A Confluence Criterion for Non Left-Linearity in a $\beta\eta$-Free Reformulation of HRSs

Thiago Felicissimo

Université Paris-Saclay, INRIA project Deducteam, LMF, ENS Paris-Saclay
`thiago.felicissimo@inria.fr`

Abstract

We give a criterion for showing confluence of non-left linear (Pattern) Higher-order Rewrite Systems (HRSs). More precisely, our criterion concerns 2nd order signatures and allows one to show object confluence, that is, confluence when restricting to terms only with variables of order 0. As a second contribution, we give a reformulation of HRSs in which one never needs to speak of meta-level $\beta\eta$-equality.

1 Introduction

When proving confluence in Nipkow's (Pattern) Higher-order Rewrite Systems (HRSs) [8], one generally has to be in one of the two following cases. If the rewrite system being considered is (1) strongly normalizing (s.n.), then by the critical pair lemma it suffices to check that critical pairs are joinable [8]. In most situations this is too strong of a requirement, but fortunately if the system is (2) left-linear then many criteria exist, such as orthogonality [9] and development-closedness [10]. If however neither (1) nor (2) hold, then no known criteria allows for showing confluence. This is problematic even when non-linearity is only necessary for silly reasons, such as in the following example.

$$\begin{aligned}\Sigma_{\lambda\pi\uparrow} = \quad & \lambda : (\mathtt{t} : \mathsf{tm} \to \mathsf{tm}) \to \mathsf{tm}, \quad @ : (\mathtt{t} : \mathsf{tm}, \mathtt{u} : \mathsf{tm}) \to \mathsf{tm}, \quad \uparrow : (\mathtt{n} : \mathsf{lvl}, \mathtt{m} : \mathsf{lvl}, \mathtt{t} : \mathsf{tm}) \to \mathsf{tm},\\ & \pi : (\mathtt{n} : \mathsf{lvl}, \mathtt{a} : \mathsf{tm}, \mathtt{b} : \mathsf{tm} \to \mathsf{tm}) \to \mathsf{tm}, \quad \mathsf{0} : \mathsf{lvl}, \quad \mathsf{S} : (\mathtt{n} : \mathsf{lvl}) \to \mathsf{lvl}\end{aligned}$$

$$\begin{aligned}\mathcal{R}_{\lambda\pi\uparrow} = \quad & \uparrow^{\mathtt{m}}_{\mathtt{n}} (\pi_{\mathtt{n}}(\mathtt{A}, x.\mathtt{B}(x))) \longmapsto \pi_{\mathtt{m}}(\uparrow^{\mathtt{m}}_{\mathtt{n}} (\mathtt{A}), x. \uparrow^{\mathtt{m}}_{\mathtt{n}} (\mathtt{B}(x))) \qquad && \uparrow^{\mathtt{n}}_{\mathtt{n}} (\mathtt{t}) \longmapsto \mathtt{t}\\ & \lambda(x.\mathtt{t}(x))_{@}\mathtt{u} \longmapsto \mathtt{t}(\mathtt{u}) && \uparrow^{\mathtt{p}}_{\mathtt{m}} (\uparrow^{\mathtt{m}}_{\mathtt{n}} (\mathtt{t})) \longmapsto \uparrow^{\mathtt{p}}_{\mathtt{n}} (\mathtt{t})\end{aligned}$$

Example 1. Consider the signature (where $\mathsf{tm}, \mathsf{lvl}$ are sorts) and the rewrite system given above, containing an excerpt of the rules used when defining a cumulative Tarski-style universe — similar ones can be found in [3]. Because of the beta rule, the rewrite system is not s.n. and because of the other rules the system is also non left-linear. Non-linearity is only used to obtain a finite signature, a pre-requisite for some applications.

Actually, there is no hope of showing confluence for $\mathcal{R}_{\lambda\pi\uparrow}$, given that it is possible to simulate the rewrite system $\mathcal{R}_k = \{\lambda(x.\mathtt{t}(x))_{@}\mathtt{u} \longmapsto \mathtt{t}(\mathtt{u}),\ f(\mathtt{t},\mathtt{t}) \longmapsto a\}$ shown by Klop to be not confluent [7]. Indeed, by taking variables $x : \mathsf{tm} \to \mathsf{lvl}, y : \mathsf{tm}$ we can translate $[\![f(t,u)]\!] :=\uparrow^{x([\![u]\!])}_{x([\![t]\!])} (y)$ and $[\![a]\!] := y$, and then show that $t \longrightarrow u$ implies $[\![t]\!] \longrightarrow [\![u]\!]$ and $[\![t]\!] \longrightarrow u$ implies $t \longrightarrow u'$ for some u' with $[\![u']\!] = u$. Using these two facts it is easy to see that the confluence of $\mathcal{R}_{\lambda\pi\uparrow}$ implies that of $\mathcal{R}_k$.

This counterexample however makes essential use of the fact that we have access to a variable $x : \mathsf{tm} \to \mathsf{lvl}$, allowing us to perform a beta step inside a non-linear position of the lhs in $\uparrow^{\mathtt{n}}_{\mathtt{n}} (\mathtt{t}) \longmapsto \mathtt{t}$. However, in most applications one is only interested in terms containing 0-order variables, and higher-order variables are only used as metavariables to define the rewrite rules. If we instead restrict our attention to confluence over terms containing only variables of order 0 (a property we will call *object confluence*), can we prove that $\mathcal{R}_{\lambda\pi\uparrow}$ satisfies this property?

In this article, we propose a criterion that allows us to do that. More precisely, given two rewrite systems $\mathcal{R}_l$ and $\mathcal{R}_{nl}$ over a signature of order at most 2 such that (1) both are object confluent, (2) $\mathcal{R}_l$ is linear, (3) there are no critical pairs between them and (4) the sorts of non-linear lhs variables of $\mathcal{R}_{nl}$ are inaccessible from the sorts of the rules in $\mathcal{R}_l$, our criterion allows one to conclude object confluence of their union. The proof is a simple adaptation of the proof of confluence by orthogonality, by using condition (4) to show that a $\mathcal{R}_l$ step cannot destroy a $\mathcal{R}_{nl}$ redex. As shown in the end of the article, our criterion proves the object confluence of Example 1.

As a second contribution, we give a reformulation of Nipkow's HRSs in which one never needs to talk about $\beta\eta$-equivalence. This is achieved by adopting a canonical forms only presentation of the simply-typed λ-calculus, and replacing regular substitution by hereditary substitution [6]. This avoids the technicalities of switching $\beta\eta$ representatives and allows for a presentation of higher-order rewriting that we believe can be clearer.

Related work The problem of higher-order confluence with non-left linear rules has been studied by [2] and [3] in the setting of rewriting union β. The notion of *confinement* introduced in [2] was an essential inspiration for us. We omit a detailed discussion because of size constraints, but remark that our criterion's proof is much shorter and less technical.

2 Higher-order rewriting

We start by introducing our reformulation of Nipkow's (Pattern) Higher-order Rewriting Systems (HRSs). We suppose we are given three infinite and disjoint sets of *variables* $\mathcal{V}$, refered to by x, y, z or by letters in typewriter font such as $\mathtt{a}, \mathtt{b}, \mathtt{t}$, *(syntactic) constructors* C, refered to by c, d, f, g, and *sorts* $\mathcal{S}$, refered to by s. A *head* h is either a constructor c or a variable x. We define *arities*, *scopes* and *signatures* by the grammars

$$\boxed{\mathsf{Arity}} \ni \sigma, \tau ::= \delta \to s \qquad \boxed{\mathsf{Scope}} \ni \gamma, \delta ::= \cdot \mid \gamma, x : \tau \qquad \boxed{\mathsf{Sig}} \ni \Sigma ::= \cdot \mid \Sigma, c : \tau$$

and abbreviate $\cdot \to s$ as simply s. We write $\vec{x}_\gamma$ for the sequence of variables in γ, and $\gamma.\delta$ for concatenation. A subscope γ' of γ is a subsequence of γ, written $\gamma' \sqsubseteq \gamma$.

In other works, one usually calls γ a context and τ a simple type. We however prefer to insist here on a different point of view, in which τ is seen as a higher-order generalization of the regular notion of arity.

Given a fixed signature Σ, *terms* and *spines* are mutually defined by the following inference rules. From the perspective of the λ-calculus, our terms can be seen as the simply-typed λ-terms of some base type, and in β-normal η-long form (or canonical form). However, our definition allows us to capture directly the terms of interest, and unlike [8] we never need to speak about the non canonical forms, which play only a bureaucratic role. The definition also clarifies the fact that higher-order rewriting is not (or at least does not need to be seen as) a form of rewriting modulo, but instead rewriting in which one adopts a different notion of substitution (as we will see next).

In the following, when convenient we abbreviate $h(\varepsilon)$ as h. We write $e \in \mathsf{Expr}\ \gamma$ when either $e \in \mathsf{Tm}\ \gamma\ s$ or $e \in \mathsf{Sp}\ \gamma\ \delta$, and we call e an expression. Finally, given a spine $\mathbf{t} \in \mathsf{Sp}\ \gamma\ \delta$ and a variable $x : \gamma_x \to s_x \in \delta$, we write $\mathbf{t}_x \in \mathsf{Tm}\ \gamma.\gamma_x\ s_x$ for the term at position x.

$$h : \delta \to s \in \Sigma \cup \gamma \frac{\mathbf{t} \in \mathsf{Sp}\ \gamma\ \delta}{h(\mathbf{t}) \in \mathsf{Tm}\ \gamma\ s} \qquad \frac{}{\varepsilon \in \mathsf{Sp}\ \gamma\ \cdot} \qquad \frac{\mathbf{t} \in \mathsf{Sp}\ \gamma\ \delta \qquad t \in \mathsf{Tm}\ \gamma.\gamma'\ s}{\mathbf{t}, \vec{x}_{\gamma'}.t \in \mathsf{Sp}\ \gamma\ (\delta, x : \gamma' \to s)}$$

Example 2. If we take $\Sigma = \lambda : (\mathtt{t} : \mathsf{tm} \to \mathsf{tm}) \to \mathsf{tm}, @ : (\mathtt{t} : \mathsf{tm}, \mathtt{u} : \mathsf{tm}) \to \mathsf{tm}$, then $\mathsf{Tm}\ (\vec{x} : \vec{\mathsf{tm}})\ \mathsf{tm}$ contains exactly the λ-term with free variables in $\vec{x}$. This justifies why, unlike in the original formulation of HRS, we do not consider $x.t$ to be a term, as $\lambda(x.t)$ corresponds to a term in the λ-calculus, but $x.t$ or $y.\lambda(x.t)$ do not. Moreover, this makes the restriction of rules to base types in [8] completely automatic in our formulation.

Remark 1. We present the syntax informally using names and α-equivalence as a convenience. However, we expect that everything can be formally carried out using de-Bruijn indices. This requires being more careful with some definitions. For instance, when using indices $\gamma' \sqsubseteq \gamma$ becomes a proof relevant relation: there are for example two inclusions of $(\mathsf{tm}, \mathsf{ty})$ into $(\mathsf{tm}, \mathsf{tm}, \mathsf{ty})$, and thus we would need to be explicit about which inclusion we take. We would also need to introduce a weakening function $\mathsf{wk} : \mathsf{Expr}\ \gamma' \to \gamma' \sqsubseteq \gamma \to \mathsf{Expr}\ \gamma$, whereas in this article we allow ourselves to weaken terms silently.

Substitution Because of our definition of terms, naive substitution would not work: for instance, syntactically replacing x by $z.\mathsf{S}(z)$ and y by $\mathsf{0}$ in $x(y)$ would yield $(z.\mathsf{S}(z))(\mathsf{0})$, which is not a valid term. We instead use *hereditary substitution* [6], which in this case recursively replaces z by $\mathsf{0}$, giving $\mathsf{S}(\mathsf{0})$. This is defined by the following clauses, by lexographic induction on γ_2 and the expression being substituted. In the following, we write just $e[\mathbf{t}]$ instead of $e[\mathbf{t}/\delta]$ if no ambiguity arises.

$$\begin{aligned}
&_[_/\gamma_2] : \mathsf{Tm}\ \gamma_1.\gamma_2.\gamma_3\ s \to \mathsf{Sp}\ \gamma_1\ \gamma_2 \to \mathsf{Tm}\ \gamma_1.\gamma_3\ s \\
&x_i(\mathbf{v})[\mathbf{u}/\gamma_2] := v[\mathbf{v}[\mathbf{u}/\gamma_2]/\delta] && \text{if } x_i : \delta \to s \in \gamma_2 \text{ and } \mathbf{u}_i = \vec{x}_\delta.v \\
&h(\mathbf{v})[\mathbf{u}/\gamma_2] := h(\mathbf{v}[\mathbf{u}/\gamma_2]) && \text{if } h \in \Sigma, \gamma_1, \gamma_3 \\
&_[_/\gamma_2] : \mathsf{Sp}\ \gamma_1.\gamma_2.\gamma_3\ \delta \to \mathsf{Sp}\ \gamma_1\ \gamma_2 \to \mathsf{Sp}\ \gamma_1.\gamma_3\ \delta \\
&\varepsilon[\mathbf{u}/\gamma_2] := \varepsilon \\
&(\mathbf{v}, \vec{y}.t)[\mathbf{u}/\gamma_2] := \mathbf{v}[\mathbf{u}/\gamma_2], \vec{y}.t[\mathbf{u}/\gamma_2]
\end{aligned}$$

We can verify that for all expressions $e \in \mathsf{Expr}\ \gamma_1.\gamma_2.\gamma_3.\gamma_4$ and spines $\mathbf{u} \in \mathsf{Sp}\ \gamma_1\ \gamma_2$ and $\mathbf{t} \in \mathsf{Sp}\ \gamma_1.\gamma_2\ \gamma_3$ we have $e[\mathbf{t}/\gamma_3][\mathbf{u}/\gamma_2] = e[\mathbf{u}/\gamma_2][\mathbf{t}[\mathbf{u}/\gamma_2]/\gamma_3]$.

Sometimes we need a spine $\mathbf{v} \in \mathsf{Sp}\ \gamma\ \gamma$ that satisfies $e[\mathbf{v}] = e$ for all e. Normally one takes $\mathbf{v} = \vec{x}_\gamma$, but in general this is not a valid spine. Instead, we need to define the *identity spine* $\mathsf{id}_\gamma \in \mathsf{Sp}\ \gamma\ \gamma$ by $\mathsf{id}_{(\cdot)} := \varepsilon$ and $\mathsf{id}_{\gamma, x:\delta\to s} := \mathsf{id}_\gamma, \vec{y}_\delta.x(\mathsf{id}_\delta)$. Intuitively, it η-expands each variable in $\vec{x}_\gamma$ so that the resulting sequence is indeed a valid spine. We can now verify that $e[\mathsf{id}_\gamma] = e$ for all $e \in \mathsf{Expr}\ \gamma$, and moreover $\mathsf{id}_\delta[\mathbf{t}] = \mathbf{t}$ for all $\mathbf{t} \in \mathsf{Sp}\ \gamma\ \delta$.

Rewriting Given an expression e, we write $\mathsf{Pos}\ e$ for its set of positions and $\mathsf{FPos}\ e$ for its set of functional positions. For each $p \in \mathsf{Pos}\ e$ let $\gamma_p \in \mathsf{Scope}$ be the scope introduced between the root and p, and $s_p \in \mathcal{S}$ the sort at p. Given $e \in \mathsf{Expr}\ \gamma$ and $p \in \mathsf{Pos}\ e$ we write $e|_p \in \mathsf{Tm}\ \gamma.\gamma_p\ s_p$ for the subterm at position p, and given a term $t \in \mathsf{Tm}\ \gamma.\gamma_p\ s_p$ we write $e\{t\}_p$ for the result of replacing $e|_p$ by t in e.

Remark 2. When dealing with terms that are not explicitly scoped the operation of taking the subterm at position p is not well-defined. Indeed, x is a subterm of $\lambda(x.x)$ but not of $\lambda(y.y)$, however they are equal modulo α-equivalence.

A pattern $e \in \mathsf{Patt}\ \gamma\ \gamma'$ is an expression $e \in \mathsf{Expr}\ \gamma.\gamma'$ in which each variable $x \in \gamma$ appearing at position p occurs applied to $\mathsf{id}_{\gamma''}$ where $\gamma'' \sqsubseteq \gamma'.\gamma_p$. We think of variables in γ as flexible and γ' as rigid. We write $t \in \mathsf{Tm}^\mathsf{P}\ \gamma\ \gamma'\ s$ for a term pattern and $\mathbf{t} \in \mathsf{Sp}^\mathsf{P}\ \gamma\ \gamma'\ \delta$ for a spine pattern. We have $\mathsf{Tm}^\mathsf{P}\ \gamma\ \gamma'\ s \subseteq \mathsf{Tm}\ \gamma.\gamma'\ s$ and $\mathsf{Sp}^\mathsf{P}\ \gamma\ \gamma'\ \delta \subseteq \mathsf{Sp}\ \gamma.\gamma'\ \delta$.

Given a pattern $e \in \mathsf{Patt}\ \gamma\ \gamma'$, we write $\mathsf{ffv}(e)$ for the subscope of γ containing exactly the free flexible variables of e. A pattern is linear if any $x \in \mathsf{ffv}(e)$ occurs only once.

A rewrite rule $\gamma \Vdash t \longmapsto u : s$ is given by $t \in \mathsf{Tm}^{\mathsf{P}}\ \gamma\ \cdot\ s$ and $u \in \mathsf{Tm}\ \gamma\ s$ st $\gamma = \mathsf{ffv}(t)$ and t is not a variable. It is linear if l is a linear pattern. We define the rewrite relation $e \longrightarrow e'$ for $e, e' \in \mathsf{Expr}\ \gamma$ if there is a rewrite rule $\delta \Vdash l \longmapsto r : s$ and position $p \in \mathsf{Pos}\ e$ and spine $\mathbf{v} \in \mathsf{Sp}\ \gamma.\gamma_p\ \delta$ such that $s = s_p$ and $e|_p = l[\mathbf{v}]$ and $e' = e\{r[\mathbf{v}]\}_p$.

Critical pairs Given $t, u \in \mathsf{Tm}\ \delta.\gamma\ s$ we call $\delta \mid \gamma \Vdash t =^? u : s$ a unification problem. A unifier is a spine $\mathbf{v} \in \mathsf{Sp}\ \delta'\ \delta$ st $t[\mathbf{v}] = u[\mathbf{v}]$. When $t, u \in \mathsf{Tm}^{\mathsf{P}}\ \delta\ \gamma\ s$, the problem has either a most general unifer (mgu) or no unifier.

It is known that one of the difficulties when going from first order to higher-order rewriting is adapting the definition of critical pairs. Given $\delta_i \Vdash l_i \longmapsto r_i : s_i$ and $p \in \mathsf{FPos}\ l_1$, if one tries naively to unify $\delta_1.\delta_2 \mid \gamma_p \Vdash l_1|_p =^? l_2 : s_2$, then because the variables in γ_p introduced between the root and p in l_1 do not appear in l_2, any unifier must throw dependencies on such variables away, which is not what is intended. Instead, we first need to add γ_p as dependencies to the variables appearing in l_2. This is achieved by a substitution Nipkow calls a $\vec{x}_{\gamma_p}$-*lifter*, but it can also be understood more algebraically.

Given $\gamma, \delta \in \mathsf{Scope}$, we define the exponential scope $\gamma \rightarrowtail \delta \in \mathsf{Scope}$ by replacing each entry $x : \gamma_x \to s \in \delta$ by $x^* : \gamma.\gamma_x \to s$. We have an evaluation spine pattern $\mathsf{eval}_{\gamma,\delta} \in \mathsf{Sp}^{\mathsf{P}}\ (\gamma \rightarrowtail \delta)\ \gamma\ \delta$, containing at entry $x : \gamma_x \to s \in \delta$ the argument $\vec{x}_{\gamma_x}.x^*(\mathsf{id}_\gamma, \mathsf{id}_{\gamma_x})$. For each $\mathbf{t} \in \mathsf{Sp}\ \gamma'.\gamma\ \delta$ we define its curryfication $\mathsf{cur}\ \mathbf{t} \in \mathsf{Sp}\ \gamma'\ (\gamma \rightarrowtail \delta)$ by replacing each entry $\vec{x}_{\gamma_x}.t$ in $\mathbf{t}$ by $\vec{x}_{\gamma.\gamma_x}.t$. The important property we have is that for all $\mathbf{t} \in \mathsf{Sp}\ \gamma'.\gamma\ \delta$, $\mathsf{cur}\ \mathbf{t}$ is the unique spine satisfying $\mathbf{t} = \mathsf{eval}_{\gamma,\delta}[\mathsf{cur}\ \mathbf{t}/\gamma \rightarrowtail \delta]$. Our notation evidences the fact that $\gamma \rightarrowtail \delta$ is the exponential object in the category of scope and spines, with $\mathsf{eval}_{\gamma,\delta}$ its corresponding evaluation morphism.

We can now define overlaps and critical pairs. A pattern $e \in \mathsf{Patt}\ \gamma\ \gamma'$ overlaps a rewrite rule $\delta \Vdash l \longmapsto r : s$ at functional position $p \in \mathsf{FPos}\ e$ if the unification problem

$$\gamma.(\gamma'.\gamma_p \rightarrowtail \delta) \mid \gamma'.\gamma_p \Vdash e|_p =^? l[\mathsf{eval}_{\gamma'.\gamma_p,\delta}] : s$$

has a unifier — in which case, it also has a most general one. A pattern overlap is proper if $e = l$ implies $p \neq \varepsilon$. A rule overlap is given by two rules $\delta_1 \Vdash l_1 \longmapsto r_1 : s_1$ and $\delta_2 \Vdash l_2 \longmapsto r_2 : s_2$ and a functional position $p \in \mathsf{FPos}\ l_1$ st l_1 properly overlaps $l_2 \longmapsto r_2$ at position p. Each rule overlap gives rise to a critical pair $\langle r_1[\mathbf{v}], l_1\{r_2[\mathsf{eval}_{\gamma_p,\delta_2}]\}_p[\mathbf{v}]\rangle$, where $\mathbf{v}$ is the mgu of the associated unification problem.

Superdevelopments Like with many proofs of confluence, ours will employ Aczel's superdevelopments [1], defined by the following rules.

$$\textsc{Rule}\ \ \delta \Vdash l \longmapsto r : s \in \mathcal{R}\ \frac{f(\mathbf{u}) = l[\mathbf{v}] \qquad \mathbf{t} \Longrightarrow \mathbf{u} \in \mathsf{Sp}\ \gamma\ \delta_f}{f(\mathbf{t}) \Longrightarrow r[\mathbf{v}] \in \mathsf{Tm}\ \gamma\ s} \qquad \textsc{Head}\ \ h : \delta \to s \in \gamma \text{ or } \Sigma\ \frac{\mathbf{v} \Longrightarrow \mathbf{v}' \in \mathsf{Sp}\ \gamma\ \delta}{h(\mathbf{v}) \Longrightarrow h(\mathbf{v}') \in \mathsf{Tm}\ \gamma\ s}$$

$$\textsc{EmptySp}\ \frac{}{\varepsilon \Longrightarrow \varepsilon \in \mathsf{Sp}\ \gamma\ \cdot} \qquad \textsc{ExtSp}\ \frac{\mathbf{t} \Longrightarrow \mathbf{t}' \in \mathsf{Sp}\ \gamma\ \delta \qquad t \Longrightarrow t' \in \mathsf{Tm}\ \gamma.\gamma_x\ s}{\mathbf{t}, \vec{x}_{\gamma_x}.t \Longrightarrow \mathbf{t}', \vec{x}_{\gamma_x}.t' \in \mathsf{Sp}\ \gamma\ (\delta, x : \gamma_x \to s)}$$

Recall that we have $\longrightarrow \subseteq \Longrightarrow \subseteq \longrightarrow^*$ and thus $\longrightarrow^* = \Longrightarrow^*$. Moreover, superdevelopments are closed under substitution: if $e \Longrightarrow e' \in \mathsf{Expr}\ \gamma_1.\gamma_2.\gamma_3$ and $\mathbf{u} \Longrightarrow \mathbf{u}' \in \mathsf{Sp}\ \gamma_1\ \gamma_2$ then $e[\mathbf{u}/\gamma_2] \Longrightarrow e'[\mathbf{u}'/\gamma_2] \in \mathsf{Expr}\ \gamma_1.\gamma_3$. The following proposition is at the heart of most proofs of confluence by orthogonality [8]. We will also need it to show our criterion.

Proposition 1. *Let $e \in \mathsf{Patt}\ \delta\ \gamma'$ be a linear pattern that does not overlap any lhs of $\mathcal{R}$ and suppose that for some $\mathbf{v} \in \mathsf{Sp}\ \gamma\ \delta$ and e' we have $e[\mathbf{v}/\delta] \Longrightarrow e'$. Then we have $\mathbf{v}' \Longrightarrow \mathbf{v}'' \in \mathsf{Sp}\ \gamma\ \mathsf{ffv}(e)$ with $\mathbf{v}'$ a subspine of $\mathbf{v}$ and $e[\mathbf{v}''/\mathsf{ffv}(e)] = e'$.*

Proof. By induction on e.

- $e = x(\mathsf{id}_{\gamma_x})$ for some $x :: \gamma_x \to s \in \delta$: Then $\mathbf{v}_x = \mathbf{v}_x[\mathsf{id}_{\gamma_x}] = x(\mathsf{id}_{\gamma_x})[\mathbf{v}] \Longrightarrow e'$. Because rewriting cannot introduce new free variables and $\mathbf{v}_x \in \mathsf{Tm}\ \gamma.\gamma_x\ s$, then we must have $e' \in \mathsf{Tm}\ \gamma.\gamma_x\ s$. We thus have $\vec{x}_{\gamma_x}.\mathbf{v}_x \Longrightarrow \vec{x}_{\gamma_x}.e' \in \mathsf{Sp}\ \gamma\ (x : \gamma_x \to s)$ and $x(\mathsf{id}_{\gamma_x})[\vec{x}_{\gamma_x}.e'] = e'$.
- $e = h(\mathbf{t})$ with $h \neq x \in \delta$ and $\mathbf{t} \in \mathsf{Sp}^{\mathsf{P}}\ \delta\ \gamma'$ linear : We reason by case analysis on the superdevelopement.
  - Rew : Then for some rule $\delta_{lr} \Vdash l \longmapsto r :: s_{lr}$ and spine $\mathbf{s} \in \mathsf{Sp}\ \gamma\ \delta_{lr}$ we have $e[\mathbf{v}/\delta] = h(\mathbf{t}[\mathbf{v}/\delta]) = f(\mathbf{u}) \Longrightarrow r[\mathbf{s}] = e'$ with $\mathbf{u} \Longrightarrow \mathbf{u}' \in \mathsf{Sp}\ \gamma\ \delta_f$ and $f(\mathbf{u}') = l[\mathbf{s}]$. We thus have $h = f$ and $\mathbf{t}[\mathbf{v}/\delta] \Longrightarrow \mathbf{u}'$. Because $f(\mathbf{t})$ does not overlap any lhs of $\mathcal{R}$, then $\mathbf{t}$ also does not. Therefore, by the i.h. we have $\mathbf{v}' \Longrightarrow \mathbf{v}'' \in \mathsf{Sp}\ \gamma\ \mathsf{ffv}(\mathbf{t})$ with $\mathbf{v}'$ a subspine of $\mathbf{v}$ and $\mathbf{t}[\mathbf{v}''/\mathsf{ffv}(\mathbf{t})] = \mathbf{u}'$. But then $f(\mathbf{t})[\mathbf{v}''/\mathsf{ffv}(\mathbf{t})] = f(\mathbf{t}[\mathbf{v}''/\mathsf{ffv}(\mathbf{t})]) = f(\mathbf{u}') = l[\mathbf{s}]$, implying that $f(\mathbf{t})$ overlaps l at position ε, a contradiction.
  - Head: We have $e[\mathbf{v}/\delta] = h(\mathbf{t}[\mathbf{v}/\delta]) \Longrightarrow h(\mathbf{t}') = e'$ with $\mathbf{t}[\mathbf{v}/\delta] \Longrightarrow \mathbf{t}'$. The result follows easily by using the i.h.
- $e = \varepsilon$: We have $e' = \varepsilon$, and thus we have $\varepsilon \Longrightarrow \varepsilon \in \mathsf{Sp}\ \gamma\ \cdot$ with $e'[\varepsilon] = \varepsilon[\varepsilon] = \varepsilon$.
- $e = \mathbf{t}, \vec{x}_{\gamma_x}.t \in \mathsf{Sp}^{\mathsf{P}}\ \delta\ \gamma\ (\delta', x : \gamma_x \to s)$: We have $\mathbf{t} \in \mathsf{Sp}^{\mathsf{P}}\ \delta\ \gamma'\ \delta'$ and $t \in \mathsf{Tm}^{\mathsf{P}}\ \delta\ \gamma'.\gamma_x\ s$, and $e' = \mathbf{t}', \vec{x}_{\gamma_x}.t'$ with $\mathbf{t}[\mathbf{v}/\delta] \Longrightarrow \mathbf{t}'$ and $t[\mathbf{v}/\delta] \Longrightarrow t'$. By the i.h. we have $\mathbf{v}_1 \Longrightarrow \mathbf{v}'_1 \in \mathsf{Sp}\ \gamma\ \mathsf{ffv}(\mathbf{t})$ and $\mathbf{t}[\mathbf{v}'_1/\delta_1] = \mathbf{t}'$ and $\mathbf{v}_2 \Longrightarrow \mathbf{v}'_2 \in \mathsf{Sp}\ \gamma\ \mathsf{ffv}(t)$ and $t[\mathbf{v}'_2/\delta_2] = t'$, with $\mathbf{v}_1$ and $\mathbf{v}_2$ subspines of $\mathbf{v}$. Because $\mathbf{t}, \vec{x}_{\gamma_x}.t$ is linear, $\mathsf{ffv}(\mathbf{t})$ and $\mathsf{ffv}(t)$ are disjoint, and we can merge $\mathbf{v}_1$ and $\mathbf{v}_2$ into $\mathbf{v}_{12} \in \mathsf{Sp}\ \gamma\ \mathsf{ffv}(\mathbf{t}, \vec{x}_{\gamma_x}.t)$, and merge $\mathbf{v}'_1$ and $\mathbf{v}'_2$ into $\mathbf{v}'_{12} \in \mathsf{Sp}\ \gamma\ \mathsf{ffv}(\mathbf{t}, \vec{x}_{\gamma_x}.t)$ such that $\mathbf{v}_{12} \Longrightarrow \mathbf{v}'_{12}$. Finally, we have $(\mathbf{t}, \vec{x}_{\gamma_x}.t)[\mathbf{v}'_{12}/\mathsf{ffv}(\mathbf{t}, \vec{x}_{\gamma_x}.t)] = \mathbf{t}[\mathbf{v}'_1/\mathsf{ffv}(\mathbf{t})], \vec{x}_{\gamma_x}.t[\mathbf{v}'_2/\mathsf{ffv}(t)] = \mathbf{t}', \vec{x}_{\gamma_x}.t'$. □

Corollary 1. *Let $\delta \Vdash l \longmapsto r : s$ be a linear rule that does not overlap any rule in $\mathcal{R}$. If $l[\mathbf{v}] = f(\mathbf{t})$ and $\mathbf{t} \Longrightarrow \mathbf{t}' \in \mathsf{Sp}\ \gamma\ \delta_f$, then there is $\mathbf{v}'$ with $\mathbf{v} \Longrightarrow \mathbf{v}' \in \mathsf{Sp}\ \gamma\ \delta$ st $l[\mathbf{v}'] = f(\mathbf{t}')$.*

Proof. We have $\delta \Vdash l \longmapsto r : s$ and $l[\mathbf{v}] = f(\mathbf{t}) \in \mathsf{Tm}\ \gamma\ s$ for some $\mathbf{t} \Longrightarrow \mathbf{t}' \in \mathsf{Sp}\ \gamma\ \delta_f$ and $\mathbf{v} \in \mathsf{Sp}\ \gamma\ \delta$. This implies that l must be of the form $f(\mathbf{u})$ with $\mathbf{u} \in \mathsf{Sp}\ \gamma\ \delta_f$, and thus we have $l[\mathbf{v}] = f(\mathbf{u}[\mathbf{v}]) = f(\mathbf{t})$ and therefore $\mathbf{u}[\mathbf{v}] \Longrightarrow \mathbf{t}'$. Because $f(\mathbf{u}) \longmapsto r$ does not overlap any rule in $\mathcal{R}$, then if the pattern $f(\mathbf{u})$ overlaps a rule the overlap can only be improper (at the head). Therefore, the pattern $\mathbf{u}$ does not overlap any rule. By Proposition 1, we get $\mathbf{v}' \Longrightarrow \mathbf{v}'' \in \mathsf{Sp}\ \gamma\ \mathsf{ffv}(\mathbf{u})$ with $\mathbf{v}'$ a subspine of $\mathbf{v}$ and $\mathbf{u}[\mathbf{v}''/\mathsf{ffv}(\mathbf{u})] = \mathbf{t}'$. Because $\delta \Vdash f(\mathbf{u}) \longmapsto r : s$ is a rule, we have by hypothesis $\mathsf{ffv}(\mathbf{u}) = \mathsf{ffv}(f(\mathbf{u})) = \delta$, and thus we have $\mathbf{v} = \mathbf{v}' \Longrightarrow \mathbf{v}'' \in \mathsf{Sp}\ \gamma\ \delta$. Finally, we have $l[\mathbf{v}''] = f(\mathbf{u}[\mathbf{v}'']) = f(\mathbf{t}')$. □

3 A confluence criterion for non-left linearity

Define the order of an arity, of a scope and of a signature by $\mathsf{ord}(\gamma \to s) = 1 + \mathsf{ord}(\gamma)$, $\mathsf{ord}(\cdot) = -1$, $\mathsf{ord}(\gamma, x : \tau) = \max\{\mathsf{ord}(\gamma), \mathsf{ord}(\tau)\}$, $\mathsf{ord}(\Sigma, c : \tau) = \max\{\mathsf{ord}(\Sigma), \mathsf{ord}(\tau)\}$. Note then that the variables of order zero are the ones whose arity is just a sort.

One can remark that for most rewrite systems of interest the underlying signature Σ is of order ≤ 2, and variables of order > 0 are only needed for defining the rewrite rules. For instance, this is the case of the λ-calculus, where one needs a 1st order variable $\mathtt{t}$ to play the role of a metavariable in the rule $\lambda(x.\mathtt{t}(x))_@\mathtt{u} \longmapsto \mathtt{t}(\mathtt{u})$, but when translating a λ-term into its HRS representation one only uses zero order variables. This is also the case of most logics and type theories.

The restriction to signatures of order ≤ 2 is even baked into the definition of second-order formalisms, such as in [5]. There, one distinguishes between variables (order 0) and metavariables (order 1), and confluence of terms without metavariables is called *object confluence*. In the following, we rephrase this notion in the setting of 2nd order HRSs.

Suppose now that the underlying signature Σ is of order ≤ 2. A term $t \in \mathsf{Tm}\ \gamma\ s$ is an object term t if $\mathsf{ord}(\gamma) \leq 0$. A spine $\mathbf{t} \in \mathsf{Sp}\ \gamma\ \delta$ is an object spine if $\mathsf{ord}(\gamma) \leq 0$ and $\mathsf{ord}(\delta) \leq 1$. We refer to them generically as object expressions. Note that if e is an object expression then all terms and spines appearing in e are object expressions. Indeed, because Σ is of order at most 2, each constructor f can only bind 0-order variables. A rewrite system $\mathcal{R}$ is object confluent if the rewriting relation restricted to object expressions is confluent.

Given two sorts s, s' we say that s is accessible from s' (written $s' \preceq s$) if there is some object term t of sort s and position p such that $t|_p$ is of sort s'. Given a rewrite system $\mathcal{R}$ we write $\mathcal{R} \preceq s$ if for some $\delta \Vdash l \longmapsto r : s'$ we have $s' \preceq s$. Note that this notion is only interesting because we only consider object terms: if one has access to all variables, the condition is always verified by taking $t = x(y)$ with $x : s' \to s$ and $y : s'$. It is easy to see that accessibility is decidable when the signature is finite.

Lemma 1. *If $t \in \mathsf{Tm}\ \gamma\ s$ is an object term and $t \Longrightarrow_{\mathcal{R}} t'$ with $\mathcal{R} \npreceq s$ then $t = t'$.*

Proof. We prove this by induction on the superdevelopement, simultaneously with a similar result for spines: if $\mathbf{t} \in \mathsf{Sp}\ \gamma\ \delta$ is an object spine st for all $x : \gamma_x \to s \in \delta$ we have $\mathcal{R} \npreceq s$, then $\mathbf{t} \Longrightarrow \mathbf{t}'$ implies $\mathbf{t} = \mathbf{t}'$. ◻

The heart of our proof is the following proposition, which is very similar to Proposition 1 but replaces linearity by an inaccessibility condition.

Proposition 2. *Let $e \in \mathsf{Patt}\ \delta\ \gamma'$ be a pattern that does not overlap any lhs of $\mathcal{R}$, with $\mathsf{ord}(\delta) \leq 1$ and such that for any $x : \gamma_x \to s_x \in \delta$ occurring non-linearly in e we have $\mathcal{R} \npreceq s_x$. If for some $\mathbf{v} \in \mathsf{Sp}\ \gamma\ \delta$ and e' we have $e[\mathbf{v}/\delta] \Longrightarrow e'$ with $e[\mathbf{v}/\delta]$ an object expression, then we have $\mathbf{v}' \Longrightarrow \mathbf{v}'' \in \mathsf{Sp}\ \gamma\ \mathsf{ffv}(e)$ with $\mathbf{v}'$ a subspine of $\mathbf{v}$ and $e[\mathbf{v}''/\mathsf{ffv}(e)] = e'$.*

Proof. By induction on e.

- $e = x(\mathsf{id}_{\gamma_x})$ for some $x :: \gamma_x \to s \in \delta$: Then $\mathbf{v}_x = \mathbf{v}_x[\mathsf{id}_{\gamma_x}] = x(\mathsf{id}_{\gamma_x})[\mathbf{v}] \Longrightarrow e'$. Because rewriting cannot introduce new free variables and $\mathbf{v}_x \in \mathsf{Tm}\ \gamma.\gamma_x\ s$, then we must have $e' \in \mathsf{Tm}\ \gamma.\gamma_x\ s$. We thus have $\vec{x}_{\gamma_x}.\mathbf{v}_x \Longrightarrow \vec{x}_{\gamma_x}.e' \in \mathsf{Sp}\ \gamma\ (x : \gamma_x \to s)$ and $x(\mathsf{id}_{\gamma_x})[\vec{x}_{\gamma_x}.e'] = e'$.
- $e = h(\mathbf{t})$ with $h \neq x \in \delta$ and $\mathbf{t} \in \mathsf{Sp}^{\mathsf{P}}\ \delta\ \gamma'$: We reason by case analysis on the superdevelopement.
  - Rew : Then for some rule $\delta_{lr} \Vdash l \longmapsto r :: s_{lr}$ and spine $\mathbf{s} \in \mathsf{Sp}\ \gamma\ \delta_{lr}$ we have $e[\mathbf{v}/\delta] = h(\mathbf{t}[\mathbf{v}/\delta]) = f(\mathbf{u}) \Longrightarrow r[\mathbf{s}] = e'$ with $\mathbf{u} \Longrightarrow \mathbf{u}' \in \mathsf{Sp}\ \gamma\ \delta_f$ and $f(\mathbf{u}') = l[\mathbf{s}]$. We thus have $h = f$ and $\mathbf{t}[\mathbf{v}/\delta] \Longrightarrow \mathbf{u}'$. Because $f(\mathbf{t})$ does not overlap any lhs of $\mathcal{R}$, then $\mathbf{t}$ also does not. Therefore, by the i.h. we have $\mathbf{v}' \Longrightarrow \mathbf{v}'' \in \mathsf{Sp}\ \gamma\ \mathsf{ffv}(\mathbf{t})$ with $\mathbf{v}'$ a subspine of $\mathbf{v}$ and $\mathbf{t}[\mathbf{v}''/\mathsf{ffv}(\mathbf{t})] = \mathbf{u}'$. But then $f(\mathbf{t})[\mathbf{v}''/\mathsf{ffv}(\mathbf{t})] = f(\mathbf{t}[\mathbf{v}''/\mathsf{ffv}(\mathbf{t})]) = f(\mathbf{u}') = l[\mathbf{s}]$, implying that $f(\mathbf{t})$ overlaps l at position ε, a contradiction.
  - Head: We have $e[\mathbf{v}/\delta] = h(\mathbf{t}[\mathbf{v}/\delta]) \Longrightarrow h(\mathbf{t}') = e'$ with $\mathbf{t}[\mathbf{v}/\delta] \Longrightarrow \mathbf{t}'$. The result follows easily by using the i.h.
- $e = \varepsilon$: We have $e' = \varepsilon$, and thus we have $\varepsilon \Longrightarrow \varepsilon \in \mathsf{Sp}\ \gamma\ \cdot$ with $e'[\varepsilon] = \varepsilon[\varepsilon] = \varepsilon$.
- $e = \mathbf{t}, \vec{x}_{\gamma_x}.t \in \mathsf{Sp}^{\mathsf{P}}\ \delta\ \gamma\ (\delta', x : \gamma_x \to s)$: We have $\mathbf{t} \in \mathsf{Sp}^{\mathsf{P}}\ \delta\ \gamma'\ \delta'$ and $t \in \mathsf{Tm}^{\mathsf{P}}\ \delta\ \gamma'.\gamma_x\ s$, and $e' = \mathbf{t}', \vec{x}_{\gamma_x}.t'$ with $\mathbf{t}[\mathbf{v}/\delta] \Longrightarrow \mathbf{t}'$ and $t[\mathbf{v}/\delta] \Longrightarrow t'$. By the i.h. we have $\mathbf{v}_1 \Longrightarrow \mathbf{v}'_1 \in \mathsf{Sp}\ \gamma\ \mathsf{ffv}(\mathbf{t})$ and $\mathbf{t}[\mathbf{v}'_1/\delta_1] = \mathbf{t}'$ and $\mathbf{v}_2 \Longrightarrow \mathbf{v}'_2 \in \mathsf{Sp}\ \gamma\ \mathsf{ffv}(t)$ and $t[\mathbf{v}'_2/\delta_2] = t'$, with $\mathbf{v}_1$ and $\mathbf{v}_2$ subspines of $\mathbf{v}$. We can merge $\mathbf{v}_1$ and $\mathbf{v}_2$ into a spine $\mathbf{v}_{12} \in \mathsf{Sp}\ \gamma\ \mathsf{ffv}(\mathbf{t}, \vec{x}_\gamma.t)$, however it is not immediately clear that we can do the same for $\mathbf{v}'_1$ and $\mathbf{v}'_2$. Let us show why.

Because $\mathsf{ord}(\delta) \leq 1$ and $\mathsf{ffv}(\mathbf{t}) \sqsubseteq \delta$ and $\mathsf{ffv}(t) \sqsubseteq \delta$ then $\mathsf{ord}(\mathsf{ffv}(\mathbf{t})) \leq 1$ and $\mathsf{ord}(\mathsf{ffv}(t)) \leq 1$. Moreover, because $(\mathbf{t}, \vec{x}_{\gamma_x}.t)[\mathbf{v}/\delta] \in \mathsf{Sp}\ \gamma.\gamma'\ (\delta', x : \gamma_x \to s)$ is an object expression, we have $\mathsf{ord}(\gamma) \leq 0$. Therefore, $\mathbf{v}_1, \mathbf{v}_2$ are object spines. If $y : \gamma_y \to s_y$ occurs in both $\mathsf{ffv}(\mathbf{t})$ and $\mathsf{ffv}(t)$ then by hypothesis we have $\mathcal{R} \not\preceq s_y$ and thus by Lemma 1 we must have $(\mathbf{v}'_1)_y = (\mathbf{v}_1)_y = (\mathbf{v}_2)_y = (\mathbf{v}'_2)_y$. Therefore, $\mathbf{v}'_1$ and $\mathbf{v}'_2$ agree on their intersection, hence it is possible to merge them into $\mathbf{v}'_{12} \in \mathsf{Sp}\ \gamma\ \mathsf{ffv}(\mathbf{t}, \vec{x}_{\gamma_x}.t)$. We then have $\mathbf{v}_{12} \Longrightarrow \mathbf{v}'_{12}$ and $(\mathbf{t}, \vec{x}_{\gamma_x}.t)[\mathbf{v}'_{12}/\mathsf{ffv}(\mathbf{t}, \vec{x}_{\gamma_x}.t)] = \mathbf{t}[\mathbf{v}'_1/\mathsf{ffv}(\mathbf{t})], \vec{x}_{\gamma_x}.t[\mathbf{v}'_2/\mathsf{ffv}(t)] = \mathbf{t}', \vec{x}_{\gamma_x}.t'$ as required. □

Corollary 2. *Let $\delta \Vdash l \longmapsto r : s$ be a rule that does not overlap any rule in $\mathcal{R}$, and such that for any $x : \gamma_x \to s_x \in \delta$ occurring non-linearly in l we have $\mathcal{R} \not\preceq s_x$. If $l[\mathbf{v}] \in \mathsf{Tm}\ \gamma\ s$ is an object term with $l[\mathbf{v}] = f(\mathbf{t})$ and $\mathbf{t} \Longrightarrow_{\mathcal{R}} \mathbf{t}' \in \mathsf{Sp}\ \gamma\ \delta_f$ then we have $\mathbf{v} \Longrightarrow_{\mathcal{R}} \mathbf{v}' \in \mathsf{Sp}\ \gamma\ \delta$ and $l[\mathbf{v}'] = f(\mathbf{t}')$.*

Proof. We have $\delta \Vdash l \longmapsto r : s$ and $l[\mathbf{v}] = f(\mathbf{t}) \in \mathsf{Tm}\ \gamma\ s$ for some $\mathbf{t} \Longrightarrow \mathbf{t}' \in \mathsf{Sp}\ \gamma\ \delta_f$ and $\mathbf{v} \in \mathsf{Sp}\ \gamma\ \delta$. This implies that l must be of the form $f(\mathbf{u})$ with $\mathbf{u} \in \mathsf{Sp}\ \gamma\ \delta_f$, and thus we have $l[\mathbf{v}] = f(\mathbf{u}[\mathbf{v}]) = f(\mathbf{t})$ and therefore $\mathbf{u}[\mathbf{v}] \Longrightarrow \mathbf{t}'$. Because $f(\mathbf{u}) \longmapsto r$ does not overlap any rule in $\mathcal{R}$, then if the pattern $f(\mathbf{u})$ overlaps a rule the overlap can only be improper (at the head). Therefore, the pattern $\mathbf{u}$ does not overlap any rule. Moreover, because $l[\mathbf{v}]$ is an object term we have $\mathsf{ord}(\gamma) \leq 0$, and because $\mathsf{ord}(\Sigma) \leq 2$ then $\mathsf{ord}(\delta_f) \leq 1$, showing that $\mathbf{u}[\mathbf{v}] \in \mathsf{Sp}\ \gamma\ \delta_f$ is an object spine. Finally, because l is a pattern, $\mathsf{ord}(\Sigma) \leq 2$ and $\delta = \mathsf{ffv}(l)$ it is not hard to see that $\mathsf{ord}(\delta) \leq 1$ — indeed, each occurrence of a flexible variable $\mathtt{x} : \gamma_x \to s_x \in \delta$ must be of the form $\mathtt{x}(\mathsf{id}_{\gamma_x})$, but because 2nd order constructors only bind variables of order 0, and because γ_x is a subscope of the scope of bound variables at this occurrence of $\mathtt{x}$, then γ_x must be of order at most 0. Thus, by Proposition 2 we get $\mathbf{v}' \Longrightarrow \mathbf{v}'' \in \mathsf{Sp}\ \gamma\ \mathsf{ffv}(\mathbf{u})$ with $\mathbf{v}'$ a subspine of $\mathbf{v}$ and $\mathbf{u}[\mathbf{v}''/\mathsf{ffv}(\mathbf{u})] = \mathbf{t}'$. Because $\delta \Vdash f(\mathbf{u}) \longmapsto r : s$ is a rule, we have by hypothesis $\mathsf{ffv}(\mathbf{u}) = \mathsf{ffv}(f(\mathbf{u})) = \delta$, and thus we have $\mathbf{v} = \mathbf{v}' \Longrightarrow \mathbf{v}'' \in \mathsf{Sp}\ \gamma\ \delta$. Finally, we have $l[\mathbf{v}''] = f(\mathbf{u}[\mathbf{v}'']) = f(\mathbf{t}')$. □

We are now ready to give our criterion.

Theorem 1. *Let $\mathcal{R}_l$ and $\mathcal{R}_{nl}$ be two rewriting systems on Σ with $\mathsf{ord}(\Sigma) \leq 2$ such that*

(A) $\mathcal{R}_{nl}$ and $\mathcal{R}_l$ are object confluent

(B) $\mathcal{R}_l$ is left-linear

(C) There are no critical pairs between $\mathcal{R}_l$ and $\mathcal{R}_{nl}$

(D) For each $\delta \Vdash t \longmapsto u : s \in \mathcal{R}_{nl}$ and $x : \gamma_x \to s_x \in \delta$ with x occurring non-linearly in t, we have $\mathcal{R}_l \not\preceq s_x$

Then $\mathcal{R}_l \cup \mathcal{R}_{nl}$ is object confluent.

Proof. Because $\mathcal{R}_{nl}$ and $\mathcal{R}_l$ are both object confluent, it suffices to show that $\mathcal{R}_{nl}$ and $\mathcal{R}_l$ commute on object expressions. We show ${}_{\mathcal{R}_{nl}}\!\Longleftarrow\Longrightarrow_{\mathcal{R}_l} \subseteq \Longrightarrow_{\mathcal{R}_l}{}_{\mathcal{R}_{nl}}\!\Longleftarrow$ on object expressions, by induction on the superdevelopments. The proof is essentially the same as the one for orthogonal systems [8], but using Corollary 2 for the case $f(\mathbf{t}) \Longrightarrow_{\mathcal{R}_{nl}} r[\mathbf{v}]$. The case EmptySp/EmptySp is trivial and the cases ExtSp/ExtSp and Head/Head follow easily by the i.h., so we consider only the cases Head/Rew, Rew/Head and Rew/Rew. They are illustrated in Figure 1.

- Head/Rew: We have $f(\mathbf{u})\ {}_{\mathcal{R}_{nl}}\!\Longleftarrow f(\mathbf{t}) \Longrightarrow_{\mathcal{R}_l} r[\mathbf{v}]$ where $\mathbf{u}\ {}_{\mathcal{R}_{nl}}\!\Longleftarrow \mathbf{t} \Longrightarrow_{\mathcal{R}_l} \mathbf{t}'$ and $f(\mathbf{t}') = l[\mathbf{v}]$ for some $l \longmapsto r \in \mathcal{R}_l$. By i.h. we have $\mathbf{u} \Longrightarrow_{\mathcal{R}_l} \mathbf{s}\ {}_{\mathcal{R}_{nl}}\!\Longleftarrow \mathbf{t}'$ for some $\mathbf{s}$. Because $l[\mathbf{v}] = f(\mathbf{t}')$ and $\mathbf{t}' \Longrightarrow_{\mathcal{R}_{nl}} \mathbf{s}$ and l does not properly overlap no rule in $\mathcal{R}_{nl}$ then by Corollary 1 there is $\mathbf{v}'$ with $\mathbf{v} \Longrightarrow_{\mathcal{R}_{nl}} \mathbf{v}'$ and $f(\mathbf{s}) = l[\mathbf{v}']$. Thus we also have $r[\mathbf{v}] \Longrightarrow_{\mathcal{R}_{nl}} r[\mathbf{v}']$. We thus conclude $f(\mathbf{u}) \Longrightarrow_{\mathcal{R}_l} r[\mathbf{v}']\ {}_{\mathcal{R}_{nl}}\!\Longleftarrow r[\mathbf{v}]$.

- Head/Rew: We have $r[\mathbf{v}] \;{}_{\mathcal{R}_{nl}}\!\Longleftarrow f(\mathbf{t}) \Longrightarrow_{\mathcal{R}_l} f(\mathbf{u})$ where $\mathbf{t}' \;{}_{\mathcal{R}_{nl}}\!\Longleftarrow \mathbf{t} \Longrightarrow_{\mathcal{R}_l} \mathbf{u}$ and $f(\mathbf{t}') = l[\mathbf{v}]$ for some $l \longmapsto r \in \mathcal{R}_{nl}$. By i.h. we have $\mathbf{t}' \Longrightarrow_{\mathcal{R}_l} \mathbf{s} \;{}_{\mathcal{R}_{nl}}\!\Longleftarrow \mathbf{u}$ for some $\mathbf{s}$. l does not properly overlap no rule in $\mathcal{R}_l$ and for all variables $x : \gamma_x \to s_x$ appearing non-linearly in l we have $\mathcal{R}_l \not\lesssim s_x$. Moreover, $l[\mathbf{v}]$ is an object term. Therefore, from $l[\mathbf{v}] = f(\mathbf{t}')$ and $\mathbf{t}' \Longrightarrow_{\mathcal{R}_l} \mathbf{s}$ and Corollary 2, we get $\mathbf{v}'$ with $\mathbf{v} \Longrightarrow_{\mathcal{R}_l} \mathbf{v}'$ and $f(\mathbf{s}) = l[\mathbf{v}']$. Thus we also have $r[\mathbf{v}] \Longrightarrow_{\mathcal{R}_l} r[\mathbf{v}']$. We thus conclude $r[\mathbf{v}] \Longrightarrow_{\mathcal{R}_l} r[\mathbf{v}'] \;{}_{\mathcal{R}_{nl}}\!\Longleftarrow f(\mathbf{u})$.

- Rew/Rew: We have $r_2[\mathbf{v}_2] \;{}_{\mathcal{R}_{nl}}\!\Longleftarrow f(\mathbf{t}) \Longrightarrow_{\mathcal{R}_l} r_1[\mathbf{v}_1]$ where $\mathbf{t}_2 \;{}_{\mathcal{R}_{nl}}\!\Longleftarrow \mathbf{t} \Longrightarrow_{\mathcal{R}_l} \mathbf{t}_1$ and $f(\mathbf{t}_1) = l_1[\mathbf{v}_1]$ for some $l_1 \longmapsto r_1 \in \mathcal{R}_l$ and $f(\mathbf{t}_2) = l_2[\mathbf{v}_2]$ for some $l_2 \longmapsto r_2 \in \mathcal{R}_{nl}$. By i.h. we have $\mathbf{t}_2 \Longrightarrow_{\mathcal{R}_l} \mathbf{s} \;{}_{\mathcal{R}_{nl}}\!\Longleftarrow \mathbf{t}_1$ for some $\mathbf{s}$. Because $f(\mathbf{t}_1) = l_1[\mathbf{v}_1]$ and $\mathbf{t}_1 \Longrightarrow_{\mathcal{R}_{nl}} \mathbf{s}$ and l_1 does not properly overlap no rule in $\mathcal{R}_{nl}$ then by Corollary 1 there is $\mathbf{v}_1'$ with $\mathbf{v}_1 \Longrightarrow_{\mathcal{R}_{nl}} \mathbf{v}_1'$ and $f(\mathbf{s}) = l_1[\mathbf{v}_1']$. Moreover, l_2 does not properly overlap no rule in $\mathcal{R}_l$ and for all variables $x : \gamma_x \to s_x$ appearing non-linearly in l_2 we have $\mathcal{R}_l \not\lesssim s_x$, and $l_2[\mathbf{v}_2]$ is an object term. Therefore, from $l_2[\mathbf{v}_2] = f(\mathbf{t}_2)$ and $\mathbf{t}_2 \Longrightarrow_{\mathcal{R}_l} \mathbf{s}$ and Corollary 2, we get $\mathbf{v}_2'$ with $\mathbf{v}_2 \Longrightarrow_{\mathcal{R}_l} \mathbf{v}_2'$ and $f(\mathbf{s}) = l_2[\mathbf{v}_2']$. We therefore have $l_1[\mathbf{v}_1'] = l_2[\mathbf{v}_2']$, but because $\mathcal{R}_l$ and $\mathcal{R}_{nl}$ do not properly overlap, this ovelap can only be improper, meaning that $l_1 = l_2$ and thus $\mathbf{v}_1' = \mathbf{v}_2'$. Finally, using $\mathbf{v}_1 \Longrightarrow_{\mathcal{R}_l} \mathbf{v}_1' = \mathbf{v}_2' \;{}_{\mathcal{R}_{nl}}\!\Longleftarrow \mathbf{v}_2$ we close the diagram with $r_1[\mathbf{v}_1] \Longrightarrow_{\mathcal{R}_l} r_1[\mathbf{v}_1'] = r_2[\mathbf{v}_2'] \;{}_{\mathcal{R}_{nl}}\!\Longleftarrow r_2[\mathbf{v}_2]$. □

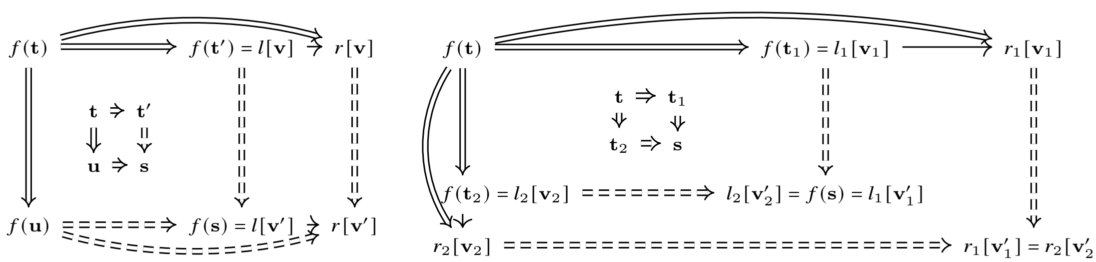

Figure 1: Cases Head/Rew and Rew/Head (left), and case Rew/Rew (right)

We can now prove that the system of Example 1 is object confluent. First note that Σ is second-order as required. Then, by taking $\mathcal{R}_l = \{\lambda(x.\mathtt{t}(x))_{@}\mathtt{u} \longmapsto \mathtt{t}(\mathtt{u})\}$ and $\mathcal{R}_{nl} = \mathcal{R}_{\lambda\pi\uparrow} \setminus \mathcal{R}_l$, we have that $\mathcal{R}_l$ is confluent by orthogonality and $\mathcal{R}_{nl}$ is confluent by joining its critical pairs and seeing that it is also strongly normalizing. Therefore, both are also object confluent. Moreover, $\mathcal{R}_l$ is linear and there are no critical pairs between $\mathcal{R}_l$ and $\mathcal{R}_{nl}$. Finally, we can easily verify that there is no object term of sort lvl containing a subterm of sort $\mathtt{tm}$, and thus we have $\mathcal{R}_l \not\lesssim \mathsf{lvl}$. Hence, by our criterion $\mathcal{R}_{\lambda\pi\uparrow} = \mathcal{R}_l \cup \mathcal{R}_{nl}$ is object confluent.

We remark that one could for instance extend $\mathcal{R}_{\lambda\pi\uparrow}$ with other rules such as

$$\mathtt{n} \vee 0 \longmapsto \mathtt{n} \qquad\qquad \mathtt{n} + 0 \longmapsto \mathtt{n}$$
$$0 \vee \mathtt{m} \longmapsto \mathtt{m} \qquad\qquad \mathtt{n} + \mathsf{S}(\mathtt{m}) \longmapsto \mathsf{S}(\mathtt{n} + \mathtt{m})$$
$$\mathsf{S}(\mathtt{n}) \vee \mathsf{S}(\mathtt{m}) \longmapsto \mathsf{S}(\mathtt{n} \vee \mathtt{m})$$

where $\vee : (\mathtt{n} : \mathsf{lvl}, \mathtt{m} : \mathsf{lvl}) \to \mathsf{lvl}$ and $+ : (\mathtt{n} : \mathsf{lvl}, \mathtt{m} : \mathsf{lvl}) \to \mathsf{lvl}$, while preserving object confluence. Indeed, by placing them in $\mathcal{R}_{nl}$ one can redo the same reasoning as above. This is important because such kind of rules are sometimes also used when defining universes in type theory.

Final remarks Our theorem could instead be stated for second-order formalisms like [5, 4] — these correspond roughly to the second-order fragment of HRSs. There, the notion of object confluence is arguably more natural, as the restriction to 0 order variables is built in the formalism.

By instantiating $\mathcal{R}_{nl}$ with a left-linear system, condition (D) is verified trivially, and our criterion reduces to van Oostrom's result for orthogonal combinations, but restricted to the 2nd order case.

Finally, the criterion is designed for situations in which we can split the rewrite system into two parts: one that is s.n. but not left-linear (whose confluence can hopefully be shown with the critical pair lemma), and one that is left-linear but (possibly) not s.n. (whose confluence can hopefully be shown with criteria assuming left-linearity). Nevertheless, the condition (B) could also be replaced by a condition similar to (D), making symmetric the roles of the two systems in the theorem. However, we do not know of any interesting examples for which this generalization would apply.

Acknowledgments The author would like to thank Gilles Dowek for his careful reading of a preliminary version of the paper, Jean-Pierre Jouannaud, Gaspard Férey and Frédéric Blanqui for their thoughtful remarks on this work, and the anonymous reviewers for their very helpful comments and suggestions.

References

[1] Peter Aczel. A general Church–Rosser theorem. 1978.

[2] Ali Assaf, Gilles Dowek, Jean-Pierre Jouannaud, and Jiaxiang Liu. Untyped Confluence In Dependent Type Theories. working paper or preprint, April 2017.

[3] Gaspard Ferey. *Higher-Order Confluence and Universe Embedding in the Logical Framework*. These, Université Paris-Saclay, June 2021.

[4] Makoto Hamana. Universal algebra for termination of higher-order rewriting. In *Term Rewriting and Applications: 16th International Conference, RTA 2005, Nara, Japan, April 19-21, 2005. Proceedings 16*, pages 135–149. Springer, 2005.

[5] Makoto Hamana, Tatsuya Abe, and Kentaro Kikuchi. Polymorphic computation systems: Theory and practice of confluence with call-by-value. *Science of Computer Programming*, 187:102322, 2020.

[6] Robert Harper and Daniel R Licata. Mechanizing metatheory in a logical framework. *Journal of functional programming*, 17(4-5):613–673, 2007.

[7] Jan Willem Klop. *Combinatory reduction systems*. PhD thesis, Rijksuniversiteit Utrecht, 1963.

[8] Richard Mayr and Tobias Nipkow. Higher-order rewrite systems and their confluence. *Theoretical computer science*, 192(1):3–29, 1998.

[9] Vincent van Oostrom. Confluence for abstract and higher-order rewriting. *Ph. D. Thesis, Vrije Universiteit*, 1984.

[10] Vincent Van Oostrom. Developing developments. *Theoretical Computer Science*, 175(1):159–181, 1997.