Succinctness of Regular Expressions with Interleaving, Intersection and Counting

Wouter Gelade

Hasselt University and transnational University of Limburg

August 25, 2008
Introduction

Regular Expressions

Regular expressions are used in many applications: Text processors, programming languages, XML schema languages, . . .

Motivating Question

What is the consequence of adding these operators?
Introduction

Regular Expressions

Regular expressions are used in many applications: Text processors, programming languages, XML schema languages, . . .

Operators?

- **Standard** regular expressions use disjunction (+), concatenation (·) and star (⋆).
- Most applications use additional operators: counting, interleaving, intersection, . . .
Regular Expressions

Regular expressions are used in many applications: Text processors, programming languages, XML schema languages, ... 

Operators?

- **Standard** regular expressions use disjunction (+), concatenation (·) and star (*).
- Most applications use additional operators: counting, interleaving, intersection, ...

Motivating Question

What is the **consequence** of adding these operators?
Counting Operator: Definition

- If \( r \) is a \( \text{RE(\#)} \), then \( r^{[i,j]} \), with \( i \leq j \) \( (i, j \in \mathbb{N}) \), also is a \( \text{RE(\#)} \).
- Example: \( a^{[3..5]} b + (cd)^{[0,10]} \)
- Used for instance in XML Schema, egrep and Perl patterns.
Counting Operator: Definition
- If $r$ is a $\text{RE}(#)\,$, then $r_{[i,j]}$, with $i \leq j \ (i,j \in \mathbb{N})$, also is a $\text{RE}(\#)$.
- Example: $a_{[3..5]}b + (cd)_{[0,10]}$
- Used for instance in XML Schema, egrep and Perl patterns.

Intersection Operator: Definition
- If $r, s$ are $\text{RE}(\cap)$ then $r \cap s$ also is a $\text{RE}(\cap)$.
- $L(r \cap s) = L(r) \cap L(s)$.
- Well studied extension of the regular expressions (also referred to as semi-extended regular expressions)
Interleaving Operator: Definition

- For words $w$, $u$, $v$, and symbols $a$, $b$:
  - $w \& \varepsilon = \varepsilon \& w = w$, and
  - $au \& bv = (a(u \& bv)) \cup (b(au \& v))$
- Allows the words of its operands to be *interleaved*.
- Example: $r = ab \& CD$
  - $abCD$, $CDab$, $aCbD \in L(r)$, $baCD \notin L(r)$
  - $L(r \& s) = \{ w \mid u \in L(r), v \in L(s), w \in L(u \& v) \}$
- Used in XML schema language Relax NG.
Outline

1. Operators
2. Questions
3. NFAs
4. DFAs
5. Regular Expressions
Succinctness w.r.t. Finite Automata

- Problem concerning standard regular expressions \(\Rightarrow\) translate to (deterministic or non-deterministic) automaton.
- Still feasible for additional operators? That is, what do we lose by adding operators?
- **Question**: what is the complexity of translating extended regular expressions to NFAs and DFAs?
Questions

Succinctness w.r.t. Finite Automata

- Problem concerning standard regular expressions ⇒ translate to (deterministic or non-deterministic) automaton.
- Still feasible for additional operators? That is, what do we lose by adding operators?
- **Question**: what is the complexity of translating extended regular expressions to NFAs and DFAs?
- Additional motivation: Negative results justify dedicated techniques.
Succinctness w.r.t. Regular Expressions

- How much easier/shorter can we define certain languages, i.e. what do we gain?
- **Question**: What is the complexity of translating extended regular expressions to standard regular expressions?
Questions

Succinctness w.r.t. Regular Expressions

- How much easier/shorter can we define certain languages, i.e. what do we gain?

- **Question**: What is the complexity of translating extended regular expressions to standard regular expressions?

- Additional Motivation: Establish complexity of translations among formalisms which use different operators. For instance, XML Schema vs. Relax NG.
Complement Operator?

Theorem [Stockmeyer, Meyer 1973]

RE(\(\neg\)) are non-elementary more succinct than standard REs and finite automata.
Outline

1. Operators
2. Questions
3. NFAs
4. DFAs
5. Regular Expressions
Proposition

Let $r$ be a $\text{RE}(\&, \cap, \#)$. An NFA $A$, with $L(r) = L(A)$, can be constructed in time $2^{O(|r|)}$. 
Proposition

Let \( r \) be a \( \text{RE}(\&, \cap, \#) \). An NFA \( A \), with \( L(r) = L(A) \), can be constructed in time \( 2^{O(|r|)} \).

Proof

- Construct \( A \) by induction on structure of \( r \).
- For instance, \( r = r_1 \cap r_2 \) and \( A_1, A_2 \) NFAs for \( r_1, r_2 \).
- \( \Rightarrow \) \( A \) is product of \( A_1 \) and \( A_2 \).
- Each step at most quadratic \( \Rightarrow \) exponential construction.
NFAs: Lower Bounds

**Proposition**
For any $n \in \mathbb{N}$, there exist an $\text{RE}(\cap) \ r$ of size $O(n)$, such that any NFA accepting $L(r)$ contains at least $2^n$ states.

**Other Operators**
**Proposition**

Let \( r \) be a \( \text{RE}(\&, \cap, \#) \). An NFA \( A \), with \( L(r) = L(A) \), can be constructed in time \( 2^{O(|r|)} \).

**Corollary**

Let \( r \) be a \( \text{RE}(\&, \cap, \#) \). A DFA \( A \), with \( L(r) = L(A) \), can be constructed in time \( 2^{2^{O(|r|)}} \).
**Proposition**

Let \( r \) be a \( \text{RE}(\&, \cap, \#) \). An NFA \( A \), with \( L(r) = L(A) \), can be constructed in time \( 2^{O(|r|)} \).

**Corollary**

Let \( r \) be a \( \text{RE}(\&, \cap, \#) \). A DFA \( A \), with \( L(r) = L(A) \), can be constructed in time \( 2^{2^{O(|r|)}} \).

**Proposition**

For any \( n \in \mathbb{N} \), there exist an \( \text{RE}(\#) \) \( r \) of size \( \mathcal{O}(n) \), such that any DFA accepting \( L(r) \) contains at least \( 2^{2^n} \) states.
DFAs: Lower Bound Intersection

Theorem

For any \( n \in \mathbb{N} \), there exist an \( \text{RE}(\cap)\ r \) of size \( \mathcal{O}(n) \), such that any DFA accepting \( L(r) \) contains at least \( 2^{2^n} \) states.

Proof

Let \( \Sigma = \{a, b\} \) and \( L_n = \{ww \mid |w| = 2^n\} \) DFA accepting \( L_n \) contains at least \( 2^{2^n} \) states.

\( \Rightarrow \) construct \( \text{RE}(\cap) \) of size \( \mathcal{O}(n) \) accepting \( L_n \).
For any $n \in \mathbb{N}$, there exist an RE(∩) $r$ of size $O(n)$, such that any DFA accepting $L(r)$ contains at least $2^{2^n}$ states.

Proof

- Let $\Sigma = \{a, b\}$ and $L_n = \{ww \mid |w| = 2^n\}$
- DFA accepting $L_n$ contains at least $2^{2^n}$ states.
- $\Rightarrow$ construct RE(∩) of size $O(n)$ accepting $L_n$. 
A marked binary number is a binary number in which the rightmost 1 and all following 0’s are marked.

\[
\begin{align*}
\text{enc}(0) &= \bar{0}\bar{0} \\
\text{enc}(1) &= 0\bar{1} \\
\text{enc}(2) &= \bar{1}\bar{0} \\
\text{enc}(3) &= 1\bar{1}
\end{align*}
\]

Property: Difference between two subsequent numbers are exactly marked bits of second number.
A marked binary number is a binary number in which the rightmost 1 and all following 0's are marked.

\[
\begin{align*}
\text{enc}(0) &= \overline{00} \\
\text{enc}(2) &= \overline{10} \\
\text{enc}(1) &= 0\overline{1} \\
\text{enc}(3) &= 1\overline{1}
\end{align*}
\]

Property: Difference between two subsequent numbers are exactly marked bits of second number

Proof (continued)

For \( w = a_0a_1 \ldots a_{2^n-1} \), let

\[
\text{enc}(w) = \text{enc}^R(0)a_0\text{enc}(0)\cdots\text{enc}^R(2^n-1)a_{2^n-1}\text{enc}(2^n-1)
\]

Let \( L'_n = \{ \#\text{enc}(w)\#\text{enc}(w) \mid |w| = 2^n \} \)

Construct \( \text{RE}(\cap) \) of size \( O(n) \) defining (complement of) \( L'_n \)
Proof (continued)

For $w = a_0a_1 \ldots a_{2^n-1}$, let

$$
\text{enc}(w) = \text{enc}^R(0)a_0\text{enc}(0)\cdot \cdot \cdot \text{enc}^R(2^n-1)a_{2^n-1}\text{enc}(2^n-1)
$$

Let $L'_n = \{\#\text{enc}(w)\#\text{enc}(w) \mid |w| = 2^n\}$

Construct $\text{RE}(\cap)$ of size $O(n)$ defining (complement of) $L'_n$

**RE$(\cap)$**

Check whether

- string is of proper format and numbers properly marked: easy.
- difference between subsequent numbers is 1: use property.
- elements at “exponential distance” are equal: use fact that they are surrounded by same marked number (intersection operator).
For any $n \in \mathbb{N}$, there exist an $\text{RE}(\&)$ $r$ of size $O(n^2)$, such that any DFA accepting $L(r)$ contains at least $2^{2n}$ states.
**Theorem**

For any \( n \in \mathbb{N} \), there exist an \( \text{RE}(\&) \) \( r \) of size \( \mathcal{O}(n^2) \), such that any DFA accepting \( L(r) \) contains at least \( 2^{2^n} \) states.

**Proof idea**

- **Simulate** intersection operator using interleaving operator [Mayer, Stockmeyer 1994.]
- Use succinctness of \( \text{RE}(\cap) \) obtained before.
Trick Mayer and Stockmeyer

- Let \( r \) be a (standard) regular expression and \( c \notin \Sigma \).
- Let \( r^c \) be obtained from \( r \) by replacing every symbol \( a \) by \( ac \).
- Example: \( r = (ab)^* \Rightarrow r^c = (acbc)^* \)
- \( a_1 \cdots a_n \in L(r) \iff a_1 c \cdots a_n c \in L(r^c) \)
Trick Mayer and Stockmeyer

Let $r$ be a (standard) regular expression and $c \notin \Sigma$.
Let $r^c$ be obtained from $r$ by replacing every symbol $a$ by $ac$.
Example: $r = (ab)^* \Rightarrow r^c = (acbc)^*$

$\Rightarrow a_1 \cdots a_n \in L(r) \iff a_1 c \cdots a_n c \in L(r^c)$

Simulate Intersection

Let $r = r_1 \cap r_2$. Define $r^c = r_1^c \& r_2^c$

$\Rightarrow a_1 \cdots a_n \in L(r) \iff a_1 c \cdots a_n c \in L(r_1^c) \cap L(r_2^c) \iff \quad a_1 a_1 cc \cdots a_n a_n cc \in L(r^c)$
Lemma [Mayer, Stockmeyer 1994]

Let $r$ be an $\text{RE}(\cap)$ containing $k \cap$-operators. Then, there exists an $\text{RE}(\&)$ $r^c$ of size at most $|r|^2$ such that $a_1 \cdots a_n \in L(r)$ iff $a_1^k c^k \cdots a_n^k c^k \in L(r^c)$. 

Theorem

For any $n \in \mathbb{N}$, there exist an $\text{RE}(\cap)$ $r$ of size $O(n)$, such that any DFA accepting $L(r)$ contains at least $2^{2n}$ states.

Corollary

For any $n \in \mathbb{N}$, there exist an $\text{RE}(\&)$ $r$ of size $O(n^2)$, such that any DFA accepting $L(r)$ contains at least $2^{2n}$ states.
Lemma [Mayer, Stockmeyer 1994]

Let $r$ be an $\text{RE}(\cap)$ containing $k \cap$-operators. Then, there exists an $\text{RE}(\&)$ $r^c$ of size at most $|r|^2$ such that $a_1 \ldots a_n \in L(r)$ iff $a_1^k c^k \cdots a_n^k c^k \in L(r^c)$.

Theorem

For any $n \in \mathbb{N}$, there exist an $\text{RE}(\cap) r$ of size $O(n)$, such that any DFA accepting $L(r)$ contains at least $2^{2^n}$ states.

Corollary

For any $n \in \mathbb{N}$, there exist an $\text{RE}(\&) r$ of size $O(n^2)$, such that any DFA accepting $L(r)$ contains at least $2^{2^n}$ states.
Outline

1. Operators
2. Questions
3. NFAs
4. DFAs
5. Regular Expressions
Counting [Kilpelainen, Tuhkanen]

Let $r$ be a $\text{RE}(\#)$, an equivalent standard expression can be constructed in time $O(2^n)$.

This exponential blow-up can not be avoided.
Counting [Kilpelainen, Tuhkanen]

- Let \( r \) be a \( \text{RE}(\#) \), an equivalent standard expression can be constructed in time \( O(2^n) \).
- This exponential blow-up cannot be avoided.

Proposition

Let \( r \) be a \( \text{RE}(\&, \cap, \#) \). An NFA \( A \), with \( L(r) = L(A) \), can be constructed in time \( 2^{O(|r|)} \).

Corollary

Let \( r \) be a \( \text{RE}(\&, \cap, \#) \). An equivalent (standard) RE, can be constructed in time \( 2^{2^{O(|r|)}} \).
Theorem

For any $n \in \mathbb{N}$, there exist an $\text{RE}(\cap) r$ of size $\mathcal{O}(n^2)$, such that any (standard) RE accepting $L(r)$ is of size $2^{2^{\Omega(n)}}$. 
Theorem

For any $n \in \mathbb{N}$, there exist an $\text{RE}(\cap) r$ of size $O(n^2)$, such that any (standard) RE accepting $L(r)$ is of size $2^{2\Omega(n)}$.

Difficulty

Show that certain languages cannot be defined by short (standard) regular expressions.
Definition

- The star height of a regular expression $r$, denoted $\text{sh}(r)$, is the maximal number of nested stars in $r$.
- $\text{sh}((a^*b)^* + c^*) = 2$, $\text{sh}(a^{***}) = 3$
- The star height of a regular language $L$, denoted $\text{sh}(L)$, is the minimal star height among all regular expressions defining $L$.
- $\text{sh}(L(a^{***})) = \text{sh}(a^*) = 1$

Lemma [Gruber, Holzer ’08]

Let $L$ be a regular language. Every regular expression defining $L$ must be of size at least $2^{1/3(\text{sh}(L)-1)} - 1$. 
Lemma [Gruber, Holzer ’08]

Let $L$ be a regular language. Every regular expression defining $L$ must be of size at least $2^{1/3(\text{sh}(L)-1)} - 1$.

Determining star height

... is very hard, in general.
Lemma [Gruber, Holzer '08]

Let $L$ be a regular language. Every regular expression defining $L$ must be of size at least $2^{1/3(\text{sh}(L) - 1)} - 1$.

Determining star height

- ... is very hard, in general.
- A language is bideterministic if the inverse of its minimal DFA is again deterministic.
- Star height of bideterministic language can be determined by looking at its minimal DFA.
**Definition**

For every $n \in \mathbb{N}$, let $K_n$ be defined by the complete DFA on $n$ states \{0, \ldots, n – 1\} with

- state 0 the initial state, and state $n – 1$ the final state; and
- a different label on every edge. ($\Sigma_n = \{a_{i,j} | 0 \leq i, j < n\}$)

**Example: $K_3$**

![Diagram of $K_3$ DFA]
A Language by Ehrenfeucht and Zeiger

Definition
For every \( n \in \mathbb{N} \), let \( K_n \) be defined by the complete DFA on \( n \) states \( \{0, \ldots, n - 1\} \) with
- state 0 the initial state, and state \( n - 1 \) the final state; and
- a different label on every edge. (\( \Sigma_n = \{a_{i,j} \mid 0 \leq i, j < n\} \))

Example: \( a_{0,0} a_{0,2} a_{2,2} \in K_3 \)
Properties $K_n$

- $K_n$ is bideterministic and $\text{sh}(K_n) = n$
- $\Rightarrow$ Any regular expression defining $K_n$ is of exponential size (proved already in a different manner by Ehrenfeucht and Zeiger)
- Can not be described succinctly by $\text{RE}(\cap)$. 
An encoding of $K_n$

For every $a_{i,j} \in \Sigma_n$ define

$$\rho_n(a_{i,j}) = \#\text{enc}(j)\triangle \text{enc}(i) \triangle \text{enc}(i+1) \triangle \cdots \triangle \text{enc}(n-1) \triangle.$$ 

Extend $\rho_n$ to strings as $\rho_n(a_{i_0,i_1} \cdots a_{i_{k-1},i_k}) = \rho_n(a_{i_0,i_1}) \cdots \rho_n(a_{i_{k-1},i_k}).$

The Language $L_n$: Definition

$L_n = \{\rho_n(w) \mid w \in K_n\}.$
Properties $L_n$

- $L_n$ is bideterministic and $\text{sh}(L_n) = n$
- $\Rightarrow$ Any regular expression defining $L_n$ is of exponential size (proved already in a different manner by Ehrenfeucht and Zeiger)
- $\Rightarrow$ Any regular expression defining $L_{2^n}$ is of double exponential size.
- Can construct $\text{RE}(\cap)$ of size $O(n^2)$ defining $L_{2^n}$.
Properties $L_n$

- $L_n$ is bideterministic and $\text{sh}(L_n) = n$
- $\Rightarrow$ Any regular expression defining $L_n$ is of exponential size (proved already in a different manner by Ehrenfeucht and Zeiger)
- $\Rightarrow$ Any regular expression defining $L_{2^n}$ is of double exponential size.
- Can construct $\text{RE}(\cap)$ of size $O(n^2)$ defining $L_{2^n}$.

Remark

Similar result already obtained [G., Neven 2008]. Preparation for lower bound interleaving.
Theorem

For any $n \in \mathbb{N}$, there exist an $\text{RE}(\&)$ $r$ of size $\mathcal{O}(n^2)$, such that any RE accepting $L(r)$ is of size $2^{2\Omega(n)}$.
Regular Expressions: Lower Bound Interleaving

**Theorem**

For any \( n \in \mathbb{N} \), there exist an \( \text{RE}(\& \ r) \) of size \( \mathcal{O}(n^2) \), such that any \( \text{RE} \) accepting \( L(r) \) is of size \( 2^{2^{\Omega(n)}} \).

**Theorem**

For any \( n \in \mathbb{N} \), there exist an \( \text{RE}(\cap \ r_n) \) of size \( \mathcal{O}(n^2) \), such that any \( \text{RE} \) accepting \( L(r_n) \) is of size \( 2^{2^{\Omega(n)}} \).

**Proof idea**

- For any \( n \in \mathbb{N} \) take \( \text{RE}(\& \ r_n^c) \), the simulation of the \( \text{RE}(\cap \ r_n) \).
- Show that star height \( L(r_n^c) \) is still large
  - Use bideterminism property
  - Exploit structural properties of \( L(r_n^c) \)
Remark
Similar result independently obtained by Gruber and Holzer.
Conclusion

Overview

Translation of extended regular expressions to
- NFAs is exponential.
- DFAs is double exponential.
- Regular expressions is double exponential, except for RE(\#).