Ground Confluence and Strong Commutation Modulo Alpha-Equivalence in Nominal Rewriting

Kentaro Kikuchi(✉)

RIEC, Tohoku University, Sendai, Japan
kentaro.kikuchi@riec.tohoku.ac.jp

Abstract. Nominal rewriting was introduced as an extension of first-order term rewriting by a binding mechanism based on the nominal approach. A distinctive feature of nominal rewriting is that α-equivalence is not implicitly dealt with at the meta-level but explicitly dealt with at the object-level. In this paper, we introduce the notion of strong commutation modulo α-equivalence and give a sufficient condition for it. Using the condition, we present a new criterion for confluence modulo α-equivalence (on ground terms) of possibly non-terminating left-linear nominal rewriting systems.

1 Introduction

In languages with variable binding and variable names, α-equivalence needs to be treated. Usually α-equivalence is implicitly dealt with at the meta-level, but in the literature some authors seriously take it into account at the object-level (e.g. [6,20]). The nominal approach [5,13] is one of such studies, where variables that are possibly bound are called atoms. It deals with α-equivalence explicitly at the object-level, incorporating permutations and freshness constraints as basic ingredients.

Nominal rewriting [3,4] is a framework introduced as an extension of first-order term rewriting by a binding mechanism based on the nominal approach. It has a device to avoid accidental capture of free atoms on the way of rewriting, using the explicit α-equivalence and freshness constraints in rewrite rules.

Confluence is a fundamental property of rewriting systems that guarantees uniqueness of results of computation. Confluence of nominal rewriting systems has been discussed in [1,3,9,16,17]. Their aim is to provide confluence criteria for particular classes of nominal rewriting systems in the same way as discussed in the field of first-order term rewriting.

In the present paper, we study confluence of nominal rewriting systems that are defined by rewrite rules with atom-variables in the style of [10], where rewriting is performed only on ground nominal terms (so confluence properties discussed in this paper correspond to ground confluence etc. in terms of traditional nominal rewriting). In previous work [8], we have proved (ground) confluence

H. Seidl et al. (Eds.): ICTAC 2022, LNCS 13572, pp. 255–271, 2022.
https://doi.org/10.1007/978-3-031-17715-6_17

for this style of nominal rewriting systems whose rewrite rules have no overlaps which are computed using nominal unification with atom-variables [15]. In this paper, we present a sufficient condition for (ground) confluence of the same style of nominal rewriting systems possibly with overlaps of rewrite rules.

To discuss confluence in nominal rewriting, it is necessary to examine whether two terms can rewrite to α-equivalent terms. For doing this, we make use of suitable notions that are defined modulo an equivalence relation in terms of abstract reduction systems [11,12]. Such an approach was suggested in [20] (page 220). Precisely speaking, we present a sufficient condition for (ground) Church-Rosser modulo α-equivalence rather than confluence. The proof method using the sufficient condition can be seen as a generalisation of that for confluence of first-order term rewriting systems using the lemma of Hindley [6] and Rosen [14]. We will explain details of the methods in Sect. 3.

Contributions of the Paper. The contributions of the present paper are summarised as follows:

– We introduce the notion of strong commutation modulo α-equivalence and give a sufficient condition for it in left-linear uniform nominal rewriting systems. This notion has not been treated in [11,12] (in the case of a general equivalence relation $\sim$).
– Using the sufficient condition, we present a new criterion for Church-Rosser modulo α-equivalence (on ground nominal terms) of left-linear uniform nominal rewriting systems that are possibly non-terminating and may have overlaps of rewrite rules.

Organisation of the Paper. The present paper is organised as follows. In Sect. 2, we explain basic notions of nominal rewriting systems with atom-variables. In Sect. 3, we give a sufficient condition for strong commutation modulo α-equivalence, and use it to present a criterion for Church-Rosser modulo α-equivalence. In Sect. 4, we conclude with suggestions for further work.

2 Nominal Rewriting Systems with Atom-Variables

Nominal rewriting [3,4] is a framework that extends first-order term rewriting by a binding mechanism. In this section, we recall basic notions and notations concerning nominal rewriting systems with atom-variables [10]. For differences from the system of [3], see [8]. For simplicity, we treat a subset of the systems in [8,10].

2.1 Preliminaries

We fix a countably infinite set $\mathcal{X}$ of *variables* ranged over by $X, Y, \ldots$, a countably infinite set $\mathcal{A}$ of *atoms* ranged over by $a, b, \ldots$, and a countably infinite set $\mathcal{X}_A$ of *atom-variables* ranged over by $A, B, \ldots$. A *nominal signature* Σ is a set of *function symbols* ranged over by $f, g, \ldots$. Each $f \in \Sigma$ has a unique non-negative integer $arity(f)$. We assume that $\mathcal{X}$, $\mathcal{A}$, $\mathcal{X}_A$ and Σ are pairwise disjoint. Unless otherwise stated, different meta-variables for objects in $\mathcal{X}$, $\mathcal{A}$, $\mathcal{X}_A$ or Σ denote different objects.

The domain $dom(\phi)$ of a mapping $\phi : D \to E$ is defined as the set $\{d \in D \mid \phi(d) \neq d\}$ if $D \subseteq E$, and D otherwise. A mapping $\phi : D \to E$ is *finite* if its domain $dom(\phi)$ is a finite set.

Let $\bowtie$ be a binary relation. We write $\bowtie^=$ for the reflexive closure and $\bowtie^*$ for the reflexive transitive closure. If $\bowtie$ is written using $\to$, then the inverse $\bowtie^{-1}$ is written using $\leftarrow$. We use $\circ$ for the composition of two binary relations.

2.2 Ground Nominal Terms

In this subsection, we introduce the set of ground nominal terms, which we call NL_a following [8,10,15].

The set NL_a of *ground nominal terms*, or simply *ground terms*, is generated by the following grammar:

$$t, s ::= a \mid [a]t \mid f\langle t_1, \ldots, t_n\rangle$$

where $n = arity(f)$. Ground terms of the forms in the right-hand side are called, respectively, atoms, abstractions and function applications. We assume that function applications bind more strongly than abstractions. We abbreviate $f\langle\,\rangle$ as f, and refer to it as a *constant*. An abstraction $[a]t$ is intended to represent t with a bound. The set $FA(t)$ of *free* atoms occurring in t is defined as follows: $FA(a) = \{a\}$; $FA([a]t) = FA(t) \setminus \{a\}$; $FA(f\langle t_1, \ldots, t_n\rangle) = \bigcup_i FA(t_i)$.

Example 1. The nominal signature of the lambda calculus has two function symbols $\mathtt{lam}$ with $arity(\mathtt{lam}) = 1$, and $\mathtt{app}$ with $arity(\mathtt{app}) = 2$. The ground nominal term $\mathtt{app}\langle\mathtt{lam}\langle[a]\mathtt{lam}\langle[b]\mathtt{app}\langle b, a\rangle\rangle\rangle, b\rangle$ represents the lambda term $(\lambda a.\lambda b.ba)b$ in the usual notation. For this ground term t, we have $FA(t) = \{b\}$. $\square$

A *swapping* is a pair of atoms, written $(a\ b)$. *Permutations* π are bijections on $\mathcal{A}$ such that $dom(\pi)$ is finite. Permutations are represented by lists of swappings applied in the right-to-left order. For example, $((b\ c)(a\ b))(a) = c$, $((b\ c)(a\ b))(b) = a$, $((b\ c)(a\ b))(c) = b$. The permutation action $\pi{\cdot}t$, which operates on terms extending a permutation on atoms, is defined as follows: $\pi{\cdot}a = \pi(a)$; $\pi{\cdot}([a]t) = [\pi{\cdot}a](\pi{\cdot}t)$; $\pi{\cdot}(f\langle t_1, \ldots, t_n\rangle) = f\langle \pi{\cdot}t_1, \ldots, \pi{\cdot}t_n\rangle$.

Positions are finite sequences of positive integers. The empty sequence is denoted by ε. The set $Pos(t)$ of positions in a ground term t is defined as follows: $Pos(a) = \{\varepsilon\}$; $Pos([a]t) = \{1p \mid p \in Pos(t)\} \cup \{\varepsilon\}$; $Pos(f\langle t_1, \ldots, t_n\rangle) = \bigcup_i \{ip \mid p \in Pos(t_i)\} \cup \{\varepsilon\}$. The subterm of t at a position $p \in Pos(t)$ is written as $t|_p$. For positions p and q, we say that p is deeper than q if there exists a position o such that $p = qo$. In that case, o is denoted by $p \setminus q$.

A *context* is a ground term in which a distinguished constant $\square$ occurs. The ground term obtained from a context C by replacing each $\square$ at positions p_i by ground terms t_i is written as $C[t_1, \ldots, t_n]_{p_1,\ldots,p_n}$ or simply $C[t_1, \ldots, t_n]$.

$$\frac{}{\vdash_{NL_a} a\#b} \qquad \frac{\vdash_{NL_a} a\#t_1 \quad \cdots \quad \vdash_{NL_a} a\#t_n}{\vdash_{NL_a} a\#f\langle t_1,\ldots,t_n\rangle}$$

$$\frac{}{\vdash_{NL_a} a\#[a]t} \qquad \frac{\vdash_{NL_a} a\#t}{\vdash_{NL_a} a\#[b]t}$$

Fig. 1. Rules for freshness relations on NL_a

$$\frac{}{\vdash_{NL_a} a \approx_\alpha a} \qquad \frac{\vdash_{NL_a} t_1 \approx_\alpha s_1 \quad \cdots \quad \vdash_{NL_a} t_n \approx_\alpha s_n}{\vdash_{NL_a} f\langle t_1,\ldots,t_n\rangle \approx_\alpha f\langle s_1,\ldots,s_n\rangle}$$

$$\frac{\vdash_{NL_a} t \approx_\alpha s}{\vdash_{NL_a} [a]t \approx_\alpha [a]s} \qquad \frac{\vdash_{NL_a} (a\ b)\cdot t \approx_\alpha s \quad \vdash_{NL_a} b\#t}{\vdash_{NL_a} [a]t \approx_\alpha [b]s}$$

Fig. 2. Rules for α-equivalence on NL_a

A pair $a\#t$ of an atom a and a ground term t is called a *freshness relation*. The rules in Fig. 1 define the validity of freshness relations. Note that the defined $\vdash_{NL_a} a\#t$ coincides with $a \notin FA(t)$.

The rules in Fig. 2 define the relation $\vdash_{NL_a} t \approx_\alpha s$. This is a congruence relation [3] and coincides with usual α-equivalence (i.e. the relation reached by renamings of bound atoms) [5]. The bottom-right rule in the figure is about the case where the ground terms t and s are abstracted by different atoms. In $(a\ b)\cdot t$, the free occurrences of a in t are replaced by b which is fresh in t under the right premise of the rule. We often write $t \approx_\alpha s$ for $\vdash_{NL_a} t \approx_\alpha s$.

The following properties are shown in [3,19].

Proposition 1. *1.* $\vdash_{NL_a} a\#t$ *if and only if* $\vdash_{NL_a} \pi\cdot a\#\pi\cdot t$.
2. $\vdash_{NL_a} t \approx_\alpha s$ *if and only if* $\vdash_{NL_a} \pi\cdot t \approx_\alpha \pi\cdot s$.
3. If $\vdash_{NL_a} a\#t$ *and* $\vdash_{NL_a} t \approx_\alpha s$ *then* $\vdash_{NL_a} a\#s$.

2.3 Nominal Term Expressions

Next we introduce the set of term expressions, which we call NL_{AX}. Each rewrite rule is defined using them to represent a schema of rules.

The set NL_{AX} of *nominal term expressions*, or simply *term expressions*, is generated by the following grammar:

$$e ::= A \mid X \mid [A]e \mid f\langle e_1,\ldots,e_n\rangle$$

where $n = arity(f)$. We write $Var_{\mathcal{X}}(e)$ and $Var_{\mathcal{X}_A}(e)$ for the sets of variables and atom-variables occurring in a term expression e, respectively. Also, we write $Var_{\mathcal{X},\mathcal{X}_A}(e)$ for $Var_{\mathcal{X}}(e) \cup Var_{\mathcal{X}_A}(e)$. A term expression e is *linear* if each variable $X \in Var_{\mathcal{X}}(e)$ occurs only once in e.

The set $Pos(e)$ of positions in a term expression e is defined similarly to that for a ground term (using atom-variables for atoms) with the additional clause that $Pos(X) = \{\varepsilon\}$. The subexpression of e at a position $p \in Pos(e)$ is written as $e|_p$. A position $p \in Pos(e)$ is called a *variable position* if $e|_p$ is a variable, and a *non-variable position* otherwise.

A *ground substitution* is a finite mapping that assigns ground terms to variables and atoms to atom-variables. We use σ, δ for ground substitutions. We write $\sigma_{\mathcal{X}}$ and $\sigma_{\mathcal{X}_A}$ for ground substitutions obtained from σ by restricting the domain to $dom(\sigma) \cap \mathcal{X}$ and $dom(\sigma) \cap \mathcal{X}_A$, respectively. When $Var_{\mathcal{X},\mathcal{X}_A}(e) \subseteq dom(\sigma)$, the application of σ on e is written as $e\sigma$ and called a *ground instance* of e. The application of σ simply replaces the variables X and atom-variables A occurring in e by $\sigma(X)$ and $\sigma(A)$, respectively, without considering capture of free atoms. Then we have $e\sigma \in NL_a$ for every ground instance $e\sigma$.

A pair $A\#e$ of an atom-variable A and a term expression e is called a *freshness constraint*. A finite set of freshness constraints is called a *freshness context*. For a freshness context ∇, we define $Var_{\mathcal{X},\mathcal{X}_A}(\nabla) = \bigcup_{A\#e\in\nabla}(\{A\} \cup Var_{\mathcal{X},\mathcal{X}_A}(e))$ and $\nabla\sigma = \{A\sigma\#e\sigma \mid A\#e \in \nabla\}$.

2.4 Nominal Rewriting Systems with Atom-Variables

Next we define nominal rewrite rules and nominal rewriting systems with atom-variables.

Definition 1. A *nominal rewrite rule with atom-variables*, or simply *rewrite rule*, is a triple of a freshness context ∇ and term expressions $l, r \in NL_{AX}$ such that $Var_{\mathcal{X},\mathcal{X}_A}(\nabla) \cup Var_{\mathcal{X},\mathcal{X}_A}(r) \subseteq Var_{\mathcal{X},\mathcal{X}_A}(l)$ and l is not a variable. We write $\nabla \vdash l \to r$ for a rewrite rule, and identify rewrite rules modulo renaming of variables and atom-variables. A rewrite rule $\nabla \vdash l \to r$ is *left-linear* if l is linear.

Definition 2 (Nominal rewriting system with atom-variables). A *nominal rewriting system with atom-variables* (NRS_{AX} for short) is a finite set of rewrite rules. An NRS_{AX} is *left-linear* if so are all its rewrite rules.

The following example of an NRS_{AX} corresponds to the system in Example 43 of [3] written in the style of traditional nominal rewriting. Note that the freshness constraint $A\#B$ in the rule ($\mathsf{sub_{lam}}$) below is used to mean that the atom-variables A and B should be instantiated by distinct atoms.

Example 2. We extend the nominal signature in Example 1 by a function symbol $\mathtt{sub}$ with $arity(\mathtt{sub}) = 2$. By $\mathtt{sub}\langle[a]t, s\rangle$, we represent an explicit substitution $t\langle a := s\rangle$. Then, an NRS_{AX} to perform β-reduction is defined by the rule (Beta):

$$\vdash \mathtt{app}\langle\mathtt{lam}\langle[A]X\rangle, Y\rangle \to \mathtt{sub}\langle[A]X, Y\rangle \qquad (\mathsf{Beta})$$

together with an NRS_{AX} $\mathcal{R}_{\mathsf{sub}}$ to execute substitution:

$$\begin{array}{rcll}
 & \vdash & \mathtt{sub}\langle[A]\mathtt{app}\langle X, Y\rangle, Z\rangle \to \mathtt{app}\langle\mathtt{sub}\langle[A]X, Z\rangle, \mathtt{sub}\langle[A]Y, Z\rangle\rangle & (\mathsf{sub_{app}}) \\
 & \vdash & \mathtt{sub}\langle[A]A, X\rangle \to X & (\mathsf{sub_{var}}) \\
A\#X & \vdash & \mathtt{sub}\langle[A]X, Y\rangle \to X & (\mathsf{sub_{\epsilon}}) \\
A\#B, B\#Y & \vdash & \mathtt{sub}\langle[A]\mathtt{lam}\langle[B]X\rangle, Y\rangle \to \mathtt{lam}\langle[B]\mathtt{sub}\langle[A]X, Y\rangle\rangle & (\mathsf{sub_{lam}})
\end{array}$$

In a standard notation, the system $\mathcal{R}_{\mathsf{sub}}$ is represented as follows:

$$\begin{array}{rll} \vdash & (XY)\langle A := Z\rangle \to (X\langle A := Z\rangle)(Y\langle A := Z\rangle) & (\mathsf{sub_{app}}) \\ \vdash & A\langle A := X\rangle \to X & (\mathsf{sub_{var}}) \\ A\#X \vdash & X\langle A := Y\rangle \to X & (\mathsf{sub_{\epsilon}}) \\ A\#B, B\#Y \vdash & (\lambda B.X)\langle A := Y\rangle \to \lambda B.(X\langle A := Y\rangle) & (\mathsf{sub_{lam}}) \end{array}$$

□

In the sequel, $\vdash_{NL_a}$ is extended to mean to hold for all members of the set in the right-hand side.

Definition 3 (Rewrite relation). Let $R = \nabla \vdash l \to r$ be a rewrite rule. For ground terms $s, t \in NL_a$, the *rewrite relation* is defined by

$$s \to_{\langle R,p,\sigma\rangle} t \overset{\text{def}}{\iff} \vdash_{NL_a} \nabla\sigma,\ s = C[s']_p,\ \vdash_{NL_a} s' \approx_\alpha l\sigma,\ t = C[r\sigma]_p$$

Here the subterm s' of s is called the *R-redex*, or simply *redex* if R is understood. We write $s \xrightarrow{p}_R t$ if there exists σ such that $s \to_{\langle R,p,\sigma\rangle} t$. We write $s \to_R t$ if there exist p and σ such that $s \to_{\langle R,p,\sigma\rangle} t$. For an NRS_{AX} $\mathcal{R}$, we write $s \to_{\mathcal{R}} t$ if there exists $R \in \mathcal{R}$ such that $s \to_R t$.

An example of rewriting can be found in Example 4 of [8].

Lemma 1. *Let $R = \nabla \vdash l \to r$ be a rewrite rule, and let s, t be ground terms. If $p \in Pos(s)$ and $s \xrightarrow{p}_R t$ then $\pi{\cdot}s \xrightarrow{p}_R \pi{\cdot}t$ for every permutation π.*

Proof. This is proved in the same way as Lemma 2 of [8]. □

2.5 Overlaps

The notion of overlap is useful for analysing confluence properties of rewriting systems. In the setting of the present paper, it can be defined using nominal unification with atom-variables [15]. Here we restrict the language of unification problems to NL_{AX}.

Definition 4 (Variable-atom nominal unification problem). Let Γ be a finite set of equations of the form $e_1 \approx e_2$ where e_1 and e_2 are term expressions, and let ∇ be a freshness context. Then the pair (Γ, ∇) is called a *variable-atom nominal unification problem* (*VANUP* for short).

Definition 5 (Solution of a VANUP). A ground substitution σ is a *solution* of a VANUP (Γ, ∇) if $\vdash_{NL_a} e_1\sigma \approx_\alpha e_2\sigma$ for every equation $e_1 \approx e_2 \in \Gamma$ and $\vdash_{NL_a} A\sigma\#e\sigma$ for every freshness constraint $A\#e \in \nabla$. A VANUP (Γ, ∇) is *solvable* if there exists a solution of (Γ, ∇).

Example 3. Consider the nominal signature of the lambda calculus in Example 1, and let P be the VANUP $(\{\mathtt{lam}\langle[A]\mathtt{app}\langle X, A\rangle\rangle \approx \mathtt{lam}\langle[B]Y\rangle\}, \{A\#X\})$. Then, the ground substitution $[A := a, B := b, X := c, Y := \mathtt{app}\langle c, b\rangle]$ is a solution of P. □

Definition 6 (Overlap). Let $R_i = \nabla_i \vdash l_i \to r_i$ $(i = 1, 2)$ be rewrite rules. We assume without loss of generality that $Var_{\mathcal{X},\mathcal{X}_A}(l_1) \cap Var_{\mathcal{X},\mathcal{X}_A}(l_2) = \emptyset$. If the variable-atom nominal unification problem $(\{l_1 \approx l_2|_p\}, \nabla_1 \cup \nabla_2)$ is solvable for some non-variable position p of l_2, then we say that R_1 *overlaps* on R_2, and the situation is called an *overlap* of R_1 on R_2. If R_1 and R_2 are identical modulo renaming of variables and atom-variables, and $p = \varepsilon$, then the overlap is said to be *self-rooted*. An overlap that is not self-rooted is said to be *proper*.

Example 4. Let R_1 and R_2 be the rules (Eta) $A\#X \vdash \mathtt{lam}\langle[A]\mathtt{app}\langle X, A\rangle\rangle \to X$ and (Beta) $\vdash \mathtt{app}\langle\mathtt{lam}\langle[B]Y\rangle, Z\rangle \to \mathtt{sub}\langle[B]Y, Z\rangle$, respectively. Then, R_1 overlaps on R_2, since the VANUP $(\{\mathtt{lam}\langle[A]\mathtt{app}\langle X, A\rangle\rangle \approx \mathtt{app}\langle\mathtt{lam}\langle[B]Y\rangle, Z\rangle|_1 (= \mathtt{lam}\langle[B]Y\rangle)\}, \{A\#X\})$ is solvable as seen in Example 3. This overlap is proper. □

Example 5. There exists a self-rooted overlap of the rule (Beta) on its renamed variant, since the VANUP $(\{\mathtt{app}\langle\mathtt{lam}\langle[A]X\rangle, Y\rangle \approx \mathtt{app}\langle\mathtt{lam}\langle[B]Z\rangle, W\rangle\}, \emptyset)$ is solvable by taking the ground substitution $[A := a, B := b, X := a, Y := c, Z := b, W := c]$ as a solution. □

Unlike in first-order term rewriting, self-rooted overlaps need to be analysed in the case of nominal rewriting (cf. [1,16]). We check the cases corresponding to self-rooted overlaps too in the sufficient conditions given in the next section.

2.6 Parallel Reduction

A key notion for proving confluence of left-linear rewriting systems is parallel reduction. Here we define it inductively, using grammatical contexts [8,16].

Definition 7. The *grammatical contexts*, ranged over by G, are the contexts defined by

$$G ::= a \mid [a]\Box \mid f\langle\Box_1, \ldots, \Box_n\rangle$$

where $n = arity(f)$. For each rewrite rule R, the relation $\twoheadrightarrow\!\!\!\!\!\!+\,_R$ is defined inductively by the following rules:

$$\frac{s \xrightarrow{\varepsilon}_R t}{s \mathrel{+\!\!\!\!\twoheadrightarrow}_R t}\ (\mathsf{B}) \qquad \frac{s_1 \mathrel{+\!\!\!\!\twoheadrightarrow}_R t_1 \quad \cdots \quad s_n \mathrel{+\!\!\!\!\twoheadrightarrow}_R t_n}{G[s_1, \ldots, s_n] \mathrel{+\!\!\!\!\twoheadrightarrow}_R G[t_1, \ldots, t_n]}\ (\mathsf{C})$$

where $n\,(\geq 0)$ depends on the form of G.

The following properties of parallel reduction hold.

Lemma 2. *1.* $s \mathrel{+\!\!\!\!\twoheadrightarrow}_R s$.
2. If $s \mathrel{+\!\!\!\!\twoheadrightarrow}_R t$ *then* $C[s] \mathrel{+\!\!\!\!\twoheadrightarrow}_R C[t]$.
3. If $s \to_{\langle R,p,\sigma\rangle} t$ *then* $s \mathrel{+\!\!\!\!\twoheadrightarrow}_R t$.
4. If $s \mathrel{+\!\!\!\!\twoheadrightarrow}_R t$ *then* $s \to^*_R t$.

Proof. 1. By induction on the structure of s.
2. By induction on the context C.
3. By 2 and the rule (B).
4. By induction on the derivation of $s \mathrel{+\!\!\!\!\twoheadrightarrow}_R t$. □

3 Confluence Criteria by Strong Commutation

In this section, we present a proof method for confluence of left-linear NRS_{AX}'s using strong commutation modulo α-equivalence. First we review a basic proof method in rewriting systems with first-order rules. Then we introduce notions to discuss confluence in nominal rewriting, and give a criterion for NRS_{AX}'s.

3.1 Proof Method for Confluence by Strong Commutation

In this subsection, we survey the proof method for confluence by strong commutation. For first-order TRS's, the method is known, e.g. in [18]. Here we consider a restricted class of NRS_{AX}'s consisting only of first-order rules. Note however that the rewrite relation is still defined for ground nominal terms in NL_a.

Definition 8. An NRS_{AX} $\mathcal{R}$ is called a TRS_{AX} if for every $\nabla \vdash l \to r \in \mathcal{R}$, $\nabla = \emptyset$, and l and r are term expressions with neither atom-variables nor abstractions.

For a TRS_{AX}, we restrict the rewrite relation to the one with matching by identity instead of modulo α-equivalence (i.e. $s' = l\sigma$ instead of $\vdash_{NL_a} s' \approx_\alpha l\sigma$ in Definition 3).

Definition 9. Let $\mathcal{R}$ be a TRS_{AX}. $\to_\mathcal{R}$ is *confluent* if for all ground terms s and t, $s\ ({\leftarrow^*_\mathcal{R}} \circ {\to^*_\mathcal{R}})\ t$ implies $s\ ({\to^*_\mathcal{R}} \circ {\leftarrow^*_\mathcal{R}})\ t$.

The basic strategy in the proof method is to show commutation of any combination of two rules of the TRS_{AX}. We recall definitions and lemmas on commutation (cf. [2, pp. 31–33]).

Definition 10. Let R_1 and R_2 be rewrite rules of a TRS_{AX}.

1. $\to_{R_1}$ and $\to_{R_2}$ *commute* iff for all ground terms s_1 and s_2, if $s_1\ ({\leftarrow^*_{R_1}} \circ {\to^*_{R_2}})\ s_2$ then $s_1\ ({\to^*_{R_2}} \circ {\leftarrow^*_{R_1}})\ s_2$.
2. $\to_{R_1}$ *strongly commutes with* $\twoheadrightarrow\!\!\!\!\!\!+\ _{R_2}$ iff for all ground terms s_1 and s_2, if $s_1\ ({\leftarrow_{R_1}} \circ {\twoheadrightarrow\!\!\!\!\!\!+\ _{R_2}})\ s_2$ then $s_1\ ({\twoheadrightarrow\!\!\!\!\!\!+\ _{R_2}} \circ {\leftarrow^*_{R_1}})\ s_2$.

By Hindley's results [6] and the properties shown in Lemma 2, we have the following.

Lemma 3. *If $\to_{R_1}$ strongly commutes with $\twoheadrightarrow\!\!\!\!\!\!+\ _{R_2}$ then $\to_{R_1}$ and $\to_{R_2}$ commute.*

Lemma 4. *Let $\mathcal{R}$ be a TRS_{AX}. If $\to_{R_i}$ and $\to_{R_j}$ commute for every $R_i, R_j \in \mathcal{R}$ then $\to_\mathcal{R}$ is confluent.*

By Lemmas 3 and 4, to prove confluence of $\to_\mathcal{R}$, it is sufficient to show that for every combination of two rules $R_i, R_j \in \mathcal{R}$ (including the case $R_i = R_j$), $\to_{R_i}$ strongly commutes with $\twoheadrightarrow\!\!\!\!\!\!+\ _{R_j}$, or $\to_{R_j}$ strongly commutes with $\twoheadrightarrow\!\!\!\!\!\!+\ _{R_i}$.

Next we give conditions for strong commutation of $\to_{R_1}$ with $\twoheadrightarrow\!\!\!\!\!\!+\ _{R_2}$.

Definition 11. Let R_1 and $R_2 (= \vdash l_2 \to r_2)$ be rewrite rules of a TRS_{AX}. The conditions $\mathsf{sc}_1(R_1, R_2)$ and $\mathsf{sc}_2(R_1, R_2)$ are defined as follows:

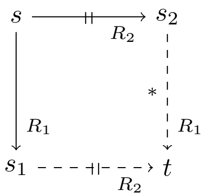
$\mathsf{sc}_1(R_1, R_2) \overset{\text{def}}{\Longleftrightarrow}$ If $s \xrightarrow{\varepsilon}_{R_1} s_1$ and $s \mathrel{\dashv\!\!\!\mapstochar\!\!\rightarrow}_{R_2} s_2$ is derived with (C) as the last applied rule, then there exists t such that $s_1 \mathrel{\dashv\!\!\!\mapstochar\!\!\rightarrow}_{R_2} t$ and $s_2 \to^*_{R_1} t$.

$\mathsf{sc}_2(R_1, R_2) \overset{\text{def}}{\Longleftrightarrow}$ If $s \xrightarrow{p}_{R_1} s_1$ and $s \xrightarrow{\varepsilon}_{R_2} s_2$ where p is a non-variable position of l_2, then there exists t such that $s_1 \mathrel{\dashv\!\!\!\mapstochar\!\!\rightarrow}_{R_2} t$ and $s_2 \to^*_{R_1} t$.

Note that the conditional part of $\mathsf{sc}_2(R_1, R_2)$ arises only when R_1 overlaps on R_2.

The next lemma guarantees that $\mathsf{sc}_1(R_1, R_2)$ and $\mathsf{sc}_2(R_1, R_2)$ are a sufficient condition for strong commutation of $\to_{R_1}$ with $\mathrel{\dashv\!\!\!\mapstochar\!\!\rightarrow}_{R_2}$. In Subsect. 3.3, we present a version of this lemma generalised to the case of NRS_{AX}.

Lemma 5. *Let R_1 and R_2 be left-linear rewrite rules of a TRS_{AX}. If the conditions $\mathsf{sc}_1(R_1, R_2)$ and $\mathsf{sc}_2(R_1, R_2)$ hold, then $\to_{R_1}$ strongly commutes with $\mathrel{\dashv\!\!\!\mapstochar\!\!\rightarrow}_{R_2}$:*



Proof. We prove by induction on the derivation of $s \mathrel{\dashv\!\!\!\mapstochar\!\!\rightarrow}_{R_2} s_2$ that if $s \to_{R_1} s_1$ and $s \mathrel{\dashv\!\!\!\mapstochar\!\!\rightarrow}_{R_2} s_2$ then there exists t such that $s_1 \mathrel{\dashv\!\!\!\mapstochar\!\!\rightarrow}_{R_2} t$ and $s_2 \to^*_{R_1} t$.

– Suppose that the last part of the derivation of $s \mathrel{\dashv\!\!\!\mapstochar\!\!\rightarrow}_{R_2} s_2$ has the form

$$\frac{u_1 \mathrel{\dashv\!\!\!\mapstochar\!\!\rightarrow}_{R_2} v_1 \quad \cdots \quad u_n \mathrel{\dashv\!\!\!\mapstochar\!\!\rightarrow}_{R_2} v_n}{G[u_1, \dots, u_n] \mathrel{\dashv\!\!\!\mapstochar\!\!\rightarrow}_{R_2} G[v_1, \dots, v_n]}\ (\mathsf{C})$$

- First we consider the case where the reduction $s \to_{R_1} s_1$ takes place in $G[u_1, \dots, u_n]$ with $u_i \to_{R_1} u'_i$ for some $i \in \{1, \dots, n\}$. Then by the induction hypothesis, there exists v'_i such that $u'_i \mathrel{\dashv\!\!\!\mapstochar\!\!\rightarrow}_{R_2} v'_i$ and $v_i \to^*_{R_1} v'_i$. Hence by applying the rule (C), we have

$$s_1 = G[u_1, \dots, u'_i, \dots, u_n] \mathrel{\dashv\!\!\!\mapstochar\!\!\rightarrow}_{R_2} G[v_1, \dots, v'_i, \dots, v_n]$$

  Also, from $v_i \to^*_{R_1} v'_i$ we have

$$s_2 = G[v_1, \dots, v_i, \dots, v_n] \to^*_{R_1} G[v_1, \dots, v'_i, \dots, v_n]$$

  Thus the claim follows by taking $t = G[v_1, \dots, v'_i, \dots, v_n]$.
- Next we consider the case where the redex of $s \to_{R_1} s_1$ is not in any u_i of $G[u_1, \dots, u_n]$. Then we can assume that the R_1-redex is at the root (i.e. $s \xrightarrow{\varepsilon}_{R_1} s_1$). Hence the claim follows from the condition $\mathsf{sc}_1(R_1, R_2)$.

– Suppose that $s \mathrel{-\!\!\shortparallel\!\!\!\rightarrow}_{R_2} s_2$ is derived by the rule (B)

$$\frac{s \xrightarrow{\varepsilon}_{R_2} s_2}{s \mathrel{-\!\!\shortparallel\!\!\!\rightarrow}_{R_2} s_2}\ (\mathsf{B})$$

where $R_2 = \vdash l_2 \to r_2$. Then by the definition of rewrite relation, there exists σ such that $s = l_2\sigma$ and $s_2 = r_2\sigma$.

- First we consider the case where the reduction $s \to_{R_1} s_1$ takes place in s with $X\sigma \to_{R_1} X\delta$ for some $X \in \mathit{Var}_{\mathcal{X}}(l_2)$, and $Y\sigma = Y\delta$ for all $Y(\neq X) \in \mathit{Var}_{\mathcal{X}}(l_2)$. Then by the left-linearity of R_2, we have $s_1 = l_2\delta \xrightarrow{\varepsilon}_{R_2} r_2\delta$, and so $s_1 \mathrel{-\!\!\shortparallel\!\!\!\rightarrow}_{R_2} r_2\delta$ by the rule (B). Also, we have $s_2 = r_2\sigma \to^*_{R_1} r_2\delta$. Hence the claim follows by taking $t = r_2\delta$.
- Otherwise, the reduction $s \to_{R_1} s_1$ takes place in s with $s \xrightarrow{p}_{R_1} s_1$ where p is a non-variable position of l_2. Then the claim follows from the condition $\mathsf{sc}_2(R_1, R_2)$. □

By Lemmas 3, 4 and 5, we have the following theorem.

Theorem 1. *Let $\mathcal{R}$ be a left-linear TRS$_{AX}$. If for every $R_i, R_j \in \mathcal{R}$, $\mathsf{sc}_1(R_i, R_j)$ and $\mathsf{sc}_2(R_i, R_j)$, or $\mathsf{sc}_1(R_j, R_i)$ and $\mathsf{sc}_2(R_j, R_i)$, then $\to_{\mathcal{R}}$ is confluent.*

We give an example of application of the theorem.

Example 6. We extend the nominal signature in Example 1 by function symbols (constants) S and K. Consider the *TRS$_{AX}$* $\mathcal{R}_{\mathsf{CL}}$ consisting of the rewrite rules of combinatory logic (CL):

$$\begin{array}{lr} \vdash \mathtt{app}\langle \mathtt{app}\langle \mathtt{app}\langle \mathtt{S}, X\rangle, Y\rangle, Z\rangle \to \mathtt{app}\langle \mathtt{app}\langle X, Z\rangle, \mathtt{app}\langle Y, Z\rangle\rangle & (\mathtt{S}) \\ \vdash \qquad\quad \mathtt{app}\langle \mathtt{app}\langle \mathtt{K}, X\rangle, Y\rangle \to X & (\mathtt{K}) \end{array}$$

We check the condition $\mathsf{sc}_1((\mathtt{S}), (\mathtt{K}))$. Suppose $\mathtt{app}\langle \mathtt{app}\langle \mathtt{app}\langle \mathtt{S}, u_1\rangle, u_2\rangle, u_3\rangle \xrightarrow{\varepsilon}_{\mathtt{S}}$ $\mathtt{app}\langle \mathtt{app}\langle u_1,\ u_3\rangle, \mathtt{app}\langle u_2, u_3\rangle\rangle$ and $\mathtt{app}\langle \mathtt{app}\langle \mathtt{app}\langle \mathtt{S}, u_1\rangle, u_2\rangle, u_3\rangle \mathrel{-\!\!\shortparallel\!\!\!\rightarrow}_{\mathtt{K}} s_2$ with its last applied rule (C). Then the derivation of the latter must have the form

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\mathtt{S} \mathrel{-\!\!\shortparallel\!\!\!\rightarrow}_{\mathtt{K}} \mathtt{S}}\,(\mathsf{C}) \qquad \begin{array}{c}\vdots\ D_1 \\ u_1 \mathrel{-\!\!\shortparallel\!\!\!\rightarrow}_{\mathtt{K}} v_1\end{array}}{\mathtt{app}\langle \mathtt{S}, u_1\rangle \mathrel{-\!\!\shortparallel\!\!\!\rightarrow}_{\mathtt{K}} \mathtt{app}\langle \mathtt{S}, v_1\rangle}\,(\mathsf{C}) \qquad \begin{array}{c}\vdots\ D_2 \\ u_2 \mathrel{-\!\!\shortparallel\!\!\!\rightarrow}_{\mathtt{K}} v_2\end{array}}{\mathtt{app}\langle \mathtt{app}\langle \mathtt{S}, u_1\rangle, u_2\rangle \mathrel{-\!\!\shortparallel\!\!\!\rightarrow}_{\mathtt{K}} \mathtt{app}\langle \mathtt{app}\langle \mathtt{S}, v_1\rangle, v_2\rangle}\,(\mathsf{C}) \qquad \begin{array}{c}\vdots\ D_3 \\ u_3 \mathrel{-\!\!\shortparallel\!\!\!\rightarrow}_{\mathtt{K}} v_3\end{array}}{\mathtt{app}\langle \mathtt{app}\langle \mathtt{app}\langle \mathtt{S}, u_1\rangle, u_2\rangle, u_3\rangle \mathrel{-\!\!\shortparallel\!\!\!\rightarrow}_{\mathtt{K}} \mathtt{app}\langle \mathtt{app}\langle \mathtt{app}\langle \mathtt{S}, v_1\rangle, v_2\rangle, v_3\rangle\ (= s_2)}\,(\mathsf{C})$$

Hence we can construct a derivation of $\mathtt{app}\langle \mathtt{app}\langle u_1, u_3\rangle, \mathtt{app}\langle u_2, u_3\rangle\rangle \mathrel{-\!\!\shortparallel\!\!\!\rightarrow}_{\mathtt{K}}$ $\mathtt{app}\langle \mathtt{app}\langle v_1, v_3\rangle, \mathtt{app}\langle v_2, v_3\rangle\rangle$ from D_1, D_2 and D_3 by using the rule (C). We also have $s_2 = \mathtt{app}\langle \mathtt{app}\langle \mathtt{app}\langle \mathtt{S}, v_1\rangle, v_2\rangle, v_3\rangle \to_{\mathtt{S}} \mathtt{app}\langle \mathtt{app}\langle v_1, v_3\rangle, \mathtt{app}\langle v_2, v_3\rangle\rangle$, and so the condition $\mathsf{sc}_1((\mathtt{S}), (\mathtt{K}))$ is satisfied. On the other hand, it is seen that the condition $\mathsf{sc}_2((\mathtt{S}), (\mathtt{K}))$ is vacuously satisfied.

Next we consider the case where both rules are the rule (S). The condition $\mathsf{sc}_1((\mathtt{S}), (\mathtt{S}))$ can be checked similarly to the above case. For the condition

$\mathsf{sc}_2((\mathtt{S}),(\mathtt{S}))$, we only have to check the case where both redexes are at the root, and in that case the claim clearly holds.

The case where both rules are the rule $(\mathtt{K})$ can be checked similarly.

Therefore by Theorem 1, $\rightarrow_{\mathcal{R}_{\mathsf{CL}}}$ $(= \rightarrow_{\mathtt{S}} \cup \rightarrow_{\mathtt{K}})$ is confluent. □

Note that ground terms here include countably many atoms (cf. Subsects. 2.1 and 2.2). By considering atoms as variables (rather than constants), we can see that confluence of a TRS_{AX} discussed above is an extension of confluence of the standard first-order TRS with the same function symbols and rewrite rules (so it is a property stronger than ground confluence of the first-order TRS). The TRS_{AX} $\mathcal{R}_{\mathsf{CL}}$ in Example 6 treats ground terms with atoms, and this is natural when considering an operator like λ^* (Definition 2.14 of [7]).

3.2 Confluence Properties in Nominal Rewriting

To discuss confluence in nominal rewriting, it is necessary to examine whether two terms can rewrite to the same term modulo α-equivalence. For doing this, we make use of suitable notions that are defined modulo an equivalence relation in terms of abstract reduction systems [11,12].

Definition 12. Let $\mathcal{R}$ be an NRS_{AX}.

1. $\rightarrow_{\mathcal{R}}$ is *confluent modulo* $\approx_\alpha$ iff for all ground terms s and t, if $s\ (\leftarrow^*_{\mathcal{R}} \circ \rightarrow^*_{\mathcal{R}})\ t$ then $s\ (\rightarrow^*_{\mathcal{R}} \circ \approx_\alpha \circ \leftarrow^*_{\mathcal{R}})\ t$.
2. $\rightarrow_{\mathcal{R}}$ is *Church-Rosser modulo* $\approx_\alpha$ iff for all ground terms s and t, if $s\ (\leftarrow_{\mathcal{R}} \cup \rightarrow_{\mathcal{R}} \cup \approx_\alpha)^*\ t$ then $s\ (\rightarrow^*_{\mathcal{R}} \circ \approx_\alpha \circ \leftarrow^*_{\mathcal{R}})\ t$.

In general, Church-Rosser modulo an equivalence relation $\sim$ is a stronger property than confluence modulo $\sim$ [11]. So, in the rest of this section, we aim to give a sufficient condition for Church-Rosser modulo $\approx_\alpha$ of left-linear NRS_{AX}'s.

To this end, we restrict the class of NRS_{AX}'s further by (an adaptation of) the uniformity condition [3]. Intuitively, uniformity means that if an atom a is not free in s and s rewrites to t then a is not free in t.

Definition 13. A rewrite rule $\nabla \vdash l \rightarrow r$ is *uniform* if the following holds: for every atom a and every ground substitution σ such that $Var_{\mathcal{X},\mathcal{X}_A}(l) \subseteq dom(\sigma)$, if $\vdash_{NL_a} \nabla\sigma$ and $\vdash_{NL_a} a\#l\sigma$ then $\vdash_{NL_a} a\#r\sigma$. A rewriting system is *uniform* if so are all its rewrite rules.

For uniform rewrite rules, the following properties hold.

Lemma 6. *Suppose $s \rightarrow_R t$ for a uniform rewrite rule R. Then, for every atom a, if $\vdash_{NL_a} a\#s$ then $\vdash_{NL_a} a\#t$.*

Proof. This is proved in the same way as Proposition 2 of [8]. □

Definition 14. A relation $\rightarrow$ on ground terms is *strongly compatible with* $\approx_\alpha$ iff for all ground terms s and t, if $s\ (\approx_\alpha \circ \rightarrow)\ t$ then $s\ (\rightarrow^= \circ \approx_\alpha)\ t$.

Lemma 7. *If R is a uniform rewrite rule, then $\rightarrow_R$ is strongly compatible with $\approx_\alpha$ and $\twoheadrightarrow\!\!\!\!\!\!+\ _R$ is strongly compatible with $\approx_\alpha$.*

Proof. This is proved in the same way as Lemmas 3 and 8 of [8]. □

3.3 A Sufficient Condition for Church-Rosser Modulo α-equivalence

Now we present a sufficient condition for Church-Rosser modulo $\approx_\alpha$ extending the sufficient condition for confluence in Theorem 1. First we introduce the notions of commutation and strong commutation modulo $\approx_\alpha$. The latter is not treated in [11,12] (in the case of a general equivalence relation $\sim$).

Definition 15. Let R_1 and R_2 be rewrite rules of an NRS_{AX}.

1. $\to_{R_1}$ and $\to_{R_2}$ *commute modulo* $\approx_\alpha$ iff for all ground terms s_1 and s_2, if $s_1\ (\leftarrow^*_{R_1} \circ \to^*_{R_2})\ s_2$ then $s_1\ (\to^*_{R_2} \circ \approx_\alpha \circ \leftarrow^*_{R_1})\ s_2$.
2. $\to_{R_1}$ *strongly commutes with* $\twoheadrightarrow\!\!\!\!\!\!\!+\ _{R_2}$ *modulo* $\approx_\alpha$ iff for all ground terms s_1 and s_2, if $s_1\ (\leftarrow_{R_1} \circ \twoheadrightarrow\!\!\!\!\!\!\!+\ _{R_2})\ s_2$ then $s_1\ (\twoheadrightarrow\!\!\!\!\!\!\!+\ _{R_2} \circ \approx_\alpha \circ \leftarrow^*_{R_1})\ s_2$.

The following lemmas are counterparts of Lemmas 3 and 4 in Subsect. 3.1.

Lemma 8. *If $\to_{R_1}$ strongly commutes with $\twoheadrightarrow\!\!\!\!\!\!\!+\ _{R_2}$ modulo $\approx_\alpha$, and both $\to_{R_1}$ and $\twoheadrightarrow\!\!\!\!\!\!\!+\ _{R_2}$ are strongly compatible with $\approx_\alpha$, then $\to_{R_1}$ and $\to_{R_2}$ commute modulo $\approx_\alpha$.*

Proof. First we consider the claim that for all ground terms s, s_1 and s_2, if $s_1 \leftarrow^*_{R_1} s \twoheadrightarrow\!\!\!\!\!\!\!+\ _{R_2} s_2$ then there exist ground terms t_1 and t_2 such that $s_1 \twoheadrightarrow\!\!\!\!\!\!\!+\ _{R_2} t_1 \approx_\alpha t_2 \leftarrow^*_{R_1} s_2$. This is proved by induction on the length of the steps of $s_1 \leftarrow^*_{R_1} s$.

Next we show that for all ground terms s, s_1 and s_2, if $s_1 \leftarrow^*_{R_1} s \twoheadrightarrow\!\!\!\!\!\!\!+\ ^*_{R_2} s_2$ then there exist ground terms t_1 and t_2 such that $s_1 \twoheadrightarrow\!\!\!\!\!\!\!+\ ^*_{R_2} t_1 \approx_\alpha t_2 \leftarrow^*_{R_1} s_2$. This is proved by induction on the length of the steps of $s \twoheadrightarrow\!\!\!\!\!\!\!+\ ^*_{R_2} s_2$. By Lemma 2, $\twoheadrightarrow\!\!\!\!\!\!\!+\ ^*_{R_2} = \to^*_{R_2}$, so we have that $\to_{R_1}$ and $\to_{R_2}$ commute modulo $\approx_\alpha$. □

Lemma 9. *Let $\mathcal{R}$ be a uniform NRS_{AX}. If $\to_{R_i}$ and $\to_{R_j}$ commute modulo $\approx_\alpha$ for every $R_i, R_j \in \mathcal{R}$, then $\to_\mathcal{R}$ is Church-Rosser modulo $\approx_\alpha$.*

Proof. By Lemma 7, $\to_{R_i}$ is strongly compatible with $\approx_\alpha$ for every $R_i \in \mathcal{R}$. Then the claim follows by Corollary 2.6.5 of [12]. □

Next we give conditions for strong commutation of $\to_{R_1}$ with $\twoheadrightarrow\!\!\!\!\!\!\!+\ _{R_2}$ modulo $\approx_\alpha$.

Definition 16. Let R_1 and R_2 $(= \nabla \vdash l_2 \to r_2)$ be rewrite rules of an NRS_{AX}. The conditions $\mathsf{sc}_1(R_1, R_2, \approx_\alpha)$ and $\mathsf{sc}_2(R_1, R_2, \approx_\alpha)$ are defined as follows:

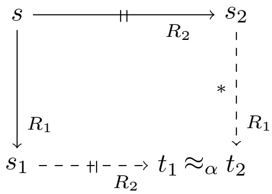
$\mathsf{sc}_1(R_1, R_2, \approx_\alpha) \overset{\text{def}}{\iff}$ If $s \overset{\varepsilon}{\to}_{R_1} s_1$ and $s \twoheadrightarrow\!\!\!\!\!\!\!+\ _{R_2} s_2$ is derived with (C) as the last applied rule, then there exist t_1 and t_2 such that $s_1 \twoheadrightarrow\!\!\!\!\!\!\!+\ _{R_2} t_1$, $s_2 \to^*_{R_1} t_2$ and $t_1 \approx_\alpha t_2$.

$\mathsf{sc}_2(R_1, R_2, \approx_\alpha) \overset{\text{def}}{\iff}$ If $s \overset{p}{\to}_{R_1} s_1$ and $s \overset{\varepsilon}{\to}_{R_2} s_2$ where p is a non-variable position of l_2, then there exist t_1 and t_2 such that $s_1 \twoheadrightarrow\!\!\!\!\!\!\!+\ _{R_2} t_1$, $s_2 \to^*_{R_1} t_2$ and $t_1 \approx_\alpha t_2$.

Note that the conditional part of $\mathsf{sc}_2(R_1, R_2, \approx_\alpha)$ arises only when R_1 overlaps on R_2.

The next lemma guarantees that $\mathsf{sc}_1(R_1, R_2, \approx_\alpha)$ and $\mathsf{sc}_2(R_1, R_2, \approx_\alpha)$ are a sufficient condition for strong commutation of $\rightarrow_{R_1}$ with $\mathrel{+\!\!\!\!\!\rightarrow}_{R_2}$ modulo $\approx_\alpha$.

Lemma 10. *Let R_1 and R_2 be left-linear uniform rewrite rules of an NRS_{AX}. If the conditions $\mathsf{sc}_1(R_1, R_2, \approx_\alpha)$ and $\mathsf{sc}_2(R_1, R_2, \approx_\alpha)$ hold, then $\rightarrow_{R_1}$ strongly commutes with $\mathrel{+\!\!\!\!\!\rightarrow}_{R_2}$ modulo $\approx_\alpha$:*



Proof. We prove by induction on the derivation of $s \mathrel{+\!\!\!\!\!\rightarrow}_{R_2} s_2$ that if $s \rightarrow_{R_1} s_1$ and $s \mathrel{+\!\!\!\!\!\rightarrow}_{R_2} s_2$ then there exist t_1 and t_2 such that $s_1 \mathrel{+\!\!\!\!\!\rightarrow}_{R_2} t_1$, $s_2 \rightarrow^*_{R_1} t_2$ and $t_1 \approx_\alpha t_2$.

– Suppose that the last part of the derivation of $s \mathrel{+\!\!\!\!\!\rightarrow}_{R_2} s_2$ has the form

$$\frac{u_1 \mathrel{+\!\!\!\!\!\rightarrow}_{R_2} v_1 \quad \cdots \quad u_n \mathrel{+\!\!\!\!\!\rightarrow}_{R_2} v_n}{G[u_1, \ldots, u_n] \mathrel{+\!\!\!\!\!\rightarrow}_{R_2} G[v_1, \ldots, v_n]} \ (\mathsf{C})$$

- First we consider the case where the reduction $s \rightarrow_{R_1} s_1$ takes place in $G[u_1, \ldots, u_n]$ with $u_i \rightarrow_{R_1} u'_i$ for some $i \in \{1, \ldots, n\}$. Then by the induction hypothesis, there exist v'_{i1} and v'_{i2} such that $u'_i \mathrel{+\!\!\!\!\!\rightarrow}_{R_2} v'_{i1}$, $v_i \rightarrow^*_{R_1} v'_{i2}$ and $v'_{i1} \approx_\alpha v'_{i2}$. Hence by applying the rule (C), we have

$$s_1 = G[u_1, \ldots, u'_i, \ldots, u_n] \mathrel{+\!\!\!\!\!\rightarrow}_{R_2} G[v_1, \ldots, v'_{i1}, \ldots, v_n]$$

  Also, from $v_i \rightarrow^*_{R_1} v'_{i2}$ we have

$$s_2 = G[v_1, \ldots, v_i, \ldots, v_n] \rightarrow^*_{R_1} G[v_1, \ldots, v'_{i2}, \ldots, v_n]$$

  Thus the claim follows by taking $t_1 = G[v_1, \ldots, v'_{i1}, \ldots, v_n]$ and $t_2 = G[v_1, \ldots, v'_{i2}, \ldots, v_n]$.
- Next we consider the case where the redex of $s \rightarrow_{R_1} s_1$ is not in any u_i of $G[u_1, \ldots, u_n]$. Then we can assume that the R_1-redex is at the root (i.e. $s \xrightarrow{\varepsilon}_{R_1} s_1$). Hence the claim follows from the condition $\mathsf{sc}_1(R_1, R_2, \approx_\alpha)$.

– Suppose that $s \mathrel{+\!\!\!\!\!\rightarrow}_{R_2} s_2$ is derived by the rule (B)

$$\frac{s \xrightarrow{\varepsilon}_{R_2} s_2}{s \mathrel{+\!\!\!\!\!\rightarrow}_{R_2} s_2} \ (\mathsf{B})$$

where $R_2 = \nabla \vdash l_2 \rightarrow r_2$. Then by the definition of rewrite relation, there exists σ such that $\vdash_{NL_a} \nabla\sigma$, $\vdash_{NL_a} s \approx_\alpha l_2\sigma$ and $s_2 = r_2\sigma$.

- First we consider the case where the reduction $s \xrightarrow{p}_{R_1} s_1$ takes place at a position p that is a variable position q of l_2 or deeper. Let $l_2|_q = X$. Then by Lemma 11 below, there exists δ such that $\vdash_{NL_a} s_1 \approx_\alpha l_2\delta$, $X\sigma \rightarrow_{R_1} X\delta$ and $Y\sigma = Y\delta$ for all $Y(\neq X) \in \mathit{Var}_{\mathcal{X}}(l_2)$. Since we can see $\vdash_{NL_a} \nabla\delta$ using Lemma 6 (cf. Lemma 7(2) of [8]), we have $s_1 \xrightarrow{\varepsilon}_{R_2} r_2\delta$, and so $s_1 \twoheadrightarrow\!\!\!\!\!+\ _{R_2} r_2\delta$ by the rule (B). Also, we have $s_2 = r_2\sigma \rightarrow^*_{R_1} r_2\delta$. Hence the claim follows by taking $t_1 = t_2 = r_2\delta$.
- Otherwise, the reduction $s \rightarrow_{R_1} s_1$ takes place in s with $s \xrightarrow{p}_{R_1} s_1$ where p is a non-variable position of l_2. Then the claim follows from the condition $\mathsf{sc}_2(R_1, R_2, \approx_\alpha)$. □

Lemma 11. *Let R_1 and R_2 $(= \nabla \vdash l_2 \rightarrow r_2)$ be left-linear uniform rewrite rules of an NRS_{AX}. Suppose that σ is a ground substitution with $\mathit{Var}_{\mathcal{X},\mathcal{X}_A}(l_2) \subseteq \mathit{dom}(\sigma)$ and $\vdash_{NL_a} \nabla\sigma$. Suppose also that a reduction $s \xrightarrow{p}_{R_1} s_1$ takes place at a position p that is a variable position q of l_2 or deeper, and $l_2|_q = X$. Then, for every position q' from ε to q, if $\vdash_{NL_a} s|_{q'} \approx_\alpha l_2|_{q'}\sigma$ then there exists δ such that $\vdash_{NL_a} s_1|_{q'} \approx_\alpha l_2|_{q'}\delta$, $X\sigma \rightarrow_{R_1} X\delta$, and $Y\sigma = Y\delta$ for all $Y(\neq X) \in \mathit{Var}_{\mathcal{X}}(l_2)$.*

Proof. By induction on the length of $q \setminus q'$.

First we consider the case $q' = q$. Then $l_2|_{q'} = l_2|_q = X$. Suppose $\vdash_{NL_a} s|_{q'} \approx_\alpha l_2|_{q'}\sigma = X\sigma$. Since $s \xrightarrow{p}_{R_1} s_1$ with a deeper position p than q, we have $s|_q \rightarrow_{R_1} s_1|_q$. So by the strong compatibility of $\rightarrow_{R_1}$ (with $\rightarrow_{R_1}$ instead of $\rightarrow^{=}_{R_1}$) there exists t such that $X\sigma \rightarrow_{R_1} t \approx_\alpha s_1|_q$. Hence we can take δ with $X\delta = t$ and $\vdash_{NL_a} s_1|_q \approx_\alpha X\delta = l_2|_q\delta$.

For the other cases, the proof is by case analysis according to the form of $l_2|_{q'}$. This is shown analogously to the case analysis in the proof of Lemma 10 of [8]. □

By Lemmas 7, 8, 9 and 10, we have the following theorem.

Theorem 2. *Let $\mathcal{R}$ be a left-linear uniform NRS_{AX}. If for every $R_i, R_j \in \mathcal{R}$, $\mathsf{sc}_1(R_i, R_j, \approx_\alpha)$ and $\mathsf{sc}_2(R_i, R_j, \approx_\alpha)$, or $\mathsf{sc}_1(R_j, R_i, \approx_\alpha)$ and $\mathsf{sc}_2(R_j, R_i, \approx_\alpha)$ then $\rightarrow_{\mathcal{R}}$ is Church-Rosser modulo $\approx_\alpha$.*

In practice, if we know that R_1 does not overlap on R_2 and vice versa, we may use, instead of Lemma 8, Theorem 1 of [8] to show commutation modulo $\approx_\alpha$ of $\rightarrow_{R_1}$ and $\rightarrow_{R_2}$. So, to apply Theorem 2, we can concentrate on R_i and R_j such that there exists an overlap of R_i on R_j or R_j on R_i. Moreover, for $R_i = R_j$, we may skip checking the case $p = \varepsilon$ in $\mathsf{sc}_2(R_i, R_j, \approx_\alpha)$ when R_i is α-stable [16], so that we have only to check rules with proper overlaps when the NRS_{AX} is α-stable.

Definition 17 (α-stability). *A rewrite rule $R = \nabla \vdash l \rightarrow r$ is* α-stable *if $\vdash_{NL_a} s \approx_\alpha s'$, $s \rightarrow_{\langle R,\varepsilon,\sigma\rangle} t$ and $s' \rightarrow_{\langle R,\varepsilon,\sigma'\rangle} t'$ imply $\vdash_{NL_a} t \approx_\alpha t'$. An NRS_{AX} $\mathcal{R}$ is* α-stable *if so are all its rewrite rules.*

We demonstrate Theorem 2 on two examples.

Example 7. The NRS_{AX} $\mathcal{R}_{\mathsf{sub}}$ in Example 2 is left-linear, uniform and α-stable. In this NRS_{AX}, there are two pairs of rules that have proper overlaps: $((\mathsf{sub}_{\mathsf{app}}), (\mathsf{sub}_{\epsilon}))$ and $((\mathsf{sub}_{\mathsf{lam}}), (\mathsf{sub}_{\epsilon}))$.

For the pair $((\mathsf{sub}_{\mathsf{app}}), (\mathsf{sub}_{\epsilon}))$, we first check the condition $\mathsf{sc}_1((\mathsf{sub}_{\mathsf{app}}), (\mathsf{sub}_{\epsilon}), \approx_\alpha)$. Suppose $\mathtt{sub}\langle[a]\mathtt{app}\langle u_1, u_2\rangle, u_3\rangle \xrightarrow{\varepsilon}_{\mathsf{sub}_{\mathsf{app}}} \mathtt{app}\langle\mathtt{sub}\langle[a]u_1, u_3\rangle, \mathtt{sub}\langle[a]u_2, u_3\rangle\rangle$ and $\mathtt{sub}\langle[a]\mathtt{app}\langle u_1, u_2\rangle, u_3\rangle \mathrel{+\!\!\!\!\!+\!\!\!\!\rightarrow}_{\mathsf{sub}_{\epsilon}} s_2$ with its last applied rule (C). Then the derivation of the latter must have the form

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots\, D_1}{u_1 \mathrel{+\!\!\!\!\!+\!\!\!\!\rightarrow}_{\mathsf{sub}_\epsilon} v_1} \qquad \dfrac{\vdots\, D_2}{u_2 \mathrel{+\!\!\!\!\!+\!\!\!\!\rightarrow}_{\mathsf{sub}_\epsilon} v_2}}{\mathtt{app}\langle u_1, u_2\rangle \mathrel{+\!\!\!\!\!+\!\!\!\!\rightarrow}_{\mathsf{sub}_\epsilon} \mathtt{app}\langle v_1, v_2\rangle}\,(\mathsf{C})}{[a]\mathtt{app}\langle u_1, u_2\rangle \mathrel{+\!\!\!\!\!+\!\!\!\!\rightarrow}_{\mathsf{sub}_\epsilon} [a]\mathtt{app}\langle v_1, v_2\rangle}\,(\mathsf{C}) \qquad \dfrac{\vdots\, D_3}{u_3 \mathrel{+\!\!\!\!\!+\!\!\!\!\rightarrow}_{\mathsf{sub}_\epsilon} v_3}}{\mathtt{sub}\langle[a]\mathtt{app}\langle u_1, u_2\rangle, u_3\rangle \mathrel{+\!\!\!\!\!+\!\!\!\!\rightarrow}_{\mathsf{sub}_\epsilon} \mathtt{sub}\langle[a]\mathtt{app}\langle v_1, v_2\rangle, v_3\rangle\ (= s_2)}\,(\mathsf{C})$$

Hence we can construct a derivation of $\mathtt{app}\langle\mathtt{sub}\langle[a]u_1, u_3\rangle, \mathtt{sub}\langle[a]u_2, u_3\rangle\rangle \mathrel{+\!\!\!\!\!+\!\!\!\!\rightarrow}_{\mathsf{sub}_\epsilon} \mathtt{app}\langle\mathtt{sub}\langle[a]v_1, v_3\rangle, \mathtt{sub}\langle[a]v_2, v_3\rangle\rangle$ from D_1, D_2 and D_3 by using the rule (C). We also have $s_2 = \mathtt{sub}\langle[a]\mathtt{app}\langle v_1, v_2\rangle, v_3\rangle \rightarrow_{\mathsf{sub}_{\mathsf{app}}} \mathtt{app}\langle\mathtt{sub}\langle[a]v_1, v_3\rangle, \mathtt{sub}\langle[a]v_2, v_3\rangle\rangle$, and so the condition $\mathsf{sc}_1((\mathsf{sub}_{\mathsf{app}}), (\mathsf{sub}_{\epsilon}))$ is satisfied.

Next we check the condition $\mathsf{sc}_2((\mathsf{sub}_{\mathsf{app}}), (\mathsf{sub}_{\epsilon}), \approx_\alpha)$. Suppose $\mathtt{sub}\langle[a]\mathtt{app}\langle u_1, u_2\rangle, u_3\rangle \xrightarrow{\varepsilon}_{\mathsf{sub}_{\mathsf{app}}} \mathtt{app}\langle\mathtt{sub}\langle[a]u_1, u_3\rangle, \mathtt{sub}\langle[a]u_2, u_3\rangle\rangle$ and $\mathtt{sub}\langle[a]\mathtt{app}\langle u_1, u_2\rangle, u_3\rangle \xrightarrow{\varepsilon}_{\mathsf{sub}_\epsilon} \mathtt{app}\langle u_1, u_2\rangle\ (= s_2)$. From the latter, we see $\vdash_{NL_a} a\#u_1$ and $\vdash_{NL_a} a\#u_2$, and so we have $\mathtt{sub}\langle[a]u_1, u_3\rangle \rightarrow_{\mathsf{sub}_\epsilon} u_1$ and $\mathtt{sub}\langle[a]u_2, u_3\rangle \rightarrow_{\mathsf{sub}_\epsilon} u_2$. Hence we can construct a derivation of $\mathtt{app}\langle\mathtt{sub}\langle[a]u_1, u_3\rangle, \mathtt{sub}\langle[a]u_2, u_3\rangle\rangle \mathrel{+\!\!\!\!\!+\!\!\!\!\rightarrow}_{\mathsf{sub}_\epsilon} \mathtt{app}\langle u_1, u_2\rangle$. Thus $\mathsf{sc}_2((\mathsf{sub}_{\mathsf{app}}), (\mathsf{sub}_{\epsilon}), \approx_\alpha)$ holds by taking $t_1 = t_2 = \mathtt{app}\langle u_1, u_2\rangle$.

For the other pair of rules with a proper overlap, we can analogously check $\mathsf{sc}_1((\mathsf{sub}_{\mathsf{lam}}), (\mathsf{sub}_{\epsilon}), \approx_\alpha)$ and $\mathsf{sc}_2((\mathsf{sub}_{\mathsf{lam}}), (\mathsf{sub}_{\epsilon}), \approx_\alpha)$.

Therefore by Theorem 2, we see that $\rightarrow_{\mathcal{R}_{\mathsf{sub}}}$ is Church-Rosser modulo $\approx_\alpha$. □

Example 8. Consider the NRS_{AX} $\mathcal{R}_{\mathsf{subdup}}$ obtained from $\mathcal{R}_{\mathsf{sub}}$ in Example 2 by adding the following rewrite rule:

$$A\#Y \vdash \mathtt{sub}\langle[A]X, Y\rangle \rightarrow \mathtt{sub}\langle[A]\mathtt{sub}\langle[A]X, Y\rangle, Y\rangle \qquad (\mathsf{sub}_{\mathsf{dup}})$$

This NRS_{AX} $\mathcal{R}_{\mathsf{subdup}}$ is left-linear, uniform and α-stable. Also we see that it is non-terminating due to the rule $(\mathsf{sub}_{\mathsf{dup}})$. By applying Theorem 2, we can show that $\rightarrow_{\mathcal{R}_{\mathsf{subdup}}}$ is Church-Rosser modulo $\approx_\alpha$. □

4 Conclusion

We presented a sufficient condition for Church-Rosser modulo α-equivalence (on ground nominal terms) of left-linear uniform NRS_{AX}'s that may have overlaps of rewrite rules and may be non-terminating. This was achieved by introducing the notion of strong commutation modulo α-equivalence and giving a sufficient condition for it.

Currently, we are working on implementation of a tool that verifies sufficient conditions as developed in this paper. To compute overlaps in NRS_{AX}'s and extract useful information, it is necessary to construct an appropriate unification procedure for variable-atom nominal unification problems.

Acknowledgements. I am grateful to the anonymous referees for valuable comments. This work was partly supported by JSPS KAKENHI Grant Numbers JP19K11891 and JP20H04164.

References

1. Ayala-Rincón, M., Fernández, M., Gabbay, M.J., Rocha-Oliveira, A.C.: Checking overlaps of nominal rewriting rules. Electron. Notes Theor. Comput. Sci. **323**, 39–56 (2016)
2. Baader, F., Nipkow, T.: Term Rewriting and All That. Cambridge University Press, Cambridge (1998)
3. Fernández, M., Gabbay, M.J.: Nominal rewriting. Inform. Comput. **205**, 917–965 (2007)
4. Fernández, M., Gabbay, M.J., Mackie, I.: Nominal rewriting systems. In: Proceedings of the 6th PPDP, pp. 108–119. ACM (2004)
5. Gabbay, M.J., Pitts, A.M.: A new approach to abstract syntax with variable binding. Formal Aspects Comput. **13**, 341–363 (2002)
6. Hindley, J.R.: The church-rosser property and a result in combinatory logic. Ph.D. thesis, University of Newcastle-upon-Tyne (1964)
7. Hindley, J.R., Seldin, J.P.: Introduction to Combinators and Lambda-Calculus. Cambridge University Press, Cambridge (1986)
8. Kikuchi, K., Aoto, T.: Confluence and commutation for nominal rewriting systems with atom-variables. In: Fernández, M. (ed.) LOPSTR 2020. LNCS, vol. 12561, pp. 56–73. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-68446-4_3
9. Kikuchi, K., Aoto, T., Toyama, Y.: Parallel closure theorem for left-linear nominal rewriting systems. In: Dixon, C., Finger, M. (eds.) FroCoS 2017. LNCS (LNAI), vol. 10483, pp. 115–131. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66167-4_7
10. Kutz, Y., Schmidt-Schauß, M.: Rewriting with generalized nominal unification. Math. Struct. Comput. Sci. **30**, 710–735 (2020)
11. Ohlebusch, E.: Church-Rosser theorems for abstract reduction modulo an equivalence relation. In: Nipkow, T. (ed.) RTA 1998. LNCS, vol. 1379, pp. 17–31. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0052358
12. Ohlebusch, E.: Advanced Topics in Term Rewriting. Springer, New York (2002). https://doi.org/10.1007/978-1-4757-3661-8
13. Pitts, A.M.: Nominal logic, a first order theory of names and binding. Inform. Comput. **186**, 165–193 (2003)
14. Rosen, B.: Tree-manipulating systems and Church-Rosser theorems. J. ACM **20**(1), 160–187 (1973)
15. Schmidt-Schauß, M., Sabel, D., Kutz, Y.D.K.: Nominal unification with atom-variables. J. Symb. Comput. **90**, 42–64 (2019)
16. Suzuki, T., Kikuchi, K., Aoto, T., Toyama, Y.: Confluence of orthogonal nominal rewriting systems revisited. In: Proceedings of the 26th RTA. LIPIcs, vol. 36, pp. 301–317 (2015)

17. Suzuki, T., Kikuchi, K., Aoto, T., Toyama, Y.: Critical pair analysis in nominal rewriting. In: Proceedings of the 7th SCSS. EPiC, vol. 39, pp. 156–168. EasyChair (2016)
18. Toyama, Y.: On the Church-Rosser property of term rewriting systems. Technical Report 17672, NTT ECL (1981). (in Japanese)
19. Urban, C., Pitts, A.M., Gabbay, M.J.: Nominal unification. Theoret. Comput. Sci. **323**, 473–497 (2004)
20. Vestergaard, R., Brotherston, J.: A formalised first-order confluence proof for the λ-calculus using one-sorted variable names. Inform. Comput. **183**, 212–244 (2003)