Decision Methods for Concurrent Kleene Algebra with Tests : Based on Derivative

Yoshiki Nakamura

Tokyo Instutute of Technology, Oookayama, Meguroku, Japan,
nakamura.y.ay@m.titech.ac.jp

Abstract. *Concurrent Kleene Algebra with Tests* (**CKAT**) were introduced by Peter Jipsen[Jip14]. We give derivatives for **CKAT** to decide word problems, for example emptiness, equivalence, containment problems. These derivative methods are expanded from derivative methods for Kleene Algebra and Kleene Algebra with Tests[Brz64][Koz08][ABM12]. Additionally, we show that the equivalence problem of **CKAT** is in EXPSPACE.

Keywords: concurrent kleene algebras with tests, series-parallel strings, Brzozowski derivative, computational complexity

1 Introduction

In this paper, we assume [Jip14, theorem 1] and we use **CKAT** terms as expressions of guarded series-parallel language.

Let Σ be a set of *basic program* symbols $\mathbf{p}_1, \mathbf{p}_2, \ldots$ and T a set of *basic boolean test* symbols $\mathbf{t}_1, \mathbf{t}_2, \cdots$, where we assume that $\Sigma \cap T = \emptyset$. Each $\alpha_1, \alpha_2, \ldots$ denotes a subset of T. *Boolean term* b and **CKAT** *term* p over T and Σ are defined by the following grammar, respectively.

$$b := \mathbf{0} \mid \mathbf{1} \mid \mathbf{t} \in T \mid b_1 + b_2 \mid b_1 b_2 \mid \overline{b_1}$$

$$p := b \mid \mathbf{p} \in \Sigma \mid p_1 + p_2 \mid p_1 p_2 \mid p_1^* \mid p_1 \parallel p_2$$

The guarded series-parallel strings set $GS_{\Sigma,T}$ over Σ and T is a smallest set such that follows

- $\alpha \in GS_{\Sigma,T}$ for any $\alpha \subseteq T$
- $\alpha_1 \mathbf{p} \alpha_2 \in GS_{\Sigma,T}$ for any $\alpha_1, \alpha_2 \subseteq T$ and any basic program $\mathbf{p} \in \Sigma$
- if $w_1\alpha, \alpha w_2 \in GS_{\Sigma,T}$, then $w_1 \alpha w_2 \in GS_{\Sigma,T}$.
- if $\alpha_1 w_1 \alpha_2, \alpha_1 w_2 \alpha_2 \in GS_{\Sigma,T}$, then $\alpha_1\{|w_1, w_2|\}\alpha_2 \in GS_{\Sigma,T}$.

Definition 1 (guarded series-parallel language). *Let $\diamond$ and $\parallel$ be binary operators over $GS_{\Sigma,T}$, respectably. They are defined as follows.*

$$w_1 \diamond w_2 = \begin{cases} w'_1 \alpha w'_2 & (w_1 = w'_1\alpha \text{ and } w_2 = \alpha w'_2) \\ undefined & (o.w.) \end{cases}$$

In particular, if $w_1 = w_2 = \alpha$, then $w_1 \diamond w_2 = \alpha$.

$$w_1 \parallel w_2 = \begin{cases} \alpha_1\{|w'_1, w'_2|\}\alpha_2 & (w_1 = \alpha_1 w'_1 \alpha_2 \textit{ and } w_2 = \alpha_1 w'_2 \alpha_2) \\ \alpha & (w_1 = w_2 = \alpha) \\ undefined & (o.w.) \end{cases}$$

L is a map from **CKAT** *terms over Σ and T to this concrete model by*

- $L(\mathbf{0}) = \emptyset$, $L(\mathbf{1}) = 2^T$
- $L(\mathbf{t}) = \{\alpha \subseteq T \mid \mathbf{t} \in \alpha\}$ *for* $\mathbf{t} \in T$
- $L(\bar{b}) = 2^T \setminus L(b)$
- $L(\mathbf{p}) = \{\alpha_1 \mathbf{p} \alpha_2 \mid \alpha_1, \alpha_2 \subseteq T\}$ *for* $\mathbf{p} \in \Sigma$
- $L(p_1 + p_2) = L(p_1) \cup L(p_2)$
- $L(p_1 p_2) = \{w_1 \diamond w_2 \mid\ w_1 \in G(p_1) \textit{ and } w_2 \in L(p_2) \textit{ and } w_1 \diamond w_2 \textit{ is defined }\}$
- $L(p^*) = \bigcup_{n<\omega} \{\alpha_0 \diamond w_1 \diamond \cdots \diamond w_n \mid \alpha_0 \subseteq T \textit{ and } w_1, \ldots, w_n \in L(p) \textit{ and } \alpha_0 \diamond w_1 \cdots \diamond w_n \textit{ is defined }\}$
- $L(p_1 \parallel p_2) = \{w_1 \parallel w_2 \mid w_1 \in L(p_1) \textit{ and } w_2 \in L(p_2) \textit{ and }\ w_1 \parallel w_2 \textit{ is defined }\}$

We expand L to $\overline{L}(P) = \bigcup_{p \in P} L(p)$, where P is a set of **CKAT** *terms. Furthermore, let $L_\alpha(p) = \{\alpha w \mid \alpha w \in L(p)\}$.*

In guarded series-parallel strings, $\alpha_1\{|w_1, w_2|\}\alpha_2$ has commutative(i.e. $\alpha_1\{|w_1, w_2|\}\alpha_2 = \alpha_1\{|w_2, w_1|\}\alpha_2$). We define $p_1 = p_2$ for two **CKAT** terms p_1 and p_2 as $L(p_1) = L(p_2)$ (by means of [Jip14, Theorem 1]).

2 The Brzozowski derivative for CKAT

Now, we give the naive derivative for **CKAT**. Derivative has applications to many language theoretic problems (e.g. membership problem, emptiness problem, equivalence problem, and so on).

Definition 2 (Naive Derivative). *We define E_α and D_w. They are maps from a* **CKAT** *term to a set of* **CKAT** *terms, respectively. E_α is inductively defined as follows. We expand E_α and D_w to $\overline{E}_\alpha(P) = \bigcup_{p \in P} E_\alpha(p)$ and $\overline{D}_w(P) = \bigcup_{p \in P} D_w(p)$, where P is a set of* **CKAT** *terms, respectively.*

- $E_\alpha(\mathbf{0}) = E_\alpha(\mathbf{p}) = \emptyset$
- $E_\alpha(\mathbf{1}) = E_\alpha(p_1^*) = \{\mathbf{1}\}$
- $E_\alpha(\mathbf{t}) = \begin{cases} \{\mathbf{1}\} & (\mathbf{t} \in \alpha) \\ \emptyset & (o.w.) \end{cases}$
- $E_\alpha(\bar{b}) = \{\mathbf{1}\} \setminus E_\alpha(b)$
- $E_\alpha(p_1 + p_2) = E_\alpha(p_1) \cup E_\alpha(p_2)$
- $E_\alpha(p_1 p_2) = E_\alpha(p_1 \parallel p_2) = E_\alpha(p_1) E_\alpha(p_2)$

D_w is inductively defined as follows.
For $w = \mathbf{q} \mid \{|w'_1, w'_2|\}$ and any series-parallel string w',

- $D_{\alpha w \alpha' w' \alpha''}(p) = \overline{D}_{\alpha' w' \alpha''}(D_{\alpha w \alpha'}(p))$
- $D_{\alpha w \alpha'}(p_1 + p_2) = D_{\alpha w \alpha'}(p_1) \cup D_{\alpha w \alpha'}(p_2)$
- $D_{\alpha w \alpha'}(p_1 p_2) = D_{\alpha w \alpha'}(p_1)\{p_2\} \cup E_\alpha(p_1) D_{\alpha w \alpha'}(p_2)$

- $D_{\alpha w \alpha'}(p_1^*) = D_{\alpha w \alpha'}(p_1)\{p_1^*\}$
- $D_{\alpha w \alpha'}(b) = \emptyset$ *for any boolean term* b
- $D_{\alpha \mathbf{q} \alpha'}(\mathbf{p}) = \begin{cases} \{\mathbf{1}\} & (\mathbf{p} = \mathbf{q}) \\ \emptyset & (o.w.) \end{cases}$
- $D_{\alpha \mathbf{q} \alpha'}(p_1 \parallel p_2) = \emptyset$
- $D_{\alpha \{|w_1, w_2|\} \alpha'}(\mathbf{p}) = \emptyset$
- $D_{\alpha \{|w_1, w_2|\} \alpha'}(p_1 \parallel p_2) = E_{\alpha'}((D_{\alpha w_1 \alpha'}(p_1) \parallel D_{\alpha w_2 \alpha'}(p_2)) \cup (D_{\alpha w_1 \alpha'}(p_2) \parallel D_{\alpha w_2 \alpha'}(p_1)))$

The *left-quotient* of $L \subseteq GS_{\Sigma,T}$ with regard to $w \in GS_{\Sigma,T}$ is the set $w^{-1}L = \{w' \mid w \diamond w' \in L\}$.

Lemma 1. *For any series-parallel string* $\alpha w \alpha'$*,*

1. $\mathbf{1} \in E_\alpha(p) \iff \alpha \in L_\alpha(p)$
2. $(\alpha w \alpha')^{-1} L_\alpha(p) = \overline{L}_{\alpha'}(D_{\alpha w \alpha'}(p))$

Proof (Sketch). 1. is proved by induction on the size of p*.*
2. is proved by double induction on the size of w *and the size of* p*.*

We can decide whether $\alpha w \alpha' \in L(p)$ to check $\mathbf{1} \in \overline{E}_{\alpha'}(D_{\alpha w \alpha'}(p))$ by Lemma 1. We now define *efficient derivative*. This derivative is another definition of derivative for **CKAT**. This derivative is useful for giving more efficient algorithm than naive derivative in computational complexity. (In naive derivative, we should memorize w_1 and w_2 to get $D_{\alpha\{|w_1,w_2|\}\alpha'}(p)$. In particular, the size of w_1 and w_2 can be double exponential size of input size in equivalence problem.) We expand **CKAT** terms to express efficient derivative. We say these terms *intermediate* **CKAT** *terms*. *Intermediate* **CKAT** *term* is defined as following.

Definition 3 (intermediate CKAT term). Intermediate **CKAT** term *is defined by the following grammar.*

$$q := b \mid \mathbf{p} \in \Sigma \mid q_1 + q_2 \mid q_1 q_2 \mid q_1^* \mid q_1 \parallel q_2 \mid D_x(q_1)$$

We call x a *derivative variable* of $D_x(q_1)$.

The *efficient derivative* $d_{pr}(q)$ is defined in Definition 4, where q is an intermediate **CKAT** term, pr is a sequence of assignments formed $x \mathrel{+}= \alpha\mathbf{p}$ or $x \mathrel{+}= \alpha\mathcal{T}$ (The sequence of assignments pr is formed $x_1 \mathrel{+}= term_1; \ldots; x_m \mathrel{+}= term_m$.) and $\mathcal{T}$ is formed by the following grammar. $\mathcal{T} := \{|x_l\mathcal{T}_l, x_r\mathcal{T}_r|\} \mid \{|x_l\mathcal{T}_l, \mathbf{p}_r x_r|\} \mid \{|\mathbf{p}_l x_l, x_r\mathcal{T}_r|\} \mid \{|\mathbf{p}_l x_l, \mathbf{p}_r x_r|\}$. Intuitively, $d_{x+=\alpha w}(\ldots D_x(q) \ldots)$ means $(\ldots D_x(\breve{D}_{\alpha w}(\mathrm{join}_\alpha(q))) \ldots)$.

Definition 4. *The* efficient derivative $d_{pr}(q)$ *is inductively defined as follows, where we assume that any derivative variable occurred in* $\mathcal{T}$ *are different. To define* $d_{pr}(q)$*, we also define* $\breve{D}_{\alpha w}$ *and* $join_\alpha$*. We expand* d_{pr} *to* $\overline{d}_{pr}(Q) = \bigcup_{q \in Q} d_{pr}(q)$*, where* Q *is a set of intermediate* **CKAT** *terms. We also expand* $join_\alpha$ *to* $\overline{join}_\alpha(Q) = \bigcup_{q \in Q} \overline{join}_\alpha(q)$*.*

- $d_{x+=\alpha w;pr'}(q) = \overline{d}_{pr'}(d_{x+=\alpha w}(q))$
- $d_{x+=\alpha w}(b) = \{b\}$
- $d_{x+=\alpha w}(\mathbf{p}) = \{\mathbf{p}\}$
- $d_{x+=\alpha w}(q_1 + q_2) = d_{x+=\alpha w}(q_1) \cup d_{x+=\alpha w}(q_2)$
- $d_{x+=\alpha w}(q_1 q_2) = d_{x+=\alpha w}(q_1) d_{x+=\alpha w}(q_2)$
- $d_{x+=\alpha w}(q_1^*) = d_{x+=\alpha w}(q_1)^*$
  $(= \{q'^* \mid q' \in d_{x:=\alpha w}(q_1)\})$
- $d_{x+=\alpha w}(q_1 \parallel q_2) = d_{x+=\alpha w}(q_1) \parallel d_{x+=\alpha w}(q_2)$
  $(= \{q'_1 \parallel q'_2 \mid q'_1 \in d_{x+=\alpha w}(q_1), q'_2 \in d_{x+=\alpha w}(q_2)\})$
- $d_{x+=\alpha w}(D_y(q_1)) = \overline{D}_y(d_{x+=\alpha w}(q_1))$
- $d_{x+=\alpha w}(D_x(q_1)) = \overline{D}_x(\breve{D}_{\alpha w}(\overline{join}_\alpha(q_1)))$
- $\breve{D}_{\alpha \mathbf{p}}(q) = D_{\alpha \mathbf{p}}(q)$
- $\breve{D}_{\alpha \mathcal{T}}(b) = \breve{D}_{\alpha \mathcal{T}}(\mathbf{p}) = \emptyset$
- $\breve{D}_{\alpha \mathcal{T}}(q_1 + q_2) = \breve{D}_{\alpha \mathcal{T}}(q_1) \cup \breve{D}_{\alpha \mathcal{T}}(q_2)$
- $\breve{D}_{\alpha \mathcal{T}}(q_1 q_2) = \breve{D}_{\alpha \mathcal{T}}(q_1)\{q_2\} \cup E_\alpha(q_1)\breve{D}_{\alpha \mathcal{T}}(q_2)$
- $\breve{D}_{\alpha \mathcal{T}}(q_1^*) = \breve{D}_{\alpha \mathcal{T}}(q_1)\{q_1^*\}$
- $\breve{D}_{\alpha \mathcal{T}}(q_1 \parallel q_2) = \begin{cases} (\overline{D}_{x_l}(\breve{D}_{\alpha \mathbf{p}_l}(q_1)) \parallel \overline{D}_{x_r}(\breve{D}_{\alpha \mathbf{p}_r}(q_2))) \\ \cup(\overline{D}_{x_r}(\breve{D}_{\alpha \mathbf{p}_r}(q_1)) \parallel \overline{D}_{x_l}(\breve{D}_{\alpha \mathbf{p}_l}(q_2))) & (\mathcal{T} = \{|\mathbf{p}_l x_l, \mathbf{p}_r x_r|\}) \\ (\overline{D}_{x_l}(\breve{D}_{\alpha \mathcal{T}_l}(q_1)) \parallel \overline{D}_{x_r}(\breve{D}_{\alpha \mathbf{p}_r}(q_2))) \\ \cup(\overline{D}_{x_r}(\breve{D}_{\alpha \mathbf{p}_r}(q_1)) \parallel \overline{D}_{x_l}(\breve{D}_{\alpha \mathcal{T}_l}(q_2))) & (\mathcal{T} = \{|\mathcal{T}_l x_l, \mathbf{p}_r x_r|\}) \\ (\overline{D}_{x_l}(\breve{D}_{\alpha \mathbf{p}_l}(q_1)) \parallel \overline{D}_{x_r}(\breve{D}_{\alpha \mathcal{T}_r}(q_2))) \\ \cup(\overline{D}_{x_r}(\breve{D}_{\alpha \mathcal{T}_r}(q_1)) \parallel \overline{D}_{x_l}(\breve{D}_{\alpha \mathbf{p}_l}(q_2))) & (\mathcal{T} = \{|\mathbf{p}_l x_l, \mathcal{T}_r x_r|\}) \\ (\overline{D}_{x_l}(\breve{D}_{\alpha \mathcal{T}_l}(q_1)) \parallel \overline{D}_{x_r}(\breve{D}_{\alpha \mathcal{T}_r}(q_2))) \\ \cup(\overline{D}_{x_r}(\breve{D}_{\alpha \mathcal{T}_r}(q_1)) \parallel \overline{D}_{x_l}(\breve{D}_{\alpha \mathcal{T}_l}(q_2))) & (\mathcal{T} = \{|\mathcal{T}_l x_l, \mathcal{T}_r x_r|\}) \end{cases}$
- $join_\alpha(b) = \{b\}$, $join_\alpha(\mathbf{p}) = \{\mathbf{p}\}$
- $join_\alpha(q_1 + q_2) = join_\alpha(q_1) \cup join_\alpha(q_2)$, $join_\alpha(q_1 q_2) = join_\alpha(q_1) join_\alpha(q_2)$
- $join_\alpha(q_1 \parallel q_2) = join_\alpha(q_1) \parallel join_\alpha(q_2)$
- $join_\alpha(q_1^*) = join_\alpha(q_1)^*$
- $join_\alpha(D_y(q)) = \overline{E}_\alpha(join_\alpha(q))$

Efficient derivative is essentially equal to the derivative of Definition 1. Let $sp_x(pr)$ be the string corresponded to x of pr. (For example, $sp_{x_0}(x_0 += \alpha\{\mathbf{p}_1 x_1, \mathbf{p}_2 x_2\}; x_1 += \alpha'\mathbf{p}_3; x_0 += \alpha''\mathbf{p}_4) = \alpha\{\mathbf{p}_1 \alpha' \mathbf{p}_3, \mathbf{p}_2\}\alpha''\mathbf{p}_4$. $sp_{x_1}(x_0 += \alpha\{\mathbf{p}_1 x_1, \mathbf{p}_2 x_2\}; x_1 += \alpha'\mathbf{p}_3; x_0 += \alpha''\mathbf{p}_4) = \alpha \mathbf{p}_1 \alpha' \mathbf{p}_3$)

Lemma 2. $\overline{join}_{\alpha'}(\overline{d}_{pr}(D_x(p))) = \overline{E}_{\alpha'}(D_{sp_x(pr)\alpha'}(p))$

By Lemma 1 and Lemma 2, $sp_x(pr)\alpha' \in L(p) \iff \mathbf{1} \in \overline{\mathrm{join}}_{\alpha'}(\overline{d}_{pr}(D_x(p)))$. Therefore, we can use effective derivative instead of naive derivative.

Next, we define the *size* of a intermediate **CKAT** term q, denoted by $|q|$ as follows.

- $|\mathbf{0}| = |\mathbf{1}| = |\mathbf{t}| = |\mathbf{p}| = 1$
- $|\overline{b}| = 1 + |b|$
- $|q_1^*| = |D_x(q_1)| = 1 + |q_1|$

- $|q_1 + q_2| = |q_1 q_2| = |q_1 \parallel q_2| = 1 + |q_1| + |q_2|$

Definition 5 (Closure). *Cl_X is a map from a intermediate* **CKAT** *term to a set of intermediate* **CKAT** *terms, where X is a set of intersection variables. Cl_X is inductively defined as follows.*

- $Cl_X(a) = \{a\}$ *for* $a = \mathbf{0} \mid \mathbf{1} \mid \mathbf{t}$
- $Cl_X(\overline{b}) = \{\overline{b}\} \cup Cl_X(b)$ *for any boolean term* b
- $Cl_X(\mathbf{p}) = \{\mathbf{p}, \mathbf{1}\}$
- $Cl_X(q_1 + q_2) = \{q_1 + q_2\} \cup Cl_X(q_1) \cup Cl_X(q_2)$
- $Cl_X(q_1 q_2) = \{q_1 q_2\} \cup Cl_X(q_1)\{q_2\} \cup Cl_X(q_2)$
- $Cl_X(q_1^*) = \{q_1^*\} \cup Cl_X(q_1)\{q_1^*\}$
- $Cl_X(q_1 \parallel q_2) = \{q_1 \parallel q_2\} \cup \{D_{x_1}(q_1') \parallel D_{x_2}(q_2') \mid q_1' \in Cl_X(q_1), q_2' \in Cl_X(q_2), x_1, x_2 \in X\}$
- $Cl_X(D_x(q_1)) = \{D_x(q_1)\} \cup \overline{D}_x(Cl_X(q_1))$

We expand Cl_X to $\overline{Cl}_X(Q) = \bigcup_{q \in Q} Cl_X(q)$, where Q is a set of intermediate **CKAT** terms. $\overline{Cl}_X$ is a closed operator. In other words, $\overline{Cl}_X$ satisfies (1) $Q \subseteq \overline{Cl}_X(Q)$, (2) $Q_1 \subseteq Q_2 \Rightarrow \overline{Cl}_X(Q_1) \subseteq \overline{Cl}_X(Q_2)$ and (3) $\overline{Cl}_X(\overline{Cl}_X(Q)) = \overline{Cl}_X(Q)$. We also define the intersection width $iw(q)$ over intermediate **CKAT** terms and $iw(w)$ over $GI_{\Sigma,T}$ as follows.

- $iw(b) = iw(\mathbf{p}) = 1$ for any boolean term b and any basic program $\mathbf{p} \in \Sigma$
- $iw(q_1 + q_2) = iw(q_1 q_2) = \max(iw(q_1), iw(q_2))$
- $iw(q_1^*) = iw(D_x(q_1)) = iw(q_1)$
- $iw(q_1 \parallel q_2) = 1 + iw(q_1) + iw(q_2)$
- $iw(\alpha) = 1$ for any $\alpha \subseteq T$
- $iw(\alpha_1 \mathbf{p} \alpha_2) = 1$
- $iw(w_1 \alpha w_2) = \max(iw(w_1 \alpha), iw(\alpha w_2))$
- $iw(\alpha_1 \{|w_1, w_2|\} \alpha_2) = 1 + iw(w_1) + iw(w_2)$

Lemma 3 (closure is bounded). *For any intermediate* **CKAT** *term q and any sequence of program pr and any set of derivative variables X, where X contains any derivative variables in pr,*

$|Cl_X(q)| \leq 2 * |X|^{2*iw(q)} * |q|^{iw(q)}$

Proof (Sketch). This is proved by induction on the structure of q. We only consider the case of $q = q_1 \parallel q_2$.

$$\begin{aligned}
|Cl_X(q_1 \parallel q_2)| &\leq 1 + |X| * |Cl_X(q_1)| * |X| * |Cl_X(q_2)| \\
&\leq 1 + |X|^2 * 2 * |q_1|^{iw(q_1)} * |X|^{2*iw(q_1)} * 2 * |q_2|^{iw(q_2)} * |X|^{2*iw(q_2)} \\
&= 1 + 4 * |X|^{2*iw(q_1 \parallel q_2)} * |q_1|^{iw(q_1)} * |q_2|^{iw(q_2)} \\
&\leq 2 * |X|^{2*iw(q_1 \parallel q_2)} * (|q_1| + |q_2|)^{iw(q_1)+iw(q_2)} \\
&\leq 2 * |X|^{2*iw(q_1 \parallel q_2)} * |q_1 \parallel q_2|^{iw(q_1 \parallel q_2)}
\end{aligned}$$

Lemma 4 (derivative is closed). *For any intermediate* **CKAT** *term q and any sequence of program pr and any set of derivative variables X, where X contains any derivative variables in pr,*

$d_{pr}(q) \subseteq Cl_X(q)$

Proof (Sketch). This is proved by double induction on the size of pr and the size of q.

3 CKAT equational theory is in EXPSPACE

By Lemma 1 and Lemma 2, $L(p_1) = L(p_2)$ iff $\overline{\text{join}}_{\alpha'}(\overline{d}_{pr}(D_x(p_1))) = \overline{\text{join}}_{\alpha'}(\overline{d}_{pr}(D_x(p_2)))$ for any pr and any α'. Thus we find some pr such that $\overline{\text{join}}_{\alpha'}(\overline{d}_{pr}(D_x(p_1))) \neq \overline{\text{join}}_{\alpha'}(\overline{d}_{pr}(D_x(p_2)))$ to decide $p_1 \neq p_2$. We must consider all the patterns of pr at first glance. But, we need not to check if pr is too long. We are enough to check the cases of $iw(sp(pr)) \leq \max(iw(p_1), iw(p_2))(\leq l)$ by the following Lemma 5.

Lemma 5. *If $iw(sp(pr)) > iw(q)$, $d_{pr}(q) = \emptyset$.*

By Lemma 5, we are enough to check the case of $iw(sp(pr)) \leq \max(iw(p_1), iw(p_2)) \leq l$. By $iw(sp(pr)) \leq l$, We are enough to prepare $1 + 3 * (l - 1)$ derivative variables. By Lemma 3, $|Cl_X(q)| \leq 2 * |q|^{iw(q)} * |X|^{2*iw(q)} \leq 2 * l^l * (1 + 3 * (l-1))^{2*l}$. Therefore, $|Cl_X(D_x(p_1))| = O(2^{p(l)})$ and $|Cl_X(D_x(p_2))| = O(2^{p(l)})$, where $p(l)$ is a polynomial function of l.

We can give a nondeterministic algorithm. We nondeterministically select the syntax of pr. (pr is $x \mathrel{+}= \alpha\mathbf{p}$ or $x \mathrel{+}= \alpha\mathcal{T}$.) If there exists a sequent of assignments pr and α' such that $\overline{\text{join}}_{\alpha'}(\overline{d}_{pr}(D_x(p_1))) \neq \overline{\text{join}}_{\alpha'}(\overline{d}_{pr}(D_x(p_2)))$, $p_1 \neq p_2$. Otherwise, $p_1 = p_2$. (See Algorithm 1 if you know more details.)

It holds the Theorem 1 by this algorithm.

Theorem 1. **CKAT** *equivalence problem is in EXPSPACE.*

Corollary 1. *if $iw(p)$ is a fixed parameter, then* **CKAT** *equivalence problem is PSPACE-complete.*

Note that PSPACE-hardness is derived by [Hun73].

4 Concluding Remarks

We have given the derivative for **CKAT** and shown that **CKAT** equational theory is in EXPSPACE. We finish with the following some of our future works.

- Is this equivalence problem EXPSPACE-complete? (We expect that this claim is $True$.)
- If we allow ϵ (for example, $\alpha\{|\mathbf{p}, \epsilon|\}\alpha$), can we give efficient derivative? (It become a little difficult because we have to memorize α in the case of $x \mathrel{+}= \alpha\{|\mathbf{p}_1 x_1, \epsilon|\}$. We should give another derivative to show the result like Corollary 1.)

A Pseudo Code

Algorithm 1 Decide $p_1 = p_2$, given two CKAT terms p_1 and p_2

Ensure: Whether $p_1 \neq p_2$ or not?(True or False)
  $step \Leftarrow 0$, $P_1 \Leftarrow \{D_{x_0}(p_1)\}$, $P_2 \Leftarrow \{D_{x_0}(p_2)\}$
  while $step \leq 2^{|Cl_X(D_{x_0}(p_1))|} * 2^{|Cl_X(D_{x_0}(p_2))|}$ **do**
    Let α be a subset of T, which is picked up nondeterministically.
    if $\overline{\text{join}}_\alpha(P_1) \neq \overline{\text{join}}_\alpha(P_2)$ **then**
      return $True$
    end if
    Let pr be $x \mathrel{+}= \alpha\mathbf{p}$ or $x \mathrel{+}= \alpha\mathcal{T}$, which is picked up nondeterministically, where $iw(pr) \leq \max(iw(p_1), iw(p_2))$.
    $step \Leftarrow step + 1$, $P_1 \Leftarrow d_{pr}(P_1)$, $P_2 \Leftarrow d_{pr}(P_2)$
  end while
  return $False$

References

[ABM12] Ricardo Almeida, Sabine Broda, and Nelma Moreira. "Deciding KAT and Hoare Logic with Derivatives". In: *Proceedings Third International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2012, Napoli, Italy, September 6-8, 2012*. 2012, pp. 127–140.

[Brz64] Janusz A Brzozowski. "Derivatives of regular expressions". In: *Journal of the ACM (JACM)* 11.4 (1964), pp. 481–494.

[Hun73] Harry B Hunt III. "On the time and tape complexity of languages I". In: *Proceedings of the fifth annual ACM symposium on Theory of computing*. ACM. 1973, pp. 10–19.

[Jip14] Peter Jipsen. "Concurrent Kleene algebra with tests". In: *Relational and Algebraic Methods in Computer Science*. Springer, 2014, pp. 37–48.

[Koz08] Dexter Kozen. *On the Coalgebraic Theory of Kleene Algebra with Tests*. Tech. rep. `http://hdl.handle.net/1813/10173`. Computing and Information Science, Cornell University, Mar. 2008.