Triangulation in rewriting

Vincent van Oostrom[1] and Hans Zantema[2]

1 Department of Philosophy, Utrecht University, The Netherlands
`Vincent.vanOostrom@phil.uu.nl`
2 Department of Computer Science, TU Eindhoven, The Netherlands
Institute for Computing and Information Sciences, Radboud University
Nijmegen, The Netherlands
`h.zantema@tue.nl`

⸺ Abstract ⸺

We introduce a process, dubbed *triangulation*, turning any rewrite relation into a confluent one. It is more direct than usual completion, in the sense that objects connected by a peak are directly related rather than their normal forms. We investigate conditions under which this process preserves desirable properties such as termination.

1 Introduction

We study the problem of deciding whether two objects are equivalent with respect to the equivalence relation generated by some rewrite relation. We do this in a fully abstract setting, that is, any binary relation on any set of objects may serve as a rewrite relation. The standard idea in such a setting is to compute the normal forms of both objects with respect to the rewrite relation, subsequently comparing whether these normal forms are equal or not. If and only if the normal forms are equal the original objects are deemed to be equivalent.

For the above to work, i.e. for it to yield a sound and complete decision procedure for equivalence, normal forms should *exist* and be *unique* within equivalence classes. For a given rewrite relation neither needs to be the case. On the one hand, the objects b and c are distinct normal forms with respect to the rewrite relation $b \twoheadleftarrow a \twoheadrightarrow c$ despite them belonging to the same equivalence class $\{a, b, c\}$ generated by $\twoheadrightarrow$ (see Figure 1 left); normal forms are *not unique* so the procedure is not sound. On the other hand, the object a does not have a normal form in the rewrite relation $a \twoheadrightarrow a$; normal forms need *not exist*, even if they are unique when they exist, hence the procedure need not be complete.
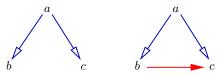

Figure 1 $b \twoheadleftarrow a \twoheadrightarrow c$ before (left) and after (right) triangulation

First we introduce in Section 2 a process called *triangulation* to stepwise extend an arbitrary rewrite relation in such a way that normal forms are *unique* within each equivalence

class, without altering the generated equivalence relation. That is, after triangulation the decision procedure is sound: if two objects both rewrite to a normal form, then they are equivalent if and only if these normal forms are equal. For instance, triangulation would extend the above rewrite relation $b \leftarrow\!\!\!\triangleleft\ a \triangleright\!\!\!\rightarrow c$ by either of the steps $b \rightarrow\!\!\!\blacktriangleright c$ (see Figure 1 right) or $b \blacktriangleleft\!\!\!\leftarrow c$, forming a triangle, hence our naming. The idea of triangulation is similar to that of abstract Knuth–Bendix completion, see e.g. [2]; the main difference is that by triangulation two not directly related objects that are related by a peak will be related directly, while in completion their normal forms will be related.

Next, in Section 3, we provide two sufficient conditions for triangulation to preserve termination, guaranteeing *existence* of normal forms when starting out with a terminating rewrite relation, thus giving rise to a sound and complete decision procedure for the generated equivalence. The first condition is a *compatibility* condition, requiring the union of the original rewrite relation and the relation used to relate the adjoined steps to be terminating. The second condition requires both these relations to be terminating on their own, and moreover the original rewrite relation to be *codeterministic*; at most one object rewrites to any given object.

Finally, in Section 4, we reflect on how the triangulation process brings about completeness. That is, looking back from the result of the completion process, i.e. from the original rewrite relation and the relation by which it is extended, we provide sufficient conditions on these two relations for their union to be complete. These conditions are abstract in that they do not require the latter being stepwise generated from the former. We also provide sufficient conditions for the result of triangulation to be cocomplete, i.e. coconfluent and coterminating.

The origin of this research was in a question of Jan Friso Groote, relevant for the typing mechanism in the tool-set of the specification language mCRL2 [5]. In a part of that mechanism types $a, b, c, d, \ldots$ are specified, some of which are equated, expressed by definitions of the shape $x \leftarrow\!\!\!\triangleleft\ y$ with x the definiendum and y the definiens. A natural question that came up is whether two types are equivalent in the sense that the one can be reached by the other by means of a series of definition foldings and unfoldings. For instance, in the system having six types a, b, c, d, e, f with four definitions

$$a \leftarrow\!\!\!\triangleleft\ b,\ b \leftarrow\!\!\!\triangleleft\ c,\ d \leftarrow\!\!\!\triangleleft\ c,\ f \leftarrow\!\!\!\triangleleft\ e$$

the types a and d are seen to be equivalent since they can be connected by the chain of definition foldings and unfoldings $a \leftarrow\!\!\!\triangleleft\ b \leftarrow\!\!\!\triangleleft\ c \triangleright\!\!\!\rightarrow d$. Triangulation arose here as a method to answer the equivalence question, as it constructs unique representatives of equivalence classes, hence checking equivalence of two types reduces to checking equality of their respective representatives ($\rightarrow$-normal forms). Executing it on this example, first the peak $b \leftarrow\!\!\!\triangleleft\ c \triangleright\!\!\!\rightarrow d$ is turned into a triangle by adjoining a definition $b \rightarrow\!\!\!\blacktriangleright d$, the direction being determined by some given order on the objects, here the alphabetic order. This then gives rise to a new peak $a \leftarrow\!\!\!\triangleleft\ b \rightarrow\!\!\!\blacktriangleright d$ which in turn is made into a triangle by adjoining $a \rightarrow\!\!\!\blacktriangleright d$, after which d, f have become the unique representatives of their respective equivalence classes. The types a and b are $\rightarrow\!\!\!\triangleright$-equivalent since d is their common $\rightarrow$-normal form. The example also motivates our interest in studying rewrite relations that are codeterministic, as codeterminism of a system of definitions $\rightarrow\!\!\!\triangleright$ corresponds to the natural requirement, satisfied by the example, that types are not defined twice.

2 Triangulation

We introduce triangulation as a process to turn an arbitrary rewrite relation into a confluent one, without altering the generated equivalence relation. We first quickly recapitulate the few basic notions on rewrite relations needed for this, referring the reader to [1, 6] for background information. Throughout, we will use *relation* to mean a binary relation. A relation R is said to have property *co-P* if its converse has property P.[1] A *rewrite* relation is a binary relation on a set of objects, that is, a relation having the same domain and codomain. Rewrite relations may denote notions of computation steps, and we will use arrow-like notations like $\rightarrow$, $\dashrightarrow$, $\blacktriangleright$ to denote them. For a rewrite relation $\rightarrow$, we inductively define an object a to be *terminating*, if for all objects b such that $a \rightarrow b$, b is terminating. The rewrite relation $\rightarrow$ is *terminating* if all its objects are. For a rewrite relation denoted by an arrow-like notation $\rightarrow$, its converse is denoted by the converse $\leftarrow$ of the notation. We denote the union of two rewrite relations by the union of their notations, e.g. $\leftrightarrow$ denotes $\leftarrow \cup \rightarrow$, the symmetric closure of $\rightarrow$. We say $\rightarrow$ is *total* if $a \leftrightarrow b$ for all objects a, b. We use $\dashrightarrow \cdot \blacktriangleright$ to denote the composition of $\dashrightarrow$ and $\blacktriangleright$, use $\rightarrow^n$ to denote the n-fold composition of $\rightarrow$ with itself, and $\rightarrow^=$ and $\rightarrow^+$ to denote respectively the reflexive and transitive closure of $\rightarrow$. To denote the reflexive–transitive closure of $\rightarrow$, i.e. its 'repetition', we employ the 'repetition' $\twoheadrightarrow$ of its notation, or, if clutter would arise from repeating the notation, $\rightarrow^*$. If $a \twoheadrightarrow b$ then we say that a *reduces* or *rewrites* to b. We define $\rightarrow$-*expansion* as $\rightarrow$-coreduction, i.e. $\leftarrow$-reduction, and $\rightarrow$-*convertibility* as $\leftrightarrow^*$ which is easily shown to be the equivalence closure of $\rightarrow$. A rewrite relation $\rightarrow$ is *confluent* if $\leftarrow \cdot \twoheadrightarrow \subseteq \twoheadrightarrow \cdot \twoheadleftarrow$. A rewrite relation is *complete* if it is both terminating and confluent. Further notions and notations will be introduced on a by-need basis.

Now we are ready for our main definition: we want to extend an arbitrary rewrite relation $\dashrightarrow$ to achieve desired properties like confluence. In doing so, we use a total relation R, typically a total order, to order the added new pairs.

▶ **Definition 2.1** (Triangulation). Let R be a total relation on the objects of a rewrite relation $\dashrightarrow$. The *triangulation* $\mathrm{Tr}(\dashrightarrow)$ of $\dashrightarrow$ with respect to R is the rewrite relation $\rightarrow = \bigcup_{n \geq 1} \rightarrow_n$, where $\rightarrow_n$ is defined inductively as follows:

▪ $\rightarrow_1 = \dashrightarrow$; and
▪ $a \rightarrow_n b$ holds for $n > 1$ if and only if
  ▪ $a \mathrel{R} b$ and for some $m, k \geq 1$ with $m + k = n$ there is a c such that $a \leftarrow_m c \rightarrow_k b$; and
  ▪ for no $k < n$, $a \leftrightarrow^=_k b$ holds.

Below we will assume the relation R to be total.

▶ **Example 2.2.** Triangulating the rewrite relation $\dashrightarrow$ displayed on the left in Figure 2 with respect to the usual alphabetic order on its objects $\{a, \ldots, h\}$ gives rise to the rewrite relation displayed on the right of that figure. The original arrows are labelled by 1. Next for every peak $x \leftarrow \cdot \rightarrow y$ with $x \neq y$ for which there is not yet an arrow between x and y an arrow $x \rightarrow_2 y$ is created or conversely, depending on whether $x \mathrel{R} y$ or $y \mathrel{R} x$ holds. This is continued until after creating the $\rightarrow_5$ arrow, there is a direct arrow between x and y for all peaks $x \leftarrow \cdot \rightarrow y$ with $x \neq y$.

▶ Remark. The index n of a step $a \rightarrow_n b$ is the number of $\dashrightarrow$-steps in a proof showing that a and b are $\dashrightarrow$-convertible. Counting each axiom ($\dashrightarrow$-step) and transitivity rule employed

---

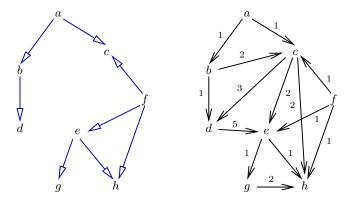[1] We will in particular be interested in codeterminism, coconfluence, cotermination and cocompleteness.

Figure 2 A rewrite relation $\rightarrow\!\!\triangleright$ (left) and its triangulation (right)

in such a proof, this amounts to the same thing as half of (one plus the size of that proof). In Example 2.2 the step $d \rightarrow_5 e$ witnesses a proof of size 9 ($= 2 \times 5 - 1$) based on the $\rightarrow\!\!\triangleright$-conversion $d \triangleleft\!\!- b \triangleleft\!\!- a \rightarrow\!\!\triangleright c \triangleleft\!\!- f \rightarrow\!\!\triangleright e$ consisting of 5 steps and 4 applications of the transitivity rule as witnessed by first $b \rightarrow_2 c$ and $c \rightarrow_2 e$, then $d \leftarrow_3 c$, and finally $d \rightarrow_5 e$.

The rewrite relation in the example is finite, hence triangulation must stop in the sense that no new steps will be adjoined from some stage on (in fact, from stage 5 on). But also for infinite rewrite relations the process may well stop. Triangulation always results in what we will call an *affluent* rewrite relation without altering the generated equivalence relation.

▶ **Definition 2.3** (Affluence). A pair $\rightarrow\!\!\triangleright$, $\rightarrow\!\!\blacktriangleright$ of rewrite relations is *affluent*[2] if $\triangleleft\!\!\triangleleft\!\!- \cdot \rightarrow\!\!\blacktriangleright\!\!\blacktriangleright \subseteq$ $\triangleleft\!\!\triangleleft\!\!- \cup \rightarrow\!\!\blacktriangleright\!\!\blacktriangleright$, *one-step* affluent if $\triangleleft\!\!- \cdot \rightarrow\!\!\blacktriangleright \subseteq \triangleleft\!\!- \cup \rightarrow\!\!\blacktriangleright$, and *locally* affluent if $\triangleleft\!\!- \cdot \rightarrow\!\!\blacktriangleright \subseteq \triangleleft\!\!\triangleleft\!\!- \cup \rightarrow\!\!\blacktriangleright\!\!\blacktriangleright$. These notions pertain to a single rewrite relation $\rightarrow$ via the pair $\rightarrow, \rightarrow$.

Observe that in term rewriting affluence seldomly occurs: even with respect to the single rule $a \rightarrow b$, affluence does not hold for the term $f(a, a)$. So both affluence and triangulation are about rewrite relations in abstract settings, typically not described by a term rewriting system.

▶ **Remark.** One-step affluence of $\rightarrow\!\!\triangleright^=$, $\rightarrow\!\!\blacktriangleright^=$ is equivalent to $\triangleleft\!\!- \cdot \rightarrow\!\!\blacktriangleright \subseteq (\triangleleft\!\!- \cup \rightarrow\!\!\blacktriangleright)^=$, hence implies sub-commutation $\triangleleft\!\!- \cdot \rightarrow\!\!\blacktriangleright \subseteq \rightarrow\!\!\blacktriangleright^= \cdot \triangleleft\!\!-^=$. In turn, local affluence of $\rightarrow\!\!\triangleright$, $\rightarrow\!\!\blacktriangleright$ implies local commutation $\triangleleft\!\!- \cdot \rightarrow\!\!\blacktriangleright \subseteq \rightarrow\!\!\blacktriangleright\!\!\blacktriangleright \cdot \triangleleft\!\!\triangleleft\!\!-$ and affluence of $\rightarrow\!\!\triangleright$, $\rightarrow\!\!\blacktriangleright$ implies commutation $\triangleleft\!\!\triangleleft\!\!- \cdot \rightarrow\!\!\blacktriangleright\!\!\blacktriangleright \subseteq \rightarrow\!\!\blacktriangleright\!\!\blacktriangleright \cdot \triangleleft\!\!\triangleleft\!\!-$. In case of a single rewrite relation, the same holds, replacing commutation by confluence.

The standard example showing that local commutation does not imply commutation, $b \triangleleft\!\!- a \rightleftarrows a' \rightarrow\!\!\blacktriangleright c$, shows local affluence does not imply affluence. Analogous to the fact that one-step commutation implies commutation by the former being preserved under taking reflexive and transitive closures, one-step affluence implies affluence:

▶ **Lemma 2.4.** *If $\rightarrow\!\!\triangleright$, $\rightarrow\!\!\blacktriangleright$ is one-step affluent, then so are $\rightarrow\!\!\triangleright^=$, $\rightarrow\!\!\blacktriangleright^=$ and $\rightarrow\!\!\triangleright^+$, $\rightarrow\!\!\blacktriangleright^+$.*

Proof. Suppose the pair $\rightarrow\!\!\triangleright$, $\rightarrow\!\!\blacktriangleright$ is one-step affluent.

That $\rightarrow\!\!\triangleright^=$, $\rightarrow\!\!\blacktriangleright^=$ is one-step affluent follows from $\triangleleft\!\!-^= \cdot \rightarrow\!\!\blacktriangleright^= = \mathsf{id} \cup \triangleleft\!\!- \cup \rightarrow\!\!\blacktriangleright \cup (\triangleleft\!\!- \cdot \rightarrow\!\!\blacktriangleright)$.

---

[2] The idea is that whereas the standard notion of *confluence* expresses that rewrite sequences (viewed as streams) may 'flow together', we use *affluence* in its original (archaic) sense to express that one of them may 'flow to' (is a tributary of) the other.

To prove that $\rightarrowtriangle^+$, $\blacktriangleright^+$ is one-step affluent, we show for $n, m \geq 0$, $a \leftarrowtail^n \cdot \leftarrowtriangle \cdot \rightarrowtriangle \cdot \blacktriangleright^m$ b implies $a\ (\leftarrowtriangle^+ \cup \blacktriangleright^+)\ b$, by induction on $n+m$. By assumption $a \leftarrowtail^n \cdot (\leftarrowtriangle \cup \blacktriangleright) \cdot \blacktriangleright^m$ b. Suppose w.l.o.g. $a \leftarrowtail^n \cdot \leftarrowtriangle \cdot \blacktriangleright^m\ b$. Then we conclude to $a \leftarrowtriangle^+ b$ if $m = 0$. Otherwise we conclude to $a\ (\leftarrowtriangle^+ \cup \blacktriangleright^+)\ b$ by the induction hypothesis. ◀

▶ **Theorem 2.5.** *Let $\rightarrow = Tr(\rightarrowtriangle)$ be the triangulation of any rewrite relation $\rightarrowtriangle$. Then $\rightarrow$ is affluent and $\leftrightarrow^* = \leftarrowtriangle\rightarrowtriangle^*$.*

Proof. To prove affluence of $\rightarrowtriangle$ it suffices by Lemma 2.4 and the remark above it, to prove $\leftarrow \cdot \rightarrow \subseteq \leftrightarrow^=$. To that end, we show that for all natural numbers $n, m \geq 1$, $a \leftarrow_n c \rightarrow_m b$ implies $a \leftrightarrow^= b$. By the triangulation construction then either a is $\bigcup_{1 \leq k < n+m} \leftrightarrow_k^=$-related to b and we are done, or a is not so related to b and then $a \leftrightarrow_{n+m} b$ by the assumed totality of the relation R.

To prove $\leftrightarrow^* = \leftarrowtriangle\rightarrowtriangle^*$ it suffices, since $\rightarrow = \bigcup_{n \geq 1} \rightarrow_n$, to show $\rightarrow_n \subseteq \leftarrowtriangle\rightarrowtriangle^*$ for all $n \geq 1$ by induction on n. In the base case $\rightarrow_1 = \rightarrowtriangle \subseteq \leftarrowtriangle\rightarrowtriangle^*$. If $a \rightarrow_{n+m} b$ because $a \leftarrow_n c \rightarrow_m b$ for some object c and natural numbers n, m, then by the induction hypothesis $a \leftarrowtriangle\rightarrowtriangle^* c \leftarrowtriangle\rightarrowtriangle^* b$, and we conclude by transitivity of $\leftarrowtriangle\rightarrowtriangle^*$. ◀

▶ **Corollary 2.6.** *The triangulation $Tr(\rightarrowtriangle)$ is confluent for every rewrite relation $\rightarrowtriangle$.*

Having established this basic result, we investigate in the next section on which rewrite relations triangulation is a completion process, i.e. for which rewrite relation does triangulation preserve termination?

3 Completion

When does triangulation yield a *complete* rewrite relation? That is, when does triangulation yield a rewrite relation that is both confluent and terminating? We first present an example showing that, in general, triangulation fails to do so. Analysing the example, we then propose two sufficient conditions for triangulation to preserve termination of the rewrite relation, i.e. for triangulation to be a completion process.
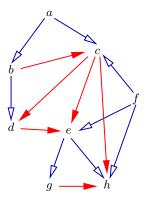

Figure 3 Rendering of Figure 2 using convention to decompose $\rightarrow$ into $\rightarrowtriangle, \blacktriangleright$

To ease discussing the various examples and conditions, we will from now on, when discussing triangulation, use $\blacktriangleright$ to denote $\bigcup_{n>1} \rightarrow_n$. Hence $\rightarrow = \rightarrowtriangle \cup \blacktriangleright$ and this union is disjoint; the triangulation $\rightarrow$ consists of the original relation $\rightarrowtriangle$ and the *triangulating* relation $\blacktriangleright$, cf. Figure 3.

► **Example 3.1.** Consider the rewrite relation $\dashrightarrow$ and its triangulation with respect to the usual greater-than relation $>$ as displayed in Figure 4. The resulting rewrite relation $\rightarrow = \dashrightarrow \cup \twoheadrightarrow$ is not terminating; it is even cyclic: $0 \dashrightarrow 4 \twoheadrightarrow 3 \dashrightarrow 1 \twoheadrightarrow 0$.
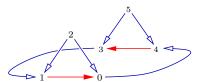

Figure 4 Failure of triangulation to yield a complete rewrite relation

What causes that triangulation succeeds in yielding a complete rewrite relation in Example 2.2? Below we present two conditions, compatibility and codeterminism, each being sufficient for triangulation to be a completion process.

3.1 Compatibility

Our first (trivial) condition is based on the observation that despite that both the original rewrite relation $\dashrightarrow$ and the greater-than relation $>$ with respect to which triangulation takes place in Example 3.1 are terminating, they are not *compatible* in the sense that their union is not terminating; it even is cyclic, e.g. $0 \dashrightarrow 4 > 0$. Since the $\twoheadrightarrow$-steps adjoined by triangulation conform to the relation with respect to which triangulation takes place, the latter being compatible with $\dashrightarrow$ guarantees that termination is preserved:

► **Theorem 3.2.** *The triangulation $Tr(\dashrightarrow)$ of $\dashrightarrow$ with respect to R is terminating if $\dashrightarrow \cup R$ is terminating.*

Proof. Combining $\rightarrow_1 = \dashrightarrow$ with $\rightarrow_n \subseteq R$, for all $n > 1$, we conclude to $\rightarrow = \bigcup_{n \geq 1} \rightarrow_n \subseteq \dashrightarrow \cup R$, hence to termination of $\rightarrow$. ◄

► Remark. The termination assumption on $\dashrightarrow \cup R$ could have been rephrased as: $\dashrightarrow \subseteq R$ and R is terminating. Clearly, the latter entails the former. To see the converse, note that if $a \dashrightarrow b$ but $a \not\mathrel{R} b$, then $b \mathrel{R} a$ by totality of R, hence $a \dashrightarrow b \mathrel{R} a$ contradicting termination of $\dashrightarrow \cup R$. Therefore $\dashrightarrow \subseteq R$ and R is terminating.

► Remark. In ordinary completion, instead of adjoining a step between b and c as in triangulation, a step between their $\dashrightarrow$-normal forms is adjoined if they are distinct. Under the conditions of the theorem, also ordinary completion yields a complete rewrite relation. However, ordinary completion results in confluence, not the stronger affluence guaranteed by triangulation.

► Remark. If triangulation results in a complete rewrite relation, then, although *unboundedly* many stages may have been needed, there cannot be *unbounded creation*. More precisely, there is no infinite sequence of triangulating steps such that each step in the sequence is a cause, one of the two (see Figure 5 left), in the triangulation process for the next step in the sequence, as unbounded creation would contradict termination of the obtained rewrite relation (see Figure 5 right).
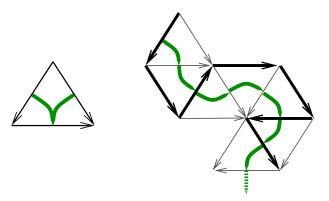
Figure 5 How steps in a peak cause its triangulating step (left) and how an infinite causal chain would give rise to an infinite sequence of steps through the sources of the triangulating steps (right)

3.2 Codeterminism

Our second (nontrivial) condition is based on the observation that the original rewrite relation $\dashrightarrow$ in Example 3.1 is not *codeterministic*: there are objects that are the target of more than one rewrite step, viz. $0 \dashrightarrow 4 \dashleftarrow 5$.

▶ **Definition 3.3.** A binary relation R is *deterministic* if for all objects a in its domain, and all objects b,c in its codomain, $a\ R\ b$ and $a\ R\ c$ imply $b = c$.

Forbidding the above configurations is captured by requiring the rewrite relation to be *codeterministic* per our earlier convention of a relation having a property co-P if its converse has property P.
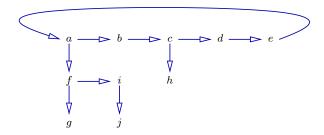

Figure 6 Codeterministic rewrite relation

▶ **Example 3.4.** Neither of the rewrite relations of Example 2.2 (Figure 2 left) and Example 3.1 is codeterministic. For the latter this was observed above. The former fails to be codeterministic because, e.g., $a \dashrightarrow c \dashleftarrow f$. Both the system of definitions given in the introduction in Figure 1 and the rewrite relation displayed in Figure 6 are codeterministic.

Note that the graph in Figure 6 consists of a number of trees branching off from a cycle ($a \dashrightarrow b \dashrightarrow c \dashrightarrow d \dashrightarrow e \dashrightarrow a$). It is easy to see (branching 'off' is allowed by codeterminism, but branching 'in' is not) that this holds in general: each component of the graph of a codeterministic rewrite relation consists of a number (possibly 1) of trees branching off (if at all) from pairwise distinct objects lying at a (possibly empty) cycle. Thus the graph of

an *acyclic* codeterministic rewrite relation is a forest of trees.[3]

▶ Remark. Although it is not unreasonable to require acyclicity, in any case for systems of definitions as in the introduction, one can well imagine an infinite setting without maximal types, that is, for every type there's a definition allowing it to be folded further. In Section 4 we also provide a dual approach, choosing maximally *unfolded* representatives. For the moment we will view termination simply as necessary for the approach via maximally *folded* representatives to make sense.

Intuitively, triangulating is a completion process for codeterministic rewrite relations, since despite that triangulating the different branches from a node in a forest will create new steps 'spanning the gaps' between these branches, no cycles will be created nor will disjoint trees be joined by triangulating, hence termination will be preserved. As formalising this intuition exactly turned out to be tedious,[4] we instead provide a short proof based on the following result due to Doornbos and von Karger. Variations on this result are given by Dershowitz in [3].
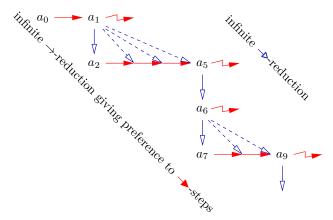

Figure 7 Ramsey-style construction in proof of Lemma 3.5

▶ **Lemma 3.5** ([4]). *For rewrite relations* $\dashrightarrow$*,* $\blacktriangleright$ *let* $\rightarrow = \dashrightarrow \cup \blacktriangleright$*. Then* $\rightarrow$ *is terminating if* $\dashrightarrow$ *and* $\blacktriangleright$ *are terminating, and* $\dashrightarrow \cdot \blacktriangleright \subseteq \dashrightarrow \cup (\blacktriangleright \cdot \twoheadrightarrow)$*.*[5]

Proof. We provide a Ramsey-style proof, see Figure 7, as an alternative to the calculational proof in [4]. Without loss of generality we may assume $\dashrightarrow$ and $\blacktriangleright$ to be disjoint (otherwise consider $\dashrightarrow - \blacktriangleright$ and $\blacktriangleright$).

For a proof by contradiction, suppose an object a_0 such that $\infty(a_0)$ were to exist, writing $\infty(a)$ to denote that a allows an infinite $\rightarrow$-reduction. Such an infinite $\rightarrow$-reduction sequence may then be constructed through objects a_n with $\infty(a_n)$, while giving preference to $\blacktriangleright$-steps. Formally: suppose the sequence has been constructed up to a_n. If there exists an object a such that $a_n \blacktriangleright a$ and $\infty(a)$, then we set a_{n+1} to a. Otherwise we set a_{n+1} to any object a such that $a_n \dashrightarrow a$ and $\infty(a)$ (which must exist since $\infty(a_n)$ and $\rightarrow = \dashrightarrow \cup \blacktriangleright$).

We show one can construct an infinite $\dashrightarrow$-subsequence through objects *that do not allow a $\blacktriangleright$-step to an object b with $\infty(b)$*. Formally: we start out with the empty subsequence for some object a_n for a pair of indices $n, n+1$ such that $a_n \dashrightarrow a_{n+1}$. Such a pair must exist

---

[3] The trees may be infinite: both infinitely branching and non-rooted trees are allowed.
[4] Still, that formalisation may be interesting, e.g. from a proof-theoretic point of view.
[5] The latter property was dubbed *lazy commutation* in [3].

since otherwise the original reduction sequence would be an infinite $\to_\blacktriangleright$-reduction sequence. Per construction of the original sequence, $a_n \to_\triangleright a_{n+1}$ entails that a_n does not allow a $\to_\blacktriangleright$-step to an object b with $\infty(b)$. Next, suppose to have a subsequence ending in a_n for some pair of indices $n < m$ such that $a_n \to_\triangleright a_m$.

- In case $a_m \to_\triangleright a_{m+1}$ (e.g. $a_1 \to_\triangleright a_5 \to_\triangleright a_6$ for $n = 1$, $m = 5$ in Figure 7), we extend the subsequence with a_m and continue with the pair of indices $m, m+1$.
- Otherwise $a_n \to_\triangleright a_m \to_\blacktriangleright a_{m+1}$ (e.g. $a_6 \to_\triangleright a_7 \to_\blacktriangleright a_8$ in Figure 7), hence by assumption $a_n \ (\to_\triangleright \cup (\to_\blacktriangleright \cdot \twoheadrightarrow)) \ a_{m+1}$. Since by construction a_n does not allow a $\to_\blacktriangleright$-step to an object b with $\infty(b)$, but $\infty(a_{m+1})$, in fact $a_n \to_\triangleright a_{m+1}$ must hold and we may continue with the pair of indices $n, m+1$.

Noting that we always eventually end up in the first case (otherwise the original sequence would contain an infinite $\to_\blacktriangleright$-reduction sequence, cf. $a_6 \to_\triangleright a_9$ in Figure 7), the process yields an infinite $\to_\triangleright$-reduction sequence (in Figure 7 the sequence $a_1, a_5, a_6, a_9, \ldots$), contradicting termination of $\to_\triangleright$. ◀

▶ **Remark.** The proof of Lemma 3.5 in fact shows a stronger property than preservation of termination of rewrite *relations*: it shows that any *object* allowing an infinite $\to$-reduction also allows an infinite $\to$-reduction having either an infinite $\to_\triangleright$-tail or an infinite $\to_\blacktriangleright$-tail.

▶ **Theorem 3.6.** *Triangulating with respect to a terminating relation preserves termination of codeterministic rewrite relations.*

Proof. Let $\to$ be the triangulation $\mathrm{Tr}(\to_\triangleright) = \bigcup_{n \geq 1} \to_n$ of the codeterministic terminating rewrite relation $\to_\triangleright$ and let $\to_\blacktriangleright = \bigcup_{n > 1} \to_n$ be the triangulating rewrite relation, which is terminating since it is contained in the terminating relation R. Since $\to \ = \ \to_\triangleright \cup \to_\blacktriangleright \ \subseteq \ \to_\triangleright^+ \cup \to_\blacktriangleright \ \subseteq \ \to^+$, to prove termination of $\to$ it suffices by Lemma 3.5 to prove $\to_\triangleright^+ \cdot \to_\blacktriangleright \ \subseteq \ \to_\triangleright^+ \cup (\to_\blacktriangleright \cdot \twoheadrightarrow)$. This is an immediate consequence of the following claim.

    Claim: $\to_\triangleright^+ \cdot \leftrightarrow_n \ \subseteq \ \to_\triangleright^+ \cup (\to_\blacktriangleright \cdot \twoheadrightarrow)$ for all $n > 1$.

We prove this claim by induction on n, for $n > 1$. Suppose $a \to_\triangleright^+ b \leftrightarrow_n c$ for some $n > 1$. We have to prove that either $a \to_\triangleright^+ c$ or $a \to_\blacktriangleright \cdot \twoheadrightarrow c$. By definition of triangulation $b \leftrightarrow_n c$ implies that there is an object d and there are natural numbers m, k both ≥ 1 and $< n$ such that $b \leftarrow_m d \to_k c$. We distinguish cases on whether or not $m = 1$.

- If $m = 1$, then $b \leftarrow_\triangleright d$ by definition of triangulation, hence $a \to_\triangleright\!\!\triangleright d$ by codeterminism of $\to_\triangleright$. If $a = d$ then we are done since then $a \to_\blacktriangleright c$. Otherwise we conclude by the induction hypothesis for $a \to_\triangleright^+ d \to_k c$.
- If $m > 1$, then by the induction hypothesis for $a \to_\triangleright^+ b \leftarrow_m d$ we conclude that $a \ (\to_\triangleright^+ \cup (\to_\blacktriangleright \cdot \twoheadrightarrow)) \ d$. If $a \to_\triangleright^+ d$ then we conclude by the induction hypothesis as in the previous item. Otherwise $a \to_\blacktriangleright \cdot \twoheadrightarrow d \to c$. ◀

▶ **Corollary 3.7.** *Triangulating a terminating codeterministic rewrite relation with respect to a terminating relation is a completion process.*
The corollary does not necessarily yield a *decision procedure* for deciding equivalence of terminating codeterministic rewrite relations, as the triangulation process may well be infinite itself, as it is the case for the rewrite relation in Figure 8.

    Acyclicity too is preserved by triangulation of codeterministic relations.

▶ **Corollary 3.8.** *Triangulating an acyclic codeterministic rewrite relation with respect to an acyclic relation, yields an acyclic relation.*
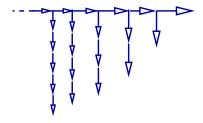
Figure 8 Rewrite relation requiring infinite triangulation

Proof. If a cycle were to exist in the triangulation, then this cycle would be generated from a finite part of the rewrite relation. On this finite part both the rewrite relation and the total relation with respect to which it is triangulated, are terminating by acyclicity, hence by Theorem 3.6 there can be no cycle. ◀

4   Completed

The previous section provided sufficient conditions on the original rewrite relation $\dashrightarrow$ and the relation R with respect to which it was triangulated, for the triangulation process to yield a complete rewrite relation. In this section, we reverse the process and look for conditions on rewrite relations $\dashrightarrow$, $\blacktriangleright$ that are sufficient for their union to be complete. Although one still may think of $\dashrightarrow$ as the original rewrite relation and $\blacktriangleright$ as the triangulating rewrite relation, the conditions provided will be abstract in that they only concern $\dashrightarrow$ and $\blacktriangleright$. In particular, the conditions refer neither to the triangulation process nor to the relation R used in it; they may (and will) of course refer to the characteristic properties the triangulation process brings about. Throughout $\rightarrow$ will be used to denote $\dashrightarrow \cup \blacktriangleright$.[6]

On the one hand, triangulation guarantees every peak to give rise to a triangle:

$$\leftarrow \cdot \rightarrow \,\subseteq\, \leftrightarrow^= \qquad\qquad\qquad \textbf{(affluence)}$$

On the other hand, it guarantees that every $\blacktriangleright$-step is caused by triangulation:

$$\blacktriangleright \,\subseteq\, \leftarrow \cdot \rightarrow \qquad\qquad\qquad \textbf{(triangulation)}$$

As these properties are characteristic of the triangulation process of Section 2, we will assume both throughout this section.

Since **(affluence)** states one-step affluence of $\rightarrow^=$ which in turn implies that $\rightarrow$ is confluent (Lemma 2.4), to obtain completeness it only remains to investigate which conditions on $\dashrightarrow$, $\blacktriangleright$ are sufficient to guarantee termination of $\rightarrow$. In the case that $\dashrightarrow$ is not codeterministic, one cannot do much more than crudely require that $\rightarrow$ itself is terminating. Of course, one may try to employ compatibility results such as Lemma 3.5, to reduce termination of $\rightarrow$ to that of its constituting relations $\dashrightarrow$, $\blacktriangleright$, but as the following example shows, employing that lemma is tied closely to the codeterministic case.

▶ **Example 4.1.** The union of $a \dashleftarrow b \dashrightarrow c \dashleftarrow d \dashrightarrow e$ and $c \blacktriangleright e \blacktriangleleft a \blacktriangleleft c$ is terminating, but Lemma 3.5 cannot be used to show this as $b \dashrightarrow a \blacktriangleright e$ but neither $b \dashrightarrow^+ e$ nor $b \blacktriangleright \cdot \twoheadrightarrow e$. Note that $\dashrightarrow$ is not codeterministic.

---

[6] Assuming $\dashrightarrow$ and $\blacktriangleright$ to be disjoint would not affect the results in this section.

Hence(forth), we restrict ourselves to the codeterministic case below. That is, we will assume the following property on top of the **(affluence)** and **(triangulation)** assumptions:

$$\rightarrowtriangle \cdot \mathbin{\reflectbox{$\rightarrowtriangle$}} \subseteq \mathsf{id}; \qquad\qquad\qquad\qquad\qquad\qquad \textbf{(codeterminism)}$$

4.1 Preservation of acyclicity by finiteness

We first show that for *finite* rewrite relations, as is the case for the motivating systems of definitions in the introduction, the above three conditions are already sufficient to guarantee completeness of $\rightarrow$, if $\rightarrowtriangle, \blacktriangleright$ are assumed terminating. After that we show this does not hold for infinite rewrite relations.

Noting that for finite rewrite relations termination coincides with acyclicity, our first result can be formulated as a preservation of acyclicity result.

▶ **Theorem 4.2.** *For finite rewrite relations* $\rightarrowtriangle, \blacktriangleright$ *and* $\rightarrow = \rightarrowtriangle \cup \blacktriangleright$, *if properties **(triangulation)** and **(codeterminism)** hold and* $\rightarrowtriangle, \blacktriangleright$ *are acyclic, then* $\rightarrow$ *is acyclic.*

Proof. Assume $\rightarrow$ admits a cycle. With a cycle we associate the multiset of elements on it, by which cycles can be compared by a well-founded order: the multiset order induced by $\blacktriangleright$. Now we take a $\rightarrow$-cycle that is minimal with respect to this multiset order, that is, there exists no cycle that is strictly smaller with respect to the multiset order.

As $\rightarrowtriangle$ and $\blacktriangleright$ are acyclic, our cycle contains both $\rightarrowtriangle$ and $\blacktriangleright$-steps. Moreover, it also contains a $\rightarrowtriangle$-step followed by a $\blacktriangleright$-step:

$$a \rightarrowtriangle b \blacktriangleright c \twoheadrightarrow a.$$

Now we choose c_1 such that $b \leftarrow c_1 \rightarrow c$, using property **(triangulation)**. As long as $b \blacktriangleleft c_i$ for increasing i we repeat the following: choose c_{i+1} such that $b \leftarrow c_{i+1} \rightarrow c_i$, again using property **(triangulation)**. Now one of the following two cases holds:

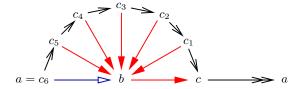▬ This process stops. Then for the last chosen c_i we have $c_i \rightarrowtriangle b$, see Figure 9. By

🟧 **Figure 9** Smaller cycle if erecting cotriangles stops

property **(codeterminism)** we conclude $a = c_i$. Since $c_j \blacktriangleright b$ for all j satisfying $1 \le j < i$, we conclude that the cycle via $\{c_j \mid 1 \le j < i\}$

$$a = c_i \twoheadrightarrow c \twoheadrightarrow a$$

is smaller than our original minimal cycle: the element b has been replaced by the (possibly empty) multiset $\{c_j \mid 1 \le j < i\}$, contradiction.

▬ This process goes on forever. Then by finiteness of the set we obtain $c_j = c_i$ for some $j > i$, yielding a new cycle $c_j \rightarrow c_{j-1} \twoheadrightarrow c_i = c_j$. Since $b \blacktriangleleft c_\kappa$ for all κ, all elements in this new cycle are less than the element b occurring in the original minimal cycle, again contradicting minimality.

As both cases contradict the assumption of the existence of a $\rightarrow$-cycle, we have proved that $\rightarrow$ is acyclic. ◄

▶ **Corollary 4.3.** *For finite rewrite relations* $\dashrightarrow$, $\blacktriangleright$ *and* $\rightarrow = \dashrightarrow \cup \blacktriangleright$, *if properties (affluence), (triangulation) and (codeterminism) hold and* $\dashrightarrow$, $\blacktriangleright$ *are terminating, then* $\rightarrow$ *is complete.*

Somewhat surprisingly, this result does not generalise to rewrite relations on infinite sets of objects as illustrated by the following counterexamples.

▶ Counterexamples 4.4. Take as objects the set of natural numbers and let
- $\dashrightarrow = \{(n+1,n) \mid n \geq 2\} \cup \{(2,0),(0,1)\}$; and
- $\blacktriangleright = \{(n,1) \mid n \geq 2\} \cup \{(1,0)\}$.

Then except for finiteness, all conditions of Corollary 4.3 are satisfied yet $\rightarrow$ is not terminating; it admits the cycle $0 \dashrightarrow 1 \blacktriangleright 0$ (see Figure 10 left).
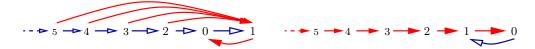

Figure 10 Loss of termination by infinite $\dashrightarrow$-expansion (left) and infinite $\blacktriangleright$-expansion (right)

For another example, take as objects the set of natural numbers and let
- $\dashrightarrow$ relate 0 to 1;
- $\blacktriangleright$ be the greater-than relation.

Again all conditions of Corollary 4.3 except for finiteness are satisfied, yet $\rightarrow$ is not terminating; it admits the cycle $0 \dashrightarrow 1 \blacktriangleright 0$ (see Figure 10 right where transitive $\blacktriangleright$-edges have been omitted).
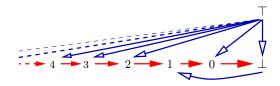
In the next two sections, we investigate how to regain the preservation result for (potentially) infinite rewrite relations, for two natural generalisation of acyclicity, termination respectively cotermination.

4.2 Preservation of termination by strong triangulation

To obtain completeness for infinite rewrite relations as well, we bar the counterexamples of the previous section by requiring, on top of the **(affluence)** and **(codeterminism)**, the following strengthening of **(triangulation)**

$$\blacktriangleright \subseteq ((\blacktriangleleft\!\!- \cdot \rightarrow) \cup (\leftarrow \cdot \dashrightarrow)) \cap (\blacktriangleleft\!\!\blacktriangleleft\!- \cdot ((\blacktriangleleft\!\!- \cdot \dashrightarrow) - \mathsf{id}) \cdot \dashrightarrow\!\!\vartriangleright) \qquad \textbf{(strong triangulation)}$$

The first conjunct of **(strong triangulation)** captures the idea that every $\blacktriangleright$-step is caused by a peak containing at least one $\dashrightarrow$-step, while the second conjunct captures that the $\blacktriangleright$-step originates with some non-trivial peak of $\dashrightarrow$-reductions. We proved that **(strong triangulation)** captures an essential aspect of triangulation in the sense that for any *codeterministic* rewrite relation its triangulation satisfies **(strong triangulation)**; the proof is found in the report version [7] of this paper.

▶ Remark. That **(strong triangulation)** is a proper strengthening of **(triangulation)** can be seen by considering the rewrite relations in Counterexamples 4.4. That the triangulation of a *non-codeterministic* rewrite relation may fail to satisfy **(strong triangulation)** can be seen by viewing $\blacktriangleright$ in Example 4.1 as arising from triangulating $\dashrightarrow$, and considering the step $a \blacktriangleright e$.

Figure 11 Non-terminating example satisfying all conditions of Corollary 4.6 except for the *first* conjunct of the **(strong triangulation)** property.

▶ **Theorem 4.5.** *For rewrite relations* $\rightarrowtriangle, \twoheadrightarrow$ *and* $\rightarrow \, = \, \rightarrowtriangle \cup \twoheadrightarrow$*, if properties **(strong triangulation)** and **(codeterminism)** hold and* $\rightarrowtriangle, \twoheadrightarrow$ *are terminating, then* $\rightarrow$ *is terminating.*

Proof. We proceed as in the proof of Theorem 3.6. That is, to prove termination of $\rightarrow$ under the assumption that $\rightarrowtriangle, \twoheadrightarrow$ are terminating, it suffices by Lemma 3.5 to prove $\rightarrowtriangle^{+} \cdot \twoheadrightarrow \, \subseteq \, \rightarrowtriangle^{+} \cup (\twoheadrightarrow \cdot \twoheadrightarrow\!\!\!\twoheadrightarrow)$. The latter property follows from the following claim.

    Claim: If $a \rightarrowtriangle^{+} b \leftarrowtriangle\!\!\twoheadrightarrow c$ then $a \, (\rightarrowtriangle^{+} \cup (\twoheadrightarrow \cdot \twoheadrightarrow\!\!\!\twoheadrightarrow)) \, c$, for all a, b, c.

We prove this claim by induction on $n + m$ for the natural numbers n, m such that it holds $b \, (\twoheadleftarrow^{n} \cdot ((\leftarrowtriangle \cdot \rightarrowtriangle) - \mathsf{id}) \cdot \rightarrowtriangle^{m}) \, c$. Note that the pair exists by the second conjunct of the **(strong triangulation)** property, and that it is unique by the **(codeterminism)** property and the assumption that $\rightarrowtriangle$ is terminating. From the **(strong triangulation)** property for $b \leftarrowtriangle\!\!\twoheadrightarrow c$ we conclude that there exists b' such that one of the following three cases holds.

- $b \leftarrowtriangle b' \rightarrowtriangle c$. Then by the **(codeterminism)** property, $a \rightarrowtriangle\!\!\!\rightarrowtriangle b' \rightarrowtriangle c$ and we conclude.
- $b \leftarrowtriangle b' \twoheadrightarrow c$. Then by the **(codeterminism)** property, $a \rightarrowtriangle\!\!\!\rightarrowtriangle b' \twoheadrightarrow c$. If $a = b'$ then we conclude. Otherwise we conclude by the induction hypothesis for $a \rightarrowtriangle^{+} b' \twoheadrightarrow c$, which applies since by the second conjunct of the **(strong triangulation)** property for $b' \twoheadrightarrow c$, for some pair n', m' of positive natural numbers $b' \, (\twoheadleftarrow^{n'} \cdot ((\leftarrowtriangle \cdot \rightarrowtriangle) - \mathsf{id}) \cdot \rightarrowtriangle^{m'}) \, c$, hence also $b \, (\twoheadleftarrow^{n'+1} \cdot ((\leftarrowtriangle \cdot \rightarrowtriangle) - \mathsf{id}) \cdot \rightarrowtriangle^{m'}) \, c$, so $n = n' + 1$ and $m = m'$ by uniqueness of the pair n, m;
- $b \twoheadleftarrow b' \rightarrowtriangle c$. Then $a \, (\rightarrowtriangle^{+} \cup (\twoheadrightarrow \cdot \twoheadrightarrow\!\!\!\twoheadrightarrow)) \, b'$ by the induction hypothesis for $a \rightarrowtriangle^{+} b \twoheadleftarrow b'$, which applies for the same reason as in the previous item. From that we conclude again. ◀

▶ **Corollary 4.6.** *For rewrite relations* $\rightarrowtriangle, \twoheadrightarrow$ *and* $\rightarrow \, = \, \rightarrowtriangle \cup \twoheadrightarrow$*, if properties **(strong triangulation)** and **(codeterminism)** hold and* $\rightarrowtriangle, \twoheadrightarrow$ *are terminating, then* $\rightarrow$ *is complete.* Neither conjunct of the **(strong triangulation)** property can be dispensed with.

▶ Counterexamples 4.7. The first example of Counterexamples 4.4 satisfies all conditions of Corollary 4.6 except for the *second* conjunct of the **(strong triangulation)** property, yet $\rightarrow$ is not complete; it is not terminating as we have seen before.

For an example satisfying all conditions of Corollary 4.6 except for the *first* conjunct of the **(strong triangulation)** property, consider the natural numbers extended with $\bot, \top$, where $\rightarrowtriangle$ is given by $\bot \rightarrowtriangle 1$, $\top \rightarrowtriangle \bot$, and $\top \rightarrowtriangle n$ for all $n \neq 1$, and where $\twoheadrightarrow$ is the greater-than relation on natural numbers extended with $0 \twoheadrightarrow \bot$ (see Figure 11 where transitive $\twoheadrightarrow$-edges have been omitted). That the conditions hold is easy to check, yet $\rightarrow$ is not complete; it is not terminating as it admits the cycle $1 \twoheadrightarrow 0 \twoheadrightarrow \bot \rightarrowtriangle 1$. Although the example does not satisfy the first conjunct $\twoheadrightarrow \, \subseteq \, (\leftarrowtriangle \cdot \rightarrow) \cup (\leftarrow \cdot \rightarrowtriangle)$ of the **(strong triangulation)** property for $1 \twoheadrightarrow 0$, it does satisfy the weaker condition $\twoheadrightarrow \, \subseteq \, \leftarrow \cdot \rightarrow$.

4.3 Preservation of cotermination by triangulation

There are two natural generalisations of acyclicity from finite to infinite rewrite relations, termination and cotermination. In the previous section we have seen that for termination to be preserved **(triangulation)** had to be strengthened to **(strong triangulation)**.[7] In this section we show that for cotermination no such strengthening is needed: it is preserved without extra conditions. That is, investigating the properties of the triangulation process, we answer the dual question whether triangulating a *coterminating* rewrite relation $\dashrightarrow$ with respect to a *coterminating* relation R yields a *coterminating* rewrite relation, affirmatively.

That the triangulation of $\dashrightarrow$ with respect to R is coterminating if their union $\dashrightarrow \cup R$ is coterminating, follows by the same proof as the one of Theorem 3.2. The interesting situation is again when $\dashrightarrow$ is assumed to be codeterministic, but no compatibility or finiteness condition is put on the coterminating relations $\dashrightarrow, R$.[8] We first show an auxiliary result, characterising the $\dashrightarrow$-reduction peaks that cause $\rightarrow$-steps.

▶ **Lemma 4.8** ($\triangle$). *For rewrite relations $\dashrightarrow$, $\twoheadrightarrow$ and $\rightarrow = \dashrightarrow \cup \twoheadrightarrow$, if properties **(triangulation)** and **(codeterminism)** hold and $\dashrightarrow, \twoheadrightarrow$ are coterminating, then for all $a_0 \rightarrow a_1$ a common $\dashrightarrow$-expansion $a^{0,1}$ of a_0, a_1 exists, such that $a_1 \twoheadleftarrow^+ b$, for $i \in \{0,1\}$ and all b with $a_i \dashleftarrow^+ b \dashleftarrow^+ a^{0,1}$ (see Figure 12).*
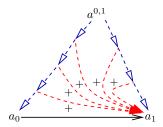

Figure 12 Illustration of the statement of the $\triangle$-lemma

Proof. The proof is by well-founded induction on the pair consisting of a_1 and the multiset of all $\dashrightarrow$-expansions of a_0, finite by **(codeterminism)** and cotermination of $\dashrightarrow$, well-foundedly ordered by the lexicographic product of $\twoheadleftarrow$ and its multiset extension. We distinguish cases on the step $a_0 \rightarrow a_1$.

($a_0 \twoheadrightarrow a_1$) We distinguish cases on the peak obtained by **(triangulation)**:
- if $a_0 \dashleftarrow a_2 \dashrightarrow a_1$, then, setting $a^{0,1} = a_2$, we conclude immediately;
- if $a_0 \dashleftarrow a_2 \twoheadrightarrow a_1$, then we conclude by the induction hypothesis for $a_2 \twoheadrightarrow a_1$, which applies by a decrease in the second component as the multiset of $\dashrightarrow$-expansions of a_2 is a proper submultiset of that of a_0, since $a_0 \dashleftarrow a_2$;
- if $a_0 \twoheadleftarrow a_2 \dashrightarrow a_1$, then we conclude by the induction hypothesis for $a_2 \twoheadrightarrow a_0$, which applies by a decrease in the first component, since $a_1 \twoheadleftarrow a_0$;
- if $a_0 \twoheadleftarrow a_2 \twoheadrightarrow a_1$ then the induction hypothesis applies first to $a_2 \twoheadrightarrow a_0$ yielding a common $\dashrightarrow$-expansion $a^{2,0}$ (decrease in the first component: $a_1 \twoheadleftarrow a_0$) and next to

---

[7] Counterexamples 4.4 showed that the **(triangulation)** property was *not* sufficient to show preservation of *termination* in the case of infinite rewrite relations.

[8] By Corollary 3.8 cotermination is seen to be preserved in the case of *finite* rewrite relations, since then termination, acyclicity, and cotermination all coincide.

$a_2 \twoheadrightarrow a_1$ yielding a common $\dashrightarrow$-expansion $a^{2,1}$ (decrease in the second component: the objects on the $\dashrightarrow$-expansion from a_2 up to $a^{2,0}$ all are $\twoheadrightarrow$-expansions of a_0 via a_2). By **(codeterminism)** $a^{2,1-j} \twoheadleftarrow\!\!\!\!\dashleftarrow a^{2,j}$ for one of $j \in \{0,1\}$, so we may set $a^{0,1} = a^{2,j}$ as common $\dashrightarrow$-expansion of a_0, a_1. If $a_i \dashleftarrow^+ b \dashleftarrow^+ a^{0,1}$ for some $i \in \{0,1\}$ and some b, then either $a_i \dashleftarrow^+ b \dashleftarrow^+ a^{2,i}$ or $a_2 \twoheadleftarrow\!\!\!\!\dashleftarrow a^{2,i} \twoheadleftarrow\!\!\!\!\dashleftarrow b \dashleftarrow^+ a^{0,1} = a^{2,j}$ and we conclude by either induction hypothesis and $a_1 \twoheadleftarrow a_0$;

$(a_0 \dashrightarrow a_1)$ Then we conclude, setting $a^{0,1} = a_0$, trivially. ◀

Nothing in the conditions of the Δ-lemma entails *completeness* of $\rightarrow$, and indeed it may fail to hold as can be seen by letting $\dashrightarrow$ be the (coterminating) predecessor relation on natural numbers and $\twoheadrightarrow$ the empty relation. We do however have *cocompleteness* (coconfluence and cotermination) even without requiring **(affluence)**.

▶ **Theorem 4.9.** *For rewrite relations* $\dashrightarrow$, $\twoheadrightarrow$ *and* $\rightarrow \,=\, \dashrightarrow \cup \twoheadrightarrow$, *if properties* **(codeterminism)** *and* **(triangulation)** *hold and* $\dashrightarrow$, $\twoheadrightarrow$ *are coterminating, then* $\rightarrow$ *is cocomplete.*

Proof. First we consider cotermination of $\rightarrow$. For a proof by contradiction, suppose $a_1 \leftarrow a_2 \leftarrow \ldots$ were an infinite expansion. Then adjoining an object a_0, and steps $a_{i+1} \twoheadrightarrow a_0$, for all i, would yield relations still satisfying the assumptions. The Δ-lemma yields a contradiction as a_0 and a_1 do not have a common $\dashrightarrow$-expansion.[9]

For the proof of coconfluence of $\rightarrow$, note that the Δ-lemma yields that $\rightarrow$-convertibility is contained in $\dashrightarrow$-convertibility, from which one concludes as $\dashrightarrow$ is trivially coconfluent by **(codeterminism)**. ◀

Although we think the proof of the result, in particular that of the Δ-lemma, is interesting, as of yet this is just a theoretical result and we do not have applications (except for providing an alternative more general/complex proof to Corollary 4.3). Still, the result lends itself well for a reformulation as a nice puzzle, i.e. one that is easy to understand but hard to solve.

▶ Puzzle 4.10. Consider a city with Red ($\twoheadrightarrow$) and Blue ($\dashrightarrow$) bus lines (see Figure 13 left):

- Blue buses are *deterministic*, i.e. the next stop of a Blue bus (if it can go anywhere at all) is completely determined by the stop it's currently at;
- Red buses can be *triangulated*, i.e. if a Red bus can go directly from stop a to stop b, then there is a stop c (not necessarily distinct from a,b) such that one can go directly from both a and b to c, in each case by either taking a Red or a Blue bus.

Show that if one can make an infinite trip using buses of either company, then one can make an infinite monochrome trip, i.e. a trip using buses of one and the same company only.[10]

To solve Puzzle 4.10 for the particular case displayed in Figure 13 it suffices to apply, the contrapositive of, the construction in the proof of the Δ-lemma (for the converse of the relations):

▶ Solution 4.11. Applying the construction to the bus route from top–right to top–left in the circuit on the left in Figure 13 leads to adjoining an infinite succession of triangles to $\twoheadrightarrow$-steps, as indicated by the numbers on the right in the figure, and thereby to an infinite monochrome trip, the Red trip indicated by the thick arrows.

A general solution to the puzzle requires considering arbitrary length bichrome trips instead of single hops, corresponding exactly to our result.

---

[9] For an alternative constructive proof, one may show, measuring objects by their multiset of $\dashrightarrow$-expansions, a decrease in the multiset extension of $\twoheadleftarrow$, along any step.

[10] So one can save lots of money by buying an ∞-*pass* from one of the companies only.
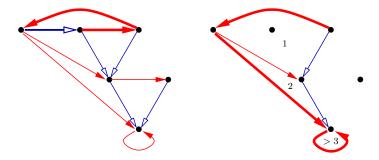
Figure 13 On the left Red and Blue bus lines satisfying the conditions of the puzzle are shown. On the right it is shown how to construct an infinite monochrome Red trip (the thick arrows).

5 Conclusion

In diagram completion in rewriting theory [6] the focus is on square diagrams. For instance, for proving confluence if a rewrites to both b and c we are looking for d such that both b and c rewrite to d, as depicted in a square diagram as is in the RTA logo. In contrast, in this paper not these squares are the basic building blocks, but the even more basic triangles: in the above setting we directly want to relate b and c. Inspired by computational geometry where splitting up polygons into triangles is called triangulation, we defined triangulation to be the process of extending a rewrite relation by extending every peak to a triangle. In this paper we studied some basic properties of triangulation, and its relation to confluence and termination. It turned out that the result of triangulation is always confluent (Theorem 2.5), even affluent, being the natural strengthening of confluence with respect to triangulation. We showed that in general triangulation does not preserve termination (Example 3.1), but that in case the initial rewrite relation is codeterministic termination is preserved (Theorem 3.6); the proof of this property was remarkably hard.

References

1. F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.

2. L. Bachmair and N. Dershowitz. Equational Inference, Canonical Proofs, and Proof Orderings. *Journal of the ACM*, 41(2):236–276, 1994.

3. N. Dershowitz. On Lazy Commutation. In *Languages: From Formal to Natural*, volume 5533 of *Lecture Notes in Computer Science*, pages 59–82. Springer, 2009.

4. H. Doornbos and B. von Karger. On the Union of Well-Founded Relations. *Logic Journal of the IGPL*, 6(2):195–201, 1998.

5. J.F. Groote, J. Keiren, A. Mathijssen, B. Ploeger, F. Stappers, C. Tankink, Y. Usenko, M. van Weerdenburg, W. Wesselink, T. Willemse, and J. van der Wulp. The mCRL2 toolset. In *Proceedings of the International Workshop on Advanced Software Development Tools and Techniques (WASDeTT 2008)*, 2008.

6. Terese. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.

7. H. Zantema and V. van Oostrom. Triangulation in rewriting. Logic Group Preprint Series 292, Utrecht University, 2012. `http://www.phil.uu.nl/preprints/lgps/`.