ANALYZING TIMED SYSTEMS USING TREE AUTOMATA

S. AKSHAY, IIT BOMBAY
PAUL GASTIN, LSV ENS PARIS-SACLAY (ENS CACHAN)
S. KRISHNA, IIT BOMBAY

CONCUR 2016
TIMED AUTOMATON

TIMED WORD

TIMED WORD LANGUAGE

NON-EMPTINESS / REACHABILITY PROBLEM
EMPTINESS FOR (PUSHDOWN) TIMED AUTOMATON

- Well-studied problem with standard approach
  - Timed automata (TA): Region construction [Alur-Dill’90]
    Many optimizations
  - Pushdown timed automata (PDTA): Lifting region construction [Bouajjani et al. ’94] [Abdulla et al. ’12]
- Common feature:
  - represent behaviors as timed words
  - use abstractions to reduce to finite automata

\[
\begin{align*}
  y \leq 2 & \quad x \geq 1 \\
  a & \quad y := 0 \\
  x \leq 3 & \quad \text{push}() \\
  2 \leq \hat{x} \leq 4 & \quad \text{pop} \\
  y := 0 & \quad c \\
  y \geq 1 & \quad \text{c}
\end{align*}
\]
EMPTINESS FOR (PUSHDOWN) TIMED AUTOMATON

- Well-studied problem with standard approach
  - Timed automata (TA): Region construction [Alur-Dill’90]
    Many optimizations
  - Pushdown timed automata (PDTA): Lifting region construction
    [Bouajjani et al. ‘94] [Abdulla et al. ’12]
- Common feature:
  - represent behaviors as timed words
  - use abstractions to reduce to finite automata
- Our new approach
  - represent behaviors as graphs: words with timing constraints
  - Interpret graphs in trees to reduce to tree automata
    - High level and powerful technique
    - Simpler and uniform proofs for more complicated systems
    - New technique not relying on regions/zones
OUTLINE

▸ BEHAVIOURS AS GRAPHS

▸ DECIDING GRAPH PROPERTIES

▸ DEFINABILITY OF PROPERTIES FOR TIMED SYSTEMS

▸ TREE-WIDTH FOR TIMED SYSTEMS

▸ INTERPRETING GRAPHS IN TREES

▸ CONCLUSION
BEHAVIORS AS GRAPHS: TIMED SYSTEMS

\[
\begin{align*}
  &y \leq 2 \\
  &x \geq 1 \\
  &x \geq 4 \\
  &x \leq 3 \\
  &y := 0 \\
  &y := 0 \\
  &y \geq 1
\end{align*}
\]

TC-WORDS: WORDS WITH TIMING CONSTRAINTS
BEHAVIORS AS GRAPHS: TIMED SYSTEMS

TC-WORDS: WORDS WITH TIMING CONSTRAINTS

TC-WORD LANGUAGE: $\mathcal{L}_{TCW}(\mathcal{A})$

- Every accepting path $\rho$ in the timed system generates one TC-word $tcw(\rho) \in \mathcal{L}_{TCW}(\mathcal{A})$
BEHAVIORS AS GRAPHS: TIMED SYSTEMS

TC-WORDS: WORDS WITH TIMING CONSTRAINTS

TC-WORD LANGUAGE: $\mathcal{L}_{TCW}(A)$

REALIZABLE TC-WORDS: $\text{Real}_{TCW}$
BEHAVIORS AS GRAPHS: TIMED SYSTEMS

TC-WORDS: WORDS WITH TIMING CONSTRAINTS

TC-WORD LANGUAGE:

REALIZABLE TC-WORDS:
REALIZATIONS OF TC-WORDS

ANALYZING TIMED SYSTEMS USING TREE AUTOMATA

TC-WORD

REALIZATION

TIMED-WORDS
TIMED WORDS vs TC-WORDS

TIMED-WORDS
- Uncountably many realizations
- No realizations
- Words over an infinite alphabet

TC-WORDS
- One TC-word
- Graphs over a finite signature

\[ \mathcal{L}_T(A) = \text{Realizations}(\mathcal{L}_{TCW}(A)) \]

\[ \mathcal{L}_T(A) \neq \emptyset \iff \mathcal{L}_{TCW}(A) \cap \text{Real}_{TCW} \neq \emptyset \]
**TIMED WORDS vs TC-WORDS**

**TIMED-WORDS**
- Uncountably many realizations
- No realizations
- Words over an infinite alphabet

**TC-WORDS**
- One TC-word
- Graphs over a finite signature

\[ \mathcal{L}_T(A) = \text{Realizations}(\mathcal{L}_{TCW}(A)) \]

\[ \mathcal{L}_T(A) \neq \emptyset \quad \iff \quad \mathcal{L}_{TCW}(A) \cap \text{Real}_{TCW} \neq \emptyset \]

This is a graph property!
OUTLINE

▸ BEHAVIOURS AS GRAPHS
▸ DECIDING GRAPH PROPERTIES
▸ DEFINABILITY OF PROPERTIES FOR TIMED SYSTEMS
▸ TREE-WIDTH FOR TIMED SYSTEMS
▸ INTERPRETING GRAPHS IN TREES
▸ CONCLUSION
**COURCELLE’S THEOREM**

- Let $TW_k$ be the set of graphs of tree-width at most $k$
- Let $P$ be a property of graphs
- If $P$ is MSO-definable then $P \cap TW_k \neq \emptyset$ is decidable
- Graphs in $TW_k$ can be interpreted in trees (k-terms)
- Let $P$ be an MSO-definable property of graphs
- $\Phi_P$ MSO over graphs $\iff$ $\Phi^k_P$ MSO over trees (k-terms)
- Then $P \cap TW_k \neq \emptyset$ iff $\Phi^k_P$ satisfiable over trees (k-terms)
- **THATCHER&WRIGHT’68**: REDUCTION TO EMPTINESS OF TREE AUTOMATA
COURCELLE’S THEOREM

- Let $TW_k$ be the set of graphs of tree-width at most $k$
- Let $P$ be a property of graphs
- If $P$ is MSO-definable then $P \cap TW_k \neq \emptyset$ is decidable

- Graphs in $TW_k$ can be interpreted in trees ($k$-terms)
- Let $P$ be a property of graphs
- Build directly a tree automaton $A^k_P$ accepting $k$-terms denoting graphs satisfying $P$
- Then $P \cap TW_k \neq \emptyset$ iff $L(A^k_P) \neq \emptyset$

CONCUR’16
We want to solve \( \mathcal{L}_{TCW}(A) \cap \text{Real}_{TCW} \neq \emptyset \):

- Show that TC-words have bounded tree-width
- Show that our properties are MSO-definable
- Build directly tree automata for our properties

Courcelle’s Theorem:

- Let \( TW_k \) be the set of graphs of tree-width at most \( k \)
- Let \( P \) be a property of graphs
- If \( P \) is MSO-definable then \( P \cap TW_k \neq \emptyset \) is decidable
ANALYZING TIMED SYSTEMS USING TREE AUTOMATA

RELATED WORKS

- P. Madhusudan & G. Parlato, POPL’11
  The tree-width of auxiliary storage

- C. Aiswarya, PG & K. Narayan Kumar, CONCUR’12
  MSO decidability of multi-pushdown systems via split-width

- C. Aiswarya PhD’14
  Verification of communicating recursive programs via split-width

- C. Aiswarya & PG, FSTTCS’14
  Reasoning about distributed systems: WYSIWYG
OUTLINE

- BEHAVIOURS AS GRAPHS
- DECIDING GRAPH PROPERTIES
- DEFINABILITY OF PROPERTIES FOR TIMED SYSTEMS
- TREE-WIDTH FOR TIMED SYSTEMS
- INTERPRETING GRAPHS IN TREES
- CONCLUSION
MSO-DEFINABLE GRAPH PROPERTIES

\( G = (V, \to) \) IS A WORD: LINEAR ORDER

\[
\text{Word}(\to) ::= \forall x, y, z \left( \neg (x \to^+ x) \land (x = y \lor x \to^+ y \lor y \to^+ x) \land \neg (x \to z \lor x \to y \to^+ z) \right)
\]
MSO-DEFINABLE GRAPH PROPERTIES

$G = (V, \rightarrow, \bowtie)$ IS A 1-CLOCK TC-WORD

Forward ($\bowtie$) $::= \forall x, y \ (x \bowtie y \implies x < y)$

Clock ($\bowtie$) $::= \neg \exists x, y, x', y' \ (x \bowtie y \land x' \bowtie y' \land x < x' < y)$
MSO-DEFINABLE GRAPH PROPERTIES

\( G = ( V, \rightarrow, \rightsquigarrow) \) IS A 1-STACK TC-WORD

Forward (\( \rightsquigarrow \)) ::= \( \forall x, y (x \rightsquigarrow y \implies x < y) \)

Stack (\( \rightsquigarrow \)) ::= \( \neg \exists x, y, x', y' (x \rightsquigarrow y \land x' \rightsquigarrow y' \land (y = x' \lor x \leq x' < y < y' \lor x < x' < y \leq y')) \)
MSO-DEFINABLE GRAPH PROPERTIES

$G = (V, \rightarrow, \curvearrowright)$ IS AN M-STACKS N-CLOCKS TC-WORD

Forward$(\curvearrowright) \land \exists \bar{R} = (R_1, \ldots, R_n) \exists \bar{S} = (S_1, \ldots, S_m)$

Partition$(\curvearrowright, \bar{R}, \bar{S}) \land \bigwedge_{i=1}^{n} \text{Clock}(R_i) \land \bigwedge_{i=1}^{m} \text{Stack}(S_i)$

???
G = (V, →, ∘) is an M-stacks N-clocks TC-word

Forward(♩) ∧ ∃R = (R₁, ..., Rₙ) ∃S = (S₁, ..., Sₘ)

Partition(♩, R, S) ∧ ∨ₙ Clock(Rᵢ) ∧ ∨ₘ Stack(Sᵢ)
ANALYZING TIMED SYSTEMS USING TREE AUTOMATA

MSO-DEFINABLE GRAPH PROPERTIES

\( G = (V, \rightarrow, \curvearrowright) \) IS A TC-WORD ACCEPTED BY A TA \( \mathcal{A} \)

\[ \exists X = (X_\delta)_{\delta \in \Delta} \text{ Partition}(X) \land \text{AcceptingPath}(X) \land \exists (c \curvearrowright)_{c \in \text{Clocks}} \text{ Partition}((\curvearrowright, (c \curvearrowright)_{c \in \text{Clocks}}) \land \forall x, y \left( x \curvearrowright \Rightarrow \text{Reset}_c(x) \land \neg \exists (z \ x < z < y \land \text{Reset}_c(z)) \right) \land \forall y \left( X_\delta(y) \Rightarrow \exists x \left( x \curvearrowright y \land x \curvearrowright^I y \right) \right) \]

\[ \text{Reset}_c(x) ::= \bigvee_{\delta \in \Delta (c:=0) \in \delta} X_\delta(x) \quad \curvearrowright ::= \bigvee_I \curvearrowright^I \]
**MSO-DEFINABLE GRAPH PROPERTIES**

**THEOREM:** REALIZABILITY OF TC-WORDS IS MSO-DEFINABLE

\[ \exists ts : V \rightarrow \mathbb{R}, \ \forall x, y \ (x \sim^I y \implies ts(y) - ts(x) \in I) \land (x \rightarrow y \implies ts(x) \leq ts(y)) \]
**MSO-DEFINABLE GRAPH PROPERTIES**

**THEOREM:** REALIZABILITY OF TC-WORDS IS MSO-DEFINABLE

\[ \exists \text{ts} : V \rightarrow \mathbb{R}, \ \forall x, y \ (x \sim^I y \implies \text{ts}(y) - \text{ts}(x) \in I) \land (x \rightarrow y \implies \text{ts}(x) \leq \text{ts}(y)) \]

\[ \exists \text{tsm} : V \rightarrow [M] = \{0, \ldots, M - 1\}, \ \forall x, y \ x \sim^I y \implies (\text{Big}(x,y) \land I.\uparrow = \infty) \lor (\neg \text{Big}(x,y) \land (\text{tsm}(y) - \text{tsm}(x))[M] \in I) \]

\[ \text{M} = 5 \]

\[ \text{Big}(x,y) = \exists z, z', \ x < z < z' \leq y \land \bigvee_{a,b,c}^{a,b,c} \text{tsm}(x) = a \land \text{tsm}(z) = b \land \text{tsm}(z') = c \]

\[ (b-a)[M] + (c-b)[M] \geq M \]
MSO-DEFINABLE GRAPH PROPERTIES

\( G = (V, \rightarrow, \rightarrow) \) IS REALIZABLE

Realizability is not MSO-definable without the linear order.
COURCELLE’S THEOREM

- Let $TW_k$ be the set of graphs of tree-width at most $k$
- Let $P$ be a property of graphs
- If $P$ is MSO-definable then $P \cap TW_k \neq \emptyset$ is decidable

WE WANT TO SOLVE $\mathcal{L}_{TCW}(A) \cap \text{Real}_{TCW} \neq \emptyset$

- Show that TC-words have bounded tree-width
- Show that our properties are MSO-definable
- Build directly tree automata for our properties
OUTLINE

▸ BEHAVIOURS AS GRAPHS
▸ DECIDING PROPERTIES OF GRAPHS
▸ DEFINABILITY OF PROPERTIES FOR TIMED SYSTEMS
▸ TREE-WIDTH FOR TIMED SYSTEMS
▸ INTERPRETING GRAPHS IN TREES
▸ CONCLUSION
TREE-WIDTH ALGEBRA

\[ \tau ::= i \mid i \rightarrow j \mid f_{g_i}(\tau) \mid \tau \oplus \tau \]

ATOMIC

FORGET
TREE-WIDTH ALGEBRA

\[ \tau ::= i \mid i \rightarrow j \mid f_{g_i}(\tau) \mid \tau \oplus \tau \]

ATOMIC

FORGET
TREE-WIDTH ALGEBRA

\[ \tau ::= i \mid i \xrightarrow{} j \mid f_{g_{i}}(\tau) \mid \tau \oplus \tau \]

**Atomic**

**Forget**

**Combine**
TREE-WIDTH ALGEBRA

\[ \tau ::= i \mid i \rightarrow j \mid f_{g_i}(\tau) \mid \tau \oplus \tau \]

**ATOMIC**

**FORGET**

**COMBINE**
**TREE-WIDTH ALGEBRA**

$\tau ::= i \mid i \rightarrow j \mid f_{g_i}(\tau) \mid \tau \oplus \tau$

$TW_k :$ graphs of tree-width at most $k$

**ATOMIC**

- $\bullet$

**FORGET**

- $\bullet$

**COMBINE**

- $\bigoplus$

Graph $G$ has tree-width at most $k$ if it can be constructed using $k+1$ colors.
\[ \tau ::= i \mid i \rightarrow j \mid \text{fg}_i(\tau) \mid \tau \oplus \tau \]
TREE-WIDTH ALGEBRA

\[ \tau ::= i | i \rightarrow j | \text{fg}_i(\tau) | \tau \oplus \tau \]

GRAPH DECOMPOSITION

FORGET

ADD

COMBINE

DIVIDE
ANALYZING TIMED SYSTEMS USING TREE AUTOMATA

1-STACK TC-WORDS $\subseteq TW_2$

$G = (V, \to, \tau)$ TC-WORD

$\tau ::= i \mid i \to j \mid i \tau \mid f_{g_i}(\tau) \mid \tau \oplus \tau$

Diagram of a 1-stack TC-word with transitions and states.
1-STACK TC-WORDS $\subseteq TW_2$

$G = (V, \rightarrow, \rightsquigarrow)$ TC-WORD

$\tau ::= i \mid i \rightarrow j \mid i \rightsquigarrow j \mid fg_i(\tau) \mid \tau \oplus \tau$
$G = (V, \to, \ derechos) \ TC\text{-}WORD$

1-STACK TC-WORDS $\subseteq TW_2$

$$\tau ::= i \mid i \rightarrow j \mid i \rightsquigarrow j \mid fg_i(\tau) \mid \tau \oplus \tau$$
1-STACK TC-WORDS $\subseteq TW_2$

$$G = (V, \rightarrow, \rightsquigarrow) \text{ TC-WORD}$$

$$\tau ::= i \mid i \rightarrow j \mid i \rightsquigarrow j \mid fg_i(\tau) \mid \tau \oplus \tau$$
1-STACK TC-WORDS $\subseteq TW_2$

$\tau ::= i \mid i \to j \mid i \bowtie j \mid fg_i(\tau) \mid \tau \oplus \tau$

$G = (V, \to, \bowtie)$ TC-WORD
1-STACK TC-WORDS $\subseteq TW_2$

$$\tau ::= i \mid i \rightarrow j \mid i \nuparrow j \mid fg_i(\tau) \mid \tau \oplus \tau$$

$G = (V, \rightarrow, \nuparrow) TC-WORD$
1-STACK TC-WORDS $\subseteq TW_2$

$$G = (V, \rightarrow, \rightsquigarrow) \text{ TC-WORD}$$

$$\tau ::= i | i \rightarrow j | i \rightsquigarrow j | fg_i(\tau) | \tau \oplus \tau$$
1-STACK TC-WORDS $\subseteq \mathcal{TW}_2$

$\mathcal{T} ::= i \mid i \rightarrow j \mid i \bowtie j \mid f_{g_i}(\tau) \mid \tau \oplus \tau$

$\mathbf{G} = (V, \rightarrow, \bowtie) \text{ TC-WORD}$
1-STACK TC-WORDS $\subseteq TW_2$

$G = (V, \rightarrow, \rightsquigarrow)$ TC-WORD

$\tau ::= i \mid i \rightarrow j \mid i \rightsquigarrow j \mid f_{g_i}(\tau) \mid \tau \oplus \tau$
1-STACK TC-WORDS $\subseteq \mathbf{TW}_2$

$$G = (V, \rightarrow, \rightsquigarrow) \text{ TC-WORD}$$

$$\tau ::= i \mid i \rightarrow j \mid i \rightsquigarrow j \mid fg_i(\tau) \mid \tau \oplus \tau$$
1-STACK TC-WORDS $\subseteq \text{TW}_2$

\[
\tau ::= i \ | \ i \rightarrow j \ | \ i \延 \ j \ | \ \text{fg}_i(\tau) \ | \ \tau \oplus \tau
\]
1-STACK TC-WORDS $\subseteq TW_2$

$$ \tau ::= i \mid i \rightarrow j \mid i \prec j \mid \text{fg}_i(\tau) \mid \tau \oplus \tau $$

G = (V, →, ⊢) TC-WORD
1-STACK TC-WORDS $\subseteq \text{TW}_2$

$G = (V, \rightarrow, \rightsquigarrow) \text{ TC-WORD}$

$$\tau ::= i \mid i \rightarrow j \mid i \rightsquigarrow j \mid fg_i(\tau) \mid \tau \oplus \tau$$

2-STACKS $\Rightarrow$ UNBOUNDED TREE-WIDTH
\[ \tau ::= i \mid i \rightarrow j \mid i \rightsquigarrow j \mid fg_i(\tau) \mid \tau \oplus \tau \]
1-STACK $k$-CLOCKS TC-WORDS $\subseteq \mathbf{TW}_{3k+2}$

$G = (V, \to, \to)$ TC-WORD

$$\tau ::= i \mid i \rightarrow j \mid i \leftarrow j \mid f_{g_i}(\tau) \mid \tau \oplus \tau$$
1-STACK $k$-CLOCKS TC-WORDS $\subseteq TW_{3k+2}$

$G = (V, \to, \leadsto)$ TC-WORD

$$\tau ::= i \mid i \to j \mid i \leadsto j \mid fg_i(\tau) \mid \tau \oplus \tau$$
ANALYZING TIMED SYSTEMS USING TREE AUTOMATA

1-STACK $k$-CLOCKS TC-WORDS $\subseteq TW_{3k+2}$

$G = (V, \rightarrow, \Rightarrow)$ TC-WORD

$$\tau ::= i \mid i \rightarrow j \mid i \Rightarrow j \mid fg_i(\tau) \mid \tau \oplus \tau$$
1-STACK $k$-CLOCKS TC-WORDS $\subseteq \mathcal{TW}_{3k+2}$

$G = (V, \to, \rtimes) \text{ TC-WORD}$

$\tau ::= i \mid i \to j \mid i \rtimes j \mid fg_i(\tau) \mid \tau \oplus \tau$
1-STACK $k$-CLOCKS TC-WORDS $\subseteq \text{TW}_{3k+2}$  
$G = (V, \rightarrow, \curvearrowright)$ TC-WORD

$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \text{fg}_i(\tau) \mid \tau \oplus \tau$
1-STACK k-CLOCKS TC-WORDS $\subseteq TW_{3k+2}$

$G = (V, \rightarrow, \tau)$ TC-WORD

$$\tau ::= i | i \rightarrow j | i \curvearrowright j | fg_i(\tau) | \tau \oplus \tau$$
$\mathbf{G} = (V, \rightarrow, \rightsquigarrow) \text{ TC-WORD}$

$\tau ::= i \mid i \rightarrow j \mid i \rightsquigarrow j \mid f_{g_i}(\tau) \mid \tau \oplus \tau$

$1$-STACK $k$-CLOCKS TC-WORDS $\subseteq \text{TW}_{3k+2}$
ANALYZING TIMED SYSTEMS USING TREE AUTOMATA

$1$-STACK $k$-CLOCKS TC-WORDS $\subseteq TW_{3k+2}$

$G = (V, \rightarrow, \mathcal{R})$ TC-WORD

$\tau ::= i \mid i \rightarrow j \mid i \bowtie j \mid fg_i(\tau) \mid \tau \oplus \tau$
ANALYZING TIMED SYSTEMS USING TREE AUTOMATA

1-STACK \( k \)-CLOCKS TC-WORDS \( \subseteq TW_{3k+2} \) \( G = (V, \rightarrow, \rightsquigarrow) \) TC-WORD

\[
\tau ::= i \mid i \rightarrow j \mid i \rightsquigarrow j \mid fg_i(\tau) \mid \tau \oplus \tau
\]
ANALYZING TIMED SYSTEMS USING TREE AUTOMATA

1-STACK $k$-CLOCKS TC-WORDS $\subseteq \text{TW}_{3k+2}$

$G = (V, \rightarrow, \circlearrowright) \text{ TC-WORD}$

$\tau ::= i \mid i \rightarrow j \mid i \bowtie j \mid fg_i(\tau) \mid \tau \oplus \tau$
1-STACK $k$-CLOCKS TC-WORDS $\subseteq \text{TW}_{3k+2}$

$G = (V, \rightarrow, \leftarrow) \text{ TC-WORD}$

$\tau ::= i \mid i \rightarrow j \mid i \leftarrow j \mid f_{g_i}(\tau) \mid \tau \oplus \tau$
$\tau ::= i \mid i \rightarrow j \mid i \blacklozenge j \mid f_{g_i}(\tau) \mid \tau \oplus \tau$
1-STACK \( k \)-CLOCKS TC-WORDS \( \subseteq \text{TW}_{3k+2} \)

\[ G = (V, \to, \rho) \] TC-WORD

\[ \tau ::= i \mid i \to j \mid i \curvearrowright j \mid fg_{i}(\tau) \mid \tau \oplus \tau \]

- At most one \textit{hanging} reset node for each clock
- At most one \textit{Last} reset node for each clock
- First and last points
- \( k+1 \) extra \textit{colors} to maintain this invariant
$\mathsf{1\text{-STACK } k\text{-CLOCKS TC-WORDS} \subseteq \mathcal{TW}_{3k+2}}$

$G = (V, \to, \gamma)$ TC-WORD

$\tau ::= i \mid i \to j \mid i \leadsto j \mid \mathsf{fg}_i(\tau) \mid \tau \oplus \tau$
1-STACK $k$-CLOCKS TC-WORDS $\subseteq TW_{3k+2}$

$G = (V, \rightarrow, \rightsquigarrow)$ TC-WORD

$\tau ::= i \mid i \rightarrow j \mid i \rightsquigarrow j \mid fg_i(\tau) \mid \tau \oplus \tau$
1-STACK k-CLOCKS TC-WORDS $\subseteq \text{TW}_{3k+2}$  

$G = (V, \rightarrow, \rightsquigarrow) \text{ TC-WORD}$

$$\tau ::= i \mid i \rightarrow j \mid i \rightsquigarrow j \mid f_{g_i}(\tau) \mid \tau \oplus \tau$$
$\mathcal{T} ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid f_{g_i}(\tau) \mid \tau \oplus \tau$
1-STACK k-CLOCKS TC-WORDS $\subseteq TW_{3k+2}$

$G = (V, \to, \circlearrowright)$ TC-WORD

$\tau ::= i \mid i \to j \mid i \circlearrowright j \mid fg_i(\tau) \mid \tau \oplus \tau$
1-STACK $k$-CLOCKS TC-WORDS $\subseteq TW_{3k+2}$

$G = (V, \to, \leadsto)$ TC-WORD

$\tau ::= i \mid i \to j \mid i \leadsto j \mid fg_i(\tau) \mid \tau \oplus \tau$
ANALYZING TIMED SYSTEMS USING TREE AUTOMATA

1-STACK k-CLOCKS TC-WORDS ⊆ TW_{3k+2} \quad G = ( V, \to, \rightsquigarrow ) TC-WORD

\[ \tau ::= i \mid i \to j \mid i \rightsquigarrow j \mid f_g i(\tau) \mid \tau \circ \tau \]
1-STACK $k$-CLOCKS TC-WORDS $\subseteq TW_{3k+2}$

$G = (V, \rightarrow, \rightsquigarrow) TC\text{-WORD}$

\[
\tau ::= i \mid i \rightarrow j \mid i \rightsquigarrow j \mid fg_i(\tau) \mid \tau \oplus \tau
\]
1-STACK k-CLOCKS TC-WORDS $\subseteq \text{TW}_{3k+2}$

$G = (V, \to, \rightarrow)$ TC-WORD

$$\tau ::= i \mid i \to j \mid i \xrightarrow{\circ} j \mid f_{g_{i}}(\tau) \mid \tau \oplus \tau$$
ANALYZING TIMED SYSTEMS USING TREE AUTOMATA

$1$-STACK $k$-CLOCKS TC-WORDS $\subseteq \text{TW}_{3k+2} \; G = (V, \rightarrow, \rightsquigarrow) \; \text{TC-WORD}$

$\tau ::= i \mid i \rightarrow j \mid i \rightsquigarrow j \mid \text{fg}_i(\tau) \mid \tau \oplus \tau$

- At most one hanging reset node for each clock
- At most one Last reset node for each clock
- First and last points
- $k+1$ extra colors to maintain this invariant
ANALYZING TIMED SYSTEMS USING TREE AUTOMATA

COURCELLE’S THEOREM

- Let $TW_k$ be the set of graphs of tree-width at most $k$
- Let $P$ be a property of graphs
- If $P$ is MSO-definable then $P \cap TW_k \neq \emptyset$ is decidable

WE WANT TO SOLVE $\mathcal{L}_{TCW}(A) \cap \text{Real}_{TCW} \neq \emptyset$

- Show that TC-words have bounded tree-width
- Show that our properties are MSO-definable
- Build directly tree automata for our properties
ANALYZING TIMED SYSTEMS USING TREE AUTOMATA

OUTLINE

▸ BEHAVIOURS AS GRAPHS
▸ DECIDING PROPERTIES OF GRAPHS
▸ DEFINABILITY OF PROPERTIES FOR TIMED SYSTEMS
▸ TREE-WIDTH FOR TIMED SYSTEMS
▸ INTERPRETING GRAPHS IN TREES
▸ CONCLUSION
TREE INTERPRETATION

\[ \tau ::= i \mid i \rightarrow j \mid fg_i(\tau) \mid \tau \oplus \tau \]
TREE INTERPRETATION

\[
\tau ::= i \mid i \xrightarrow{j} j \mid \text{fg}_i(\tau) \mid \tau \oplus \tau
\]

- Edge = leaf
Edge = leaf

Vertex = leaf + color

\[ \tau ::= i \mid i \xrightarrow{j} j \mid \text{fg}_i(\tau) \mid \tau \oplus \tau \]
TREE INTERPRETATION

- Edge = leaf
- Vertex = leaf + color
- One vertex = several leaves

\[ \tau ::= i \mid i \rightarrow j \mid fg_i(\tau) \mid \tau \oplus \tau \]
TREE INTERPRETATION

- Edge = leaf
- Vertex = leaf + color
- One vertex = several leaves

\[
\tau ::= i \mid i \overset{1}{\longrightarrow} j \mid f_{g_i}(\tau) \mid \tau \oplus \tau
\]
**TREE INTERPRETATION**

- Edge = leaf
- Vertex = leaf + color
- One vertex = several leaves
- SameVertex_i(x,y)

\[
\text{SameVertex}_i(x, y) ::= \exists z \left( z < x \land z < y \land \forall z' \left( (z < z' < x \lor z < z' < y) \implies \neg f_{g_i}(z') \right) \right)
\]
TREE INTERPRETATION

- Edge = leaf
- Vertex = leaf + color
- One vertex = several leaves
- $\text{SameVertex}_i(x,y)$

$$\tau ::= i \mid i \rightarrow j \mid f_{g_i}(\tau) \mid \tau \oplus \tau$$

$$\text{SameVertex}_i(x,y) ::= \exists z \left( z < x \land z < y \land \forall z' \left( (z < z' < x \lor z < z' < y) \implies \neg f_{g_i}(z') \right) \right)$$
ANALYZING TIMED SYSTEMS USING TREE AUTOMATA

**TREE INTERPRETATION**

\[ \tau ::= i \mid i \xrightarrow{j} j \mid fg_i(\tau) \mid \tau \oplus \tau \]

- Edge = leaf
- Vertex = leaf + color
- One vertex = several leaves
- \( \text{SameVertex}_i(x, y) \)

\[ \text{SameVertex}_i(x, y) ::= \exists z \left( z < x \land z < y \land \forall z' \left( (z < z' < x \lor z < z' < y) \implies \neg fg_i(z') \right) \right) \]
TREE INTERPRETATION

\[ \tau ::= i \mid i \rightarrow j \mid f_{g_{i}}(\tau) \mid \tau \oplus \tau \]

- Edge = leaf
- Vertex = leaf + color
- One vertex = several leaves
- \text{SameVertex}_{i}(x, y)

\[ \text{SameVertex}_{i}(x, y) ::= \exists z \left( z < x \land z < y \land \forall z' \left( (z < z' < x \lor z < z' < y) \implies \neg f_{g_{i}}(z') \right) \right) \]
DIRECTLY BUILDING TREE AUTOMATA

We can build a tree automaton $A^k_{\text{valid}}$ of size $2^{O(k^2)}$ which accepts all $k$-terms denoting valid TCWs.

We can build a tree automaton $A^k_{\text{real}}$ of size $M^{\text{poly}(k)}$ which accepts all $k$-terms denoting realizable TCWs using constants at most $M$.

Let $S$ be a pushdown timed automaton with set of clocks $X$. Let $|S|$ be its size (constants encoded in unary) and $k = 3|X| + 2$.

We can build a tree automaton $A^k_S$ of size $|S|^{\text{poly}(k)}$ which accepts all $k$-terms denoting TCWs in $L_{\text{TCW}}(S)$.

\[ L(S) \neq \emptyset \iff L(A^k_{\text{valid}} \cap A^k_{\text{real}} \cap A^k_S) \neq \emptyset \]
ANALYZING TIMED SYSTEMS USING TREE AUTOMATA

COURCELLE’S THEOREM

- Let $TW_k$ be the set of graphs of tree-width at most $k$
- Let $P$ be a property of graphs
- If $P$ is MSO-definable then $P \cap TW_k \neq \emptyset$ is decidable

WE WANT TO SOLVE $L_{TCW}(A) \cap \text{Real}_{TCW} \neq \emptyset$

- Show that TC-words have bounded tree-width ✓
- Show that our properties are MSO-definable ✓
- Build directly tree automata for our properties ✓
OUTLINE

▸ BEHAVIOURS AS GRAPHS
▸ DECIDING PROPERTIES OF GRAPHS
▸ DEFINABILITY OF PROPERTIES FOR TIMED SYSTEMS
▸ TREE-WIDTH FOR TIMED SYSTEMS
▸ INTERPRETING GRAPHS IN TREES
▸ CONCLUSION
CONCLUSION

1. Write behaviors as graphs with timing constraints
2. Show a bound on tree-width for these graphs
3. Show MSO-definability of the relevant properties, or
4. Build Tree automata directly

RESULTS

- PSPACE decision procedure for timed automata
- EXPTIME decision procedure for pushdown timed automata
- EXPTIME decision procedure for multi-pushdown timed automata with bounded rounds
CONCLUSION

1. Write behaviors as graphs with timing constraints
2. Show a bound on tree-width for these graphs
3. Show MSO-definability of the relevant properties, or
4. Build Tree automata directly

FUTURE WORK

- Efficient implementation
- Concurrent recursive timed programs
- MSO/LCPDL-definability of realizability and non-realizability
- Model-Checking wrt. timed specifications
THANK YOU