

Laziness and Lifting*

Nachum Dershowitz

School of Computer Science, Tel Aviv University, Ramat Aviv, Israel
nachum.dershowitz@cs.tau.ac.il

Abstract. A new commutation property – related to “lazy” commutation, and a weakened version of the “lifting” property – requiring only an *eventual* lifting, can be used for proving well-foundedness of unions of binary relations and generic path orderings.

Keywords: path orderings, well-founded orderings, well quasi-orderings, termination, commutation properties

*At the foundation of well-founded beliefs
lies belief that is not well-founded.*

– Ludwig Wittgenstein, *On Certainty*, §253

1 Motivation

The various path orderings, like the recursive path ordering [6] and its higher-order versions [3], provide a convenient and popular method of proving termination, particularly of term-rewriting systems. Here, we set out to prove the well-foundedness of a very abstract “path” ordering that includes the usual path orderings on first-order terms as special cases. Abstract versions of the path orderings have been proposed in [15,4]. We aim for a more general version yet, and we develop new proof techniques for this purpose.

Suppose we have a set V (think terms, if you wish) that is equipped with a well-founded ordering \triangleright that provides some structure on V (think subterms). Suppose further that we have some kind of “surface” relation \gg on V (think precedences, lexicographically ordered tuples, and the like). Let’s call t' a *descendant* of V if $t \triangleright t'$ and say that u' *follows* u if $u \gg u'$.

We will define t to be bigger than u in the (*abstract*) *path ordering*, written here as $t \blacktriangleright u$ for $t, u \in V$, if either of two cases holds, in a “two-dimensional” recursive way:

- (a) If t has a descendant that is bigger than u , then t is also bigger than u .
- (b) If t is followed by u and, moreover, all its descendants are either bigger than u or themselves have a descendant that is no smaller than u , then t is also bigger than u .

* An earlier version of this work was presented at the last International Workshop on Termination.

In Sect. 3, we give a more general definition. For this relation – it need not be a partial ordering – to be a useful one, we need to connect the surface, “followed-by” relation \gg with the “bigger-than” relation \blacktriangleright on \triangleright -descendants, as we will see.

The followed-by relation can be something like a multiset ordering of unordered tuples [10] – as in the multiset path ordering [5], or a lexicographic comparison of ordered tuples – as in the lexicographic path ordering [18]. So, in the former case, an ordering on elements is lifted to one on multisets; in the latter, it is lifted to tuples. But there is a noticeable difference between the two cases. Given an infinite descending sequence in the multiset ordering, one can be sure that there is an infinite descending sequence of elements of the underlying set that starts from some element of the first multiset. In the lexicographic case, on the other hand, one only knows that some tuple along the way has some component that initiates an infinite descending sequence of elements of the underlying set. This distinction leads us to investigate, in Sect. 6, sequences with a property that we call “escaping”, as opposed to the simpler “lifting” property, which suffices for multisets.

To this end, we employ the method of constriction, due originally to Plaisted [25], as explained in Sect. 5. We also need to consider, in Sect. 5, various commutation properties, generalizing those of [8]. Commutation is applicable to sequences mixing two kinds of steps, such as we have here.

These various tools are combined to prove that the abstract version of the path orderings given in detail in Sect. 3 is strongly normalizing (terminating). Some basic notions are first explained in the next section. And a hint of future plans concludes this paper.

In addition, in the Appendix, we propose some convenient notations for properties of sequences and illustrate their use with some properties of well quasi-orderings. See [22]; compare [15].

2 Mortality

We require methods of showing that, under appropriate conditions, the directed graph of the abstract path ordering lacks infinite paths.

2.1 Relations

Let V be some finite or infinite set of elements, called *points*. Let Σ be a set of colors (labels) and $U = V \times \Sigma \times V$ colored *steps* (pairs of points). Think vertices of a finite or infinite graph for V and all possible labelled edges U between them. Any binary relation R over points V can be viewed as steps colored R .

In what follows, the underlying set V is fixed, with colored relations A , B , R , S , etc.

2.2 Notation

We denote composition of binary relations by “.” or juxtaposition, union by “+”, and the reflexive, transitive, and reflexive-transitive closures of binary relation R by R^ε , R^+ , and R^* , respectively. The n -fold composition of R is R^n . Let \emptyset be the empty relation and 1 , the identity relation. So, as usual, $R^0 = \emptyset$, $R^1 = R$, $1 \cdot R = R \cdot 1 = R$, for any $R \subseteq \Omega$.

If P is a property of points, that is, if $P \subseteq V$, then we will use $P?$ to denote the restriction of the identity relation to P only.

For binary relations A and B , the following division-like notation will also be quite convenient:

$$B/A = \{\langle x, y \rangle : \forall z. yAz \Rightarrow xBz\}.$$

In other words, $x(B/A)y$ means that xBz for all z such that yAz . It follows that

$$\begin{aligned} B/1 &= B \\ B/\emptyset &= \Omega \\ B/B &\supseteq 1 \\ (B/A) \cdot A &\subseteq B. \end{aligned}$$

In the Appendix, we develop a more complete system of notation for sequences, their operations, and their properties.

2.3 Immortality

Let R be a colored relation over V , that is, a subset of U .

The *immortal* elements of a set V equipped with a relation R are those $t \in V$ that initiate infinite R -chains of (not necessarily distinct) elements of V ,

$$t R t' R t'' R \dots$$

We will say that a step $t R t'$ is *immortalizing* if t' is immortal. Define

$$R^\infty = \{\langle u, v \rangle \mid u, v \in V, u \text{ is immortal for } R\},$$

relating immortal elements to all vertices, as is commonly done in relational program semantics (e.g. [11]). Thus,

$$R^\infty C \subseteq R^\infty,$$

for any relation C .

We say that a relation R is *terminating* or *well-founded* (regardless of whether R is transitive) if it admits no cases of immortality, that is, if $R^\infty = \emptyset$. The domain (first projection) of the relation AB^∞ are those elements with an A -descendant that is immortal for B .

We will also use R_∞ to represent the set (monadic relation) of immortal elements with respect to R . Thus, to say that R is terminating, or – what is equivalent – that its transitive closure is well-founded, one can write

$$R_\infty = \emptyset.$$

In the graph



A is dashed **Azure** and B is solid **Bordeaux**. The vertex s is immortal in A and mortal in B , while t and u are mortal in both. In the following



every vertex is mortal in each of the two colored relations, but immortal in their union.

3 Abstract Path Ordering

What distinguishes the various path orderings in the literature is the use of a surface ordering, like a precedence or numeric interpretation, together with a recursive comparison of subterms. More abstractly, we have the following:

Definition 1 (Abstract Path Ordering). *The (abstract) path ordering is a relation \blacktriangleright (not necessarily transitive) on some set V , parameterized by two other abstract relations, \gg and well-founded \triangleright , and by conditions C and D . It is defined as:*

$$t \blacktriangleright u \quad \text{if} \quad \begin{cases} t \succ s \text{ and } t C s & \text{(a)} \\ \text{or} & \\ t \gg s, t (\succ + \blacktriangleright) / \triangleright s, \text{ and } t D s, & \text{(b)} \end{cases}$$

where \succ is short for $\triangleright^+ \blacktriangleright^*$.

This is a generalization of the abstract ordering given in [15].

Strictly speaking, this abstract path ordering need not be an ordering, as it can be non-transitive.

The purpose of C and D is just to optionally constrain the two cases. So, the definition means that

$$\triangleright^+ \blacktriangleright^* \subseteq \blacktriangleright$$

and

$$\gg \cap (\triangleright^+ \blacktriangleright^* + \blacktriangleright) / \triangleright \subseteq \blacktriangleright.$$

The version described in the first section has \triangleright transitive and C and D universal. In other words,

$$\triangleright = \triangleright^* \cup [\gg \cap (\triangleright^* + \triangleright) / \triangleright].$$

4 Constriction

The point of “constriction” is to consider the impact on immortality of a color preference for edges. This is very useful in arguments for well-foundedness, as we will see.

Definition 2 (Constricting Chains [25]). *An infinite sequence of A - and B -steps is constricting in A if from every point in the sequence where there is a B -step every possible alternate A -step would yield a mortal element.*

The use of constriction was first suggested by Plaisted [25] for rewriting and by Sørensen [26] for the lambda calculus. See also [16,21].

Let

$$B_{\#} = B \setminus (A(A+B)^{\infty})?B.$$

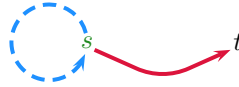
That is a B -step from which no A -step leads to immortality. These are the constricting B -steps, which is the only kind of B -step in a constricting sequence.

One can always build a constricting sequence for an immortal element. The idea is to ignore B -steps whenever an A -step is available.

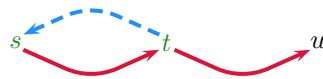
Theorem 1 ([25]). *For any binary relations A and B , if s is immortal in $A+B$, then there is sequence constricting in A , beginning with s .*

Proof. Just take an A -step whenever possible, that is, whenever there is one that leads to immortality. Only take B if there is no choice, so the B -steps are constricting. \square

If there is an infinite A -path, then preferring A can be a bad idea. Witness:



Even if the sole infinite path involves both relations, A 's can preclude reaching the goal, as in



5 Commutation Properties

Since the abstract path ordering is defined in terms of two other orderings, following (\gg) and descending (\triangleright), we will need to reason about reorderings of sequences of steps. That’s where commutation, some versions of which were investigated in [2], will come into play.

5.1 Selection

Definition 3 (Selection [8, Def. 42]). *Binary relation B selects binary relation A if*

$$B \cdot A^+ \subseteq A \cdot (A + B)^* + B^+.$$

Selection insists that if one can get from an element s to an element V by a B -step followed by one or more A -steps, then one can also get from s to V by first taking an A -step and then some combination of A - and B -steps, or else one can get there by one or more B -steps alone.

This is a weaker requirement than the following “local” condition explored in [11,12] and called “lazy commutation” in [8],¹ which, in turn, is much weaker than the “commuting over” condition in [2].

Definition 4 (Laziness [11]). *Binary relation B commutes lazily over binary relation A if*

$$B \cdot A \subseteq A \cdot (A + B)^* + B.$$

Proposition 1. *If binary relation B commutes lazily over binary relation A , then B selects A .*

Theorem 2. *If binary relation B selects binary relation A , then*

$$(A + B)_\infty = A^* \cdot B^* \cdot (A_\infty + B_\infty).$$

This is meant to convey that one can get from any element that is immortal in the union $A + B$ to an element that is immortal in one of the two component relations by taking some number of A -steps followed by some number of B -steps. This implies, of course, that the union is well-founded whenever both A and B are.

*Proof.*² Consider an infinite constricting sequence, per Theorem 1. If the sequence has only one kind of step from some point on, then the mixed prefix up to that point can be rearranged as A^*B^* , unless the attempted rearranging produces an infinite sequence of A -steps. Otherwise, it includes infinitely many A s and infinitely many B s and looks like $A^* \cdot (B \cdot A^*)^\infty$. By selection, that is A^* followed by $(A \cdot (A + B)^* + B^+)^\infty$. But, by constriction, that can only be $(B^+)^\infty$. \square

¹ In [8, Def. 61], we also defined a weak form of laziness, “lackadaisical commutation”, which is somewhat similar to the property in [24].

² The proof of this theorem given in [8, Theorem 72] relied on an overly broad formulation of Theorem 1. I am grateful to Ori Brostovski for pointing this out.

In particular,

Corollary 1 ([11]). *If well-founded binary relation B commutes lazily over well-founded binary relation A , then their union $A + B$ is also well-founded.*

Proposition 2. *For tricolored graphs, with steps colored A, B, C , each of which is well-founded, the following commutation conditions suffice for well-foundedness of $A + B + C$:*

$$\begin{aligned} B \cdot A &\subseteq A \cdot (A + B)^* + B + C \\ C \cdot A &\subseteq A \cdot (A + B)^* + B \cdot (A + B)^* + C \\ C \cdot B &\subseteq A \cdot (A + B)^* + B \cdot (A + B)^* + C. \end{aligned}$$

The argument is also by constriction.

5.2 Jumping

When A is transitive, as it will be in the cases of interest here (where A is some sort of subterm relation), selection is the same as the following local condition:

Definition 5 (Jumping). *Binary relation A jumps over binary relation B if*

$$B \cdot A \subseteq A \cdot (A + B)^* + B^+.$$

This is noticeably weaker than lazy commutation [11,12,8], which allows only one B rather than B^+ .

As a corollary of Theorem 2, we have the following:

Corollary 2. *If a transitive binary relation A jumps over a binary relation B , then*

$$(A + B)_\infty = A^\varepsilon B^* (A_\infty + B_\infty).$$

This, too, implies “separation” of termination of the union $A + B$.

6 Lifting and Escaping

We need to relate the infinite behaviors of mixed relation $A + B$ and pure B .

Definition 6 (Lifting). *Binary relation A lifts to binary relation B if*

$$B_\infty \subseteq A \cdot (A + B)_\infty.$$

In general,

Theorem 3 ([8, Lemma 84]). *If binary relation A lifts to binary relation B and B selects A , then*

$$(A + B)_\infty = (A + B)^* \cdot A_\infty.$$

It turns out, however, that oftentimes we need a weaker alternative to lifting, in which the A -step need only take place *eventually*. This is captured by the next definition.

Definition 7 (Escaping). *Binary relation A escapes from binary relation B if there is a point in every infinite B -chain at which an A -step would lead to an immortal element in the union $A + B$.*

Using the proposed notation of the appendix, escaping may be expressed as

$$B^\infty \models \diamond \llbracket A(A+B)^\infty \rrbracket.$$

Here, B^∞ is being used to denote the set of all infinite B -chains and the \diamond modality means that some suffix looks like $\llbracket A(A+B)^\infty \rrbracket B^\infty$. The double-bracket notation turns the set (of sequences) $A(A+B)^\infty$ into the relation between those elements having immortal descendants and everything.³

Theorem 4. *If binary relation A escapes from binary relation B and B commutes lazily over A , then*

$$B_\infty \subseteq (A+B)^* \cdot A_\infty.$$

So, under these conditions, B is well-founded as long as A is.

Proof. Let $E = A + B$. On account of laziness and Theorem 2, we know that

$$E^\infty \subseteq E^* \cdot A^\infty + A^* \cdot B^\infty.$$

Also, by laziness, we know that for all constricting B_\sharp -steps appearing in an infinite sequence of A - and B -steps, we may replace $B_\sharp A$, and – by induction – $B_\sharp A^*$, by just B , which is actually B_\sharp .

Consider, now, an infinite B -chain, but take the first A -escape possible. That looks like $B_\sharp^* A E^\infty$, which expands to $E^* A^\infty$ or $B_\sharp^* A^+ B^\infty$. In the second case, we repeat the expansion. In the final analysis, we either get $E^* A^\infty$, as desired, or else $(B_\sharp^* A^+)^\infty$. But $(B_\sharp^* A^+)^\infty = (A^+)^\infty + (B_\sharp^* B_\sharp A^+)^\infty$. As noted, $B_\sharp A^+$ may be replaced by B_\sharp . So, $(B_\sharp^* A^+)^\infty = (A^+)^\infty + (B_\sharp^+)^\infty = A^\infty + B_\sharp^\infty$. Since B_\sharp^∞ is precluded by the escaping condition, the result follows. \square

7 Well-foundedness

In this section, we establish conditions under which the (abstract) path ordering (Def. 1) is well-founded and can be used in termination proofs.

Let \sqsupset be short for $\gg \cap (\succ + \blacktriangleright) / \triangleright$, as appears in case (b) of the definition of the path ordering. Bear in mind, then, $\sqsupset \subseteq \gg$.

By the cases of the definition, we have the following.

³ Escaping is somewhat reminiscent of the ‘‘bar induction’’ criterion in [15].

Lemma 1.

$$\blacktriangleright \subseteq \gamma + \sqsupset .$$

Also, the recursive definition must expand, so:

Lemma 2.

$$\blacktriangleright \subseteq (\triangleright + \sqsupset)^+ .$$

Lemma 3. *For the path ordering, relation \triangleright commutes lazily over \sqsupset .*

Proof. By the terms of the second case (b), one has

$$\sqsupset \cdot \triangleright \subseteq \gamma + \blacktriangleright .$$

Using the above lemmata, and pasting the various facts together, we get

$$\sqsupset \cdot \triangleright \subseteq \triangleright^+ \cdot \blacktriangleright^* + \blacktriangleright \subseteq \triangleright^+ \cdot \blacktriangleright^* + \triangleright^+ \cdot \blacktriangleright^* + \sqsupset \subseteq \triangleright \cdot (\triangleright + \sqsupset)^* + \sqsupset .$$

So, in fact, \triangleright commutes lazily over \sqsupset , which implies selection. \square

It follows from Theorem 2 that \blacktriangleright is well-founded if \sqsupset is. Of course, \sqsupset is well-founded if \gg is. So:

Proposition 3. *A path ordering is well-founded whenever the “followed by” relation (\gg) is.*

This works, as is, for some interpretation-based termination orderings. The problem is that, for path orderings, \gg (following) is normally defined, recursively, in terms of \blacktriangleright applied to subterms.

Theorem 5. *A path ordering \blacktriangleright defined on a set V is well-founded if \blacktriangleright -mortality of all the \triangleright -descendants of an element $s \in V$ implies well-foundedness of \sqsupset (or of \gg) for s .*

Proof. We are presuming that \triangleright is well-founded. Let s be a \triangleright -minimal element of V that is immortal in \blacktriangleright . By Theorem 2 (with \triangleright for A and \sqsupset for B) and Lemma 3, there must be an $s' \triangleleft^* s$ that is immortal in \sqsupset (and, therefore, in \gg). By the assumption of this theorem, there must be a descendant $s'' \triangleleft s'$ that is immortal in \blacktriangleright , contradicting minimality of s . \square

As a corollary of Theorem 3 and Lemma 3, we have

Corollary 3. *A path ordering \blacktriangleright is well-founded if \triangleright lifts to \sqsupset .*

A minori ad maius if it lifts to \gg .

This applies to the nested multiset ordering [10], where \gg is the multiset ordering, and to lexicographic orderings. The general case of such “lifted” definitions was first studied in [18] and was pursued further in [14,15].

Presuming well-foundedness of \triangleright , and using Theorem 4, the inescapable conclusion is that

Theorem 6. *A path ordering \blacktriangleright is well-founded if \triangleright escapes from \sqsubset .*

The multiset path ordering [5], lexicographic path ordering [18], and recursive path ordering [20,6] are all special cases, where \triangleright is the proper subterm relation (so, $\triangleright^+ = \triangleright$), C and D are always true, and \gg is a recursive lifting of \blacktriangleright to multisets, precedence (first) and (then) multisets, precedence and tuples lexicographically, and a mixture thereof, respectively.

8 Discussion

We can apply this commutation method to analyze the dependency-pair method of proving termination. (See [1]; compare [7].) We also hope to analyze minimal bad sequence arguments for well-quasi-orderings in a similar fashion. (See [22]; compare [15].)

We are optimistic that the commutation-based approach taken here will likewise help for advanced path orderings, like the general path ordering [9] and higher-order recursive-path-ordering [13,17,3], without recourse to reducibility/computability predicates, because (as pointed out in [7]) there is an analogy between the use of reducibility predicates and the use in proofs of well-foundedness of the “constricting” derivations used in the proof of Theorem 2 cited above.

Acknowledgements. Thank you, Ori Brostovski and Bernhard Gramlich, for your suggestions.

References

1. Thomas Arts and Jürgen Giesl, 2000, “Termination of term rewriting using dependency pairs”, *Theoretical Computer Science*, vol. 236, pp. 133–178.
2. Leo Bachmair and Nachum Dershowitz, July 1986, “Commutation, transformation, and termination”, *Proc. of the Eighth International Conference on Automated Deduction (CADE)*, Oxford, England, J. H. Siekmann, ed., Lecture Notes in Computer Science, vol. 230, Springer-Verlag, Berlin, pp. 5–20.
3. Frédéric Blanqui, Jean-Pierre Jouannaud, and Albert Rubio, Oct. 2007, “HORPO with computability closure: A reconstruction”, *Proceedings of the 14th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2007)*, Yerevan, Armenia, N. Dershowitz and A. Voronkov, eds., Lecture Notes in Computer Science, vol. 4790, Springer-Verlag, Berlin, pp. 138–150.
4. Jeremy E. Dawson and Rajeev Gore, 2007, “Termination of abstract reduction systems”, *Proceedings of Computing: The Australasian Theory Symposium (CATS 2007)*, Ballarat, Australia, pp. 35–43.
5. Nachum Dershowitz, Mar. 1982, “Orderings for term-rewriting systems”, *Theoretical Computer Science*, vol. 17, no. 3, pp. 279–301.
6. Nachum Dershowitz, 1987, “Termination of rewriting”, *J. of Symbolic Computation*, vol. 3, pp. 69–116.

7. Nachum Dershowitz, Sept. 2004, “Termination by abstraction”, *Proceedings of the Twentieth International Conference on Logic Programming (ICLP)*, St. Malo, France, Lecture Notes in Computer Science, vol. 3132, Springer-Verlag, Berlin, pp. 1–18. Available at <http://www.cs.tau.ac.il/~nachum/papers/TerminationByAbstraction.pdf>.
8. Nachum Dershowitz, 2009, “On lazy commutation”, *Languages: From Formal to Natural*, O. Grumberg, M. Kaminski, S. Katz, and S. Wintner, eds., Lecture Notes in Computer Science, vol. 5533, Springer-Verlag, Berlin, pp. 59–82. Available at <http://www.cs.tau.ac.il/~nachum/papers/LazyCommutation.pdf>.
9. Nachum Dershowitz and Charles Hoot, May 1995, “Natural termination”, *Theoretical Computer Science*, vol. 142, no. 2, pp. 179–207. Available at <http://www.cs.tau.ac.il/~nachum/papers/natural-sterm94.pdf>.
10. Nachum Dershowitz and Zohar Manna, Aug. 1979, “Proving termination with multiset orderings”, *Communications of the ACM*, vol. 22, no. 8, pp. 465–476.
11. Henk Doornbos, Roland Backhouse, and Jaap van der Woude, 1997, “A calculational approach to mathematical induction”, *Theoretical Computer Science*, vol. 179, pp. 103–135.
12. Henk Doornbos and Burghard von Karger, 1998, “On the union of well-founded relations”, *Logic Journal of the IGPL*, vol. 6, no. 2, pp. 195–201.
13. Maribel Fernández and Jean-Pierre Jouannaud, 1994, “Modular termination of term rewriting systems revisited”, in E. Astesiano, G. Reggio, and A. Tarlecki, eds., *Recent Trends in Data Type Specification*, Lecture Notes in Computer Science, vol. 906, Springer-Verlag, Berlin, pp. 255–272.
14. Maria C. F. Ferreira and Hans Zantema, 1995, “Well-foundedness of term orderings”, in: N. Dershowitz, ed., *Proceedings of the 4th International Workshop on Conditional Term Rewriting Systems (CTRS '94)*, Lecture Notes in Computer Science, vol. 968, Springer-Verlag, Berlin, pp. 106–123.
15. Jean Goubault-Larrecq, Sept. 2001, “Well-founded recursive relations”, *Proceedings of the 15th International Workshop on Computer Science Logic (CSL '01)*, Paris, France, Lecture Notes in Computer Science, vol. 2142, Springer-Verlag, Berlin, pp. 484–497.
16. Bernhard Gramlich, 1996, *Termination and Confluence Properties of Structured Rewrite Systems*. Ph.D. thesis, Universität Kaiserslautern.
17. Jean-Pierre Jouannaud and Albert Rubio, 1999, “Higher-order recursive path orderings”, *Proceedings 14th Annual IEEE Symposium on Logic in Computer Science (LICS)*, Trento, Italy, pp. 402–411.
18. Sam Kamin and Jean-Jacques Lévy, February 1980, “Two generalizations of the recursive path ordering”, unpublished note, Department of Computer Science, University of Illinois, Urbana, IL. Available at <http://pauillac.inria.fr/~levy/pubs/80kamin.pdf>.
19. Claude Kirchner and Florent Kirchner and Hélène Kirchner, June 2010, “Constraint Based Strategies”, *Proceedings of the 18th International Workshop on Functional and (Constraint) Logic Programming (WFLP 2009, Brasilia, Brazil)*, Escobar, S., ed., Lecture Notes in Computer Science, vol. 5979, Springer Verlag, Berlin, pp. 13–26.
20. Pierre Lescanne, 1990, “On the recursive decomposition ordering with lexicographical status and other related orderings”, *J. Automated Reasoning*, vol. 6, no. 1, pp. 39–49.
21. Paul-André Mellès, 1996, *Description Abstraite des Systèmes de Réécriture*, Thèse de doctorat, Université Paris.

22. Paul-André Melliès, 1998, “On a duality between Kruskal and Dershowitz theorems”, *Proceedings of the 25th International Conference on Automata, Languages and Programming (ICALP)*, Lecture Notes in Computer Science, vol. 1443, Springer-Verlag, Berlin, pp. 518–529.
23. Crispin St. J. A. Nash-Williams, 1963, “On well-quasi-ordering finite trees”, *Proceedings Cambridge Phil. Soc.*, vol. 59, pp. 833–835.
24. Vincent van Oostrum, Apr. 2011, “Preponement”. Unpublished draft. Available at <http://www.phil.uu.nl/~oostrom/publication/pdf/preponement.pdf>.
25. David A. Plaisted, 1993, “Polynomial Time Termination and Constraint Satisfaction Tests”, *Proceedings of the 5th Intl. Conference on Rewriting Techniques and Applications (RTA)*, Lecture Notes in Computer Science, vol. 690, Springer-Verlag, pp. 405–420.
26. Morten Heine Sørensen. “Properties of infinite reduction paths in untyped λ -calculus”, *Proceedings of the Tbilisi Symposium on Language, Logic, and Computation*, CLSI Lecture Notes, 1996.

A Properties of Sequences

We have been dealing with sets of sequences. In this appendix, we propose some notational conveniences for reasoning about such situations.

Let’s use the term *gene* for a sequence, be it finite or infinite, and let’s use *genome* for a set of genes of unrestricted lengths. We need to devise appropriate notations for characterizing finite or infinite length genomes.

We will use lower-case for individual elements and for genes, and UPPER-CASE for relations and genomes. And we will use **boldface** for genes and for genomes to distinguish them from tuples and relations, respectively.

A.1 Genes

A path in the complete graph $\Omega = V \times V$ corresponds to a *sequence* of zero or more steps, connecting one or more points. (We have no need for empty sequences, devoid of points.)

For any binary relation $R \subseteq \Omega$, we use (boldface) \mathbf{R}^n for the set of genes with exactly n R -steps and, consequently, $n + 1$ points; \mathbf{R}^* for all finite genes; \mathbf{R}^∞ for infinite genes; and \mathbf{R}^\sim for all genes, finite or infinite. In particular, \mathbf{R} ($= \mathbf{R}^1$) is the binary relation itself, viewed as a set of one-step, two-point genes, and $\mathbf{\Omega}$ is the genome containing all possible two-point genes.

Just like a binary relation is a set of single steps, an n -ary *relation* is a set of genes (paths) of length $n - 1$. At the low end, if P is a (unary) property of points, then \mathbf{P} is also the set of one-element, zero-length genes, one for each $p \in P$. A single point $v \in V$ may be viewed also as a single-item gene.

The length $|\mathbf{r}|$ of a gene \mathbf{r} is the number of its steps, or ∞ if there are infinitely many. (We do not deal with longer transfinite sequences.) We index the elements of a gene $\mathbf{r} \in \mathbf{R}^\sim$ from 1, as in $\mathbf{r} = r_1 r_2 \dots$. We are interested in the points in sequences, not in the labels of the steps, per se.

It will be convenient to imagine a fictitious “wildcard” element, which we will denote $\mathbf{1}$. For infinite genes, let $\mathbf{s}_{|\mathbf{s}|} = \mathbf{s}_\infty = \mathbf{1}$ by fiat. By the same token,

we now have a special gene, consisting just of the single wildcard element, and also denoted $\mathbf{1}$.

A.2 Joins

Suppose A and B are binary relations, colorings (labels) of edges in the graph. Rather than viewing their juxtaposition AB as the binary relation $A \cdot B$, obtained by composing A and B (namely, $\{\langle x, z \rangle : \exists y. (xAyBz)\}$), or the 4-letter strings obtained by concatenation, viewed as the ternary relation obtained by joining an A -step with a B -step, that is, $AB = \{\langle x, y, z \rangle : xAyBz\}$.

More generally, the *join* of two genes \mathbf{r}, \mathbf{s} (not mere concatenation) is defined as follows:

$$\mathbf{rs} = \begin{cases} r_1 r_2 \cdots r_{|\mathbf{r}|} s_2 s_3 \cdots & |\mathbf{r}| < \infty, r_{|\mathbf{r}|} = s_1 \\ \mathbf{r} & |\mathbf{r}| = \infty. \end{cases}$$

Compare the notion of composition of derivations in [19]. If, however, $r_{|\mathbf{r}|} \neq s_1$, then \mathbf{rs} is nonexistent, a “non-gene”. The join of an infinite gene \mathbf{r} with anything is just \mathbf{r} , because we are uninterested in transfinite sequences of length beyond ω . This join operation is associative, since $\mathbf{rs} = \mathbf{r}$, even if \mathbf{s} is a non-gene, as long as \mathbf{r} is infinite.

The wildcard gene acts like a multiplicative unit, in that $\mathbf{1r} = \mathbf{r1} = \mathbf{r}$ for all \mathbf{r} . And we can use exponentiation for repeated joins:

$$\mathbf{r}^0 = \mathbf{1}, \quad \mathbf{r}^1 = \mathbf{r}, \quad \mathbf{r}^2 = \mathbf{rr},$$

etc. Also, $\mathbf{r}^\infty = \mathbf{rrr} \cdots$. In general, $\mathbf{r}^m \mathbf{r}^n = \mathbf{r}^{m+n}$. For a gene \mathbf{r} of length 1, $\mathbf{rr} = \mathbf{r}$, as well as $\mathbf{r}^\infty = \mathbf{r}$.

A.3 Genomes

More generally, let $\mathbf{R}, \mathbf{S} \subseteq \Omega^\sim$ be genomes over V . Define their *join* (in the sense of database theory) as follows:

$$\mathbf{RS} = \{\mathbf{rs} : \mathbf{r} \in \mathbf{R}, \mathbf{s} \in \mathbf{S}\},$$

that is, an R -gene glued to an S -gene at their matching extreme points. Joining of genomes is associative.

The empty set \emptyset acts as a zero element: $\emptyset \mathbf{S} = \mathbf{S} \emptyset = \emptyset$, for all \mathbf{S} . (So, we have a monoid with a zero.)

In analogy with regular expressions for strings, we avoid set formers for singletons. So $p\mathbf{R}$, for point p and relation \mathbf{R} , denotes those genes in \mathbf{R} that begin with p .

Let $\mathbf{1} = \{\mathbf{1}\}$ be the set containing just the wildcard gene. This relation acts as a unit: $\mathbf{1S} = \mathbf{S1} = \mathbf{S}$, for all \mathbf{S} . One can identify $\mathbf{1}$ with the full set of singleton (single-point) genes, V , since it too acts as an identity element.

Sets of singleton genes are monadic predicates, and act as filters. In particular, $PV = VP = PP = P$ for all $P \subseteq V$. The binary relation $P\Omega$ is the set of steps whose source is in P , while ΩP are edges whose endpoint is in P . For predicate P and genome \mathbf{R} , $P\mathbf{R}$ is the set of those genes in \mathbf{R} that begin with a point in P , while $\mathbf{R}P$ are those \mathbf{R} -genes that end with a point in P , or are infinite.

Exponentiating works as expected: $\mathbf{R}^0 = \mathbf{1}$, $\mathbf{R}^1 = \mathbf{R}$, $\mathbf{R}^2 = \mathbf{R}\mathbf{R}$, etc., for any genome $\mathbf{R} \subseteq \Omega^\sim$. So, \mathbf{R}^i , for $i \in \mathbb{N}$, is the set of i -fold joins of paths in \mathbf{R} . Define also the following:

- \mathbf{R}^- contains all finite genes in \mathbf{R} in reverse order.
- $\mathbf{R}^* = \bigcup_{i \geq 0} \mathbf{R}^i$ is the set of all joins of finitely many genes of \mathbf{R} .
- $\mathbf{R}^+ = \bigcup_{i > 0} \mathbf{R}^i$, so $\mathbf{R}^* = \mathbf{R}^+ + \mathbf{1}$, where $+$ is being used for set union.
- \mathbf{R}^∞ is the set of joins of infinitely many genes of \mathbf{R} .
- \mathbf{R}^\sim is the set $\mathbf{R}^* + \mathbf{R}^\infty$ of finite and infinite genes.

For any predicate, P , $\mathbf{P}^+ = \mathbf{P}^\infty = \mathbf{P}$.

A.4 Collapse and Expansion

We sometimes need to *collapse* a gene \mathbf{r} into the pair of its endpoints $\langle r_1, r_{|\mathbf{r}|} \rangle$, which we will denote $\llbracket \mathbf{r} \rrbracket$. When \mathbf{r} is infinite, we have $\llbracket \mathbf{r} \rrbracket = \langle r_1, \mathbf{1} \rangle$, pairing the initial point with $\mathbf{1}$, which acts like *any* point of the underlying set V .

To avoid confusion, we indicate by $\llbracket \mathbf{R} \rrbracket$ the binary relation that relates endpoints of genes in \mathbf{R} , recalling that the initial points of infinite genes relate to the wildcard element. This *collapsed* relation is

$$\llbracket \mathbf{R} \rrbracket = \{ \llbracket r \rrbracket : r \in \mathbf{R} \}.$$

Unfortunately, $\llbracket \mathbf{R}\mathbf{S} \rrbracket$ need not be equal to $\llbracket \mathbf{R} \rrbracket \cdot \llbracket \mathbf{S} \rrbracket$, on account of the infinite case. For example, let \mathbf{a} be an infinite gene of a 's, that is, $\mathbf{a} = \llbracket a \rrbracket^\infty$. Then, $\llbracket \mathbf{a}\mathbf{b} \rrbracket = \llbracket \mathbf{a} \rrbracket = \langle a, \mathbf{1} \rangle$, whereas $\llbracket \mathbf{a} \rrbracket \cdot \llbracket \mathbf{b} \rrbracket = \langle a, b \rangle$.

Viewed as tuples, $\llbracket \mathbf{R} \rrbracket$ and \mathbf{R} are the same for purely binary \mathbf{R} . One may think of composition $A \cdot B$ of binary relations as their collapsed join, $\llbracket AB \rrbracket$.

Now, let the *expansion* of a binary relation R , $\langle\langle R \rangle\rangle$, contain all finite genes whose beginning and end points are in R ,

$$\langle\langle R \rangle\rangle = \{ \mathbf{r} \in \Omega^\sim : |\mathbf{r}| > 1, r_1 R r_{|\mathbf{r}|} \}.$$

This suggests defining a dual notion, the *fan* of R , as those genes

$$\llbracket R \rrbracket = \{ \mathbf{r} \in \Omega^\sim : r_1 R r_i, \forall i > 1 \},$$

in which all pairs of the first element and a later element are related by R .

A.5 Properties

We adapt notations from modal logic for properties of genomes. If $\mathbf{R} \subseteq \Omega^\sim$ is a genome, then $\neg\mathbf{R}$ denotes its complement $\Omega^\sim \setminus \mathbf{R}$, $\diamond\mathbf{R}$ is the genome containing all genes having some suffix in \mathbf{R} , and $\square\mathbf{R} = \neg\diamond\neg\mathbf{R}$ are those every suffix of which is in \mathbf{R} .

Let $\mathbf{R}, \mathbf{S} \subseteq \Omega^\sim$ be genomes. The notation

$$\mathbf{S} \models \mathbf{R}$$

will mean that each individual gene in \mathbf{S} has a *prefix* in \mathbf{R} . We also allow singletons in place of \mathbf{S} , as in $s \models \diamond ab$, meaning that the gene s contains an adjacent pair (step) ab , or $s \models \square a$, every subsequence begins with a , implying that s is a finite or infinite sequence purely of a 's.

A.6 Example: Well Quasi-Orderings

Suppose Q is a quasi-ordering over V . Call an infinite sequence $s \in \Omega^\infty$ *good* if

$$s \models \diamond\langle\langle Q \rangle\rangle,$$

that is, if it contains two (not necessarily adjacent) points, a and b , that are (quasi-) ordered as $a Q b$. The earlier one a is less than, or equivalent to, the later one b . Call a sequence s *bad* if it is not good, that is, if

$$s \models \square\llbracket\neg Q\rrbracket,$$

and *very good* if all its infinite subsequences are good:

$$s \models \square\diamond\langle\langle Q \rangle\rangle.$$

Clearly, all subsequences of a bad sequence are bad. (A bad sequence is by its very nature very bad!) By (an easy instance of) Ramsey's Theorem, if a sequence s is very good, then it has an infinite subsequence that is a Q -chain:

$$s \models \diamond\langle\langle Q \rangle\rangle^\infty.$$

A quasi-ordering Q is *well*, if

$$\Omega^\infty \models \diamond\langle\langle Q \rangle\rangle,$$

meaning that every infinite sequence is good. It goes without saying that this implies

$$\Omega^\infty \models (\Omega^\sim\langle\langle Q \rangle\rangle)^\infty.$$

But, as we just saw, it is actually equivalent to the stronger

$$\Omega^\infty \models \diamond\langle\langle Q \rangle\rangle^\infty$$

and the even stronger-looking

$$\Omega^\infty \models \square \diamond \langle\langle Q \rangle\rangle^\infty,$$

meaning that infinite sequences are all very good and hence possess infinite chains.

Suppose that A is well-founded:

$$A_\infty = \emptyset.$$

Then, if there is any bad sequence, then there is also a *minimal* bad sequence, \mathbf{b} , such that, in addition to being bad, taking an A -step from any point can lead only to a good sequence.

Let A_1, A_2, \dots, A_n be a fixed finite number ($n \geq 0$) of binary relations over V . (For terms, say, A_i might give the i th immediate subterm.) Let well-founded $A = \bigcup_i A_i$ be their union; A^- is its inverse. Let $\bar{A} \subseteq V \times V^n$ denote the set of pairs $(a, \langle a_1, \dots, a_n \rangle)$, such that $a A_i a_i$ for all i , and let \bar{A}^- be its inverse. Call a point that has an A -successor (that is, a neighbor at the arrowhead side of an outgoing “arrow”) a *parent* and its successor (descendant), a *child*. So, the relation A gives any child a_i of a parent $a \in V$, while \bar{A} gives the tuple $\langle a_1, \dots, a_n \rangle$ of all its children. Finally, let

$$\bar{Q} = \underbrace{Q \times \dots \times Q}_{n \text{ times}}$$

denote the n -fold direct (Cartesian) product of Q , which is the component-wise comparison of n elements in Q .

Suppose further all the following:

$$\begin{aligned} Q \cdot A^- &\subseteq Q \\ \bar{A} \cdot \bar{Q} \cdot \bar{A}^- &\subseteq Q \\ \Omega^\infty &\models \diamond \llbracket A \Omega \rrbracket + \diamond \langle\langle Q \rangle\rangle. \end{aligned}$$

Supposition (2) means that what is smaller than a child is smaller than its parent. It is satisfied, as a special case, whenever $A^- \subseteq Q$ as sets of pairs, since Q is transitive. Supposition (2) means that two parents are ordered if *all* their children are. The last supposition (2) means that every infinite sequence includes a parent or else is good. By Ramsey’s Theorem, again, this is equivalent to

$$\Omega^\infty \models \diamond \langle\langle A \cdot \Omega \rangle\rangle^\infty + \diamond \langle\langle Q \rangle\rangle^\infty. \quad (5')$$

That is, every infinite sequence includes infinitely many parents, or else is very good.

It turns out that Q must be a well quasi-ordering of V if all four suppositions (2–2) hold. The proof follows the lines of Nash-Williams’ proof [23] of Kruskal’s Tree Theorem.