Tight Bounds for Regular Expression Pattern Matching and Membership

Master's Seminar

Philipp J. Schepper

January 16, 2020

Department of Computer Science, Saarland University — IMPRS-CS, MPI Informatik

Motivation

Regular Expressions are used for

- Text analysis and manipulation (e.g. unix tools `grep` and `sed`)
- Network analysis
- Searching for proteins in DNA sequences
- Human-computer interaction

Definition (Membership)

Input: Text t of length n and pattern p of size m.

Question: Does p generate t, i.e. $t \in \mathcal{L}(p)$?

Definition (Pattern Matching)

Question: Does p generate some *substring* of t, i.e. $t \in \mathcal{M}(p) := \Sigma^*\mathcal{L}(p)\Sigma^*$?

How fast can these problems be solved? $\mathcal{O}(nm)$! $o(nm)$? $\Omega(nm)$?

Agenda

1. Introduction and Results
2. Lower Bounds
3. Upper Bounds
4. Conclusion

Introduction and Results

Recap: Regular Expressions

Name	Regular Expression	Language
Symbol	σ	$\mathcal{L}(\sigma) := \{\sigma\}$
Alternative	$(p \mid q)$	$\mathcal{L}(p \mid q) := \mathcal{L}(p) \cup \mathcal{L}(q)$
Concatenation	$p \circ q$	$\mathcal{L}(p \circ q) := \{tu \mid t \in \mathcal{L}(p) \wedge u \in \mathcal{L}(q)\}$
Kleene Plus	p^+	$\mathcal{L}(p^+) := \bigcup_{i=1}^{\infty} \mathcal{L}(\underbrace{p \circ \ldots \circ p}_{i})$
Kleene Star	p^*	$\mathcal{L}(p^*) := \{\varepsilon\} \cup \mathcal{L}(p^+)$

n text length, m pattern size (=number of operators and symbols)

Current Results

Upper bounds

- Classical (Thompson 1968): $\mathcal{O}(nm)$
- Myers 1992: $\mathcal{O}(nm/\log n)$
- Bille and Thorup 2009: $\bar{\mathcal{O}}(nm/\log^{3/2} n)$ ($\bar{\mathcal{O}}$ hides $\operatorname{poly}\log\log n$ factors)

Lower bounds

- Backurs and Indyk 2016: $\Omega((nm)^{1-\epsilon})$, assuming SETH
- Abboud and Bringmann 2018: $\Omega(nm/\log^{7+\epsilon} n)$, assuming FSH

$\rightarrow$ Matching lower and upper bound up to a constant number of log-factors for the **general** case!

What about "**easier**" patterns? What is an "easy" pattern?

Homogeneous Patterns

Represent the patterns as trees:

- Leaves are labeled with symbols from Σ
- Inner nodes are labeled with operations $(|, \circ, +, \star)$

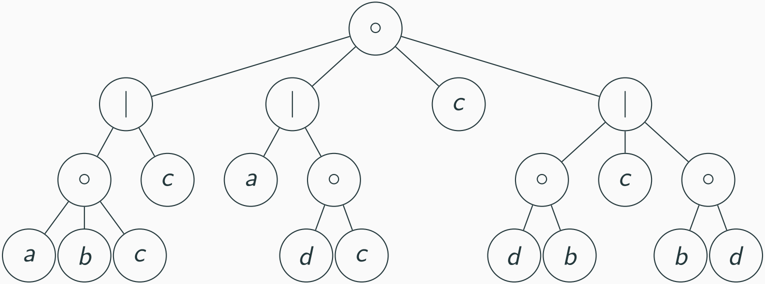
Definition (Homogeneous Patterns)

A pattern is *homogeneous* if for each level of the tree the inner nodes are labeled with the same operation. The *type* is the sequence of operators on the longest path path from the root to the deepest leaf.

Homogeneous Patterns – Example i

Definition (Homogeneous Patterns)

A pattern is *homogeneous* if for each level of the tree the inner nodes are labeled with the same operation. The *type* is the sequence of operators on the path from the root to the deepest leaf.



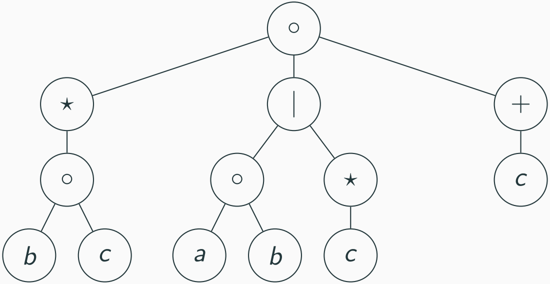
$(abc \mid c)(a \mid dc)c(db \mid c \mid bd)$:

Homogeneous pattern of type $\circ|\circ$ and depth 3.

Homogeneous Patterns – Example ii

Definition (Homogeneous Patterns)

A pattern is *homogeneous* if for each level of the tree the inner nodes are labeled with the same operation. The *type* is the sequence of operators on the path from the root to the deepest leaf.



$(bc)^*(ab \mid c^*)c^+$: Not a homogeneous pattern!

Current Results

General Patterns

$\bar{\mathcal{O}}(nm/\log^{3/2} n)$ and $\Omega(nm/\log^{7+\epsilon} n)$, assuming FSH

Homogeneous Patterns

Several common problems can be formulated as problems with homogeneous pattern types (e.g. dictionary, superset and string matching). And are solvable in $\mathcal{O}(n \log^2 m + m)$ time.

Dichotomy for homogeneous types [BI16], [BGL17]

- It suffices to analyse few pattern types of constant depth
- For "easy" patterns: strongly sub-quadratic time algorithms
- For "hard" patterns: $\Omega((nm)^{1-\epsilon})$ lower bound, assuming SETH

Questions: Does the general lower bound transfer to the hard pattern? Are there super-poly-logarithmic improvements as for APSP and OV?

Satisfiability Problems i

SETH rules out **polynomial** improvements.

We want to rule out **log-factor** improvements!

$\rightarrow$ We need a stronger assumption!

FORMULA-SAT (Abboud and Bringmann 2018)

Input: A De Morgan formula F of size s on n variables.

Task: Check whether there is a satisfying assignment for F.

De Morgan Formula

A node labeled tree. Each inner node is labeled with AND or OR. Each leaf is labeled with a variable or its negation. Size = number of leaves

FORMULA-SAT HYPOTHESIS (FSH) (Abboud and Bringmann 2018)

FORMULA-SAT on De Morgan formulas of size $s = n^{3+\Omega(1)}$ cannot be solved in $\mathcal{O}(2^n/n^\epsilon)$ time, for some $\epsilon > 0$, in the Word-RAM model.

Satisfiability Problems ii

Definition (Formula-Pair (Abboud and Bringmann 2018))

Input: A *monotone* De Morgan formula F with s inputs and $A, B \subseteq \{0,1\}^{s/2}$ of size n and m, respectively.
Task: Check whether there are $a \in A, b \in B$ such that $F(a, b) = \text{true}$.

Formula-SAT and Formula-Pair are related by writing down all half-assignments explicitly and we can ensure each input is used exactly once.

Consequence of Formula-Sat Hypothesis:

Formula-Pair Hypothesis (FPH)

For all $k \geq 1$: For a monotone De Morgan formula F of size s and sets $A, B \subseteq \{0,1\}^{s/2}$ of n half-assignments each, Formula-Pair cannot be solved in time $\mathcal{O}(n^2 s^k / \log^{3k+2} n)$, in the Word-RAM model.

Our Results

Before:

In general: $\bar{\mathcal{O}}(nm/\log^{3/2} n)$

For "hard" homogeneous patterns: $\Omega((nm)^{1-\epsilon})$, assuming SETH.

New bounds assuming FPH:

<table>
<tr><th></th><th>$\circ\star$, $\circ{+}\circ$, $\circ|\circ$, $\circ{+}|$, $\circ|{+}$</th><th>$|\circ|$, $|\circ{+}$</th><th>$|{+}|\circ$</th></tr>
<tr><td>Pattern matching</td><td rowspan="2">$\Omega\left(\frac{nm}{\log^{81} n}\right)$</td><td>$\frac{nm}{2^{\Theta\left(\sqrt{\log n}\right)}}$</td><td>$\Theta(n+m)$</td></tr>
<tr><td>Membership</td><td>$\Theta(n+m)$</td><td>$\Omega\left(\frac{nm}{\log^{17} n}\right)$</td></tr>
</table>

- $2^{\Theta(\sqrt{\log n})} \in \omega(\text{poly} \log n)$:
  Currently fastest algorithm and best we can hope for under SETH.
- For "ultra-hard" pattern types: The general algorithm is optimal up to a constant number of log-factors.

Tight classification for these pattern types (up to log-factors).

Lower Bounds

General Idea

Definition (Formula-Pair)

Input: A *monotone* De Morgan formula F with s inputs and $A, B \subseteq \{0,1\}^\ell$ of size n and m, respectively.

Task: Check whether $\exists a \in A, b \in B : F(a, b) = \text{true}$.

Reduce **Formula-Pair** to **pattern matching** with a text t and pattern p of a specific type:

$$(\exists a \in A, b \in B : F(a, b) = \text{true}) \iff t \in \mathcal{M}(p)$$

We first encode the formula such that

$$F(a, b) = \text{true} \iff t(a) \in \mathcal{L}(p(b)) \quad \forall a \in A, b \in B$$

Encode the INPUT, AND and OR gates of the formula inductively.

Encoding the Formula i

Focus on patterns with type ∘+∘: $ab(bc)^+b^+(cd)^+$

INPUT Gate $F_g(a,b) = a_i$

if $a_i = 1$ ↑

$t_g := 0a_i1$ $\qquad$ $u_g := 0011$

$p_g := 0^+11^+$ $\qquad$ $q_g := 0^+1^+$

For $F_g(a,b) = b_i$ define: $t_g := 011 \quad p_g := 0^+b_i1^+$

AND Gate $F_g(a,b) = F_{g_1}(a,b) \wedge F_{g_2}(a,b)$

if $t_i \in \mathcal{L}(p_i)$
for $i = 1, 2$ ↑

$t_g := t_1 G t_2$ $\qquad$ $u_g := u_1 G u_2$

$p_g := p_1 G p_2$ $\qquad$ $q_g := q_1 G q_2$

Define separator gadget $G := 2\langle g \rangle 2$.

Need universal text u_g and pattern q_g for OR gate.

Encoding the Formula ii

OR Gate $F_g(a, b) = F_{g_1}(a, b) \vee F_{g_2}(a, b)$

$$t_g := (u_1 G u_2) G (u_1 G u_2) G (\boldsymbol{t_1} G \boldsymbol{t_2}) G (u_1 G u_2) G (u_1 G u_2)$$

$$p_g := (u_1 G u_2 G)^+ (\boldsymbol{p_1} G q_2) G (q_1 G \boldsymbol{p_2}) (u_1 G u_2 G)^+$$

$$u_g := (u_1 G u_2) G (u_1 G u_2) G (\boldsymbol{u_1} G \boldsymbol{u_2}) G (u_1 G u_2) G (u_1 G u_2)$$

$$q_g := (u_1 G u_2) G (u_1 G u_2) G (\boldsymbol{q_1} G \boldsymbol{q_2}) G (u_1 G u_2) G (u_1 G u_2)$$

Lemma (Size bound for the final encoding)

$|u_r|, |t_r|, |p_r|, |q_r| \in \mathcal{O}(5^{d(F)} s \log s)$, *with* r *as root of* F.

Proof.

$|p_g| \in \mathcal{O}(|u_g|) = \mathcal{O}(|t_g|) = \mathcal{O}(|q_g|)$.
By definition: $|u_g| \leq 5|u_1| + 5|u_2| + \mathcal{O}(\log s)$
Inductively over the $d(F_g)$ levels of F_g: $|u_g| \leq \mathcal{O}(5^{d(F_g)} s \log s)$. □

Outer OR

Definition (Formula-Pair)

Input: A *monotone* De Morgan formula F, where each input is used exactly once, $A, B \subseteq \{0,1\}^\ell$ of size n and m.
Task: Check whether $\exists a \in A, b \in B : F(a,b) = \text{true}$.

$$t := \bigodot_{a \in A} \left(\gamma\gamma u_r \gamma u_r \gamma u_r \gamma t(a) \gamma u_r \gamma u_r \gamma u_r \gamma u_r\right)$$

$$p := \gamma u_r \gamma u_r \gamma u_r \gamma u_r \bigodot_{b \in B} \left(\gamma^+ (u_r\gamma)^+ u_r \gamma^+ q_r \gamma p(b) \gamma (u_r \gamma)^+ q_r\right) \gamma u_r \gamma u_r \gamma u_r \gamma u_r$$

Lemma (Correctness)

$(\exists a \in A, b \in B : F(a,b) = true) \iff t \in \mathcal{M}(p)$

The Lower Bound

- Text length and pattern size is $\mathcal{O}(n5^d s \log s) \subseteq \mathcal{O}(n5^s s \log s)$.
- We can reduce the depth of a formula by increasing its size:
  $d \to d' = \mathcal{O}(\log s)$ and $s \to s' = \mathcal{O}(s^2)$. (e.g. Bonet and Buss 1994)
- We obtain text length and pattern size of $\mathcal{O}(ns^{15})$.
- Assume a $\mathcal{O}(NM/\log^{92} N)$ algorithm for $\circ{+}\circ$-pattern matching:

$$\mathcal{O}\left(\frac{ns^{15} \cdot ns^{15}}{\log^{92}(ns^{15})}\right) \subseteq \mathcal{O}\left(\frac{n^2 s^{30}}{\log^{92} n}\right)$$

FORMULA-PAIR HYPOTHESIS (FPH)

For all $k \geq 1$: For a monotone De Morgan formula F of size s and sets $A, B \subseteq \{0,1\}^{s/2}$ of n half-assignments each, FORMULA-PAIR cannot be solved in time $\mathcal{O}(n^2 s^k / \log^{3k+2} n)$, in the Word-RAM model.

Reducing Pattern Matching to Membership

We can reduce pattern matching to membership for the pattern types for which we showed improved lower bounds.

Example for patterns of type $\circ{+}|$:

- $t' := \sigma t \sigma$ where $\sigma \in \Sigma$
- $p' := \Sigma^+ p \Sigma^+ = (\sigma_1 \mid \sigma_2 \mid \ldots \mid \sigma_{\mathfrak{s}})^+ p (\sigma_1 \mid \sigma_2 \mid \ldots \mid \sigma_{\mathfrak{s}})^+$

$t \in \mathcal{M}(p) \implies t' \in \mathcal{L}(p')$

The first Σ^+ matches initial σ and not matched prefix of t.
Analogous for the second Σ^+.

$t' \in \mathcal{L}(p') \implies t \in \mathcal{M}(p)$

The two Σ^+ match at least the initial and final σ.
Thus, p has to match some substring of t (possibly the empty string).

Upper Bounds

The Polynomial Method

- Originally used for circuit lower bounds (Razborov 1987 and Smolensky 1987)
- Method to transform boolean circuits into polynomials
- Adopted by Williams in 2014 for improved algorithm for APSP
- Yields super-poly-logarithmic runtime improvements
- Idea: Solve the task for many small sub-problems in parallel

Orthogonal Vectors (OV)

Input: Sets $U, V \subseteq \{0,1\}^d$ of n vectors each.
Question: Are there $u \in U, v \in V$ such that $\langle u, v \rangle = 0$?

Lemma (Chan and Williams 2016)

For $d = 2^{\Theta(\sqrt{\log n})}$ OV can be solved in time $\mathcal{O}(n^2/2^{\Theta(\sqrt{\log n})})$ deterministically.

Fast Algorithm

Focus on patterns with type $|\circ|$: $[(a \mid b)(b \mid c)d] \mid [ab(a \mid c \mid d)] \mid [bd]$

Main observation: Patterns can be split into independent sub-patterns!

(a) Define threshold $f \in 2^{\Theta(\sqrt{\log n})}$

(b) Split p into *large* (matching $> f$ symbols) and *small* sub-patterns

(c) Solve each of the $\leq k = m/f$ large sub-patterns of type $\circ|$ with the near-linear time algorithm:

$$\mathcal{O}\left(\sum_{i=1}^{k} n \log^2 m_i + m_i\right) \subseteq \mathcal{O}\left(\frac{m}{f} n \log^2 n + m\right) = \mathcal{O}\left(\frac{nm}{2^{\Theta\left(\sqrt{\log n}\right)}}\right)$$

(d) Reduce small sub-patterns to OV with dimension $d = 2^{\Theta(\sqrt{\log n})}$:

$$\mathcal{O}\left(\frac{nm}{2^{\Theta\left(\sqrt{\log n}\right)}}\right)$$

Small Sub-Patterns

Assume all sub-patterns match exactly f symbols (else pad with Σs).

Check whether there is a sub-pattern q and an offset $i \in [n]$ such that:

$$\begin{array}{cccccccc} t_1 t_2 \ldots\ldots & t_i & t_{i+1} & \ldots & t_{i+f-1} & \ldots\ldots t_{n-1} t_n \\ & | & | & & | & \\ & q_1 & q_2 & \ldots & q_f & \end{array}$$

- Use all length f substrings of t as one set of vectors
- Use sub-patterns as the other set of vectors
- Encode orthogonality using characteristic vector for Σ:

$$\begin{array}{l} t = a \\ p = a \mid b \end{array} \longrightarrow \begin{array}{c} (1,0,0) \\ (1,1,0) \end{array} \longrightarrow \begin{array}{c} (1,0,0) \\ (0,0,1) \end{array}$$

$$\Sigma = \{a, b, c\}$$

- n text-vectors and $\leq m$ pattern-vectors
- dimension $d = f \cdot |\Sigma| \in 2^{\Theta(\sqrt{\log n})}$ if $|\Sigma| = \mathcal{O}(1)$

Conclusion

Conclusion

Before: Upper bound: $\bar{\mathcal{O}}(nm/\log^{3/2} n)$
Lower bound: $\Omega((nm)^{1-\epsilon})$, assuming SETH.

Now:
Improved matching upper and lower bounds for homogeneous pattern types up to a constant number of log-factors. $\rightarrow$ A tight dichotomy.

<table>
<tr><th>Assume FPH and SETH</th><th>$\circ{+}|$, $\circ|{+}$, $\circ{+}\circ$, $\circ|\circ$, $\circ\star$</th><th>$|\circ|$, $|\circ{+}$</th><th>$|{+}|\circ$</th><th>${+}|\circ|$, ${+}|\circ{+}$</th></tr>
<tr><td>Pattern matching</td><td rowspan="2">$\Theta\left(\frac{nm}{\text{poly}\log n}\right)$</td><td>$\frac{nm}{2^{\Theta\left(\sqrt{\log n}\right)}}$</td><td>$\Theta(n+m)$</td><td>same as $|\circ|$, $|\circ{+}$</td></tr>
<tr><td>Membership</td><td>$\Theta(n+m)$</td><td>$\Theta\left(\frac{nm}{\text{poly}\log n}\right)$</td><td>$\frac{nm}{2^{\Theta\left(\sqrt{\log n}\right)}}$</td></tr>
</table>

Questions?