Reasoning about Term Rewriting in Kleene Categories with Converse

Toshinori Takai[*1][*2], Hitoshi Furusawa[*2] and Wolfram Kahl[*3]

[*1] CREST, Japan Science and Technology Corporation (JST)
[*2] CVS, National Institute of Advanced Industrial Science and Technology (AIST)
[*3] Department of Computing and Software, McMaster University

Reasoning about Term Rewriting in Kleene Categories with Converse

Toshinori Takai[1,2], Hitoshi Furusawa[2], and Wolfram Kahl[3]

[1] CREST, Japan Science and Technology Corporation (JST)
[2] CVS, National Institute of Advanced Industrial Science and Technology (AIST)
[3] Department of Computing and Software, McMaster University

Abstract. This paper shows that "root-only" rewrite relations with respect to term rewriting systems can be expressed using Kleene star operations in a gs-monoidal Kleene category with converse. In our framework, we can analyze some properties of term rewriting systems by computing rewrite descendants of tree languages. As an application, we consider an infinite state model-checking problem given by a term rewriting system and an LTL formula.

1 Introduction

Term rewriting systems (TRSs)[1] can be used for expressing models of infinite state model checking problems. For example, some cryptographic protocols can be modelled by TRSs[13]. In the term rewriting setting, computing images of tree languages (sets of terms) under a rewrite relation is an important issue for solving verification problem. Because in many cases the problem to show safety properties of systems can be reduced into the problem of tree language operations (e.g. intersection, complement and deciding emptiness) on images of tree languages. From this point of view, the class of *recognizable tree languages*[4] plays an important role and many authors have investigated a class of TRSs which *effectively preserve recognizability*. A TRS effectively preserves recognizability if for any recognizable tree language, the image of the language under the rewrite relation is always recognizable and computable. If a TRS effectively preserves recognizability, then an infinite model-checking problem given by the TRS and an invariant condition is decidable due to the properties of the class recognizable tree languages. However, it is undecidable whether a given TRS effectively preserves recognizability or not. Also it seems that known decidable subclasses are too restricted to apply practical verification problems. Moreover, those known decidable subclasses are defined and investigated in a syntactic way. In this paper, we try to deal with rewrite descendants (an image of a tree language under a rewrite relation) in an algebraic way.

A *strict gs-monoidal category*[5, 6] is introduced for dealing with term graphs in an algebraic way. For our purpose, i.e. to express term rewriting, term graphs are not needed. However, we will consider a "converse" of a function symbol, which is intended to be interpreted as the inverse relation of the function. Corradini and Gadducci[7] showed that, in case when an operator symbol in a signature

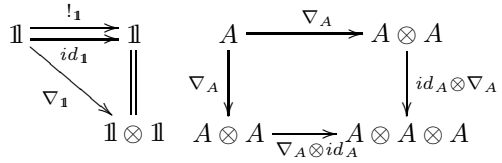
may be interpreted as not only a function but also a relation, the corresponding algebraic theory is a strict gs-monoidal category. For a given signature, we can construct a strict gs-monoidal category such that each arrow corresponds to a tuple of terms. On the other hand, a *Kleene category*[11] is essentially a typed Kleene algebra [14]. Considering a strict gs-monoidal Kleene category, we can regard an arrow with Kleene star as terms obtained by adding contexts for finitely many times. Moreover, a *category with converse*[11] can be considered as that each homset has a "subtraction" structure. Adding the converse structure to a strict gs-monoidal Kleene category, we can express "root-only" rewrite relation using converses and Kleene stars.

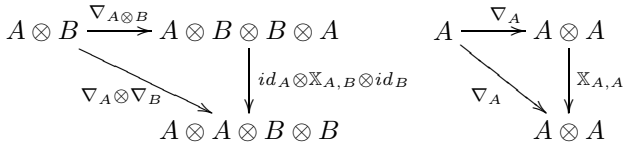
One application is an infinite model-checking for LTL formulas. Esparza et al.[8, 9] showed that LTL model-checking for push-down systems is decidable and Nitta et al.[15] extended the result to TRSs. Along the line of their studies, in this paper, we show that the problem can be reduced into the problem of deciding an order on a homset of strict gs-monoidal Kleene category with converse.

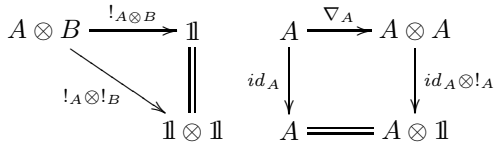
Here are some basic notations and notions on term rewriting. Let $\Sigma = (\underline{S}, \underline{F})$ be a signature where $\underline{S}$ be a finite set of *sorts* and $\underline{F}$ be a finite set of *function symbols* with their sorts. The set of terms constructed from a signature Σ and variables X is denoted as $\mathsf{T}_\Sigma(X)$. We write T_Σ for $\mathsf{T}_\Sigma(\emptyset)$. A *rewrite rule* is an ordered pair (l, r) of terms such that any variable in r also appears in l and denoted by $l \Rightarrow r$. A TRS is a finite set of rewrite rules and a rewrite relation of a TRS $\mathbb{R}$, denoted as $\Rightarrow_{\mathbb{R}}$ is the smallest relation which includes $\mathbb{R}$ and is closed under both contexts and substitutions. A tree language is a set of terms and for a tree language L and a TRS $\mathbb{R}$, the rewrite descendants of L by $\mathbb{R}$ is the image of L under the reflexive and transitive closure of $\Rightarrow_{\mathbb{R}}$, i.e. $(\Rightarrow_{\mathbb{R}}^*)(L)$.

2 GS-Monoidal Kleene Category with Converse

In this section, we first review strict gs-monoidal categories[5] in order to deal with terms. A *strict gs-monoidal category* $\mathbb{C}$ is a tuple $(\mathscr{C}_0, \otimes, \mathbb{1}, \mathbb{X}, \nabla, !)$ where (1) $(\mathscr{C}_0, \otimes, \mathbb{1}, \mathbb{X})$ is a strict symmetric monoidal category with monoidal operator $\otimes\colon \mathscr{C}_0 \times \mathscr{C}_0 \to \mathscr{C}_0$, unit $\mathbb{1} \in \mathrm{ob}\,\mathscr{C}_0$ and symmetric operator $\mathbb{X}$ and (2) $!$ and ∇ are collections of arrows such that for any object A, $!_A\colon A \to \mathbb{1}$ and $\nabla_A\colon A \to A \otimes A$ are arrows and the following diagrams commute:







Intuitively, an object S corresponds to the set of terms of sort S and an object $S_1 \otimes \cdots \otimes S_n$ corresponds to n-tuples of terms with sorts $S_1, \ldots, S_n$. For an object A of the form $A = A_1 \otimes \cdots \otimes A_n$ in $\mathscr{C}$, we write $\mathsf{elm}(A)$ for the set $\{A_1, \ldots, A_n\}$.

Proposition 1 *For two objects A and B of a strict gs-monoidal category, if $\mathsf{elm}(B) \subseteq \mathsf{elm}(A)$, then there is an arrow $A \to B$ consisting of $\mathbb{X}, \nabla$ and $!$.* □

We omit the proof which can be done by induction on the number of $|\mathsf{elm}(A)|$.

Next we introduce Kleene category[11, 12] with converse to deal with term rewriting. An ordered category $\mathscr{C}$ is a *Kleene category with converse* if for all $A, B \in \mathrm{ob}\,\mathscr{C}$, $f\colon A \to B$ and $r\colon A \to A$, (1) $\mathscr{C}(A, B)$ is an upper semilattice, (2) $\mathscr{C}(A, B)$ has a least element $\bot_{A,B}$ and $\bot_{A,B}$ is a left- and right-zero morphism for composition, (3) there is an arrow $f^\smile : B \to A$, called *converse of* f, such that for any $A, B, C \in \mathrm{ob}\,\mathscr{C}$ and any $f\colon A \to B$ and $g\colon B \to C$, the function $^\smile\colon \mathscr{C}(A, B) \to \mathscr{C}(B, A)$ is monotone and

$$(f^\smile)^\smile = f \qquad (f; g)^\smile = g^\smile; f^\smile$$

and (4) there is an arrow $r^*\colon A \to A$, called *Kleene star of* r, such that for any $r\colon A \to A$, $q\colon B \to A$, $s\colon A \to C$,

$$r^* = id_A \sqcup r \sqcup r^*; r^*$$
$$q; r \sqsubseteq q \text{ implies } q; r^* \sqsubseteq q \qquad s; r \sqsubseteq s \text{ implies } s; r^* \sqsubseteq s.$$

Remark that we can show that ${id_A}^\smile = id_A$ for any identity arrow id_A. Kleene category has been proposed by Y. Kinoshita[12] in 2001 and independently reintroduced by W. Kahl[11] in 2004. For an arrow $r\colon A \to B$ in a Kleene category with converse we say (1) r is *univalent* if $r^\smile; r \sqsubseteq id_B$, (2) r is *total* if $id_A \sqsubseteq r; r^\smile$, (3) r is *injective* if $r; r^\smile \sqsubseteq id_A$ and (4) r is a *mapping* if r is univalent and total. A *strict gs-monoidal Kleene category with converse* is a Kleene category with converse which is strict gs-monoidal.

3 Term rewriting by Monoidal Kleene Category with Converse

In this section, we will show how to represent term rewriting by monoidal Kleene category with converse. Let **GSM-Cat** be the category of strict gs-monoidal categories with gs-monoidal functors[6] and $\mathbf{GSM_c\text{-}Cat}$ be the category of strict gs-monoidal categories with converse with structure preserving functors. The forgetful functor $\mathcal{U}_1\colon \mathbf{GSM_c\text{-}Cat} \to \mathbf{GSM\text{-}Cat}$ has a left adjoint $\mathcal{F}_1$, which is an obvious free construction.

Let us consider a pair $(\mathscr{C}, R)$ of a strict gs-monoidal category $\mathscr{C}$ and R be a set of ordered pairs of parallel arrows (p, q), i.e. $\mathsf{source}(p) = \mathsf{source}(q)$ and $\mathsf{target}(p) = \mathsf{target}(q)$. The collection of such pairs forms a category, say **GSM-Comp**, where an arrow $(\mathscr{C}, R) \to (\mathscr{C}', R')$ is a pair of gs-monoidal functor $\mathcal{H}\colon \mathscr{C} \to \mathscr{C}'$ and a mapping $h\colon R \to R'$ such that for any $(p, q) \in R$ with $p, q\colon A \to B$, $(h)(p, q)$ is a pair of arrows of $\mathcal{H}A \to \mathcal{H}B$. Let $\mathbf{GSK_c\text{-}Cat}$ be the category of strict gs-monoidal Kleene category with converse and structure preserving functors. Then, the forgetful functor $\mathcal{U}_2\colon \mathbf{GSK_c\text{-}Cat} \to \mathbf{GSM\text{-}Comp}$ has a left adjoint $\mathcal{F}_2$ because $\mathbf{GSK_c\text{-}Cat}$ is obtained by adding operators $\sqcup$, *, and $\bot$ to **GSM-Comp** and the structure can be axiomatized by equations and Horn clauses[2].

Let $\Sigma = (\underline{S}, \underline{F})$ be a signature where $\underline{S}$ is a set of sorts and $\underline{F}$ is a set of function symbols. Using the free constructions $\mathcal{F}_1$ and $\mathcal{F}_2$ defined above with the notion of *free gs-monoidal theory* $\mathbf{GS\text{-}Th}(\Sigma)$ *of a signature* Σ[6], we define *free term category* $\mathscr{C}_\Sigma$ *generated by a signature* Σ as $(\mathcal{F}_2)(\mathcal{F}_1\mathbf{GS\text{-}Th}(\Sigma), R)$ where

$$
\begin{aligned}
R = \{&(f; f^\smile, id_{S_1 \otimes \cdots \otimes S_n}), (id_{S_1 \otimes \cdots \otimes S_n}, f; f^\smile), (f^\smile; f, id_S) \\
&\mid f \in \underline{F} \text{ with } f\colon S_1 \times \cdots \times S_n \to S\} \cup \\
&\{(f; g^\smile, \bot_{S_1 \otimes \cdots \otimes S_n, S'_1 \otimes \cdots \otimes S'_m}), (\bot_{S_1 \otimes \cdots \otimes S_n, S'_1 \otimes \cdots \otimes S'_m}, f; g^\smile) \\
&\mid f, g \in \underline{F} \text{ with } f\colon S_1 \times \cdots \times S_n \to S \text{ and } g\colon S'_1 \times \cdots \times S'_m \to S\}.
\end{aligned}
$$

The set R is used to say that each arrow corresponding to a function symbol in Σ is an injective mapping and for every two arrows corresponding two different function symbols, there ranges are disjoint. Since $\mathbf{GS\text{-}Th}(\Sigma)$ has no converse arrow, we can see that any two different arrow f and g in $\mathbf{GS\text{-}Th}(\Sigma)$ will not be collapsed into one arrow by the set R defined above; this means that we can distinguish the two arrows in $(\mathcal{F}_2)(\mathcal{F}_1\mathbf{GS\text{-}Th}(\Sigma), R)$ which correspond to f and g.

Example 1 *Let* $\Sigma = (\{S\}, \{f, g, a, b\})$ *with* $f\colon S \times S \to S, g\colon S \to S, a\colon S$ *and* $b\colon S$. *For terms* $f(g(a), g(b))$, $f(g(x), g(y))$ *and* $f(g(x), g(x))$, *the arrows* $((a; g) \otimes (b; g)); f\colon \mathbb{1} \to S$, $(g \otimes g); f\colon S \otimes S \to S$ *and* $\nabla_S; (g \otimes g); f\colon S \to S$ *are respectively associated.*

Note that ground terms of sort S are arrows of $\mathbb{1} \to S$. For the free term category $\mathscr{C}_\Sigma$ generated by Σ, we write $\overline{t}$ for the corresponding arrow in $\mathscr{C}$ of a term t or a set t of terms and define $\mathsf{var\_list} : \mathsf{T}_\Sigma(X) \to X^*$ as $\mathsf{var\_list}(x) = x$ for $x \in X$, $\mathsf{var\_list}(f(t_1, \ldots, t_n)) = \mathsf{var\_list}(t_1) \cdots \mathsf{var\_list}(t_n)$ and $\mathsf{to\_mon\_obj}\colon X^* \to \mathrm{ob}\,\mathscr{C}_\Sigma$ as a translation from a list $x_1 \cdots x_n$ of variables to the object of $\mathscr{C}_\Sigma$ which is a monoidal product $S_1 \otimes \cdots \otimes S_n$ of the list of sorts with $x_i\colon S_i$ for $1 \le i \le n$.

Definition 1 *For a (linear) rewrite rule* $l \Rightarrow r$ *constructed from a signature* Σ, *we associate with an arrow* $\overline{l \Rightarrow r}$ *in the term category* $\mathscr{C}$ *as* $\overline{l \Rightarrow r} = (\overline{l})^\smile; h; \overline{r}$ *where* $h\colon (\mathsf{to\_mon\_obj} \circ \mathsf{var\_list})(l) \to (\mathsf{to\_mon\_obj} \circ \mathsf{var\_list})(r)$ *is the arrow constructed in the proof of Proposition 1. Note that since* $l \Rightarrow r$ *is a rewrite rule, any variable in* r *also appears in* l. *For a (linear) TRS* $\mathbb{R}$, *we write* $\overline{\mathbb{R}}$ *for the arrow* $\bigsqcup\{\overline{l \to r} \mid l \to r \in \mathbb{R}\}$.

Intuitively, for an arrow $\overline{L}$ representing a set L of terms and a TRS $\mathbb{R}$, the arrow $\overline{L};(\overline{\mathbb{R}})^*$ corresponds to the set of terms which can be obtained by "root rewriting", defined below, from a term in L.

Definition 2 *For TRS $\mathbb{R}$, the* root rewrite relation*, written as $\Rightarrow_{\mathsf{root},\mathbb{R}}$, is the smallest relation which contains $\mathbb{R}$ and is closed under substitutions.*

Example 2 *Let $\mathbb{R}_1 = \{f(x,y) \Rightarrow f(g(x),g(y))\}$. Then, $\overline{\mathbb{R}_1} = f^\smile;(g \otimes g);f$. For a set $L = \{f(x,y)\}$, the image by the root rewrite relation by $\mathbb{R}_1$, i.e. $(\Rightarrow^*_{\mathsf{root},\mathbb{R}_1})(L)$ can be obtained as $\overline{L};\overline{\mathbb{R}_1}^* = f;(f^\smile;(g \otimes g);f)^* = (f;f^\smile;(g \otimes g))^*;f = (g \otimes g)^*;f$.*

Example 3 *Let $\mathbb{R}_2 = \{f(x,h(g(y))) \Rightarrow f(g(k(y)),h(x))\}$, then we have $\overline{\mathbb{R}_2} = (id_S \otimes g;h);f;\mathbb{X};(k;g \otimes h);f$. For a set $L = \{f(g(x),h(g(y)))\}$, the image by the root rewrite relation by $\mathbb{R}_2$, i.e. $(\Rightarrow^*_{\mathsf{root},\mathbb{R}_2})(L)$ can be obtained as follows:*

$$\begin{aligned}
\overline{\mathbb{R}_2} &= ((id_S \otimes g;h);f)^\smile;\mathbb{X};(k;g \otimes h);f \\
&= f^\smile;(id_S \otimes h^\smile;g^\smile);\mathbb{X};(k;g \otimes h);f \\
&= f^\smile;(id_S \otimes h^\smile;g^\smile);(h \otimes k;g);\mathbb{X};f \\
&= f^\smile;(h \otimes h^\smile;g^\smile;k;g);\mathbb{X};f
\end{aligned}$$

$$\begin{aligned}
\overline{L};\overline{\mathbb{R}_2}^* &= (g \otimes g;h);f;(f^\smile;(h \otimes h^\smile;g^\smile;k;g);\mathbb{X};f)^* \\
&= (g \otimes g);(id_S \otimes h);(f;f^\smile;(id_S \otimes h^\smile;g^\smile;k;g);\mathbb{X};(id_S \otimes h))^*;f \\
&= (g \otimes g);((id_S \otimes h);(id_S \otimes h^\smile;g^\smile;k;g);\mathbb{X})^*;(id_S \otimes h);f \\
&= (g \otimes g);((id_S \otimes h;h^\smile;g^\smile;k;g);\mathbb{X})^*;(id_S \otimes h);f \\
&= (g \otimes g);((id_S \otimes g^\smile;k;g);\mathbb{X})^*;(id_S \otimes h);f \\
&= (g \otimes g);((id_S \otimes g^\smile;k;g);\mathbb{X};(id_S \otimes g^\smile;k;g);\mathbb{X})^*;(id_S \sqcup \\
&\quad (id_S \otimes g^\smile;k;g);\mathbb{X});(id_S \otimes h);f \ (\textit{since } \forall f\colon A \to A.\ f^* = (f;f)^*;(id_S \sqcup f)) \\
&= (g \otimes g);(g^\smile;k;g \otimes g^\smile;k;g)^*;(id_S \sqcup (id_S \otimes g^\smile;k;g);\mathbb{X});(id_S \otimes h);f \\
&= (g \otimes g);((g^\smile;k \otimes g^\smile;k);(g \otimes g))^*;(id_S \sqcup (id_S \otimes g^\smile;k;g);\mathbb{X});(id_S \otimes h);f \\
&= ((g \otimes g);(g^\smile;k \otimes g^\smile;k))^*;(g \otimes g);(id_S \sqcup (id_S \otimes g^\smile;k;g);\mathbb{X});(id_S \otimes h);f \\
&= (g;g^\smile;k \otimes g;g^\smile;k)^*;((g \otimes g) \sqcup (g \otimes g;g^\smile;k;g);\mathbb{X});(id_S \otimes h);f \\
&= (k \otimes k)^*;((g \otimes g) \sqcup (g \otimes k;g);\mathbb{X});(id_S \otimes h);f \\
&= (k \otimes k)^*;((g \otimes g) \sqcup \mathbb{X};(k;g \otimes g));(id_S \otimes h);f \\
&= (k \otimes k)^*;((g \otimes g) \sqcup \mathbb{X};(k \otimes id_S);(g \otimes g));(id_S \otimes h);f \\
&= (k \otimes k)^*;(id_S \sqcup \mathbb{X};(k \otimes id_S));(g \otimes g;h);f
\end{aligned}$$

Remark that for TRSs $\mathbb{R}_1$ and $\mathbb{R}_2$ in the examples above, both $(\Rightarrow^*_{\mathsf{root},\mathbb{R}_1})(L) = (\Rightarrow^*_{\mathbb{R}_1})(L)$ and $(\Rightarrow^*_{\mathsf{root},\mathbb{R}_2})(L) = (\Rightarrow^*_{\mathbb{R}_2})(L)$ hold.

4 Application: Infinite LTL Model-Checking

In this section, we consider in our framework a verification problem whose system is given by a TRS and the property to be verified is given by an LTL formula. The verification problem can be seen as an infinite model checking problem; it is in general undecidable. The problem can be formalized as follows. Let Atom be a set of atomic proposition, $(S, \rightarrow_S, \nu)$ be a Kripke structure where S is a set of states, $\rightarrow_S \subseteq S \times S$ is a transition relation and $\nu \colon \mathsf{Atom} \to \mathcal{P}(S)$ is a valuation mapping. For an LTL formula ϕ and a state $t_0 \in S$, we write

$$(S, \rightarrow_S), t_0 \models^{\nu} \phi$$

for that ϕ is valid in $(S, \rightarrow_S, \nu)$ at t_0.

Esparza et al. showed that LTL model-checking for push-down systems, which can be regarded as a special class of TRSs, is decidable by reducing to finite number of membership problems of images of recognizable tree languages[8, 9] of the rewrite relation. Along the line of their works, Nitta et al. extended the class of systems which can be verified effectively to subclass of TRSs[15]. This paper follows the study by Nitta et al. and try to re-formalize the theorem by using our framework.

First, we show how recognizable tree languages can be dealt with in our setting. A *tree automaton*[4] is a tuple $(\Sigma, Q, Q_{final}, \Delta)$ where Q is the finite set of states in which each state is associated with one sort, $Q_{final} \subseteq Q$ is the set of final states and Δ is the set of transition rules of the form $f(q_1, \ldots, q_n) \Rightarrow q$ where $f \colon S_1 \times \cdots \times S_n \to S$, q has sort S and q_i has sort S_i for $1 \leq i \leq n$ or the form $q' \Rightarrow q$ where q' and q have the same sort. Let Σ be a signature and $\mathbb{A} = (\Sigma, Q, Q_{final}, \Delta)$ be a tree automaton. The *term category* $\mathscr{C}_{\Sigma,\mathbb{A}}$ *generated by* Σ *with* $\mathbb{A}$ is defined as $(\mathcal{F}_2)(\mathcal{F}_1\mathbf{GS\text{-}Th}(\Sigma \cup Q), R \cup R')$ where $\mathcal{F}$ and R are the same staffs as in the previous section, Q is regarded as a set of constant symbols and $R' = \{((q_1 \otimes \cdots \otimes q_n); f, q) \mid f(q_1, \ldots, q_n) \to q \in \Delta\} \cup \{(q', q) \mid q' \to q \in \Delta\}$. For tree automata $\mathbb{A}_1, \ldots, \mathbb{A}_n$, we write $\mathscr{C}_{\Sigma,\mathbb{A}_1,\ldots,\mathbb{A}_n}$ for the term category where all $\mathbb{A}_1, \ldots, \mathbb{A}_n$ are assumed.

Proposition 2 *A term t of sort S is accepted by a tree automaton $\mathbb{A}$ if and only if $\overline{t} \sqcap (\bigsqcup\{\overline{q} \mid$ q is a final state of $\mathbb{A}$ of sort $S\} \neq \bot_{\mathbb{1},S}$ in $\mathscr{C}_{\Sigma,\mathbb{A}}$.* □

The proposition above says that the arrow $\bigsqcup\{\overline{q} \mid q \in Q_{final}, q \text{ has sort } S\}$ corresponds to the set of terms of sort S accepted by the tree automaton $\mathbb{A}$.

Next, we recall the simplest version of a valuation mapping. Let Atom be a set of atomic propositions and $\mathbb{R}$ be a TRS. A *simple valuation* is a mapping $\mathsf{Atom} \to \mathcal{P}(\mathsf{T}_\Sigma)$ which is given by a mapping $\mu \colon \mathsf{Atom} \to \mathsf{T}_\Sigma(X)$ satisfying for any $\mathsf{p} \in \mathsf{Atom}$ and any $l \Rightarrow r \in \mathbb{R}$, (1) l is an instance of $\mu(\mathsf{p})$ or (2) l and $\mu(\mathsf{p})$ are not unifiable and defined as follows: each $\mathsf{p} \in \mathsf{Atom}$ is valued by the set of all ground instances of $\mu(\mathsf{p})$, i.e. $\mathsf{p} \mapsto \{\sigma(\mu(\mathsf{p})) \in \mathsf{T}_\Sigma \mid \sigma \text{ is a substitution}\}$. Nitta et al. showed that more complex valuations, which are called regular valuations, i.e. each atomic proposition is sent to a recognizable tree language, can be translated into the simple one.

Usually, a transition system for model-checking problems is assumed to be total, which means that there is no dead-lock state. For a TRS $\mathbb{R}$, we can consider a relation $\Rightarrow_{\mathbb{R},\mathsf{total}} = \Rightarrow_{\mathbb{R}} \cup \{(t,t) \mid t \in \mathsf{NF}_{\mathbb{R}}\}$ where $\mathsf{NF}_{\mathbb{R}}$ is the set of all normal forms with respect to $\mathbb{R}$. Note that for a TRS $\mathbb{R}$, we can effectively obtain $\mathbb{R}_{\mathsf{total}}$ such that $\Rightarrow_{\mathbb{R}_{\mathsf{total}}} = \Rightarrow_{\mathbb{R},\mathsf{total}}$, which is total. In the following, without loss of generality, we assume that TRSs are total.

Let $\mathbb{R}$ be a TRS on Σ and $\nu\colon \mathsf{Atom} \to \mathcal{P}(\mathsf{T}_\Sigma)$ be a valuation, then we can consider a Kripke structure $(\mathsf{T}_\Sigma, \Rightarrow_{\mathbb{R}}, \nu)$. Note that since we have assumed that a TRS is total, so is the Kripke structure. To solve a model checking problem given by a TRS, we construct another TRS from a given instance of the problem.

Definition 3 ([15]) *Let $\mathbb{R}$ be a TRS, ϕ be an LTL formula, $\nu\colon \mathsf{Atom} \to \mathcal{P}(\mathsf{T}_\Sigma)$ be a simple valuation defined by $\mu\colon \mathsf{Atom} \to \mathsf{T}_{\Sigma,\mathcal{V}}$ and $\mathbb{B}_\phi$ be a Büchi automaton corresponding to ϕ. The* Büchi TRS $\mathbb{R}^\nu_\phi$ for $\mathbb{R}$, ν and $\mathbb{B}_\phi$ *is a TRS on signature $\Sigma \cup Q$ where Q is the set of states of $\mathbb{B}_\phi$ and an element in Q is regarded as a unary function symbol and defined as follows: for any transition rule $q \overset{a}{\Rightarrow} q'$ of $\mathbb{B}_\phi$ and any rewrite rule $l \Rightarrow r$ of $\mathbb{R}$ such that $a \subseteq \{\mathsf{p} \in \mathsf{Atom} \mid\ l$ and $\mu(\mathsf{p})$ are unifiable$\}$, add a rewrite rule $q(l) \Rightarrow q'(r)$ to $\mathbb{R}^\nu_\phi$.*

Note that rewrite relation by a Büchi TRS always coincides with that of root rewrite relation. Nitta et al. characterized a class of TRSs in which the model-checking problem can be reduced into the finite number of problems to decide whether a term is contained in the image of certain recognizable tree language under rewrite relation of Büchi TRSs. A left-linear rewrite rule $l \Rightarrow r$ is *generalized growing*[15] if any two variables x and y in $\mathcal{V}ar(l) \cap \mathcal{V}ar(r)$ and positions $o_1, o_2 \in \mathcal{P}os(l)$ and $o_3, o_4 \in \mathcal{P}os(r)$ satisfy that $l/o_1 = x, l/o_2 = y, r/o_3 = x$ and $r/o_4 = y$, then $|o_1| - |o_1 \sqcup o_2| \leq |o_3| - |o_3 \sqcup o_4|$. A TRS consisting of generalized growing rewrite rules is called *generalized growing*.

Theorem 3 ([15]) *Let $(\mathsf{T}_\Sigma, \Rightarrow_{\mathbb{R}}, \nu)$ be a Kripke structure given by a generalized growing TRS $\mathbb{R}$ on a signature Σ and a simple valuation $\nu\colon \mathsf{Atom} \to \mathcal{P}(\mathsf{T}_\Sigma)$, $t_0 \in \mathsf{T}_\Sigma$ be a state, ϕ be an LTL formula and $\mathbb{B}_{\neg\phi} = (\Sigma, Q, \Delta, q_0, Q_{acc})$ be a Büchi automaton corresponding to $\neg\phi$. Then, $(\mathsf{T}_\Sigma, \Rightarrow_{\mathbb{R}}), t_0 \models^\nu \phi$ if and only if there are an integer c and a term $t_R \in \{q(t) \mid q \in Q, t \in \mathsf{T}_\Sigma(X)$ is linear$\}$ such that $|t_R| \leq c$, $(\Rightarrow^+_{\mathbb{R}^\nu_{\neg\phi}})((\Rightarrow^*_{\mathbb{R}^\nu_{\neg\phi}})(\{t_R\}) \cap L_{acc}) \cap L'_R \neq \emptyset$ and $L_R \cap (\Rightarrow^*_{\mathbb{R}^\nu_{\neg\phi}})(\{q_0(t_0)\}) \neq \emptyset$ where $\mathbb{R}^\nu_{\neg\phi}$ is the Büchi TRS constructed from $\mathbb{R}$, ν and $\mathbb{B}_{\neg\phi}$, $L_{acc} = \{q(t) \mid q \in Q_{acc}, t \in \mathsf{T}_\Sigma(X)$ is linear$\}$, $L_R = \{\theta(t_R) \mid \theta\colon X \to \mathsf{T}_\Sigma$ is a substitution$\}$ and $L'_R = \{\theta(t_R) \mid \theta\colon X \to \mathsf{T}_\Sigma(X)$ is a linear substitution$\}$.*

To formalize the theorem above in our setting, we first consider a Kleene category in which each homset has a lattice structure. It has been shown [10] that the categories of Kleene algebras and *action lattices*, which have lattice structure, are related by an adjunction. Therefore, the discussion of this paper can be applied to a freely generated strict gs-monoidal Kleene category with converse whose homsets have not only joins but also meets. For a ground term t and a partial function $\sigma\colon \mathcal{P}os(t) \to \mathsf{T}_\Sigma$, we write $\mathsf{replace}(t, \sigma)$ for a term which can be obtained by replacing subterm t/o with $\sigma(o)$ for $o \in \mathsf{dom}\ \sigma$ and for a constant c, we write $\mathcal{P}os_c(t)$ for the subset of $\mathcal{P}os(t)$ such that $t/o = c$ for $o \in \mathcal{P}os_c(t)$.

Fact 4 *Let* $(\mathsf{T}_\Sigma, \Rightarrow_\mathbb{R}, \nu)$ *be a Kripke structure given by a generalized growing TRS* $\mathbb{R}$ *on a signature* Σ *and a simple valuation* $\nu\colon \mathsf{Atom} \to \mathcal{P}(\mathsf{T}_\Sigma)$, $t_0 \in \mathsf{T}_\Sigma$ *be a state,* ϕ *be an LTL formula and* $\mathbb{B}_{\neg\phi} = (\Sigma, Q, \Delta, q_0, Q_{acc})$ *be a Büchi automaton corresponding to* $\neg\phi$. *Then,* $(\mathsf{T}_\Sigma, \Rightarrow_\mathbb{R}), t_0 \models^\nu \phi$ *if and only if there are an integer* n *and a term* $t_R \in \{q(t) \mid q \in Q, t \in \mathsf{T}_{\Sigma\cup\{c\}}\}$ *such that* $|t_R| \leq n$, $((\overline{t_R}; (\overline{\mathbb{R}^\nu_{\neg\phi}})^*) \sqcap \overline{q_{acc}}; (\overline{\mathbb{R}^\nu_{\neg\phi}})^+) \sqcap \overline{q'_R} \neq \bot$ *and* $\overline{q_R} \sqcap \overline{t_0}; \overline{q_0}; (\overline{\mathbb{R}^\nu_{\neg\phi}})^* \neq \bot$ *in* $\mathscr{C}_{\Sigma\cup\{c\},\mathbb{A}_R,\mathbb{A}'_R,\mathbb{A}_{acc}}$ *where* c *is a new constant,* $\mathbb{R}^\nu_{\neg\phi}$ *is the Büchi TRS constructed from* $\mathbb{R}$, ν *and* $\mathbb{B}_{\neg\phi}$, $\mathcal{L}(\mathbb{A}_{acc}) = \{q(t) \mid q \in Q_{acc}, t \in \mathsf{T}_{\Sigma\cup\{c\}}\}$, $\mathcal{L}(\mathbb{A}_R) = \{\mathsf{replace}(t_R, \sigma) \mid \sigma\colon \mathcal{P}os_c(t) \to \mathsf{T}_\Sigma\}$, $\mathcal{L}(\mathbb{A}'_R) = \{\mathsf{replace}(t_R, \sigma) \mid \sigma\colon \mathcal{P}os_c(t) \to \mathsf{T}_{\Sigma\cup\{c\}}\}$ *and* q_R, q'_R *and* q_{acc} *are the final states of* $\mathbb{A}_R$, $\mathbb{A}'_R$ *and* $\mathbb{A}_{acc}$, *respectively.*

5 Future work

In this paper, we express root only rewriting by the strict gs-monoidal Kleene category with converse and show infinite state model checking problem for LTL formulas can be described by our setting. To solve the model checking problem effectively, we have to investigate the decidability of an order on a homset of strict gs-monoidal Kleene category with converse. Moreover, we will consider how to deal with "usual" rewrite relations of term rewriting systems in our setting. Another direction is to compare the class of tree languages represented by arrows in the freely generated category $\mathscr{C}_\Sigma$ and the class of recognizable tree languages.

References

1. Baader, F. and Nipkow, T.: *Term Rewriting and All That*, Cambridge University Press, 1998.
2. Barr, M. and Wells, C.: *Toposes, Triples and Theories*, 2000. Available from `ftp://ftp.math.mcgill.ca/pub/barr/ttt.ps`
3. Bouajjani, A., Esparza, J. and Male, O.: Reachability Analysis of Pushdown Automata: Application to Model-Checking, *CONCUR97*, 1997.
4. Comon, H., Dauchet, M., Gilleron, R., Jacquemard, F., Lugiez, D., Tison, S. and Tommasi, M.: *Tree Automata Techniques and Applications*, draft, 1999.
5. Corradini, A. and Gadducci, F.: A 2-Categorical Presentation of Term Graph Rewriting, *Category Theory and Computer Science*, pp. 87–105, 1997.
6. Corradini, A. and Gadducci, F.: An Algebraic Presentation of Term Graphs via GsMonoidal Categories, *Applied Categorical Structures*, Vol. 7, pp. 299–321, 1999.
7. Corradini, A. and Gadducci, F.: Functorial Semantics for Multi-algebras, *Recent Trends in Algebraic Development Techniques*, LNCS, Vol. 1589, pp. 78–90, 1999.
8. Esparza, J., Hansel, D., Rossmanith, P. Schwoon, S.: Efficient Algorithms for Model Checking Pushdown Systems, *CAV2000, LNCS*, Vol. 1855, pp. 232–247, 2000.
9. Esparza, J., Kučera, A. and Schwoon, S.: Model-Checking LTL with Regular Valuations for Pushdown Systems, *TACS01, LNCS*, Vol. 2215, pp. 316–339, 2001.
10. Furusawa, H.: The Categories of Kleene Algebras, Action Algebras and Action Lattices are Related by Adjunctions, *Workshop of Kleene Algebra, RelMiCS, LNCS,* Vol. 3051, pp. 124–136, 2004.

11. Kahl, W.: Refactoring Heterogeneous Relation Algebras around Ordered Categories and Converse, *JoRMiCS*, Vol. 1, 2004. To appear.
12. 木下佳樹: 不動点をめぐる代数構造たち, 日本ソフトウェア科学会第 18 回大会 (2001 年度) 論文集, 日本ソフトウェア科学会, 2001.
13. Klay, F. and Genet, T.: Rewriting for Cryptographic Protocol Verification, *CADE00*, pp. 271–290, 2000.
14. Kozen, D.: Typed Kleene Algebra, *Computer Science Department, Cornell University,* No. 98-1669, 1998.
15. Nitta, N. and Seki, H.: An Extension of Pushdown System and Its Model Checking Method, *CONCUR03*, 2003,

逆演算付きクリーニ圏による項書換えの解析 (in English)
(算譜科学研究速報)
発行日：2004 年 12 月 10 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

Reasoning about Term Rewriting in Kleene Categories with Converse
(Programming Science Technical Report)
Dec 10, 2004
Research Center for Verification and Semantics
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