On the Complexity of k-Piecewise Testability and the Depth of Automata

Tomáš Masopust and Michaël Thomazo

TU Dresden, Germany

DLT 2015

Problems

Problems

Problem (k-PiecewiseTestability)

Input: *an automaton (min. DFA or NFA)* $\mathscr{A}$

Output: YES *if and only if* $\mathscr{L}(\mathscr{A})$ *is k-PT*

Quest.: *complexity*

Problem (Bounds on automata of k-PT languages)

Input: $\Sigma = \{a_1, a_2, \ldots, a_n\}$, $n \geq 1$, *and* $k \geq 1$

Quest.: *length of a longest word,* w*, such that*

1. $\mathsf{sub}_k(w) := \{u \in \Sigma^* \mid u \preccurlyeq v, |u| \leq k\}$[1] $= \Sigma^{\leq k}$,
2. *prefixes* $w_1 \neq w_2$ *of* w, $\mathsf{sub}_k(w_1) \neq \mathsf{sub}_k(w_2)$

[1] $abc \preccurlyeq bacabbabca$

Piecewise testable languages (PT)

Definition

A regular language is piecewise testable if it is a finite boolean combination of languages of the form

$$\Sigma^* a_1 \Sigma^* a_2 \Sigma^* \cdots \Sigma^* a_n \Sigma^*$$

where $n \geq 0$ and $a_i \in \Sigma$.

It is k-piecewise testable (k-PT) if $n \leq k$.

Piecewise testable languages (PT)

Definition

A regular language is piecewise testable if it is a finite boolean combination of languages of the form

$$\Sigma^* a_1 \Sigma^* a_2 \Sigma^* \cdots \Sigma^* a_n \Sigma^*$$

where $n \geq 0$ and $a_i \in \Sigma$.

It is k-piecewise testable (k-PT) if $n \leq k$.

Example (PT language)

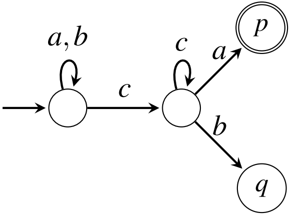
$$\bigcup_{a_1 a_2 \cdots a_n \in L} \Sigma^* a_1 \Sigma^* a_2 \Sigma^* \cdots \Sigma^* a_n \Sigma^*$$

PT recognition

$$\mathfrak{Bool}\left(\Sigma^* a_1 \Sigma^* a_2 \Sigma^* \cdots \Sigma^* a_n \Sigma^*\right)$$

Theorem (min. DFA characterization[2])

1. *Partially ordered* – *acyclic, but with self-loops*
2. *Confluent* – $\forall q \in Q, \forall a,b \in \Sigma, \exists w \in \{a,b\}^*$ *s.t.* $(qa)w = (qb)w$



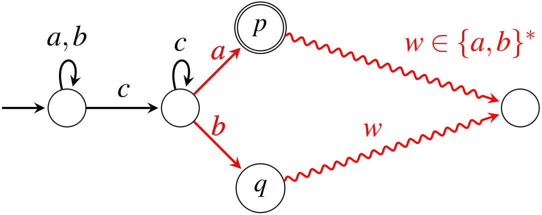
[2] Version by Klíma + Polák 2013

PT recognition

$$\mathfrak{Bool}\left(\Sigma^* a_1 \Sigma^* a_2 \Sigma^* \cdots \Sigma^* a_n \Sigma^*\right)$$

Theorem (min. DFA characterization[2])

1. *Partially ordered – acyclic, but with self-loops*
2. *Confluent* – $\forall q \in Q, \forall a,b \in \Sigma$, $\exists w \in \{a,b\}^*$ **s.t.** $(qa)w = (qb)w$



[2] Version by Klíma + Polák 2013

What do we know about PT languages?

What do we know about PT languages?

- Simon 1975
  piecewise testable = $\mathscr{J}$-trivial

What do we know about PT languages?

- Simon 1975
  piecewise testable = $\mathscr{J}$-trivial
- Stern 1985
  piecewise testability for DFAs is in PTIME, $O(n^5)$

What do we know about PT languages?

- Simon 1975
  piecewise testable = $\mathscr{J}$-trivial
- Stern 1985
  piecewise testability for DFAs is in PTIME, $O(n^5)$
- Trahtman 2001
  piecewise testability for DFAs in quadratic time

What do we know about PT languages?

- Simon 1975
  piecewise testable = $\mathscr{J}$-trivial
- Stern 1985
  piecewise testability for DFAs is in PTIME, $O(n^5)$
- Trahtman 2001
  piecewise testability for DFAs in quadratic time
- Klíma, Polák 2013
  another quadratic-time algorithm for DFAs

What do we know about PT languages?

- Simon 1975
  piecewise testable = $\mathscr{J}$-trivial
- Stern 1985
  piecewise testability for DFAs is in PTIME, $O(n^5)$
- Trahtman 2001
  piecewise testability for DFAs in quadratic time
- Klíma, Polák 2013
  another quadratic-time algorithm for DFAs
- Cho and Huynh 1991
  piecewise testability for DFAs is NL-complete

What do we know about PT languages?

- Simon 1975
  piecewise testable = $\mathscr{J}$-trivial
- Stern 1985
  piecewise testability for DFAs is in PTIME, $O(n^5)$
- Trahtman 2001
  piecewise testability for DFAs in quadratic time
- Klíma, Polák 2013
  another quadratic-time algorithm for DFAs
- Cho and Huynh 1991
  piecewise testability for DFAs is NL-complete
- Holub, M., Thomazo 2014$^+$
  piecewise testability for NFAs is PSPACE-complete

What do we know about PT languages?

- Simon 1975
  piecewise testable = $\mathscr{J}$-trivial
- Stern 1985
  piecewise testability for DFAs is in PTIME, $O(n^5)$
- Trahtman 2001
  piecewise testability for DFAs in quadratic time
- Klíma, Polák 2013
  another quadratic-time algorithm for DFAs
- Cho and Huynh 1991
  piecewise testability for DFAs is NL-complete
- Holub, M., Thomazo 2014$^+$
  piecewise testability for NFAs is PSPACE-complete
- Bojańczyk, Segoufin, Straubing 2012
  PT tree languages

Problem 1

k-PiecewiseTestability

$$\mathfrak{Bool}\big(\Sigma^* a_1 \Sigma^* a_2 \Sigma^* \cdots \Sigma^* a_n \Sigma^*\big) \quad \text{with } n \leq k$$

Problem (k-PiecewiseTestability)

Input: *An automaton (min. DFA or NFA)* $\mathscr{A}$

Output: YES *if and only if* $\mathscr{L}(\mathscr{A})$ *is k-PT*

Trivially decidable – finite number of k-PTL over $\Sigma_{\mathscr{A}}$

DFAs

Complexity of k-Piecewise Testability for DFAs

Theorem

The following problem

NAME: K-PIECEWISETESTABILITY
INPUT: *a minimal DFA $\mathscr{A}$*
OUTPUT: YES *if and only if $\mathscr{L}(\mathscr{A})$ is k-PT*

*belongs to co-*NP*.*

0-Piecewise Testability DFAs

$$L(\mathscr{A}) \text{ is } \textcolor{red}{\text{0-PT}} \text{ iff } L(\mathscr{A}) = \begin{cases} \Sigma^* \\ \emptyset \end{cases}$$

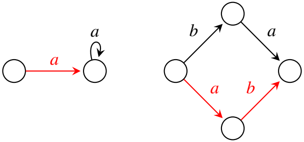
Complexity $O(1)$

1-Piecewise Testability

Theorem

To decide whether a min. DFA recognizes a 1-PT language is in LOGSPACE.

$L(\mathscr{A})$ 1-PT iff the two patterns hold in every state and letter(s)



Syntactic monoids of 1-PTL defined by equations $x = x^2$ and $xy = yx$.[3]

[3] Simon, Blanchet-Sadri

2-Piecewise Testability

Theorem

To decide whether a min. DFA recognizes a 2-PT language is NL*-complete.*

$\mathscr{A}$ min. acyclic and confluent DFA (checked in NL); $L(\mathscr{A})$ 2-PT iff $\forall a \in \Sigma, \forall s \in Q$ s.t. $q_0 w = s$ for a $w \in \Sigma^*$ with $|w|_a \geq 1$, $sba = saba$ $\forall b \in \Sigma \cup \{\varepsilon\}$.

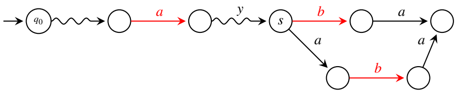
Synt. monoids of 2-PT defined by $xyzx = xyxzx$ and $(xy)^2 = (yx)^2$ (Blanchet-Sadri)

2-Piecewise Testability

Theorem

To decide whether a min. DFA recognizes a 2-PT language is NL*-complete.*

$\mathscr{A}$ min. acyclic and confluent DFA (checked in NL); $L(\mathscr{A})$ 2-PT iff
$\forall a \in \Sigma, \forall s \in Q$ s.t. $q_0 w = s$ for a $w \in \Sigma^*$ with $|w|_a \geq 1$, $sba = saba$ $\forall b \in \Sigma \cup \{\varepsilon\}$.



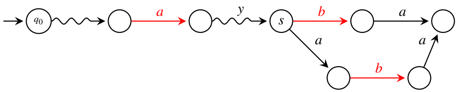
Synt. monoids of 2-PT defined by $xyzx = xyxzx$ and $(xy)^2 = (yx)^2$ (Blanchet-Sadri)

2-Piecewise Testability

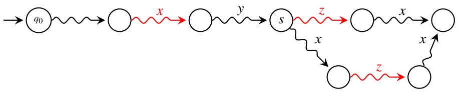
Theorem

To decide whether a min. DFA recognizes a 2-PT language is NL*-complete.*

$\mathscr{A}$ min. acyclic and confluent DFA (checked in NL); $L(\mathscr{A})$ 2-PT iff
$\forall a \in \Sigma, \forall s \in Q$ s.t. $q_0 w = s$ for a $w \in \Sigma^*$ with $|w|_a \geq 1$, $sba = saba$ $\forall b \in \Sigma \cup \{\varepsilon\}$.



Synt. monoids of 2-PT defined by $xyzx = xyxzx$ and $(xy)^2 = (yx)^2$ (Blanchet-Sadri)



3-Piecewise Testability

Theorem

To decide whether a min. DFA recognizes a 3-PT language is NL*-complete.*

Blachet-Sadri: Equations $(xy)^3 = (yx)^3$, $xzyxvxwy = xzxyxvxwy$ and $ywxvxyzx = ywxvxyxzx$

k-Piecewise Testability

Theorem

NAME: K-PIECEWISETESTABILITY
INPUT: *a minimal DFA $\mathscr{A}$*
OUTPUT: YES *if and only if $\mathscr{L}(\mathscr{A})$ is k-PT*

*Complexity: in co-*NP

- $O(1)$ *for* $k = 0$,
- LOGSPACE *for* $k = 1$,
- NL*-complete for* $k = 2, 3$,

k-Piecewise Testability

Theorem

NAME: K-PIECEWISETESTABILITY
INPUT: *a minimal DFA* $\mathscr{A}$
OUTPUT: YES *if and only if* $\mathscr{L}(\mathscr{A})$ *is k-PT*

*Complexity: in co-*NP

- $O(1)$ *for* $k = 0$,
- LOGSPACE *for* $k = 1$,
- NL*-complete for* $k = 2, 3$,

Recently, a co-NP upper bound in terms of separability
Hofman, Martens, "Separability by Short Subsequences and Subwords", ICDT 2015

k-Piecewise Testability

Theorem

NAME: K-PIECEWISETESTABILITY
INPUT: *a minimal DFA $\mathscr{A}$*
OUTPUT: YES *if and only if $\mathscr{L}(\mathscr{A})$ is k-PT*

Complexity: in co-NP

- $O(1)$ *for $k = 0$,*
- LOGSPACE *for $k = 1$,*
- NL*-complete for $k = 2, 3$,*
- *co-*NP*-complete for $k \geq 4$.*

Recently, a co-NP upper bound in terms of separability
Hofman, Martens, "Separability by Short Subsequences and Subwords", ICDT 2015

Even more recently, co-NP-completeness for $k \geq 4$
Klíma, Kunc, Polák, "Deciding k-piecewise testability", submitted, unaccessible ☺
Thanks to an anonymous reviewer and the authors

NFAs

Complexity of k-PT for NFAs

Theorem

The following problem

NAME: K-PIECEWISETESTABILITYNFA

INPUT: *an NFA* $\mathscr{A}$

OUTPUT: YES *if and only if* $\mathscr{L}(\mathscr{A})$ *is k-PT*

is PSPACE*-complete.*

Problem 2

Bounds on min. DFAs of k-PT languages

Problem (Bounds on automata of k-PT languages)

Input: $\Sigma = \{a_1, a_2, \ldots, a_n\}$, $n \geq 1$, *and* $k \geq 1$

Quest.: *length of a longest word,* w*, s.t.*

1. *sub*$_k(w) := \{u \in \Sigma^* \mid u \preccurlyeq v, |u| \leq k\}$[4] $= \Sigma^{\leq k}$,
2. *prefixes* $w_1 \neq w_2$ *of* w, *sub*$_k(w_1) \neq$ *sub*$_k(w_2)$

[4] $abc \preccurlyeq bacabbabca$

Bounds on min. DFAs of k-PT languages

Problem (Bounds on automata of k-PT languages)

Input: $\Sigma = \{a_1, a_2, \ldots, a_n\}$, $n \geq 1$, *and* $k \geq 1$

Quest.: *length of a longest word,* w*, s.t.*

1. *sub*$_k(w) := \{u \in \Sigma^* \mid u \preccurlyeq v, |u| \leq k\}$[4] $= \Sigma^{\leq k}$,
2. *prefixes* $w_1 \neq w_2$ *of* w, *sub*$_k(w_1) \neq$ *sub*$_k(w_2)$

Solution

$$|w| = \binom{k+n}{k} - 1$$

[4] $abc \preccurlyeq bacabbabca$

Consequences

Theorem (Klíma + Polák 2013)

Given a min. DFA recognizing a PT language. If the depth is k, then the language is k-PT.

[5] depth = # states on longest simple path -1; simple path = all states pairwise different

[6] states are $\sim_k$ classes: $u \sim_k v$ iff $sub_k(u) = sub_k(v)$

Consequences

Theorem (Klíma + Polák 2013)

Given a min. DFA recognizing a PT language. If the depth is k, then the language is k-PT.

Opposite does not hold.
Ex.: $(4\ell - 1)$-PTL with the min. DFA of depth $4\ell^2$, for $\ell > 1$.

[5] depth = # states on longest simple path-1; simple path = all states pairwise different

[6] states are $\sim_k$ classes: $u \sim_k v$ iff $sub_k(u) = sub_k(v)$

Consequences

Theorem (Klíma + Polák 2013)

Given a min. DFA recognizing a PT language. If the depth is k, then the language is k-PT.

Opposite does not hold.
Ex.: $(4\ell - 1)$-PTL with the min. DFA of depth $4\ell^2$, for $\ell > 1$.

Corollary (of Problem 2)

Depth[5] of min. DFA for a k-PTL over an n-letter alphabet is at most $\binom{k+n}{k} - 1$. The bound is tight.

---

[5] depth = # states on longest simple path-1; simple path = all states pairwise different

[6] states are $\sim_k$ classes: $u \sim_k v$ iff $sub_k(u) = sub_k(v)$

Consequences

Theorem (Klíma + Polák 2013)

Given a min. DFA recognizing a PT language. If the depth is k, then the language is k-PT.

Opposite does not hold.
Ex.: $(4\ell - 1)$-PTL with the min. DFA of depth $4\ell^2$, for $\ell > 1$.

Corollary (of Problem 2)

Depth[5] of min. DFA for a k-PTL over an n-letter alphabet is at most $\binom{k+n}{k} - 1$. The bound is tight.

= depth of the $\sim_k$-canonical DFA[6]

Number of equiv. classes of $\sim_k$ investigated by Karandikar, Kufleitner, Schnoebelen, "On the index of Simon's congruence for piecewise testability", IPL 2015

[5] depth = # states on longest simple path-1; simple path = all states pairwise different

[6] states are $\sim_k$ classes: $u \sim_k v$ iff $sub_k(u) = sub_k(v)$

Stirling Numbers

For positive integers k and n,

$$\binom{k+n}{k} - 1 = \frac{1}{k!}\sum_{i=1}^{k} \begin{bmatrix} k+1 \\ i+1 \end{bmatrix} n^i,$$

where $\begin{bmatrix} k \\ n \end{bmatrix}$ denotes the Stirling cyclic numbers.

k-PT, NFAs and DFAs

Theorem

For every $k \geq 2$, there exists a language L such that

- L *is* k*-PT*
- L *is not* $(k-1)$*-PT*
- L *is recognized by an NFA of depth* $k-1$*, and*
- L *is recognized by the min. DFA of depth* $2^k - 1$*.*

k-PT, NFAs and DFAs

Theorem

For every $k \geq 2$, there exists a language L such that

- *L is k-PT*
- *L is not $(k-1)$-PT*
- *L is recognized by an NFA of depth $k-1$, and*
- *L is recognized by the min. DFA of depth 2^k-1.*

Note

NFA has k states $\rightsquigarrow$ there are NFAs s.t. 2^k states of their min. DFAs form a simple path

Are NFAs better?

Are NFAs more convenient for upper bounds on k?

Are NFAs better?

Are NFAs more convenient for upper bounds on k?

No

Are NFAs better?

Are NFAs more convenient for upper bounds on k?

No

Even for 1-PT, the depth of NFA depends on the alphabet.

The language

$$L = \bigcap_{a \in \Sigma} \Sigma^* a \Sigma^*$$

is 1-PT and any NFA requires at least $2^{|\Sigma|}$ states and depth $|\Sigma|$.

Thank you!

Summary of main results

- k-PT of DFAs is in co-NP

	$k=0$	$k=1$	$k=2,3$	$k\geq 4$
Comp.	$O(1)$	LOGSPACE	NL-complete	co-NP-complete[7]

- k-PT for NFAs is PSPACE-complete
- $k,n\geq 1$, the depth of min. DFA of any k-PTL over n letters $\leq \binom{k+n}{k}-1$
- For every $k\geq 2$, there exists L s.t. L is k-PT and not (k−1)-PT, L is recognized by an NFA with k states and depth $k-1$, and the min. DFA for L has depth 2^k-1.

[7] Klíma, Kunc, Polák