A Logical Characterization of Timed Pushdown Languages

Manfred Droste and Vitaly Perevoshchikov[1]

Leipzig University

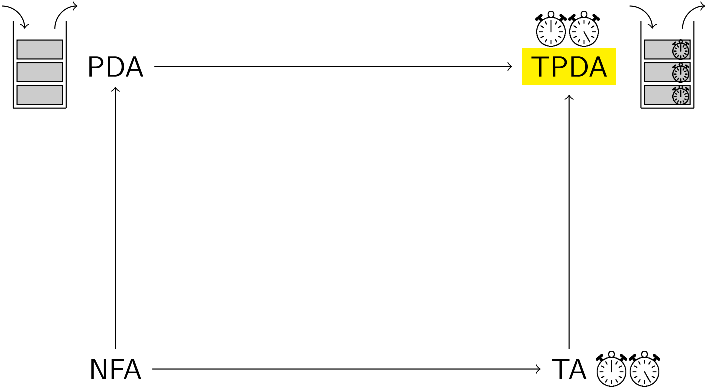
CSR 2015, Listvyanka

[1]Supported by the DFG Research Training Group "QuantLA"

(Dense-)Timed Pushdown Automata[1] (TPDA)

TPDA are nondeterministic finite automata (NFA) equipped with:

- real-valued clocks
- timed stack



[1] Abdulla, Atig, Stenman '12

Timed Pushdown Automata[1] (TPDA)

Definition

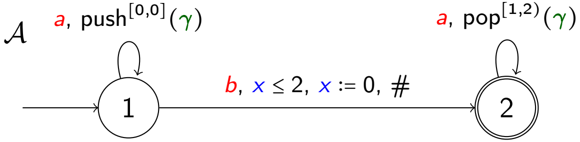
A TPDA over an alphabet Σ: $\mathcal{A} = (Q, C, \Gamma, I, T, F)$ where

- Q is a finite set of states
- C is a finite set of clocks
- Γ is a stack alphabet
- $I, F \subseteq Q$ are initial and final state
- T is a finite set of edges of the form $q \xrightarrow[s]{a,\,\phi,\,\Lambda} q'$ where:
  - $q, q' \in Q$, $a \in \Sigma$
  - ϕ is a clock constraint over C, $\Lambda \subseteq C$ is a set of clocks to be reset
  - s is: $\text{push}^{\mathcal{I}}(\gamma)$, $\#$ or $\text{pop}^{\mathcal{I}}(\gamma)$ where $\gamma \in \Gamma$ and $\mathcal{I}$ is an interval

[1] Abdulla, Atig, Stenman '12

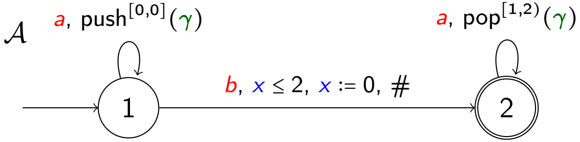
TPDA: Example

$\Sigma = \{a, b\}$
$C = \{x\}$
$\Gamma = \{\gamma\}$.



TPDA: Example

$\Sigma = \{a, b\}$
$C = \{x\}$
$\Gamma = \{\gamma\}$.

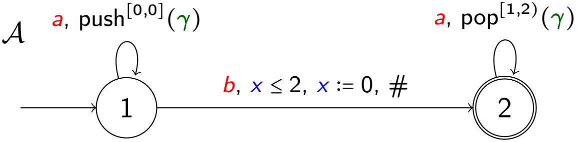


A run of $\mathcal{A}$:

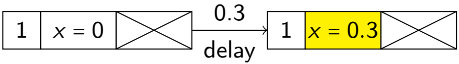
1	$x = 0$	

TPDA: Example

$\Sigma = \{a, b\}$
$C = \{x\}$
$\Gamma = \{\gamma\}$.

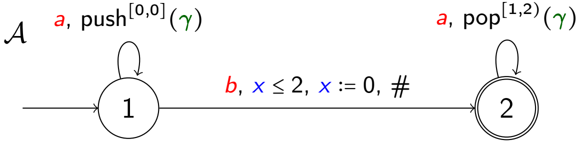


A run of $\mathcal{A}$:

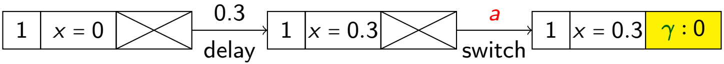


TPDA: Example

$\Sigma = \{a, b\}$
$C = \{x\}$
$\Gamma = \{\gamma\}$.

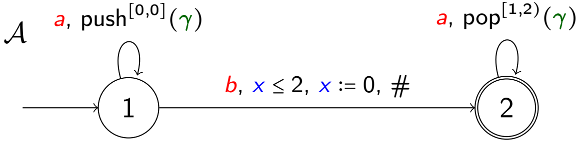


A run of $\mathcal{A}$:

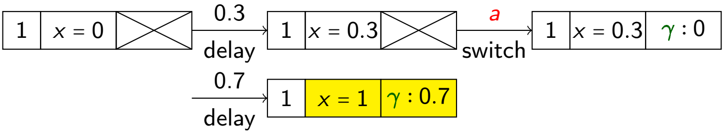


TPDA: Example

$\Sigma = \{a, b\}$
$C = \{x\}$
$\Gamma = \{\gamma\}$.

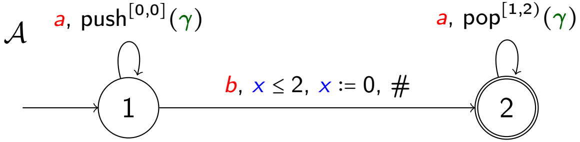


A run of $\mathcal{A}$:

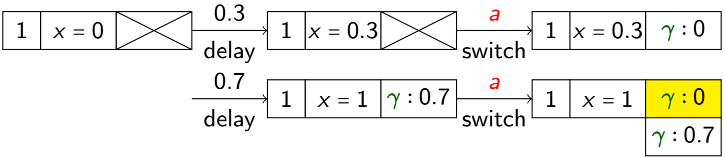


TPDA: Example

$\Sigma = \{a, b\}$
$C = \{x\}$
$\Gamma = \{\gamma\}$.

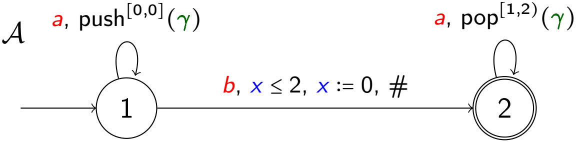


A run of $\mathcal{A}$:

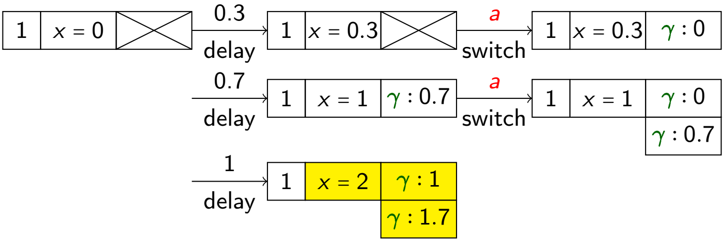


TPDA: Example

$\Sigma = \{a, b\}$
$C = \{x\}$
$\Gamma = \{\gamma\}$.

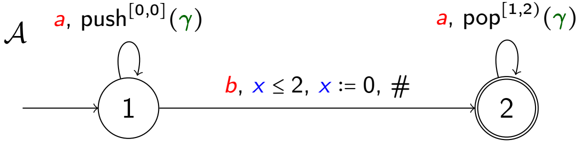


A run of $\mathcal{A}$:

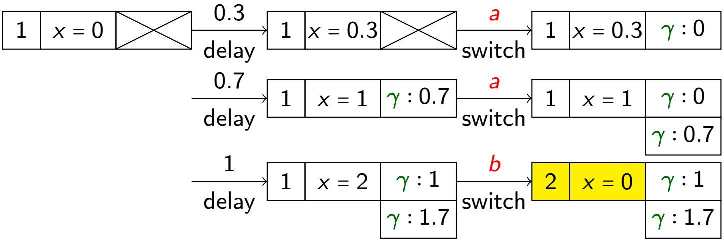


TPDA: Example

$\Sigma = \{a, b\}$
$C = \{x\}$
$\Gamma = \{\gamma\}$.

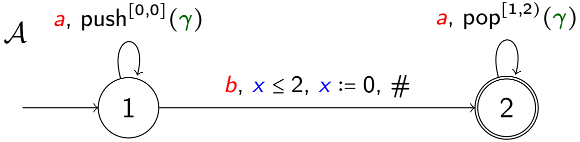


A run of $\mathcal{A}$:

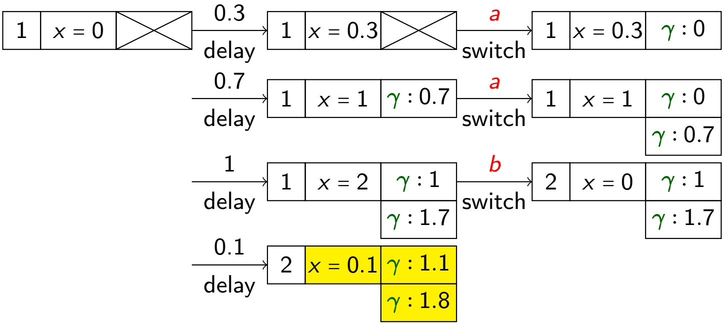


TPDA: Example

$\Sigma = \{a, b\}$
$C = \{x\}$
$\Gamma = \{\gamma\}$.

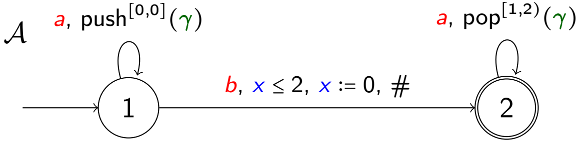


A run of $\mathcal{A}$:

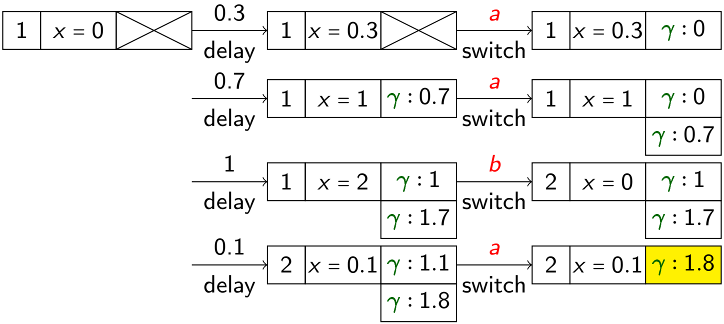


TPDA: Example

$\Sigma = \{a, b\}$
$C = \{x\}$
$\Gamma = \{\gamma\}$.

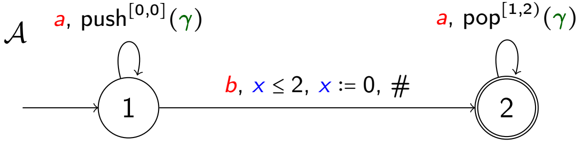


A run of $\mathcal{A}$:

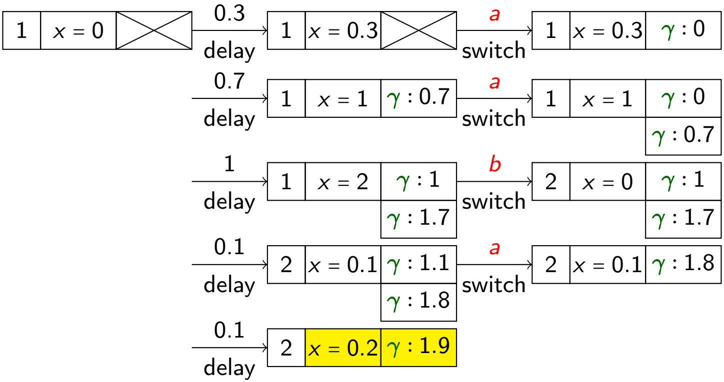


TPDA: Example

$\Sigma = \{a, b\}$
$C = \{x\}$
$\Gamma = \{\gamma\}$.

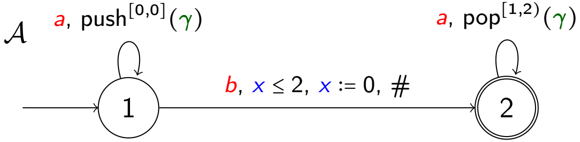


A run of $\mathcal{A}$:

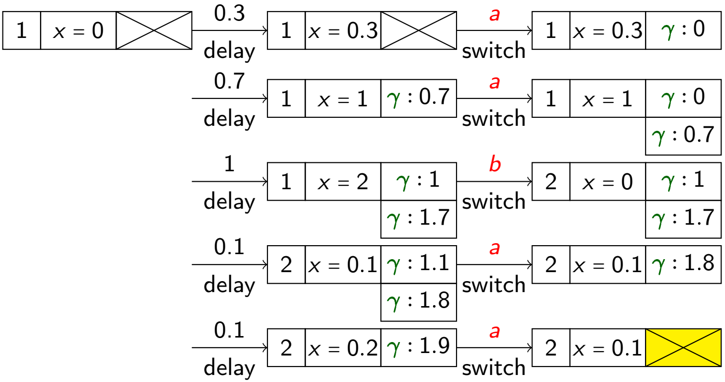


TPDA: Example

$\Sigma = \{a, b\}$
$C = \{x\}$
$\Gamma = \{\gamma\}$.

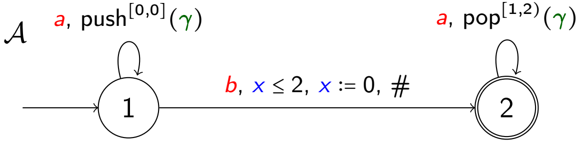


A run of $\mathcal{A}$:

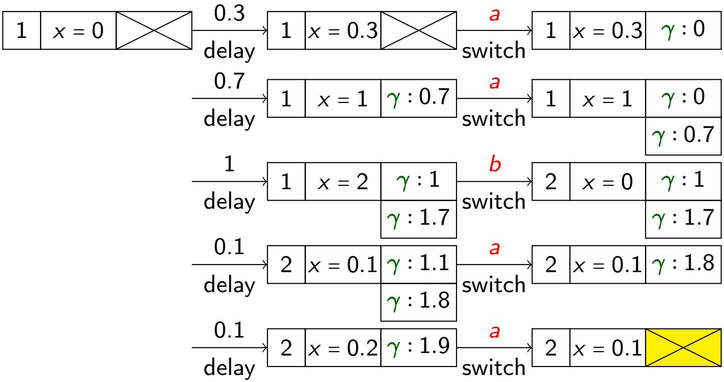


TPDA: Example

$\Sigma = \{a, b\}$
$C = \{x\}$
$\Gamma = \{\gamma\}$.



A run of $\mathcal{A}$:



Accepted timed word: $(a, 0.3)(a, 1)(b, 2)(a, 2.1)(a, 2.2)$

Relative Distance Logic (RDL)[1]

Let Σ be an alphabet.

Definition

Relative distance logic $\mathrm{RDL}(\Sigma)$: consists of formulas of the form $\exists X_1 \ldots \exists X_n . \varphi$ where

$$\varphi ::= P_a(x) \mid x \leq y \mid x \in X \mid \boxed{d(X, x) \sim c} \mid \varphi \vee \varphi \mid \neg\varphi \mid \exists x. \varphi$$

with $a \in \Sigma$, $\sim \in \{<, =, >\}$, $c \in \mathbb{N}$.

[1] Wilke '94

Relative Distance Logic (RDL)[1]

Let Σ be an alphabet.

Definition

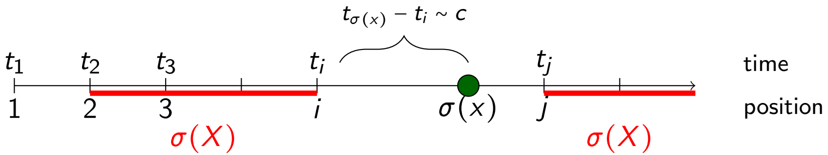
Relative distance logic $\mathrm{RDL}(\Sigma)$: consists of formulas of the form $\exists X_1 \ldots \exists X_n . \varphi$ where

$$\varphi ::= P_a(x) \mid x \le y \mid x \in X \mid \boxed{d(X,x) \sim c} \mid \varphi \vee \varphi \mid \neg \varphi \mid \exists x . \varphi$$

with $a \in \Sigma$, $\sim \in \{<, =, >\}$, $c \in \mathbb{N}$.

Model: a timed word $w = (a_1, t_1) \ldots (a_n, t_n) \in \mathbb{T}\Sigma^+$.

$$(w, \sigma) \models d(X,x) \sim c$$



[1]Wilke '94

Relative Distance Logic (RDL)[1]

Let Σ be an alphabet.

Definition

Relative distance logic $\mathrm{RDL}(\Sigma)$: consists of formulas of the form $\exists X_1 \ldots \exists X_n . \varphi$ where

$$\varphi ::= P_a(x) \mid x \leq y \mid x \in X \mid \boxed{d(X, x) \sim c} \mid \varphi \vee \varphi \mid \neg\varphi \mid \exists x . \varphi$$

with $a \in \Sigma$, $\sim \in \{<, =, >\}$, $c \in \mathbb{N}$.

Theorem (Wilke '94)

Let $L \subseteq \mathbb{T}\Sigma^+$ be a timed language. TFAE:

1. L is recognizable by a timed automaton
2. L is definable by a $\mathrm{RDL}(\Sigma)$-sentence.

[1] Wilke '94

Logic for Pushdown Automata[1]

Matching logic ML(Σ): $\exists^{\text{match}}\mu.\text{FO}(\Sigma, <, \mu)$

Definition (Matching).

A relation $M \subseteq \{1, \ldots, n\}^2$ is a matching if:

- $(x, y) \in M \Rightarrow x < y$;
- every $x \in \{1, \ldots, n\}$ belongs to at most one pair in M;

[1] Lautemann, Schwentick, Thérien '94

Logic for Pushdown Automata[1]

Matching logic ML(Σ): $\exists^{\text{match}}\mu.\text{FO}(\Sigma, <, \mu)$

Definition (Matching).

A relation $M \subseteq \{1, ..., n\}^2$ is a matching if:

- $(x, y) \in M \Rightarrow x < y$;
- every $x \in \{1, ..., n\}$ belongs to at most one pair in M;
- M is non-crossing:

[1] Lautemann, Schwentick, Thérien '94

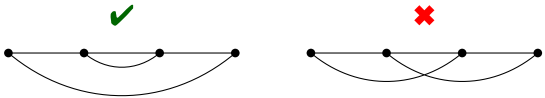
Logic for Pushdown Automata[1]

Matching logic ML(Σ): $\exists^{\text{match}}\mu.\mathrm{FO}(\Sigma, <, \mu)$

Definition (Matching).

A relation $M \subseteq \{1, \ldots, n\}^2$ is a matching if:

- $(x, y) \in M \Rightarrow x < y$;
- every $x \in \{1, \ldots, n\}$ belongs to at most one pair in M;
- M is non-crossing:



[1] Lautemann, Schwentick, Thérien '94

Timed Matching Logic (TML)

Definition

TML(Σ) is the set of formulas of the form $\exists^{\text{match}}\mu.\exists X_1.\ \dots\ \exists X_n.\varphi$ where φ is defined by the grammar:

$$\varphi ::= P_a(x) \mid x \le y \mid x \in X \mid \boxed{\mu(x,y) \sim c} \mid d(X,x) \sim c \mid$$
$$\varphi \vee \varphi \mid \neg\varphi \mid \exists x.\varphi$$

where $a \in \Sigma$, $\sim \in \{<,=,>\}$ and $c \in \mathbb{N}$.

Timed Matching Logic (TML)

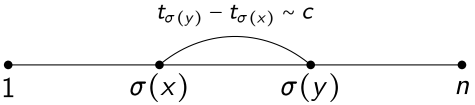
Definition

TML(Σ) is the set of formulas of the form $\exists^{\text{match}}\mu.\exists X_1.\ \ldots\ \exists X_n.\varphi$ where φ is defined by the grammar:

$$\varphi \ ::= P_a(x) \mid x \le y \mid x \in X \mid \boxed{\mu(x,y) \sim c} \mid d(X,x) \sim c \mid$$
$$\varphi \vee \varphi \mid \neg\varphi \mid \exists x.\varphi$$

where $a \in \Sigma$, $\sim \in \{<, =, >\}$ and $c \in \mathbb{N}$.

Let $w = (a_1, t_1)\ldots(a_n, t_n) \in \mathbb{T}\Sigma^+$. Then, $(w, \sigma) \models \mu(x,y) \sim c$ iff:

- $(\sigma(x), \sigma(y)) \in \sigma(\mu)$
- 

Example: Timed Dyck Languages

Definition

- Let $\Sigma = \{a_1, \ldots, a_m\}$ be a set of opening brackets
- Let $\overline{\Sigma} = \{\overline{a}_1, \ldots, \overline{a}_m\}$ be a set of corresponding closing brackets

Example: Timed Dyck Languages

Definition

- Let $\Sigma = \{a_1, \ldots, a_m\}$ be a set of opening brackets
- Let $\overline{\Sigma} = \{\overline{a}_1, \ldots, \overline{a}_m\}$ be a set of corresponding closing brackets
- Let $I_1, \ldots, I_m$ be intervals (e.g., $(0, 3], [2, \infty)$, etc.)

Example: Timed Dyck Languages

Definition

- Let $\Sigma = \{a_1, \ldots, a_m\}$ be a set of opening brackets
- Let $\overline{\Sigma} = \{\bar{a}_1, \ldots, \bar{a}_m\}$ be a set of corresponding closing brackets
- Let $I_1, \ldots, I_m$ be intervals (e.g., $(0,3], [2,\infty)$, etc.)
- A timed Dyck language $\mathcal{D}_\Sigma(I_1, \ldots, I_m)$ consists of all timed words $(b_1, t_1)\ldots(b_n, t_n) \in \mathbb{T}(\Sigma \cup \overline{\Sigma})^+$ such that:

Example: Timed Dyck Languages

Definition

- Let $\Sigma = \{a_1, \dots, a_m\}$ be a set of opening brackets
- Let $\overline{\Sigma} = \{\bar{a}_1, \dots, \bar{a}_m\}$ be a set of corresponding closing brackets
- Let $I_1, \dots, I_m$ be intervals (e.g., $(0, 3]$, $[2, \infty)$, etc.)
- A timed Dyck language $\mathcal{D}_\Sigma(I_1, \dots, I_m)$ consists of all timed words $(b_1, t_1) \dots (b_n, t_n) \in \mathbb{T}(\Sigma \cup \overline{\Sigma})^+$ such that:
  - $b_1 \dots b_n \in (\Sigma \cup \overline{\Sigma})^+$ is a correctly nested sequence of brackets

Example: Timed Dyck Languages

Definition

- Let $\Sigma = \{a_1, \ldots, a_m\}$ be a set of opening brackets
- Let $\overline{\Sigma} = \{\overline{a}_1, \ldots, \overline{a}_m\}$ be a set of corresponding closing brackets
- Let $I_1, \ldots, I_m$ be intervals (e.g., $(0,3], [2,\infty)$, etc.)
- A timed Dyck language $\mathcal{D}_\Sigma(I_1, \ldots, I_m)$ consists of all timed words $(b_1, t_1) \ldots (b_n, t_n) \in \mathbb{T}(\Sigma \cup \overline{\Sigma})^+$ such that:
  - $b_1 \ldots b_n \in (\Sigma \cup \overline{\Sigma})^+$ is a correctly nested sequence of brackets
  - the time distance between any two matching brackets a_j and $\overline{a}_j$ is in I_j.

Example: Timed Dyck Languages

Definition

- Let $\Sigma = \{a_1, \dots, a_m\}$ be a set of opening brackets
- Let $\overline{\Sigma} = \{\overline{a}_1, \dots, \overline{a}_m\}$ be a set of corresponding closing brackets
- Let $I_1, \dots, I_m$ be intervals (e.g., $(0, 3], [2, \infty)$, etc.)
- A timed Dyck language $\mathcal{D}_\Sigma(I_1, \dots, I_m)$ consists of all timed words $(b_1, t_1) \dots (b_n, t_n) \in \mathbb{T}(\Sigma \cup \overline{\Sigma})^+$ such that:
  - $b_1 \dots b_n \in (\Sigma \cup \overline{\Sigma})^+$ is a correctly nested sequence of brackets
  - the time distance between any two matching brackets a_j and $\overline{a}_j$ is in I_j.

Example: Timed Dyck Languages

Definition

- Let $\Sigma = \{a_1, \ldots, a_m\}$ be a set of opening brackets
- Let $\overline{\Sigma} = \{\overline{a}_1, \ldots, \overline{a}_m\}$ be a set of corresponding closing brackets
- Let $I_1, \ldots, I_m$ be intervals (e.g., $(0,3], [2,\infty)$, etc.)
- A timed Dyck language $\mathcal{D}_\Sigma(I_1, \ldots, I_m)$ consists of all timed words $(b_1, t_1)\ldots(b_n, t_n) \in \mathbb{T}(\Sigma \cup \overline{\Sigma})^+$ such that:
  - $b_1 \ldots b_n \in (\Sigma \cup \overline{\Sigma})^+$ is a correctly nested sequence of brackets
  - the time distance between any two matching brackets a_j and $\overline{a}_j$ is in I_j.

$\mathcal{D}_\Sigma(I_1, \ldots, I_m)$ is defined by the sentence:

$$\varphi = \exists^{\text{match}} \mu . \Bigg(\forall x . \exists y . (\mu(x,y) \vee \mu(y,x)) \wedge$$
$$\forall x . \forall y . \Bigg(\mu(x,y) \rightarrow \bigvee_{j=1}^{m} (P_{a_j}(x) \wedge P_{\overline{a}_j}(y) \wedge \mu^{I_j}(x,y)) \Bigg) \Bigg)$$

Main Result

Theorem

Let Σ be an alphabet and $\mathcal{L} \subseteq \mathbb{T}\Sigma^+$ a timed language. TFAE:

1. *$\mathcal{L}$ is recognizable by a TPDA.*
2. *$\mathcal{L}$ is definable by a $\mathsf{TML}(\Sigma)$-sentence.*

Main Result

Theorem

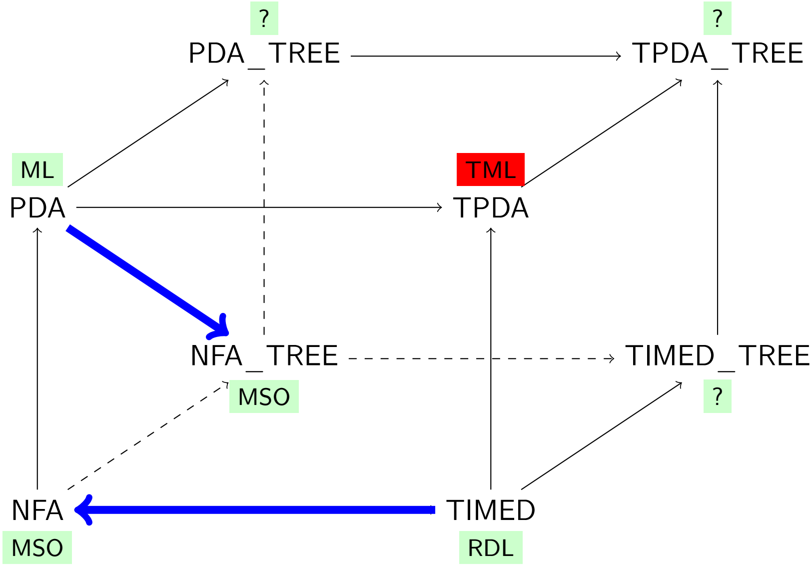
Let Σ be an alphabet and $\mathcal{L} \subseteq \mathbb{T}\Sigma^+$ a timed language. TFAE:

1. *$\mathcal{L}$ is recognizable by a TPDA.*
2. *$\mathcal{L}$ is definable by a $\mathsf{TML}(\Sigma)$-sentence.*

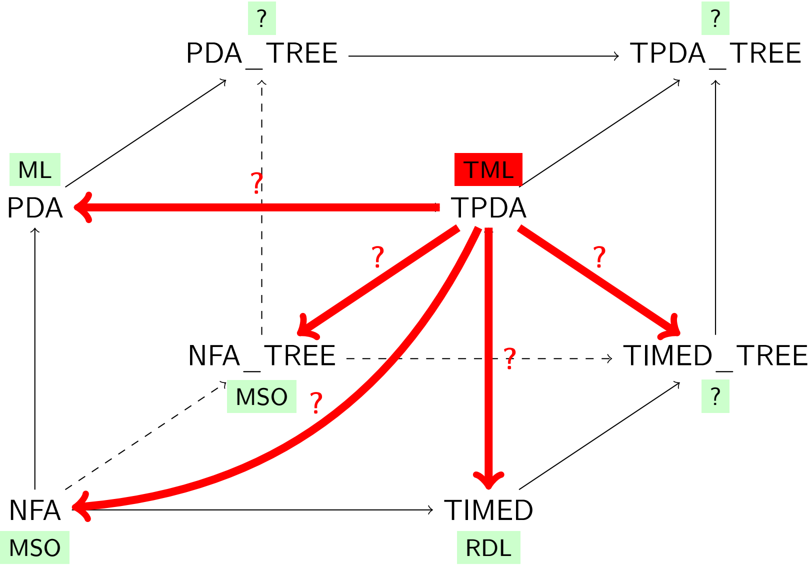
Corollary

It is decidable, given an alphabet Σ and a sentence $\psi \in \mathsf{TML}(\Sigma)$, whether there exists a timed word $w \in \mathbb{T}\Sigma^+$ with $w \models \psi$.

Proof

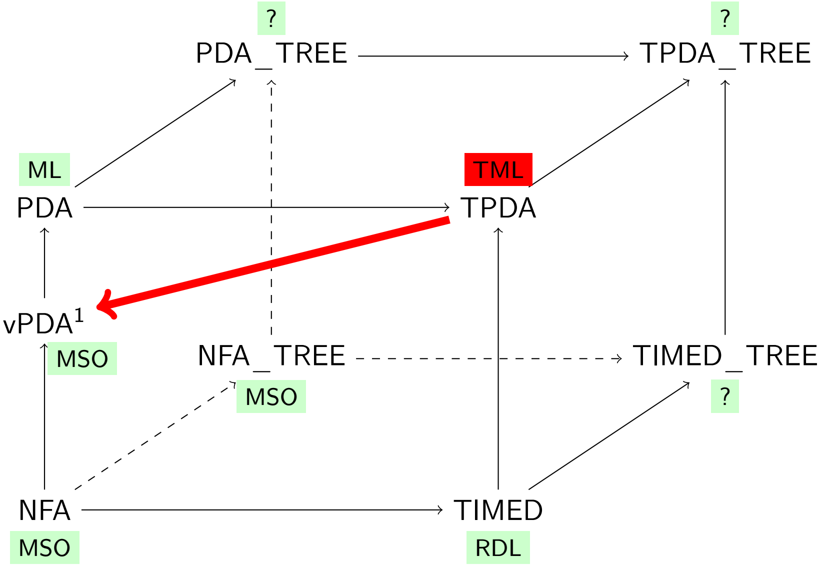

?
PDA_TREE
?
TPDA_TREE
ML
PDA
TML
TPDA
NFA_TREE
MSO
TIMED_TREE
?
NFA
MSO
TIMED
RDL

Proof



?
?
PDA_TREE
TPDA_TREE
ML
TML
PDA
?
TPDA
?
?
NFA_TREE
?
TIMED_TREE
MSO
?
?
NFA
TIMED
MSO
RDL

Proof



[1] Alur, Madhusudan '04

Visibly Pushdown Automata[1] (vPDA)

- Let Σ^{push}, $\Sigma^{\#}$ and Σ^{pop} be pairwise disjoint alphabets
- Let $\Sigma = \Sigma^{\text{push}} \cup \Sigma^{\#} \cup \Sigma^{\text{pop}}$ and $\tilde{\Sigma} = \langle \Sigma^{\text{push}}, \Sigma^{\#}, \Sigma^{\text{pop}} \rangle$

Definition

A vPDA over $\tilde{\Sigma}$ is a tuple $\mathcal{A} = (Q, \Gamma, I, T, F)$ where:

- Q is a finite set of states, Γ is a stack alphabet
- $I, F \subseteq Q$ are sets of initial resp. final states
- $T = T^{\text{push}} \uplus T^{\#} \uplus T^{\text{pop}}$ where:
  - $T^{\text{push}} \subseteq Q \times \Sigma^{\text{push}} \times \Gamma \times Q$
  - $T^{\#} \subseteq Q \times \Sigma^{\#} \times Q$
  - $T^{\text{pop}} \subseteq Q \times \Sigma^{\text{pop}} \times (\Gamma \uplus \{\bot\}) \times Q$

Accepted language: $\mathcal{L}(\mathcal{A}) \subseteq \Sigma^{+}$.

[1] Alur, Madhusudan '04

Logic for Visibly Pushdown Languages[1]

- Let Σ^{push}, $\Sigma^{\#}$ and Σ^{pop} be pairwise disjoint alphabets
- Let $\Sigma = \Sigma^{\text{push}} \cup \Sigma^{\#} \cup \Sigma^{\text{pop}}$ and $\tilde{\Sigma} = \langle \Sigma^{\text{push}}, \Sigma^{\#}, \Sigma^{\text{pop}} \rangle$

Definition

Logic $\mathrm{MSO}(\tilde{\Sigma})$ is defined as:

$$\varphi \ ::= \ P_a(x) \mid x \leq y \mid x \in X \mid \mathrm{match}(x,y) \mid \varphi \vee \varphi \mid \neg\varphi \mid \exists x.\varphi \mid \exists X.\varphi$$

where $a \in \Sigma$.

Defined language of a sentence $\varphi \in \mathrm{MSO}(\tilde{\Sigma})$: $\mathcal{L}(\varphi) \subseteq \Sigma^+$.

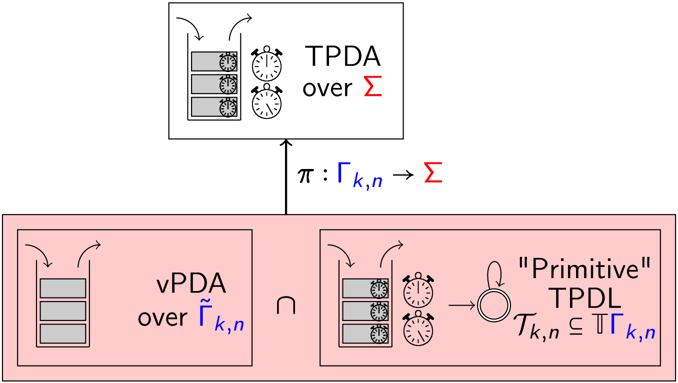
Theorem[1]

Let $\mathcal{L} \subseteq \Sigma^+$ be a language. TFAE:

1. $\mathcal{L}$ is recognizable by a vPDA over $\tilde{\Sigma}$.
2. $\mathcal{L}$ is $\mathrm{MSO}(\tilde{\Sigma})$-definable.

[1] Alur, Madhusudan '04

Decomposition of TPDA

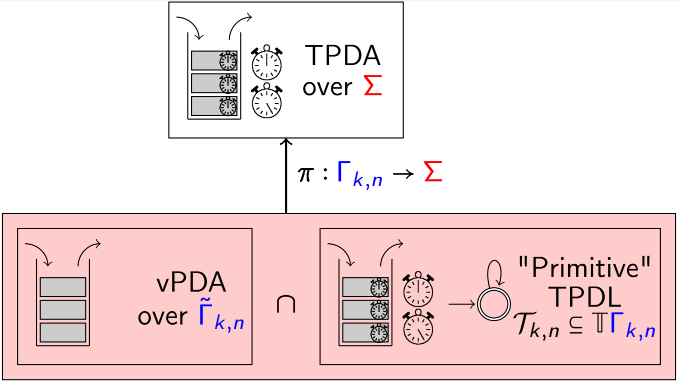


Extended alphabet

$$\Gamma_{k,n} = \Sigma \times \underbrace{(\mathbb{P}(k))^n}_{\text{clock constraints}} \times \underbrace{\{0,1\}^n}_{\text{clock resets}} \times \underbrace{\mathbb{P}(k)}_{\text{stack constraints}} \times \underbrace{\{\text{push}, \#, \text{pop}\}}_{\text{stack commands}}$$

- $n :=$ number of global clocks
- $k :=$ maximal number appearing in constraints
- $\mathbb{P}(k) := \{[0,0], (0,1), [1,1], \ldots, (k-1,k), [k,k], (k,\infty)\}$

Decomposition of TPDA



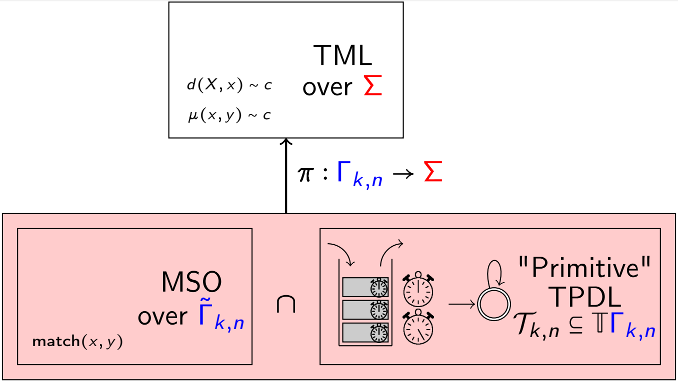
Theorem

Let $\mathcal{L} \subseteq \mathbb{T}\Sigma^+$. TFAE:

1. *$\mathcal{L}$ is a timed pushdown language.*
2. *There exist $k, n \in \mathbb{N}$ and a vPDL $\mathcal{L}' \subseteq {\Gamma_{k,n}}^+$ with*

$$\mathcal{L} = \pi(\mathcal{L}' \cap \mathcal{T}_{k,n})$$

Decomposition of TML



Theorem

Let $\mathcal{L} \subseteq \mathbb{T}\Sigma^+$. *TFAE:*

1. $\mathcal{L}$ *is* TML*-definable.*
2. *There exist* $k, n \in \mathbb{N}$ *and a vPDL* $\mathcal{L}' \subseteq (\Gamma_{k,n})^+$ *with*

$$\mathcal{L} = \pi(\mathcal{L}' \cap \mathcal{T}_{k,n})$$

Future work

1. Weighted TPDA
2. TPDA with ε-transitions

Future work

1. Weighted TPDA
2. TPDA with ε-transitions
3. Are TDPA without global clocks expressively equivalent to $\exists^{\text{match}}\mu.\text{FO}(<, \mu_{\sim c})$?

Future work

1. Weighted TPDA
2. TPDA with ε-transitions
3. Are TDPA without global clocks expressively equivalent to $\exists^{\mathsf{match}}\mu.\mathsf{FO}(<, \mu_{\sim c})$?
4. Connection to timed tree automata

Future work

1. Weighted TPDA
2. TPDA with ε-transitions
3. Are TDPA without global clocks expressively equivalent to $\exists^{\text{match}}\mu.\text{FO}(<, \mu_{\sim c})$?
4. Connection to timed tree automata
5. A Chomsky-Schützenberger characterization

Future work

1. Weighted TPDA
2. TPDA with ε-transitions
3. Are TDPA without global clocks expressively equivalent to $\exists^{\mathsf{match}}\mu.\mathrm{FO}(<,\mu_{\sim c})$?
4. Connection to timed tree automata
5. A Chomsky-Schützenberger characterization

THANK YOU!