Rewriting of Regular Path Queries: The first paper of the four Italians

Maurizio Lenzerini

SAPIENZA
UNIVERSITÀ DI ROMA

Based on the paper
"Rewriting of Regular Expressions and Regular Path Queries",
by D. Calvanese, G. De Giacomo, M. Lenzerini, M. Y. Vardi

VardiFest22: On the not so unusual effectiveness of Logic

July 31 – August 1, 2022

Italian last names of the form "_ardi"

- Bardi
- Cardi
- Dardi
- Gardi
- Lardi
- Nardi
- Pardi
- Sardi
- Tardi
- Zardi

"Vardi" is missing, but it really sounds like an Italian last name ...

Given:

- a query Q over a database with alphabet Σ
- k view definitions (where each Q_i is a query over Σ):

$$q_1 \doteq Q_1, \quad \dots \quad , q_k \doteq Q_k$$

can we re-express Q in terms of the views $q_1, \dots, q_k$?

Several applications, such as:

- Answer Q by relying only on the precomputed answers to the views?
- Process queries in LAV data integration system, where sources are characterized as views over the global schema Σ

Graph databases and regular path queries

A **graph database** is a labeled graph, where nodes represents objects and each edge represents the fact that a certain relation holds between two nodes.

A **regular path query** Q is specified through a regular language $L(Q)$ over the edge labels.

The **answer to** Q **over a database** DB is:
$\{(x, y) \mid \exists(x \stackrel{a_1}{\to} x_1 \cdots \stackrel{a_n}{\to} y)$ in DB s.t. $a_1 \cdots a_n$ is a word in $L(Q)\}$

Example: $(child^* \cdot friend) + (friend \cdot child^*)$

Rewriting of regular expressions

- $\Sigma_{\mathcal{E}} = \{e_1, \ldots, e_k\}$
- $re(e_i) = E_i$ r.e. over Σ

For a language ℓ over $\Sigma_{\mathcal{E}}$, we define

$$exp_{\Sigma}(\ell) = \bigcup_{e_1 \cdots e_n \in \ell} \{w_1 \cdots w_n \mid w_i \in L(re(e_i))\}$$

Example: $\Sigma_{\mathcal{E}} = \{e_1, e_2\}$
$re(e_1) = a + b, \quad re(e_2) = c{\cdot}d + f$
$\ell = (e_1{\cdot}e_2) + e_1$
$exp_{\Sigma}(\ell) = \{acd, af, bcd, bf, a, b\}$

Rewriting of regular expressions

Intuitively:

Given a r.e. E_0 over Σ, an alphabet $\Sigma_{\mathcal{E}} = \{e_1, \ldots, e_k\}$, and one r.e. $re(e_i)$ over Σ for each $e_i \in \Sigma_{\mathcal{E}}$, we aim at re-expressing E_0 by a combination of $re(e_1), \ldots, re(e_k)$.

Formally:

A formalism R for defining a language $L(R)$ over $\Sigma_{\mathcal{E}}$ is a **rewriting of** E_0 wrt $\mathcal{E}$ if

$$exp_{\Sigma}(L(R)) \subseteq L(E_0).$$

Computing the rewriting of a regular expression

Input: r.e. E_0 over Σ, $\Sigma_{\mathcal{E}} = \{e_1, \ldots, e_k\}$, set $\{re(e_1), \ldots, re(e_k)\}$ of r.e. over Σ

Output: automaton $R_{\mathcal{E},E_0}$ (maximal rewriting of E_0 wrt $\mathcal{E}$)

1. Construct a **deterministic** automaton $A_d = (\Sigma, S, s_0, \rho, F)$ accepting $L(E_0)$.
2. Let $A' = (\Sigma_{\mathcal{E}}, S, s_0, \rho', S - F)$, where

   $$s_j \in \rho'(s_i, e) \quad \text{iff} \quad \exists w \in L(re(e)) \text{ s.t. } s_j \in \rho^*(s_i, w).$$

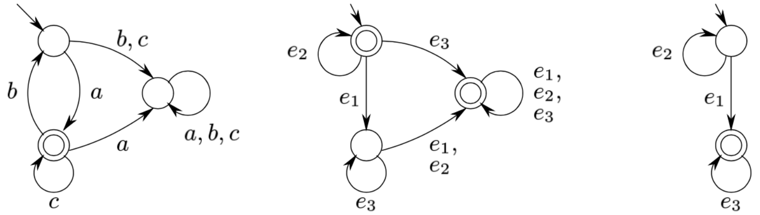
   A' **accepts** a $\Sigma_{\mathcal{E}}$-word $e_1 \cdots e_n$ iff there is a Σ-word in $exp_\Sigma(\{e_1 \cdots e_n\})$ **rejected** by A_d.
3. $R_{\mathcal{E},E_0}$ is $\overline{A'}$, i.e. the complement of A'.

Observation:

Any maximal rewriting of E_0 wrt $\mathcal{E}$ defines a **regular language**.

Rewriting of regular expressions (example)

$$E_0 = a\cdot(b\cdot a + c)^* \qquad \mathcal{E} = \{\underbrace{a}_{e_1}, \underbrace{a\cdot c^*\cdot b}_{e_2}, \underbrace{c}_{e_3}\}$$



A_d $\qquad$ A' $\qquad$ $\overline{A'} = R_{\mathcal{E},E_0}$

Complexity analysis

For a regular expression E_0 and a set $\mathcal{E}$ of regular expressions:

Generating the $\Sigma_{\mathcal{E}}$-maximal rewriting is in 2EXPTIME
(Two determinization steps.)

Checking existence of a nonempty rewriting is EXPSPACE-complete
(Hence the upper bound for generating the $\Sigma_{\mathcal{E}}$-maximal rewriting is optimal.)

Verifying the existence of an exact rewriting is 2EXPSPACE-complete

What came next

After this paper, Moshe visited regularly Rome, sometimes with Pam, often during the Christmas Holydays.

The group had such a great time, enjoying the city and working on several aspects of graph databases, including

- Answering queries using views (see Diego's talk)
- Two-way RPQs (containment, rewriting and answering)
- Conjunctive RPQs (containment, rewriting and answering)
- Two-way Conjunctive RPQs (containment, rewriting and answering)
- View-based containment
- ...

Moshe, I am looking forward to another ride in the traffic of Rome!