A SYMBOLIC CALCULUS OF REGULAR EXPRESSIONS

Mizoguchi, Yoshihiro
Department of Mathematics, Kyushu University

Otsuka, Hiroshi
Department of Mathematics, Kyushu University

Kawahara, Yasuo
Department of Mathematics, Kyushu University

https://doi.org/10.5109/13383

出版情報：Bulletin of informatics and cybernetics. 22 (3/4), pp.165-170, 1987-03. Research Association of Statistical Sciences

Bulletin of Informatics and Cybernetics Vol. 22, No. 3~4, 1987

A SYMBOLIC CALCULUS OF REGULAR EXPRESSIONS

By

Yoshihiro MIZOGUCHI*, Hiroshi OHTSUKA*
and Yasuo KAWAHARA*

Abstract

To implement Ginzburg's equality check procedure for regular expressions by using personal computers, we propose a new and more efficient axiom system consisting of an axiom and inference rules concerning a new relational symbol $\subset$ in addition to a part of Salomaa's axiom system.

1. Introduction

The researches of axiom systems for the regular expressions started in the 60's. Redco [4] showed that it is impossible to make a complete axiom system with finite axioms and only one inference rule **R1**. He made a complete system with countably many axioms and an inference rule **R1**. Salomaa [5] showed a complete and consistent axiom system by using two inference rules **R1** and **R2**. In his system every tautology has a constructive proof. But the construction of a proof is so complicated that it is not useful for practice. After a few years, Ginzburg [2] showed a simple mechanical procedure for checking equality of regular expressions by using derivatives. This method uses transition graphs so it is neither symbolic nor axiomatic. In the present paper, by using symbolic derivatives, we improve Ginzburg's procedure to a symbolic and axiomatic one. Moreover we examine which axiom in Salomaa's system is essential for showing the validity of the procedure. We choose some axioms which is useful for mechanical procedure, and we show that it is sufficient to show the termination of the checking procedure by using only those axioms.

In section 2, we recall Salomaa's system F_1 and introduce a new system F_*. In section 3, we show that in the system F_*, the set of all symbolic derivatives of a regular expression is *finite*. This result guarantees the termination of improved Ginzburg's equality check procedure. In section 4, we introduce a new relational symbol $\subset$, and we improve the system F_* to a more efficient system F_+. Since we do not use an axiom of associative laws in the system F_+, we can check equalities by a straightforward method.

We implemented the efficient procedure on a personal computer. So we can easily check the equality of two regular expressions. Moreover we get the transition graphs which accept the expressions together with the answer of the equality. We show a brief example in the Appendix.

* Department of Mathematics, Kyushu University 33 Fukuoka 812, Japan.

2. Preliminaries

DEFINITION 2.1. *Let Γ be an alphabet set. Assume $\phi \notin \Gamma$. We define regular expressions over Γ as follows:*

(1) *ϕ is a regular expression.*

(2) *An element x of Γ is a regular expression.*

(3) *If X and Y are regular expressions, then so are $(X+Y)$, $(Y \cdot Y)$ and X^*.*

(4) *Nothing else a regular expression unless its being so follows from a finite number of applications of* (1), (2) *and* (3).

In the rest of this paper, if no confusion occurs we omit parentheses and $\cdot$ from $(X \cdot Y)$. We use symbol e for ϕ^*; a, b, c, x, y, z for elements of Γ; A, B, C, X, Y, Z for regular expressions and u, v, w for strings of elements of Γ. If X and Y are the same regular expressions we denote $X \equiv Y$. We denote by $X[A:=B]$ a regular expression substituted A in X by B.

DEFINITION 2.2. *We define a property Eps of regular expressions as follows:*

(1) *ϕ is not Eps.*

(2) *An element x of Γ is not Eps.*

(3) *X^* is Eps.*

(4) *$X+Y$ is Eps iff X is Eps or Y is Eps.*

(5) *$X \cdot Y$ is Eps iff X is Eps and Y is Eps.*

We note that for any regular expression X, we can check whether X is Eps or not in a finite number of steps.

Now we recall Salomaa's axiom system.

Axioms:

- ***A0** $X=X$,
- ***A1** $X+(Y+Z)=(X+Y)+Z$,
- **A2** $X(YZ)=(XY)Z$,
- ***A3** $X+Y=Y+X$,
- **A4** $X(Y+Z)=XY+XZ$,
- **A5** $(X+Y)Z=XZ+YZ$,
- ***A6** $X+X=X$,
- ***A7** $Xe=X$,
- ***A8** $X\phi=\phi$,
- ***A9** $X+\phi=X$,
- **A10** $X^*=e+XX^*$,
- **A11** $X^*=(e+X)^*$,
- ***A12** $eX=X$,
- ***A13** $\phi X=\phi$,
- ***A14** $\phi+X=X$.

Inference rules:

- **R1** From the equations $A=B$ and $X=Y$ one may infer the equations $Y=X$ and $X[A:=B]=Y$.
- **R2** Assume that A is not Eps. Then from the equation $X=AX+B$ one may infer $X=A^*B$.

DEFINITION 2.3.

(1) *Salomaa's axiom system* F_1 *has* 12 *axioms* **A0-A11** *and two inference rules* **R1** *and* **R2**. *If an equation* $X=Y$ *is derivable within the system* F_1 *we denote* $\vdash_1 X=Y$.

(2) *An axiom system* F_* *has* 10 *axioms which are marked by* * *and an inference rule* **R1**. *If an equation* $X=Y$ *is derivable within the system* F_* *we denote* $\vdash X=Y$.

PROPOSITION 2.4. *If an equation* $X=Y$ *is derivable within* F_* *then it is derivable within* F_1. □

For any two regular expressions X and Y, we can get a mechanical procedure for checking whether an equation $X=Y$ is derivable or not. First we find a normal form X'(resp. Y') of X(resp. Y) by using axioms **A6-A9** and **A12-A14**. Then we check whether an equation $X'=Y'$ is derivable or not by using axioms **A0-A3**. The second procedure is not straitforward in the system F_*, but we solve the difficulty in section 4 by reforming the system.

3. Derivatives

In this section, we recall the symbolic derivatives of regular expressions [1]. We show that for any regular expression, the set of all derivatives in the system F_* is finite, by which we can find all derivatives of an expression in a finite number of steps. Moreover, we can implement the Ginzburg's equality check proceedure by using the symbolic derivatives. For two expressions, we first find all derivatives of them, and then check for each derivatives the property Eps. By comparing the properties Eps of corresponding derivatives with each other, we can check the equality of the expressions.

DEFINITION 3.1. *We define symbolic derivatives of regular expressions as follows*:

(1) $a/x := \begin{cases} e & (\text{if } a=x) \\ \phi & (\text{otherwise}), \end{cases}$

(2) $a/(X+Y) := (a/X)+(a/Y)$,

(3) $a/(X\cdot Y) := (a/X)Y+\bar{X}(a/Y)$,

(4) $a/X^* := (a/X)X^*$,

(5) $(ua)/X := a/(u/X))$,

where $\bar{X} := \begin{cases} e & (\text{if } X \text{ is Eps}) \\ \phi & (\text{otherwise}). \end{cases}$

LEMMA 3.2.

(1) $\bar{\phi}\equiv\phi$,

(2) $\vdash(\overline{X+Y})=\bar{X}+\bar{Y}$,

(3) $\vdash(\overline{X\cdot Y})=\bar{X}\cdot\bar{Y}$,

(4) $\vdash a\cdot\bar{X}=\bar{X}\cdot a$. □

PROPOSITION 3.3. *For any regular expression* X, *we have* $\vdash_1 X=\bar{X}+\sum_{a\in\Gamma}a(a/X)$.

PROOF. We omit a subscript $a\in\Gamma$ from $\sum_{a\in\Gamma}$. It is trivial for ϕ and $x\in\Gamma$, since $\vdash\phi=\phi+\sum\phi=\phi+\sum a(a/\phi)$ and $\vdash x=\phi+xe=\bar{x}+\sum a(a/x)$.

Assume $\vdash_1 X=\bar{X}+\sum a(a/X)$ and $\vdash_1 Y=\bar{Y}+\sum a(a/Y)$, then we obtain following equations.

$$\vdash_1 X+Y=\bar{X}+\sum a(a/X)+\bar{Y}+\sum a(a/Y)=\bar{X}+\bar{Y}+\sum a((a/X)+(a/Y))$$
$$=(\overline{X+Y})+\sum a(a/(X+Y)),$$

$$\vdash_1 XY=(\bar{X}+\sum a(a/X))(\bar{Y}+\sum a(a/Y))=\bar{X}\cdot\bar{Y}+\bar{X}\sum a)a/Y)+(\sum a(a/X))Y$$
$$=\bar{X}\cdot\bar{Y}+\sum a(a/X)Y+\sum a\bar{X}(a/Y)=(\overline{X\cdot Y})+\sum a(a/(XY)),$$
$$\vdash_1 X^*=(\bar{X}+\sum a(a/X))^*=(\sum a(a/X))^*=e+(\sum a(a/X))(\sum a(a/X))^*$$
$$=e+\sum a(a/X)X^*=\bar{X}^*+\sum a(a/X^*). \quad \square$$

THEOREM 3.4. *For any regular expression X, there exists a finite set $[X]=\{X_1, X_2, \cdots, X_n\}$ of regular expressions such that for any string w of Γ, there exists a regular expression X_i in $[X]$ and $\vdash X_i=(w/X)$. That is, the set of all symbolic derivatives of X is finite in F_*.*

PROOF. It is trivial for $[\phi]:=\{\phi\}$ and $[X]:=\{e, x, \phi\}$. Assume that $[X]=\{X_1, X_2, \cdots, X_n\}$ and $[Y]=\{Y_1, Y_2, \cdots, Y_m\}$. We define

$$[X+Y]:=\{X_i+Y_j \mid 1\leqq i\leqq n,\ 1\leqq j\leqq m\},$$
$$[XY]:=\{X_iY+Y_{j_1}+Y_{j_2}+\cdots+Y_{j_k} \mid 1\leqq i\leqq n,\ 1\leqq k\leqq m,\ 1\leqq j_1<j_2<\cdots<j_k\leqq m\}$$

and

$$[X^*]:=\{X_{i_1}X^*+X_{i_2}X^*+\cdots+X_{i_k}X^* \mid 1\leqq k\leqq n,\ 1\leqq i_1<i_2\cdots<i_k\leqq n\}.$$

Then for any string w of Γ, $\vdash w/(X+Y)=(w/X)+(w/Y)\in[X+Y]$. For any element $a\in\Gamma$, we obtain $\vdash a/(XY)=(a/X)Y+\bar{X}(a/Y)\in[XY]$,

$$a/(X_iY+Y_{j_1}+Y_{j_2}+\cdots+Y_{j_k})$$
$$=(a/X_i)Y+\bar{X}_i(a/Y)+(a/Y_{j_1})+(a/Y_{j_2})+\cdots+(a/Y_{j_k})\in[XY],$$
$$a/(X^*)=(a/X)X^*\in[X^*],$$

and

$$a/(X_{i_1}X^*+X_{i_2}X^*+\cdots+X_{i_k}X^*)$$
$$=(a/X_{i_1})X^*+\bar{X}_{i_1}(a/X)X^*+(a/X_{i_2})X^*+\cdots+(a/X_{i_k})X^*+\bar{X}_{i_k}(a/X)X^*\in[X^*]. \quad \square$$

By Theorem 3.4, our process of finding all derivatives of a regular expression in the system F_* terminates in a finite number of steps.

PROPOSITION 3.5. *If the system F_* does not contain axioms* **A1** *and* **A6** *then Theorem 3.4 is not valid.*

PROOF. We show a counter example. Let $A=(a^*b+a)^*$. $a/A=(a^*b+e)(a^*b+a)^*$, $aa/A=a^*b(a^*b+a)^*+((a^*b+e)(a^*b+a)^*)=B+(a/A)$, where $B=a^*b(a^*b+a)^*$. By $a/B=B$, we have $aaa/A=B+(aa/A)=B+(B+(a/A))$. Since one can not use axiom **A6** without axiom **A1**, a^n/A has $n-1$ copies of B. So the set of all symbolic derivatives of A is an infinite set. $\square$

4. The Axiom System F_+

From the view of an equality check procedure, axioms **A0**, **A1** and **A3** in the system F_* are not practical. So we modify those axioms and make new system F_+. To avoid the application of those axioms, we introduce a new relational symbol $\subset$ and some new inference rules.

DEFINITION 4.1. *The axiom system F_+ consists of axioms* **A7**, **A8**, **A9**, **A12**, **A13** *and* **A14**, *an inference rule* **R1**, *a new axiom* **A15** *and new inference rules* **R3**-**R6** *below.*

Axioms:

A15 $X \subset X$.

Inference rules:

R3 From $X \subset A$ and $Y \subset B$, one may infer $X^* \subset A^*$ and $(X \cdot Y) \subset (A \cdot B)$.

R4 From $X \subset A$ and $Y \subset A$, one may infer $(X+Y) \subset A$.

R5 From $X \subset A$ or $X \subset B$, one may infer $X \subset (A+B)$.

R6 From $X \subset Y$ and $Y \subset X$, one may infer $X=Y$.

We note that for arbitrary two regular expressions X and Y, we can show in a finite number of steps whether the equation $X=Y$ is derivable or not.

PROPOSITION 4.2. *If an equation $X=Y$ is derivable within F_*, then it is derivable within F_+.*

PROOF. The axioms **A0**, **A1**, **A3** and **A6** are infered from inference rules **R4**, **R5** and **R6**. □

The equality check procedure in the system F_+ is simple. So we can improve the Ginzburg's equality check procedure into a more efficient procedure.

References

[1] BRZOZOWSKI, J. A.: *Derivatives of regular expressions*, J. A. C. M., 11 (1964), 481–494.

[2] GINZBURG, A.: *A procedure for checking equality of regular expressions*, J. A. C. M., 14 (1967), 355–362.

[3] KOBAYASHI, K. and TAKAHASHI, M.: *Theory of automata*, Kyouritsu Shuppan, 1983. (Japanese)

[4] REDKO, V. N.: *On defining relations for the algebra of regular events*, Ukrain. Mat. Z., 16 (1964), 122–126. (Russian)

[5] SALOMAA, A.: *Two complete axiom systems for the algebra of regular events*, J. A. C. M., 13 (1966), 158–169.

Communicated by S. Arikawa
Received July 30, 1986
Revised September 12, 1986

Appendix

Followings are execution examples of the Ginzburg's equality check procedure with symbolic derivatives. For an expression X, $[w]$ expresses w/X. The symbol # indicates the Eps property of w/X. The return value means the equality of two expressions.

```
*(GINZ)
(EXP1)>(a+ab+ba)*
(EXP2)>(ba+a*ab)*a*
(SIGMA)>(a b)
[e]=#(a+ab+ba)*
    #(ba+a*ab)*a*
[a]=#(e+b)(a+ab+ba)*
    #(a*ab+b)(ba+a*ab)*a*+a*
```

[aa]=#[a]
#[a]
[ab]=#(a+ab+ba)*+a(a+ab+ba)*
#[e]
[aba]=#(e+b)(a+ab+ba)*+(a+ab+ba)*
#[a]
[$abaa$]=#[a]
#[a]
[$abab$]=#[ab]
#[e]
[abb]= [b]
[b]
[b]= a(a+ab+ba)*
a(ba+a*ab)*a*
[ba]=#[e]
#[e]
[bb]= %
%
[bba]= [bb]
[bbb]= [bb]
T