Cuts in Regular Expressions

Frank Drewes
Umeå University (Sweden)

Joint work with M. Berglund, H. Björklund, B.v.d. Merwe, and B. Watson

FASTAR/Espresso Workshop 2013

Perhaps some remember last year's talk about cuts by Martin Berglund?

- Intuitively, $E!E'$ is concatenation, but with a greedy E.
- E grabs as much as it can; the rest must match E.
- This is motivated by *REGEXP* packages with concepts such as possessive quantifiers and pruning, that are usually imprecisely defined.

After FASTAR/Espresso 2012, we had a closer look and finally wrote a paper (for DLT 2013).

This talk is about the results of that paper.

Later on, in his talk Martin will take the visionary role again.

$$L!L' = \{uv \mid u \in L, v \in L',$$
$$uv' \notin L \text{ for all nomempty prefixes } v' \text{ of } v\}.$$

Examples of cut expressions:

$ab^* ! b \equiv \emptyset$

$(a^* \mid b^*) ! (ac \mid bc) \equiv a^+ bc \mid b^+ ac$

$((ab)^* ! a) ! b \equiv (ab)^* ab$ but $(ab)^* ! (a ! b) \equiv (ab)^* ! ab \equiv \emptyset$

Note (and remember for later):

If $\varepsilon \notin \mathscr{L}(E')$ then $\mathscr{L}(E) \cap \mathscr{L}(E!E') = \emptyset$. E.g., $\mathscr{L}(E) \cap \mathscr{L}(E!\Sigma) = \emptyset$.

Can we use $*$ to iterate the cut?

"Take the longest prefix matching E, then iterate."

An attempt: What does $(E!\varepsilon)^*$ yield? – Just E^*.

$\Rightarrow$ if we want an iterated cut, we must define it explicitly:

$L^{!*}$ is the smallest language s.t. $\{\varepsilon\} \cup (L!(L^{!*})) \subseteq L^{!*}$

Cut expressions consist of ordinary regular operators, $!$, and $^{!*}$.

In the following, we will mostly be interested in $!$.

We do not gain expressive power...

Theorem (closedness of REG under $!$ and $^{!*}$)

If L and L' are regular, then so are $L!L'$ and $L^{!*}$.
Thus, $\mathscr{L}(E)$ is regular for every cut expression E.

Open question: What is the maximum size of a minimal DFA accepting $\mathscr{L}(E)$ for a cut expression E of size n? Does it matter whether we use the iterated cut?

The upper bound we get is non-elementary. ☹

We conjecture that something much better is possible.

We do have a lower bound...

Let $\Sigma = \{0, 1\}$ and

$$E_n = ([\Sigma^* 0 \Sigma^{n-1} 1 \Sigma^*] \mid [\Sigma^* 1 \Sigma^{n-1} 0 \Sigma^*] \mid \varepsilon) \; ! \; [\Sigma^{2n}]$$

$\Rightarrow$ $\mathscr{L}(E_n)$ consists of strings $[x][y]$ and $[vv]$ where $v \in \Sigma^n$.

$\Rightarrow$ every NFA accepting $\mathscr{L}(E_n)$ has at least $2^{\Omega(n)}$ states

Theorem (succinctness of cut expressions)

There exist cut expressions $E = E_1 ! E_2$ of size n, where E_1, E_2 are ordinary regular expressions, such that the minimal NFAs accepting $\mathscr{L}(E)$ are of size $2^{\Omega(n)}$.

How efficiently can we solve the uniform membership problem?

$Match(E,i,j)$ matches E to substring $a_i \cdots a_{j-1}$ of $a_1 \cdots a_n$

---

if $E = \emptyset$ **then return** *false*

$\vdots$

else if $E = E_1 \,!\, E_2$ **then**

  for $k = j-i, \ldots, 0$ **do**

    if $Match(E_1, i, i+k)$ **then return** $Match(E_2, i+k, j)$

  return *false*

else if $E = E_1^{!*}$ **then**

  for $k = j-i, \ldots, 1$ **do**

    if $Match(E_1, i, i+k)$ **then return** $Match(E, i+k, j)$

  return $i = j$

Theorem (complexity of the uniform membership problem)

The uniform membership problem for cut expressions is solvable in time $O(mn^3)$, where m is the size of the expression and n is the length of the input string.

Finally, emptiness testing...

Recall Regular Expression Universality:

Deciding $\mathscr{L}(E) = \Sigma^*$ for regular expressions E is PSPACE-hard.

Consider a regular expression E with $\varepsilon \in \mathscr{L}(E)$.

1. The shortest $ua \notin \mathscr{L}(E)$ is in $\mathscr{L}(E!\Sigma)$ (if it exists) as $u \in \mathscr{L}(E)$.
2. $\mathscr{L}(E) \cap \mathscr{L}(E!\Sigma) = \emptyset$ (by the earlier remark, as $\varepsilon \notin \Sigma$).

$\Rightarrow \mathscr{L}(E!\Sigma) = \emptyset \iff \mathscr{L}(E) = \Sigma^*$

Theorem (PSPACE-Hardness of Emptiness)

The emptiness problem for expressions of the form $E!\Sigma$, where E is regular, is PSPACE-hard.

Summary

1. Cut expressions define regular languages.
2. Equivalent NFAs are exponentially large even with only one '!'.
3. The only upper bound we know is non-elementary (huge gap!).
4. Uniform membership is in $O(mn^3)$. (Can we do better?)
5. Emptiness is PSPACE-hard even with only one '!'. (We do not know a reasonable upper bound for the general case.)

CUT!