Parameterized Regular Expressions and Their Languages

Pablo Barceló
Univ. de Chile

Leonid Libkin
U. of Edinburgh

Juan Reutter
U. of Edinburgh

Parameterized regular expressions (PREs) are regular expressions with variables.

Parameterized regular expressions (PREs) are regular expressions with variables.

Given:

- Σ: a finite alphabet
- $\mathcal{V}$: a countably infinite set of variables $x, y, z, \ldots,$

a PRE over Σ is a regular expression over alphabet $\Sigma \cup \mathcal{V}$.

Parameterized regular expressions (PREs) are regular expressions with variables.

Given:

- Σ: a finite alphabet
- $\mathcal{V}$: a countably infinite set of variables $x, y, z, \ldots,$

a PRE over Σ is a regular expression over alphabet $\Sigma \cup \mathcal{V}$.

$(0x)^*1(xy)^*$ and $(0|1)^*xy(0|1)^*$ are PREs over $\{0, 1\}$.

Language of PREs?

$$(0x)^*1(xy)^* \qquad (0|1)^*xy(0|1)^*.$$

Language of PREs?

$$(0x)^*1(xy)^* \qquad\qquad (0|1)^*xy(0|1)^*.$$

Each PRE defines a regular language over $(\Sigma \cup \mathcal{V})^*$.

Language of PREs?

$$(0x)^*1(xy)^* \qquad (0|1)^*xy(0|1)^*.$$

Each PRE defines a regular language over $(\Sigma \cup \mathcal{V})^*$.

We want PREs to define languages over Σ.

How to interpret variables in PREs

For now, variables are interpreted as symbols from Σ.

Given a PRE e over Σ that uses variables $\mathcal{W} \subset \mathcal{V}$:

- A valuation for e is a mapping $\nu : \mathcal{W} \to \Sigma$.

How to interpret variables in PREs

For now, variables are interpreted as symbols from Σ.

Given a PRE e over Σ that uses variables $\mathcal{W} \subset \mathcal{V}$:

- A valuation for e is a mapping $\nu : \mathcal{W} \to \Sigma$.

Example:

$$e = (0x)^*1(xy)^*$$

How to interpret variables in PREs

For now, variables are interpreted as symbols from Σ.

Given a PRE e over Σ that uses variables $\mathcal{W} \subset \mathcal{V}$:

- A valuation for e is a mapping $\nu : \mathcal{W} \to \Sigma$.

Example:

$$e \quad = \quad (0x)^* 1 (xy)^* \qquad\qquad \nu : \; x \mapsto 0, \;\; y \mapsto 1$$

How to interpret variables in PREs

For now, variables are interpreted as symbols from Σ.

Given a PRE e over Σ that uses variables $\mathcal{W} \subset \mathcal{V}$:

- A valuation for e is a mapping $\nu : \mathcal{W} \to \Sigma$.

Example:

$$\begin{array}{rcl} e & = & (0x)^*1(xy)^* \\ \nu(e) & = & (00)^*1(01)^* \end{array} \qquad \nu : \; x \mapsto 0, \;\; y \mapsto 1$$

Semantics for PREs over Σ: Two alternatives

Let e be a PRE over Σ. Then

- $\mathcal{L}_\diamond(e) := \bigcup\{\mathcal{L}(\nu(e)) \mid \nu \text{ is a valuation for } e\}$ (possibility)

Semantics for PREs over Σ: Two alternatives

Let e be a PRE over Σ. Then

- $\mathcal{L}_\diamond(e) := \bigcup\{\mathcal{L}(\nu(e)) \mid \nu \text{ is a valuation for } e\}$ (possibility)

Example:

$e = (0x)^*1(xy)^*$

$\mathcal{L}_\diamond(e) =$

Semantics for PREs over Σ: Two alternatives

Let e be a PRE over Σ. Then

- $\mathcal{L}_\diamond(e) := \bigcup\{\mathcal{L}(\nu(e)) \mid \nu \text{ is a valuation for } e\}$ (possibility)

Example:

$e = (0x)^*1(xy)^*$

$\mathcal{L}_\diamond(e) = (00)^*1(00)^*$

Semantics for PREs over Σ: Two alternatives

Let e be a PRE over Σ. Then

- $\mathcal{L}_\diamond(e) := \bigcup\{\mathcal{L}(\nu(e)) \mid \nu \text{ is a valuation for } e\}$ (possibility)

Example:

$e = (0x)^*1(xy)^*$

$$\mathcal{L}_\diamond(e) = (00)^*1(00)^* \cup (00)^*1(01)^*$$

Semantics for PREs over Σ: Two alternatives

Let e be a PRE over Σ. Then

- $\mathcal{L}_\diamond(e) := \bigcup\{\mathcal{L}(\nu(e)) \mid \nu \text{ is a valuation for } e\}$ (possibility)

Example:

$e = (0x)^*1(xy)^*$

$\mathcal{L}_\diamond(e) = (00)^*1(00)^* \cup (00)^*1(01)^* \cup (01)^*1(10)^*$

Semantics for PREs over Σ: Two alternatives

Let e be a PRE over Σ. Then

- $\mathcal{L}_\diamond(e) := \bigcup\{\mathcal{L}(\nu(e)) \mid \nu \text{ is a valuation for } e\}$ (possibility)

Example:

$e = (0x)^*1(xy)^*$

$\mathcal{L}_\diamond(e) = (00)^*1(00)^* \cup (00)^*1(01)^* \cup (01)^*1(10)^* \cup (01)^*1(11)^*$

Semantics for PREs over Σ: Two alternatives

Let e be a PRE over Σ. Then

- $\mathcal{L}_\diamond(e) := \bigcup\{\mathcal{L}(\nu(e)) \mid \nu \text{ is a valuation for } e\}$ (possibility)

Example:

$e = (0x)^*1(xy)^*$

$$\mathcal{L}_\diamond(e) = (00)^*1(00)^* \cup (00)^*1(01)^* \cup (01)^*1(10)^* \cup (01)^*1(11)^*$$

- 00101 is in $\mathcal{L}_\diamond(e)$.

Semantics for PREs over Σ: Two alternatives

Let e be a PRE over Σ. Then

- $\mathcal{L}_{\square}(e) := \bigcap\{\mathcal{L}(\nu(e)) \mid \nu \text{ is a valuation for e}\}$ (certainty)

Semantics for PREs over Σ: Two alternatives

Let e be a PRE over Σ. Then

- $\mathcal{L}_{\Box}(e) := \bigcap\{\mathcal{L}(\nu(e)) \mid \nu \text{ is a valuation for e}\}$ (certainty)

Example:

$e = (0|1)^* xy (0|1)^*$

Semantics for PREs over Σ: Two alternatives

Let e be a PRE over Σ. Then

- $\mathcal{L}_{\Box}(e) := \bigcap\{\mathcal{L}(\nu(e)) \mid \nu \text{ is a valuation for e}\}$ (certainty)

Example:

$e = (0|1)^* xy (0|1)^*$

$\mathcal{L}_{\Box}(e) = (0|1)^* 00 (0|1)^*$

Semantics for PREs over Σ: Two alternatives

Let e be a PRE over Σ. Then

- $\mathcal{L}_\Box(e) := \bigcap\{\mathcal{L}(\nu(e)) \mid \nu \text{ is a valuation for e}\}$ (certainty)

Example:

$e = (0|1)^* xy (0|1)^*$

$\mathcal{L}_\Box(e) = (0|1)^* 00 (0|1)^* \cap (0|1)^* 01 (0|1)^*$

Semantics for PREs over Σ: Two alternatives

Let e be a PRE over Σ. Then

- $\mathcal{L}_{\Box}(e) := \bigcap\{\mathcal{L}(\nu(e)) \mid \nu \text{ is a valuation for e}\}$ (certainty)

Example:

$e = (0|1)^* xy (0|1)^*$

$$\mathcal{L}_{\Box}(e) = (0|1)^* 00 (0|1)^* \cap (0|1)^* 01 (0|1)^* \cap (0|1)^* 10 (0|1)^*$$

Semantics for PREs over Σ: Two alternatives

Let e be a PRE over Σ. Then

- $\mathcal{L}_{\Box}(e) := \bigcap\{\mathcal{L}(\nu(e)) \mid \nu \text{ is a valuation for e}\}$ (certainty)

Example:

$$e = (0|1)^* xy (0|1)^*$$

$$\mathcal{L}_{\Box}(e) = (0|1)^* 00 (0|1)^* \cap (0|1)^* 01 (0|1)^* \cap (0|1)^* 10 (0|1)^* \cap (0|1)^* 11 (0|1)^*$$

Semantics for PREs over Σ: Two alternatives

Let e be a PRE over Σ. Then

- $\mathcal{L}_{\Box}(e) := \bigcap\{\mathcal{L}(\nu(e)) \mid \nu \text{ is a valuation for e}\}$ (certainty)

Example:

$e = (0|1)^* xy (0|1)^*$

$$\mathcal{L}_{\Box}(e) = (0|1)^*00(0|1)^* \cap (0|1)^*01(0|1)^* \cap (0|1)^*10(0|1)^* \cap (0|1)^*11(0|1)^*$$

- 10011 is in $\mathcal{L}_{\Box}(e)$.

Semantics for PREs over Σ: Two alternatives

Let e be a PRE over Σ. Then

- $\mathcal{L}_{\Box}(e) := \bigcap\{\mathcal{L}(\nu(e)) \mid \nu$ is a valuation for e$\}$ (certainty)

Example:

$e = (0|1)^* xy (0|1)^*$

$$\mathcal{L}_{\Box}(e) = (0|1)^*00(0|1)^* \cap (0|1)^*01(0|1)^* \cap (0|1)^*10(0|1)^* \cap (0|1)^*11(0|1)^*$$

- 10011 is in $\mathcal{L}_{\Box}(e)$.
- No word of length ≤ 4 is in $\mathcal{L}_{\Box}(e)$

Semantics for PREs over Σ: Two alternatives

Let e be a PRE over Σ. Then

- $\mathcal{L}_\Diamond(e) := \bigcup\{\mathcal{L}(\nu(e)) \mid \nu \text{ is a valuation for } e\}$ (possibility)
- $\mathcal{L}_\Box(e) := \bigcap\{\mathcal{L}(\nu(e)) \mid \nu \text{ is a valuation for e}\}$ (certainty)

Semantics for PREs over Σ: Two alternatives

Let e be a PRE over Σ. Then

- $\mathcal{L}_\Diamond(e) := \bigcup\{\mathcal{L}(\nu(e)) \mid \nu \text{ is a valuation for } e\}$ (possibility)
- $\mathcal{L}_\Box(e) := \bigcap\{\mathcal{L}(\nu(e)) \mid \nu \text{ is a valuation for e}\}$ (certainty)

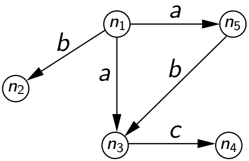
Finite unions or intersections of regular languages:

$\mathcal{L}_\Diamond(e)$ and $\mathcal{L}_\Box(e)$ are regular languages

Applications of PREs: Graph databases

Graph DBs:

- Applications: RDF, SNs, Scientific data, etc.
- Model: Edge-labeled directed graphs (that is: NFAs).



Applications of PREs: Graph databases

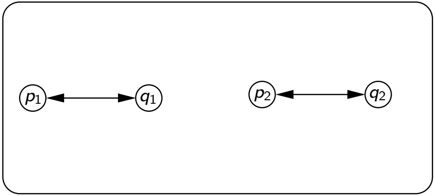
As it is usual, some data in the graph DB may be missing [Barceló. et al. 2011,Calvanese et al. 2011].

Applications of PREs: Graph databases

As it is usual, some data in the graph DB may be missing [Barceló. et al. 2011,Calvanese et al. 2011].

Example: Biological DB

- Proteins $p_1, q_1,\ p_2, q_2$

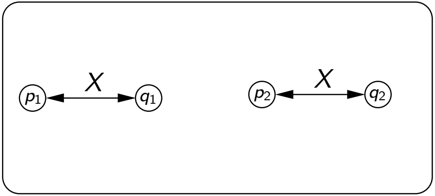


Applications of PREs: Graph databases

As it is usual, some data in the graph DB may be missing [Barceló. et al. 2011,Calvanese et al. 2011].

Example: Biological DB

- Proteins $p_1, q_1,\ p_2, q_2$
- we do not know the actual relationship

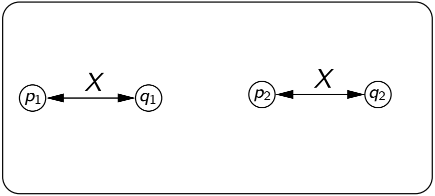


Applications of PREs: Graph databases

As it is usual, some data in the graph DB may be missing [Barceló. et al. 2011,Calvanese et al. 2011].

Example: Biological DB

- Proteins $p_1, q_1,\ p_2, q_2$
- we do not know the actual relationship



Incomplete graph DBs are graph DBs with edges labeled in $\mathcal{V}$.

- They can be represented as NFAs over $\Sigma \cup \mathcal{V}$.
- Equivalently, as PREs over Σ.

Applications of PREs: Graph Databases

Standard semantics for incomplete DBs: Certain answers.

- Answers that hold regardless of the interpretation of the variables.

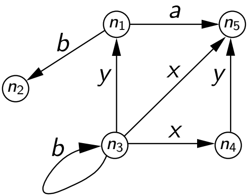
Applications of PREs: Graph Databases

Standard semantics for incomplete DBs: Certain answers.

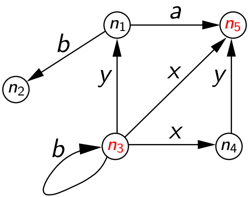
- Answers that hold regardless of the interpretation of the variables.

How to use PREs to compute certain answers over graph DBs?

PRE's for querying incomplete graph DB's

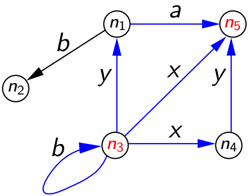


PRE's for querying incomplete graph DB's



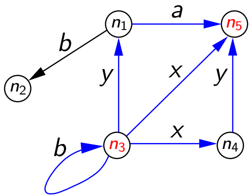
- Paths from n_3 to n_5.

PRE's for querying incomplete graph DB's



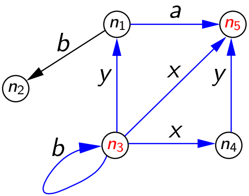
- Paths from n_3 to n_5.
- $e = b^*ya|\ b^*x|\ b^*xy$.

PRE's for querying incomplete graph DB's



- Paths from n_3 to n_5.
- $e = b^* ya|\ b^* x|\ b^* xy$.
- We can be certain about a word $w \in \Sigma^*$ labeling a path from n_3 to n_5 in G iff $w \in \mathcal{L}_{\Box}(e)$.

PRE's for querying incomplete graph DB's



The certainty semantics is essential for computing certain answers over incomplete graph DBs.

Applications of PREs: Program analysis

PREs naturally arise in program analysis [Liu & Stoller 2004, de Moor at al. 2003].

- Alphabet: Operations on variables; e.g. **def**, **use**, **open**, etc.
- Variables: Program variables, pointers, files, etc.

PREs are used in this setting to specify undesired behavior.

Example: The undesired property "A variable is used without being defined" can be expressed as follows:

$$(\neg \mathbf{def}(x))^* \mathbf{use}(x).$$

Applications of PREs: Program analysis

PREs naturally arise in program analysis [Liu & Stoller 2004, de Moor at al. 2003].

- Alphabet: Operations on variables; e.g. **def**, **use**, **open**, etc.
- Variables: Program variables, pointers, files, etc.

PREs are used in this setting to specify undesired behavior.

Example: The undesired property "A variable is used without being defined" can be expressed as follows:

$$(\neg \mathbf{def}(x))^* \mathbf{use}(x).$$

These expressions are evaluated over graphs that serve as an abstraction of the program behavior.

Applications of PREs: Program analysis

PREs specify undesired behavior: Assignments of the variables that "satisfy" the PRE represent *bugs* of the program.

In the program analysis context the possibility semantics is essential for finding where the program fails a specification.

We study basic computational problems of PREs

Despite its importance, basic computational problems associated with PREs have not been addressed.

In this paper: Study standard language-theoretical problems for PREs divided as follows:

- Decision problems: Emptiness, universality, containment and membership.
- Computational problems: Minimal-size NFAs representing $\mathcal{L}_\Box(e)$ and $\mathcal{L}_\Diamond(e)$.

- Upper bound techniques
- Decision problems
- Computational problems
- Extending the semantics
- Future work

- Upper bound techniques
- Decision problems
- Computational problems
- Extending the semantics
- Future work

NFAs for $\mathcal{L}_\Diamond(e)$ and $\mathcal{L}_\Box(e)$

- Exponentially many valuations: $|\Sigma|^{(\#\text{ of variables})}$.
- Taking the union gives an exponential NFA for $\mathcal{L}_\Diamond(e)$.
- Taking the intersection gives a doubly-exponential NFA for $\mathcal{L}_\Box(e)$.

We shall see that these are tight bounds...

- Upper bound techniques
- Decision problems
- Computational problems
- Extending the semantics
- Future work

In order to do a finer analysis we study two restrictions of PREs:

- Simple: No repetition of variables; e.g. $e = (0|1)^*xy(0|1)^*$.
- Star-height 0: No Kleene-star: i.e. finite languages.

Decision problems: Nonemptiness

- $\textsc{Nonemptiness}_{\Diamond}$: $\mathcal{L}_{\Diamond}(e) \neq \emptyset$?

- $\textsc{Nonemptiness}_{\Box}$: $\mathcal{L}_{\Box}(e) \neq \emptyset$?

Decision problems: Nonemptiness

- $\textsc{Nonemptiness}_{\diamond}$: $\mathcal{L}_{\diamond}(e) \neq \emptyset$?

  Not different from the case without variables:

  $$(0x)^*1^*(xy)^*$$

- $\textsc{Nonemptiness}_{\Box}$: $\mathcal{L}_{\Box}(e) \neq \emptyset$?

Decision problems: Nonemptiness

- $\textsc{Nonemptiness}_\Diamond$: $\mathcal{L}_\Diamond(e) \neq \emptyset$?

  Not different from the case without variables:

$$(0x)^*1^*(xy)^*$$

- $\textsc{Nonemptiness}_\Box$: $\mathcal{L}_\Box(e) \neq \emptyset$?

Theorem

$\textsc{Nonemptiness}_\Box$ is $\textsc{Expspace}$-complete.

Decision problems: Nonemptiness

- $\textsc{Nonemptiness}_{\Diamond}$: $\mathcal{L}_{\Diamond}(e) \neq \emptyset$?

  Not different from the case without variables:

$$(0x)^* 1^* (xy)^*$$

- $\textsc{Nonemptiness}_{\Box}$: $\mathcal{L}_{\Box}(e) \neq \emptyset$?

Theorem

$\textsc{Nonemptiness}_{\Box}$ is $\textsc{Expspace}$-complete.

1. Remains $\textsc{Expspace}$-hard even over the class of simple expressions.
2. For PREs of star-height 0: $\textsc{Nonemptiness}_{\Box}$ is Σ_2^P-complete.

PREs and succinct intersection

Main tool for EXPSPACE-hardness:

Given PRE's $e_1, \ldots, e_n$ we can construct in polynomial time a PRE e' such that

$$\mathcal{L}_\Box(e') \text{ is empty iff } \mathcal{L}_\Box(e_1) \cap \cdots \cap \mathcal{L}_\Box(e_n) \text{ is empty.}$$

PREs and succinct intersection

Main tool for EXPSPACE-hardness:

Given PRE's $e_1, \ldots, e_n$ we can construct in polynomial time a PRE e' such that

$$\mathcal{L}_\square(e') \text{ is empty iff } \mathcal{L}_\square(e_1) \cap \cdots \cap \mathcal{L}_\square(e_n) \text{ is empty.}$$

Gives us PSPACE-hardness for NONEMPTINESS$_\square$, since regular expressions are PRE's

Minimal size of words in $\mathcal{L}_{\square}(e)$

Corollary ($\textsc{Nonemptiness}_{\square}$)

There exists a sequence of parameterized regular expressions $\{e_n\}_{n \in \mathbb{N}}$ such that:

1. *Each e_n is of size polynomial in n.*
2. *Every word in the language $\mathcal{L}_{\square}(e_n)$ has size at least 2^{2^n}.*

Minimal size of words in $\mathcal{L}_\square(e)$: exponential bound

Consider PREs of the form:

$$(0 \mid 1)^* \, x_1 \cdot x_2 \cdots x_n \, (0 \mid 1)^* \qquad (n \geq 1).$$

Minimal size of words in $\mathcal{L}_{\square}(e)$: exponential bound

Consider PREs of the form:

$$(0 \,|\, 1)^* \, x_1 \cdot x_2 \cdots x_n \, (0 \,|\, 1)^* \qquad (n \geq 1).$$

- If $w \in \mathcal{L}_{\square}(e)$, then it contains as a subword each $w' \in \{0, 1\}^n$.

Minimal size of words in $\mathcal{L}_{\square}(e)$: exponential bound

Consider PREs of the form:

$$(0 \mid 1)^* \, x_1 \cdot x_2 \cdots x_n \, (0 \mid 1)^* \qquad (n \geq 1).$$

- If $w \in \mathcal{L}_{\square}(e)$, then it contains as a subword each $w' \in \{0, 1\}^n$.

de Bruijn sequences of order n, which are of size $\geq 2^n$.

Decision problems: Universality

$\textsc{Universality}_\circ$: Is $\mathcal{L}_\circ(e) = \Sigma^*$?

As opposed to nonemptiness, universality is more difficult for the $\Diamond$-semantics than for the $\Box$-semantics:

- $\textsc{Universality}_\Box$ is $\textsc{Pspace}$-complete.
- $\textsc{Universality}_\Diamond$ is $\textsc{Expspace}$-complete.
  - It remains $\textsc{Expspace}$-complete even over the class of simple expressions.

Decision problems: Containment

$\textsc{Containment}_\circ$: Is $\mathcal{L}_\circ(e_1) \subseteq \mathcal{L}_\circ(e_2)$?

We can reduce from other problems, since:

- L is empty iff $L \subseteq \emptyset$

Decision problems: Containment

$\textsc{Containment}_\circ$: Is $\mathcal{L}_\circ(e_1) \subseteq \mathcal{L}_\circ(e_2)$?

We can reduce from other problems, since:

- L is empty iff $L \subseteq \emptyset$
- L is Σ^* iff $\Sigma^* \subseteq L$.

Thus,

$\textsc{Containment}_\Box$ and $\textsc{Containment}_\Diamond$ are $\textsc{Expspace}$-complete.

- Even if restricted to simple expressions.

Decision problems: Membership

Is w in $\mathcal{L}_\Diamond(e)$ or $\mathcal{L}_\Box(e)$?

Guess a valuation ν:

- $w \in \mathcal{L}(\nu(e))$ (possibility)
- $w \notin \mathcal{L}(\nu(e))$ (certainty)

Gives us NP and coNP bounds

Decision problems: Membership

Is w in $\mathcal{L}_\Diamond(e)$ or $\mathcal{L}_\Box(e)$?

Guess a valuation ν:

- $w \in \mathcal{L}(\nu(e))$ (possibility)
- $w \notin \mathcal{L}(\nu(e))$ (certainty)

Gives us NP and coNP bounds (tight).

Theorem

- $\textsc{Membership}_\Diamond$ is NP-complete.
- $\textsc{Membership}_\Box$ is coNP-complete.

Decision problems: Membership

We can do a finer analysis:

Proposition

- The complexity of $\textsc{Membership}_{\Diamond}$ is as follows:
  1. Simple expressions: NP-complete.
  2. Star-height 0 expressions: NP-complete.
  3. Simple and star-height 0 expressions: $\textsc{Ptime}$.
- The complexity of $\textsc{Membership}_{\Box}$ is as follows:
  1. Simple expressions: $\textsc{co}\mathrm{NP}$-complete.
  2. Star-height 0 expressions: $\textsc{co}\mathrm{NP}$-complete.
  3. Simple and star-height 0 expressions: $\textsc{Ptime}$.

Also in the paper:

- CONTAINMENT when one expression is fixed.

Also in the paper:

- Containment when one expression is fixed.
- Membership when the word is fixed.

Also in the paper:

- CONTAINMENT when one expression is fixed.
- MEMBERSHIP when the word is fixed.
- Emptiness of the intersection with a regular language

Also in the paper:

- CONTAINMENT when one expression is fixed.
- MEMBERSHIP when the word is fixed.
- Emptiness of the intersection with a regular language

These problems are motivated by the application of PREs

- Upper bound techniques
- Decision problems
- Computational problems
- Extending the semantics
- Future work

Computational problems

What is the size of the minimal NFA $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}_\diamond(e)$ or $\mathcal{L}(\mathcal{A}) = \mathcal{L}_\Box(e)$?

Computational problems

What is the size of the minimal NFA $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}_\diamond(e)$ or $\mathcal{L}(\mathcal{A}) = \mathcal{L}_\Box(e)$?

Theorem

The sizes of minimal NFAs are:

- necessarily double-exponential for $\mathcal{L}_\Box$
- necessarily exponential for $\mathcal{L}_\diamond$.

Proof sketch of minimal size NFA for $\mathcal{L}_\Box$

We use the following result by Glaister and Shallit:

If L is a regular language, and there exists a set of pairs

$$P = \{(u_i, v_i) \mid 1 \leq i \leq m\} \subseteq \Sigma^* \times \Sigma^*, \text{ such that}$$

1. $u_i v_i \in L$,
2. $u_j v_i \notin L$ for $i \neq j$,

then every NFA accepting L has at least m states.

Proof skecth of minimal size NFA for $\mathcal{L}_\square$

Consider the following family of PREs:

$$e_n = ((0 \mid 1)^{n+1})^* \cdot x_1 \cdots x_n \cdot x_{n+1} \cdot ((0 \mid 1)^{n+1})^* \qquad (n \geq 1)$$

- Each e_n is of linear size on n.

We shall construct a Fooling Set for $\mathcal{L}_\square(e_n)$.

Proof sketch of minimal size NFA for $\mathcal{L}_\Box$

Given a set $S \subset \{0,1\}^{n+1}$ of size 2^n:

- w_S is the concatenation in lexicographical order of all words in S; and
- $w_{\bar{S},n}$ is the concatenation in lexicographical order of all words in $\{0,1\}^{n+1}$ that are not in S.

Proof skecth of minimal size NFA for $\mathcal{L}_\square$

We define:

$$P_n \;:=\; \{(w_S, w_{\bar{S},n}) \mid S \subset \{0,1\}^{n+1} \text{ and } |S| = 2^n\},$$

Proof sketch of minimal size NFA for $\mathcal{L}_\Box$

We define:

$$P_n \;:=\; \{(w_S, w_{\bar{S},n}) \mid S \subset \{0,1\}^{n+1} \text{ and } |S| = 2^n\},$$

1. There are $\binom{2^{n+1}}{2^n} \geq 2^{2^n}$ different subsets of $\{0,1\}^{n+1}$ of size 2^n, and thus $|P_n| \geq 2^{2^n}$.
2. $(w_S, w_{\bar{S},n})$ belongs to $\mathcal{L}_\Box(e_n)$, but
3. $(w_{S_1}, w_{\bar{S}_2,n})$ are not in $\mathcal{L}_\Box(e_n)$, for distinct S_1 and S_2.

- Upper bound techniques
- Decision problems
- Computational problems
- Extending the semantics
- Future work

Extending Semantics

We can extend semantics and allow replacement of variables by words that belong to some regular language.

- $\Diamond$-semantics: Easily becomes non-regular (e.g. xx = squared words). Regular for finite languages.

Extending Semantics

We can extend semantics and allow replacement of variables by words that belong to some regular language.

- $\Diamond$-semantics: Easily becomes non-regular (e.g. xx = squared words). Regular for finite languages.
- $\Box$-semantics: Keeps being regular. Same complexity bounds apply.

- Upper bound techniques
- Decision problems
- Computational problems
- Extending the semantics
- Future work

Future work

Closure properties:

- The minimal NFA $\mathcal{A}$ that accepts $\mathcal{L}_{\Box}(e_1) \cap \mathcal{L}_{\Box}(e_2)$ is necessarily of double-exponential size.

Future work

Closure properties:

- The minimal NFA $\mathcal{A}$ that accepts $\mathcal{L}_{\square}(e_1) \cap \mathcal{L}_{\square}(e_2)$ is necessarily of double-exponential size.
- Perhaps it is possible to construct in polynomial time a PRE e such that $\mathcal{L}_{\square}(e) = \mathcal{L}_{\square}(e_1) \cap \mathcal{L}_{\square}(e_2)$.