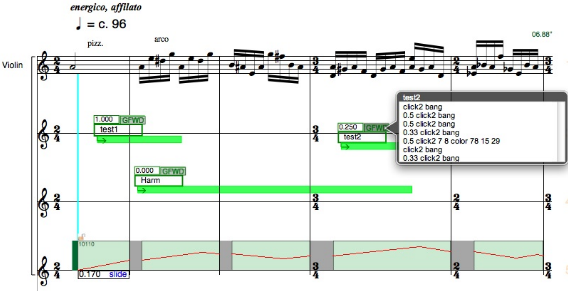
Formal Timed Analysis of Mixed Music Scores

Léa Fanchon, Florent Jacquemard

Ircam & INRIA

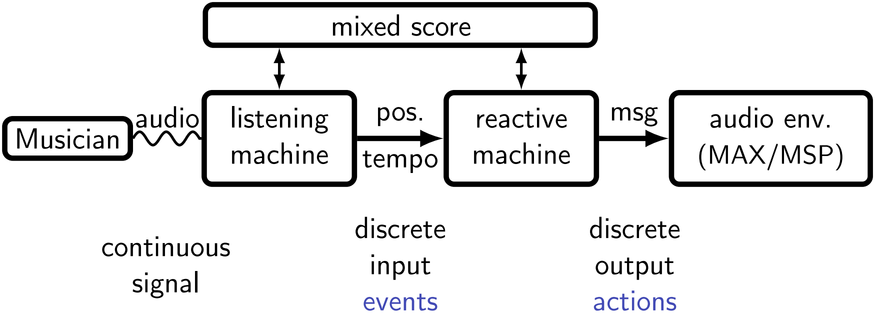
ICMC 2013 – Perth

Writing Mixed Music



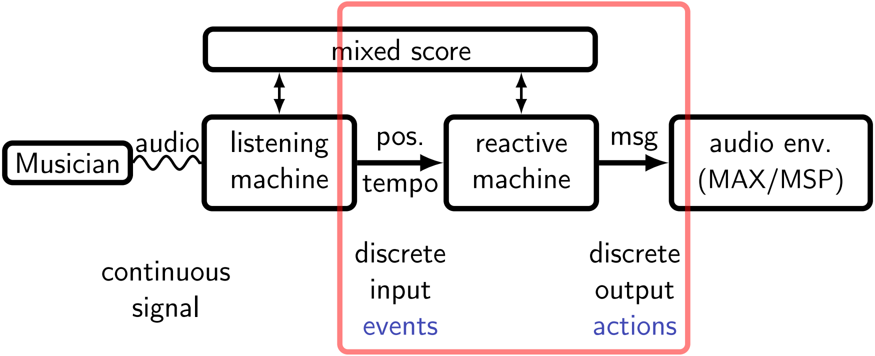
Antescofo (*Anticipatory Score Following*, Ircam 2007-2013)

http://repmus.ircam.fr/antescofo



Antescofo (*Anticipatory Score Following*, Ircam 2007-2013)

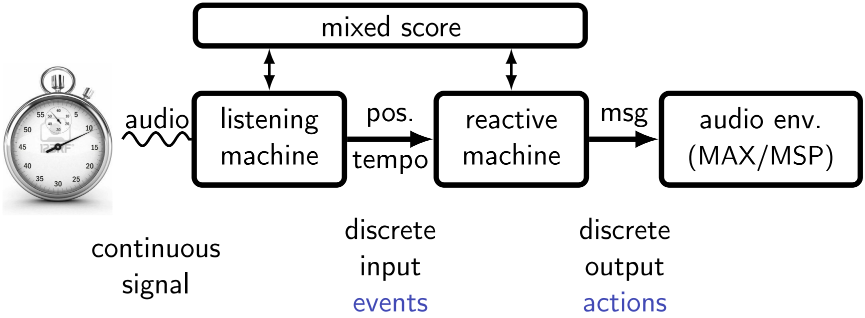
http://repmus.ircam.fr/antescofo



interpretation trace =
interleaving of input events and output actions with dates

"Ideal" Trace

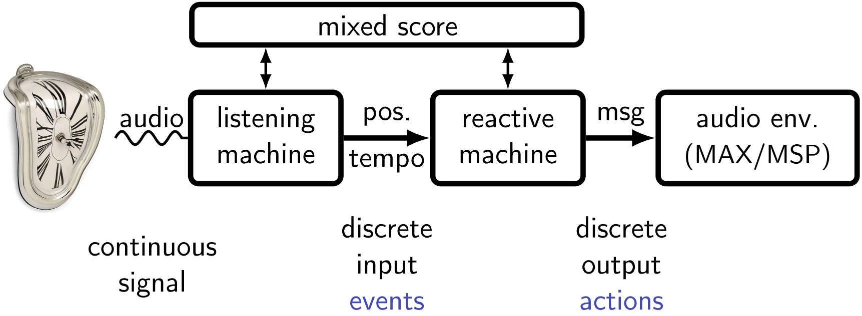
http://repmus.ircam.fr/antescofo



ideal trace $\simeq$ the mixed score

Real Trace

http://repmus.ircam.fr/antescofo



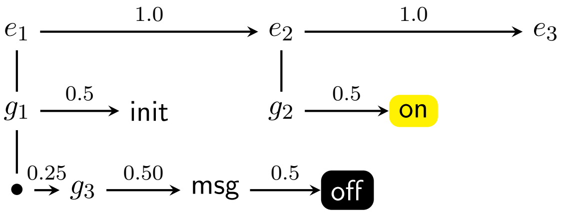
real trace ≠ ideal trace

Example

Microscore

note e_1 1.0
0.0 group g_1 $\{(0.5 \text{ init})\}$
0.25 group g_3 $\{(0.5 \text{ msg})\ (0.5 \text{ off})\}$
note e_2 1.0
0.0 group g_2 $\{(0.5 \text{ on})\}$
note e_3 0.5

Ideal performance

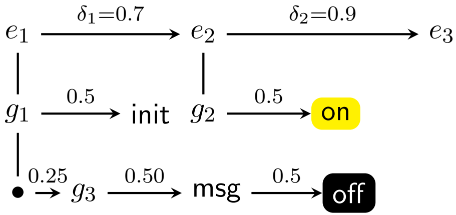


Example

Microscore

note e_1 1.0
0.0 group g_1 $\{(0.5 \text{ init})\}$
0.25 group g_3 $\{(0.5 \text{ msg}) \ (0.5 \text{ off})\}$
note e_2 1.0
0.0 group g_2 $\{(0.5 \text{ on})\}$
note e_3 0.5

Unexpected performance



Objective: Parameter Synthesis

Static Analysis of Mixed Music Scores

Compute a linear constraint K on δ_1 and δ_2, ensuring that

- trace = ideal trace, or
- on after off in trace.

K = indication of the robustness of mixed scores to the tempo variations during performances.

Computed using software verification techniques developed for realtime embedded software based on timed automata model.

Outline

1. Timed Automata: an abstract model for realtime systems
2. Compilation of mixed scores into timed automata
3. Parameter Synthesis
4. Perspectives

1. Timed Automata

Timed Automata

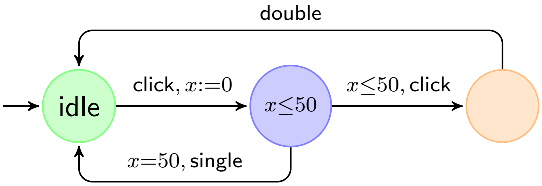
[Alur & Dill 1990]

- $+$ describes logical ordering of events of the system
- $+$ quantitative timing info (duration and time between events)

modeling realtime systems: transport, embedded systems, communication networks, manufacturing, circuits... for

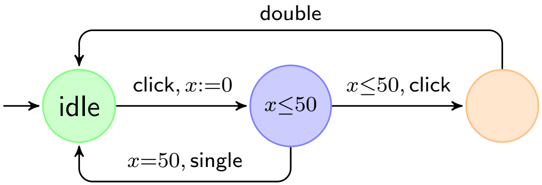
- $\rightarrow$ simulation
- $\rightarrow$ verification
- $\rightarrow$ optimization
- $\rightarrow$ controller synthesis

Timed Automata Example



- x is a clock
- transitions: location$_1$ $\xrightarrow{guard,\, symbol,\, \{clocks\ to\ reset\}}$ location$_2$
- $symbol$ belongs to a finite alphabet
- clock constraints $e := x\{<, \leq, \geq, >\}c \mid e \wedge e \mid true$, $c \in \mathbb{Z}$, for
  - transitions' guards
  - locations' invariants (condition to stay)

Timed Automata Example



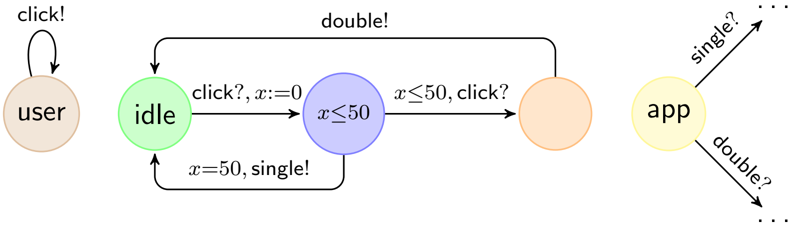
trace:

$$\begin{array}{ccccccccccccc} \mathbf{1} & \xrightarrow{40} & \mathbf{1} & \xrightarrow{\text{click}} & \mathbf{2} & \xrightarrow{25} & \mathbf{2} & \xrightarrow{\text{click}} & \mathbf{3} & \xrightarrow{\text{double}} & \mathbf{1} \\ x = 0 & & 40 & & 0 & & 25 & & 25 & & 25 \\ t = 0 & & 40 & & 40 & & 65 & & 65 & & 65 \end{array}$$

untimed trace: click click double

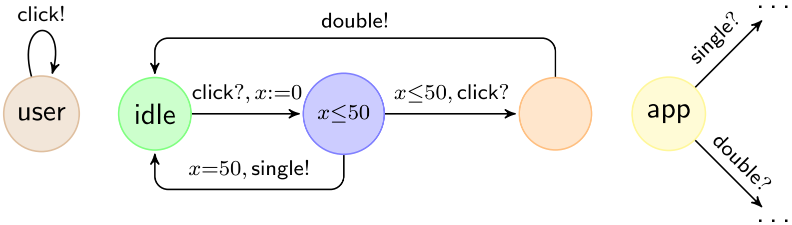
Timed Automata Networks

$\mathcal{A}_1 \parallel \ldots \parallel \mathcal{A}_n$ with disjoint locations and clocks
non-disjoint alphabets $\Sigma_1, \ldots, \Sigma_n$.



Timed Automata Networks

$\mathcal{A}_1 \parallel \ldots \parallel \mathcal{A}_n$ with disjoint locations and clocks
non-disjoint alphabets $\Sigma_1, \ldots, \Sigma_n$.



Cartesian product: $\langle \ell_1, \ldots, \ell_n \rangle \xrightarrow{g,a,R} \langle \ell'_1, \ldots, \ell'_n \rangle$ such that

- for all $1 \leq i \leq n$, if $a \notin \Sigma_i$, then $\ell'_i = \ell_i$
  otherwise there exists g_i and R_i such that $\ell_i \xrightarrow{g_i,a,R_i} \ell'_i \in \mathcal{A}_i$
- $g = \bigwedge_{a \in \Sigma_i} g_i$ and $R = \bigcup_{a \in \Sigma_i} R_i$

Timed Automata

infinite state model

state: $\langle \ell, v \rangle$ where ℓ location and v : clocks $\to \mathbb{R}^+$

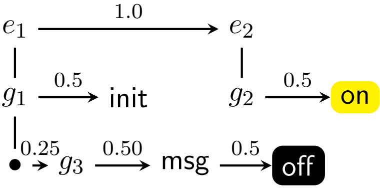
but... region abstraction: transformable into NFA

+ decision of reachability in EXPTIME, PSPACE-complete
+ symbolic algorithms (zones): Uppaal, HyTech
– $\subseteq$ of traces languages undecidable
– no closure under complement
– not determinizable

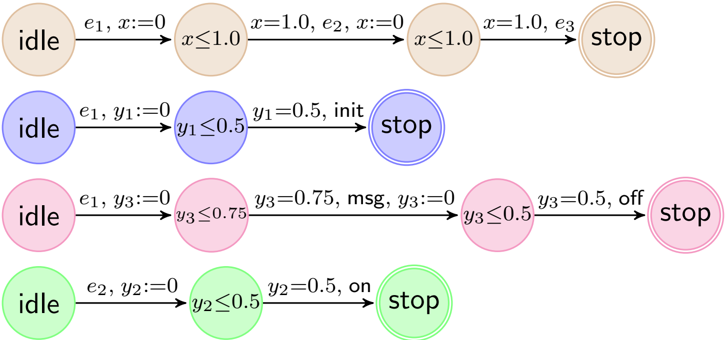
2. Compilation of Antescofo Scores into Timed Automata Automata Networks

Compilation: Antescofo Scores → Timed Automata

note e_1 1.0
0.0 group g_1 $\{$(0.5 init)$\}$
0.25 group g_3 $\{$(0.5 msg) (0.5 off)$\}$
note e_2 1.0
0.0 group g_2 $\{$(0.5 on)$\}$
note e_3 0.5

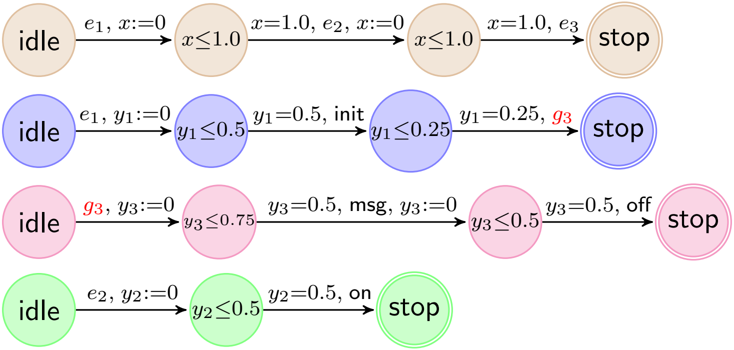


Timed Automata Network



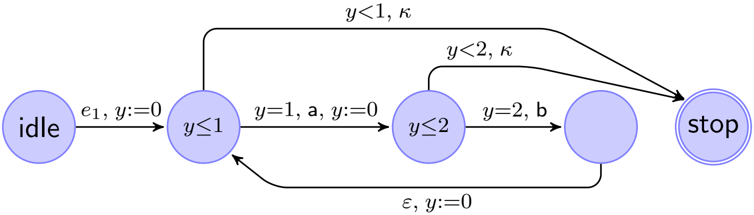
Nested Groups

note e_1 1.0
0.0 group g_1 $\{(0.5 \text{ init})$
$(0.25 \text{ group } g_3 \{(0.5 \text{ msg})\ (0.5 \text{ off})\})\}$
note e_2 1.0
0.0 group g_2 $\{(0.5 \text{ on})\}$
note e_3 0.5



Loops

$0 \quad \text{loop} \quad \ell \quad \{(1\ \mathsf{a})\ (2\ \mathsf{b})\}\ \text{until}\ \kappa$



Group Attribute "tight"

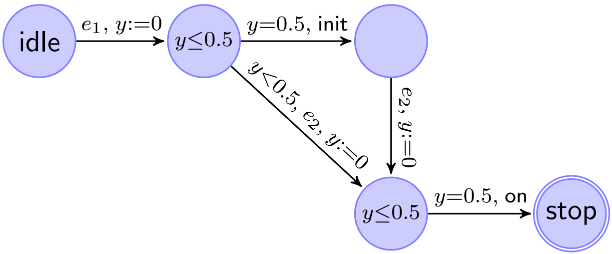
Score

note e_1 1.0
0.0 group g_1 {(0.5 init)}
0.25 group g_3 {(0.5 msg) (0.5 off) }
note e_2 1.0
0.0 group g_2 {(0.5 on)}
note e_3 0.5

Almost equivalent score

note e_1 1.0
0.0 group g[tight] {(0.5 init) (1.0 on) }
0.25 group g_3 {(0.5 msg) (0.5 off) }
note e_2 1.0
note e_3 0.5

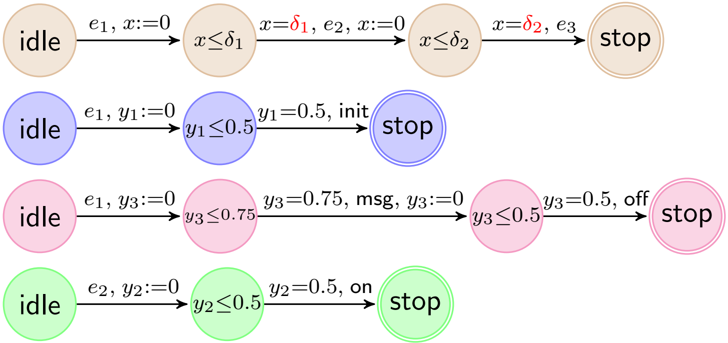
Timed Automaton for g



3. Parameter Synthesis

Parametric Timed Automata

[Alur,Henzinger,Vardi 1993]
parameters δ for (unknown) time bounds in guards and invariants.



Valuation of Parameters δ_1, δ_2 = performance

Good Parameters Problem

[Frehse, Jha, Krogh 2008]
Find a set of parameter values within a rectangular domain $\subseteq \mathbb{R}_+^m$ for which the behavior of $\mathcal{A}$ is acceptable.

i.e. find a linear constraint K on parameters such that every for valuation of parameters π satisfying K, the behavior of $\mathcal{A}$ instantiated by π ($\mathcal{A}[\pi]$) is acceptable.

Symbolic Semantics of Parametric Timed Automata

$\mathcal{A}(K)$: PTA with a constraint K over parameters.

symbolic states $\langle \ell, C \rangle$ where ℓ location, C constraint over clocks and parameters.

initial state $\langle \ell_0, C_0 \rangle$ with $C_0 = K \wedge inv(\ell_0) \wedge \bigwedge_{i=1}^{p-1} x_{i+1} = x_i$,

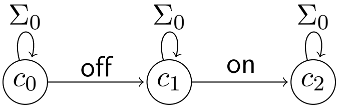
transitions $\langle \ell, C \rangle \xrightarrow{a} \langle \ell', C' \rangle$ such that $\exists\ \ell \xrightarrow{g,a,Reset} \ell'$ in $\mathcal{A}(K)$ and

$$
\begin{array}{rcl}
C'(\boldsymbol{x}') & = & \exists \boldsymbol{x}, d \quad C(\boldsymbol{x}) \wedge g(\boldsymbol{x} + d) \\
& & \wedge \quad \bigwedge_{x_i \in Reset} x'_i = 0 \wedge \bigwedge_{x_i \notin Reset} x'_i = x_i + d \wedge inv(\ell')(\boldsymbol{x}')
\end{array}
$$

where $\boldsymbol{x} = x_1, \ldots, x_p$, and $\boldsymbol{x}' = x'_1, \ldots, x'_p$.

3.2 Naive Approach

Definition of acceptable behavior by a control automaton $\mathcal{C}$



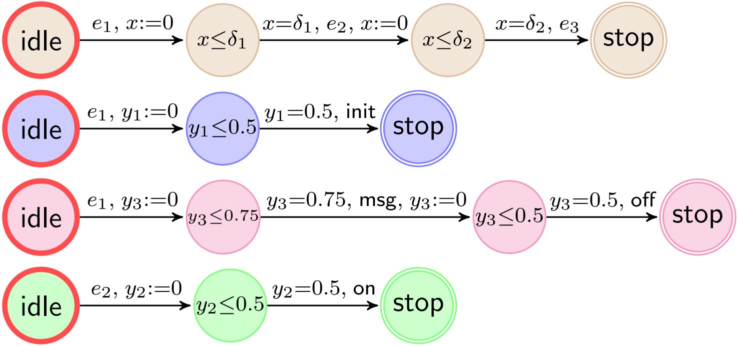
$\Sigma_0 = \{e_1, e_2, e_3, \text{init}, \text{msg}\}$.

Naive Algorithm

Depth first exploration of the symbolic state space of $\mathcal{A} \parallel \mathcal{C}$.
Prune branches with unsatisfiable constraints.

constraint $K =$
disjunction of the constraints of final symbolic states reached.

Depth First Symbolic State Exploration (example 1)

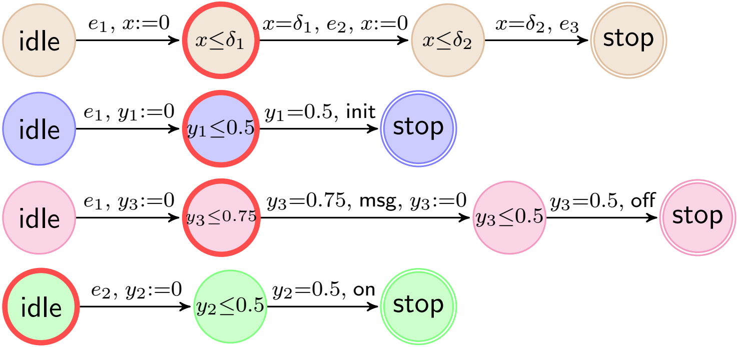


untimed trace:

constraint:

1

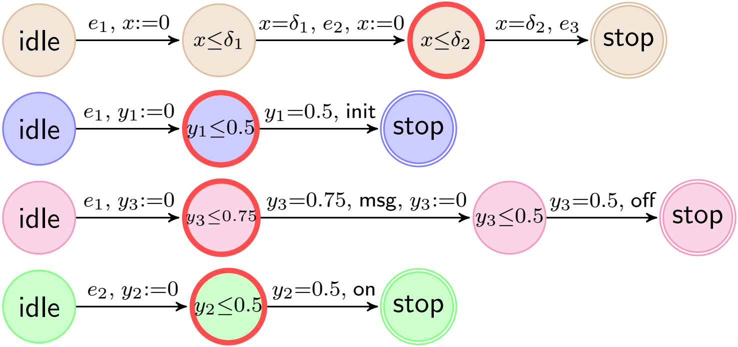
Depth First Symbolic State Exploration (example 1)



untimed trace: e_1

constraint:

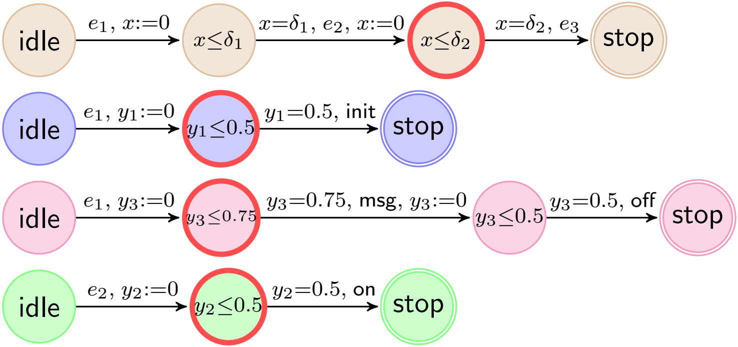
Depth First Symbolic State Exploration (example 1)



untimed trace: e_1, e_2

constraint: $\exists y_1, y_3 \,.\, y_1 = \delta_1 \le 0.5 \land y_3 = \delta_1 \le 0.75$

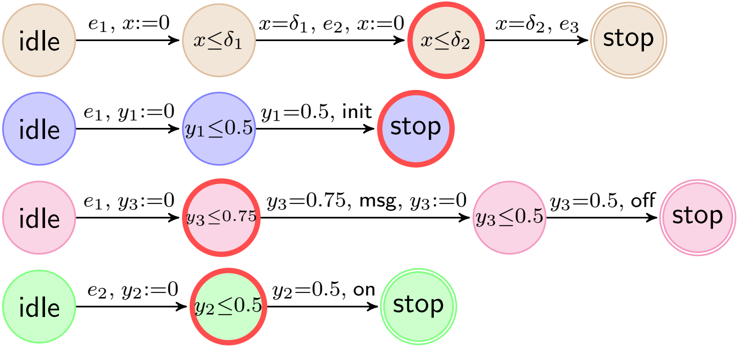
Depth First Symbolic State Exploration (example 1)



untimed trace: e_1, e_2

constraint: $\delta_1 \le 0.5$

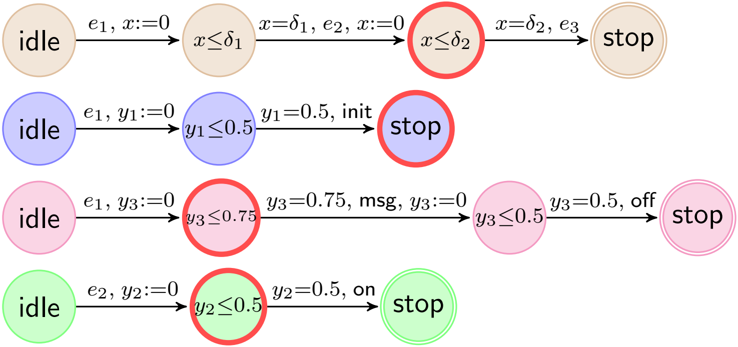
Depth First Symbolic State Exploration (example 1)



untimed trace: $e_1, e_2,$ init

constraint: $\delta_1 \le 0.5 \land \exists x \,.\, x = 0.5 - \delta_1 \le \delta_2$

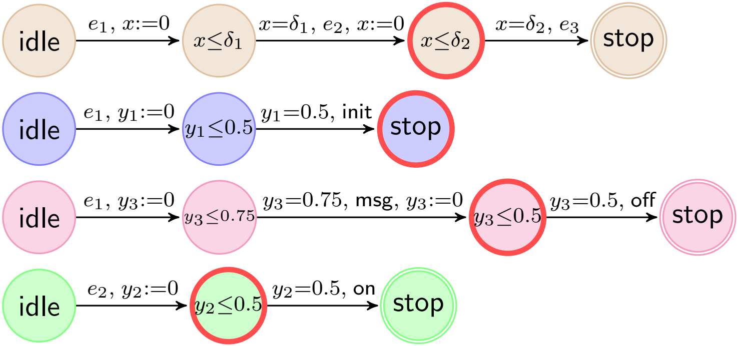
Depth First Symbolic State Exploration (example 1)



untimed trace: $e_1, e_2,$ init

constraint: $\delta_1 \leq 0.5 \wedge 0.5 - \delta_2 \leq \delta_1$

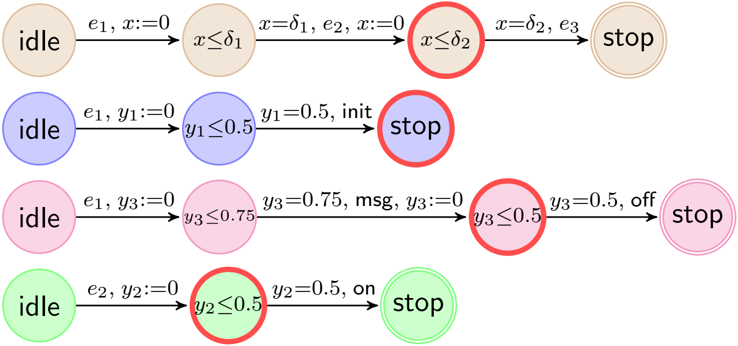
Depth First Symbolic State Exploration (example 1)



untimed trace: $e_1, e_2, \mathsf{init}, \mathsf{msg}$

constraint: $\delta_1 \leq 0.5 \wedge 0.5 - \delta_2 \leq \delta_1 \wedge$
$\exists x, y_2 \,.\, x = 0.75 - \delta_1 \leq \delta_2 \wedge y_2 = 0.75 - \delta_1 \leq 0.5$

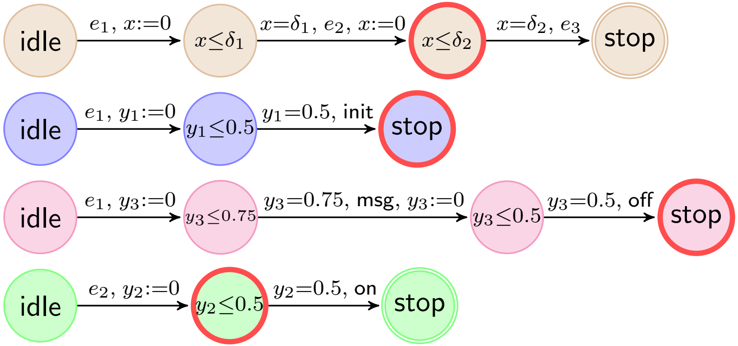
Depth First Symbolic State Exploration (example 1)



untimed trace: $e_1, e_2, \text{init}, \text{msg}$

constraint: $\delta_1 \leq 0.5 \wedge 0.75 - \delta_2 \leq \delta_1 \wedge 0.25 \leq \delta_1$

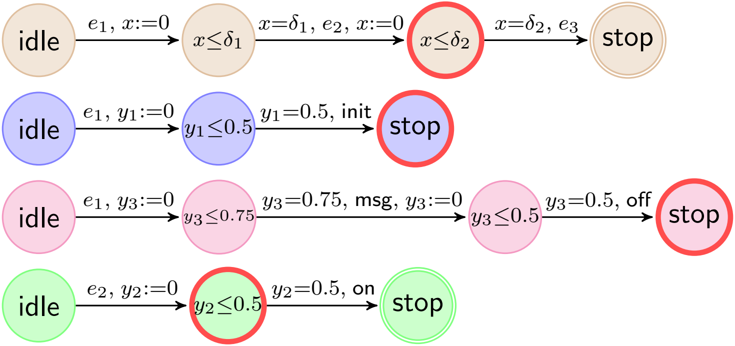
Depth First Symbolic State Exploration (example 1)



untimed trace: $e_1, e_2,$ init, msg, off

constraint: $\delta_1 \leq 0.5 \wedge 0.75 - \delta_2 \leq \delta_1 \wedge 0.25 \leq \delta_1 \wedge$
$\exists x, y_2 \,.\, x = 1.25 - \delta_1 \leq \delta_2 \wedge y_2 = 1.25 - \delta_1 \leq 0.5$

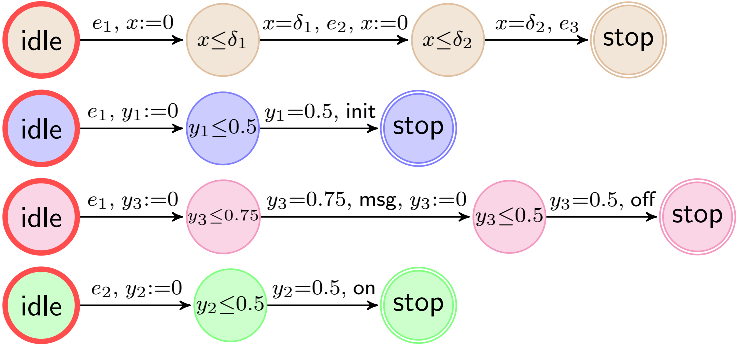
Depth First Symbolic State Exploration (example 1)



untimed trace: $e_1, e_2, \mathsf{init}, \mathsf{msg}, \mathsf{off}$

constraint: $\delta_1 \leq 0.5 \wedge 0.75 - \delta_2 \leq \delta_1 \wedge 0.25 \leq \delta_1 \wedge$
$1.25 - \delta_2 \leq \delta_1 \wedge 0.75 \leq \delta_1$ unsatisfiable!

Depth First Symbolic State Exploration (example 2)

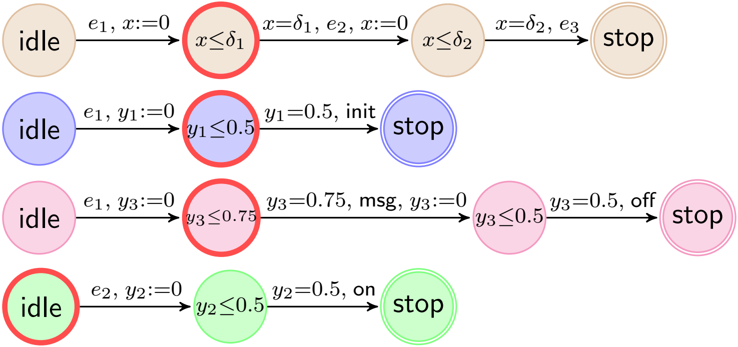


untimed (ideal) trace:

constraint:

1

Depth First Symbolic State Exploration (example 2)

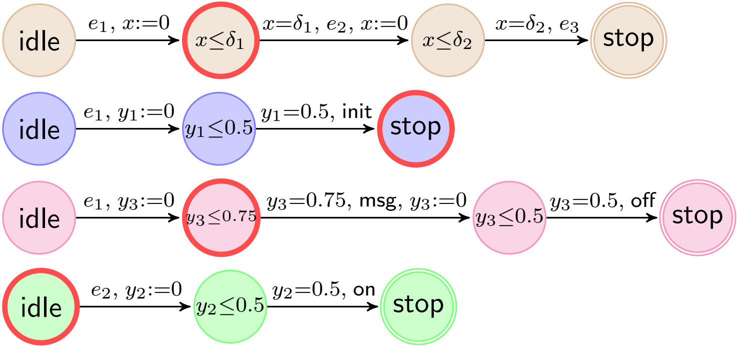


untimed (ideal) trace: e_1

constraint:

2

Depth First Symbolic State Exploration (example 2)

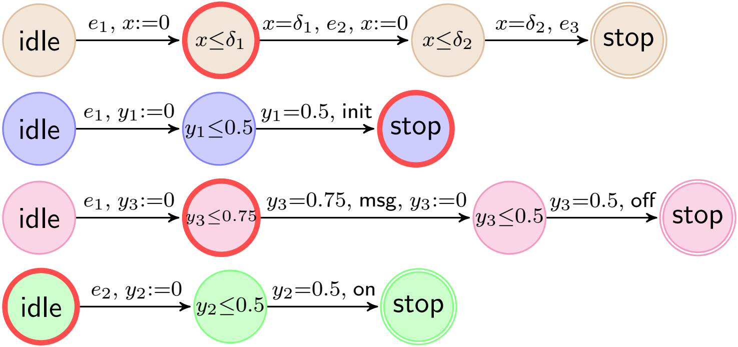


untimed (ideal) trace: e_1, init

constraint: $\exists x \,.\, x = 0.5 \leq \delta_1$

3

Depth First Symbolic State Exploration (example 2)

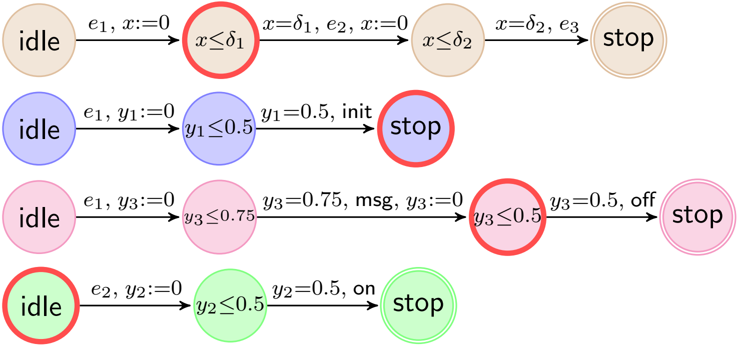


untimed (ideal) trace: e_1, init

constraint: $0.5 \leq \delta_1$

4

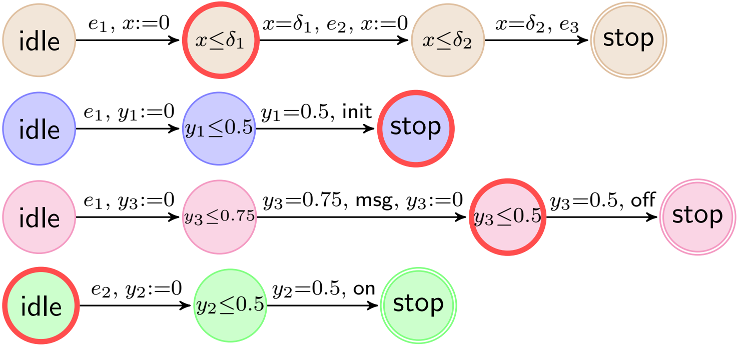
Depth First Symbolic State Exploration (example 2)



untimed (ideal) trace: e_1, init, msg

constraint: $0.5 \leq \delta_1 \wedge \exists x \,.\, x = 0.75 \leq \delta_1$

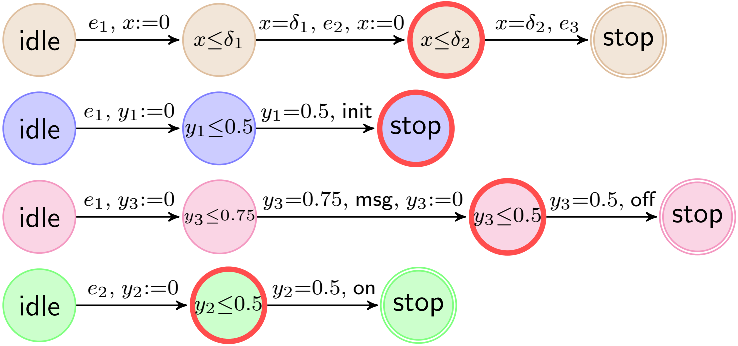
Depth First Symbolic State Exploration (example 2)



untimed (ideal) trace: $e_1, \text{init}, \text{msg}$

constraint: $0.75 \leq \delta_1$

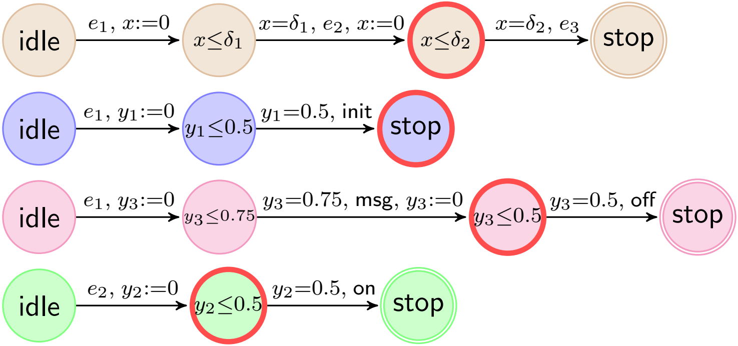
Depth First Symbolic State Exploration (example 2)



untimed (ideal) trace: $e_1, \text{init}, \text{msg}, e_2$

constraint: $0.75 \leq \delta_1 \wedge \exists y_3 \,.\, y_3 = \delta_1 - 0.75 \leq 0.5$

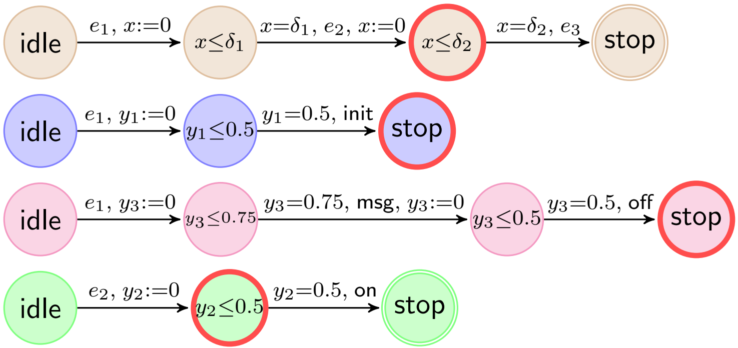
Depth First Symbolic State Exploration (example 2)



untimed (ideal) trace: $e_1, \text{init}, \text{msg}, e_2$

constraint: $0.75 \leq \delta_1 \wedge \delta_1 \leq 1.25$

Depth First Symbolic State Exploration (example 2)

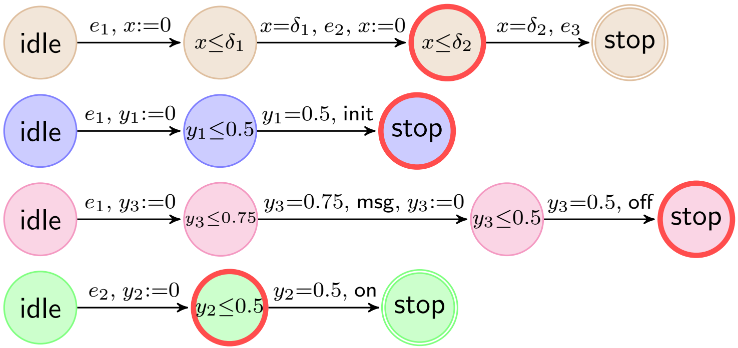


untimed (ideal) trace: e_1, init, msg, e_2, off

constraint: $0.75 \leq \delta_1 \wedge \delta_1 \leq 1.25 \wedge$

$\exists x, y_2 \,.\, x = 1.25 - \delta_1 \leq \delta_2 \wedge y_2 = 1.25 - \delta_1 \leq 0.5$

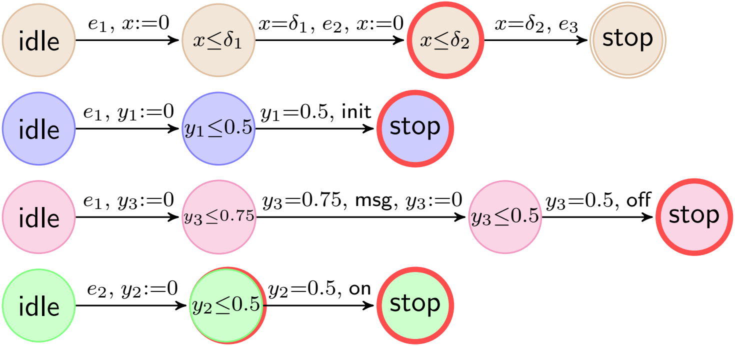
Depth First Symbolic State Exploration (example 2)



untimed (ideal) trace: $e_1, \text{init}, \text{msg}, e_2, \text{off}$

constraint: $0.75 \leq \delta_1 \wedge \delta_1 \leq 1.25 \wedge 1.25 - \delta_2 \leq \delta_1$

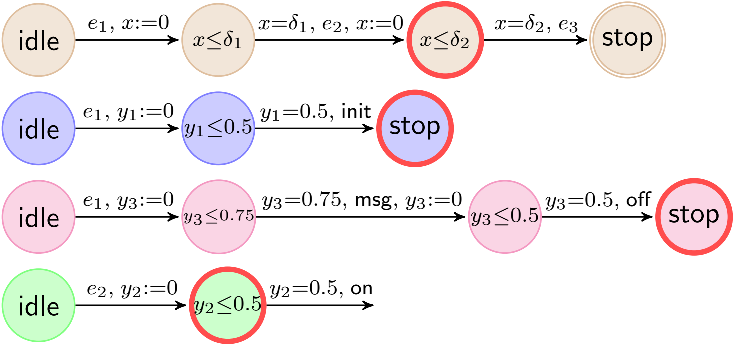
Depth First Symbolic State Exploration (example 2)



untimed (ideal) trace: e_1, init, msg, e_2, off, on

constraint: $0.75 \leq \delta_1 \wedge \delta_1 \leq 1.25 \wedge 1.25 - \delta_2 \leq \delta_1 \wedge \exists x \, . \, x = 0.5 \leq \delta_2$

Depth First Symbolic State Exploration (example 2)



untimed (ideal) trace: $e_1, \text{init}, \text{msg}, e_2, \text{off}, \text{on}$

constraint: $0.75 \leq \delta_1 \wedge \delta_1 \leq 1.25 \wedge 0.5 \leq \delta_2$

3.2 Inverse Method

[André phd 2010, Soulat phd 2012, Fribourg]
Given a reference valuation π_0 of the parameters,
synthesize a constraint K_0 on the parameters guaranteeing
the same untimed traces as with π_0.

start with $K := true$
computes iteratively a set S of reachable states of $\mathcal{A}(K)$
when $\langle \ell, C \rangle \in S$ and $\pi_0 \not\models \exists \boldsymbol{x}\, C(\boldsymbol{x})$
then $K := K \wedge \neg J$ for chosen π_0-incompatible conjunct J.

tool IMITATOR: fixpoint after 6 iterations on example
returns $K = 0.75 < \delta_1 \leq 1.0 \wedge 1.25 < \delta_1 + \delta_2$.

Conclusion

- compilation of Antescofo scores into a formal model.
- static analysis to characterize the range of performances for which the reactive engine of Antescofo will behave as expected.
- outcome of analysis:
  linear constraints on performance delays
  - warn performers about dangerous parts in the score
  - assistance for adjusting delays and structuring groups
  - user friendly presentation on score?

not seen / ongoing / perspectives

- Dealing with error-handling instructions (missed events)
- Non-linear scores
- Game theoretic approach for a quantitative estimation of robustness

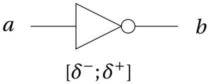
Thank you!

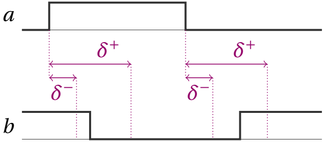
Perspectives: Verification

- delays = linear expressions on parameters
  (variables in scores)
- complexity wrt restrictions
- reachability analysis or timed model checking
  over the unparametric TA model
  (verification of properties of idealized performance)
- timed games: controller synthesis, dynamic scheduling
  open scores (straightforward extension)

Parametric Timed Automata for Circuits

[Alur,Henzinger,Vardi 1993] parameters δ for (unknown) time bounds in guards and invariants





[André 2010]