust kernel is not regular





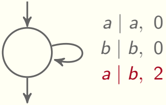
Let $\nu = 1$

At most half the input swap

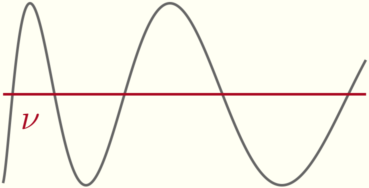
One a remain if $\#(a) > \#(b)$

Avg operation

Robust kernel is not regular



Let $\nu = 1$
At most half the input swap
One a remain if $\#(a) > \#(b)$

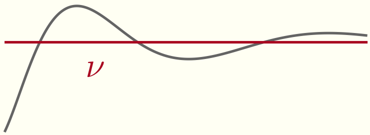


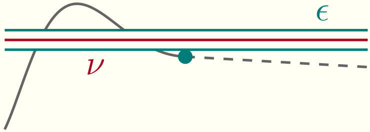
Cost at runtime

Threshold crossed
arbitrarily many times

Kernel emptiness is undecidable

Discounted-sum operation





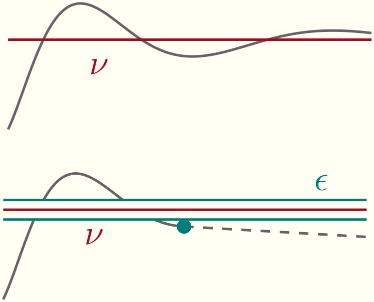
Cost at runtime

Automata open problem

$$\forall u \quad A_{Disc}(u) \leq \nu$$

Henzinger et al. CSL'08 & LICS'15

Discounted-sum operation



Cost at runtime

Automata open problem

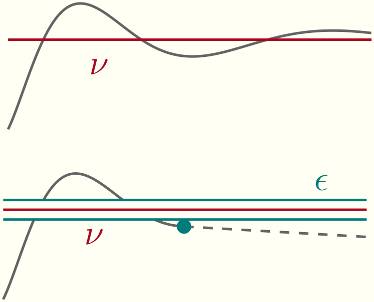
$$\forall u \quad A_{Disc}(u) \leq \nu$$

Henzinger et al. CSL'08 & LICS'15

ν-Isolation hypothesis

$$\exists \epsilon \forall u \quad \begin{cases} \nu - \epsilon > A_{Disc}(u) \\ \nu + \epsilon < A_{Disc}(u) \end{cases}$$

Discounted-sum operation



Cost at runtime

Automata open problem

$$\forall u \quad A_{Disc}(u) \leq \nu$$

Henzinger et al. CSL'08 & LICS'15

ν-Isolation hypothesis

$$\exists \epsilon \forall u \quad \begin{cases} \nu - \epsilon > A_{Disc}(u) \\ \nu + \epsilon < A_{Disc}(u) \end{cases}$$

Regular ν-isolated Kernel

$$\overline{\hat{K}} = N^{\leq \nu}_{P,\not\$}$$

Kernel emptiness is decidable

Emmanuel Filiot*

Jean-François Raskin*

Nicolas Mazzocchi*

Sriram Sankaranarayanan†

Ashutosh Trivedi†

* Université libre de Bruxelles

† University Colorado Boulder

CONCUR 2020

Weighted Transducers for Robustness Verification