Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 323 (2016) 39–56

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Checking Overlaps of Nominal Rewriting Rules

Mauricio Ayala-Rincón [a,2] Maribel Fernández [b,3]
Murdoch James Gabbay[c,4] Ana Cristina Rocha-Oliveira[a,1]

[a] *Departamentos de Matemática e Ciência da Computação*
Universidade de Brasília, Brasília D.F., Brasil

[b] *Department of Informatics*
King's College London, London, UK

[c] *School of Mathematics and Computer Sciences*
Heriot-Watt University, Einburgh, UK

Abstract

Nominal rewriting generalises first-order rewriting by providing support for the specification of binding operators. In this paper, we give sufficient conditions for (local) confluence of closed nominal rewriting theories, based on the analysis of rule overlaps. More precisely, we show that closed nominal rewriting rules where all proper critical pairs are joinable are locally confluent. We also show how to refine the notion of rule overlap to derive confluence of the closed rewriting relation. The conditions that we define are easy to check using a nominal unification algorithm.

Keywords: nominal syntax, rewriting, confluence, binding

1 Introduction

Two key properties of rewrite theories are *termination* ('the computation is finite') and *confluence* ('it is deterministic'). Termination and confluence are undecidable in general, but decidable criteria do exist that are sufficient, and so can be used to check that a rewrite theory satisfies these properties.

Criteria for guaranteeing confluence of rewriting theories were first investigated in the context of the λ-calculus and abstract rewrite theories in works such as [15], in which the famous Newman's Lemma was stated: confluence and local confluence

[1] Email: anacrismarie@gmail.com. Author supported by a Ph.D. scholarship from CAPES Brazil.
[2] Email: ayala@unb.br. Work partially supported by grant CNPq UNIVERSAL 476952/2013-1.
[3] Email: maribel.fernandez@kcl.ac.uk. Work partially supported by grant CsF PVE CAPES 146/2012.
[4] Web: gabbay.org.uk

http://dx.doi.org/10.1016/j.entcs.2016.06.004

coincide for terminating rewrite theories. Nowadays this is seen as a combinatorial property of abstract rewrite theories that strictly depends on Noetherianity, that is, well-foundedness of the rewrite relation [11].

Without termination, the Critical Pair Lemma, which is the kernel of the famous Knuth-Bendix completion procedure, guarantees local confluence of term rewriting theories [13]. The most famous sufficient condition for confluence without termination, giving also rise to a programming discipline, is *orthogonality*. Orthogonality essentially avoids ambiguity through two easily verifiable syntactic constraints on the rewriting rules: left-linearity, that constrains each variable occurring in the left-hand side of each rule to appear only once, and non-ambiguity, that constrains left-hand sides of rules to have no overlaps (except for trivial ones, at variable positions or between a rule and its copy at the root position). With these syntactic restrictions confluence of orthogonal rewriting theories is guaranteed [17].

Nominal rewriting generalises first-order rewriting by providing support for the specification of languages with binding operators. In nominal syntax, there are two kinds of variables: *atoms*, which are used to represent object-level variables and can be abstracted but not be substituted, and meta-variables, called simply *variables* or *unknowns*, which can be substituted but cannot be abstracted. Substitution of a variable by a term can capture atoms (unlike higher-order theories, where substitution is non-capturing). The nominal rewriting relation is defined using equivariant nominal matching, that is, matching modulo α-equivalence and atom permutations. If rules are closed, then nominal matching is sufficient to generate the rewrite relation. Nominal matching is matching modulo α-equivalence only, and it is efficient (it can be solved in linear time [3]).

For nominal rewriting theories, the Critical Pair Lemma and confluence of orthogonal theories were first investigated in [7], where it was shown that the above-mentioned results extend to the class of uniform nominal rewriting theories, that is, theories where rules do not generate new atoms. More precisely, in [7] it is shown that for the class of uniform theories, if all the non-trivial critical pairs are joinable then the theory is locally confluent, and therefore confluent if it is also terminating (by Newman's Lemma). Another sufficient condition for confluence of uniform theories is orthogonality: if the rules are left-linear and have no non-trivial critical pairs then the theory is confluent [7]. Trivial critical pairs are defined by overlaps at variable positions, or overlaps at the root between a rule and its copy (as for first-order rewrite theories). However, overlaps at the root between a rule and a permuted variant are not trivial. Both of these criteria rely on checking all non-trivial critical pairs. It is important to check also the overlaps at the root between a rule and its permuted variants, because if we miss those overlaps the theory might not be confluent (see Example 4.3).

In [19], the orthogonality condition given in [7] was relaxed, to permit overlaps at the root between a rule and its permuted copies, but only for uniform rules that satisfy an additional condition, called α-stability.

In this paper we give new criteria for (local) confluence of nominal rewriting. We show that also the conditions in the Critical Pair Lemma can be relaxed if rules

are uniform and α-stable: if all the non-trivial critical pairs, except possibly those caused by overlaps at the root between a rule and its permuted variants, are joinable, then the theory is locally confluent. Moreover, we give a new sufficient condition for α-stability, which is easy to check as it relies simply on nominal matching.

In addition, we give new improved criteria for closed nominal rewriting: it is sufficient to check the overlaps generated using just one variant of each rule.

Summarising, the main contributions of this paper are:

(i) We relax the conditions in the Critical Pair Lemma for uniform rules that are α-stable: it is not necessary to consider critical pairs generated by overlaps at the root between a rule and a permuted variant. See Subsection 4.1. [5]

(ii) We show that closedness is a sufficient condition for α-stability. Since closedness is easy to check (by simply solving a nominal matching problem), we get an easy to check condition for α-stability. See Subsection 4.2.

(iii) We show that for closed rewriting, the criteria can be relaxed even more: it is sufficient to check overlaps by using one freshened version of each rule; overlaps between permuted variants of rules (at the root or otherwise) do not need to be considered at all. See Section 5.

1.1 Related work

First-order rewriting systems and the λ-calculus provide two useful notions of terms and reduction. However, both have limitations, which motivated extensions such as higher-order rewriting systems (see, e.g., [12,14]). Nominal rewriting systems are at an intermediate level between higher-order rewriting systems and their explicit substitution versions, which implement in a first-order setting the capture-avoiding substitution operation together with α-conversion. For the latter, indices and rewrite rules are used to deal with the management of bound variables (see, e.g., [18]). Using nominal rewriting, we can specify capture-avoiding substitutions without the need to manage indices, since names and α-equivalence are primitive notions.

Two notions of 'orthogonality' exist in previous work for nominal rewriting: In [7], orthogonality was left-linearity plus no non-trivial critical pairs. This was proved a sufficient condition for confluence of uniform rewrite rules. The notion of orthogonality was relaxed in [19] to allow overlaps at the root between permuted variants of rules. This weaker notion does not ensure confluence of uniform rules. If we also have α-stability then confluence is guaranteed [19].

A sufficient condition for α-stability was given in [19], called "abstract skeleton preserving" (ASP). This is a strong restriction: it only allows identity permutations to be suspended on variables, and it requires the use of different atoms in nested abstractions. Here we show that closedness, which does not impose such restrictions and can be checked simply by solving a nominal matching problem, is a sufficient condition for α-stability. In addition, for closed rewriting the criteria for confluence

[5] This result was independently obtained by T. Suzuki, K. Kikuchi, T. Aoto and Y. Toyama, "On confluence of nominal rewriting systems", 16th JSSST Workshop on Programming and Programming Languages, 2014, in Japanese.

can be simplified, by checking only overlaps of freshened rules. Closedness and the ASP criterion are complementary in the sense that none of them implies the other.

2 Syntax

We fix disjoint countably infinite collections of **atoms**, **unknowns** (or variables), and **term-formers** (or function symbols). We write $\mathbb{A}$ for the set of atoms; $a, b, c, \ldots$ will range over distinct atoms. $X, Y, Z, \ldots$ will range over distinct unknowns. $\mathsf{f}, \mathsf{g}, \ldots$ will range over distinct term-formers. We assume that to each f is associated an **arity** $n \geq 0$. A **signature** Σ is a set of term-formers with their arities.

Definition 2.1 A **permutation** π is a bijection on atoms such that $nontriv(\pi) = \{a \mid \pi(a) \neq a\}$ is finite. We write $(a\ b)$ for the **swapping** permutation that maps a to b, b to a and all other c to themselves, and id for the **identity permutation**, so $id(a) = a$. The notation $\pi \circ \pi'$ is used for **functional composition** of permutations, so $(\pi \circ \pi')(a) = \pi(\pi'(a))$, and π^{-1} for **inverse**, so $\pi(a) = b$ if and only if $a = \pi^{-1}(b)$.

Permutations are represented by lists of swappings; thus, composition is list concatenation, and the inverse is obtained simply by reversing the list.

Definition 2.2 Define **(nominal) terms** inductively by:

$$s, t, l, r, u \quad ::= \quad a \ \mid\ \pi \cdot X \ \mid\ [a]t \ \mid\ \mathsf{f}(t_1, \ldots, t_n)$$

Call $\pi \cdot X$ a **(suspended) variable** and $[a]t$ an **(atom-)abstraction**; it represents '$x.e$' or '$x.\phi$' in expressions like '$\lambda x.e$' or '$\forall x.\phi$'. We write $\equiv$ for syntactic identity.

Definition 2.3 Define $\pi \cdot t$ a **permutation action** by:

$$\begin{array}{rr} \pi \cdot a \equiv \pi(a) & \pi \cdot (\pi' \cdot X) \equiv (\pi \circ \pi') \cdot X \\ \pi \cdot [a]t \equiv [\pi(a)](\pi \cdot t) & \pi \cdot \mathsf{f}(t_1, \ldots, t_n) \equiv \mathsf{f}(\pi \cdot t_1, \ldots, \pi \cdot t_n) \end{array}$$

A **substitution (on unknowns)**, ranged over by $\theta, \sigma, \ldots$, is a partial function from unknowns to terms with finite domain. We write id for the substitution with $dom(id) = \varnothing$ (it will always be clear whether we mean 'id the identity substitution' or 'id the identity permutation'). If $X \notin dom(\sigma)$ then $\sigma(X)$ denotes $id \cdot X$.

Define $t\sigma$ a**(n unknowns) substitution action** by:

$$\begin{array}{rlr} a\sigma \equiv a & (\pi \cdot X)\sigma \equiv \pi \cdot X & (X \notin dom(\sigma)) \\ ([a]t)\sigma \equiv [a](t\sigma) & (\pi \cdot X)\sigma \equiv \pi \cdot \sigma(X) & (X \in dom(\sigma)) \\ \mathsf{f}(t_1, \ldots, t_n)\sigma \equiv \mathsf{f}(t_1\sigma, \ldots, t_n\sigma) & & \end{array}$$

If σ and θ are substitutions, $\sigma \circ \theta$ maps each X to $(X\sigma)\theta$.

Definition 2.4 The set $Pos(t)$ of **positions** of a term t is defined below. Note that ϵ is the only position in atoms and variables.

$$\frac{}{\epsilon \in Pos(t)}\,(\mathbf{p}_\epsilon) \quad \frac{p \in Pos(t)}{1 \cdot p \in Pos([a]t)}\,(\mathbf{p}_{[\mathbf{a}]}) \quad \frac{p \in Pos(t_i) \quad (1 \leq i \leq n)}{i \cdot p \in Pos(\mathsf{f}(t_1, \ldots, t_i, \ldots, t_n))}\,(\mathbf{p}_\mathsf{f})$$

Call $t|_p$ a **subterm** of t **at position** p when

$$t|_\epsilon = t \qquad [a]t|_{1\cdot p} = t|_p \qquad \mathsf{f}(t_1,\ldots,t_i,\ldots,t_n)|_{i\cdot p} = t_i|_p \quad (1 \le i \le n)$$

If $p \in Pos(s)$, then $s[p \leftarrow t]$ denotes the replacement of $s|_p$ by t in s.

Definition 2.5 A **freshness (constraint)** is a pair $a\#t$ of an atom a and a term t. We call a freshness of the form $a\#X$ **primitive**, and a finite set of primitive freshnesses a **freshness context**. Δ, Γ and ∇ will range over freshness contexts.

We denote by $\nabla\sigma$ the set $\{a\#\sigma(X) |\ a\#X \in \nabla\}$ of freshness constraints.

A **freshness judgement** is a tuple $\Delta \vdash a\#t$ of a freshness context and a freshness constraint. An **α-equivalence judgement** is a tuple $\Delta \vdash s \approx_\alpha t$ of a freshness context and two terms. The **derivable** freshness and α-equivalence judgements are defined by the rules in Figure 1, where $ds(\pi,\pi') = \{a \in \mathbb{A} \mid \pi(a) \neq \pi'(a)\}$. We call $ds(\pi,\pi')$ the difference set of permutations π and π'.

$$\frac{}{\Delta \vdash a\#b}\,(\#\mathbf{ab}) \qquad \frac{}{\Delta \vdash a\#[a]t}\,(\#[\mathbf{a}])$$

$$\frac{(\pi^{-1}(a)\#X) \in \Delta}{\Delta \vdash a\#\pi\cdot X}\,(\#\mathbf{X}) \qquad \frac{\Delta \vdash a\#t}{\Delta \vdash a\#[b]t}\,(\#[\mathbf{b}])$$

$$\frac{\Delta \vdash a\#t_1 \ \cdots\ \Delta \vdash a\#t_n}{\Delta \vdash a\#\mathsf{f}(t_1,\ldots,t_n)}\,(\#\mathsf{f}) \qquad \frac{}{\Delta \vdash a \approx_\alpha a}\,(\approx_\alpha\mathbf{a})$$

$$\frac{\Delta \vdash b\#t \quad \Delta \vdash (b\ a)\cdot t \approx_\alpha u}{\Delta \vdash [a]t \approx_\alpha [b]u}\,(\approx_\alpha[\mathbf{b}]) \qquad \frac{a\#X \in \Delta \text{ for all } a \in ds(\pi,\pi')}{\Delta \vdash \pi\cdot X \approx_\alpha \pi'\cdot X}\,(\approx_\alpha\mathbf{X})$$

$$\frac{\Delta \vdash t \approx_\alpha u}{\Delta \vdash [a]t \approx_\alpha [a]u}\,(\approx_\alpha[\mathbf{a}]) \qquad \frac{\Delta \vdash t_i \approx_\alpha u_i \quad (1 \le i \le n)}{\Delta \vdash \mathsf{f}(t_1,\ldots,t_n) \approx_\alpha \mathsf{f}(u_1,\ldots,u_n)}\,(\approx_\alpha\mathsf{f})$$

Figure 1: Freshness and α-equality

Definition 2.6 The functions $atms(t)$ and $unkn(t)$ will be used to compute the set of atoms and unknowns in a term, respectively. They are defined by:

$$atms(a) = \{a\} \qquad atms(\pi\cdot X) = nontriv(\pi)$$
$$atms([a]t) = atms(t) \cup \{a\} \qquad atms(\mathsf{f}(t_1,\ldots,t_n)) = \textstyle\bigcup_i atms(t_i)$$
$$unkn(a) = \varnothing \qquad unkn(\pi\cdot X) = \{X\}$$
$$unkn([a]t) = unkn(t) \qquad unkn(\mathsf{f}(t_1,\ldots,t_n)) = \textstyle\bigcup_i unkn(t_i)$$

3 Nominal Rewriting

This section introduces the main concepts related with nominal rewriting, including the nominal rewriting relation itself, confluence, closedness of terms in context and rules and the closed rewriting relation.

Definition 3.1 A **rewrite judgement** is a tuple $\nabla \vdash l \to r$ of a freshness context and two terms. We may write '$\varnothing \vdash$' as '$\vdash$'.

A **rewrite theory** $\mathsf{R} = (\Sigma, Rw)$ is a pair of a signature Σ and a possibly infinite set of rewrite judgements Rw in that signature; we call these **rewrite rules**.

A rewrite rule $\nabla \vdash l \to r$ is **left-linear** if each unknown occurs at most once in l.

Definition 3.2 Define t^π the **meta-action** of π on t by:

$$a^\pi = \pi(a) \qquad (\rho \cdot X)^\pi = \rho^\pi \cdot X \qquad ([a]t)^\pi = [a^\pi]t^\pi \qquad \mathsf{f}(t_1, \ldots, t_n)^\pi = \mathsf{f}(t_1^\pi, \ldots, t_n^\pi),$$

where $id^\pi = id$ and $((a\ b) \circ \rho)^\pi = (\pi(a)\ \pi(b)) \circ \rho^\pi$.

Extend the meta-action to contexts by $\nabla^\pi = \{\pi(a)\#X \mid a\#X \in \nabla\}$.

The meta-action of permutations affects only atoms in terms (it does not suspend on variables, in contrast with the permutation action of Definition 2.3). We use it to define the *equivariant closure* of a set of rules, needed to generate the rewrite relation (Definition 3.4; see [7,8] for more details).

Definition 3.3 The **equivariant closure** of a set Rw of rewrite rules is the closure of Rw by the meta-action of permutations, that is, it is the set of all the permutative variants of rules in Rw. We write $eq\text{-}closure(Rw)$ for the equivariant closure of Rw.

Below we write $\Delta \vdash (\phi_1, \ldots, \phi_n)$ for the judgements $\Delta \vdash \phi_1, \ldots, \Delta \vdash \phi_n$.

Definition 3.4 Nominal rewriting: Let $\mathsf{R} = (\Sigma, Rw)$ be a rewrite theory. The **one-step rewrite relation** $\Delta \vdash s \xrightarrow{\mathsf{R}} t$ is the least relation such that for every $(\nabla \vdash l \to r) \in Rw$, position p, permutation π, and substitution θ,

$$\frac{\Delta \vdash \left(\nabla^\pi\theta, \quad s|_p \approx_\alpha l^\pi\theta, \quad s[p \leftarrow r^\pi\theta] \approx_\alpha t\right)}{\Delta \vdash s \xrightarrow{\mathsf{R}} t}\ (\mathbf{Rew}_{\nabla \vdash \mathbf{l} \to \mathbf{r}})$$

The notation $\Delta \vdash s \to_{\langle R,p,\pi,\theta\rangle} t$ highlights the fact that the rewrite step from s to t occurs with some specific rule R, position p, permutation π and substitution θ, under the freshness context Δ.

The **rewrite relation** $\Delta \vdash_{\mathsf{R}} s \to t$ is the reflexive transitive closure of the one-step rewrite relation, that is, the least relation that includes the one-step rewrite relation and such that:

- for all Δ and s: $\Delta \vdash s \approx_\alpha s'$ implies $\Delta \vdash_{\mathsf{R}} s \to s'$; and
- for all Δ, s, t, u: $\Delta \vdash_{\mathsf{R}} s \to t$ and $\Delta \vdash_{\mathsf{R}} t \to u$ implies $\Delta \vdash_{\mathsf{R}} s \to u$.

If $\Delta \vdash_{\mathsf{R}} s \to t$ holds, we say that s rewrites to t in the context Δ.

The rewrite relation is defined in a freshness context since it takes into account α-equivalence, which depends on freshness information for the term unknowns.

Example 3.5 The following rewrite theory, using a signature containing term-formers λ of arity 1, and *app* and *subst* of arity 2, defines β-reduction for the λ-calculus. Below, application is denoted by juxtaposition and $subst([a]X, Y)$ is written

$X[a \mapsto Y]$ as usual (syntactic sugar). In this theory, we can derive $\vdash_{\mathsf{R}} (\lambda[a]a)Y \to Y$ and also $a\#Z \vdash_{\mathsf{R}} (\lambda[a]Z)Y \to Z$.

$$\begin{array}{lll}
(\mathsf{Beta}) & \vdash (\lambda[a]X)Y & \to X[a \mapsto Y] \\
(\sigma_{app}) & \vdash (XX')[a \mapsto Y] & \to X[a \mapsto Y]X'[a \mapsto Y] \\
(\sigma_{var}) & \vdash a[a \mapsto X] & \to X \\
(\sigma_{lam}) & b\#Y \vdash (\lambda[b]X)[a \mapsto Y] & \to \lambda[b](X[a \mapsto Y]) \\
(\sigma_{\epsilon}) & a\#X \vdash X[a \mapsto Y] & \to X
\end{array}$$

Definition 3.6 A rewrite theory R is **terminating** if there are no infinite rewriting sequences, .e., there is no term in context from which infinite rewriting steps can be performed. It is **locally confluent** if $\Delta \vdash s \overset{\mathsf{R}}{\to} u$ and $\Delta \vdash s \overset{\mathsf{R}}{\to} v$ implies that there exists w such that $\Delta \vdash_{\mathsf{R}} u \to w$ and $\Delta \vdash_{\mathsf{R}} v \to w$. It is **confluent** when, if $\Delta \vdash_{\mathsf{R}} s \to t$ and $\Delta \vdash_{\mathsf{R}} s \to t'$, then u exists such that $\Delta \vdash_{\mathsf{R}} t \to u$ and $\Delta \vdash_{\mathsf{R}} t' \to u$.

We call the situation $\Delta \vdash s \overset{\mathsf{R}}{\to} u$ and $\Delta \vdash s \overset{\mathsf{R}}{\to} v$ a **peak**.

Remark 3.7 Since the definition of the rewriting relation generated by a rewrite theory $\mathsf{R} = (\Sigma, Rw)$ takes into account permuted variants of rules (via the use of the permutation π in the one-step rewrite relation, see Definition 3.4), it is not necessary to include permuted variants of rules in Rw. For convenience, in the rest of the paper we assume that for any $R \in Rw$, if R and R^{π} are both in Rw then $\pi = id$; in other words, Rw does not contain permuted variants of the same rule.

According to Definition 3.4, to generate a rewrite step we need to solve an equivariant matching problem, that is, we need to find a permutation and a substitution such that $\Delta \vdash s|_p \approx_{\alpha} l^{\pi}\theta$. This problem is decidable, but exponential over the number of different atoms of the terms in context [4]. However, for **closed rules** [7], a simpler matching problem of the form $\Delta \vdash s|_p \approx_{\alpha} l\theta$, called nominal matching [20], suffices to generate the rewrite relation. Nominal matching is decidable and unitary [20] and efficient (it can be solved in linear time [3,2]).

Closed rules roughly correspond to rules without free atoms, where rewriting cannot change the binding status of an atom. They are the counterpart of rules in standard higher-order rewriting formats (see [6]). Below we first recall the definition of nominal matching and then give a structural definition and an operational characterisation of closed terms.

Definition 3.8 A **term-in-context** is a pair $\Delta \vdash s$ of a freshness context and a term. A **nominal matching problem** is a pair of terms-in-context

$$(\nabla \vdash l) \;{}_{?}{\approx}\; (\Delta \vdash s) \quad \text{where} \quad unkn(\nabla \vdash l) \cap unkn(\Delta \vdash s) = \varnothing.$$

A **solution** to this problem is a substitution σ such that $\Delta \vdash \nabla\sigma$, $\Delta \vdash l\sigma \approx_{\alpha} s$, and $dom(\sigma) \subseteq unkn(\nabla \vdash l)$.

The following structural definition of closedness follows [6,5].

Definition 3.9 Call a term-in-context $\Delta \vdash t$ **closed** when

(i) every occurrence of an atom subterm a in t is under an abstraction of a;
(ii) if $\pi \cdot X$ occurs under an abstraction of $\pi \cdot a$ then any occurrence of $\pi' \cdot X$ occurs under an abstraction of $\pi' \cdot a$ or $a\#X \in \Delta$;
(iii) for any pair $\pi_1 \cdot X, \pi_2 \cdot X$ occurring in t, and $a \in ds(\pi_1, \pi_2)$, if neither $\pi_1 \cdot X$ nor $\pi_2 \cdot X$ occurs in the scope of an abstraction of $\pi_1 \cdot a$ or $\pi_2 \cdot a$, respectively, then $a\#X \in \Delta$.

Call $R = (\nabla \vdash l \to r)$ **closed** when $\nabla \vdash (l, r)$ is closed. [6]

It is easy to check whether a term is closed, using nominal matching and a freshened variant of the term [7] (see Proposition 3.11 below).

Definition 3.10 A **freshened variant** t^ν of a nominal term t is a term with the same structure as t, except that the atoms and unknowns are replaced by 'fresh' atoms and unknowns (so they are not in $atms(t)$ and $unkn(t)$, and perhaps are also fresh with respect to some atoms and unknowns from other syntax, which we will always specify). We omit an inductive definition.

Similarly, if ∇ is a freshness context then ∇^ν denotes a freshened variant of ∇ (so if $a\#X \in \nabla$ then $a^\nu\#X^\nu \in \nabla^\nu$, where a^ν and X^ν are chosen fresh for the atoms and unknowns appearing in ∇).

We may extend this to other syntax, like equality and rewrite judgements.

Note that if $\nabla^\nu \vdash l^\nu \to r^\nu$ is a freshened variant of $\nabla \vdash l \to r$ then $unkn(\nabla^\nu \vdash l^\nu \to r^\nu) \cap unkn(\nabla \vdash l \to r) = \varnothing$.

Proposition 3.11 *A term-in-context $\nabla \vdash l$ is closed if and only if there exists a solution for the matching problem*

$$(\nabla^\nu \vdash l^\nu) \quad {}_?{\approx} \quad (\nabla, atms(\nabla^\nu, l^\nu)\#unkn(\nabla, l) \vdash l). \tag{1}$$

Due to the link between closedness of terms-in-context and solvability of a nominal matching problem, made explicit by the proposition above, the definition of closed rewriting (Definition 3.12) is based on nominal matching instead of using equivariant matching as in Definition 3.4.

Definition 3.12 Given a rewrite rule $R = (\nabla \vdash l \to r)$ and a term-in-context $\Delta \vdash s$, write $\Delta \vdash s \xrightarrow{R}_c t$ when there is some R^ν a freshened variant of R (so, fresh for R, Δ, s, and t), position p and substitution θ such that

$$\Delta, atms(R^\nu) \# unkn(\Delta, s, t) \vdash (\nabla^\nu\theta,\ s|_p{\approx_\alpha} l^\nu\theta,\ s[p{\leftarrow}r^\nu\theta]{\approx_\alpha} t). \tag{2}$$

We call this (one-step) **closed rewriting**.

The **closed rewrite relation** $\Delta \vdash_{\mathsf{R}} s \to_c t$ is the reflexive transitive closure of the one-step closed rewrite relation (as in Definition 3.4, but notice the extended freshness context).

[6] Here we use pair as a term former and apply the definition above.

Example 3.13 Any rule with free atoms, such as $\vdash \mathsf{f}(a,a) \to a$, is not closed (it is impossible to match it with a freshened variant). The rule $R = \vdash [a]\mathsf{f}(a,X) \to 0$ is closed, since taking a freshened version $R^{\prime} = \vdash [b]\mathsf{f}(b,Y) \to 0$, it is possible to solve the matching problem $(\vdash ([b]\mathsf{f}(b,Y),0)) \ {}_{?}{\approx} \ (b\#X \vdash ([a]\mathsf{f}(a,X),0))$ with the substitution $\sigma = [Y \mapsto (a\ b)\cdot X]$. Notice that $b\#X \vdash [b]\mathsf{f}(b,(a\ b)\cdot X) \approx_\alpha [a]\mathsf{f}(a,X)$.

We refer to [7,8] for more examples and properties of closed rewriting.

To compute overlaps of rules, we use a nominal unification algorithm [20].

Definition 3.14 A **nominal unification problem** is a set of freshness constraints and pairs of terms, written $\{a_1\#t_1,\ldots,a_k\#t_k, s_1 \ {}_{?}{\approx}_{?} \ u_1,\ldots,s_m \ {}_{?}{\approx}_{?} \ u_m\}$. It is unifiable if there exists a **solution** $\langle \Gamma,\theta\rangle$ (freshness context and substitution) such that $\Gamma \vdash (a_1\#t_1\theta,\ldots,a_k\#t_k\theta, s_1\theta \approx_\alpha u_1\theta,\ldots,s_m\theta \approx_\alpha u_m\theta)$. In this case, $\langle \Gamma,\theta\rangle$ is said to be a **unifier** for the problem.

Nominal unification is decidable and unitary, that is, if there is a solution for a nominal unification problem there exists a most general one.

4 Confluence of Nominal Rewriting

In this section we consider two well-known criteria for confluence of first-order rewriting based on the notion of overlapping rewrite steps [1]. They can be extended to nominal rewrite theories, but it is necessary to add some conditions.

4.1 Critical Pair Criterion and Orthogonality

The notion of overlap has been extended from the first-order setting to the nominal rewriting setting [7]. In the first-order case, overlaps are computed by unification of a left-hand side of a rule R_1 with a non-variable subterm of the left-hand side of a rule R_2 (which could be a copy of R_1 with renamed variables, in which case the subterm must be strict, that is, overlaps at the root between a left-hand side and its copy are not considered). With nominal rules the nominal rewrite relation is generated by the *equivariant closure* of a set of rules (see Definitions 3.3 and 3.4) so we must consider permuted variants of rules, and use nominal unification instead of first-order unification. This is Definition 4.1, which follows [7]:

Definition 4.1 (Overlaps and CPs) Let $R_i = \nabla_i \vdash l_i \to r_i$ $(i=1,2)$ be copies of rewrite rules in *eq-closure*(Rw) (so R_1 and R_2 could be copies of the same rule), where $unkn(R_1) \cap unkn(R_2) = \varnothing$, as usual. If the nominal unification problem $\nabla_1 \cup \nabla_2 \cup \{l_2 \ {}_{?}{\approx}_{?} \ l_1|_p\}$ has a most general solution $\langle \Gamma,\theta\rangle$ for some position p, then we say that R_1 **overlaps** with R_2, and we call the pair of terms-in-context $\Gamma \vdash (r_1\theta, l_1\theta[p{\leftarrow}r_2\theta])$ a **critical pair**. If p is a variable position, or if R_1 and R_2 are identical modulo renaming of variables and $p=\epsilon$, then we call the overlap and critical pair **trivial**, otherwise we call it **non-trivial**.

The critical pair $\Gamma \vdash (r_1\theta, l_1\theta[p{\leftarrow}r_2\theta])$ is **joinable** if there is a term u such that $\Gamma \vdash_{\mathsf{R}} r_1\theta \to u$ and $\Gamma \vdash_{\mathsf{R}} (l_1\theta[p{\leftarrow}r_2\theta]) \to u$.

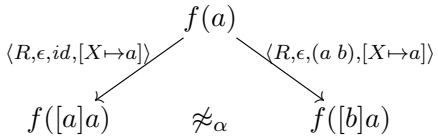
We distinguish between different kinds of overlaps and critical pairs:

Definition 4.2 (Permutative Overlaps and CPs) Let $R_i = \nabla_i \vdash l_i \to r_i$ $(i = 1, 2)$ be copies of rewrite rules in *eq-closure*(Rw), such that there is an overlap. If R_2 is a copy of R_1^π, we say that the overlap is **permutative**. We call a permutative overlap at the root position **root-permutative**. We call an overlap that is not trivial and not root-permutative **proper**. We use the same terminology to classify critical pairs; e.g. we call a critical pair generated by a permutative overlap **permutative**.

A permutative overlap indicates that there is a critical pair generated by a rule and one of its permuted variants.

Note that only the root-permutative overlaps where π is id are trivial. While overlaps at the root between variable-renamed versions of first-order rules can be discarded (they generate equal terms), in nominal rewriting we must also consider overlaps at the root between permuted variants of rules. Indeed, they do not necessarily produce the same result, as the following example shows (see also [19]).

Example 4.3 Consider $R = (\ \vdash f(X) \to f([a]X))$. There is an overlap at the root between this rule and its variant $R^{(a\ b)} = (\ \vdash f(X) \to f([b]X))$, i.e., a root-permutative overlap, which is not trivial. It generates the critical pair $\vdash (f([a]X), f([b]X))$. Note that the terms $f([a]X)$ and $f([b]X)$ are not α-equivalent. This theory is not confluent; we have for instance:



Definition 4.4 introduces *uniformity*. In [7] a Critical Pair Lemma was proved for uniform nominal rewrite theories, that joinability of non-trivial critical pairs implies local confluence; confluence follows by Newman's Lemma if the theory is terminating. Uniformity features in this paper in Theorem 4.6. Intuitively, uniformity means that if a is not free in s and s rewrites to t then a is not free in t.

Definition 4.4 (Uniformity) We call a nominal rewrite theory $\mathsf{R} = (\Sigma, Rw)$ **uniform** [7] when if $\Delta \vdash_{\mathsf{R}} s \to t$ and $\Delta, \Delta' \vdash a\#s$ for some Δ', then $\Delta, \Delta' \vdash a\#t$.

Note that in the Critical Pair Lemma of [7], joinability is assumed for all non-trivial critical pairs. Joinability of proper critical pairs is insufficient for local confluence, even for a uniform theory: the rule in Example 4.3 is uniform. However, an additional condition allows us to prove that uniform rewrite theories with joinable proper critical pairs are locally confluent. Recall the notion of α-stability from [19]:

Definition 4.5 (α-stability) Call a rewrite rule $R = \nabla \vdash l \to r$ **α-stable** when, for all $\Delta, \pi, \sigma, \sigma'$, $\Delta \vdash \nabla\sigma, \nabla^\pi\sigma', l\sigma \approx_\alpha l^\pi\sigma'$ implies $\Delta \vdash r\sigma \approx_\alpha r^\pi\sigma'$.

A rewrite theory $\mathsf{R} = (\Sigma, Rw)$ is **α-stable** if every rule in Rw is α-stable.

α-stability is hard to check in general because of the quantification over all σ and σ'. α-stability is related to *closedness* (Definition 3.9): we show in Section 4.2

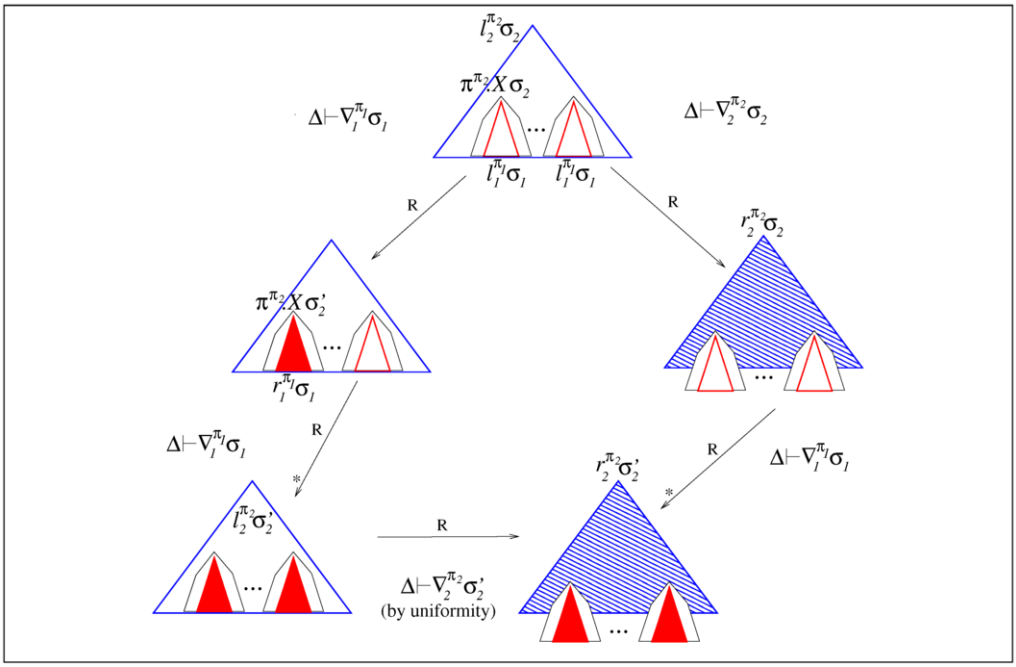


Figure 2: Critical Pair Lemma - case of overlap at a variable position

that closed rules are α-stable. The reverse implication does not hold: for example $\vdash f(a) \to a$ is α-stable but not closed.

Theorem 4.6 (Critical Pair Lemma for uniform α-stable theories) *Let $\mathsf{R} = (\Sigma, Rw)$ be a uniform rewrite theory where all the rewrite rules in Rw are α-stable. If every proper critical pair is joinable, then R is locally confluent.*

Proof We consider cases. There are four kinds of peaks:

- *If the rewrite steps occur at disjoint positions,* then the peak is trivially joinable by applying the same rules, permutations and substitutions.
- *If the peak is an instance of a proper critical pair* (joinable by assumption) then it is joinable since rewriting is compatible with instantiation [7, Theorem 49].
- *If the peak is generated by an overlap at a variable position,* without loss of generality assume $\nabla \vdash s \approx_\alpha l_1^{\pi_1}\sigma_1$ and s occurs inside $l_2^{\pi_2}\sigma_2$ under an instance of an unknown $(\pi^{\pi_2} \cdot X)\sigma_2$ (see Figure 2). Then we can change the action of σ_2 over X, replacing s by t, such that $\nabla \vdash t \approx_\alpha r_1^{\pi_1}\sigma_1$, as it is done in the first-order case. Here we rely on uniformity to ensure that no free atoms are introduced by the rewrite step, so freshness constraints are preserved when replacing s by t.
- *If there is a root-permutative overlap* then joinability follows by α-stability.

□

Definition 4.7 Call a rewrite theory $\mathsf{R} = (\Sigma, Rw)$ **orthogonal** when all the rules in Rw are left-linear and there are no non-trivial critical pairs.

Call $\mathsf{R} = (\Sigma, Rw)$ **quasi-orthogonal** when all rules are left-linear and there are

no proper overlaps.

So orthogonal theories are left-linear and can have trivial overlaps only, whereas quasi-orthogonal theories are left-linear and can have trivial overlaps and root-permutative overlaps (Definition 4.2).

Orthogonal theories were defined in [7]. Quasi-orthogonal theories were defined in [19] and called orthogonal (we changed the name here to avoid confusion).

For uniform nominal rewrite theories, orthogonality implies confluence [7].

Quasi-orthogonality is insufficient for confluence of uniform theories; see Example 4.3. If a theory is uniform, quasi-orthogonal, and α-stable, then it is confluent [19].

4.2 Criterion for α-stability

This section presents closedness as a sufficient condition for α-stability. Closedness is easy to check using a nominal matching algorithm (see Proposition 3.11).

An easy technical lemma will be useful, that substitutions that coincide modulo α on the unknowns in a term yield α-equivalent instances, and vice-versa (i.e., if the instances are α-equivalent, the substitutions must coincide modulo α on the unknowns of the term):

Lemma 4.8 $\Delta \vdash t\sigma \approx_\alpha t\theta \Leftrightarrow \forall X \in unkn(t).\Delta \vdash X\sigma \approx_\alpha X\theta$.

Lemma 4.9 *If R is a closed rule, then R is α-stable.*

Proof It is sufficient to prove the following property: $R = \nabla \vdash l \to r$ closed and $\Delta \vdash s \approx_\alpha l\sigma \to r\sigma$ and $\Delta \vdash s \approx_\alpha l^\pi\sigma' \to r^\pi\sigma'$ implies $\Delta \vdash r\sigma \approx_\alpha r^\pi\sigma'$.

The matching problems $(\nabla^\nu \vdash (l^\nu, r^\nu)) \stackrel{?}{\approx} (\nabla, atms(R^\nu)\#unkn(R) \vdash (l, r))$ and $(\nabla^\nu \vdash (l^\nu, r^\nu)) \stackrel{?}{\approx} (\nabla^\pi, atms(R^\nu)\#unkn(R) \vdash (l^\pi, r^\pi))$ are solvable with solutions θ and θ^π, respectively, insofar as R is closed. Hence, we can infer:

- $\nabla, atms(R^\nu)\#unkn(R) \vdash \nabla^\nu\theta,\ (l^\nu\theta, r^\nu\theta) \approx_\alpha (l, r)$
- $\nabla^\pi, atms(R^\nu)\#unkn(R) \vdash \nabla^\nu\theta^\pi,\ (l^\nu\theta^\pi, r^\nu\theta^\pi) \approx_\alpha (l^\pi, r^\pi)$
- $\Delta \vdash \nabla\sigma, \nabla^\pi\sigma', l\sigma \approx_\alpha l^\pi\sigma' \implies \Delta, atms(R^\nu)\#unkn(R\sigma) \vdash l^\nu\theta\sigma \approx_\alpha l^\nu\theta^\pi\sigma'$

From Lemma 4.8 ($\Rightarrow$), it follows that $\forall X \in unkn(l^\nu) : \Delta, atms(R^\nu)\#unkn(R\sigma) \vdash X\theta\sigma \approx_\alpha X\theta^\pi\sigma'$.

Since $unkn(r^\nu) \subseteq unkn(l^\nu)$, Lemma 4.8 ($\Leftarrow$) can be used to demonstrate the equivalences

$$\Delta, atms(R^\nu)\#unkn(R\sigma) \vdash r^\nu\theta\sigma \approx_\alpha r\sigma, r^\nu\theta^\pi\sigma' \approx_\alpha r^\pi\sigma', r^\nu\theta\sigma \approx_\alpha r^\nu\theta^\pi\sigma'$$

and, finally, $\Delta, atms(R^\nu)\#unkn(R\sigma) \vdash r\sigma \approx_\alpha r^\pi\sigma'$ is obtained by transitivity. Notice that atoms in $atms(R^\nu)$ do not appear in $r\sigma, r^\pi\sigma'$, so that the previous judgement can be strengthened taking only Δ as context. □

5 Better Criteria for Confluence of Closed Rewriting

In this section we study confluence of closed rewriting (Definition 3.12). Closed rewriting uses freshened versions of rules and nominal matching, instead of the computationally more expensive equivariant matching used in Definition 3.4. Closed rewriting is complete for equational reasoning if the axioms are closed [8].

The following three lemmas state properties of closed rules and closed rewriting, and will be useful for Theorems 5.6 and 5.8. The first two state that if a rule has no free atoms then its freshness context can be extended to obtain a closed rule, and closed rewriting with either rule is equivalent. The third lemma states that a rule with free atoms generates an empty closed rewriting relation.

Lemma 5.1 *Let $R = \nabla \vdash l \to r$ be a rule such that every occurrence of an atom subterm a in l or r is under the scope of an abstraction of a (i.e., no atom occurs free as a subterm in R). Then there exists a minimal context $\Delta \subseteq atms(R)\#unkn(R)$ such that $\Delta, \nabla \vdash l \to r$ is closed.*

Proof By definition of closed term (see Definition 3.9), we must check:

(i) Every occurrence of an atom subterm a is under an abstraction of a.
(ii) If $\pi \cdot X$ occurs under an abstraction of $\pi \cdot a$, then any occurrence of $\pi' \cdot X$ is in the scope of an abstraction of $\pi' \cdot a$ or $a\#X \in \nabla \cup \Delta$.
(iii) For any pair $\pi_1 \cdot X, \pi_2 \cdot X$ occurring in R and $a \in ds(\pi_1, \pi_2)$, if $\pi_1 \cdot a$ and $\pi_1 \cdot a$ are not abstracted over the respective occurrences of X, then $a\#X \in \nabla \cup \Delta$.

The first point holds by assumption. For the second and third points, if $a\#X \notin \nabla$ it is sufficient to include $a\#X$ in Δ. □

Lemma 5.2 *Suppose $R = \nabla \vdash l \to r$ and $R' = \Delta, \nabla \vdash l \to r$ are rules such that R has no free atom-subterms and $\Delta \subseteq atms(R)\#unkn(R)$ is the minimal set of freshness constraints that makes R' closed. Then, $\Gamma \vdash s \xrightarrow{R}_c t \Leftrightarrow \Gamma \vdash s \xrightarrow{R'}_c t$.*

Proof *The left-to-right direction.* If $\Gamma \vdash s \xrightarrow{R}_c t$, then $\Gamma, atms(R^\nu)\#unkn(\Gamma, s, t) \vdash s \xrightarrow{R^\nu} t$, i.e., there is θ such that

$$\Gamma, atms(R^\nu)\#unkn(\Gamma, s, t) \vdash s|_p \approx_\alpha l^\nu\theta, t \approx_\alpha s[p \leftarrow r^\nu\theta], \nabla^\nu\theta.$$

Since $atms(R) = atms(R')$, it suffices to show that $\Gamma, atms(R^\nu)\#unkn(\Gamma, s, t) \vdash \Delta^\nu\theta$ to obtain $\Gamma, atms(R^\nu)\#unkn(\Gamma, s, t) \vdash s \xrightarrow{R'^\nu} t$ as required.

To prove $\Gamma, atms(R^\nu)\#unkn(\Gamma, s, t) \vdash \Delta^\nu\theta$, observe that $a^\nu\#X^\nu$ is in Δ^ν if $\pi_1^\nu \cdot X^\nu$ and $\pi_2^\nu \cdot X^\nu$ occur in (l^ν, r^ν) and at least one of the following holds:

- $\pi_1^\nu \cdot a^\nu$ is abstracted over $\pi_1^\nu \cdot X^\nu$ and $\pi_2^\nu \cdot a^\nu$ is not abstracted over $\pi_2^\nu \cdot X^\nu$. We know

$$\Gamma, atms(R^\nu)\#unkn(\Gamma, s, t) \vdash \pi_2^\nu \cdot a^\nu\#(s|_p, t|_p),\ (s|_p, t|_p) \approx_\alpha (l^\nu\theta, r^\nu\theta).$$

  Then, since $\pi_2^\nu \cdot a^\nu$ is not abstracted over $\pi_2^\nu \cdot X^\nu$, the same freshness context allows us to derive $\pi_2^\nu \cdot a^\nu\#\pi_2^\nu \cdot X^\nu\theta$ and, consequently, $a^\nu\#X^\nu\theta$.

- a^{ν} is in $ds(\pi_1^{\nu}, \pi_2^{\nu})$ and neither $\pi_1^{\nu} \cdot a^{\nu}$ nor $\pi_2^{\nu} \cdot a^{\nu}$ are abstracted over the respective occurrences of X^{ν}. The same argument is valid in this case.

The right-to-left direction. If $\Gamma \vdash s \xrightarrow{R'}_c t$, then $\Gamma, atms(R^{\nu})\#unkn(\Gamma, s, t) \vdash s \xrightarrow{R'^{\nu}} t$, i.e., there is θ such that

$$\Gamma, atms(R'^{\nu})\#unkn(\Gamma, s, t) \vdash s|_p \approx_\alpha l^{\nu}\theta, t \approx_\alpha s[p \leftarrow r^{\nu}\theta], \nabla^{\nu}\theta, \Delta^{\nu}\theta.$$

So $atms(R) = atms(R')$. It follows that $\Gamma, atms(R^{\nu})\#unkn(\Gamma, s, t) \vdash s \xrightarrow{R^{\nu}} t$. □

Lemma 5.3 *Suppose $R = \nabla \vdash l \to r$ is a nominal rule and there exist Δ, s, t and a closed-rewriting step $\Delta \vdash s \xrightarrow{R}_c t$. Then every occurrence of an atom subterm a in l or r is under an abstraction of a (i.e., no atom occurs free as a subterm in R).*

Proof By contradiction. Assume R has a free atom subterm a; without loss of generality, we assume $l|_q = a$ (if it occurs in r we reason in the same way). By definition of closed-rewriting, there exists R^{ν}, a fresh variant of R, such that $\Delta, atms(R^{\nu})\#unkn(\Delta, s, t) \vdash s|_p \approx_\alpha l^{\nu}\theta, t \approx_\alpha s[p \leftarrow r^{\nu}\theta], \nabla^{\nu}\theta$. But $l^{\nu}|_q = a^{\nu}$ is free, and a^{ν} does not occur in s, contradicting $\Delta, atms(R^{\nu})\#unkn(\Delta, s, t) \vdash s|_p \approx_\alpha l^{\nu}\theta$.□

The following definitions of *fresh overlap* and *fresh critical pair* will be used to derive sufficient conditions for confluence of closed rewriting.

Definition 5.4 (Fresh Overlaps and CPs) Let $R_i = \nabla_i \vdash l_i \to r_i$ $(i = 1, 2)$ be freshened versions of rewrite rules in Rw (R_1 and R_2 could be two freshened versions of the same rule), where $unkn(R_1) \cap unkn(R_2) = \varnothing$, as usual. If the nominal unification problem $\nabla_1 \cup \nabla_2 \cup \{l_2 \stackrel{?}{\approx}_? l_1|_p\}$ has a most general solution $\langle \Gamma, \theta \rangle$ for some position p, then we say that R_1 **fresh overlaps** with R_2, and we call the pair of terms-in-context $\Gamma \vdash (r_1\theta, l_1\theta[p \leftarrow r_2\theta])$ a **fresh critical pair**.

If p is a variable position, or if R_1 and R_2 are equal modulo renaming of variables and $p = \epsilon$, then we call the overlap and critical pair **trivial**.

If R_1 and R_2 are freshened versions of the same rule and $p = \epsilon$, then we call the overlap and critical pair **fresh root-permutative**.

A fresh overlap (resp. fresh critical pair) that is not trivial and not root-permutative is **proper**.

The fresh critical pair $\Gamma \vdash (r_1\theta, l_1\theta[p \leftarrow r_2\theta])$ is **joinable** if there is a term u such that $\Gamma \vdash_{\mathsf{R}} r_1\theta \to_c u$ and $\Gamma \vdash_{\mathsf{R}} (l_1\theta[p \leftarrow r_2\theta]) \to_c u$.

Definition 5.5 Call a rewrite theory $\mathsf{R} = (\Sigma, Rw)$ **fresh quasi-orthogonal** when all rules are left-linear and there are no proper fresh critical pairs.

Theorem 5.6 (Critical Pair Lemma for Closed Rewriting) *Let $\mathsf{R} = (\Sigma, Rw)$ be a rewrite theory where every proper fresh critical pair is joinable. Then the closed rewriting relation generated by R is locally confluent.*

Proof Since rules with free atom-subterms do not generate closed rewriting steps (Lemma 5.3), without loss of generality we can assume that the rules in Rw do not have free atom-subterms. Consider $\mathsf{R}' = (\Sigma, Rw')$ the closed rewrite theory obtained

by extending the freshness contexts of rules in Rw, as described in Lemma 5.1. Then, by Lemma 5.2, the closed rewriting relation generated by R is equivalent to the one generated by R'. Thus, joinability of proper fresh critical pairs in R implies joinability of proper fresh critical pairs in R' and it suffices to prove local confluence for the closed rewriting relation generated by R'. Also note that since all rules in Rw' are closed, they are uniform and α-stable (Lemma 4.9).

We consider the kinds of peaks that may arise:

- *If the rewrite steps defining the peak occur at disjoint positions* then the peak is trivially joinable by applying the same rules and substitutions.
- *If the peak is generated by an overlap at a variable position* then consider $R_1 = \nabla_1^{\nu} \vdash l_1^{\nu} \to r_1^{\nu}$ and $R_2 = \nabla_2^{\nu} \vdash l_2^{\nu} \to r_2^{\nu}$ freshened versions of two rules (see Figure 2, but here we do not need permuted versions for the rules are already freshened). Let Δ be the context used to rewrite $l_2^{\nu}\sigma_2$ with R_1 and R_2. Without loss of generality, we assume $\Delta, atms(R_1)\#unkn(\Delta, s) \vdash \nabla_1^{\nu}\sigma_1, s \approx_\alpha l_1^{\nu}\sigma_1, t \approx_\alpha r_1^{\nu}\sigma_1$ and s occurs inside $l_2^{\nu}\sigma_2$ under an instance of an unknown $(\pi^{\nu} \cdot X^{\nu})\sigma_2$. Then we can change the action of σ_2 over X^{ν}, replacing s by t, such that $\nabla_1 \vdash t \approx_\alpha r_1^{\pi_1}\sigma_1$, as it is done in the first-order case. Here we rely on the assumption of uniformity, which ensures that no free atoms are introduced by the rewrite step, therefore no freshness constraint will be violated when replacing s by t.
- *If there are freshened rules* $R_1 = \nabla_1^{\nu} \vdash l_1^{\nu} \to r_1^{\nu}$ *and* $R_2 = \nabla_2^{\nu} \vdash l_2^{\nu} \to r_2^{\nu}$ *and a term-in-context* $\Delta \vdash s$*, such that there is a rewrite step at position* p_1 *in* s *using* R_1 *and at position* p_2 *using* R_2 then $\Delta, \Gamma_1 \vdash \nabla_1^{\nu}\sigma, l_1^{\nu}\sigma \approx_\alpha s|_{p_1}$ and $\Delta, \Gamma_2 \vdash \nabla_2^{\nu}\sigma', l_2^{\nu}\sigma' \approx_\alpha s|_{p_2}$. Without loss of generality we assume that $p_2 = p_1 q$. Since the sets of variables in the freshened rules are disjoint, without loss of generality we can assume $dom(\sigma) \cap dom(\sigma') = \emptyset$, and define the substitution $\mu = \sigma \circ \sigma'$ such that $dom(\mu) = dom(\sigma) \cup dom(\sigma')$. Then, $\Delta, \Gamma_1, \Gamma_2 \vdash \nabla_1^{\nu}\mu, \nabla_2^{\nu}\mu, l_1^{\nu}|_q\mu \approx_\alpha l_2^{\nu}\mu$. Therefore the unification problem $\nabla_1^{\nu}, \nabla_2^{\nu}, l_1^{\nu}|_q \stackrel{?}{\approx_?} l_2^{\nu}$ has a solution. Hence, by Definition 5.4, there is a fresh critical pair between R_1 and R_2. Observe that, if $q = \epsilon$ and R_1 is a permuted copy of R_2 (equal or not), then the terms of divergence t_1 and t_2 are α-equivalent by triviality or α-stability. If the critical pair is proper it is joinable by assumption. Therefore the peak is joinable since the rewriting relation is compatible with instantiation [7, Theorem 49].

□

Since it is sufficient to consider just one freshened version of each rule when computing overlaps of closed rules, the number of fresh critical pairs for a rewrite theory with a finite number of rules is finite. Thus, Theorem 5.6 provides an effective criterion for local confluence, similar to the criterion for first-order systems.

We can deduce from Theorem 5.6 that the closed rewriting relation for the closed theory defining explicit substitution in Example 3.5 (i.e., all the rules except Beta) is locally confluent: every proper fresh critical pair is joinable. If we consider also the rule (Beta) then the system is not locally confluent. This does not contradict the previous theorem, because there is a proper fresh critical pair between (Beta)

and (σ_{app}), obtained from $\varnothing \vdash ((\lambda[a]X)Y)[b \mapsto Z]$, which is not joinable:

$$\varnothing \vdash (((\lambda[a]X)[b \mapsto Z])(Y[b \mapsto Z]), (X[a \mapsto Y])[b \mapsto Z]).$$

Next we consider criteria for confluence based on (quasi-) orthogonality. The following lemma is used in the proof of confluence.

Lemma 5.7 *Let $\mathsf{R} = (\Sigma, Rw)$ be a closed rewrite theory.*
$\Delta \vdash_{\mathsf{R}} s \to_c t$ if, and only if, there exist $R_1, \ldots, R_n \in Rw$ such that $\Delta, atms(R_1^{\nu}, \ldots, R_n^{\nu}) \# unkn(\Delta, s) \vdash_{\mathsf{R}} s \to t$.

Proof In both directions, the proof is by induction on the number of steps in $\Delta \vdash_{\mathsf{R}} s \to_c t$ and $\Delta, atms(R_1^{\nu}, \ldots, R_n^{\nu}) \# unkn(\Delta, s) \vdash_{\mathsf{R}} s \to t$, respectively. From left to right, the result follows by definition of closed rewriting. In the other direction, it is necessary to consider closedness of rules. Any version of $R \in Rw$ can be used in one step $\Delta, atms(R_1^{\nu}, \ldots, R_n^{\nu}) \# unkn(\Delta, s) \vdash s \overset{\mathsf{R}}{\to} v$. So, the version R^{ν} freshened with respect to $\Delta, atms(R_1^{\nu}, \ldots, R_n^{\nu})$ and all the terms in the rewrite sequence could be taken in this step. Weakening the freshness context, $\Delta, atms(R_1^{\nu}, \ldots, R_n^{\nu}, R^{\nu}) \# unkn(\Delta, s) \vdash s \overset{\mathsf{R}}{\to} v$ is obtained. Since the atoms of $R_1^{\nu}, \ldots, R_n^{\nu}$ do not occur in Δ, R^{ν} and in the terms of the rewrite sequence, the freshness context can be strengthened into $\Delta, atms(R^{\nu}) \# unkn(\Delta, s) \vdash s \overset{\mathsf{R}}{\to} v$. Thus, $\Delta \vdash s \overset{\mathsf{R}}{\to}_c v$ is reached. □

Theorem 5.8 *If R is a fresh-quasi-orthogonal rewrite theory, then the closed rewriting relation generated by R is confluent.*

Proof As in the previous theorem, we prove confluence for the closed rewriting relation generated by $\mathsf{R}' = (\Sigma, Rw')$, where Rw' is obtained by extending the freshness contexts to close the rules of Rw which do not have free atom-subterms (see Lemmas 5.3, 5.1 and 5.2). Since all rules in Rw' are closed, they are also uniform and α-stable.

Now we can proceed in the usual way (see, e.g., [1,16,7]), by proving the diamond property for a parallel closed-rewriting relation (simultaneous closed rewriting steps at disjoint positions). The proof proceeds by analysis of peaks: When overlaps occur under instances of variables, we use uniformity to ensure that when we change the substitution, the rewrite step is still possible. Joinability of root-permutative overlaps is a consequence of α-stability for the rules are closed.

Alternatively, we can prove confluence by reducing to a previous result for standard nominal rewriting, using the previous lemma: Consider a peak $\Delta \vdash_{\mathsf{R}'} s \to_c t$ and $\Delta \vdash_{\mathsf{R}'} s \to_c v$. By Lemma 5.7 ($\Rightarrow$), there exist $R_1, \ldots, R_n \in Rw'$ such that $\Delta, atms(R_1^{\nu}, \ldots, R_n^{\nu}) \# unkn(\Delta, s) \vdash_{\mathsf{R}'} s \to t$ and $\Delta, atms(R_1^{\nu}, \ldots, R_n^{\nu}) \# unkn(\Delta, s) \vdash_{\mathsf{R}'} s \to v$. Theorem 28 of [19] guarantees confluence with the context $\Delta, atms(R_1^{\nu}, \ldots, R_n^{\nu}) \# unkn(\Delta, s)$, since for closed theories our notion of fresh-quasi-orthogonality coincides with the notion of orthogonality defined in [19] (in this case, it does not matter which permuted version is used

to obtain a proper critical pair). Using Lemma 5.7(⇐), we obtain confluence of $\Delta \vdash \_ \xrightarrow{\mathsf{R}'}_c \_$. □

Example 5.9 Consider a signature for first-order logic, with term-formers $\neg$, $\forall$ and $\exists$ of arity 1, and $\wedge, \vee$ of arity 2 (as usual we write them infix). The following closed rules can be used to simplify formulas:

$$\vdash \neg(X \wedge Y) \to \neg(X) \vee \neg(Y) \quad \text{and} \quad b\#X \vdash \neg(\forall[a]X) \to \exists[b]\neg((b\ a) \cdot X).$$

Why write $\exists[b]\neg((b\ a)\cdot X)$ on the right-hand side above, instead of the α-equivalent $\exists[a]\neg(X)$? We could: these are equivalent—in a nominal context. The version above directly translates the corresponding CRS rule (see [6]) which, following Barendregt's convention, must use *different* names for bound variables in a rule. Theorem 5.8 tells us that the closed rewriting relation generated by the theory in Example 5.9 is confluent. This theory is closed, but forbidden by ASP restrictions because of the permutation $(b\ a)$ on the right-hand side.

The criteria for local confluence given in Theorem 5.6 and for confluence given in Theorem 5.8 for closed rewriting are easy to check using a nominal unification algorithm: just compute overlaps for the set of rules obtained by taking one freshened copy of each given rule. For comparison, the criteria given in [7] and [19] require the computation of critical pairs for permutative variants of rules, which needs equivariant unification (exponential). Theorems 5.6 and 5.8 apply even if the rules are not closed, as long as we use closed rewriting. Consider the uniform rules $\vdash f(a) \to 0$ and $\vdash g(f(b)) \to 0$. These rules have no non-trivial fresh overlap, and closed rewriting is confluent, but the standard rewriting relation is not confluent, since the term $g(f(a))$ rewrites to both $g(0)$ and 0. Using closed rewriting, the term $g(f(a))$ is a normal form.

6 Conclusion

We have presented easy-to-check criteria for confluence of nominal rewriting theories (Theorem 4.6 and Lemma 4.9, and Theorems 5.6 and 5.8), improving previous criteria [7,19]. The Critical Pair Lemma for closed rewriting yields a completion algorithm for closed rewrite rules [9]. We intend to enlarge the PVS library on term rewriting systems [10] to formalise the results of this paper.

References

[1] F. Baader and T. Nipkow. *Term rewriting and all that.* Cambridge UP, 1998.

[2] C. Calvès. Complexity and implementation of nominal algorithms, 2010. PhD thesis, King's College London.

[3] C. Calvès and M. Fernández. Matching and alpha-equivalence check for nominal terms. *J. Comput. Syst. Sci.*, 76(5):283–301, 2010.

[4] J. Cheney. The complexity of equivariant unification. In *Automata, Languages and Programming: 31st Int. Colloquium, ICALP 2004*, volume 3142 of *LNCS*, pages 332–344. Springer, 2004.

[5] R. A. Clouston. Closed terms (unpublished notes). Available from http://cs.au.dk/~ranald/closedterms.pdf, 2007.

[6] J. Domínguez and M. Fernández. Relating nominal and higher-order rewriting. In *Mathematical Foundations of Computer Science 2014 - 39th Int. Symposium, MFCS 2014. Proc., Part I*, volume 8634 of *LNCS*, pages 244–255. Springer, 2014.

[7] M. Fernández and M. J. Gabbay. Nominal rewriting. *Information and Computation*, 205(6):917–965, June 2007.

[8] M. Fernández and M.J. Gabbay. Closed nominal rewriting and efficiently computable nominal algebra equality. In *Proc. 5th Int. Workshop on Logical Frameworks and Meta-languages: Theory and Practice, LFMTP 2010*, pages 37–51, 2010.

[9] M. Fernández and A. Rubio. Nominal Completion for Rewrite Systems with Binders. In *Automata, Languages, and Programming*, volume 7392 of *LNCS*, pages 201–213. Springer, 2012.

[10] A.L. Galdino and M. Ayala-Rincón. A Formalization of the Knuth-Bendix(-Huet) Critical Pair Theorem. *J. Autom. Reasoning*, 45(3):301–325, 2010.

[11] G. P. Huet. Confluent reductions: Abstract properties and applications to term rewriting systems: Abstract properties and applications to term rewriting systems. *J. of the ACM*, 27(4):797–821, October 1980.

[12] J.-W. Klop, V. van Oostrom, and F. van Raamsdonk. Combinatory reduction systems, introduction and survey. *Theoretical Computer Science*, 121:279–308, 1993.

[13] D. Knuth and P. Bendix. Simple word problems in universal algebras. In *Computational Problems in Abstract Algebra*. Pergamon Press, Oxford, 1970.

[14] R. Mayr and T. Nipkow. Higher-order rewrite systems and their confluence. *Theoretical Computer Science*, 192:3–29, 1998.

[15] M. H. A. Newman. On Theories with a Combinatorial Definition of "Equivalence". *The Annals of Mathematics*, 43(2):pp. 223–243, 1942.

[16] A. C. Rocha-Oliveira and M. Ayala-Rincón. Formalizing the confluence of orthogonal rewriting systems. In *Proc. 7th Workshop on Logical and Semantic Frameworks, with Applications, LSFA 2012*, pages 145–152, 2012.

[17] B. K. Rosen. Tree-manipulating systems and Church-Rosser theorems. *J. of the ACM*, 20(1):160–187, January 1973.

[18] M.-O. Stehr. `CINNI` - A Generic Calculus of Explicit Substitutions and its Application to λ- ς- and π-Calculi. *Electronic Notes in Theoretical Computer Science*, 36:70–92, 2000. The 3rd Int. Workshop on Rewriting Logic and its Applications.

[19] T. Suzuki, K. Kikuchi, T. Aoto, and Y. Toyama. Confluence of Orthogonal Nominal Rewriting Systems Revisited. In *26th Int. Conf. on Rewriting Techniques and Applications (RTA 2015)*, volume 36 of *LIPIcs*, pages 301–317, 2015.

[20] C. Urban, A. M. Pitts, and M. Gabbay. Nominal Unification. *Theoretical Computer Science*, 323(1-3):473–497, 2004.