On the Difference Set of two Transductions and PRAX Algorithms

Stavros Konstantinidis

Saint Mary's University, Halifax, Canada

One FLAT World Seminar talk, June 18, 2025

Based on the papers:

[SK, Mitja Mastnak, Nelma Moreira, Rog. Reis: PRAX, TCS 2023]

[SK, N.Moreira, R.Reis, Juraj Šebej: Diff sets of Transd, TCS2024]

[P. Andreou, SK, Taylor Smith: PRAX apps, JALC, to appear]

On the Difference Set of two Transductions and PRAX Algorithms

Stavros Konstantinidis

Saint Mary's University, Halifax, Canada

One FLAT World Seminar talk, June 18, 2025

Based on the papers:

[SK, Mitja Mastnak, Nelma Moreira, Rog. Reis: PRAX, TCS 2023]

[SK, N.Moreira, R.Reis, Juraj Šebej: Diff sets of Transd, TCS2024]

[P. Andreou, SK, Taylor Smith: PRAX apps, JALC, to appear]

Some General References

Automata and FLs: Salomaa; Rozenberg; Hopcroft & Ullman
Transducers: Berstel; Sakarovitch
Complexity: O. Goldreich; Arora & Barak
Probabilistic Computing: Mitzenmacher & Upfal

Difference Set of two Transducers

Let $\boldsymbol{s}$, $\boldsymbol{t}$ be two transducers with the same domain. Their `difference set` is

$$\Delta_{\boldsymbol{s},\boldsymbol{t}} = \{w \in \operatorname{dom} \boldsymbol{s} \mid \boldsymbol{s}(w) \neq \boldsymbol{t}(w)\}.$$

Their equality set is

$$\mathcal{E}_{\boldsymbol{s},\boldsymbol{t}} = \{w \in \operatorname{dom} \boldsymbol{s} \mid \boldsymbol{s}(w) = \boldsymbol{t}(w)\}.$$

The transducers are nondeterministic in general, so we have a set of outputs for any given input.

Difference Set of two Transducers

Let $\boldsymbol{s}$, $\boldsymbol{t}$ be two transducers with the same domain. Their `difference set` is

$$\Delta_{\boldsymbol{s},\boldsymbol{t}} = \{w \in \operatorname{dom} \boldsymbol{s} \mid \boldsymbol{s}(w) \neq \boldsymbol{t}(w)\}.$$

Their equality set is

$$\mathcal{E}_{\boldsymbol{s},\boldsymbol{t}} = \{w \in \operatorname{dom} \boldsymbol{s} \mid \boldsymbol{s}(w) = \boldsymbol{t}(w)\}.$$

The transducers are nondeterministic in general, so we have a set of outputs for any given input.

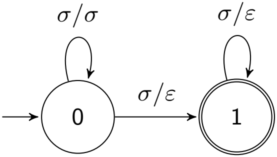
Example

Let px, sx be the prefix and suffix transducers;

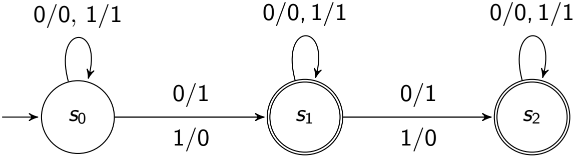
$$\mathrm{px}(w) = \text{the set of prefixes of } w.$$

Their difference set is equal to the set of all words containing at least two distinct letters.

Transducer examples

px =


E.g., $\mathtt{px}(001) = \{00, 0, \varepsilon\}$

$\mathtt{sub}_2$ =


Chomsky-like types of the languages $\Delta_{\boldsymbol{s},\boldsymbol{t}}$

Consider the language class
$\Delta(\mathrm{TR}) = \{\Delta_{\boldsymbol{s},\boldsymbol{t}} \mid \boldsymbol{s}, \boldsymbol{t} \in \mathrm{TR}, \operatorname{dom} \boldsymbol{s} = \operatorname{dom} \boldsymbol{t}\}$

Questions:

- How does this class relate to standard language classes?
- What about the classes defined when the transducers $\boldsymbol{s}$, $\boldsymbol{t}$ involved are of some restricted type?

Example

(continued) Let $\mathtt{px}$, $\mathtt{sx}$ be the prefix and suffix transducers;

$$\mathtt{px}(w) = \text{the set of prefixes of } w.$$

Their difference set is equal to the set of all words containing at least two distinct letters. Thus, $\Delta_{\mathtt{px},\mathtt{sx}}$ is a regular language: $\Delta_{\mathtt{px},\mathtt{sx}} \in \mathbf{REG}$.

Types of a transducer $\boldsymbol{t}$

- FINOUT: $\boldsymbol{t}(w)$ is finite, for all inputs w
- FINVAL: there is $k \in \mathbb{N}$ such that $|\boldsymbol{t}(w)| \leq k$ for all w
- FUNC: $|\boldsymbol{t}(w)| \leq 1$ for all w
- HOM: $\mathrm{dom}\,\boldsymbol{t} = \Sigma^*$ and the usual: $\boldsymbol{t}(xy) = \boldsymbol{t}(x)\boldsymbol{t}(y)$
- REC: recognizable (will skip this)

Types of a transducer $\boldsymbol{t}$

- FINOUT: $\boldsymbol{t}(w)$ is finite, for all inputs w
- FINVAL: there is $k \in \mathbb{N}$ such that $|\boldsymbol{t}(w)| \leq k$ for all w
- FUNC: $|\boldsymbol{t}(w)| \leq 1$ for all w
- HOM: $\mathrm{dom}\,\boldsymbol{t} = \Sigma^*$ and the usual: $\boldsymbol{t}(xy) = \boldsymbol{t}(x)\boldsymbol{t}(y)$
- REC: recognizable (will skip this)

Notation

- $\Delta(\mathtt{Y})$ denotes the class of all difference sets between transductions of type $\mathtt{Y}$.
- $\Delta(\mathtt{Y}_1, \mathtt{Y}_2)$ denotes the class of all difference sets between a transducer of type $\mathtt{Y}_1$ and one of type $\mathtt{Y}_2$.

  For example, $\Delta(\mathrm{FUNC}, \mathrm{TR})$ = the class of languages $\Delta_{\boldsymbol{s},\boldsymbol{t}}$, for some functional transducer $\boldsymbol{s}$ and some transducer $\boldsymbol{t}$.

Types of a transducer $\boldsymbol{t}$

- FINOUT: $\boldsymbol{t}(w)$ is finite, for all inputs w
- FINVAL: there is $k \in \mathbb{N}$ such that $|\boldsymbol{t}(w)| \leq k$ for all w
- FUNC: $|\boldsymbol{t}(w)| \leq 1$ for all w
- HOM: $\mathrm{dom}\,\boldsymbol{t} = \Sigma^*$ and the usual: $\boldsymbol{t}(xy) = \boldsymbol{t}(x)\boldsymbol{t}(y)$
- REC: recognizable (will skip this)

Notation

- $\Delta(\mathtt{Y})$ denotes the class of all difference sets between transductions of type $\mathtt{Y}$.
- $\Delta(\mathtt{Y}_1, \mathtt{Y}_2)$ denotes the class of all difference sets between a transducer of type $\mathtt{Y}_1$ and one of type $\mathtt{Y}_2$.

  For example, $\Delta(\mathrm{FUNC}, \mathrm{TR})$ = the class of languages $\Delta_{\boldsymbol{s},\boldsymbol{t}}$, for some functional transducer $\boldsymbol{s}$ and some transducer $\boldsymbol{t}$.
- $\mathcal{E}(\mathtt{Y})$ denotes the class of all equality sets between transductions of type $\mathtt{Y}$.

  For example, the well-known class $\mathcal{E}(\mathrm{HOM})$

Related Literature ?

Literature: We found no direct results on the $\Delta(\Upsilon)$'s, but found some in the same spirit and some that could be used to infer...

- The language that distinguishes two states of a DFA in [Cezar, Nelma, Rogerio, 2016]

Related Literature ?

Literature: We found no direct results on the $\Delta(\Upsilon)$'s, but found some in the same spirit and some that could be used to infer...

- The language that distinguishes two states of a DFA in [Cezar, Nelma, Rogerio, 2016]

- $\mathcal{E}(\text{HOM}) \subseteq \text{co}\mathbf{OCL}$. Hence, $\Delta(\text{HOM}) \subseteq \mathbf{OCL}$. [Harju, Karhumäki: Morphisms, 1997]

Related Literature ?

Literature: We found no direct results on the $\Delta(\Upsilon)$'s, but found some in the same spirit and some that could be used to infer…

- The language that distinguishes two states of a DFA in [Cezar, Nelma, Rogerio, 2016]

- $\mathcal{E}(\text{HOM}) \subseteq$ co**OCL**. Hence, $\Delta(\text{HOM}) \subseteq$ **OCL**. [Harju, Karhumäki: Morphisms, 1997]

- $\mathcal{E}(\text{FUNC}) \subseteq$ **CSL** and $\mathcal{E}(\text{FUNC}) \not\subseteq$ **CFL**, follows from [Foryś, Fixed point languages…1986]

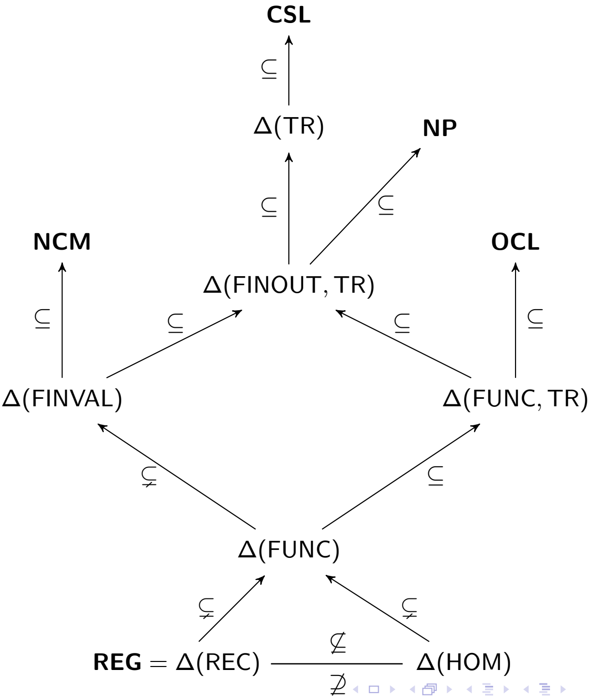
  The paper shows that the `fixed point` $\{w \mid w \in \boldsymbol{t}(w)\}$ of any transducer $\boldsymbol{t}$ is **CSL**. Then, for any $\boldsymbol{g}, \boldsymbol{h} \in \text{FUNC}$, as $\mathcal{E}_{g,h}$ is the fixed point of $\boldsymbol{h}^{-1} \circ \boldsymbol{g}$, we have that indeed $\mathcal{E}(\text{FUNC}) \subseteq$ **CSL**.

Hierarchy

The following are immediate:

$$
\begin{aligned}
\Delta(\mathsf{HOM}) &\subseteq \\
\Delta(\mathsf{FUNC}) &\subseteq \\
\Delta(\mathsf{FUNC}, \mathsf{TR}), \Delta(\mathsf{FINVAL}) &\subseteq \\
\Delta(\mathsf{FINOUT}, \mathsf{TR}) &\subseteq \\
\Delta(\mathsf{TR}) &
\end{aligned}
$$

Theorem:


CSL
Δ(TR)
NP
NCM
Δ(FINOUT, TR)
OCL
Δ(FINVAL)
Δ(FUNC, TR)
Δ(FUNC)
REG = Δ(REC)
Δ(HOM)
⊆
⊊
⊈
⊉

The proofs of the inclusions are not long, but they put together several "scattered" facts and adapt the proofs of some facts.

- $\Delta(\mathrm{FUNC}, \mathrm{TR}) \subseteq \mathbf{OCL}$: follows by adapting the proof of *"The prefix language of two* $\boldsymbol{s}, \boldsymbol{t} \in$ FUNC *is* co**OCL**" in [J. Engelfriet, H.J. Hoogeboom: IPL 1988]

  Pref($\boldsymbol{s}$,$\boldsymbol{t}$)=all w such that one of $\boldsymbol{s}(w), \boldsymbol{t}(w)$ is prefix of the other

The proofs of the inclusions are not long, but they put together several "scattered" facts and adapt the proofs of some facts.

- $\Delta(\mathrm{FUNC}, \mathrm{TR}) \subseteq \mathbf{OCL}$: follows by adapting the proof of *"The prefix language of two* $\boldsymbol{s}, \boldsymbol{t} \in \mathrm{FUNC}$ *is* co**OCL**"
  in [J. Engelfriet, H.J. Hoogeboom: IPL 1988]

  Pref($\boldsymbol{s}$,$\boldsymbol{t}$)=all w such that one of $\boldsymbol{s}(w), \boldsymbol{t}(w)$ is prefix of the other

- $\Delta(\mathrm{FUNC}) \subsetneq \Delta(\mathrm{FINVAL})$: shown using the other inclusion $\Delta(\mathrm{FUNC}) \subseteq \Delta(\mathrm{FUNC}, \mathrm{TR}) \subseteq \Delta(\mathbf{OCL})$, and the example of the finite valued transducers $\boldsymbol{s}$, $\boldsymbol{t}$ with domain $a^+b^+c^+d^+$ such that $\boldsymbol{s}(a^{n_1}b^{m_1}c^{n_2}d^{m_2}) = \{a^{n_1}, a^{m_1}\}$ and $\boldsymbol{t}(a^{n_1}b^{m_1}c^{n_2}d^{m_2}) = \{a^{n_2}, a^{m_2}\}$. Then,

$$\Delta_{\boldsymbol{s},\boldsymbol{t}} = \left\{a^{n_1}b^{m_1}c^{n_2}d^{m_2} \mid \{n_1, m_1\} \neq \{n_2, m_2\}\right\} \notin \mathbf{CFL}.$$

The proofs of the inclusions are not long, but they put together several "scattered" facts and adapt the proofs of some facts.

- $\Delta(\text{FUNC}, \text{TR}) \subseteq \textbf{OCL}$: follows by adapting the proof of *"The prefix language of two* $\boldsymbol{s}, \boldsymbol{t} \in \text{FUNC}$ *is* co**OCL**" in [J. Engelfriet, H.J. Hoogeboom: IPL 1988]

  Pref($\boldsymbol{s}$,$\boldsymbol{t}$)=all w such that one of $\boldsymbol{s}(w), \boldsymbol{t}(w)$ is prefix of the other

- $\Delta(\text{FUNC}) \subsetneq \Delta(\text{FINVAL})$: shown using the other inclusion $\Delta(\text{FUNC}) \subseteq \Delta(\text{FUNC}, \text{TR}) \subseteq \Delta(\textbf{OCL})$, and the example of the finite valued transducers $\boldsymbol{s}$, $\boldsymbol{t}$ with domain $a^+b^+c^+d^+$ such that $\boldsymbol{s}(a^{n_1}b^{m_1}c^{n_2}d^{m_2}) = \{a^{n_1}, a^{m_1}\}$ and $\boldsymbol{t}(a^{n_1}b^{m_1}c^{n_2}d^{m_2}) = \{a^{n_2}, a^{m_2}\}$. Then,

$$\Delta_{\boldsymbol{s},\boldsymbol{t}} = \{a^{n_1}b^{m_1}c^{n_2}d^{m_2} \mid \{n_1, m_1\} \neq \{n_2, m_2\}\} \notin \textbf{CFL}.$$

Corollary. We strengthen $\mathcal{E}(\text{FUNC}) \subseteq \textbf{CSL}$ to $\mathcal{E}(\text{TR}) \subseteq \textbf{CSL}$ (via our $\Delta(\text{TR}) \subseteq \textbf{CSL}$ and the known $\textbf{CSL} = \text{co}\textbf{CSL}$)

The Word Problem

The word problem (in our context) is to decide, for given transducers $\boldsymbol{s}$, $\boldsymbol{t}$ and word w whether $w \in \Delta_{\boldsymbol{s},\boldsymbol{t}}$.

The following statements hold true.

1. The word problem is **PSPACE**-complete.
2. The restriction of the word problem to the case where the first, at least, transducer has finite outputs is **NP**-complete.
3. The restriction of the word problem to the case where at least one of the transducers involved is functional is in the class **P**.

Note: Hardness reductions involve NFA universality.

PRAX Algorithms for Universality and Emptiness

PRAX: **P**olynomial-time **R**andomized **A**ppro**X**imation.

Main ideas

- View universality and emptiness as probability mass problems relative to a discrete probability distribution $T : X \to [0,1]$:

$$\sum_{x\in X} T(x) = 1 \text{ and } T(L) = \sum_{x\in L} T(x), \text{ for } L \subseteq X.$$

$T(L)$ = the probability/mass of the subset L, [Golomb, 1970]

PRAX Algorithms for Universality and Emptiness

PRAX: **P**olynomial-time **R**andomized **A**ppro**X**imation.

Main ideas

- View universality and emptiness as probability mass problems relative to a discrete probability distribution $T : X \to [0, 1]$:

  $$\sum_{x \in X} T(x) = 1 \text{ and } T(L) = \sum_{x \in L} T(x), \text{ for } L \subseteq X.$$

  $T(L)$ = the probability/mass of the subset L, [Golomb, 1970]

- Specifically:

  a subset L of the domain X is universal if $T(L) = 1$;
  a subset L of the domain X is empty if $T(L) = 0$.

PRAX Algorithms for Universality and Emptiness

PRAX: **P**olynomial-time **R**andomized **A**ppro**X**imation.

Main ideas

- View universality and emptiness as probability mass problems relative to a discrete probability distribution $T : X \to [0,1]$:

  $$\sum_{x \in X} T(x) = 1 \text{ and } T(L) = \sum_{x \in L} T(x), \text{ for } L \subseteq X.$$

  $T(L) =$ the probability/mass of the subset L, [Golomb, 1970]

- Specifically:

  a subset L of the domain X is universal if $T(L) = 1$;
  a subset L of the domain X is empty if $T(L) = 0$.

- $L \subseteq X$ is described by some algorithmic object α, that is, $L = \mathcal{L}(\alpha)$. We call α a `subset description`.

  We use $T(\alpha)$ as shorthand for $T\big(\mathcal{L}(\alpha)\big)$.

- As the problems

  $$U_T = \{\alpha : T(\alpha) = 1\} \text{ and } E_T = \{\alpha : T(\alpha) = 0\}$$

  can be hard, we seek approximate versions.

Reminder: $T(\alpha) = T\big(\mathcal{L}(\alpha)\big)$=mass of $\mathcal{L}(\alpha) \subseteq X$

Seeking approximate versions of the problems

$$U_T = \{\alpha : T(\alpha) = 1\} \quad \textit{and} \quad E_T = \{\alpha : T(\alpha) = 0\}$$

- Given (approximation) tolerance $\varepsilon \in (0,1)$, we assume we are happy to decide

$$U_{T,\varepsilon} = \{\alpha : T(\alpha) \geq 1 - \varepsilon\} \text{ and } E_{T,\varepsilon} = \{\alpha : T(\alpha) \leq \varepsilon\}.$$

(ε-close being universal) (ε-close being empty)

Reminder: $T(\alpha) = T\big(\mathcal{L}(\alpha)\big)$=mass of $\mathcal{L}(\alpha) \subseteq X$

Seeking approximate versions of the problems

$$U_T = \{\alpha : T(\alpha) = 1\} \quad \textit{and} \quad E_T = \{\alpha : T(\alpha) = 0\}$$

- Given (approximation) tolerance $\varepsilon \in (0, 1)$, we assume we are happy to decide

$$U_{T,\varepsilon} = \{\alpha : T(\alpha) \geq 1 - \varepsilon\} \text{ and } E_{T,\varepsilon} = \{\alpha : T(\alpha) \leq \varepsilon\}.$$

(ε-close being universal) (ε-close being empty)

- Unfortunately the approximate versions can be hard as well [SK, Mastnak, Moreira, Reis: TCS 2023]

Reminder: $T(\alpha) = T(\mathcal{L}(\alpha))$=mass of $\mathcal{L}(\alpha) \subseteq X$

Seeking approximate versions of the problems

$$U_T = \{\alpha : T(\alpha) = 1\} \quad \textit{and} \quad E_T = \{\alpha : T(\alpha) = 0\}$$

- Given (approximation) tolerance $\varepsilon \in (0, 1)$, we assume we are happy to decide

  $$U_{T,\varepsilon} = \{\alpha : T(\alpha) \geq 1 - \varepsilon\} \text{ and } E_{T,\varepsilon} = \{\alpha : T(\alpha) \leq \varepsilon\}.$$

  (ε-close being universal) (ε-close being empty)
- Unfortunately the approximate versions can be hard as well [SK, Mastnak, Moreira, Reis: TCS 2023]
- What about *standard* randomized complexity classes (like RP or coRP)? As all of the above problems can be hard, it is unlikely that they belong to those [S.Arora, B.Barak: Comp. Cxty 2008].

PRAX algorithm definition (universality)

The answer to [what a "good" approximation is] seems intimately related to the specific computational task at hand [O.Goldreich, 2008]

We define a `PRAX algorithm for` U_T to be a randomized decision algorithm $A(\alpha, \varepsilon)$ satisfying the following conditions:

1. if $\alpha \in U_T$ then $A(\alpha, \varepsilon) = \texttt{True}$;
2. if $\alpha \notin U_{T,\varepsilon}$ then $P[A(\alpha, \varepsilon) = \texttt{False}] \geq 3/4$;
3. $A(\alpha, \varepsilon)$ works in polynomial time w.r.t. $1/\varepsilon$ and $|\alpha|$.

PRAX algorithm definition (universality)

The answer to [what a "good" approximation is] seems intimately related to the specific computational task at hand [O.Goldreich, 2008]

We define a `PRAX algorithm for` U_T to be a randomized decision algorithm $A(\alpha, \varepsilon)$ satisfying the following conditions:

1. if $\alpha \in U_T$ then $A(\alpha, \varepsilon) = \texttt{True}$;
2. if $\alpha \notin U_{T,\varepsilon}$ then $P[A(\alpha, \varepsilon) = \texttt{False}] \geq 3/4$;
3. $A(\alpha, \varepsilon)$ works in polynomial time w.r.t. $1/\varepsilon$ and $|\alpha|$.

- When $A(\alpha, \varepsilon)$ returns the answer `False`, this answer is correct: $\alpha \notin U_T$.
- If $A(\alpha, \varepsilon)$ returns `True` then probably $\alpha \in U_{T,\varepsilon}$, in the sense that $\alpha \notin U_{T,\varepsilon}$ would imply $P[A(\alpha, \varepsilon) = \texttt{False}] \geq 3/4$.

PRAX algorithm definition (universality)

The answer to [what a "good" approximation is] seems intimately related to the specific computational task at hand [O.Goldreich, 2008]

We define a `PRAX algorithm for` U_T to be a randomized decision algorithm $A(\alpha, \varepsilon)$ satisfying the following conditions:

1. if $\alpha \in U_T$ then $A(\alpha, \varepsilon) = \texttt{True}$;
2. if $\alpha \notin U_{T,\varepsilon}$ then $P[A(\alpha, \varepsilon) = \texttt{False}] \geq 3/4$;
3. $A(\alpha, \varepsilon)$ works in polynomial time w.r.t. $1/\varepsilon$ and $|\alpha|$.

- When $A(\alpha, \varepsilon)$ returns the answer `False`, this answer is correct: $\alpha \notin U_T$.
- If $A(\alpha, \varepsilon)$ returns `True` then probably $\alpha \in U_{T,\varepsilon}$, in the sense that $\alpha \notin U_{T,\varepsilon}$ would imply $P[A(\alpha, \varepsilon) = \texttt{False}] \geq 3/4$.
- The algorithm returns the wrong answer when it returns `True` and $\alpha \notin U_{T,\varepsilon}$, but this happens with probability $\leq 1/4$.
- Running the algorithm k times reduces error probability to $\leq (1/4)^k$.

PRAX algorithm definition (emptiness)

The answer to [what a "good" approximation is] seems intimately related to the specific computational task at hand [O.Goldreich, 2008]

We define a `PRAX algorithm for` E_T to be a randomized decision algorithm $A(\alpha, \varepsilon)$ satisfying the following conditions:

1. if $\alpha \in E_T$ then $A(\alpha, \varepsilon) = \texttt{True}$;
2. if $\alpha \notin E_{T,\varepsilon}$ then $P[A(\alpha, \varepsilon) = \texttt{False}] \geq 3/4$;
3. $A(\alpha, \varepsilon)$ works in polynomial time w.r.t. $1/\varepsilon$ and $|\alpha|$.

- When $A(\alpha, \varepsilon)$ returns the answer `False`, this answer is correct: $\alpha \notin E_T$.
- If $A(\alpha, \varepsilon)$ returns `True` then probably $\alpha \in E_{T,\varepsilon}$, in the sense that $\alpha \notin E_{T,\varepsilon}$ would imply $P[A(\alpha, \varepsilon) = \texttt{False}] \geq 3/4$.
- The algorithm returns the wrong answer when it returns `True` and $\alpha \notin E_{T,\varepsilon}$, but this happens with probability $\leq 1/4$.
- Running the algorithm k times reduces error probability to $\leq (1/4)^k$.

PRAX Theorem [P.Andreou, SK, Taylor Smith: JALC, to appear]

Let T be a tractable distribution family. Let E_T be any emptiness problem and let U_T be any universality problem, where their instances α are subset descriptions with polynomially decidable $\mathcal{L}(\alpha)$. The algorithms below are PRAX algorithms (relative to T) for E_T and U_T.

Universal$_T$ (α, ε)
  $c := 1.4$;
  $n := \lceil c/(\varepsilon/2) \rceil$;
  repeat n times:
    $x :=$ select$_T(\varepsilon/2)$;
    if $(x \neq \perp$ and $x \notin \mathcal{L}(\alpha))$
      return `False`
  return `True`;

Empty$_T$ (α, ε)
  $c := 1.4$;
  $n := \lceil c/(\varepsilon/2) \rceil$;
  repeat n times:
    $x :=$ select$_T(\varepsilon/2)$;
    if $(x \neq \perp$ and $x \in \mathcal{L}(\alpha))$
      return `False`
  return `True`;

Earlier version tested for NFA universality in [SK, Mastnak, Moreira, Reis: TCS 2023]

PRAX algorithm for tractable $T = \{T_\alpha\}$

Assumptions (inspired from [S.Arora, B.Barak: Comp. Cxty 2008])

Each decision problem (U_T or E_T) and each PRAX algorithm refers to a specific family T of tractable distributions.

Each problem instance α, which we call a subset description, specifies a particular tractable distribution $T_\alpha \in T$ and a subset $\mathcal{L}(\alpha)$ of $X = \mathrm{dom}(T_\alpha)$.

Example.

- If the set of instances is the set of NFAs then each NFA α specifies the domain $X = \Sigma^*$, where Σ is the alphabet of α; and the distribution T_α is the same for any NFA α over Σ.
- If every instance α specifies a finite domain, then we use the uniform distribution on $X = \mathrm{dom}(T_\alpha)$. For example, each α is a Boolean expression in CNF with some number k, say, of variables, and $\mathcal{L}(\alpha)$ is the set of 2^k truth assignments to the k variables.

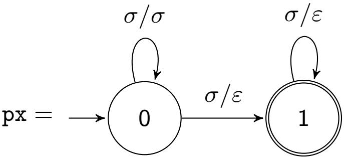
Tractable Distribution

A distribution T with domain X is called tractable if there is a (randomized) algorithm $\mathrm{select}_T(\delta)$, where $\delta \in (0,1)$, that is polynomial w.r.t. $1/\delta$ and randomly selects an element from the truncated distribution T^F such that $T(X - F) \leq \delta$, where F is a set of X-elements determined by the algorithm once (F can be defined via a static variable).

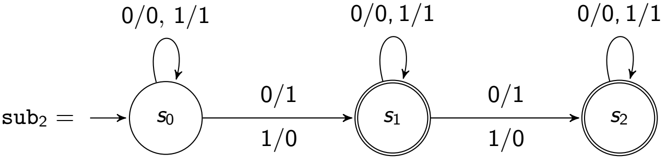
Truncated distributions. Selecting from a distribution D with infinite domain $X = \mathrm{dom} D$ could return an element of intractable size. For this reason, given a desirable "small" positive $\delta < 1$, we like to choose a finite subset F of X and then select elements from F, omitting the *tail* $X - F$ of D, provided that $D(X - F) \leq \delta$. More specifically, if $F = \{x_1, \ldots, x_k\}$, the F-truncated version of D is the distribution D^F with domain $F \cup \{\perp\}$, where '$\perp$' is an object outside of X, such that

$$D^F = \big(D(x_1), \ldots, D(x_k),\ 1 - D(F)\big).$$

Transducer examples



E.g., $\mathtt{px}(001) = \{00, 0, \varepsilon\}$



A $\boldsymbol{t}$-code is any language L s.t. $\boldsymbol{t}(L) \cap L = \emptyset$.

Maximality of Codes [SK, Mastnak, Moreira, Reis: TCS 2023]

- Let $\boldsymbol{t}$ be an input-altering transducer: $w \notin \boldsymbol{t}(w)$, $\forall$ words w.
- A $\boldsymbol{t}$-code is any language L s.t. $\boldsymbol{t}(L) \cap L = \emptyset$.

  By varying $\boldsymbol{t}$, we can model var length *and* error control codes.

Maximality of Codes [SK, Mastnak, Moreira, Reis: TCS 2023]

- Let $\boldsymbol{t}$ be an input-altering transducer: $w \notin \boldsymbol{t}(w)$, $\forall$ words w.
- A $\boldsymbol{t}$-code is any language L s.t. $\boldsymbol{t}(L) \cap L = \emptyset$.

  By varying $\boldsymbol{t}$, we can model var length *and* error control codes.
- If α is an NFA/DFA with $\mathcal{L}(\alpha)$ being a $\boldsymbol{t}$-code, then $\mathcal{L}(\alpha)$ is maximal iff the NFA $\alpha \cup \boldsymbol{t}(\alpha) \cup \boldsymbol{t}^{-1}(\alpha)$ is universal.

Theorem/Corollary. There is a PRAX algorithm for the maximality of regular $\boldsymbol{t}$-codes relative to the Dirichlet word distribution $\langle \mathrm{D}_{t,d} \rangle$.

... Maximality of Codes [SK, Mastnak, Moreira, Reis: TCS 2023]

Theorem/Corollary. There is a PRAX algorithm for the maximality of regular $\boldsymbol{t}$-codes relative to the Dirichlet word distribution $\langle \mathrm{D}_{t,d} \rangle$.

Reminder: For $t > 1$, $\zeta(t) = \sum_{n \geq 1} 1/n^t$ is a real number.

Dirichlet distribution $\mathrm{D}_{t,d}$ of [Golomb, 1970], $t > 1$, $d \geq 0$:
using it, many heuristic probability arguments based on the fictitious uniform distribution on the positive integers become rigorous statements.

$$\mathrm{D}_{t,d}(\ell) = 1/\zeta(t) \cdot 1/(1 + \ell - d)^t$$

Confirm: $\sum_{\ell \geq d} \mathrm{D}_{t,d}(\ell) \ = \ 1/\zeta(t) \cdot \sum_{\ell \geq d} 1/(1 + \ell - d)^t \ = \ 1.$

$\langle \mathrm{D}_{t,d} \rangle$: use $\mathrm{D}_{t,d}$ to select a length $\ell \geq d$, then select uniformly a word of length ℓ.

...Maximality of Codes [SK, Mastnak, Moreira, Reis: TCS 2023]

Theorem/Corollary. There is a PRAX algorithm for the maximality of regular $\boldsymbol{t}$-codes relative to the Dirichlet word distribution $\langle \mathrm{D}_{t,d} \rangle$.

Reminder: For $t > 1$, $\zeta(t) = \sum_{n \geq 1} 1/n^t$ is a real number.

Dirichlet distribution $\mathrm{D}_{t,d}$ of [Golomb, 1970], $t > 1$, $d \geq 0$:
using it, many heuristic probability arguments based on the fictitious uniform distribution on the positive integers become rigorous statements.

$$\mathrm{D}_{t,d}(\ell) = 1/\zeta(t) \cdot 1/(1 + \ell - d)^t$$

Confirm: $\sum_{\ell \geq d} \mathrm{D}_{t,d}(\ell) \;=\; 1/\zeta(t) \cdot \sum_{\ell \geq d} 1/(1 + \ell - d)^t \;=\; 1.$

$\langle \mathrm{D}_{t,d} \rangle$: use $\mathrm{D}_{t,d}$ to select a length $\ell \geq d$, then select uniformly a word of length ℓ.

<u>Nice</u>: For any $m \in \mathbb{N}$, selecting ℓ multiple of m has probability $P[\ell \text{ multiple of } m] = 1/m^t$.

Emptiness & Universality of 2D Automata

2DNFAs: [Giammarresi&Restivo], [Inque&Takanami], [Taylor],...

Theorem/Corollary. [P. Andreou, SK, Taylor: JALC, to appear]
There is a PRAX algorithm for the emptiness (and for the universality) problem of 2D automata, relative to the 2D word distribution $\langle D^2_{t,d} \rangle$, which works in time

$$O\left(1/\varepsilon \cdot \sqrt[t-1]{1/\varepsilon^2} \cdot s\right),$$

where s is the number of states of the 2D automaton used as input to the algorithm.

- The above holds for all (standard) 2D automata as they have polynomial time membership [Taylor Smith: PhD 2021]
- $\langle D^2_{t,d} \rangle$: pick lengths $k, \ell \geq d$ from $D_{t,d}$; then pick $k \times \ell$ symbols from Σ to form a 2D word over Σ.

Tautology testing

Corollary. [P. Andreou, SK, Taylor: JALC, to appear]
There is a PRAX algorithm for the problem of tautology testing that works in time $O\big(|\alpha|(1/\varepsilon)\big)$.

- Testing whether a CNF proposition α with some k variables is a tautology is a universality problem U_{B}: whether all 2^k truth assignments satisfy α, equivalently, whether $\mathrm{B}(\alpha) = 1$, where $\mathrm{B} = (\mathrm{B}_k)$ and B_k is the uniform distribution on $\{\mathrm{T}, \mathrm{F}\}^k$.

Tautology testing

Corollary. [P. Andreou, SK, Taylor: JALC, to appear]
There is a PRAX algorithm for the problem of tautology testing that works in time $O\big(|\alpha|(1/\varepsilon)\big)$.

- Testing whether a CNF proposition α with some k variables is a tautology is a universality problem U_{B}: whether all 2^k truth assignments satisfy α, equivalently, whether $\mathrm{B}(\alpha) = 1$, where $\mathrm{B} = (\mathrm{B}_k)$ and B_k is the uniform distribution on $\{\mathrm{T}, \mathrm{F}\}^k$.
- The problem is mentioned in [Fortnow: CACM 2022] as a good candidate for an interesting problem not in the class **P**. The approximate version $U_{\mathrm{B},\varepsilon}$ of U_{B} is whether $\mathrm{B}(\alpha) \geq 1 - \varepsilon$.
- Could be useful in a non rigid decision making scenario, where it is acceptable to know whether a proposition $\alpha \to \beta$ is true in "most cases" (i.e., the ratio of the satisfying truth assignments over all truth assignments is $\geq 1 - \varepsilon$).

Diophantine Equations

Motivation from: [B. Grechuk, On the smallest open diophantine equations. SIGACT News 53, 2022]

Theorem/Corollary. [P. Andreou, SK, Taylor: JALC, to appear] There is a PRAX algorithm for the emptiness (and for the universality) problem of Diophantine equations $\alpha(x, y, z) = 0$ with three nonnegative variables, relative to the 3D length distribution $\mathrm{D}^3_{t,d}$, which works in time

$$O\big(\sqrt[t-1]{1/\varepsilon^2} + 1/\varepsilon \cdot \sqrt[t-1]{1/\varepsilon} + 1/\varepsilon \cdot |\alpha| \big).$$

$\mathrm{D}^3_{t,d}$: picks three integers $k, \ell, m \geq d$ from $\mathrm{D}_{t,d}$.

Experimental "Theorem" about the Diophantine equations $x^3y^2 - z^3 \pm 6 = 0$ [B. Grechuk, SIGACT News 2022]

1. with probability $\geq 1023/1024$, the set S of solutions (of either equation) is 0.00001-close to being empty, relative to $\mathrm{D}^3_{t,d}$ with $t = 2.00$ and $d = 2$, i.e. $\mathrm{D}^3_{t,d}(S) \leq 0.00001$.
2. with probability $\geq 1023/1024$, the set S of solutions (of either equation) is 0.001-close to being empty, relative to $\mathrm{D}^3_{t,d}$ with $t = 1.50$ and $d = 2$, i.e. $\mathrm{D}^3_{t,d}(S) \leq 0.001$.
3. with probability $\geq 1023/1024$, the set S of solutions (of either equation) is 0.05-close to being empty, relative to $\mathrm{D}^3_{t,d}$ with $t = 1.25$ and $d = 2$, i.e. $\mathrm{D}^3_{t,d}(S) \leq 0.05$.

PRAX implementation: using FAdo [Reis, Moreira: FAdo]

Research Directions

- Resolve *proper containment* issues in the hierarchy of the transducer difference classes. For example, can we show that

  the **OCL** language $\{a^n b^n\}_{n \geq 0}$ is not in $\Delta(\text{FUNC})$?

  $\Delta(\text{FUNC}) \subseteq \Delta(\text{FUNC}, \text{TR})$ is proper ?

Research Directions

- Resolve *proper containment* issues in the hierarchy of the transducer difference classes. For example, can we show that

  the **OCL** language $\{a^n b^n\}_{n \geq 0}$ is not in $\Delta(\mathrm{FUNC})$?

  $\Delta(\mathrm{FUNC}) \subseteq \Delta(\mathrm{FUNC}, \mathrm{TR})$ is proper ?

- PRAX theory: One can define PRAX algorithms for the complements of the universality and emptiness problems (in a "dual" manner). Does every problem in coNP and NP, or even in PSPACE, have a PRAX algorithm ?

Research Directions

- Resolve *proper containment* issues in the hierarchy of the transducer difference classes. For example, can we show that

  the **OCL** language $\{a^n b^n\}_{n\geq 0}$ is not in $\Delta(\mathrm{FUNC})$?

  $\Delta(\mathrm{FUNC}) \subseteq \Delta(\mathrm{FUNC}, \mathrm{TR})$ is proper ?

- PRAX theory: One can define PRAX algorithms for the complements of the universality and emptiness problems (in a "dual" manner). Does every problem in coNP and NP, or even in PSPACE, have a PRAX algorithm ?

- PRAX modelling: $\mathrm{D}_{t,d}$ is biased, but approaches the fictitious uniform distribution on $\mathbb{N}$ as $t \to 1^+$.

Some details on the Dirichlet distribution [Golomb, 1970]

Below, let $d = 1$, so $\mathrm{dom}(\mathsf{D}_{t,d}) = \mathbb{N}$.

$\mathsf{D}_{t,d}$ is biased: For any $m \in \mathbb{N}$, the probability that a selected $\ell \xleftarrow{\$} \mathsf{D}_{t,d}$ is a multiple of m is $1/m^t$.

Hence, $P_t[\ell \text{ is even}] = 1/2^t$ and $P_t[\ell \text{ is odd}] = 1 - 1/2^t$.

Some details on the Dirichlet distribution [Golomb, 1970]

Below, let $d = 1$, so $\mathrm{dom}(\mathsf{D}_{t,d}) = \mathbb{N}$.

$\mathsf{D}_{t,d}$ is biased: For any $m \in \mathbb{N}$, the probability that a selected $\ell \xleftarrow{\$} \mathsf{D}_{t,d}$ is a multiple of m is $1/m^t$.
Hence, $P_t[\ell \text{ is even}] = 1/2^t$ and $P_t[\ell \text{ is odd}] = 1 - 1/2^t$.

But, if $t \to 1^+$: $\mathsf{D}_{t,d}$ approaches the fictitious uniform distribution on $\mathbb{N}$. For any m and any $r < m$, we have:

$$P_t[\ell\%m = 1] > \cdots > P_t[\ell\%m = m-1] > P_t[\ell\%m = 0] = 1/m^t$$

The above strict inequalities become $\geq$ (non-strict) when taking $\lim_{t\to 1^+} P_t[\ell\%m = r]$. Hence, $\lim_{t\to 1^+} P_t[\ell\%m = r] = 1/m$.

Some details on the Dirichlet distribution [Golomb, 1970]

Below, let $d = 1$, so $\mathrm{dom}(\mathrm{D}_{t,d}) = \mathbb{N}$.

$\mathrm{D}_{t,d}$ is biased: For any $m \in \mathbb{N}$, the probability that a selected $\ell \xleftarrow{\$} \mathrm{D}_{t,d}$ is a multiple of m is $1/m^t$.
Hence, $P_t[\ell \text{ is even}] = 1/2^t$ and $P_t[\ell \text{ is odd}] = 1 - 1/2^t$.

But, if $t \to 1^+$: $\mathrm{D}_{t,d}$ approaches the fictitious uniform distribution on $\mathbb{N}$. For any m and any $r < m$, we have:

$$P_t[\ell\%m = 1] > \cdots > P_t[\ell\%m = m-1] > P_t[\ell\%m = 0] = 1/m^t$$

The above strict inequalities become $\geq$ (non-strict) when taking $\lim_{t\to 1^+} P_t[\ell\%m = r]$. Hence, $\lim_{t\to 1^+} P_t[\ell\%m = r] = 1/m$.

Let $(m_1, m_2) = 1$. Then, for $\ell \xleftarrow{\$} \mathrm{D}_{t,d}$ we have

$$P_t[\ell\%(m_1 m_2) = 0] \;=\; P_t[\ell\%m_1 = 0] \cdot P_t[\ell\%m_2 = 0]$$

THANK YOU !