Monodic tree Kleene algebra

(Preliminary Version)

Toshinori Takai and Hitoshi Furusawa

産業技術総合研究所 システム検証研究センター

Monodic tree Kleene algebra

Toshinori Takai Hitoshi Furusawa

April 5, 2006

Abstract

We propose a quasi-equational sound axiomatization of regular tree languages, called monodic tree Kleene algebra. The algebra is weaker than Kleene algebra introduced by Kozen. We find a subclass of regular tree languages, for which monodic tree Kleene algebra is complete. While regular tree expressions may have two or more kinds of place holders, the subclass can be equipped with only one kind of them. Along the lines of the original proof by Kozen, we prove the completeness theorem based on determinization and minimization of tree automata represented by matrices on monodic tree Kleene algebra.

1 Introduction

A tree language is a set of first-order terms and a tree automaton is a natural extension of a finite automaton, in which the inputs are first-order terms[1]. The class of regular tree languages, which are recognized by tree automata, inherits some desirable properties including complexity of some decision problems and closeness under boolean operations. In the last year, the authors proposed essentially algebraic structure of a certain subclass of tree languages[2]. But the subclass and the class of regular tree languages are incomparable.

Although regular tree expressions and Kleene theorem for trees have been proposed[1, 3], they are rarely used in practice because the structure is too complicated. Regular tree expressions have two or more multiplication, Kleene stars and place holders. The goal of our study is to make clear of algebraic structure of regular tree expressions. In this paper, we propose a subclass of regular tree expressions, called *monodic regular tree expressions*, and also propose a complete axiomatization, called a *monodic tree Kleene algebra*, of the subclass of regular tree languages corresponding to monodic regular tree expressions. A monodic regular tree expression has only one kind of place holders, multiplications and Kleene stars. The subclass corresponds to tree automata in which only one kind of states occurs in the left-hand side of each transition rule.

A monodic tree Kleene algebra is a similar to a Kleene algebra by Kozen[4] and a Kleene algebra is always a monodic tree Kleene algebra. The essential difference is the lack of the right-distributivity of the multiplication over addition $+$. For example, let f be a binary function symbol, a and b be constants. Then

$f(\Box,\Box)$ and $a+b$ are regular tree expressions with place holder $\Box$ and so is $f(\Box,\Box)\cdot(a+b)$. The expression $f(\Box,\Box)\cdot(a+b)$ is interpreted as a set of terms obtained by replacing $\Box$ with a or b, i.e. $\{f(a,a), f(a,b), f(b,a), f(b,b)\}$. On the other hand, the interpretation of $f(\Box,\Box)\cdot a + f(\Box,\Box)\cdot b$ is $\{f(a,a), f(b,b)\}$. Another differences are shapes of right-unfold and right-induction laws. We compare the original laws with the new ones.

After giving preliminaries, we define monodic tree Kleene algebras and show some basic properties of them. In Section 4, a subclass of regular tree expressions and a subclass of tree automata are proposed and show the correspondence of them via matrices. Since the proof flow of the completeness theorem is the same as the original one by Kozen[4], this paper concentrates showing the lemmas used in the proof.

2 Preliminaries

We review some notions on the language theory on trees. After defining the syntax of regular tree expressions, we give some functions on tree languages. Using the functions, we give an interpretation function of regular tree expressions.

For a signature Σ, we denote the set of all first-order terms without variables constructed from Σ by T_Σ. A *tree language* is a set of terms. For a signature Σ and a set Γ of *substitution constants*, the set $\mathsf{RegExp}(\Sigma,\Gamma)$ of *regular tree expressions* is inductively defined as follows. A substitution constant is a constant not included in the signature.

1. The symbol $\mathbf{0}$ is a regular tree expression.
2. A term in $\mathsf{T}_{\Sigma\cup\Gamma}$ is a regular tree expression.
3. If e_1 and e_2 are regular tree expressions and $\Box$ is a substitution constant, so are e_1+e_2, $e_1 \cdot_\Box e_2$ and $e_1^{*\Box}$.

Let $\mathscr{P}(S)$ denote the power set of a set S. We define binary function $\circ_\Box$ and unary function ${}^{\star_\Box}$ on $\mathscr{P}(\mathsf{T}_{\Sigma\cup\Gamma})$ for any $\Box\in\Gamma$ and define *tree substitutions*. A tree substitution is given by a finite set of pairs of a substitution constant and a tree language. For a tree substitution $\theta = \{(\Box_1, L_1),\ldots,(\Box_n, L_n)\}$ and a term $t\in\mathsf{T}_{\Sigma\cup\Gamma}$, define $\theta(t)$ as follows.

1. If $t\in\Gamma$ and $t=\Box_i$, then $\theta(t)=L_i$.
2. If $t\in\Gamma\setminus\{\Box_i \mid 1\le i\le n\}$, then $\theta(t)=\{t\}$.
3. If $t=f(t_1,\ldots,t_n)$, then $\theta(t)=\{f(t'_1,\ldots,t'_n)\mid t'_i\in\theta(t_i), 1\le i\le n\}$.

For tree languages L_1 and L_2 and a substitution constant $\Box$, define $L_1\circ_\Box L_2$ as follows. If $L_1=\{t\}$, then $\{t\}\circ_\Box L_2 = t\{(\Box, L_2)\}$. If $|L_1|\neq 1$, then $L_1\circ_\Box L_2 = \bigcup_{t\in L_1}\{t\}\circ_\Box L_2$. Define $L^{\star_\Box}=\bigcup_{j\ge 0} L^{j,\Box}$ where

$$\begin{aligned} L^{0,\Box} &= \{\Box\} \\ L^{n+1,\Box} &= L^{n,\Box}\cup L\circ_\Box L^{n,\Box} \end{aligned}$$

In the book by Comon *et al.*[1], languages do not contains symbols in Γ. In the survey by Gécseg[3], substitution constants can be elements of languages. This paper follows the later one because substitution constants are essential in the algebraic structure as we see later.

Using the above functions, we define an interpretation of regular tree expressions by the following function.

$$[\![\_]\!] \colon \mathsf{RegExp}(\Sigma, \Gamma) \to \mathscr{P}(\mathsf{T}_{\Sigma\cup\Gamma})$$

Let e be a regular tree expression.

1. $[\![\mathbf{0}]\!]$ is the empty set.
2. If $e \in \mathsf{T}_{\Sigma\cup\Gamma}$, then $[\![e]\!] = \{e\}$.
3. If e has the form $e_1 + e_2$, then $[\![e_1 + e_2]\!] = [\![e_1]\!] \cup [\![e_2]\!]$.
4. If e has the form $e_1 \cdot_\square e_2$, then $[\![e_1 \cdot_\square e_2]\!] = [\![e_1]\!] \circ_\square [\![e_2]\!]$.
5. If e has the form $e_0^{*\square}$, then $[\![e_0^{*\square}]\!] = [\![e_0]\!]^{\star\square}$.

The image of the interpretation function above coincides with the class of *regular tree languages*[1]. Let $\mathsf{Reg}(\Sigma, \Gamma)$ be the set of regular tree languages on signature Σ and set Γ of substitution constants. The definition of ${}^{\star\square}$ can be changed into $L^{\underline{\star}\square} = \bigcup_{j\geq 0} L^{\underline{j}}$ where

$$\begin{aligned} L^{\underline{0},\square} &= \{\square\} \\ L^{\underline{n+1},\square} &= L^{\underline{n},\square} \circ_\square (L \cup \{\square\}) \end{aligned}$$

The proposition below can be shown by induction on n of $L^{n,\square}$ and $L^{\underline{n},\square}$.

Proposition 1. *For a tree language L, $L^{\star\square} = L^{\underline{\star}\square}$ holds.* $\square$

3 Monodic tree Kleene algebra

In this section, we give an essentially algebraic structure of the subclass of regular tree languages, which will be shown in the next section. After giving the axioms, we show some basic properties of the algebra.

Definition 1. A *monodic tree Kleene algebra* $(A, +, \cdot, 0, 1, {}^*)$ satisfies the following equations and Horn clauses where $\cdot$ is omitted.

$$\begin{aligned}
a + (b + c) &= (a + b) + c & (1)\\
a + b &= b + a & (2)\\
a + 0 &= a & (3)\\
a + a &= a & (4)\\
a(bc) &= (ab)c & (5)\\
1a &= a & (6)\\
a1 &= a & (7)\\
ac + bc &= (a + b)c & (8)\\
ab + ac &\leq a(b + c) & (9)\\
0a &= 0 & (10)\\
1 + aa^* &\leq a^* & (11)\\
1 + a^*(a + 1) &\leq a^* & (12)\\
b + ax \leq x &\rightarrow a^*b \leq x & (13)\\
b + x(a + 1) \leq x &\rightarrow ba^* \leq x & (14)
\end{aligned}$$

The order is defined as $a \leq b$ if $a + b = b$.

Operators $+$, $\cdot$ and * are respectively called an addition, a multiplication and a Kleene star. Axioms (11) and (12) are sometimes called unfold laws. Axioms (13) and (14), which are called induction laws, can be replaced by the following axioms, respectively.

$$\begin{aligned}
ax \leq x &\rightarrow a^*x \leq x & (15)\\
x(a + 1) \leq x &\rightarrow xa^* \leq x & (16)
\end{aligned}$$

The equivalence between (13) and (15) can be shown in the same way of Kleene algebras. From (14) to (16), for showing $xa^* \leq x$, it is sufficient to hold that $x + x(a + 1) \leq x$, which can be shown by the assumption. From (16) to (14), we assume $b + x(a + 1) \leq x$. From the assumption, $b \leq x$, $x(a + 1) \leq x$ and $xa^* \leq x$ hold. $ba^* \leq x$ is from $b \leq x$. Remark that if x is either 0 or 1, we have the right-induction law $b + ax \leq x \rightarrow ba^* \leq x$ of Kleene algebras.

We compare to the original Kleene algebras. The proof will be shown later.

Proposition 2. *The right-unfold law $1 + a^*a \leq a^*$ is of Kleene algebras a theorem of monodic tree Kleene algebras but the right-induction law $b + xa \leq x \rightarrow ba^* \leq x$ of Kleene algebras is not a theorem.* □

The right-unfold law $1 + a^*a \leq a^*$ of Kleene algebras holds according to the axiom $1 + a^*(a + 1) \leq a^*$ with partial right-distributivity (9).

A lazy Kleene algebra by Möller[5] also gives up right-distributivity but does not have right-unfold and right-induction laws.

Lemma 3. *Operations $\cdot$ and * in a monodic tree Kleene algebra are monotone.*

Proof. Assuming $a \le b$, we show $ac \le bc$, $ca \le cb$ and $a^* \le b^*$. From the fact that $a + b = b$, we have $bc = (a + b)c$. By (8), $ac + bc = bc$ holds and thus $ac \le bc$. From $a + b = b$, we have $cb = c(a + b)$. By (10), $ca + cb \le cb$ holds and thus $ca \le cb$. To obtain $a^* \le b^*$, we first show $1 + ab^* \le b^*$ (by (13)). From monotonicity of $+$ and $\cdot$, we have $1 + ab^* \le 1 + b \cdot b^*$. From (11), $1 + b \cdot b^* \le b^*$ holds and thus $1 + ab^* \le b^*$. □ □

The theorems below are used in the proof of the completeness theorem.

Lemma 4. *The following are theorems of monodic tree Kleene algebras.*

$$
\begin{aligned}
(a+1)^* &= a^* && (17)\\
(a^*)^* &= a^* && (18)\\
1 + aa^* &= a^* && (19)\\
1 + a^*(a+1) &= a^* && (20)\\
(a+b)^* &= a^*(ba^*)^* && (21)\\
(ab)^*a &\le a(ba)^* && (22)
\end{aligned}
$$

Proof. By (13), to show that $(a+1)^* \le a^*$, it is sufficient to show $1+(a+1)a^* \le a^*$, which is obtained by distributivity and (11). The inequation $(a^*)^* \le a^*$ can be shown as follows.

$$
\begin{aligned}
a^* + aa^* &\le a^* && \text{(by (11))}\\
1 + a^*a^* &\le a^* && \text{(by (13))}\\
(a^*)^* &\le a^* && \text{(by (13))}
\end{aligned}
$$

By (13), to show $a^* \le 1 + aa^*$, it is sufficient to show $1 + a(1 + aa^*) \le 1 + aa^*$, which is from (11). By (14), to show, $a^* \le 1 + a^*(a + 1)$ it is sufficient to show $1 + (1 + a^*(a + 1))(a + 1) \le 1 + a^*(a + 1)$, which is obtained by (12) and monotonicity.

Since the original proof of $(a + b)^* \le a^*(b^*a)^*$ in Kleene algebras does not involve the right-induction law and right-distributivity[4], the above equation also holds in our setting. The inequation $a^*(b^*a)^* \le (a + b)^*$ in (22) can be shown as follows. First, we have $(a^*b)^*a^* \le ((a+b+1)^*(a+b+1))^*(a+b+1)^*$ from monotonicity.

$$
\begin{aligned}
&((a+b+1)^*(a+b+1))^*(a+b+1)^*\\
&\quad \le ((a+b+1)^*(a+b+1+1))^*((a+b+1)^*+1)\\
&\quad \le ((a+b+1)^*)^*((a+b+1)^*+1) && \text{(by (12))}\\
&\quad \le ((a+b+1)^*)^* && \text{(by (12))}\\
&\quad \le (a+b+1)^* && \text{(by (18))}\\
&\quad \le (a+b)^* && \text{(by (17))}
\end{aligned}
$$

By (13), to show $(ab)^*a \le a(ba)^*$, it is sufficient to show $a + aba(ba)^* \le a(ba)^*$, which holds by (11). □ □

Next, we show the set of regular tree expressions satisfies the axioms of monodic tree Kleene algebras.

Lemma 5. *For two tree languages S and T in $\mathsf{T}_{\Sigma\cup\{\Box\}}$, (i) $T^{\star\Box}\circ_{\Box} S$ is the least fixed point of function $\lambda X.\ S\cup T\circ_{\Box} X$ and (ii) $S\circ_{\Box} T^{\star\Box}$ is the least fixed point of function $\lambda X.\ S\cup X\circ_{\Box}(T\cup\{\Box\})$.*

Proof. In the proof, we write $\circ$ and * for $\circ_\Box$ and $^{\star\Box}$, respectively. (i) $T^\star\circ S = S\cup T\circ(T^\star\circ S)$ can be shown easily. For tree language α, we assume

$$\alpha = S\cup T\circ\alpha \tag{23}$$

and show that $T^\star\circ S\subseteq\alpha$. For any $n\geq 0$, it is sufficient to show that $T^n\circ S\subseteq\alpha$. For the base case $n=0$, the lemma holds from (23). Assume the lemma holds for the case $n=i$, and we show the case when $n=i+1$.

$$\begin{aligned} T^{i+1}\circ S &= (T\cup T\circ T^i)\circ S \\ &= T\circ S\cup(T\circ T^i)\circ S \end{aligned}$$

By inductive hypothesis, we have $T\circ S\subseteq\alpha$. For $(T\circ T^i)\circ S$ by (23), it is sufficient to show $(T\circ T^i)\circ S\subseteq T\circ\alpha$, which can be shown by inductive hypothesis and monotonicity and associativity of $\circ$.

(ii) We can show $S\circ T^\star = S\cup(S\circ T^\star)\circ(T\cup\{\Box\})$ easily. For tree language β, assume

$$\beta = S\cup\beta\circ(T\cup\{\Box\}) \tag{24}$$

and we show $S\circ T^\star\subseteq\beta$. By Proposition 1, we can use another definition of Kleene star. For $n\geq 0$, it is sufficient to show $S\circ T^{\underline{n}}\subseteq\beta$. For the base case $n=0$, the lemma holds from (24). Assume the lemma holds for $n=i$, we show $n=i+1$.

$$\begin{aligned} S\circ T^{\underline{i+1}} &= S\circ(T^{\underline{i}}\circ(T\cup\{\Box\})) \\ &= (S\circ T^{\underline{i}})\circ(T\cup\{\Box\}) \end{aligned}$$

The lemma holds from (24), inductive hypothesis and monotonicity of $\circ$. □ □

We can show that the other axioms from (1) to (10) of monodic tree Kleene algebras are satisfied by tree languages with functions $\cup, \circ_\Box, ^{\star\Box}$ and constants $\emptyset, \{\Box\}$ via easy observations. Using the lemma above, we have the following theorem.

Theorem 6. *Let Σ be a signature and Γ be a set of substitution constants. For any $\Box\in\Gamma$,*

$$(\mathsf{Reg}(\Sigma,\Gamma),\emptyset,\{\Box\},\cup,\circ_\Box,^{\star\Box})$$

is a monodic tree Kleene algebra. □

Since Lemma 5 does not depend on the regularity of the two languages in the claim, we can obtain the following proposition.

Proposition 7. *Let Σ be a signature and Γ be a set of substitution constants. For any $\Box \in \Gamma$, $(\mathscr{P}(\mathsf{T}_{\Sigma\cup\Gamma}), \emptyset, \{\Box\}, \cup, \circ_\Box, {}^{\star_\Box})$ is a monodic tree Kleene algebra.* $\Box$

Here, we prove Proposition 2 by giving a counterexample. For a tree language L and a substitution constant $\Box$, define $L^{\overline{\star}_\Box}$ by $L^{\overline{\star}_\Box} = \bigcup_{j\geq 0} L^{\overline{j},\Box}$ where

$$\begin{aligned} L^{\overline{0},\Box} &= \{\Box\} \quad \text{and} \\ L^{\overline{n+1},\Box} &= L^{\overline{n},\Box} \circ_\Box L. \end{aligned}$$

For tree language $\{f(\Box,\Box)\}$, $\{f(\Box,\Box)\}^{\overline{\star}}$ consists of complete binary trees and we have $\{\Box\} \cup \{f(\Box,\Box)\}^{\overline{\star}} \circ_\Box \{f(\Box,\Box)\} \subseteq \{f(\Box,\Box)\}^{\overline{\star}}$, which corresponds to the assumption of the right-induction law of a Kleene algebra. On the other hand, we can see that $\{f(\Box,\Box)\}^{\star} \not\subseteq \{f(\Box,\Box)\}^{\overline{\star}}$ since $\{f(\Box,\Box)\}^{\star}$ contains terms more than complete binary trees.

Next, we introduce matrices on monodic tree Kleene algebras and operations on the matrices. Let K be a monodic tree Kleene algebra and $\mathsf{M}(n,K)$ be the class of n by n matrices on K. In the following, we assume E and X are matrices as follows.

$$E = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \qquad X = \begin{pmatrix} x & y \\ z & w \end{pmatrix} \tag{25}$$

The addition and multiplication matrices in $\mathsf{M}(n,K)$ is defined in the usual way.

Lemma 8. $\mathsf{M}(n,K)$ *satisfies axioms (1)–(10).* $\Box$

Kleene star is defined essentially in the same way of Kleene algebras.

$$E^* = \begin{pmatrix} a & b \\ c & d \end{pmatrix}^* = \begin{pmatrix} (a+bd^*c)^* & (a+bd^*c)^*bd^* \\ (d+ca^*b)^*ca^* & (d+ca^*b)^* \end{pmatrix}$$

The definition of Kleene star for n by n matrices is inductively given in a similar way of Kleene algebras.

$$\left(\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right)^* = \left(\begin{array}{c|c} (A+BD^*C)^* & (A+BD^*C)^*BD^* \\ \hline (D+CA^*B)^*CA^* & (D+CA^*B)^* \end{array}\right)$$

Lemma 9. *For matrices E and X in $\mathsf{M}(n,K)$, the matrix E^* satisfies the following monodic tree Kleene algebra axioms (11)–(13) where I is the identity matrix.*

$$I + EE^* \leq E^* \tag{26}$$
$$I + E^*(E+I) \leq E^* \tag{27}$$
$$EX \leq X \rightarrow E^*X \leq X \tag{28}$$

Proof. Since in this proof, the axiom (14) is not used, the cases for arbitrary $n \geq 1$ can be shown by induction and thus we give only the base, i.e. $n = 2$.

Let E and X be the matrices in (25). The inequation (26) can be written as the following four inequations.

$$\begin{array}{rcl}
1 + a(a+bd^*c)^* + b(d+ca^*b)^*ca^* & \leq & (a+bd^*c)^* \\
a(a+bd^*c)^*bd^* + b(d+ca^*b)^*d^* & \leq & (a+bd^*c)^*bd^* \\
c(a+bd^*c) + d(d+ca^*b)^*ca^* & \leq & (d+ca^*b)^*ca^* \\
1 + c(a+bd^*c)^*bd^* + d(d+ca^*b)^* & \leq & (d+ca^*b)^*
\end{array}$$

For example, we can show $b(d+ca^*b)^*ca^* \leq (a+bd^*c)^*$ as follows.

$$\begin{array}{rcll}
b(d+ca^*b)^*ca^* & \leq & bd^*(ca^*bd^*)^*ca^* & \text{(by (21))} \\
& \leq & bd^*c(a^*bd^*c)^*a^* & \text{(by (22))} \\
& \leq & bd^*ca^*(bd^*ca^*)^* & \text{(by (22))} \\
& \leq & (a+bd^*c)a^*(bd^*ca^*)^* & \\
& = & (a+bd^*c)(a+bd^*c)^* & \\
& \leq & (a+bd^*c)^* &
\end{array}$$

The rests can be shown in similar ways.

The inequation (27) consists of the following four inequations and each of them can be easily shown.

$$\begin{array}{rcl}
1 + (a+bd^*c)^*(a+1) + (a+bd^*c)^*bd^*c & \leq & (a+bd^*c)^* \\
(a+bd^*c)^*b + (a+bd^*c)^*bd^*(d+1) & \leq & (a+bd^*c)^*bd^* \\
(d+ca^*b)^*ca^*(a+1) + (d+ca^*b)^*c & \leq & (d+ca^*b)^*c \\
1 + (d+ca^*b)ca^*b + (d+ca^*b)^*(d+1) & \leq & (d+ca^*b)^*
\end{array}$$

For (28), we show that the assumptions $ax+by \leq x$ and $cx+dy \leq y$ imply the following inequations.

$$\begin{array}{rcl}
(a+bd^*c)^*x + (a+bd^*c)^*bd^*y & \leq & x \\
(d+ca^*b)^*ca^*x + (d+ca^*b)^*y & \leq & y
\end{array}$$

We only show $(a+bd^*c)^*x \leq x$.

$$\begin{array}{rcll}
d^*y & \leq & y & (dy \leq y \text{ and } (15)) \\
bd^*y & \leq & x & (by \leq x) \\
bd^*cx & \leq & x & (cx \leq y) \\
ax + bd^*cx & \leq & x & (ax \leq x) \\
(a+bd^*c)x & \leq & x & \\
(a+bd^*c)^*x & \leq & x &
\end{array}$$

□ □

Consequently, the set of matrices on a monodic tree Kleene algebra satisfies all the axioms of a monodic tree Kleene algebra except for (14).

According to Lemmas 8 and 9, we can see that matrices on a monodic tree Kleene algebra has a monodic tree Kleene algebra *like* structure. Some of theorems of monodic tree Kleene algebras also hold in $\mathsf{M}(n, K)$.

Lemma 10. *(i) Operations $\cdot$ and * on $\mathsf{M}(n, K)$ is monotone. (ii) The equations and inequations (17)–(19) and (21)–(22) hold in $\mathsf{M}(n, K)$.* □

A matrix in which any entries are either 0 or 1 is called a 0-1 *matrix*. Althouh the lemma below mentions properties of $\mathsf{M}(n, K)$ concerning with 0-1 matrices, each statement in the lemma can also be applied to 0-1 vectors.

Lemma 11. *Let X be a* 0-1 *matrix, P be a permutation matrix and A and B be matrices, then the following equations hold where P^T is the transpose of P.*

$$
\begin{aligned}
X(A+B) &= XA + XB && (29)\\
XA \le X &\to XA^* \le X && (30)\\
B + XA \le X &\to BA^* \le X && (31)\\
AX = XB &\to A^*X = XB^* && (32)\\
X(AX)^* &= (XA)^*X && (33)\\
(P^TAP)^* &= P^TA^*P && (34)
\end{aligned}
$$

Proof. Since in this proof, the axiom (14) is only used for the base case and for the inductive step, (31) can be used as an inductive hypothesis. Hence, we give only the base $n = 2$. The first one is obvious. Let E and X be the matrices in (25). To prove (30), we show the following statement.

$$X(A + I) \le X \to XA^* \le X \tag{35}$$

The left-hand side of (35) consists of the following inequations.

$$
\begin{aligned}
x(a + 1) + yc &\le x\\
xb + y(d + 1) &\le y\\
z(a + 1) + wc &\le z\\
zb + w(d + 1) &\le w
\end{aligned}
$$

The right-hand side consists the following inequations.

$$
\begin{aligned}
x(a + bd^*c)^* + y(d + ca^*b)^*ca^* &\le x\\
x(a + bd^*c)^*bd^* + y(d + ca^*b)^* &\le y\\
z(a + bd^*c)^* + w(d + ca^*b)^*ca^* &\le z\\
z(a + bd^*c)^*bd^* + w(d + ca^*b)^* &\le z
\end{aligned}
$$

For example, $x(a + bd^*c)^* \le x$ can be shown as follows.

$$
\begin{aligned}
yd^* &\le y && (y(d + 1) \le y)\\
yd^*c &\le x && (yd^* \le y \text{ and } yc \le x)\\
xbd^*c &\le x && (yd^*c \le x \text{ and } xb \le y)\\
x(a + bd^*c + 1) &\le x && (xbd^*c \le x \text{ and } x(a + 1) \le x)\\
x(a + bd^*c)^* &\le x && (x(a + bd^*c + 1) \le x)
\end{aligned}
$$

The last step is by the axiom (14) for the base case or by the inductive hypothesis of for the induction step. Finally, we can see that $EX \leq X$ implies $E(X+I) \leq X$ and the lemma holds.

The Horn clause (31) is directly obtained from (30).

The simulation law (32) can be shown in the same way of the original proof by Kozen[4], since as we have shown that the Kleene algebra axioms (Lemma 9 and (31)) hold in our setting if X is restricted to a 0-1 matrix.

The shift law for specific case (33), i.e. $X(AX)^* = (XA)^*X$, is obtained from (32) by replacing A with XA and B with AX, respectively.

To prove (34), we show $A^*P = P(P^TAP)^*$. Multiplying P^T from left and the facts that $P^TP = I$ and $PP^T = I$, we obtain $(P^TAP)^* = P^TA^*P$. By Lemma 10, we obtain $(PP^TA)^*P \leq P(P^TAP)^*$. Since P is a 0-1 matrix, by (31) in this lemma, to show $P(P^TAP)^* \leq (PP^TA)^*P$, it is sufficient to show that $P + (PP^TB)^*(PP^TB)P \leq (PP^TB)^*P$. □ □

4 Subclass of regular tree expressions

In this section, we give subclass of regular tree expressions, called monodic regular tree expressions.

Definition 2. Let Σ be a signature and $\Box$ be a substitution constant. The set $\mathsf{RegExp}(\Sigma, \Box)$ of *monodic regular tree expressions* is defined as follows.

1. The symbol $\mathbf{0}$ is a monodic regular tree expression.
2. A term of the form $f(\Box, \ldots, \Box)$ is a monodic regular tree expression.
3. If e_1 and e_2 are regular tree expressions, so are $e_1 + e_2$, $e_1 \cdot e_2$ and e_1^*.

The set of monodic regular tree expressions is a subclass of regular tree expressions when the multiplication $\cdot$ and the Kleene star $*$ are regarded as $\cdot_\Box$ and $^{*\Box}$, respectively. The interpretation of monodic regular tree expressions is given by functions $\cup$, $\circ_\Box$ and $^{\star\Box}$ on tree languages.

Definition 3. A tree language L which can be expressed by a monodic regular tree expressions is called *monodic regular*.

We denote the set of all monodic regular tree languages by $\mathsf{Reg}(\Sigma, \Box)$.

Proposition 12. *(i) The set* $\mathsf{Reg}(\Sigma, \Box)$ *of monodic regular tree language is closed under functions* $\cup$, $\circ_\Box$ *and* $^{\star\Box}$. *(ii)* $(\mathsf{Reg}(\Sigma, \Box), \emptyset, \{\Box\}, \cup, \circ_\Box, ^{\star\Box})$ *is a monodic tree Kleene algebra.* □

Example 1. A regular tree expression $f(\Box, \Box) \cdot_\Box (g(\Box) + h(\Box, \Box))$ is monodic but $f(a, c)$ and $(f(\Box, \Box_1) \cdot_\Box a) \cdot_{\Box_1} c$ are not. □

Next, we introduce the subclass of tree automata corresponding to monodic regular tree languages. For the definition of behaviors of tree automata, please refer to the books[1, 3].

Definition 4 ([1])**.** A *tree automaton* is a tuple $(\Sigma, Q, Q_{final}, \Delta)$ where Σ is a signature, Q is a finite set of *states*, $Q_{final} \subseteq Q$ is a set of *final states* and Δ is a set of *transition rules* of the form either $f(q_1, \ldots, q_n) \to q$ or $q' \to q$ where $f \in \Sigma_n, q_1, \ldots, q_n, q, q' \in Q$.

Definition 5. A tree automaton in which the left-hand side of each transition rule has only one kind of states is called *monodic*.

In the following, we give a matrix representation of monodic tree automaton $A = (\Sigma, Q, Q_{final}, \Delta)$. Without loss of generality, Q can be written as $\{1, \ldots, n\}$ for some integer $n \geq 1$. Let M_A be a matrix in $\mathsf{M}(n+1, \mathsf{RegExp}(\Sigma, \Box))$ where the (p, q) entry is given by the following formula.

$$\begin{gathered}\sum\{f(\Box, \ldots, \Box) \mid f(q, \ldots, q) \to p \in \Delta, \mathtt{arity}(f) \geq 1\} \cup \\ \{\Box \mid q \to p \in \Delta\} \cup \{c \mid c \to p \in \Delta, q = n+1\}\end{gathered}$$

Let v be a $n+1$ vector in which the $n+1$-th row is $\Box$ and the others are 0 and u be a $n+1$ vector in which for each $q \in Q_{final}$, the q-th row is $\Box$ and the others are 0. The triple (v, M_A, u) is called a *matrix representation* of A where v and u are vectors and M_A is a matrix on the free monodic tree Kleene algebra over Σ and $\Box$, i.e. the quotient of monodic regular tree expressions modulo provable equivalence. We call the vector u the *final vector* and v the *initial vector*.

Example 2. Let A_0 be a tree automaton in which Δ consists of the following transition rules

$$f(1,1) \to 1 \quad g(1,1) \to 2 \quad a \to 1 \quad b \to 1 \quad b \to 2$$

and the final state is just 2. Then the matrix representation is as follows.

$$\left(\begin{pmatrix} 0 \\ 0 \\ \Box \end{pmatrix}, \begin{pmatrix} f(\Box,\Box) & 0 & a+b \\ g(\Box,\Box) & 0 & b \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \Box \\ 0 \end{pmatrix} \right)$$

Since $Q = \{1, 2\}$, the matrix has size $2 + 1 = 3$. □

Next, we justify the matrix representation of tree automata via regular tree equation systems[1]. Let $X_1, \ldots, X_n$ be variables and $s_{i,j}$ $(1 \leq i \leq m_j, 1 \leq j \leq p)$ be terms in $\mathsf{T}_\Sigma(\{X_1, \ldots, X_n\})$, which is the set of terms with variables $\{X_1, \ldots, X_n\}$. A *regular tree equation system* S is given by the following set of equations.

$$\begin{aligned} X_1 &= s_{1,1} + \cdots + s_{m_1,1} \\ &\cdots \\ X_p &= s_{1,p} + \cdots + s_{m_p,p} \end{aligned}$$

A *solution of* S is a p-tuple of tree languages $(L_1, \ldots, L_p) \in \mathscr{P}(\mathsf{T}_\Sigma)^p$ satisfying the following condition.

$$\begin{aligned} L_1 &= \theta(s_{1,1}) \cup \cdots \cup \theta(s_{m_1,1}) \\ &\cdots \\ L_p &= \theta(s_{1,p}) \cup \cdots \cup \theta(s_{m_p,p}) \end{aligned}$$

where θ is a tree substitution $\{(X_1, L_1), \ldots, (X_p, L_p)\}$. For regular tree equation system S, we define $\hat{S}\colon \mathscr{P}(\mathsf{T}_\Sigma)^p \to \mathscr{P}(\mathsf{T}_\Sigma)^p$ as $\lambda(L_1, \ldots, L_p).\ (L'_1, \ldots, L'_p)$ where

$$\begin{aligned} L'_1 &= L_1 \cup \theta(s_{1,1}) \cup \cdots \cup \theta(s_{m_1,1}), \\ &\ldots \\ L'_p &= L_p \cup \theta(s_{1,p}) \cup \cdots \cup \theta(s_{m_p,p}) \end{aligned}$$

and $\theta = \{(X_1, L_1), \ldots, (X_p, L_p)\}$. The order on $\mathscr{P}(\mathsf{T}_\Sigma)^p$ is defined componentwise.

Lemma 13 ([1]). *For regular tree equation system S, the least fixed-point of $\hat{S}$ is the least solution of S* □

Theorem 14 ([1]). *For any regular tree equations, the least solution is a regular tree language. Conversely, for any regular tree language, there exists a regular tree equations representing the regular tree language.* □

In the proof of the above theorem, tree automaton $A = (\Sigma, Q, Q_{final}, \Delta)$ with $Q = \{1, \ldots, n\}$ is translated into regular tree equation system S_A consisting of

$$i = \sum\{l \mid l \to i \in \Delta\} \qquad (1 \leq i \leq n)$$

where states are regarded as variables. More precisely, $m_1, \ldots, m_p$ are p and $s_{i,j}$ is $\sum\{l \mid f(j, \ldots, j) \to i \in \Delta, f \in \Sigma\}$. If A is monodic, then we can obtain another definition of $\hat{S_A}$ as $\lambda(L_1, \ldots, L_p).\ (L'_1, \ldots, L'_p)$ where

$$\begin{aligned} L'_1 &= L_1 \cup \theta_1(s_{1,1}) \cup \cdots \cup \theta_1(s_{p,1}), \\ &\ldots \\ L'_p &= L_p \cup \theta_1(s_{1,p}) \cup \cdots \cup \theta_p(s_{p,p}) \end{aligned}$$

and $\theta_n = \{(n, L_n)\}$ for $1 \leq n \leq p$. For a substitution constant $\Box$, we have $\theta_n = \{(n, \{\Box\})\} \circ \{(\Box, L_n)\}$ for $1 \leq n \leq p$.

Summarizing the observations above, the fixed-point operator $\hat{S_A}$ of the above regular tree equation system S_A can be written as the matrix

$$X = (P + I + C)X = (P + I)X + C$$

where I is the identity matrix of size n, $X = (X_1, \ldots, X_n)^T$, $C = (C_1, \ldots, C_n)^T$, $C_i = \sum\{c \mid c \to i \in \Delta, c \in \Sigma_0\}$ and $P_{i,j} = \sum\{f(\Box, \ldots, \Box) \mid f(j, \ldots, j) \to i \in \Delta, f \in \Sigma\}$. The matrix $P + I$ corresponds to the fixed-point operator. The least solution is given by the least-fixed point and thus we have $(P + I)^* C = P^* C$. This is the language represented by the tree automaton A. More precisely, the sequence of languages represented by each state of A. Although A can be represented by matrices C and P and final 0-1 vector U, for the discussions below initial vectors also have to be given by 0-1 vectors. Henceforth, we give an initial vector, a final vector and a matrix M_A as follows.

$$\left(\left(\begin{array}{c} \mathbf{0} \\ \hline 1 \end{array}\right), \left(\begin{array}{c|c} P & C \\ \hline \mathbf{0} & 0 \end{array}\right), \left(\begin{array}{c} U \\ \hline 0 \end{array}\right) \right)$$

This matrix-represented automaton corresponds to the following regular tree equation system in which entries of C are produced from the new variable x_0.

$$\left(\begin{array}{c} X \\ \hline x_0 \end{array}\right) = \left(\begin{array}{c} 1 \\ \hline \mathbf{0} \end{array}\right) + \left(\left(\begin{array}{c|c} P & C \\ \hline \mathbf{0} & 0 \end{array}\right) + I\right) \left(\begin{array}{c} X \\ \hline x_0 \end{array}\right)$$

The least-fixed point can be computed as follows.

$$\left(\begin{array}{c|c} P & C \\ \hline \mathbf{0} & 0 \end{array}\right)^* \left(\begin{array}{c} \mathbf{0} \\ \hline 1 \end{array}\right) = \left(\begin{array}{c|c} P^* & P^*C \\ \hline \mathbf{0} & 1 \end{array}\right) \left(\begin{array}{c} \mathbf{0} \\ \hline 1 \end{array}\right) = \left(\begin{array}{c} P^*C \\ \hline 1 \end{array}\right)$$

Using the initial vector, we have the following expressions.

$$\left(\begin{array}{c|c} U & 0 \end{array}\right) \left(\begin{array}{c} P^*C \\ \hline 1 \end{array}\right)$$

Finally, by the final vector, we can retrieve the language represented by the tree automaton.

In the following, we also deal with languages including substitution constants. This means that we also consider tree automata in which a initial vector may not only be of the form $(1, \mathbf{0})^T$ but also any 0-1 vectors.

Lemma 15. *For a tree automaton A and its matrix representation (v, M_A, u), the language accepted by A is $[u^T M_A^* v]$.*

Proof. (sketch) We regard the tree automaton as a regular tree equation system, then the lemma holds because of Lemmas 5 and 13 and the discussion after Theorem 14. □ □

Example 3. Let A_0 be the tree automaton considered in Example 2. Then the corresponding expression can be obtained as follows.

$$\begin{aligned}
M_{A_0}^* &= \left(\begin{array}{cc|c} f(\square,\square) & 0 & a+b \\ g(\square,\square) & 0 & b \\ \hline 0 & 0 & 0 \end{array}\right)^* \\
&= \left(\begin{array}{c|c} (A+BD^*C)^* & (A+BD^*C)^*BD^* \\ \hline (D+CA^*B)^*CA^* & (D+CA^*B)^* \end{array}\right) \\
&= \left(\begin{array}{c|c} A^* & A^*B \\ \hline \mathbf{0} & 0 \end{array}\right) \\
&= \left(\begin{array}{ccc} f(\square,\square)^* & f(\square,\square)^* & f(\square,\square)^* \cdot (a+b) + f(\square,\square)^* \cdot b \\ g(\square,\square) \cdot f(\square,\square)^* & \square & g(\square,\square) \cdot f(\square,\square)^* \cdot (a+b) + b \\ 0 & 0 & 0 \end{array}\right)
\end{aligned}$$

Finally, we have the following expression.

$$\left(\begin{array}{ccc} 0 & \square & 0 \end{array}\right) M_{A_0}^* \left(\begin{array}{c} 0 \\ 0 \\ \square \end{array}\right) = g(\square,\square) \cdot f(\square,\square)^* \cdot (a+b) + b$$

□

The proof of the following theorem follows the original one by Kozen[4] as follows.

1. First, we construct tree automata for given two regular tree expressions.
2. Second, we translate the tree automata for deterministic ones.
3. Then, we minimamize the tree automata.

In the proof, we use the following lemmas.

- the theorems in Lemma 4 of monodic tree Kleene algebras
- the lemmas in Lemmas 8, 9 and 10 of arbitrary matrices on a monodic tree Kleene algebra
- the lemmas in Lemma 11 of 0-1 matrices on a monodic tree Kleene algebra

Theorem 16. *Let* α *and* β *be monodic regular tree expressions such that* $[\alpha] = [\beta]$ *and* $[\alpha] \subseteq \mathtt{T}_\Sigma$. *Then* $\alpha = \beta$ *is a theorem of monodic tree Kleene algebras.* $\square$

5 Conclusion

In this paper, we have not yet considered the independence of the axioms in Definition 1. Since for defining the class of regular tree languages, only one function in the statement of Lemma 5 is needed, the axiom (14) may be redundant. Moreover, the axiom (14) is used in the proof of completeness theorem only is the case that x is 0-1. However, the argument of the independence does not affect the soundness and the completeness theorems (Theorems 6 and 16).

The subclass is given by monodic regular tree expressions, i.e. the number of kinds of place-holders is restricted to one. We conjecture that the expressive power of the class coincides with the subclass of regular tree expressions defined below. Let Σ be a signature and a set Γ of substitution constants. The set of *essentially monodic regular tree expressions* is defined as follows.

1. The symbol $\mathbf{0}$ is an essentially monodic regular tree expression.
2. A term of the form $f(\square, \ldots, \square)$ is a monodic regular tree expression for any $\square \in \Gamma$.
3. If e_1 and e_2 are regular tree expressions, so are $e_1 + e_2$, $e_1 \cdot_\square e_2$ and $e_1^{*\square}$ for any $\square \in \Gamma$.

For dealing with the whole class of regular tree expressions, there may be two directions. The first one is to use modal Kleene algebras[6]. A tree can be encoded with two modalities in a modal Kleene algebra. Another direction is to consider products of two monodic tree Kleene algebras. The whole class of regular tree expressions seems a *many-sorted* monodic tree Kleene algebra.

Acknowledgments

The authors appreciate Georg Struth, who visited us with the grant from the International Information Science Foundation (IISF), for a lot of his valuable comments to this study. We also thank to Yasuo Kawahara and Yoshihiro Mizoguchi for fruitful discussions on this study. This research was supported by Core Research for Evolutional Science and Technology (CREST) Program "New High-performance Information Processing Technology Supporting Information-oriented Society" of Japan Science and Technology Agency (JST).

References

[1] Comon, H., Dauchet, M., Gilleron, R., Jacquemard, F., Lugiez, D., Tison, S., Tommasi, M.: Tree automata techniques and applications. Available on: http://www.grappa.univ-lille3.fr/tata/ (1997)

[2] Takai, T., Furusawa, H., Kahl, W.: Reasoning about term rewriting in Kleene categories with converse. In Düntsch, I., Winter, M., eds.: Proceedings of the 3rd Workshop on Applications of Kleene algebera. (2005) 259–266

[3] Gécseg, F., Steinby, M.: Tree languages. In: Handbook of formal languages, 3: beyond words. Springer (1997) 1–68

[4] Kozen, D.: A completeness theorem for Kleene algebras and the algebra of regular events. In Kahn, G., ed.: Proceedings of the Sixth Annual IEEE Symp. on Logic in Computer Science, LICS 1991, IEEE Computer Society Press (1991) 214–225

[5] Möller, B.: Lazy Kleene algebra. In Kozen, D., Shankland, C., eds.: MPC. Volume 3125 of Lecture Notes in Computer Science., Springer (2004) 252–273

[6] Möller, B., Struth, G.: Modal Kleene algebra and partial correctness. In Rattray, C., Maharaj, S., Shankland, C., eds.: AMAST. Volume 3116 of Lecture Notes in Computer Science., Springer (2004) 379–393

A Proof of the completeness theorem

The proof is along with the original one by Kozen.

Definition 6. Let (v, M, u) be a tree automaton on a monodic tree Kleene algebra with $M = n \in \mathsf{M}(n, K)$. Tree automaton (v, M, u) is simple if M has the form

$$M = J + \sum_{f \in \Sigma} M_f$$

where J and M_f is a 0-1 matrix for any $(f \in \Sigma)$. The tree automaton is ε-free if the matrix J is 0 matrix and is deterministic if it is simple, ε-free and each column has just one entry of 1.

Lemma 17. *For a monodic regular tree expression e on Σ and $\Box$, there is a integer n and a monodic tree automaton (v, M, u) such that $e = u^T M^* v$ and $M \in \mathsf{M}(n, \mathsf{RegExp}(\Sigma, \Box))$.*

Proof. The claim $M \in \mathsf{M}(n, \mathsf{RegExp}(\Sigma, \Box))$ in the lemma says just the size of a matrix may be increased from the size of matrices of induction hypothesis. The proof is shown by induction on the structure of e. If $e = c$ with $(c \in \Sigma_0)$, we consider the following tree automaton.

$$\left(\begin{pmatrix} \Box \\ 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ c & 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \Box \end{pmatrix} \right)$$

The accepting language of the above tree automaton is as follows.

$$\begin{aligned} \begin{pmatrix} 0 & \Box \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 \\ c & 0 \end{pmatrix}^* \cdot \begin{pmatrix} \Box \\ 0 \end{pmatrix} &= \\ \begin{pmatrix} 0 & \Box \end{pmatrix} \cdot \begin{pmatrix} \Box & 0 \\ c & \Box \end{pmatrix} \cdot \begin{pmatrix} \Box \\ 0 \end{pmatrix} &= \Box \cdot c \cdot \Box \\ &= c \end{aligned}$$

If $e = f(\Box, \cdots, \Box)$ for some $f \in \Sigma_n$, consider the following tree automaton.

$$\left(\begin{pmatrix} \Box \\ 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ f(\Box, \ldots, \Box) & 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \Box \end{pmatrix} \right)$$

The accepting language of the above tree automaton is as follows.

$$\begin{aligned} \begin{pmatrix} 0 & \Box \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 \\ f(\Box, \ldots, \Box) & 0 \end{pmatrix}^* \cdot \begin{pmatrix} \Box \\ 0 \end{pmatrix} &= \\ \begin{pmatrix} 0 & \Box \end{pmatrix} \cdot \begin{pmatrix} \Box & 0 \\ f(\Box, \ldots, \Box) & \Box \end{pmatrix} \cdot \begin{pmatrix} \Box \\ 0 \end{pmatrix} &= \Box \cdot f(\Box, \ldots, \Box) \cdot \Box \\ &= f(\Box, \ldots, \Box) \end{aligned}$$

If $e = \alpha \cdot_\Box \beta$, we assume (v, A, u) and (t, B, s) be tree automata such that the followings hold, respectively.

$$\alpha = u^T A^* v \qquad \beta = s^T B^* t$$

We consider the following tree automaton.

$$\left(\left(\frac{v}{t}\right), \left(\begin{array}{c|c} A & 0 \\ \hline 0 & B \end{array}\right), \left(\frac{u}{s}\right)\right)$$

The accepting language of the above tree automaton is as follows.

$$\begin{array}{rcl} (\ u^T \mid s^T\ ) \cdot \left(\begin{array}{c|c} A & 0 \\ \hline 0 & B \end{array}\right)^* \cdot \left(\frac{v}{t}\right) & = & \\ (\ u^T \mid s^T\ ) \cdot \left(\begin{array}{c|c} A^* & 0 \\ \hline 0 & B^* \end{array}\right) \cdot \left(\frac{v}{t}\right) & = & u^T A^* v + s^T B^* t \end{array}$$

If $e = \alpha^*$, let (v, A, u) be a tree automaton such that $\alpha = u^T A^* v$. In this case, we can see that the tree automaton $(v, A + vu^T, u)$ represents $\alpha\alpha^*$ as follows.

$$\begin{array}{rcl} u^T(A + vu^T)^* v & = & u^T A^*(vu^T A^*)^* v \\ & = & u^T A^* v(u^T A^* v)^* \\ & = & \alpha\alpha^* \end{array}$$

By (11), we have $\alpha^* = I + \alpha\alpha^*$ and thus the tree automaton accepting α^* can be constructed from the construction of the addition on matrices. □

Lemma 18. *Let (v, M, u) be an ε-free monodic tree automaton. Then there is a deterministic monodic tree automaton $(\hat{v}, \hat{M}, \hat{u})$ such that*

$$u^T M^* v = \hat{u}^T \hat{M}^* \hat{v}$$

holds.

Proof. Since (v, M, u) is simple, we have 0-1 matrices M_f for $f \in \Sigma$ such that

$$M = \sum_{f \in \Sigma} f \cdot M_f.$$

Let Q be the set of states of (v, M, u) and n be $|Q|$ and we represent an element in $\mathscr{P}(Q)$ by the corresponding vector $\{0, 1\}^n$. For $s \in \mathscr{P}(Q)$, let e_s be the vector of size $\mathscr{P}(Q)$ in which s-th low is $\Box$ and the others are 0. Let X be the n by $\mathscr{P}(Q)$ matrix in which s-th column ($s \in \mathscr{P}(Q)$) is s, i.e. $Xe_s = s$.

On the other hand, for any $f \in \Sigma$, $\hat{M}_f \in \mathsf{M}(\mathscr{P}(Q), \mathsf{RegExp}(\Sigma, \Box))$ be a matrix in which the $s \in \mathscr{P}(Q)$-th column vector is $e_{M_f s}$, i.e. $\hat{M}_f e_s = e_{M_f s}$ holds. Let $\hat{M} = \sum_{f \in \Sigma} f \cdot \hat{M}_f$ $\hat{u} = e_u$ and $\hat{v} = Xv$, the tree automaton $(\hat{v}, \hat{M}, \hat{u})$ is simple and deterministic. We prove the following equation.

$$MX = X\hat{M} \tag{36}$$

$$
\begin{aligned}
MXe_s &= Ms \\
&= \sum_{f\in\Sigma} f\cdot M_f s \\
&= \sum_{f\in\Sigma} fX\cdot e_{M_f s} \\
&= X\sum_{f\in\Sigma} f\cdot \hat{M}_f e_s \\
&= X(\sum_{f\in\Sigma} f\cdot \hat{M}_f) e_s \\
&= X\hat{M}e_s
\end{aligned}
$$

By (36) and the theorems of matrices on monodic tree Kleene algebras,

$$XM^* = \hat{M}^*X \tag{37}$$

holds. Summarizing the discussion above, we have the following derivation.

$$
\begin{aligned}
\hat{u}^T\hat{M}^*\hat{v} &= e_u^T\hat{M}^*Xv \\
&= e_u^TXM^*v \\
&= u^TM^*v
\end{aligned}
$$

□

Lemma 19. *Let Σ be a signature, (v, M, u) be an ε-free tree automaton, there exists a unique minimal deterministic tree automaton $(\bar{v}, \bar{M}, \bar{u})$ such that*

$$u^TM^*v = \bar{u}^T\bar{M}^*\bar{v}$$

holds.

Proof. Let M be an n by n matrix for some integer n and Q be the set of states of the tree automaton (v, M, u), i.e. $Q = \{1, \dots, n\}$. For any q with $1 \le q \le n$, e_q be a vector of size n in which the q-th row is □ and the others are 0. Since (v, M, u) is simple, there are 0-1 matrices M_f for $f \in \Sigma$ such that

$$M = \sum_{f\in\Sigma} f\cdot M_f.$$

For a function symbol $f \in \Sigma$ and an integer q with $1 \le q \le n$, let $\delta(q, f)$ be an integer such that $M_f e_q = e_{\delta(q,f)}$. Since (v, M, u) is deterministic, $\delta(q, f)$ is unique.

A state q $(1 \le q \le n)$ is *reachable* if

$$e_q^TM^*v \ne 0,$$

otherwise, it is *unreachable*. Let Q_R be the set of reachable states and let Q_U be the set of unreachable states, i.e. $Q_U = Q \setminus Q_R$. Partition M into four submatricies M_{RR}, M_{RU}, M_{UR} and M_{UU} such that for $S,T \in \{R,U\}$, M_{ST} is the $Q_S \times Q_T$ submatrices of M. Then M_{RU} is the zero matrix, otherwise a state in Q_U would be reachable.

Let $\equiv$ the congruence relation on the states obtained by the minimizing algorithm for tree automata[1]. According to the properties of $\equiv$, $p \equiv q$ implies $ue_p \equiv ue_q$ and for any $f \in \Sigma$,

$$\delta(p, f) \equiv \delta(q, f) \tag{38}$$

holds. In the following, the equivalence class of q by $\equiv$ and the set of equivalence classes by $\equiv$ are written as $[q]$ and $Q/\equiv$, respectively. For q with $1 \leq q \leq n$, let $e_{[q]}$ be a vector of size $n\equiv$ in which the $[q]$-th row is $\Box$ and the others are 0.

Y be a matrix of size $|Q/\equiv| \times n$ such that the p-th column is the vector representing $[p]$, i.e. $Ye_p = e_{[p]}$. For any $f \in \Sigma$, let $\bar{M}_f \in \mathsf{M}(|\Gamma/\equiv|, \mathsf{RegExp}(\Sigma, \Box))$ be a matrix in which the $[q]$-th column is $e_{[\delta(q,f)]}$, i.e. $\bar{M}_f e_{[q]} = e_{[\delta(q,f)]}$. The matrix $\bar{M}_f$ is well-defined by (38).

$$\begin{aligned} \bar{M} &= \sum_{f\in\Sigma} f \cdot \bar{M}_f \\ \bar{u}^T &= u^T Y \end{aligned}$$

Let $\bar{v}$ be a vector of size $|Q/\equiv|$ such that for any $q \in Q$, $\bar{v}e_{[q]} = ve_q$ holds. By (38) C $\bar{v}$ is well-defined. Moreover, we have $Ye_q\bar{v} = \bar{v}e_{[q]} = ve_p$ and thus $Y\bar{v} = v$. As in the proof of the previous lemma, we show the following equation.

$$YM = \bar{M}Y$$

For any $q \in \Gamma$, we have the following derivation.

$$\begin{aligned} YMe_q &= \sum_{f\in\Sigma} f \cdot YM_q e_q \\ &= \sum_{f\in\Sigma} f \cdot Ye_{\delta(q,f)} \\ &= \sum_{f\in\Sigma} f \cdot e_{[\delta(q,f)]} \\ &= \sum_{f\in\Sigma} f \cdot \bar{M}_f e_{[q]} \\ &= \sum_{f\in\Sigma} f \cdot Y\bar{M}_f e_q \\ &= Y\bar{M}e_q \end{aligned}$$

By (38) and the axioms of monodic tree Kleene algebras,

$$M^*Y = Y\bar{M}^* \tag{39}$$

holds. Summarizing the discussion above, we can show the lemma.

$$\begin{aligned} \bar{u}^T \bar{M}^* \bar{v} &= u^T Y \bar{M}^* \bar{v} \\ &= u^T M^* Y \bar{v} \\ &= u^T M^* v \end{aligned}$$

□

Now, we can prove the main theorem of this paper.

Proof. (proof of Theorem 16) Let $A_1 = (t, M, s)$ and $A_2 = (v, N, u)$ be the minimal deterministic monodic tree automata such that $[\alpha] = [s^T M^* t]$, $[\beta] = [u^T N^* v]$ By Lemmas 17, 18 and 19, both $\alpha = s^T M^* t$ $\beta = u^T N^* v$ are theorems of monodic tree Kleene algebras. On the other hand, from the fact that $[\alpha] = [\beta]$, the automata $A_1 = (t, M, s)$ and $A_2 = (v, N, u)$ are isomorphic. Let P be a permutation matrix giving this isomorphism, then the relation of two automata can be written as follows.

$$\begin{aligned} M &= P^T N P \\ s &= P^T u \\ t &= P^T v \end{aligned}$$

Using $\alpha = \beta$, we have the following derivation.

$$\begin{aligned} \alpha &= s^T M^* t \\ &= (P^T u)^T (P^T N P)^* (P^T v) \\ &= u^T P P^T N^* P P^T v \qquad \text{(by Lemma 11)} \\ &= u^T N^* v \\ &= \beta \end{aligned}$$

□

単口木クリーニ代数
(算譜科学研究速報)
発行日：2006 年 5 月 19 日
編集・発行：独立行政法人産業技術総合研究所関西センター尼崎事業所
システム検証研究センター
同連絡先：〒661-0974 兵庫県尼崎市若王寺 3-11-46
e-mail：informatics-inquiry@m.aist.go.jp

Monodic tree Kleene algebra (preliminary version)
(Programming Science Technical Report)
May. 19, 2006
Research Center for Verification and Semantics (CVS)
AIST Kansai, Amagasaki Site
National Institute of Advanced Industrial Science and Technology (AIST)
3-11-46 Nakouji, Amagasaki, Hyogo, 661-0974, Japan
e-mail: informatics- inquiry@m.aist.go.jp