Confluence of Layered Rewrite Systems

Jiaxiang Liu[1,2], Jean-Pierre Jouannaud[2], and Mizuhito Ogawa[3]

1 School of Software, and TNList, Tsinghua University, Beijing, China
2 Deducteam, INRIA, and LIX, École Polytechnique, Palaiseau, France
3 School of Information Science, JAIST, Ishikawa, Japan

Abstract

We investigate the new, Turing-complete class of *layered* systems, whose lefthand sides of rules can only be overlapped at a multiset of disjoint or equal positions. Layered systems define a natural notion of rank for terms: the maximal number of non-overlapping redexes along a path from the root to a leaf. Overlappings are allowed in finite or infinite trees. Rules may be non-terminating, non-left-linear, or non-right-linear. Using a novel unification technique, *cyclic unification*, we show that rank non-increasing layered systems are confluent provided their cyclic critical pairs have cyclic-joinable decreasing diagrams.

1998 ACM Subject Classification F.4.2 [Logic and computation] Rewriting

Keywords and phrases Layers, confluence, decreasing diagrams, critical pairs, cyclic unification

Digital Object Identifier 10.4230/LIPIcs.CSL.2015.423

1 Introduction

Confluence of terminating systems is well understood: it can be reduced to the joinability of local peaks by Newman's lemma, and to that of critical ones, obtained by unifying lefthand sides of rules at subterms, by Knuth-Bendix-Huet's lemma. Confluence can thus be decided by inspecting all critical pairs, see for example [5].

Many efforts notwithstanding [23, 12, 27, 20, 22, 19, 29, 10, 14, 24, 11, 30, 15, 25, 18, 1], confluence of non-terminating systems is far from being understood in terms of critical pairs. Only recently did this question make important progress with van Oostrom's complete method for checking confluence based on decreasing diagrams, a generalization of joinability [28, 29]. In particular, while Huet's result stated that linear systems are confluent provided their critical pairs are strongly confluent [12], Felgenhauer showed that right-linearity could be removed provided parallel critical pairs have decreasing diagrams [8]. Knuth-Bendix's and Felgenhauer's theorems can join forces in presence of both terminating and non-terminating rules [17].

We show here that rank non-increasing layered systems are confluent provided their critical pairs have decreasing diagrams. Our confluence result for non-terminating non-linear systems by critical pair analysis is the first we know of. Further, our result holds in case critical pairs become infinite, solving a long standing problem raised in [12]. Prior solutions to the problem existed under different assumptions that could be easily challenged [27, 10, 15].

Our results use a simplified version of sub-rewriting introduced in [17], and a simple, but essential revisitation of unification in case overlaps generate occur-check equations: *cyclic unification* is based on a new, important notion of cyclic unifiers, which enjoy all good properties of unifiers over finite trees such as existence of most general cyclic unifiers, and can therefore represent solutions of occur-check equations by simple rewriting means.

24th EACSL Annual Conference on Computer Science Logic (CSL 2015).
Editor: Stephan Kreutzer; pp. 423–440
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Terms are introduced in Section 2, labelled rewriting and decreasing diagrams in Section 3, sub-rewriting in Section 4, cyclic unification in Section 5 and layered systems in Section 6 where our main result is developed, before concluding in Section 7.

2 Terms, substitutions, and rewriting

Given a *signature* $\mathcal{F}$ of *function symbols* and a denumerable set $\mathcal{X}$ of *variables*, $\mathcal{T}(\mathcal{F},\mathcal{X})$ denotes the set of finite or infinite rational *terms* built up from $\mathcal{F}$ and $\mathcal{X}$. We reserve letters x, y, z for variables, f, g, h for function symbols, and s, t, u, v, w for terms. Terms are recognized by top-down tree automata in which some ω-states, and only those, are possibly traversed infinitely many times. Terms are identified with labelled trees. See [4] for details.

Positions are finite strings of positive integers. We use o, p, q for arbitrary positions, the empty string Λ for the root position, and "$\cdot$" for concatenation of positions or sets thereof. We use $\mathcal{FP}os(t)$ for the (possibly infinite) set of non-variable positions of t, $t(p)$ for the function symbol at position p in t, $t|_p$ for the *subterm* of t at position p, and $t[u]_p$ for the result of replacing $t|_p$ with u at position p in t. We may omit the position p, writing $t[u]$ for simplicity and calling $t[\cdot]$ a *context*. We use $\geq$ for the partial prefix order on positions (further from the root is bigger), $p\#q$ for incomparable positions p, q, called *disjoint*. The order on positions is extended to finite sets as follows: $P \geq Q$ (resp. $P > Q$) if $(\forall p \in P)(\exists q \in max(Q))\, p \geq q$ (resp. $p > q$), where $max(P)$ is the set of maximal positions in P. We use p for the set $\{p\}$.

We use $\mathcal{V}ar(t_1, \dots, t_n)$ for the set of variables occurring in $\{t_i\}_i$. A term t is *ground* if $\mathcal{V}ar(t) = \emptyset$, *linear* if no variable occurs more than once in t. Given a term t, we denote by $\bar{t}$ any linear term obtained by renaming, for each variable $x \in \mathcal{V}ar(t)$, the occurrences of x at positions $\{p_i\}_i$ in t by *linearized variable* x^{k_i} such that $i \neq j$ implies $x^{k_i} \neq x^{k_j}$. Note that $\mathcal{V}ar(\bar{s}) \cap \mathcal{V}ar(\bar{t}) = \emptyset$ iff $\mathcal{V}ar(s) \cap \mathcal{V}ar(t) = \emptyset$. Identifying x^{k_0} with x, $\bar{t} = t$ for a linear term t.

A *substitution* σ is an endomorphism from terms to terms defined by its value on its *domain* $\mathcal{D}om(\sigma) := \{x : \sigma(x) \neq x\}$. Its *range* is $\mathcal{R}an(\sigma) := \bigcup_{x \in \mathcal{D}om(\sigma)} \mathcal{V}ar(x\sigma)$. We use $\sigma_{|V}$ for the restriction of σ to $V \subseteq \mathcal{D}om(\sigma)$, and $\sigma_{|\neg X}$ for the restriction of σ to $\mathcal{D}om(\sigma) \setminus X$. The substitution σ is said to be *finite* (resp., a *variable substitution*) if for each $x \in \mathcal{D}om(\sigma)$, $\sigma(x)$ is a finite term (resp., a variable). Variable substitutions are called *renamings* when also bijective. A substitution γ is *ground* if for each $x \in \mathcal{X}$, $\gamma(x)$ is ground. We use Greek letters for substitutions and postfix notation for their application.

The strict *subsumption order* $\gtrdot$ on finite terms (resp. substitutions) associated with the quasi-order $s \mathrel{\underline{\gtrdot}} t$ (resp. $\sigma \mathrel{\underline{\gtrdot}} \tau$) iff $s = t\theta$ (resp. $\sigma = \tau\theta$) for some substitution θ, is well-founded.

A *rewrite rule* is a pair of finite terms, written $l \to r$, whose *lefthand side* l is not a variable and whose *righthand side* r satisfies $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l)$. A *rewrite system* R is a set of rewrite rules. A rewrite system R is *left-linear* (resp. *right-linear*, *linear*) if for every rule $l \to r \in R$, l is a linear term (resp. r is a linear term, l and r are linear terms). Given a rewrite system R, a term u *rewrites* to a term v *at* a position p, written $u \xrightarrow[R]{p} v$, if $u|_p = l\sigma$ and $v = u[r\sigma]_p$ for some rule $l \to r \in R$ and substitution σ. The term $l\sigma$ is a *redex* and $r\sigma$ its *reduct*. We may omit R as well as p, and also replace the former by the rule which is used and the latter by a property it satisfies, writing for example $u \xrightarrow[l \to r]{\geq P} v$. Rewriting *terminates* if there exists no infinite rewriting sequence issuing from some term. Rewriting is sometimes called *plain rewriting*.

Consider a *local peak* made of two rewrites issuing from the same term u, say $u \to^p_{l \to r} v$ and $u \to^q_{g \to d} w$. Following Huet [12], we distinguish three cases:

$p\#q$ (disjoint case), $q > p \cdot \mathcal{FP}os(l)$ (ancestor case), and $q \in p \cdot \mathcal{FP}os(l)$ (critical case). Given two, possibly different rules $l \to r, g \to d$ and a position $p \in \mathcal{FP}os(l)$ such that

$\mathcal{V}ar(l) \cap \mathcal{V}ar(g) = \emptyset$ and σ is a most general unifier of the equation $l|_p = g$, then $l\sigma$ is the *overlap* and $\langle r\sigma, l\sigma[d\sigma]_p\rangle$ the *critical pair* of $g \to d$ on $l \to r$ at p.

Rewriting extends naturally to lists of terms of the same length, hence to substitutions of the same domain. See [5, 26] for surveys.

3 Labelled rewriting and decreasing diagrams

Our goal is to reduce confluence of a non-terminating rewrite system R to that of finitely many critical pairs. Huet's analysis of linear non-terminating systems was based on Hindley's lemma, stating that a non-terminating relation is confluent provided its local peaks are joinable in at most one step from each side [12]. The more general analyses needed here have been made possible by van Oostrom's notion of decreasing diagrams for labelled relations.

▶ **Definition 1.** A *labelled rewrite relation* is a pair made of a rewrite relation $\to$ and a mapping from rewrite steps to a set of labels $\mathcal{L}$ equipped with a partial quasi-order $\trianglerighteq$ whose strict part $\triangleright$ is well-founded. We write $u \xrightarrow[R]{p,m} v$ for a rewrite step from u to v at position p with label m and rewrite system R. Indexes p, m, R may be omitted. We also write $\alpha \triangleright l$ (resp. $l \triangleright \alpha$) if $m \triangleright l$ (resp. $l \triangleright m$) for all m in a multiset α.

Given an arbitrary labelled rewrite step $\to^l$, we denote its projection on terms by $\to$, its inverse by ${}^l\!\leftarrow$, its reflexive closure by $\Rrightarrow^l$, its symmetric closure by $\overset{l}{\leftrightarrow}$, its reflexive, transitive closure by $\twoheadrightarrow^\alpha$ for some word α on the alphabet of labels, and its reflexive, symmetric, transitive closure, called *conversion*, by $\overset{\alpha}{\twoheadleftarrow\!\!\twoheadrightarrow}$. We may consider the word α as a multiset.

The triple v, u, w is said to be a *local peak* if $v \; {}^l\!\leftarrow u \to^m w$, a *peak* if $v \; {}^\alpha\!\twoheadleftarrow u \twoheadrightarrow^\beta w$, a *joinability diagram* if $v \twoheadrightarrow^\alpha u \; {}^\beta\!\twoheadleftarrow w$. The local peak $v \; {}^{p,m}_{l\to r}\!\leftarrow u \to^{q,n}_{g\to d} w$ is a *disjoint, critical, ancestor* peak if $p\#q, q \in p\cdot\mathcal{FP}os(l), q > p\cdot\mathcal{FP}os(l)$, respectively. The pair v, w is *convertible* if $v \overset{\alpha}{\twoheadleftarrow\!\!\twoheadrightarrow} w$, *divergent* if $v \; {}^\alpha\!\twoheadleftarrow u \twoheadrightarrow^\beta w$ for some u, and *joinable* if $v \twoheadrightarrow^\alpha t \; {}^\beta\!\twoheadleftarrow w$ for some t. The relation $\to$ is *locally confluent* (resp. *confluent*, *Church-Rosser*) if every local peak (resp. divergent pair, convertible pair) is joinable.

Given a labelled rewrite relation $\to^l$ on terms, we consider specific conversions associated with a given local peak called *local diagrams* and recall the important subclass of van Oostrom's decreasing diagrams and their main property: a relation all whose local diagrams are decreasing enjoys the Church-Rosser property, hence confluence. Decreasing diagrams were introduced in [28], where it is shown that they imply confluence, and further developed in [29]. The first version suffices for our needs.

▶ **Definition 2** (Local diagrams)**.** A *local diagram* D is a pair made of a *local peak* $D_{peak} = v \leftarrow u \to w$ and a *conversion* $D_{conv} = v \twoheadleftarrow\!\!\twoheadrightarrow w$. We call *diagram rewriting* the rewrite relation $\Rightarrow_{\mathcal{D}}$ on conversions associated with a set $\mathcal{D}$ of local diagrams, in which a local peak is replaced by one of its associated conversions:

$$P \; D_{peak} \; Q \underset{\mathcal{D}}{\Rightarrow} P \; D_{conv} \; Q \text{ for some } D \in \mathcal{D}$$

▶ **Definition 3** (Decreasing diagrams [28])**.** A local diagram D with peak $v \; {}^l\!\leftarrow u \to^m w$ is *decreasing* if its conversion $D_{conv} = v \overset{\alpha}{\twoheadrightarrow} s \overset{m}{\Rrightarrow} s' \overset{\delta}{\twoheadrightarrow} {}^{\delta'}\!\twoheadleftarrow t' \; {}^l\!\Lleftarrow t \; {}^\beta\!\twoheadleftarrow w$ satisfies the following *decreasingness condition*: labels in α (resp. β) are strictly smaller than l (resp. m), and labels in δ, δ' are strictly smaller than l or m. The rewrites $s \Rrightarrow^m s'$ and $t' \; {}^l\!\Lleftarrow t$ are called the *facing steps* of the diagram.

▶ **Theorem 4** ([14])**.** *The relation $\Rightarrow_{\mathcal{D}}$ terminates for any set $\mathcal{D}$ of decreasing diagrams.*

▶ **Corollary 5.** *Assume that $T \subseteq \mathcal{T}(\mathcal{F},\mathcal{X})$ and $\mathcal{D}$ is a set of decreasing diagrams in T such that the set of T-conversions is closed under $\Rightarrow_{\mathcal{D}}$. Then, the restriction of $\rightarrow$ to T is Church-Rosser if every local peak in T has a decreasing diagram in $\mathcal{D}$.*

This simple corollary of Theorem 4 is a reformulation of van Oostrom's decreasing diagram theorem which is convenient for our purpose.

4 Sub-rewriting

Consider the following famous system inspired by an abstract example of Newman, algebraized by Klop and publicized by Huet [12], NKH $= \{f(x,x) \rightarrow a,\ f(x,c(x)) \rightarrow b,\ g \rightarrow c(g)\}$. NKH is overlap-free, hence locally confluent by Huet's lemma [12]. However, it enjoys non-joinable non-local peaks such as $a \leftarrow f(g,g) \rightarrow f(g,c(g)) \rightarrow b$.

The main difficulty with NKH is that non-joinable peaks are non-local. To restore the usual situation for which the confluence of a relation can be characterized by the joinability of its local peaks, we need another rewrite relation whose local peaks capture the non-confluence of NKH as well as the confluence of its confluent variations. A major insight of [17] is that this can be achieved by the *sub-rewriting* relation, that allows us to rewrite $f(g,c(g))$ in one step to either a or b, therefore exhibiting the pair $\langle a,b\rangle$ as a sub-rewriting critical pair. Sub-rewriting is made of a preparatory *equalization* phase in which the variable instances of the lefthand side l of some rule $l \rightarrow r$ are joined, taking place before the rule is applied in the *firing phase*. In [17], sub-rewriting required a signature split to define layers in terms, the preparatory phase taking place in the lower layers. No a-priori layering is needed here:

▶ **Definition 6** (Sub-rewriting)**.** A term u *sub-rewrites* to a term v at $p \in \mathcal{P}os(u)$ for some rule $l \rightarrow r \in R$, written $u \rightarrow^{p}_{R_R} v$, if $u \twoheadrightarrow^{(>p\cdot\mathcal{FP}os(l))}_{R} u[l\theta]_p \rightarrow^{p}_{R} u[r\theta]_p = v$ for some substitution θ. The term $u|_p$ is called a *sub-rewriting redex*.

This definition of sub-rewriting allows *arbitrary rewriting* below the lefthand side of the rule until a redex is obtained. This is the major idea of sub-rewriting, ensuring that $R \subseteq R_R \subseteq R^*$. A simple, important property of a sub-rewriting redex is that it is an instance of a linearized lefthand side of rule:

▶ **Lemma 7** (Sub-rewriting redex)**.** *Assume u sub-rewrites to $u[r\sigma]_p$ with $l \rightarrow r$ at position p. Then $u|_p = \overline{l}\theta$ for some θ s.t. $(\forall x \in \mathcal{V}ar(l))\,(\forall p_i \in \mathcal{P}os(l)$ s.t. $l(p_i)=x)\,\theta(x^{p_i}) \twoheadrightarrow_R \sigma(x)$. We say that σ is an* equalizer *of l, and the rewrite steps from $\overline{l}\theta$ to $l\sigma$ are an* equalization.

Sub-rewriting differs from rewriting modulo by being directional. It differs from Klop's higher-order rewriting modulo developments [26] used by Okui for first-order computations [22], in that the preparatory phase uses arbitrary rewriting. Having non-left-linear rules with critical pairs at subterms seems incompatible with using developments. Sub-rewriting differs as well from relative rewriting [11] in that the preparatory phase must take place below variables. The latter condition is essential to obtain plain critical pairs based on plain unification.

Assuming that local sub-rewriting peaks characterize the confluence of NKH, we need to compute the corresponding critical pairs. Unifying the lefthand sides $f(x,x)$ and $f(y,c(y))$ results in the conjunction $x = y \wedge y = c(y)$ containing the *occur-check equation* $y = c(y)$, which prevents unification from succeeding on finite trees but allows it to succeed on infinite rational trees: the critical peak has therefore an infinite overlap $f(c^\omega,c^\omega)$ and a finite critical pair $\langle a,b\rangle$. At the level of infinite trees, we then have an infinite local rewriting peak

Remove	$s = s \wedge P$	$\to$	P	
Decomp	$f(\vec{s}) = f(\vec{t}) \wedge P$	$\to$	$\vec{s} = \vec{t} \wedge P$	
Conflict	$f(\vec{s}) = g(\vec{t}) \wedge P$	$\to$	$\bot$	**if** $f \neq g$
Choose	$y = x \wedge P$	$\to$	$x = y \wedge P$	**if** $x \notin Var(P)$, $y \in \mathcal{V}ar(P)$
Coalesce	$x = y \wedge P$	$\to$	$x = y \wedge P\{x \mapsto y\}$	**if** $x, y \in Var(P)$, $x \neq y$
Swap	$u = x \wedge P$	$\to$	$x = u \wedge P$	**if** $u \notin \mathcal{X}$
Merge	$x = s \wedge x = t \wedge P$	$\to$	$x = s \wedge s = t \wedge P$	**if** $x \in \mathcal{X}$, $0 < \lvert s\rvert \leq \lvert t\rvert$
Replace	$x = s \wedge P$	$\to$	$x = s \wedge P\{x \mapsto s\}$	**if** $x \in Var(P)$, $x \notin \mathcal{V}ar(s)$, $s \notin \mathcal{X}$
Merep	$y = x \wedge x = s \wedge P$	$\to$	$y = s \wedge x = s \wedge P$	**if** $x \in Var(s)$, $s \notin \mathcal{X}$, $y \notin \mathcal{V}ar(s, P)$ and no other rule applies

Figure 1 Unification Rules.

$a \leftarrow f(c^\omega, c^\omega) = f(c^\omega, c(c^\omega)) \to b$, the properties of infinite trees making the sub-rewriting preparatory phase useless. Sub-rewriting therefore captures on finite trees some properties of rewriting on infinite trees, here the existence of a local peak. Computing the critical pairs of the sub-rewriting relation is therefore related to unification over finite trees resulting possibly in solutions over infinite rational trees. In the next section, we develop a novel view of unification that will allow us to capture both finite and infinite overlaps by finite means.

5 Cyclic unification

This section is adapted from [3, 5, 13] by treating finite and infinite unifiers uniformly: equality of terms is interpreted over the set of infinite rational terms *when needed*.

An *equation* is an oriented pair of finite terms, written $u = v$. A *unification problem* P is a (finite) conjunction $\wedge_i\, u_i = v_i$ of equations, sometimes seen as a multiset of pairs written $\vec{u} = \vec{v}$. A *unifier* (resp. a *solution*) of P is a substitution (resp., a ground substitution) θ such that $(\forall i)\, u_i\theta = v_i\theta$. A unifier describes a generally infinite set of solutions via its ground instances. A major usual assumption, ensuring that solutions exist when unifiers do, is that the set $\mathcal{T}(\mathcal{F})$ of ground terms is non-empty. A unification problem P has a *most general finite unifier* $mgu(P)$, whenever a finite solution exists, which is minimal with respect to subsumption and unique up to variable renaming. Computing $mgu(P)$ can be done by the unifier-preserving transformations of Figure 1, starting with P until a *solved form* is obtained, $\bot$ denoting the absence of solution, whether finite or infinite. Our notion of solved form therefore allows for infinite unifiers (and solutions) as well as finite ones:

▶ **Definition 8.** *Solved forms* of a unification problem P different from $\bot$ are unification problems $S = \vec{x} = \vec{u} \wedge \vec{y} = \vec{v}$ such that
(i) $\mathcal{P} = \mathcal{V}ar(P) \setminus (\vec{x} \cup \vec{y})$ is the set of *parameters* of S;
(ii) variables in $\vec{x} \cup \vec{y}$ (i.e. variables at lefthand sides of equations) are all distinct;
(iii) $(\forall\, x = u \in \vec{x} = \vec{u})$, $\mathcal{V}ar(u) \subseteq \mathcal{P}$;
(iv) $(\forall\, y = v \in \vec{y} = \vec{v})$, $\mathcal{V}ar(v) \subseteq \mathcal{P} \cup \vec{y}$, $\mathcal{V}ar(v) \cap \vec{y} \neq \emptyset$ and $v \notin \mathcal{X}$.

Equations $y = v \in \vec{y} = \vec{v}$ are called *cyclic* (or *occur-check*, the vocabulary originating from [3] used so far), $\vec{x}$ is the set of *finite* variables, and $\vec{y}$ is the set of (infinite) *cyclic* (or *occur-check*) variables. A solved form is a set of equations since $\vec{x} \cup \vec{y}$ is itself a set and an equation $x = y$ between variables can only relate a finite variable x with a parameter y.

▶ **Example 9** (NKH). $f(x,x) = f(y,c(y)) \to_{\mathbf{Decomp}} x = y \wedge x = c(y) \to_{\mathbf{Coalesce}} x = y \wedge y = c(y) \to_{\mathbf{Merep}} x = c(y) \wedge y = c(y)$. Alternatively, $f(x,x) = f(y,c(y)) \to_{\mathbf{Decomp}} x = y \wedge x = c(y) \to_{\mathbf{Replace}} c(y) = y \wedge x = c(y) \to_{\mathbf{Swap}} y = c(y) \wedge x = c(y)$.

Choose and **Swap** originate from [3]. **Replace** and **Coalesce** ensure that finite variables (but parameters) do not occur in equations constraining the infinite ones. **Merep** is a sort of combination of **Merge** and **Replace** ensuring condition $v \notin \mathcal{X}$ in Definition 8, item (iv). Unification over finite trees has another failure rule, called **Occur-check**, fired in presence of cyclic equations.

▶ **Theorem 10.** *Given an input unification problem P, the unification rules terminate, fail if the input has no solution, and return a solved form $S = \vec{x} = \vec{u} \wedge \vec{y} = \vec{v}$ otherwise.*

Proof. Termination, characterization of solved forms, soundness, are all adapted from [13].

Termination. The quadruple $\langle nu, |P|, nvre, nvle \rangle$ is used to interpret a unification problem P, where

- nu is the number of unsolved variables (0 for $\perp$), where a variable x is *solved* in $x = s \wedge P'$ if $x \notin \mathcal{V}ar(s, P')$;
- $|P|$ is the multiset ($\emptyset$ for $\perp$) of natural numbers $\{max(|s|, |t|) : s = t \in P\}$;
- $nvle$ (resp. $nvre$) is the number of equations in P whose lefthand (resp., righthand) side is a variable and the other side is not.

Remove, **Decomp** and **Conflict** decrease $|P|$ without increasing nu. **Choose** and **Coalesce** both decrease nu. **Swap** decreases $nvre$ without increasing nu and $|P|$. **Merge** decreases $nvle$ without increasing nu, $|P|$ and $nvre$. **Replace** decreases nu. Now, when **Merep** applies, no other rule can apply, and we can check that no rules can apply either after **Merep** (except another possible application of **Merep**). This can happen only finitely many times, by simply reasoning on the number of equations whose both sides are variables.

Solved form. We show by contradiction that the output P, which is in normal form with respect to the unification rules, is a solved form in case **Conflict** never applies. First, P must be a conjunction of equations $x = s$, since otherwise **Decomp** or **Swap** would apply. Let $\mathcal{P} = \mathcal{V}ar(P) \setminus (\vec{x} \cup \vec{y})$.

- Condition (i) is a definition.
- Condition (ii). Let $P = x = s \wedge x = t \wedge P'$. Either s or t is a variable, since otherwise **Merge** would apply. Assume without loss of generality that $s \in \mathcal{X}$, call it y. If $x = y$, **Remove** applies. If $y \notin \mathcal{V}ar(t, P')$, then **Choose** applies. Otherwise, **Coalesce** applies. Hence $\vec{x}, \vec{y}$ are all different *sets*, and P is therefore itself a set.
  Let now $\vec{x} = \vec{u}$ be a maximal (with respect to inclusion) set of equations in P such that $\mathcal{V}ar(\vec{u}) \subseteq \mathcal{P}$, and $\vec{y} = \vec{v}$ be the remaining set of equations.
- Condition (iii). It is ensured by the definition of $\vec{x} = \vec{u}$.
- Condition (iv). Let $y = v \in \vec{y} = \vec{v}$.
  Let now $x = u \in \vec{x} = \vec{u}$, hence $x \notin \mathcal{V}ar(u)$. Assume $x \in \mathcal{V}ar(v)$. If $u \notin \mathcal{X}$, then **Replace** applies. Otherwise, if u has no other occurrence in P, then **Choose** applies, else **Coalesce** applies. Therefore $\mathcal{V}ar(v) \cap \vec{x} = \emptyset$ by contradiction.
  Assume $\mathcal{V}ar(v) \cap \vec{y} = \emptyset$. Then $\mathcal{V}ar(v) \subseteq \mathcal{P}$, which contradicts the maximality of $\vec{x} = \vec{u}$.
  We are left to show that v is not a variable. If it were, then $v \in \vec{y}$. First, $v \neq y$, otherwise **Remove** applies. Let $P = (y = v) \wedge P'$ with $v \in \vec{y} \setminus \{y\}$. Let $v = z$, there must exist $(z = w) \in P'$ for some w, otherwise $z \in \mathcal{P}$. Hence $P' = (z = w) \wedge P''$. Now, $y \notin \mathcal{V}ar(w, P'')$, otherwise **Coalesce** applies. Then we show $z \in \mathcal{V}ar(w)$: firstly, $w \neq z$,

otherwise **Remove** applies; secondly, w is not a variable, otherwise $w \notin \mathcal{V}ar(y, P'')$ lets **Choose** apply, while $w \in \mathcal{V}ar(y, P'')$ makes **Coalesce** available; then if $z \notin \mathcal{V}ar(w)$, **Replace** applies. Thus $z \in \mathcal{V}ar(w)$, allowing **Merep**, which contradicts that we have indeed a solved form.

Soundness. The set of solutions is an invariant of the unification rules. This is trivial for all rules but **Coalesce**, **Merge**, **Replace**, **Merep**, for which it follows from the fact that substitutions are homomorphisms and equality is a congruence. ◀

The solved form is a *tree solved form* if $\vec{y} = \emptyset$, and otherwise an Ω *solved form* whose solutions are infinite substitutions taking their values in the set of infinite (rational) terms. We shall now develop our notion of *cyclic unifier* capturing both solved forms by describing the infinite unifiers of a problem P as a pair of a finite unifier σ and a set of cyclic equations E constraining those variables that require infinite solutions. In case $E = \emptyset$, then P is a tree solved form and $\sigma = mgu(P)$. To avoid manipulating infinite unifiers when $E \neq \emptyset$, we shall work with the cyclic equations themselves considered as a ground rewrite system.

▶ **Definition 11** ([21]). Given a set of equations E, we denote by $=^{cc}_E$ the equational theory in which the variables in $\mathcal{V}ar(E)$ are treated as constants, also called *congruence closure* E.

We are interested in the congruence closure defined by cyclic equations, seen here as a set R of ground rewrite rules. We may sometimes consider R as a set of equations, to be either solved or used as axioms, depending on context.

▶ **Definition 12.** A *cyclic rewrite system* is a set of rules $R = \{\vec{y} \to \vec{v}\}$ such that the unification problem $\vec{y} = \vec{v}$ is its own solved form with $\vec{y}$ as the set of infinite cyclic variables. Variables in R are treated as constants.

▶ **Lemma 13.** *A cyclic rewrite system R is ground and critical pair free, hence Church-Rosser.*

We now introduce our definition of cyclic unifiers and solutions:

▶ **Definition 14.** A *cyclic unifier* of a unification problem P is a pair $\langle \eta, R\rangle$ made of a substitution η and a cyclic rewrite system $R = \{\vec{y} \to \vec{v}\}$, satisfying:
(i)) $\mathcal{D}om(\eta) \subseteq \mathcal{V}ar(P) \setminus \vec{y}$, $\mathcal{R}an(\eta) \cap \vec{y} = \emptyset$, and $\mathcal{R}an(\eta) \cap \mathcal{D}om(\eta) = \emptyset$;
(ii) P and $P \wedge R$ have identical sets of solutions ; and
(iii) **(a)** $(\forall u = v \in P)\, u\eta =^{cc}_{R\eta} v\eta$, or equivalently by Lemma 13,
(b) $(\forall u = v \in P)\, u\eta \twoheadrightarrow_{R\eta}\ {}_{R\eta}\!\twoheadleftarrow v\eta$.
A *cyclic solution* of P is a pair $\langle \eta\rho, R\rangle$ made of a cyclic unifier $\langle \eta, R\rangle$ of P and an additional substitution ρ.

We shall use (iii)(a) or (iii)(b) indifferently, depending on our needs, by referring to (iii).

The idea of cyclic unifiers is that the need for infinite values for some variables is encoded via the use of the cyclic rewrite system R, which allows us to solve the various occur-check equations generated when unifying P. Finite variables are instantiated by the finite substitution η, which ensures that cyclic unification reduces to finite unification in the absence of infinite variables. The technical restrictions on $\mathcal{D}om(\eta)$ and $\mathcal{R}an(\eta)$ aim at making η idempotent. In (iii), parameters occurring in R are instantiated by η before rewriting takes place: cyclic unification is nothing but rigid unification modulo the cyclic equations in R [9]. Instantiation of the infinite variables $\vec{y}$ is delegated to cyclic solutions via the additional substitution ρ which may also instantiate the variables introduced by η.

▶ **Example 15.** Consider the equation $f(x,z,z) = f(a,y,c(y))$. A cyclic unifier is $\langle\{x \mapsto a\}, \{y \to c(z), z \to c(z)\}\rangle$, and a cyclic solution is $\langle\{x \mapsto a, y \mapsto a, z \mapsto c(a)\}, \{y \to c(z), z \to c(z)\}\rangle$, which is clearly an instance of the former by the substitution $\{y \mapsto a, z \mapsto c(a)\}$. For the former, $f(a,z,z) =^{cc}_{\{y=c(z),z=c(z)\}} f(a,y,c(y))$. Another cyclic unifier is $\langle\{x \mapsto a\}, \{z \to c(y), y \to c(y)\}\rangle$, for which $f(a,z,z) =^{cc}_{\{z=c(y),y=c(y)\}} f(a,y,c(y))$.

The set of cyclic unifiers of a problem P is closed under substitution instance, provided the variable conditions on its substitution part are met, as is the set of its unifiers. Cyclic unifiers have indeed many interesting properties similar to those of finite unifiers, of which we are going to investigate only a few which are relevant to the confluence of layered systems.

We now focus our attention on specific cyclic unifiers sharing a same cyclic rewrite system.

▶ **Definition 16.** Given a unification problem P with solved form $S = \vec{x} = \vec{u} \wedge \vec{y} = \vec{v}$, let

- its set of parameters $\mathcal{P} = \mathcal{V}ar(P) \setminus (\vec{x} \cup \vec{y})$,
- its cyclic rewrite system $R_S = \{\vec{y} \to \vec{v}\}$ and *canonical* substitution $\eta_S = \{\vec{x} \mapsto \vec{u}\}$,
- its S-based cyclic unifiers $\langle\eta, R_S\rangle$, among which $\langle\eta_S, R_S\rangle$ is said to be *canonical*.

We now show a major property of S-based cyclic unifiers, true for any solved form S:

▶ **Lemma 17.** *Given a unification problem with solved form S, the set of S-based cyclic unifiers is preserved by the unification rules.*

Proof. The result is straightforward for **Remove**, **Choose**, and **Swap**. It is true for **Decomp** and **Conflict** since, using formulation (iii.b) of Definition 14, the rules in $R\eta$ cannot apply at the root of $\mathcal{F}$-headed terms. Next comes **Coalesce**. We need to prove that $\langle\eta, R\rangle$ is a cyclic unifier for $x = y \wedge P$ iff it is one for $x = y \wedge P\{x \mapsto y\}$. Let $u = v \in P$. For the only if case, we have $u\{x \mapsto y\}\eta = u\eta\{x\eta \mapsto y\eta\} =^{cc}_{R\eta} u\eta =^{cc}_{R\eta} v\eta =^{cc}_{R\eta} v\eta\{x\eta \mapsto y\eta\} = v\{x \mapsto y\}\eta$. The if case is similar. **Replace** is similar to **Coalesce**. Consider now **Merge** (**Merep** is similar). Showing that $\langle\eta, R\rangle$ is a cyclic unifier for $x = s \wedge x = t \wedge P$ iff it is one for $x = s \wedge s = t \wedge P$ is routine by using transitivity of the congruence closure $=^{cc}_{R\eta}$. ◀

We can now conclude:

▶ **Theorem 18.** *Given a unification problem P with solved form $S = \vec{x} = \vec{u} \wedge \vec{y} = \vec{v}$, the canonical S-based cyclic unifier is most general among the set of S-based cyclic unifiers of P.*

Proof. Let $\langle\eta, R_S\rangle$ be a cyclic unifier of P based on S.

Let $x = u \in \vec{x} = \vec{u}$. By definition of cyclic unification, $x\eta \twoheadrightarrow_{R_S\eta} {}_{R_S\eta}\twoheadleftarrow u\eta$. By definition of a solved form and cyclic unifiers, we have: $\mathcal{V}ar(x\eta, u\eta) \subseteq (\vec{x} \cup \mathcal{P} \cup \mathcal{R}an(\eta))$, $(\vec{x} \cup \mathcal{P}) \cap \vec{y} = \emptyset$, $\mathcal{R}an(\eta) \cap \vec{y} = \emptyset$, and $\vec{y} \cap \mathcal{D}om(\eta) = \emptyset$. Therefore, $x\eta$ and $u\eta$ are irreducible by $R_S\eta$. Hence $x\eta = u\eta$. Since $x\eta_S = u$, it follows that $x\eta = u\eta = (x\eta_S)\eta = x(\eta_S\eta)$.

Let now $z \in \mathcal{V}ar(P) \setminus \vec{x}$. Since $z \notin \mathcal{D}om(\eta_S)$, then $\eta(z) = z\eta = (z\eta_S)\eta = z(\eta_S\eta)$.

Therefore, $\eta = \eta_S\eta$ and we are done. ◀

This result, which suffices for our needs, is easily lifted to cyclic solutions, as they are instances of a cyclic unifier. We can further prove that η_S is more general than any S'-based cyclic unifiers, for any solved form S' of P. This is where our conditions on $\mathcal{R}an(\eta)$ become important. We conjecture that it is most general among the set of all cyclic unifiers.

6 Layered systems

NKH is non-confluent, but can be easily made confluent by adding the rule $a \to b$ (giving NKH^1), or removing the rule $g \to c(g)$ (giving NKH^2). It can be made non-right-ground by making the symbols a, b unary (using $a(c(x))$ and $b(c(x))$ in the righthand sides of rules, giving NKH^3), or even non-right-linear by making them binary (giving NKH^4). There are classes of systems containing NKH for which it is possible to conclude its non-confluence. The following classes succeed for NKH^1: simple-right-linear [27], strongly depth-preserving [10], and relatively terminating [15]. As for NKH^3, it is neither simple-right-linear nor strongly depth-preserving: only [15] can cover it. When it comes to NKH^4, relative termination becomes hard to satisfy in presence of non-right-linearity [15].

Our goal is to define a robust, Turing-complete class of rewrite systems capturing NKH and its variations, for which confluence can be analyzed in terms of critical diagrams.

▶ **Definition 19.** A rewrite system R is *layered* iff it satisfies the *disjointness* assumption (DLO) that *linearized* overlaps of some lefthand sides of rules upon a given lefthand side l can only take place at a multiset of disjoint or equal positions of $\mathcal{FP}os(l)$:

$$\begin{array}{rcl}
(\mathrm{DLO}) & := & (\forall l \to r \in R)\,(\forall p \in \mathcal{FP}os(l))\,(\forall g \to d \in R \text{ s.t. } \mathcal{V}ar(\bar{l}) \cap \mathcal{V}ar(\bar{g}) = \emptyset) \\
 & & (\forall \sigma : \mathcal{V}ar(\bar{l}|_p, \bar{g}) \to \mathcal{T}(\mathcal{F},\mathcal{X}) \text{ s.t. } \bar{l}|_p\sigma = \bar{g}\sigma)\,\mathrm{SOF}(l|_p) \wedge \mathrm{SOF}(g) \\
\mathrm{SOF}(u) & := & (\forall q \in \mathcal{FP}os(u)\backslash\{\Lambda\})\,\mathrm{OF}(u|_q) \\
\mathrm{OF}(v) & := & (\forall g \to d \in R \text{ s.t. } \mathcal{V}ar(\bar{v}) \cap \mathcal{V}ar(\bar{g}) = \emptyset) \\
 & & (\forall o \in \mathcal{FP}os(v))(\forall \sigma : \mathcal{V}ar(\bar{v}, \bar{g}) \to \mathcal{T}(\mathcal{F},\mathcal{X}))\ \bar{v}|_o\sigma \neq \bar{g}\sigma
\end{array}$$

SOF stands for *subterm overlap-free*, and OF for *overlap-free*. In words, if two lefthand sides of rules in R overlap (linearly) a lefthand side l of a rule in R at positions p and q respectively, then either $p = q$ or $p\#q$. Overlaps at different positions along a path from the root to a leaf of l are forbidden.

Layered systems is a decidable class that relates to overlay systems [6], for which overlaps computed with plain unification can only take place at the root of terms –hence their name–, and generalizes strongly non-overlapping systems [24] which admit no linearized overlaps at all. All these classes are Turing-complete since they contain a complete class [16].

▶ **Example 20.** NKH is a layered system, which is also overlay. $\{h(f(x, y)) \to a, f(x, c(x)) \to b\}$ is layered but not overlay. $\{h(f(x, x)) \to a, f(x, c(x)) \to b, g \to c(g)\}$ is layered, but not strongly non-overlapping. $\{f(h(x)) \to x, h(a) \to a, a \to b\}$ is not overlay nor layered: SOF$(h(x))$ succeeds while SOF$(h(a))$ fails, hence their conjunction fails.

6.1 Layering

We define the rank of a term t as the maximum number of non-overlapping linearized redexes traversed from the root to some leaf of t, which differs from the usual redex-depth.

▶ **Definition 21.** Given a layered rewrite system R, the *rank* $rk(t)$ of a term t is defined by induction on the size of terms as follows:

- the maximal rank of its immediate subterms if t is not a linearized redex; otherwise,
- 1 plus $max\{rk(\sigma) : (\exists l \to r \in R)\, t = \bar{l}\sigma\}$, where $rk(\sigma) := max\{rk(\sigma(x)) : x \in \mathcal{V}ar(\bar{l})\}$.

▶ **Definition 22.** A rewrite system R is *rank non-increasing* if for all terms u, v such that $u \xrightarrow[R]{} v$, then $rk(u) \geq rk(v)$.

The rewrite system $\{f(x) \to c(f(x))\}$ is rank non-increasing while $\{f(x) \to f(f(x))\}$ is rank increasing. The system $\{fib(0) \to 0, fib(S(0)) \to S(0), fib(S(S(x))) \to fib(S(x)) + fib(x)\}$ calculating the Fibonacci function is rank non-increasing. NKH is rank non-increasing. The coming decidable sufficient condition for rank non-increasingness captures our examples (but Fibonacci, for which an even more complex decidable property is needed):

▶ **Lemma 23.** *A layered rewrite system R is rank non-increasing if each rule $g \to d$ in R satisfies the following properties:*

(i) $((\forall l \to r \in R)(\forall l' \to r' \in R)$ *s.t.* $\mathcal{V}ar(d), \mathcal{V}ar(\bar{l}), \mathcal{V}ar(\overline{l'})$ *are pairwise disjoint*$)$
$(\forall p, q \in \mathcal{FP}os(d)$ *s.t.* $q > p \cdot \mathcal{FP}os(l))$
$(\forall \sigma : \mathcal{V}ar(g, \bar{l}, \overline{l'}) \to \mathcal{T}(\mathcal{F}))\ (d|_p\sigma \neq \bar{l}\sigma) \vee (d|_q\sigma \neq \overline{l'}\sigma)$ *;*

(ii) $((\forall l \to r \in R)$ *s.t.* $\mathcal{V}ar(g) \cap \mathcal{V}ar(\bar{l}) = \emptyset)(\forall p \in \mathcal{FP}os(l) \setminus \Lambda)$
$(\forall \sigma : \mathcal{V}ar(g, \bar{l}) \to \mathcal{T}(\mathcal{F})$ *s.t.* $d\sigma = \bar{l}|_p\sigma)$
$((\exists l' \to r' \in R)$ *s.t.* $\mathcal{V}ar(\overline{l'}) \cap \mathcal{V}ar(g, \bar{l}) = \emptyset)(\exists x \notin \mathcal{V}ar(\bar{l}, \overline{l'}, g))\ \bar{l}[x]_p \gtrdot \overline{l'}$*.*

We can now index term-related notions by the rank of terms. Let $\mathcal{T}_n(\mathcal{F}, \mathcal{X})$ (in short, $\mathcal{T}_n$) be the set of terms of rank at most n. Two terms in $\mathcal{T}_n$ are *n-convertible* (resp. *n-joinable*) if their R-conversion (resp. R-joinability) involves terms in $\mathcal{T}_n$ only.

6.2 Closure properties

Call a term u an *OF-term* if u satisfies OF(u), and a substitution an *OF-substitution* if it maps variables to OF-terms. OF-terms enjoy several important closure properties. Given two substitutions θ, σ and rank n, let

$\mathrm{Conv}_n^\theta(\overline{u}, \overline{v})$ iff $\overline{u}\theta$ and $\overline{v}\theta$ are n-convertible, and
$\mathrm{Equalize}_n(\overline{u})_\sigma^\theta$ iff $\overline{u}\theta \twoheadrightarrow_{R_R} u\sigma$ with terms of rank at most n.

▶ **Lemma 24.** *For all OF-terms u and substitutions γ, $u\gamma$ cannot sub-rewrite at a position $p \in \mathcal{FP}os(u)$.*

▶ **Corollary 25.** *OF-terms are preserved under instantiation by OF-substitutions.*

▶ **Lemma 26.** *Let u, v be two terms such that $\mathrm{Conv}_n^\theta(\overline{u}, \overline{v})$, $\mathrm{Equalize}_n(\overline{u})_\sigma^\theta$ and $\mathrm{Equalize}_n(\overline{v})_\sigma^\theta$. Then $u\sigma$ and $v\sigma$ are n-convertible.*

▶ **Lemma 27.** *Let $\wedge_i u_i = v_i$ be obtained by decomposition of a unification problem P. Assume all equations $u_i = v_i$ satisfy the properties $\mathrm{Conv}_n^\theta(\overline{u_i}, \overline{v_i})$, $\mathrm{Equalize}_n(\overline{u_i})_\sigma^\theta$, $\mathrm{Equalize}_n(\overline{v_i})_\sigma^\theta$, OF$(u_i)$ and OF(v_i). Assume further that n-convertible terms are joinable. Then, unification of P succeeds, and returns a solved form whose all equations satisfy these five properties.*

In this lemma and coming proof, we assume that linearizations are propagated by the unification rules, implying in particular that $\overline{u|_p} = \overline{u}|_p$. P defines the initial linearization.

Proof. We show that these five properties are invariant by the unification rules. The claim follows since the unification rules terminate. We use notations of Figure 1.

- **Remove**, **Choose**, **Swap** are straightforward.
- **Decomp**. By assumption, $\mathrm{Conv}_n^\theta(\overline{f(\vec{s})}, \overline{f(\vec{t})})$, hence $\overline{f(\vec{s})}\theta$ and $\overline{f(\vec{t})}\theta$ are joinable by using terms of rank at most n, since R is rank non-increasing. By assumption OF$(f(\vec{s}))$ and OF$(f(\vec{t}))$, hence no rewrite can take place at the root. The result follows.
- **Conflict**. By the same token, $f = g$, a contradiction. Thus **Conflict** is impossible.

- **Coalesce**. By assumption, $\text{Conv}_n^\theta(x^k,y^l)$, $\text{Equalize}_n(x^k)_\sigma^\theta$, $\text{Equalize}_n(y^l)_\sigma^\theta$, and $(\forall u = v \in P)$, $\text{Conv}_n^\theta(\overline{u},\overline{v})$, $\text{OF}(u)$, $\text{Equalize}_n(\overline{u})_\sigma^\theta$, $\text{OF}(v)$ and $\text{Equalize}_n(\overline{v})_\sigma^\theta$. Putting these things together, we get $\text{Conv}_n^\theta(\overline{u}\{x^k \mapsto y^l\},\overline{v}\{x^k \mapsto y^l\})$, hence $\text{Conv}_n^\theta(\overline{u\{x \mapsto y\}},\overline{v\{x \mapsto y\}})$. Similarly, properties $\text{Equalize}_n(\overline{u\{x \mapsto y\}})_\sigma^\theta$ and $\text{Equalize}_n(\overline{v\{x \mapsto y\}})_\sigma^\theta$ hold. Property $\text{OF}(u)$ is of course preserved by variable renaming for any u.
- **Merge**. Assume $\text{Conv}_n^\theta(x^k,\overline{s})$, $\text{Conv}_n^\theta(x^l,\overline{t})$, $\text{OF}(s)$, $\text{Equalize}_n(\overline{s})_\sigma^\theta$, $\text{Equalize}_n(x^k)_\sigma^\theta$, $\text{OF}(t)$, $\text{Equalize}_n(\overline{t})_\sigma^\theta$ and $\text{Equalize}_n(x^l)_\sigma^\theta$. $\text{Conv}_n^\theta(\overline{s},\overline{t})$ follows from $\text{Conv}_n^\theta(x^k,\overline{s})$, $\text{Conv}_n^\theta(x^l,\overline{t})$, $\text{Equalize}_n(x^k)_\sigma^\theta$ and $\text{Equalize}_n(x^l)_\sigma^\theta$. The other properties follow similarly.
- **Replace**. The proof is similar for the first 3 properties. Further, OF is preserved by replacement by Corollary 25.
- **Merep**. Similar to **Merge**. ◀

▶ **Example 28** (NKH). Let $P = f(x,x) = f(y,c(y))$. Then $P \rightarrow_{\textbf{Decomp}} x = y \wedge x = c(y) \rightarrow_{\textbf{Replace}} c(y) = y \wedge x = c(y) \rightarrow_{\textbf{Swap}} y = c(y) \wedge x = c(y)$. Successive linearizations yield $f(x^1,x^2) = f(y^1,c(y^2))$, $x^1 = y^1 \wedge x^2 = c(y^2)$, $c(y^2) = y^1 \wedge x^2 = c(y^2)$ and $y^1 = c(y^2) \wedge x^2 = c(y^2)$. The announced properties of the solved form can be easily verified.

▶ **Corollary 29.** *Let $l \to r, g \to d \in R$ and $p \in \mathcal{FP}os(l)$ such that $\mathcal{V}ar(l) \cap \mathcal{V}ar(g) = \emptyset$, and $\overline{l}|_p\theta = \overline{g}\theta$ are terms in $\mathcal{T}_{n+1}$. Then, unification of $l|_p = g$ succeeds, returning a solved form S s.t., for each $z = s \in S$, $\text{Conv}_n^\theta(\overline{z},\overline{s})$, $\text{OF}(s)$, $\text{Equalize}_n(\overline{s})_\sigma^\theta$ for all σ satisfying $(\overline{l}\theta \twoheadrightarrow^{(>\mathcal{FP}os(l))} l\sigma) \wedge (\overline{g}\theta \twoheadrightarrow^{(>\mathcal{FP}os(g))} g\sigma)$, and further, $\text{SOF}(l|_p\eta_S) \wedge \text{SOF}(g\eta_S)$.*

Proof. Unification applies first **Decomp**. Conclude by Lemmas 27 and Corollary 25. ◀

▶ **Corollary 30.** *Assume $t = \overline{l}\sigma$ for some $l \to r \in R$. Then, $rk(t) = 1 + rk(\sigma)$.*

Proof. Let $t = \overline{l_i}\sigma_i = \overline{l_i}\theta\gamma$ (note that γ does not depend on i), where $\theta = mgu(=_i \overline{l_i})$. Then, $rk(t) = 1 + max_i\{rk(\sigma_i)\} = 1 + max_i\{rk(\theta\gamma)\} = 1 + rk(\gamma) = 1 + rk(\sigma_i)$ since θ satisfies OF at all non-variable positions by Lemma 27. ◀

▶ **Example 31** (NKH). Consider $f(c(g),c(g))$ of rank 2, using either linearized lefthand side $f(x^1,x^2)$ or $f(y^1,c(y^2))$ to match $f(c(g),c(g))$. Corresponding substitutions have rank 1.

A major consequence is that the preparatory phase of sub-rewriting operates on terms of a strictly smaller rank. This would not be true anymore, of course, with a conversion-based preparatory phase. More generally, we can also show that the rank of terms does not increase –but may remain stable– when taking a subterm, a property which is not true of non-layered systems. Consider the system $\{f(g(h(x))) \to x, g(x) \to x, h(x) \to x\}$. The redex $f(g(h(a)))$ has rank 1 with our definition, but its subterm $g(h(a))$ has rank 2.

6.3 Testing confluence of layered systems via their cyclic critical pairs

Since R is rank non-increasing we shall prove confluence by induction on the rank of terms. Since rewriting is rank non-increasing, the set of $\mathcal{T}_n$-conversions is closed under diagram rewriting, hence allowing us to use Corollary 5. This is why we adopted this restricted, but complete, form of decreasing diagram rather than the more general form described in [29].

▶ **Definition 32** (Cyclic critical pairs). Given a layered rewrite system R, let $l \to r, g \to d \in R$ and $p \in \mathcal{FP}os(l)$ such that $\mathcal{V}ar(l) \cap \mathcal{V}ar(g) = \emptyset$, and $l|_p = g$ is unifiable with canonical cyclic unifier $\langle \eta_S = \{\vec{x} \mapsto \vec{u}\}, R_S = \{\vec{y} \to \vec{v}\}\rangle$. Then, $r\eta_S \;{}_R\!\leftarrow l\eta_S =^{cc}_{R_S\eta_S} l[g]_p\eta_S \to_R l[d]_p\eta_S$ is a *cyclic critical peak*, and $\langle r\eta_S, l[d]_p\eta_S\rangle$ is a *cyclic critical pair*, which is said to be *realizable* by the substitution θ iff $(\forall y \to v \in R_S)\, y\theta \twoheadrightarrow_R \;{}_R\!\twoheadleftarrow v\theta$.

The relationship between critical peaks and realizable cyclic critical pairs, usually called critical pair lemma, is more complex than usual:

▶ **Lemma 33** (Cyclic critical pair lemma). *Let $l \to r, g \to d \in R$ such that $\mathcal{V}ar(l) \cap \mathcal{V}ar(g) = \emptyset$. Let $r\sigma \; {}_{l\to r}\overset{\Lambda}{\leftarrow} l\sigma \; {}^{(>\mathcal{FP}os(l))}\twoheadleftarrow \bar{l}\theta = \bar{l}\theta[\bar{g}\theta]_p \twoheadrightarrow^{(>p\cdot\mathcal{FP}os(g))} \bar{l}\theta[g\sigma]_p \to^{p}_{g\to d} \bar{l}\theta[d\sigma]_p$ be a sub-rewriting local peak in $\mathcal{T}_{n+1}$, satisfying $p \in \mathcal{FP}os(l)$ and $\mathcal{V}ar(\bar{l}\theta) \cap \mathcal{V}ar(l,g) = \emptyset$. Assume further that R is Church-Rosser on the set $\mathcal{T}_n$. Then, there exists a cyclic solution $\langle\gamma, R_S\rangle$ such that S is a solved form of the unification problem $l|_p = g$, $\gamma = \eta_S\rho$ for some ρ of domain included in $\mathcal{V}ar(l,g)$, $\sigma \twoheadrightarrow_R \gamma$, and R_S is realizable by γ.*

Proof. Corollary 29 asserts the existence of a solved form $S = \vec{x} = \vec{u} \wedge \vec{y} = \vec{v}$ of the problem $l|_p = g$. But $\langle\sigma, R_S\rangle$ may not be a cyclic solution. We shall therefore construct a new substitution γ such that $\sigma \twoheadrightarrow_{R_R} \gamma$ and $\langle\gamma, R_S\rangle$ is a cyclic solution of the problem, obtained as an instance by some substitution ρ of the most general cyclic unifier $\langle\eta_S, R_S\rangle$ by Theorem 18.

The construction of γ has two steps. The first aims at forcing the equality constraints given by S. This step will result in each parameter having possibly many different values. The role of the second step will be to construct a single value for each parameter.

We start equalizing independently equations $z = s \in S$. Since $\text{Equalize}_n(z^j)^\theta_\sigma$, $\text{Equalize}_n(\bar{s})^\theta_\sigma$ and $\text{Conv}^\theta_n(z^j,\bar{s})$, $z\sigma$ and $s\sigma$ are n-convertible by Lemma 26. By assumption, $z\sigma$ and $s\sigma$ are joinable, hence there exists a term t^s_z such that $z\sigma \twoheadrightarrow_R t^s_z \; {}_R\twoheadleftarrow s\sigma$. Since $\text{OF}(s)$ by Corollary 29, the derivation from $s\sigma$ to t^s_z must occur at positions below $\mathcal{FP}os(s)$. Maintaining equalities in $s\sigma$ between different occurrences of each variable in $\mathcal{V}ar(s)$, we get $t^s_z = s\tau^s_z$ for some τ^s_z. For each parameter p, $p\sigma \twoheadrightarrow_R p\tau^s_z$, hence the elements of the non-empty set $\{p\tau^s_z : p \in \mathcal{V}ar(s) \text{ for some } z = s \in S\}$ are n-convertible thanks to rank non-increasingness. By our Church-Rosser assumption, they can all be rewritten to a same term t_p. We now define γ:

(i) parameters. Given $p \in \mathcal{P}$, we define $\gamma(p) = t_p$. By construction, $p\sigma \twoheadrightarrow_R t_p = p\gamma$.
(ii) finite variables. Given $x = u \in \vec{x} = \vec{u}$, let $\gamma(x) = u\gamma_{|\mathcal{P}}$, thus $x\gamma = u\gamma$. By construction, $x\sigma \twoheadrightarrow_R u\tau^u_x \twoheadrightarrow_R u\gamma$, hence $x\sigma \twoheadrightarrow_R x\gamma$.
(iii) cyclic variables. Given $y = v \in \vec{x} = \vec{u}$, let $\gamma(y) = y\sigma$, making $y\sigma \twoheadrightarrow_R y\gamma$ trivial.
(iv) variables in $\mathcal{V}ar(l,g) \setminus \mathcal{V}ar(l|_p, g)$, that is, those variables from the context $l[\cdot]_p$ which do not belong to the unification problem $l|_p = g$, hence to the solved form S. Given $z \in \mathcal{V}ar(l,g) \setminus \mathcal{V}ar(l|_p, g)$, let $\gamma(z) = z\sigma$, making $z\sigma \twoheadrightarrow_R z\gamma$ trivial.

Therefore $\sigma \twoheadrightarrow_R \gamma$. We proceed to show $\langle\gamma, R_S\rangle$ is a cyclic solution of $l|_p = g$. Take $\rho = \gamma|_{\neg\vec{x}}$. It is routine to see $\gamma = \eta_S\rho$, and to check that $\langle\eta_S, R_S\rangle$ is a cyclic unifier of S by Definition 14, hence of $l|_p = g$ by Lemma 17. Hence the statement.

We end up the proof by noting that γ is a realizer of R_S. ◀

In case of NKH, the lemma is straightforward since solved forms have no parameters.

Our proof strategy for proving confluence of layered systems is as follows: assuming that n-convertible terms are joinable, we show that $(n+1)$-convertible terms are $(n+1)$-joinable by exhibiting appropriate decreasing diagrams for all their local peaks. To this end, we need to define a labelling schema for sub-rewriting. Assuming that rules have an integer index, different rules having possibly the same index, a step $u \underset{R_R}{\overset{p}{\to}} v$ with the rule $l_i \to r_i$ is labelled by the pair $\langle rk(u|_p), i\rangle$. Pairs are compared in the order $\trianglerighteq = (\geq_N, \geq_N)_{lex}$ whose strict part is well-founded. Indexes give more flexibility (shared indexes give even more) in finding decreasing diagrams for critical pairs, this is their sole use.

▶ **Definition 34.** Let $l \to_i r, g \to_j d \in R$ and $p \in \mathcal{FP}os(l)$ such that $l|_p = g$ has a solved form S. Then, the cyclic critical pair $\langle r\eta_S, l[d]_p\eta_S\rangle$ has a *cyclic-joinable decreasing diagram* if

$r\eta_S \overset{\langle 1,I\rangle}{\underset{R}{\twoheadrightarrow}} s =^{cc}_{R_S\eta_S} t \,{}_R^{\langle 1,J\rangle}\!\twoheadleftarrow l[d]_p\eta_S$, whose sequences of indexes I and J satisfy the decreasing diagram condition, with the additional condition, in case $\mathcal{V}ar(l[\cdot]_p) \neq \emptyset$, that all steps have a rule index $k < i$.

By Corollary 29, the ranks of $l\eta_S$ and $l[g]_p\eta_S$ are 1. Thanks to rank non-increasingness and Definition 21, the cyclic-joinable decreasing diagram –but the congruence closure part– is made of terms of rank 1 except possibly s and t which may have rank 0. It follows that all redexes rewritten in the diagram have rank 1. The decreasing diagram condition is therefore ensured by the rule indexes, which justifies our formulation.

Note further that the condition $\mathcal{V}ar(l[\cdot]_p) = \emptyset$ is automatically satisfied when $p = \Lambda$, hence no additional condition is needed in case of a root overlap. In case where $\mathcal{V}ar(l[\cdot]_p) \neq \emptyset$, implying a non-root overlap, the additional condition aims at ensuring that the decreasing diagram is stable under substitution. It implies in particular that there exists no i-facing step. This may look restrictive, and indeed, we are able to prove a slightly better condition: (i) there exists no i-facing step, and (ii) each step $u \rightarrow^q_k v$ using rule k at position q satisfies $k < i$ or $\mathcal{V}ar(u|_q) \subseteq \mathcal{V}ar(g\eta_S)$. We will restrict ourselves here to the simpler condition which yields a less involved confluence proof.

We can now state and prove our main result:

▶ **Theorem 35.** *Rank non-increasing layered systems are confluent provided their realizable cyclic critical pairs have cyclic-joinable decreasing diagrams.*

Proof. Since $\rightarrow_R \subseteq \rightarrow_{R_R} \subseteq \underset{R}{\twoheadrightarrow}$, R-convertibility and R_R-convertibility coincide. We can therefore apply van Oostrom's theorem to R_R-conversions, and reason by induction on the rank. We proceed by inspection of the sub-rewriting local peaks $v \,{}^{p}_{(l\rightarrow r)_R}\!\leftarrow u \rightarrow^{q}_{(g\rightarrow d)_R} w$, with $\mathcal{V}ar(l) \cap \mathcal{V}ar(g) = \emptyset$. We also assume for convenience that $\mathcal{V}ar(l, g) \cap \mathcal{V}ar(u, v, w) = \emptyset$. This allows us to consider u, v, w as ground terms by adding their variables as new constants. We assume further that variables $x, y \in \mathcal{V}ar(l, g)$ become linearized variables x^i, y^j in $\bar{l}, \bar{g}$, and that ξ is the substitution such that $\xi(x^i) = x$ and $\xi(y^j) = y$, hence implying $\mathcal{V}ar(\bar{l}) \cap \mathcal{V}ar(\bar{g}) = \emptyset$.

By definition of sub-rewriting, $u|_p = \bar{l}\theta \twoheadrightarrow_R^{(>\mathcal{FP}os(l))} v'|_p = l\sigma$ and $v = u[r\sigma]_p$, where for all positions $o \in \mathcal{P}os(l)$ such that $l|_o = x$ and $\bar{l}|_o = x^i$, then $x^i\theta \twoheadrightarrow_R x\sigma$. Similarly, $u|_q = \bar{g}\theta \twoheadrightarrow_R^{(>\mathcal{FP}os(g))} w'|_q = g\sigma$ and $w = u[d\sigma]_q$, where for all positions $o \in \mathcal{P}os(g)$ such that $g|_o = y$ and $\bar{g}|_o = y^j$, then $y^j\theta \twoheadrightarrow_R y\sigma$. There are three cases:

1. $p\#q$. The case of disjoint redexes is as usual.
2. $q > p \cdot \mathcal{FP}os(l)$, the so-called ancestor peak case, for which sub-rewriting shows its strength. W.l.o.g. we assume $u|_p$ has some rank $n + 1$ and note that $u|_q$ has some rank $m \leq n$ by Corollary 30. Since the sub-rewriting steps from u to w occur strictly below $p \cdot \mathcal{FP}os(l)$, then $q = p \cdot o \cdot q'$ where $l|_o = \xi(y^j)$ and $\bar{l}|_o = y^j$. It follows that $w = \bar{l}\tau$ for some τ which is equal to θ for all variables in $\bar{l}$ except y^j for which $\tau(y^j) = \theta(y^j)[d\sigma]_{q'}$. We proceed as follows: we equalize all n-convertible terms $\{x\sigma : x \in \mathcal{V}ar(r)\}$ in v and $\{y\tau : y \in \mathcal{V}ar(\bar{l})\}$ in w by induction hypothesis, yielding s, t. Note that steps from v to s have ranks strictly less than the rank $n + 1$ of the step $u \rightarrow_{R_R} v$ by Corollary 30 and rank non-increasingness. Then, t is an instance of l by some γ, and s is the corresponding instance of r, hence t rewrites to s with $l \rightarrow r$. The equalization steps from w to t have ranks which are not guaranteed to be strictly less than m, hence cannot be kept to build a decreasing diagram. But they can be absorbed in a sub-rewriting step from w to s whose first label is at most $n + 1$, hence faces the step from $u \rightarrow_{R_R} v$: sub-rewriting allows us to rewrite directly from w to s, short-cutting the rewrites from w to t that would otherwise

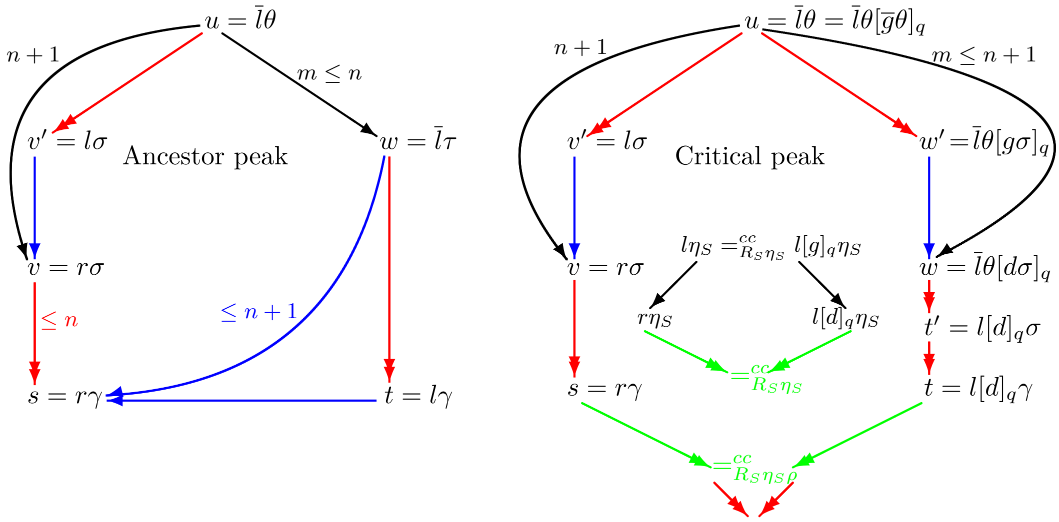

Figure 2 Ancestor and Critical Peaks.

yield a non-decreasing diagram. The proof is depicted at Figure 2 (left), assuming $p = \Lambda$ for simplicity. Black color is used for the given sub-rewriting local peak, blue for arrows whose redexes have ranks at most $n + 1$, and red when redex has rank at most n.

3. $q \in p \cdot \mathcal{FP}os(l)$, the so-called critical peak case, whose left and right rewrite steps have labels $\langle n + 1, i\rangle$ and $\langle m, j\rangle$ respectively, with rules $l \to r$ and $g \to d$ having indexes i and j. Assuming without loss of generality that $p = \Lambda$, the proof is depicted at Figure 2 (right). Most technical difficulties here originate from the fact that the context $l[\cdot]_q$ may have variables. In this case, we first rewrite w to $t' = l\sigma[d\sigma]_q = l[d]_q\sigma$ by replaying those equalization steps, of rank at most n, used in the derivation from u to v', which apply to variable positions in $\mathcal{V}ar(\bar{l}[\cdot]_q)$.
Now, since $\bar{l}\theta = \bar{l}\theta[\bar{g}\theta]_q$, by Lemma 33, there is a substitution γ and a solved form S of the unification problem $l|_q = g$, such that $\sigma \twoheadrightarrow_R \gamma$, $\gamma = \eta_S\rho$ for some ρ, and R_S is realizable by γ. By assumption, the cyclic critical pair $\langle r\eta_S, l[d]_q\eta_S\rangle$ has a cyclic-joinable decreasing diagram (modulo $=^{cc}_{R_S\eta_S}$). We can now lift this diagram to the pair $\langle s, t\rangle$ by instantiation with the substitution ρ. The congruence closure used in the lifted diagram becomes therefore $=^{cc}_{R_S\eta_S\rho}$. We are left showing that the obtained diagram for the pair $\langle v, w\rangle$ is decreasing with respect to the local peak $v \leftarrow u \to w$.
This diagram is made of three distinct parts: the equalization steps, the rewrite steps instantiating the cyclic-joinability assumption with ρ, which originate from s and t –we call them the middle part–, and the congruence closure steps. By Corollary 30, the left equalization steps $v = r\sigma \twoheadrightarrow_R r\gamma = s$ use rewrites with redexes of rank at most n, hence their labels are strictly smaller than $\langle n + 1, i\rangle$. The right equalization steps $w \twoheadrightarrow t' \twoheadrightarrow t$ are considered together with the (green-)middle-part rewrite steps. There are two cases depending on whether $l[\cdot]_q$ is variable-free or not:

(a) $\mathcal{V}ar(l[\cdot]_q) = \emptyset$, hence $m = n + 1$ by Corollary 30. In this case, $w = t'$, and by Corollary 30, the rewrite steps $w = l[d]_q\sigma \twoheadrightarrow_R l[d]_q\gamma = t$ have redexes of rank at most n, making their labels strictly smaller than $\langle m, j\rangle = \langle n + 1, j\rangle$. Let us now consider the middle-part rewrite steps. Thanks to rank non-increasingness, all terms in this part have rank at most $n + 1$. It follows that the associated labels are pairs of the form $\{\langle n', i'\rangle : n' \leq n + 1, i' \in I\}$ on the left, or $\{\langle n', j'\rangle : n' \leq n + 1, j' \in J\}$ on the right. The assumption that I, J satisfy the decreasing diagram condition for the critical peak ensures that these rewrites do satisfy the decreasing diagram condition with respect to the local peak $v \leftarrow u \rightarrow w$ as well.

(b) $\mathcal{V}ar(l[\cdot]_q) \neq \emptyset$. By Corollary 30, the right equalization steps $w \twoheadrightarrow t' \twoheadrightarrow t$ have redexes of rank at most n, making their labels strictly smaller than $\langle n+1, i\rangle$. Consider now the middle part. Thanks to rank non-increasingness and the additional condition on the cyclic-joinability assumption of the cyclic critical pair, all labels $\langle n', k\rangle$ in the middle part satisfy $n' \leq n+1$ and $k < i$, hence are strictly smaller than $\langle n+1, i\rangle$.

We are left with the congruence closure steps. Given $y = v \in R_S$, $y\gamma \twoheadrightarrow_R {}_R\!\twoheadleftarrow v\gamma$ since R_S is realizable by γ. By Lemma 27, OF(v) holds, hence $y\gamma$ and $v\gamma$ are n-convertible by rank non-increasingness. We are left with replacing the $=^{cc}_{\vec{y}\gamma=\vec{v}\gamma}$-steps by a joinability diagram whose all steps have rank at most n. The obtained diagram is therefore decreasing, which ends the proof. ◀

Using the improved condition of cyclic-joinability mentioned after Definition 34 requires modifying the discussion concerning the (green-)middle-part rewrite steps. Although this does not cause any conceptual difficulties, it is technically delicate. The interested reader can of course reconstruct this proof for himself or herself.

Our result gives an answer to NKH: confluence of critical pair free rewrite systems can be analyzed via their sub-rewriting critical pairs, which are actually the cyclic critical pairs.

NKH is critical pair free but non-confluent. Indeed, it has the Ω solved form $x = c(y) \wedge y = c(y)$ obtained by unifying $f(x,x) = f(y, c(y))$. The cyclic critical peak is then $a \leftarrow f(x,x) =^{cc} f(y,c(y)) \rightarrow b$ yielding the cyclic critical pair $\langle a, b\rangle$ which is not joinable modulo $\{x = c(y), y = c(y)\}$.

We now give a slight modification of NKH making it confluent:

▶ **Example 36.** The system $R = \{f(x,x) \rightarrow_2 a(x,x),\ f(x,c(x)) \rightarrow_2 b(x),\ f(c(x),c(x)) \rightarrow_3 f(x,c(x)),\ a(x,x) \rightarrow_1 e(x),\ b(x) \rightarrow_1 e(c(x)),\ g \rightarrow_0 c(g)\}$ is confluent. Showing that R satisfies (DLO) is routine, and it is rank non-increasing by Lemma 23. There are three cyclic critical pairs, which all have a cyclic-joinable decreasing diagram. For instance, the unification $f(x,x) = f(y,c(y))$ returns a canonical cyclic unifier $\langle \eta_S = \emptyset, R_S = \{x \rightarrow c(y), y \rightarrow c(y)\}\rangle$, the corresponding cyclic critical peak $a(x,x)\ {}^{\langle 1,2\rangle}\!\!\leftarrow f(x,x) =^{cc}_{R_S\eta_S} f(y,c(y)) \rightarrow^{\langle 1,2\rangle} b(y)$ has a cyclic-joinable decreasing diagram $a(x,x) \rightarrow^{\langle 1,1\rangle} e(x) =^{cc}_{R_S\eta_S} e(c(y))\ {}^{\langle 1,1\rangle}\!\!\leftarrow b(y)$. The unification $f(x,x) = f(c(y),c(y))$ returns $\langle \eta_S = \{x = c(y)\}, R_S = \emptyset\rangle$, the corresponding (normal) critical peak $a(c(y),c(y))\ {}^{\langle 1,2\rangle}\!\!\leftarrow f(c(y),c(y)) \rightarrow^{\langle 1,3\rangle} f(y,c(y))$ decreases by $a(c(y),c(y)) \rightarrow^{\langle 1,1\rangle} e(c(y))\ {}^{\langle 1,1\rangle}\!\!\leftarrow b(y)\ {}^{\langle 1,2\rangle}\!\!\leftarrow f(y,c(y))$. By Theorem 35, R is confluent.

Theorem 35 can be easily used positively: if all cyclic critical pairs have cyclic-joinable decreasing diagrams, then confluence is met. This was the case in Example 36. But there is another positive use that we illustrate now: showing that $\{f(x,x) \rightarrow a, f(x,c(x)) \rightarrow b, g \rightarrow d(g)\}$ is confluent requires proving that the cyclic critical pair given by unifying the first two rules is not realizable. Although realizability is undecidable in general, this is the case here since there is no term s convertible to $c(s)$. Theorem 35 can also be used negatively by exhibiting some realizable cyclic critical pair which is not joinable: this is the case of example NKH. In general, if some realizable cyclic critical pair leading to a local peak is not joinable, then the system is non-confluent. Whether a realizable cyclic critical pair always yields a local peak is still an open problem which we had no time to investigate yet.

A main assumption of our result is that rules may not increase the rank. One can of course challenge this assumption, which could be due to the proof method itself. The following counter-example shows that it is not the case.

▶ **Example 37.** Consider the critical pair free system $R = \{d(x,x) \to 0,\ f(x) \to d(x, f(x)),\ c \to f(c)\}$, which is layered but whose second rule is rank increasing since $d(x^1, x^2)$ unifies with $d(y, f(y))$. This system is non-confluent, since $f(fc) \to d(fc, ffc) \to d(ffc, ffc) \to 0$ while $f(fc) \to f(d(c, fc)) \to f(d(fc, fc)) \to f0$ which generates the regular tree language $\{S \to d(0, S), S \to f0\}$ not containing 0. Note that replacing the second rule by the right linear rule $f(x) \to d(x, f(c))$ yields a confluent system [24].

Releasing rank non-increasingness would indeed require strengthening another assumption, possibly imposing left- or right-linearity.

7 Conclusion

Decreasing diagrams opened the way for generalizing Knuth and Bendix's critical-pair test for confluence to non-terminating systems, re-igniting these questions. Our results answer open problems by allowing non-terminating rules which can also be non-linear on the left as well as on the right. The notion of layered systems is our first conceptual contribution here.

Another, technical contribution of our work is the notion of sub-rewriting, which can indeed be compared to parallel rewriting. Both relations contain plain rewriting, and are included in its transitive closure. Both can therefore be used for studying confluence of plain rewriting. Tait and Martin-Löf's parallel rewriting –as presented by Barendregt in his famous book on Lambda Calculus [2]– has been recognized as the major tool for studying confluence of left-linear non-terminating rewrite relations when they are not right-linear. We believe that sub-rewriting will be equally successful for studying confluence of non-terminating rewrite relations that are not left-linear. In the present work where no linearity assumption is made, assumption (DLO) ensuring the absence of stacked critical pairs in lefthand sides makes the combined use of sub-rewriting and parallel rewriting superfluous. Without that assumption, as is the case in [17], their combined use becomes necessary.

A last contribution, both technical and conceptual, is the notion of cyclic unifiers. Although their study is still preliminary, we have shown that they constitute a powerful new tool to handle unification problems with cyclic equations in the same way we deal with unification problems without cyclic equations, thanks to the existence of most general cyclic unifiers which generalize the usual notion of mgu. This indeed opens the way to a *uniform treatment* of problems where unification, whether finite or infinite, plays a central role.

Our long-term goal goes beyond improving the current toolkit for carrying out confluence proofs for non-terminating rewrite systems. We aim at designing new tools for showing confluence of complex type theories (with dependent types, universes and dependent elimination rules) directly on raw terms, which would ease the construction of strongly normalizing models for typed terms. Since redex-depth, the notion of rank used here, does not behave well for higher-order rules, appropriate new notions of rank are required in that setting.

Acknowledgments. This research is supported in part by NSFC Programs (No. 91218302, 61272002) and MIIT IT funds (Research and application of TCN key technologies) of China, and in part by JSPS, KAKENHI Grant-in-Aid for Exploratory Research 25540003 of Japan.

References

1 Takahito Aoto, Yoshihito Toyama, and Kazumasa Uchida. Proving confluence of term rewriting systems via persistency and decreasing diagrams. In Dowek [7], pages 46–60.

2 Hendrik P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 1984.

3 Alain Colmerauer. Equations and inequations on finite and infinite trees. In *FGCS*, pages 85–99, 1984.

4 Hubert Comon, Max Dauchet, Remi Gilleron, Christof Löding, Florent Jacquemard, Denis Lugiez, Sophie Tison, and Marc Tommasi. Tree automata techniques and applications. Available on: `http://www.grappa.univ-lille3.fr/tata`, 2007. release October, 12th 2007.

5 Nachum Dershowitz and Jean-Pierre Jouannaud. Rewrite systems. In *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics (B)*, pages 243–320. North-Holland, 1990.

6 Nachum Dershowitz, Mitsuhiro Okada, and G. Sivakumar. Confluence of conditional rewrite systems. In Stéphane Kaplan and Jean-Pierre Jouannaud, editors, *Conditional Term Rewriting Systems, 1st International Workshop, Orsay, France, July 8-10, 1987, Proceedings*, volume 308 of *Lecture Notes in Computer Science*, pages 31–44. Springer, 1987.

7 Gilles Dowek, editor. *Rewriting and Typed Lambda Calculi – Joint International Conference, RTA-TLCA 2014, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 14-17, 2014. Proceedings*, volume 8560 of *Lecture Notes in Computer Science*. Springer, 2014.

8 Bertram Felgenhauer. Rule labeling for confluence of left-linear term rewrite systems. In *IWC*, pages 23–27, 2013.

9 Jean H. Gallier, Stan Raatz, and Wayne Snyder. Theorem proving using rigid E-unification equational matings. In *Proceedings of the Symposium on Logic in Computer Science (LICS'87), Ithaca, New York, USA, June 22-25, 1987*, pages 338–346. IEEE Computer Society, 1987.

10 Hiroshi Gomi, Michio Oyamaguchi, and Yoshikatsu Ohta. On the Church-Rosser property of root-E-overlapping and strongly depth-preserving term rewriting systems. *IPSJ*, 39(4):992–1005, 1998.

11 Nao Hirokawa and Aart Middeldorp. Decreasing diagrams and relative termination. *J. Autom. Reasoning*, 47(4):481–501, 2011.

12 Gérard P. Huet. Confluent reductions: Abstract properties and applications to term rewriting systems. *J. ACM*, 27(4):797–821, 1980.

13 Jean-Pierre Jouannaud and Claude Kirchner. Solving equations in abstract algebras: A rule-based survey of unification. In *Computational Logic – Essays in Honor of Alan Robinson*, pages 257–321, 1991.

14 Jean-Pierre Jouannaud and Vincent van Oostrom. Diagrammatic confluence and completion. In Susanne Albers, Alberto Marchetti-Spaccamela, Yossi Matias, Sotiris E. Nikoletseas, and Wolfgang Thomas, editors, *Automata, Languages and Programming, 36th International Colloquium, ICALP 2009, Rhodes, Greece, July 5-12, 2009, Proceedings, Part II*, volume 5556 of *Lecture Notes in Computer Science*, pages 212–222. Springer, 2009.

15 Dominik Klein and Nao Hirokawa. Confluence of non-left-linear TRSs via relative termination. In Nikolaj Bjørner and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning – 18th International Conference, LPAR-18, Mérida, Venezuela, March 11-15, 2012. Proceedings*, volume 7180 of *Lecture Notes in Computer Science*, pages 258–273. Springer, 2012.

16 Jan Willem Klop. Term rewriting systems. In *Handbook of Logic in Computer Science, Vol.2*, pages 1–116. Oxford University Press, 1993.

17 Jiaxiang Liu, Nachum Dershowitz, and Jean-Pierre Jouannaud. Confluence by critical pair analysis. In Dowek [7], pages 287–302.

18 Jiaxiang Liu and Jean-Pierre Jouannaud. Confluence: The unifying, expressive power of locality. In Shusaku Iida, José Meseguer, and Kazuhiro Ogata, editors, *Specification,*

Algebra, and Software – Essays Dedicated to Kokichi Futatsugi, volume 8373 of *Lecture Notes in Computer Science*, pages 337–358. Springer, 2014.

19 Ken Mano and Mizuhito Ogawa. Unique normal form property of compatible term rewriting systems: a new proof of Chew's theorem. *Theor. Comput. Sci.*, 258(1-2):169–208, 2001.

20 Kunihiro Matsuura, Michio Oyamaguchi, Yoshikatsu Ohta, and Mizuhito Ogawa. On the E-overlapping property of nonlinear term rewriting systems (in Japanese). *IEICE*, 80-D-I(11):847–855, 1997.

21 Greg Nelson and Derek C. Oppen. Fast decision procedures based on congruence closure. *J. ACM*, 27(2):356–364, 1980.

22 Satoshi Okui. Simultaneous critical pairs and Church-Rosser property. In Tobias Nipkow, editor, *Rewriting Techniques and Applications, 9th International Conference, RTA-98, Tsukuba, Japan, March 30 – April 1, 1998, Proceedings*, volume 1379 of *Lecture Notes in Computer Science*, pages 2–16. Springer, 1998.

23 Barry K. Rosen. Tree-manipulating systems and Church-Rosser theorems. *J. ACM*, 20(1):160–187, 1973.

24 Masahiko Sakai and Mizuhito Ogawa. Weakly-non-overlapping non-collapsing shallow term rewriting systems are confluent. *Inf. Process. Lett.*, 110(18-19):810–814, 2010.

25 Masahiko Sakai, Michio Oyamaguchi, and Mizuhito Ogawa. Non-E-overlapping, weakly shallow, and non-collapsing TRSs are confluent. In *CADE*, 2015. to appear.

26 Terese. Term rewriting systems. In *Cambridge Tracts in Theoretical Computer Science*, volume 55. Cambridge University Press, 2003.

27 Yoshihito Toyama and Michio Oyamaguchi. Church-Rosser property and unique normal form property of non-duplicating term rewriting systems. In Nachum Dershowitz and Naomi Lindenstrauss, editors, *Conditional and Typed Rewriting Systems, 4th International Workshop, CTRS-94, Jerusalem, Israel, July 13-15, 1994, Proceedings*, volume 968 of *Lecture Notes in Computer Science*, pages 316–331. Springer, 1994.

28 Vincent van Oostrom. Confluence by decreasing diagrams. *Theor. Comput. Sci.*, 126(2):259–280, 1994.

29 Vincent van Oostrom. Confluence by decreasing diagrams converted. In Andrei Voronkov, editor, *Rewriting Techniques and Applications, 19th International Conference, RTA 2008, Hagenberg, Austria, July 15-17, 2008, Proceedings*, volume 5117 of *Lecture Notes in Computer Science*, pages 306–320. Springer, 2008.

30 Harald Zankl, Bertram Felgenhauer, and Aart Middeldorp. Labelings for decreasing diagrams. In Manfred Schmidt-Schauß, editor, *Proceedings of the 22nd International Conference on Rewriting Techniques and Applications, RTA 2011, May 30 – June 1, 2011, Novi Sad, Serbia*, volume 10 of *LIPIcs*, pages 377–392. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2011.