EPiC Series in Computing

Volume 39, 2016, Pages 156–168

SCSS 2016. 7th International Symposium on Symbolic Computation in Software Science

Critical Pair Analysis in Nominal Rewriting

Takaki Suzuki[1], Kentaro Kikuchi[1], Takahito Aoto[2], and Yoshihito Toyama[1]

[1] RIEC, Tohoku University, Sendai, Miyagi, Japan
{ takaki, kentaro, toyama }@nue.riec.tohoku.ac.jp
[2] Faculty of Engineering, Niigata University, Niigata, Japan
aoto@ie.niigata-u.ac.jp

Abstract

Nominal rewriting (Fernández, Gabbay & Mackie, 2004; Fernández & Gabbay, 2007) is a framework that extends first-order term rewriting by a binding mechanism based on the nominal approach (Gabbay & Pitts, 2002; Pitts, 2003). In this paper, we investigate confluence properties of nominal rewriting, following the study of orthogonal systems in (Suzuki et al., 2015), but here we treat systems in which overlaps of the rewrite rules exist. First we present an example where choice of bound variables (atoms) of rules affects joinability of the induced critical pairs. Then we give a proof of the critical pair lemma, and illustrate some of its applications including confluence results for non-terminating systems.

1 Introduction

Variable binding is ubiquitous in many expressive formal systems such as systems of predicate logics, λ-calculi, process calculi, etc. Every language containing variable binding needs to deal with α-equivalence. Intuitively α-equivalence may be dealt with implicitly, but much effort is required in formal treatment. To overcome the difficulty, various studies have been made in the literature, among which the nominal approach [4, 9] is a novel one—unlike other approaches, it incorporates permutations and freshness conditions on variables (atoms) as basic ingredients.

To deal with equational logics containing variable binding, various frameworks of higher-order rewriting have been proposed (e.g. [5, 6]). *Nominal rewriting* [2, 3] has been introduced as a new framework of higher-order rewriting based on the nominal approach. A distinctive feature of nominal rewriting is that α-conversion and capture-avoiding substitution are not relegated to meta-level—they are explicitly dealt with at object-level. In contrast, previous higher-order rewriting frameworks employ some meta-level calculus (e.g. the simply-typed λ-calculus) and accomplish α-conversion and capture-avoiding substitution via the meta-level calculus.

Confluence and *critical pairs* are important in automated theorem proving and inductive theorem proving based on rewriting techniques such as Knuth-Bendix completion and implicit induction. Analysis of critical pairs in rewrite rules plays a fundamental role in those techniques. There are, however, problems in doing such analysis in the setting of nominal rewriting, since a rewrite system in [2, 3] is defined as an infinite set of rewrite rules that is closed under equivariance (i.e. bijective renaming of atoms).

This motivates us to propose in [10] a refined presentation of the nominal rewriting framework. While a rewrite system in [2, 3] is defined as an infinite set of rewrite rules, we define

a rewriting system as a finite set of rewrite rules. Then, instead of appealing to the property of equivariance, we specify a permutation as a parameter in each rewrite relation. Using this formulation, we studied in [10] confluence of orthogonal systems where no overlaps (and hence no critical pairs) exist in the rewrite rules. To solve the problem of checking the existence of overlaps, we used an equivariant unification procedure [1] with one permutation variable.

In the present paper, we study confluence properties of nominal rewriting systems in which overlaps of the rewrite rules exist. To this end, we need to investigate how to check joinability of critical pairs. This is not as easy as checking the existence of overlaps (critical pairs), since one has to examine whether there exist α-equivalent terms that are reachable from two terms in each critical pair, which may be parametrised by permutations. In this paper, we illustrate the process of checking joinability of critical pairs by giving some concrete examples.

The contributions of the paper are summarised as follows:

- First we present a natural example (Example 12) where choice of bound atoms of rules varies joinability of the induced critical pairs. This is a phenomenon that never appears in other higher-order rewriting frameworks in which α-equivalent terms are identified.
- We introduce two kinds of critical pairs with permutations as parameters, following the corresponding kinds of overlaps in [10]. Using them, we give a detailed proof of the critical pair lemma and concrete examples of checking joinability of critical pairs. Such examples have never been seen in the previous style of (infinitely many) critical pairs [2, 3] without parametrisation by permutations.
- Our proof of the critical pair lemma is more precise than the previous proof [2] in that we construct required permutations and substitutions, using some lemmas and the property of an mgu in the critical pair. Analysing our proof, we also present a critical pair lemma specialised for linear rewriting systems, which seems to be new.
- As applications of the critical pair lemmas, we give several confluence criteria which are used to obtain various confluence results including those for non-terminating systems.

The paper is organised is as follows. In Section 2, we explain basic notions and notations of nominal rewriting. In Section 3, we discuss problems on confluence in nominal rewriting, introduce our notion of critical pairs, and present applications of the critical pair lemmas. In Section 4, we conclude with suggestions for further work.

2 Nominal rewriting

Nominal rewriting [2, 3] is a framework that extends first-order term rewriting by a binding mechanism. In this section, we recall basic definitions on nominal terms and nominal rewriting based on [10]. For further descriptions and examples, see [2, 3, 10].

2.1 Nominal terms

A *nominal signature* Σ is a set of *function symbols* ranged over by $f, g, \ldots$. We fix a countably infinite set $\mathcal{X}$ of *variables* ranged over by $X, Y, Z, \ldots$, and a countably infinite set $\mathcal{A}$ of *atoms* ranged over by $a, b, c, \ldots$, and assume that Σ, $\mathcal{X}$, and $\mathcal{A}$ are pairwise disjoint. Unless otherwise stated, different meta-variables for objects in Σ, $\mathcal{X}$, or $\mathcal{A}$ denote different objects. A *swapping* is a pair of atoms, written $(a\ b)$. *Permutations* π are bijections on $\mathcal{A}$ such that the set of atoms for which $a \neq \pi(a)$ is finite. Permutations are represented by lists of swappings applied in the right-to-left order. For example, $((b\ c)(a\ b))(a) = c$, $((b\ c)(a\ b))(b) = a$, $((b\ c)(a\ b))(c) = b$. We write Id for the identity permutation, $()^{-1}$ for the inverse, and $\circ$ for the composition.

Nominal terms, or simply *terms*, are generated by the grammar

$$t, s ::= a \mid \pi{\cdot}X \mid [a]t \mid f\ t \mid \langle t_1, \ldots, t_n \rangle$$

and called, respectively, atoms, moderated variables, abstractions, function applications and tuples. We assume that function applications are bound more strongly than abstractions. We abbreviate $Id{\cdot}X$ as X if there is no ambiguity. $f\ \langle\,\rangle$ is abbreviated as f, and referred to as a *constant*. An abstraction $[a]t$ is intended to represent t with a bound. The set of *free* atoms occurring in t, denoted by $FA(t)$, is defined as follows: $FA(a) = \{a\}$; $FA(\pi{\cdot}X) = \emptyset$; $FA([a]t) = FA(t) \setminus \{a\}$; $FA(f\ t) = FA(t)$; $FA(\langle t_1, \ldots, t_n \rangle) = \bigcup_i FA(t_i)$. We write $V(t)(\subseteq \mathcal{X})$ for the set of variables occurring in t. A term t is said to be *ground* if $V(t) = \emptyset$. A *linear* term is a term in which any variable occurs at most once.

Example 1. A nominal signature for a fragment of first-order predicate logic has function symbols and, forall and p. The nominal term $\mathsf{and}\ \langle \mathsf{p}\ a, \mathsf{forall}\ [a]\mathsf{forall}\ [b]X \rangle$ represents the formula $p(a) \wedge \forall a.\forall b.X$ in the usual notation. Here X is a (meta-level) variable which can be instantiated by another term (representing a formula) possibly with free atoms a and b. For this term t, we have $FA(t) = \{a\}$ and $V(t) = \{X\}$.

Positions are finite sequences of positive integers. The empty sequence is denoted by ε. For positions p, q, we write $p \preceq q$ if there exists a position o such that $q = po$. We write $p \parallel q$ for $p \not\preceq q$ and $q \not\preceq p$. The set of positions in a term t, denoted by $Pos(t)$, is defined as follows: $Pos(a) = Pos(\pi{\cdot}X) = \{\varepsilon\}$; $Pos([a]t) = Pos(f\ t) = \{1p \mid p \in Pos(t)\} \cup \{\varepsilon\}$; $Pos(\langle t_1, \ldots, t_n \rangle) = \bigcup_i \{ip \mid p \in Pos(t_i)\} \cup \{\varepsilon\}$. The subterm of t at a position $p \in Pos(t)$ is written as $t|_p$. A position $p \in Pos(t)$ is a *variable position* in t if $t|_p$ is a moderated variable. The set of variable positions in t is denoted by $Pos_{\mathcal{X}}(t)$.

A *context* is a term in which a distinguished constant $\Box$ occurs. Contexts having precisely one $\Box$ are written as $C[\,]$. The term obtained from a context C by replacing each $\Box$ at positions p_i by terms t_i is written as $C[t_1, \ldots, t_n]_{p_1,\ldots,p_n}$ or simply $C[t_1, \ldots, t_n]$. Similarly, the term obtained from a term s by replacing each subterm at positions p_i by terms t_i is written as $s[t_1, \ldots, t_n]_{p_1,\ldots,p_n}$.

Next, we define two kinds of permutation actions $\pi{\cdot}t$ and t^π, which operate on terms extending a permutation on atoms. These actions are used to define substitution, α-equivalence and rewrite relation for nominal rewriting systems. They are defined as follows:

$$\begin{array}{rlrl}
\pi{\cdot}a &= \pi(a) & a^\pi &= \pi(a) \\
\pi{\cdot}(\pi'{\cdot}X) &= (\pi \circ \pi'){\cdot}X & (\pi'{\cdot}X)^\pi &= (\pi \circ \pi' \circ \pi^{-1}){\cdot}X \\
\pi{\cdot}([a]t) &= [\pi{\cdot}a](\pi{\cdot}t) & ([a]t)^\pi &= [a^\pi]t^\pi \\
\pi{\cdot}(f\ t) &= f\ \pi{\cdot}t & (f\ t)^\pi &= f\ t^\pi \\
\pi{\cdot}\langle t_1, \ldots, t_n \rangle &= \langle \pi{\cdot}t_1, \ldots, \pi{\cdot}t_n \rangle & \langle t_1, \ldots, t_n \rangle^\pi &= \langle t_1^\pi, \ldots, t_n^\pi \rangle
\end{array}$$

The difference between the two consists in the clause for moderated variables. In particular, when $\pi' = Id$, π is suspended before X in the first action as $\pi{\cdot}(Id{\cdot}X) = (\pi \circ Id){\cdot}X = \pi{\cdot}X$, while in the second action π has no effect as $(Id{\cdot}X)^\pi = (\pi \circ Id \circ \pi^{-1}){\cdot}X = Id{\cdot}X$.

A *substitution* is a map σ from variables to terms. Substitutions act on variables, without avoiding capture of atoms. We write $t\sigma$ for the application of σ on t. Note that by replacing X of a moderated variable $\pi{\cdot}X$ in t by $\sigma(X)$, a permutation action $\pi{\cdot}(\sigma(X))$ occurs. For a permutation π and a substitution σ, we define the substitution $\pi{\cdot}\sigma$ by $(\pi{\cdot}\sigma)(X) = \pi{\cdot}(\sigma(X))$.

2.2 α-equivalence and nominal rewriting systems

The distinctive feature of nominal rewriting is that it is equipped with a mechanism to avoid accidental capture of free atoms on the way of rewriting. This is partly achieved by α-conversion built in the matching process of the LHS of a rule and a redex involving also permutations (cf. Example 9).

In this subsection, we first recall the notion of α-equivalence in the nominal setting. This is different from α-equivalence in the traditional sense in that equivalence between terms is discussed under assumptions on the freshness of atoms in variables.

A pair $a\#t$ of an atom a and a term t is called a *freshness constraint*. Intuitively, this means that a does not occur as a free atom in t, including the cases where the variables in t are instantiated by other terms. A finite set $\nabla \subseteq \{a\#X \mid a \in \mathcal{A}, X \in \mathcal{X}\}$ is called a *freshness context*. For a freshness context ∇, we define $V(\nabla) = \{X \in \mathcal{X} \mid \exists a.\, a\#X \in \nabla\}$, $\nabla^\pi = \{a^\pi\#X \mid a\#X \in \nabla\}$, and $\nabla\sigma = \{a\#\sigma(X) \mid a\#X \in \nabla\}$.

The rules in Figure 1 define the relation $\nabla \vdash a\#t$, which means that $a\#t$ is satisfied under the freshness context ∇. It can be seen that $a \notin FA(t)$ whenever $\nabla \vdash a\#t$. An example using the last rule is $\{c\#X\} \vdash a\#((a\ b)(b\ c))\cdot X$, since $((a\ b)(b\ c))^{-1}\cdot a = ((b\ c)(a\ b))(a) = c$.

$$\frac{}{\nabla \vdash a\#b} \qquad \frac{\nabla \vdash a\#t}{\nabla \vdash a\#f\ t} \qquad \frac{\nabla \vdash a\#t_1 \quad \cdots \quad \nabla \vdash a\#t_n}{\nabla \vdash a\#\langle t_1, \ldots, t_n\rangle}$$

$$\frac{}{\nabla \vdash a\#[a]t} \qquad \frac{\nabla \vdash a\#t}{\nabla \vdash a\#[b]t} \qquad \frac{\pi^{-1}\cdot a\#X \in \nabla}{\nabla \vdash a\#\pi\cdot X}$$

Figure 1: Rules for freshness constraints

The rules in Figure 2 define the relation $\nabla \vdash t \approx_\alpha s$, which means that t is α-equivalent to s under the freshness context ∇. $ds(\pi, \pi')$ in the last rule denotes the set $\{a \in \mathcal{A} \mid \pi\cdot a \neq \pi'\cdot a\}$. For example, $ds((a\ b), Id) = \{a, b\}$.

$$\frac{}{\nabla \vdash a \approx_\alpha a} \qquad \frac{\nabla \vdash t \approx_\alpha s}{\nabla \vdash f\ t \approx_\alpha f\ s} \qquad \frac{\nabla \vdash t_1 \approx_\alpha s_1 \quad \cdots \quad \nabla \vdash t_n \approx_\alpha s_n}{\nabla \vdash \langle t_1, \ldots, t_n\rangle \approx_\alpha \langle s_1, \ldots, s_n\rangle}$$

$$\frac{\nabla \vdash t \approx_\alpha s}{\nabla \vdash [a]t \approx_\alpha [a]s} \qquad \frac{\nabla \vdash (a\ b)\cdot t \approx_\alpha s \quad \nabla \vdash b\#t}{\nabla \vdash [a]t \approx_\alpha [b]s} \qquad \frac{\forall a \in ds(\pi, \pi').\ a\#X \in \nabla}{\nabla \vdash \pi\cdot X \approx_\alpha \pi'\cdot X}$$

Figure 2: Rules for α-equivalence

Example 2. Consider the nominal signature for a fragment of first-order predicate logic in Example 1, and suppose $\nabla = \{a\#X, b\#X\}$. Then we have the following derivation:

$$\dfrac{\dfrac{\dfrac{a\#X \in \nabla}{\nabla \vdash a\#X} \quad \dfrac{b\#X \in \nabla}{\nabla \vdash b\#X}}{\nabla \vdash (a\ b)\cdot X \approx_\alpha X} \quad \dfrac{b\#X \in \nabla}{\nabla \vdash b\#X}}{\nabla \vdash [a]X \approx_\alpha [b]X}}{\nabla \vdash \mathsf{forall}\ [a]X \approx_\alpha \mathsf{forall}\ [b]X}$$

The following property is shown in [2].

Proposition 3 ([2])**.** *For any freshness context ∇, the binary relation $\nabla \vdash - \approx_\alpha -$ is a congruence (i.e. an equivalence relation that is closed under any context $C[\,]$).*

For ground terms, the last rules in Figures 1 and 2 are not necessary, and so the relation $\nabla \vdash - \approx_\alpha -$ is irrelevant to the freshness context ∇. In that case, the relation coincides with the usual α-equivalence (i.e. the relation reached by renamings of bound atoms) [4].

Now we define nominal rewrite rules and nominal rewriting systems.

Definition 4 (Nominal rewrite rule)**.** A *nominal rewrite rule*, or simply *rewrite rule*, is a triple of a freshness context ∇ and terms l and r such that $V(\nabla) \cup V(r) \subseteq V(l)$. We write $\nabla \vdash l \to r$ for a rewrite rule. A rewrite rule $\nabla \vdash l \to r$ is *linear* if so are l and r. For a rewrite rule $R = \nabla \vdash l \to r$ and a permutation π, we define the rewrite rule R^π as $\nabla^\pi \vdash l^\pi \to r^\pi$.

Example 5. Consider the nominal signature for a fragment of first-order predicate logic in Example 1. The following ($\mathsf{COM}_\forall$) is a nominal rewrite rule (we omit $\emptyset$ on the LHS of $\vdash$):

$$\vdash \quad \mathsf{forall}\ [a]\mathsf{forall}\ [b]X \to \mathsf{forall}\ [b]\mathsf{forall}\ [a]X \qquad (\mathsf{COM}_\forall)$$

Definition 6 (Nominal rewriting system)**.** A *nominal rewriting system*, or simply *rewriting system*, is a finite set of rewrite rules. A rewriting system is *linear* if so are all its rewrite rules.

Example 7. Consider the nominal signature for a fragment of first-order predicate logic in Example 1. A nominal rewriting system that represents logically equivalent transformations in first-order predicate logic:

$$\begin{array}{ll} (\forall a.X) \wedge Y \equiv \forall a.(X \wedge Y) & \text{where } a \notin \mathrm{FV}(Y) \\ X \wedge (\forall a.Y) \equiv \forall a.(X \wedge Y) & \text{where } a \notin \mathrm{FV}(X) \end{array}$$

is defined by the following $\mathcal{R}_{\mathsf{pnf}}$ (we omit the braces on the LHS of $\vdash$):

$$\mathcal{R}_{\mathsf{pnf}} = \left\{ \begin{array}{lll} a\#Y & \vdash & \mathsf{and}\ \langle \mathsf{forall}\ [a]X, Y\rangle \to \mathsf{forall}\ [a]\mathsf{and}\ \langle X, Y\rangle \qquad (\forall_1) \\ a\#X & \vdash & \mathsf{and}\ \langle X, \mathsf{forall}\ [a]Y\rangle \to \mathsf{forall}\ [a]\mathsf{and}\ \langle X, Y\rangle \qquad (\forall_2) \end{array} \right.$$

In [2, 3], nominal rewrite systems are defined as infinite sets of rewrite rules that are closed under equivariance, i.e., if R is a rule of a rewrite system $\mathcal{R}$ then so is R^π for any permutation π. In [10] and the present paper, rewriting systems are defined as finite sets of rewrite rules that may not be closed under equivariance. Accordingly, our rewrite relation is defined with a permutation as a parameter unlike in the definition of rewrite relation in [2, 3].

In the sequel, $\vdash$ is extended to mean to hold for all members of a set (sequence) on the RHS.

Definition 8 (Rewrite relation)**.** Let $R = \nabla \vdash l \to r$ be a rewrite rule. For a freshness context Δ and terms s and t, the *rewrite relation* is defined by

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

where w.l.o.g. $V(l) \cap (V(\Delta) \cup V(s)) = \emptyset$. We write $\Delta \vdash s \to_{\langle R,\pi\rangle} t$ if there exist p and σ such that $\Delta \vdash s \to_{\langle R,\pi,p,\sigma\rangle} t$. We write $\Delta \vdash s \to_R t$ if there exists π such that $\Delta \vdash s \to_{\langle R,\pi\rangle} t$. For a rewriting system $\mathcal{R}$, we write $\Delta \vdash s \to_{\mathcal{R}} t$ if there exists $R \in \mathcal{R}$ such that $\Delta \vdash s \to_R t$.

Example 9. Using the rule $(\forall_1)$ of the system $\mathcal{R}_{\mathsf{pnf}}$ in Example 7, we see that the term representing $(\forall a.p(a)) \wedge (\forall a.q(a))$ rewrites to $\forall a.(p(a) \wedge (\forall a.q(a)))$, that is, we have

$$\vdash \mathsf{and}\ \langle \mathsf{forall}\ [a]\mathsf{p}\ a, \mathsf{forall}\ [a]\mathsf{q}\ a\rangle \to_{\langle \forall_1, Id, \varepsilon, \sigma_1\rangle} \mathsf{forall}\ [a]\mathsf{and}\ \langle \mathsf{p}\ a, \mathsf{forall}\ [a]\mathsf{q}\ a\rangle$$

where σ_1 is the substitution $\{X := \mathsf{p}\ a, Y := \mathsf{forall}\ [a]\mathsf{q}\ a\}$. The resulting term rewrites further to the term $\mathsf{forall}\ [a]\mathsf{forall}\ [b]\mathsf{and}\ \langle \mathsf{p}\ a, \mathsf{q}\ b\rangle$ by the rule $(\forall_2)$ of the system $\mathcal{R}_{\mathsf{pnf}}$. Here we give a detail of the rewrite step to see how capture of a free atom is avoided.

Let $s = \mathsf{forall}\ [a]\mathsf{and}\ \langle \mathsf{p}\ a, \mathsf{forall}\ [a]\mathsf{q}\ a\rangle$. Since the rule has a freshness context $\nabla = \{a\#X\}$, to apply $(\forall_2)$ to s at the position $p = 11$, it is necessary to find a permutation π and a substitution σ_2 that satisfy $\vdash \nabla^\pi\sigma_2$ and $\vdash \mathsf{and}\ \langle \mathsf{p}\ a, \mathsf{forall}\ [a]\mathsf{q}\ a\rangle \approx_\alpha (\mathsf{and}\ \langle X, \mathsf{forall}\ [a]Y\rangle)^\pi\sigma_2$. Here one cannot simply take $\pi = Id$, because then $\sigma_2(X) = \mathsf{p}\ a$ from the condition for $\approx_\alpha$, which contradicts $\vdash \nabla^\pi\sigma_2$. So we take, e.g. $\pi = (a\ b)$ and $\sigma_2 = \{X := \mathsf{p}\ a, Y := \mathsf{q}\ b\}$ to satisfy the conditions, and get $\mathsf{forall}\ [a](\mathsf{forall}\ [a]\mathsf{and}\ \langle X, Y\rangle)^\pi\sigma_2 = \mathsf{forall}\ [a]\mathsf{forall}\ [b]\mathsf{and}\ \langle \mathsf{p}\ a, \mathsf{q}\ b\rangle$ as the result of rewriting.

In the following, a binary relation $\Delta \vdash - \bowtie -$ ($\bowtie$ is $\to_R$, $\approx_\alpha$, etc.) with a fixed freshness context Δ is called the relation $\bowtie$ under Δ, or simply the relation $\bowtie$ if there is no ambiguity. If a relation $\bowtie$ is written using $\to$ then the inverse is written using $\leftarrow$. Also, we write $\bowtie^=$ for the reflexive closure, and $\bowtie^*$ for the reflexive transitive closure. We use $\circ$ for the composition of relations. We write $\Delta \vdash s_1 \bowtie_1 s_2 \bowtie_2 \ldots \bowtie_{n-1} s_n$ for $\Delta \vdash s_i \bowtie_i s_{i+1}$ $(1 \le i < n)$. A rewriting system $\mathcal{R}$ is *terminating* if there is no infinite rewrite sequence $\Delta \vdash s_1 \to_\mathcal{R} s_2 \to_\mathcal{R} \cdots$.

3 Confluence and critical pairs in nominal rewriting

Having defined basic notions on nominal terms and nominal rewriting systems, we now set out to discuss confluence and critical pairs in nominal rewriting. To be exact, we study confluence properties modulo the equivalence relation $\approx_\alpha$ in terms of abstract reduction systems [7].

Definition 10. Let $\mathcal{R}$ be a nominal rewriting system.

1. s and t are *joinable modulo* $\approx_\alpha$ under a freshness context Δ, denoted by $\Delta \vdash s \downarrow_{\approx_\alpha} t$, iff $\Delta \vdash s\ (\to_\mathcal{R}^* \circ \approx_\alpha \circ \leftarrow_\mathcal{R}^*)\ t$.
2. $\mathcal{R}$ is *locally confluent modulo* $\approx_\alpha$ iff $\Delta \vdash s\ (\leftarrow_\mathcal{R} \circ \to_\mathcal{R})\ t$ implies $\Delta \vdash s \downarrow_{\approx_\alpha} t$.
3. $\mathcal{R}$ is *confluent modulo* $\approx_\alpha$ iff $\Delta \vdash s\ (\leftarrow_\mathcal{R}^* \circ \to_\mathcal{R}^*)\ t$ implies $\Delta \vdash s \downarrow_{\approx_\alpha} t$.
4. $\mathcal{R}$ is *Church-Rosser modulo* $\approx_\alpha$ iff $\Delta \vdash s\ (\leftarrow_\mathcal{R} \cup \to_\mathcal{R} \cup \approx_\alpha)^*\ t$ implies $\Delta \vdash s \downarrow_{\approx_\alpha} t$.
5. $\mathcal{R}$ is *strongly compatible with* $\approx_\alpha$ iff $\Delta \vdash s\ (\approx_\alpha \circ \to_\mathcal{R})\ t$ implies $\Delta \vdash s\ (\to_\mathcal{R}^= \circ \approx_\alpha)\ t$.

It is known that Church-Rosser modulo an equivalence relation $\sim$ is a stronger property than confluence modulo $\sim$ [7]. So we aim to prove Church-Rosser modulo $\approx_\alpha$ for some class of nominal rewriting systems.

The next proposition, which substitutes for Newman's Lemma in the usual rewriting, follows from [8, Propositions 2.5.4 & 2.5.12].

Proposition 11. *If $\mathcal{R}$ is terminating, locally confluent modulo $\approx_\alpha$ and strongly compatible with $\approx_\alpha$, then $\mathcal{R}$ is Church-Rosser modulo $\approx_\alpha$.*

3.1 Problems on confluence in nominal rewriting

Problems on confluence in nominal rewriting emerge in diverse ways. They are caused mainly by parametrisation of rewrite steps by permutations (or, in terms of [2], equivariance of the set of rewrite rules), which is not present in usual first-order and higher-order rewriting. In [10], we solved a problem on confluence of orthogonal systems by introducing the notion of α-stability. In the present paper, we treat another problem on confluence in nominal rewriting, which also concerns permutations as parameters on two rewrite rules that induce critical pairs.

Example 12. Consider the nominal rewriting system $\mathcal{R}_{\mathsf{pnfcom}}$ with the rewrite rules of $\mathcal{R}_{\mathsf{pnf}}$ in Example 7 and the rewrite rule ($\mathsf{COM}_\forall$) in Example 5:

$$\mathcal{R}_{\mathsf{pnfcom}} = \left\{ \begin{array}{lcll} a\#Y_1 & \vdash & \mathsf{and}\ \langle \mathsf{forall}\ [a]X_1, Y_1\rangle \to \mathsf{forall}\ [a]\mathsf{and}\ \langle X_1, Y_1\rangle & (\forall_1) \\ a\#X_2 & \vdash & \mathsf{and}\ \langle X_2, \mathsf{forall}\ [a]Y_2\rangle \to \mathsf{forall}\ [a]\mathsf{and}\ \langle X_2, Y_2\rangle & (\forall_2) \\ & \vdash & \mathsf{forall}\ [a]\mathsf{forall}\ [b]X \to \mathsf{forall}\ [b]\mathsf{forall}\ [a]X & (\mathsf{COM}_\forall) \end{array} \right.$$

Then the term $\mathsf{and}\ \langle \mathsf{forall}\ [a]X_1, \mathsf{forall}\ [a]Y_2\rangle$ can be reduced in two ways:

$$\vdash \mathsf{and}\ \langle \mathsf{forall}\ [a]X_1, \mathsf{forall}\ [a]Y_2\rangle \to_{\langle \forall_1, Id\rangle} \mathsf{forall}\ [a]\mathsf{and}\ \langle X_1, \mathsf{forall}\ [a]Y_2\rangle \quad (1\text{-}1)$$
$$\vdash \mathsf{and}\ \langle \mathsf{forall}\ [a]X_1, \mathsf{forall}\ [a]Y_2\rangle \to_{\langle \forall_2, Id\rangle} \mathsf{forall}\ [a]\mathsf{and}\ \langle \mathsf{forall}\ [a]X_1, Y_2\rangle \quad (1\text{-}2)$$

where the resulting two terms are not α-equivalent and they are normal forms in $\mathcal{R}_{\mathsf{pnfcom}}$. (To apply the rule ($\forall_2$) to the resulting term of (1-1) at the position 11, it is necessary to find a permutation π and a substitution σ that satisfy $\vdash (a\#X_2)^\pi\sigma$ and $\vdash \mathsf{and}\ \langle X_1, \mathsf{forall}\ [a]Y_2\rangle \approx_\alpha (\mathsf{and}\ \langle X_2, \mathsf{forall}\ [a]Y_2\rangle)^\pi\sigma$. The latter implies $\sigma(X_2) = X_1$, but then $\vdash (a\#X_2)^\pi\sigma\,(= a^\pi\#X_1)$ is not derivable by the rules in Figure 1 under the empty freshness context.) Hence we conclude that the rewriting system $\mathcal{R}_{\mathsf{pnfcom}}$ is not locally confluent modulo $\approx_\alpha$.

On the other hand, we have rewrite steps

$$a\#Y_2, b\#X_1 \vdash \mathsf{and}\ \langle \mathsf{forall}\ [a]X_1, \mathsf{forall}\ [b]Y_2\rangle \to_{\langle \forall_1, Id\rangle} \mathsf{forall}\ [a]\mathsf{and}\ \langle X_1, \mathsf{forall}\ [b]Y_2\rangle \quad (2\text{-}1)$$
$$a\#Y_2, b\#X_1 \vdash \mathsf{and}\ \langle \mathsf{forall}\ [a]X_1, \mathsf{forall}\ [b]Y_2\rangle \to_{\langle \forall_2, (a\ b)\rangle} \mathsf{forall}\ [b]\mathsf{and}\ \langle \mathsf{forall}\ [a]X_1, Y_2\rangle \quad (2\text{-}2)$$

where the resulting two terms are joinable modulo $\approx_\alpha$ under $\{a\#Y_2, b\#X_1\}$ as follows:

$$\begin{aligned} a\#Y_2, b\#X_1 \vdash \mathsf{forall}\ [a]\mathsf{and}\ \langle X_1, \mathsf{forall}\ [b]Y_2\rangle &\to_{\langle \forall_2, (a\ b)\rangle} \mathsf{forall}\ [a]\mathsf{forall}\ [b]\mathsf{and}\ \langle X_1, Y_2\rangle \\ a\#Y_2, b\#X_1 \vdash \mathsf{forall}\ [b]\mathsf{and}\ \langle \mathsf{forall}\ [a]X_1, Y_2\rangle &\to_{\langle \forall_1, Id\rangle} \mathsf{forall}\ [b]\mathsf{forall}\ [a]\mathsf{and}\ \langle X_1, Y_2\rangle \\ &\to_{\langle \mathsf{COM}_\forall, (a\ b)\rangle} \mathsf{forall}\ [a]\mathsf{forall}\ [b]\mathsf{and}\ \langle X_1, Y_2\rangle \end{aligned}$$

It turns out that the pairs of the resulting terms in (1-1) and (1-2), and in (2-1) and (2-2), together with the freshness contexts, form two different critical pairs induced from the rules ($\forall_1$) and ($\forall_2$) (cf. Example 18). Hence the above example indicates that choice of bound atoms in the same two rewrite rules can vary joinability of the induced critical pairs, and so we have to check all combinations of atoms in the rules to guarantee confluence properties. This kind of phenomenon never appears in other higher-order rewriting frameworks (e.g. [5, 6]) where α-equivalent terms are identified.

3.2 Basic critical pairs

To make the above problem more precise, we define our notions of overlaps and critical pairs. Here we introduce two kinds of critical pairs that are induced by the corresponding kinds of overlaps considered in [10].

First, we recall unification of nominal terms. Let P be a set of equations and freshness constraints $\{s_1 \approx t_1, \ldots, s_m \approx t_m, a_1\#u_1, \ldots, a_n\#u_n\}$ (where a_i and a_j may denote the same atom). Then, P is *unifiable* if there exist a freshness context Γ and a substitution θ such that $\Gamma \vdash s_1\theta \approx_\alpha t_1\theta, \ldots, s_m\theta \approx_\alpha t_m\theta, a_1\#u_1\theta, \ldots, a_n\#u_n\theta$; the pair $\langle \Gamma, \theta\rangle$ is called a *unifier* of P. It is shown in [13] that the unification problem for nominal terms is decidable. Moreover, if P is unifiable then there exists a *most general unifier* (*mgu* for short) of P, where an mgu of P is a unifier $\langle \Gamma, \theta\rangle$ of P such that for any unifier $\langle \Delta, \sigma\rangle$ of P, there exists a substitution δ such that $\Delta \vdash \Gamma\delta$ and $\Delta \vdash X\theta\delta \approx_\alpha X\sigma$ for any variable X.

Example 13. Consider the nominal signature for a fragment of first-order predicate logic.

1. Let $P = \{\mathsf{and}\ \langle \mathsf{forall}\ [a]X_1, Y_1\rangle \approx \mathsf{and}\ \langle X_2, \mathsf{forall}\ [a]Y_2\rangle, a\#Y_1, a\#X_2\}$. Then, $\langle \emptyset, \{X_2 := \mathsf{forall}\ [a]X_1, Y_1 := \mathsf{forall}\ [a]Y_2\rangle$ is a unifier of P. Furthermore, it is an mgu of P.
2. Let $P = \{\mathsf{and}\ \langle \mathsf{forall}\ [a]X_1, Y_1\rangle \approx \mathsf{and}\ \langle X_2, \mathsf{forall}\ [b]Y_2\rangle, a\#Y_1, b\#X_2\}$. Then, $\langle \{a\#Y_2, b\#X_1\}, \{X_2 := \mathsf{forall}\ [a]X_1, Y_1 := \mathsf{forall}\ [b]Y_2\}\rangle$ is a unifier of P. Furthermore, it is an mgu of P.

As in usual first-order and higher-order rewriting, our notion of overlap is defined in terms of unification, but involving also a permutation.

Definition 14 (Overlap)**.** Let $R_i = \nabla_i \vdash l_i \to r_i$ $(i = 1, 2)$ be rewrite rules. We assume w.l.o.g. that $V(l_1) \cap V(l_2) = \emptyset$. If $\nabla_1 \cup \nabla_2^{\pi_2} \cup \{l_1 \approx l_2^{\pi_2}|_p\}$ is unifiable for some permutation π_2 and a non-variable position p, 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 $p = \varepsilon$, then the overlap is said to be *self-rooted*. An overlap that is not self-rooted is said to be *proper*.

Example 15. Consider the rules $(\forall_1)$ $a\#Y_1 \vdash \mathsf{and}\ \langle \mathsf{forall}\ [a]X_1, Y_1\rangle \to \mathsf{forall}\ [a]\mathsf{and}\ \langle X_1, Y_1\rangle$ and $(\forall_2)$ $a\#X_2 \vdash \mathsf{and}\ \langle X_2, \mathsf{forall}\ [a]Y_2\rangle \to \mathsf{forall}\ [a]\mathsf{and}\ \langle X_2, Y_2\rangle$ from Example 12. Then, $(\forall_1)$ overlaps on $(\forall_2)$ in the following two ways.

1. $\{a\#Y_1\} \cup \{a\#X_2\}^{Id} \cup \{\mathsf{and}\ \langle \mathsf{forall}\ [a]X_1, Y_1\rangle \approx (\mathsf{and}\ \langle X_2, \mathsf{forall}\ [a]Y_2\rangle)^{Id}|_\varepsilon\}$ is unifiable as seen in Example 13 (1). This overlap is proper.
2. $\{a\#Y_1\} \cup \{a\#X_2\}^{(a\ b)} \cup \{\mathsf{and}\ \langle \mathsf{forall}\ [a]X_1, Y_1\rangle \approx (\mathsf{and}\ \langle X_2, \mathsf{forall}\ [a]Y_2\rangle)^{(a\ b)}|_\varepsilon\}$ is unifiable as seen in Example 13 (2). This overlap is proper.

Example 16. There exist self-rooted overlaps of the rule $(\forall_1)$ on its renamed variant, since $P = \{\mathsf{and}\ \langle \mathsf{forall}\ [a]X_1, Y_1\rangle \approx (\mathsf{and}\ \langle \mathsf{forall}\ [a]Z_1, W_1\rangle)^\pi\}$ is unifiable for any permutation π. In the case of $\pi(a) = b$, we take $\langle \{a\#Z_1\}, \{X_1 := (a\ b)\cdot Z_1, Y_1 := W_1\}\rangle$ as an mgu of P.

An overlap always gives rise to a critical pair, which we call here a basic critical pair to distinguish from the critical pair [2, 3] that is defined without parametrisation by a permutation.

Definition 17 (Basic critical pair)**.** Let $R_i = \nabla_i \vdash l_i \to r_i$ $(i = 1, 2)$ be rewrite rules. We assume w.l.o.g. that $V(l_1) \cap V(l_2) = \emptyset$. Let $\nabla_1 \cup \nabla_2^{\pi_2} \cup \{l_1 \approx l_2^{\pi_2}|_p\}$ be unifiable for some permutation π_2 and a non-variable position p such that $l_2 = L[l_2|_p]_p$, and let $\langle \Gamma, \theta\rangle$ be an mgu. Then, $\Gamma \vdash \langle L^{\pi_2}\theta[r_1\theta]_p, r_2^{\pi_2}\theta\rangle$ is called a *basic critical pair* (*BCP* for short) of R_1 and R_2. A BCP induced by a self-rooted (proper) overlap is said to be *self-rooted* (*proper*, respectively).

$BCP(R_1, R_2)$ denotes the set of all BCPs of R_1 and R_2, and $BCP(\mathcal{R})$ denotes the set $\bigcup_{R_i, R_j \in \mathcal{R}} BCP(R_i, R_j)$. Similarly, $PBCP(\mathcal{R})$ denotes the set of all proper BCPs of $\mathcal{R}$.

We remark that any BCP $\Gamma \vdash \langle L^{\pi_2}\theta[r_1\theta]_p, r_2^{\pi_2}\theta\rangle$ of R_1 and R_2 forms a peak, i.e., we have $\Gamma \vdash L^{\pi_2}\theta[r_1\theta]_p \leftarrow_{\langle R_1, Id, p, \theta\rangle} L^{\pi_2}\theta[l_2^{\pi_2}|_p\theta]_p = (L[l_2|_p]_p)^{\pi_2}\theta = l_2^{\pi_2}\theta \to_{\langle R_2, \pi_2, \varepsilon, \theta\rangle} r_2^{\pi_2}\theta$.

Example 18. The proper BCPs induced by the proper overlaps in Example 15 (1) and (2) are, respectively,

1. $\vdash \langle \mathsf{forall}\ [a]\mathsf{and}\ \langle X_1, \mathsf{forall}\ [a]Y_2\rangle, \mathsf{forall}\ [a]\mathsf{and}\ \langle \mathsf{forall}\ [a]X_1, Y_2\rangle\rangle$
2. $a\#Y_2, b\#X_1 \vdash \langle \mathsf{forall}\ [a]\mathsf{and}\ \langle X_1, \mathsf{forall}\ [b]Y_2\rangle, \mathsf{forall}\ [b]\mathsf{and}\ \langle \mathsf{forall}\ [a]X_1, Y_2\rangle\rangle$

These are the BCPs that were mentioned in Example 12.

Example 19. The self-rooted BCP induced by the self-rooted overlap in Example 16 is $a\#Z_1 \vdash \langle \mathsf{forall}\ [a]\mathsf{and}\ \langle (a\ b)\cdot Z_1, W_1\rangle, \mathsf{forall}\ [b]\mathsf{and}\ \langle Z_1, W_1\rangle\rangle$.

In the rest of the paper, we are concerned with confluence properties for particular classes of nominal rewriting systems. For this, we restrict rewriting systems by some conditions. First we consider the uniformity condition [2], and present critical pair lemma for uniform rewriting systems, which is a central tool for proving confluence of rewriting systems with critical pairs. 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. We employ the following definition of uniformity which is equivalent to the one in [2].

Definition 20 (Uniformity)**.** A rewrite rule $\nabla \vdash l \to r$ is *uniform* if for any atom a and any freshness context Δ, $\Delta \vdash \nabla$ and $\Delta \vdash a\#l$ imply $\Delta \vdash a\#r$. A rewriting system is *uniform* if so are all its rewrite rules.

The following property of uniform rewrite rules is important and will be used in the sequel. (For the proof, see [12].)

Lemma 21. *Let R be a uniform rewrite rule. If $\Delta \vdash s' \approx_\alpha s \to_{\langle R,\pi,p,\sigma\rangle} t$, then there exist π', σ', t' such that $\Delta \vdash s' \to_{\langle R,\pi',p,\sigma'\rangle} t' \approx_\alpha t$.*

The following is an immediate consequence of Lemma 21.

Corollary 22. *Uniform rewriting systems are strongly compatible with $\approx_\alpha$.*

Now we state the critical pair lemma. (For the proof, see [12].)

Lemma 23 (Critical pair lemma)**.** *Let $\mathcal{R}$ be a uniform rewriting system, and let $R_1, R_2 \in \mathcal{R}$. If $\Delta \vdash t_1 \leftarrow_{R_1} s \to_{R_2} t_2$ then one of the following holds:*

1. *There exist terms t_1' and t_2' such that either $\Delta \vdash t_1 \ (\to_{R_1}^* \circ \to_{R_2}) \ t_1' \approx_\alpha t_2' \leftarrow_{R_1}^* t_2$ or $\Delta \vdash t_1 \to_{R_2}^* t_1' \approx_\alpha t_2' \ (\leftarrow_{R_1} \circ \leftarrow_{R_2}^*) \ t_2$.*
2. *There exist $\Gamma \vdash \langle u, v\rangle \in BCP(R_1, R_2) \cup BCP(R_2, R_1)$, π, θ and $C[\,]$ such that $\Delta \vdash \Gamma^\pi\theta$, $\Delta \vdash t_1 \approx_\alpha C[u^\pi\theta]$ and $\Delta \vdash t_2 \approx_\alpha C[v^\pi\theta]$.*

3.3 Confluence criteria

As corollaries of the critical pair lemma, we obtain several criteria for confluence properties of uniform rewriting systems. In this subsection, we present those criteria and their applications.

First we state a proposition which is not difficult to show (cf. [11]).

Proposition 24. *If $\Gamma \vdash s \bowtie t$ and $\Delta \vdash \Gamma^\pi\theta$ then $\Delta \vdash C[s^\pi\theta] \bowtie C[t^\pi\theta]$ ($\bowtie$ is $\approx_\alpha$ or $\to_R$).*

Based on the critical pair lemma, the following theorem provides a necessary and sufficient condition for local confluence modulo $\approx_\alpha$ of uniform rewriting systems.

Theorem 25. *Let $\mathcal{R}$ be a uniform nominal rewriting system. Then $\mathcal{R}$ is locally confluent modulo $\approx_\alpha$ if and only if $\Gamma \vdash u \downarrow_{\approx_\alpha} v$ for any $\Gamma \vdash \langle u, v\rangle \in BCP(\mathcal{R})$.*

Proof. ($\Rightarrow$) follows from the remark after Definition 17. ($\Leftarrow$) Suppose $\Delta \vdash t_1 \leftarrow_{R_1} s \to_{R_2} t_2$. Then, one of 1 and 2 in Lemma 23 holds. In the case of 2, we have some $\Gamma \vdash \langle u, v\rangle \in BCP(\mathcal{R})$ such that $\Delta \vdash \Gamma^\pi\theta$, $\Delta \vdash t_1 \approx_\alpha C[u^\pi\theta]$ and $\Delta \vdash t_2 \approx_\alpha C[v^\pi\theta]$. Then, from the assumption, we have $\Gamma \vdash u \downarrow_{\approx_\alpha} v$. Hence, using Lemma 21 and Proposition 24, we obtain $\Delta \vdash t_1 \downarrow_{\approx_\alpha} t_2$. □

If we restrict the class of rewriting systems to α-stable ones [10], then it suffices to consider PBCPs (proper BCPs) instead of BCPs in the condition of Theorem 25.

Definition 26 (α-stability)**.** A rewrite rule $R = \nabla \vdash l \to r$ is *α-stable* if $\Delta \vdash s \approx_\alpha s'$, $\Delta \vdash s \to_{\langle R,\pi,\varepsilon,\sigma\rangle} t$ and $\Delta \vdash s' \to_{\langle R,\pi',\varepsilon,\sigma'\rangle} t'$ imply $\Delta \vdash t \approx_\alpha t'$. A rewriting system $\mathcal{R}$ is *α-stable* if so are all its rewrite rules.

Theorem 27. *Let $\mathcal{R}$ be a uniform α-stable nominal rewriting system. Then $\mathcal{R}$ is locally confluent modulo $\approx_\alpha$ if and only if $\Gamma \vdash u \downarrow_{\approx_\alpha} v$ for any $\Gamma \vdash \langle u, v\rangle \in PBCP(\mathcal{R})$.*

Proof. ($\Rightarrow$) By Theorem 25. ($\Leftarrow$) If $\Gamma \vdash \langle u, v\rangle \in BCP(\mathcal{R})$ in the proof of Theorem 25 is not a PBCP, then the claim follows by the α-stability of $\mathcal{R}$. □

In [10], we have given a sufficient condition for uniformity and α-stability of rewrite rules, called *abstract skeleton preserving* (*ASP* for short). It is easy to judge whether a rewrite rule is ASP or not (cf. [10]). All examples of the rewrite rules in the present paper are ASP, and so in the rest of the paper, we focus our attention on uniform and α-stable rewriting systems.

Now we arrive at our Knuth-Bendix criterion for nominal rewriting systems.

Corollary 28. *Let $\mathcal{R}$ be a terminating uniform α-stable nominal rewriting system. Then $\mathcal{R}$ is Church-Rosser modulo $\approx_\alpha$ if and only if $\Gamma \vdash u \downarrow_{\approx_\alpha} v$ for any $\Gamma \vdash \langle u, v\rangle \in PBCP(\mathcal{R})$.*

Proof. ($\Rightarrow$) follows from Theorem 27, since Church-Rosser modulo $\approx_\alpha$ implies local confluence modulo $\approx_\alpha$. ($\Leftarrow$) is by Proposition 11, Corollary 22 and Theorem 27. □

Example 29. Consider a nominal signature for a fragment of first-order predicate logic with function symbols not, forall and exists. The following rewriting system $\mathcal{R}_{\mathsf{nnf}}$ computes negation normal forms.

$$\mathcal{R}_{\mathsf{nnf}} = \left\{ \begin{array}{lll} \vdash & \mathsf{not}\ \mathsf{forall}\ [a]X \to \mathsf{exists}\ [a]\mathsf{not}\ X & (\mathsf{dM}_\forall) \\ \vdash & \mathsf{not}\ \mathsf{exists}\ [a]X \to \mathsf{forall}\ [a]\mathsf{not}\ X & (\mathsf{dM}_\exists) \\ \vdash & \mathsf{not}\ \mathsf{not}\ X \to X & (\mathsf{DNE}) \end{array} \right.$$

There exist three PBCPs of $\mathcal{R}_{\mathsf{nnf}}$, each of which is induced by a proper overlap of each rule on the rule (DNE).

$$PBCP(\mathcal{R}_{\mathsf{nnf}}) = \left\{ \begin{array}{ll} \vdash & \langle \mathsf{not}\ \mathsf{exists}\ [a]\mathsf{not}\ X, \mathsf{forall}\ [a]X\rangle \\ \vdash & \langle \mathsf{not}\ \mathsf{forall}\ [a]\mathsf{not}\ X, \mathsf{exists}\ [a]X\rangle \\ \vdash & \langle \mathsf{not}\ X, \mathsf{not}\ X\rangle \end{array} \right.$$

It is seen that all the PBCPs are joinable modulo $\approx_\alpha$. For the first PBCP, we have

$$\vdash \mathsf{not}\ \mathsf{exists}\ [a]\mathsf{not}\ X \to_{\mathsf{dM}_\exists} \mathsf{forall}\ [a]\mathsf{not}\ \mathsf{not}\ X \to_{\mathsf{DNE}} \mathsf{forall}\ [a]X$$

and similarly for the second PBCP. Termination of $\mathcal{R}_{\mathsf{nnf}}$ can be shown as follows: define a map h by $h(a) = h(\pi{\cdot}X) = 1$; $h([a]t) = h(t)$; $h(\mathsf{not}\ t) = h(t) \times 2$; $h(f\ t) = h(t) + 1$ for $f \neq \mathsf{not}$; $h(\langle t_1, \ldots, t_n\rangle) = h(t_1) + \cdots + h(t_n) + 1$, and observe that if $\Delta \vdash s \to_{\mathcal{R}_{\mathsf{nnf}}} t$ then $h(s) > h(t)$. Hence by Corollary 28, $\mathcal{R}_{\mathsf{nnf}}$ is Church-Rosser modulo $\approx_\alpha$.

In the above example, the set of PBCPs is finite since the rule (DNE) has no atoms, but in general, PBCPs involve permutations as parameters (cf. Examples 33 and 34).

Next we present a confluence criterion for linear rewriting systems, which works not only for terminating systems but also for non-terminating ones. Observing the proof of Lemma 23, we can specialise the critical pair lemma for linear rewriting systems.

Lemma 30 (Critical pair lemma for linear systems)**.** *Let $\mathcal{R}$ be a uniform linear rewriting system, and let $R_1, R_2 \in \mathcal{R}$. If $\Delta \vdash t_1 \leftarrow_{R_1} s \to_{R_2} t_2$ then one of the following holds:*

1. *There exist terms* t_1' *and* t_2' *such that* $\Delta \vdash t_1 \to^{=}_{R_2} t_1' \approx_\alpha t_2' \leftarrow^{=}_{R_1} t_2$.
2. *There exist* $\Gamma \vdash \langle u, v\rangle \in BCP(R_1, R_2) \cup BCP(R_2, R_1)$, π, θ *and* $C[\,]$ *such that* $\Delta \vdash \Gamma^\pi\theta$, $\Delta \vdash t_1 \approx_\alpha C[u^\pi\theta]$ *and* $\Delta \vdash t_2 \approx_\alpha C[v^\pi\theta]$.

The criterion is obtained via strong local confluence modulo $\approx_\alpha$.

Definition 31. A nominal rewriting system $\mathcal{R}$ is *strongly locally confluent modulo* $\approx_\alpha$ if $\Delta \vdash s\ (\leftarrow_\mathcal{R} \circ \to_\mathcal{R})\ t$ implies $\Delta \vdash s\ (\to^{=}_\mathcal{R} \circ \approx_\alpha \circ \leftarrow^{*}_\mathcal{R})\ t$.

Theorem 32. *Let* $\mathcal{R}$ *be a uniform* α*-stable linear rewriting system. Then* $\mathcal{R}$ *is Church-Rosser modulo* $\approx_\alpha$ *if* $\Gamma \vdash u\ (\to^{=}_\mathcal{R} \circ \approx_\alpha \circ \leftarrow^{*}_\mathcal{R})\ v$ *and* $\Gamma \vdash u\ (\to^{*}_\mathcal{R} \circ \approx_\alpha \circ \leftarrow^{=}_\mathcal{R})\ v$ *for any* $\Gamma \vdash \langle u, v\rangle \in PBCP(\mathcal{R})$.

Proof. By Corollary 22, $\mathcal{R}$ is strongly compatible with $\approx_\alpha$. Also, in a similar way to the proofs of Theorems 25 and 27, we see that $\mathcal{R}$ is strongly locally confluent modulo $\approx_\alpha$, using Lemma 30 instead of Lemma 23. Then, by the results in [7] (see also [8, Section 2.5]), $\mathcal{R}$ is Church-Rosser modulo $\approx_\alpha$. □

Example 33. Consider the linear rewriting system $\mathcal{R}_{\mathsf{com}}$ with the only rewrite rule ($\mathsf{COM}_\forall$):

$$\mathcal{R}_{\mathsf{com}} = \{\quad \vdash\ \ \mathsf{forall}\ [a]\mathsf{forall}\ [b]X \to \mathsf{forall}\ [b]\mathsf{forall}\ [a]X \qquad (\mathsf{COM}_\forall)$$

PBCPs of $\mathcal{R}_{\mathsf{com}}$ are induced by overlaps of ($\mathsf{COM}_\forall$) on its renamed variant, all of which arise from the unification problem $\{\mathsf{forall}\ [a]\mathsf{forall}\ [b]X \approx (\mathsf{forall}\ [a]\mathsf{forall}\ [b]Y)^\pi|_{11} (= \mathsf{forall}\ [\pi(b)]Y)\}$. In the following, we write down all patterns of the PBCPs according to the permutation π (we abbreviate $\mathsf{forall}\ [a]t$ as $\forall[a](t)$).

1. Case $\pi(b) = a$. Then the problem $\{\forall[a]\forall[b]X \approx \forall[a]Y\}$ has an mgu $\langle\emptyset, \{Y := \forall[b]X\}\rangle$. Hence, the pattern of PBCPs in this case is $\vdash \langle\forall[\pi(a)]\forall[b]\forall[a]X, \forall[a]\forall[\pi(a)]\forall[b]X\rangle$, for which we have
$$\vdash \forall[\pi(a)]\forall[b]\forall[a]X \to^{=}_{\mathcal{R}_{\mathsf{com}}} \forall[\pi(a)]\forall[a]\forall[b]X \leftarrow^{=}_{\mathcal{R}_{\mathsf{com}}} \forall[a]\forall[\pi(a)]\forall[b]X$$
2. Case $\pi(b) = b$. Then the problem $\{\forall[a]\forall[b]X \approx \forall[b]Y\}$ has an mgu $\langle\emptyset, \{Y := \forall[a](a\ b)\cdot X\}\rangle$. Hence, the pattern of PBCPs in this case is $\vdash \langle\forall[\pi(a)]\forall[b]\forall[a]X, \forall[b]\forall[\pi(a)]\forall[a](a\ b)\cdot X\rangle$, for which we have
$$\begin{aligned}\vdash \forall[\pi(a)]\forall[b]\forall[a]X &\to^{=}_{\mathcal{R}_{\mathsf{com}}} \forall[\pi(a)]\forall[a]\forall[b]X\\ &\approx_\alpha\ \ \forall[\pi(a)]\forall[b]\forall[a](a\ b)\cdot X \leftarrow^{=}_{\mathcal{R}_{\mathsf{com}}} \forall[b]\forall[\pi(a)]\forall[a](a\ b)\cdot X\end{aligned}$$
3. Case $\pi(b) = c$. Then the problem $\{\forall[a]\forall[b]X \approx \forall[c]Y\}$ has an mgu $\langle\{c\#X\}, \{Y := \forall[b](a\ c)\cdot X\}\rangle$. Hence, the pattern of PBCPs in this case is $c\#X \vdash \langle\forall[\pi(a)]\forall[b]\forall[a]X, \forall[c]\forall[\pi(a)]\forall[b](a\ c)\cdot X\rangle$, for which we have
$$\begin{aligned}c\#X \vdash \forall[\pi(a)]\forall[b]\forall[a]X &\to^{=}_{\mathcal{R}_{\mathsf{com}}} \forall[\pi(a)]\forall[a]\forall[b]X\\ &\approx_\alpha\ \ \forall[\pi(a)]\forall[c]\forall[b](a\ c)\cdot X \leftarrow^{=}_{\mathcal{R}_{\mathsf{com}}} \forall[c]\forall[\pi(a)]\forall[b](a\ c)\cdot X\end{aligned}$$

Hence by Theorem 32, $\mathcal{R}_{\mathsf{com}}$ is Church-Rosser modulo $\approx_\alpha$.

Now, let us revisit Example 12 in Subsection 3.1. One of the reasons why the resulting two terms in (1-1) and (1-2) are not joinable modulo $\approx_\alpha$ is that they cannot be reduced further as freshness constraints including variables are not satisfied under the empty freshness context. If we restrict, however, rewrite relations to those on ground terms, such situation can be avoided.

For ground terms s and t, the relation $\Delta \vdash s \to_{\langle R,\pi,p,\sigma\rangle} t$ is irrelevant to Δ (cf. the remark after Proposition 3). So we simply write $s \to_{\langle R,\pi,p,\sigma\rangle} t$ etc. instead of $\Delta \vdash s \to_{\langle R,\pi,p,\sigma\rangle} t$ etc. A rewriting system $\mathcal{R}$ is *ground Church-Rosser modulo* $\sim$ if the relation $\to_{\mathcal{R}}$ on ground terms satisfies $(\leftarrow_{\mathcal{R}} \cup \to_{\mathcal{R}} \cup \sim)^* \subseteq (\to_{\mathcal{R}}^* \circ \sim \circ \leftarrow_{\mathcal{R}}^*)$. The other properties in Definition 10 are also defined on ground terms. In the next example, we assume that each relation is on ground terms.

Example 34. Let $\mathcal{R}_{\mathsf{pnfcom}}$ be the linear rewriting system in Example 12. Our aim is to prove that $\mathcal{R}_{\mathsf{pnfcom}}$ is ground Church-Rosser modulo $\approx_\alpha$. For this, define an additional symmetric relation $\mapsto$ by $\leftarrow_{\mathsf{COM}_\forall} \cup \to_{\mathsf{COM}_\forall}$ and an equivalence relation $\sim$ by $(\mapsto \cup \approx_\alpha)^*$. First we show that the system $\mathcal{R}_{\mathsf{pnf}}$ $(= \{(\forall_1), (\forall_2)\})$ is ground Church-Rosser modulo $\sim$, using [8, Corollary 2.6.10].

1. $\mathcal{R}_{\mathsf{pnf}}$ is terminating w.r.t. $\sim \circ \to_{\mathcal{R}_{\mathsf{pnf}}} \circ \sim$. This is seen by defining a map h by $h(a) = 1$; $h([a]t) = h(t)$; $h(\mathsf{and}\ t) = h(t) \times 2$; $h(f\ t) = h(t) + 1$ for $f \neq \mathsf{and}$; $h(\langle t_1, \ldots, t_n\rangle) = h(t_1) + \cdots + h(t_n) + 1$, and observing that if $s\ (\sim \circ \to_{\mathcal{R}_{\mathsf{pnf}}} \circ \sim)\ t$ then $h(s) > h(t)$.
2. $\mathcal{R}_{\mathsf{pnf}}$ is locally confluent modulo $\sim$. This is shown using the critical pair lemma as follows. Suppose $t_1 \leftarrow_{\mathcal{R}_{\mathsf{pnf}}} s \to_{\mathcal{R}_{\mathsf{pnf}}} t_2$. Then, one of 1 and 2 in Lemma 30 holds. In the case of 2, we have some $\Gamma \vdash \langle u, v\rangle \in \mathit{BCP}(\mathcal{R}_{\mathsf{pnf}})$ such that $\vdash \Gamma^\pi\theta$, $\vdash t_1 \approx_\alpha C[u^\pi\theta]$ and $\vdash t_2 \approx_\alpha C[v^\pi\theta]$. If $\Gamma \vdash \langle u, v\rangle$ is not a PBCP, then the claim follows by the α-stability of $\mathcal{R}_{\mathsf{pnf}}$. Otherwise, PBCPs are those as in Example 18 or their symmetric ones. Now, let us consider the first PBCP $\vdash \langle \mathsf{forall}\ [a]\mathsf{and}\ \langle X_1, \mathsf{forall}\ [a]Y_2\rangle, \mathsf{forall}\ [a]\mathsf{and}\ \langle \mathsf{forall}\ [a]X_1, Y_2\rangle\rangle$ in Example 18 (the second PBCP is joinable as in Example 12). Let s_1, s_2 be any ground terms and c be an atom that does not occur in s_1, s_2. Then $\vdash c\#s_1, c\#s_2$ holds, so we have rewrite steps

$$\begin{aligned}
\mathsf{forall}\ [a]\mathsf{and}\ \langle s_1, \mathsf{forall}\ [a]s_2\rangle &\to_{\langle \forall_2, (a\ c)\rangle} && \mathsf{forall}\ [a]\mathsf{forall}\ [c]\mathsf{and}\ \langle s_1, (a\ c)\cdot s_2\rangle \\
&\to_{\langle \mathsf{COM}_\forall, (b\ c)\rangle} && \mathsf{forall}\ [c]\mathsf{forall}\ [a]\mathsf{and}\ \langle s_1, (a\ c)\cdot s_2\rangle \\
&\approx_\alpha && \mathsf{forall}\ [a]\mathsf{forall}\ [c]\mathsf{and}\ \langle (a\ c)\cdot s_1, s_2\rangle \\
&\leftarrow_{\langle \forall_1, (a\ c)\rangle} && \mathsf{forall}\ [a]\mathsf{and}\ \langle \mathsf{forall}\ [a]s_1, s_2\rangle
\end{aligned}$$

   Thus we see $C[u^\pi\theta] \downarrow_\sim C[v^\pi\theta]$, and by Lemma 21, $t_1 \downarrow_\sim t_2$.
3. $\mathcal{R}_{\mathsf{pnf}}$ is locally coherent with $\mapsto$, i.e., if $t_1\ (\mapsto \circ \to_{\mathcal{R}_{\mathsf{pnf}}})\ t_2$ then $t_1 \downarrow_\sim t_2$. This is shown again using the critical pair lemma. Suppose $t_1 \leftarrow_{\mathsf{COM}_\forall} s \to_{\mathcal{R}_{\mathsf{pnf}}} t_2$. We proceed as in the previous item, and consider PBCPs induced by overlaps of $(\mathsf{COM}_\forall)$ on $(\forall_1)$ (or $(\forall_2)$ symmetrically). They arise from the unification problem $\{\pi(a)\#Y_1, \mathsf{forall}\ [a]\mathsf{forall}\ [b]X \approx \mathsf{forall}\ [\pi(a)]X_1\}$. In the following, we write down all PBCPs according to the permutation π (we abbreviate $\mathsf{forall}\ [a]t$ as $\forall[a]t$).
   (a) Case $\pi(a) = a$. Then the problem has an mgu $\langle\{a\#Y_1\}, \{X_1 := \forall[b]X\}\rangle$. Hence, the PBCP in this case is $a\#Y_1 \vdash \langle \mathsf{and}\ \langle \forall[b]\forall[a]X, Y_1\rangle, \forall[a]\mathsf{and}\ \langle \forall[b]X, Y_1\rangle\rangle$.
   (b) Case $\pi(a) = b$. Then the problem has an mgu $\langle\{b\#Y_1\}, \{X_1 := \forall[a](a\ b)\cdot X\}\rangle$. Hence, the PBCP in this case is $b\#Y_1 \vdash \langle \mathsf{and}\ \langle \forall[b]\forall[a]X, Y_1\rangle, \forall[b]\mathsf{and}\ \langle \forall[a](a\ b)\cdot X, Y_1\rangle\rangle$.
   (c) Case $\pi(a) = c$. Then the problem has an mgu $\langle\{c\#Y_1, c\#X\}, \{X_1 := \forall[b](a\ c)\cdot X\}\rangle$. Hence, the PBCP is $c\#Y_1, c\#X \vdash \langle \mathsf{and}\ \langle \forall[b]\forall[a]X, Y_1\rangle, \forall[c]\mathsf{and}\ \langle \forall[b](a\ c)\cdot X, Y_1\rangle\rangle$.

   Then, by a similar argument to the last part of item 2, we obtain $t_1 \downarrow_\sim t_2$ in each case.

From 1, 2 and 3, it follows by [8, Corollary 2.6.10] that $\mathcal{R}_{\mathsf{pnf}}$ is ground Church-Rosser modulo $\sim$, which means $(\leftarrow_{\mathcal{R}_{\mathsf{pnf}}} \cup \to_{\mathcal{R}_{\mathsf{pnf}}} \cup \mapsto \cup \approx_\alpha)^* \subseteq (\to_{\mathcal{R}_{\mathsf{pnf}}}^* \circ (\mapsto \cup \approx_\alpha)^* \circ \leftarrow_{\mathcal{R}_{\mathsf{pnf}}}^*)$. From this and Church-Rosser modulo $\approx_\alpha$ of $\mathcal{R}_{\mathsf{com}}$ (cf. Example 33), we obtain $(\leftarrow_{\mathcal{R}_{\mathsf{pnfcom}}} \cup \to_{\mathcal{R}_{\mathsf{pnfcom}}} \cup \approx_\alpha)^* \subseteq (\to_{\mathcal{R}_{\mathsf{pnfcom}}}^* \circ \approx_\alpha \circ \leftarrow_{\mathcal{R}_{\mathsf{pnfcom}}}^*)$, which means that $\mathcal{R}_{\mathsf{pnfcom}}$ is ground Church-Rosser modulo $\approx_\alpha$.

4 Conclusion

Using our notion of critical pairs, we have presented several confluence criteria as applications of the critical pair lemmas. We have obtained some confluence results for concrete examples of nominal rewriting systems, illustrating the process of checking joinability of critical pairs.

In future work, we are going to implement a procedure for checking joinability of critical pairs for automated confluence proving. Such an effort is expected to be useful in developing automated theorem proving techniques like Knuth-Bendix completion. Confluence criteria for non-terminating left-linear systems with critical pairs are also to be investigated.

Acknowledgements We thank the anonymous referees for useful comments. This research was supported by JSPS KAKENHI Grant Numbers 25330004, 25280025 and 15K00003.

References

[1] J. Cheney. Equivariant unification. *J. of Automated Reasoning*, 45:267–300, 2010.

[2] M. Fernández and M. J. Gabbay. Nominal rewriting. *Inform. and Comput.*, 205:917–965, 2007.

[3] M. Fernández, M. J. Gabbay, and I. Mackie. Nominal rewriting systems. In *Proc. of PPDP'04*, pages 108–119. ACM Press, 2004.

[4] M. J. Gabbay and A. M. Pitts. A new approach to abstract syntax with variable binding. *Formal Aspects of Computing*, 13:341–363, 2002.

[5] J. W. Klop, V. van Oostrom, and F. van Raamsdonk. Combinatory reduction systems: introduction and survey. *Theoret. Comput. Sci.*, 121:279–308, 1993.

[6] R. Mayr and T. Nipkow. Higher-order rewrite systems and their confluence. *Theoret. Comput. Sci.*, 192:3–29, 1998.

[7] E. Ohlebusch. Church-Rosser theorems for abstract reduction modulo an equivalence relation. In *Proc. of RTA'98*, LNCS 1379, pages 17–31. Springer-Verlag, 1998.

[8] E. Ohlebusch. *Advanced Topics in Term Rewriting*. Springer-Verlag, 2002.

[9] A. M. Pitts. Nominal logic, a first order theory of names and binding. *Inform. and Comput.*, 186:165–193, 2003.

[10] T. Suzuki, K. Kikuchi, T. Aoto, and Y. Toyama. Confluence of orthogonal nominal rewriting systems revisited. In *Proc. of RTA'15*, LIPIcs 36, pages 301–317, 2015.

[11] T. Suzuki, K. Kikuchi, T. Aoto, and Y. Toyama. Basic properties on nominal rewriting. http://www.nue.riec.tohoku.ac.jp/user/kentaro/cr-nominal/proofs-basic.pdf.

[12] T. Suzuki, K. Kikuchi, T. Aoto, and Y. Toyama. Proof of critical pair lemma. http://www.nue.riec.tohoku.ac.jp/user/kentaro/cr-nominal/proof-cplemma.pdf.

[13] C. Urban, A. M. Pitts, and M. J. Gabbay. Nominal unification. *Theoret. Comput. Sci.*, 323:473–497, 2004.