GPO: A Path Ordering for Graphs

Nachum Dershowitz[1] and Jean-Pierre Jouannaud[2]

1 **School of Computer Science, Tel Aviv University, Tel Aviv, Israel**
2 **LIX & Deducteam, École Polytechnique, Palaiseau, France**

Abstract

We define well-founded monotonic rewrite orderings on graphs inspired by the recursive path ordering on terms. Our graph path ordering applies to finite, directed, ordered multigraphs and provides a building block for rewriting with such graphs, which should impact the many areas in which computations take place on graphs.

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

We are interested in well-founded orderings that can be used to show termination of rewriting on first-order terms having both sharing and back-arrows, which is another way of saying that we study rewriting of rooted (multi-) graphs, each vertex of which is labeled by a function symbol whose arity governs the number of vertices it points to. Different target applications require different properties of the ordering: totality is crucial for operaders, while monotonicity with respect to graph structure is important for rewriters.

We propose a generalization of the recursive path ordering to ordered, labeled, rooted graphs. The graph ordering we end up with, *GPO*, has the very same definition as the recursive path ordering (RPO), but the computation of the "head" and "tail" of a multigraph shares little resemblance with the case of trees, for which the head is the top function symbol labeling the root of the tree and the tail is the list of its subtrees. GPO has many of the properties that are important for its various potential users. It is well-founded, total on graph expressions up to isomorphism, and monotonicity is testable.

Graph rewriting has been richly studied. Path orderings of term graphs were developed in [3]. Graph decomposition and weight-based orderings for cyclic graphs are explored in [1]. By adding a root structure to cyclic graphs, we get a natural decomposition into head and tail and a concomitant path ordering.

2 Drags

We work with *drags*, which are finite ***d***irected multi-***r***ooted vertex-l***a***beled ***g***raphs with multiple edges. We begin with a brief presentation of the algebra of drags, developed more fully in [2].

We presuppose a set of function symbols Σ, whose elements are used as vertex labels and are equipped with a fixed arity, plus a denumerable set of (nullary) *variable* symbols Ξ disjoint from Σ. Outgoing edges are ordered (from left to right, say) and their quantity depends solely on the label of the vertex. An "open" drag will include variables, while a "closed" one won't. The successors of a vertex labeled $f \in \Sigma$ in a drag are understood as the arguments of the function symbol f – in order. In contrast with the common categorical approach to graph rewriting, this multigraph model is quite standard. The novelty is that we allow for arbitrarily many roots and arbitrarily complex cycles.

Conference title on which this volume is based on.
Editors: Billy Editor and Bill Editors; pp. 1–5

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

A *(closed) drag* is a tuple $\langle V, R, L, X\rangle$, where V is a finite set of *vertices*; R is a finite (ordered) list of vertices, called *roots*, with $R(n)$ denoting the nth root in the list; $L : V \to \Sigma$ is the *labeling* function, mapping vertices to labels from the vocabulary Σ; and $X : V \to V^*$ is the *successor* function, mapping each vertex $v \in V$ to a list of vertices in V whose length equals the appropriate arity. Given $b \in X(a)$, we say that there is an *edge* from *source* a to *target* b and write aXb. The reflexive-transitive closure X^* of relation X is *accessibility*. Vertex v is *accessible* if it is accessible from a root $r \in R$ (rX^*v). A drag is *clean* if all its vertices are accessible. A root r is *maximal* if it can access all roots its accessible from ($\forall r' \in R.\ r'X^*r$ implies rX^*r').

An *open drag* is a drag over $\Sigma \cup \Xi$. The vertices labeled by a function symbol in Σ are *internal*; those labeled by a variable are called *sprouts*. When nonempty, the set S of sprouts will be added at the end of the tuple: $\langle V, R, L, X, S\rangle$. An open drag is termed *linear* if different sprouts have different labels. It is *cyclic* if each internal vertex accesses a root ($\forall v \in V \setminus S.\ \exists r \in R.\ vX^*r$) and it has no variable roots ($R \cap S = \varnothing$).

Given drag D, $\mathcal{A}cc(D)$ is its set of accessible vertices; $\mathcal{R}(D)$, its list of roots; $\mathcal{R}^\bullet(D) \subseteq \mathcal{R}(D)$ are the *maximal* roots; $[R]$ are the numbers $[1\,..\,|R|]$ referring to the roots in order; $\mathcal{S}(D)$ are its sprouts; $\mathcal{V}ar(D) = L(\mathcal{S}(D))$ are the variables labeling its sprouts; and $D^\sharp$ is the *clean* drag obtained by removing inaccessible vertices and associated edges. Drag D is *(quasi-) isomorphic* to drag D', indicated $D \cong D'$ (or $D \simeq D'$ in the quasi case) if there is a graph isomorphism between them that respects roots (as multisets in the quasi case) and sprouts up to renaming of labels.

Given drag $D = \langle V, R, L, X, S\rangle$ and a subset $W \subseteq V$ of its vertices, the *subdrag* $D_{|W}$ of D generated by vertices W is the drag $\langle V', R', L', X', S'\rangle$ where V' is the least superset of W that is closed under X; L', X', S' are the restrictions of L, X, S to V'; and R' is $(R \cap V') \cup (X(V \setminus V') \cap V')$. A subdrag is clean by construction. Its roots are obtained by adding as new roots those vertices that have an incoming edge in D that is not in $D_{|W}$. The order of elements in this additional list does not really matter for now. In particular, restricting a drag D to its maximal roots, yields $D^\sharp$ ($= D_{|\mathcal{R}^\bullet(D)}$).

The key to working with drags is that we can equip them with a parameterized binary composition operator that connects sprouts of each of two drags with roots of the other. Denote by $\mathcal{D}om(\xi)$ and $\mathcal{I}m(\xi)$ the *domain (of definition)* and *image* of a (partial) function ξ, using $\xi_{A\to B}$ for its restriction going from $A \subseteq \mathcal{D}om(\xi)$ to $B \subseteq \mathcal{I}m(\xi)$, omitting $\to B$ when irrelevant. Let $D = \langle V, R, L, X, S\rangle$ and $D' = \langle V', R', L', X', S'\rangle$ be open drags. A *switchboard* $\xi : S \cup S' \to \mathbb{N}$ for D, D' splits into a pair $\langle \xi_D : S \to [R'], \xi_{D'} : S' \to [R]\rangle$ of partial injective functions such that (i) $\forall s, t \in S.\ s \in \mathcal{D}om(\xi)$ and $L(s) = L(t)$ imply $t \in \mathcal{D}om(\xi)$ and $D_{|R'(\xi(s))} \simeq D_{|R'(\xi(t))}$; (ii) $\forall s', t' \in S'.\ s' \in \mathcal{D}om(\xi)$ and $L'(s') = L'(t')$ imply $t' \in \mathcal{D}om(\xi)$ and $D'_{|R(\xi(s'))} \simeq D'_{|R(\xi(t'))}$. We also say that $D'\xi$ is an *extension* of D.

Each switchboard induces a binary composition operation on open drags, the precise definition of which we omit (but see [2]). The essence is that the (disjoint) union of the two drags is formed, with sprouts in the domain of the switchboards merged with the vertices referred to in its images. This necessitates renaming targets of edges that had pointed to sprouts.

A switchboard ξ for D, D' is *directed* if one of ξ_D and $\xi_{D'}$ has an empty domain. Directed switchboards correspond to the tree case, with all connections from one of the drags to the other.

▸ **Lemma 1** (Unique Decomposition). *Given a drag D and a subset of its vertices W, there exists a drag A, called its* antecedent*, and a directed switchboard ξ such that $D = A\,\xi\,D_{|W}$.*

The fact that the switchboard ξ is directed expresses the property that decomposing a drag

into a subdrag and its antecedent does not break any of its cycles. If it's cyclic, it's its own antecedent.

Just as a tree can be decomposed into a head node f and subtrees, a drag has a head, viz. its smallest (nontrivial) antecedent, and one tail, possibly a list of several connected components. The *tail* $\triangledown D$ of a drag $D = \langle V, R, L, X, S\rangle$ is the subdrag generated by the set of vertices $V \setminus \{v \in V : vX^*\mathcal{R}^\bullet(D)\}$. Furthermore, for drag D, there exists a linear drag $\widehat{D}$, the *head* of D, and a directed switchboard ξ such that $D = \widehat{D}\,\xi\,\triangledown D$. The head of a drag is therefore the antecedent of its tail.

Isomorphic drags have isomorphic heads and tails. The crucial point is that decomposition into head and tail is canonical and faithful because drags are multi-rooted. Uni-rooted drags cannot represent horizontal sharing, in contrast to vertical sharing which can *sometimes* be preserved with uni-rooted drags.

3 Drag Rewriting

An extension $B\xi$ is a *context* of drag D when $\mathcal{D}om(\xi_D) = \varnothing$. It's a *substitution* when $\mathcal{D}om(\xi_B) = \varnothing$. It is a *rewriting* extension if ξ_B is linear and surjective and ξ_D is total. An extension $B\xi$ of a clean drag D is *cyclic* if B is a linear drag generated by $\mathcal{I}m(\xi_D)$, $B\,\xi\,D$ is a clean nonempty drag, and there exists $s \in \mathcal{D}om(\xi_B)$ such that tX^*s for all $t \in V$. The extension is *trivial* if B is an identity drag (a linear open drag all of whose vertices are sprouts and sans edges) and *total* if ξ_B is surjective.

The conditions for being a cyclic extension impose that ξ_D is surjective on $R \setminus S$ as a set so as to generate B. *Identity cyclic extensions* of D are such that its sprout variables are in one-to-one correspondence with the sprouts in D that are connected by the switchboard and connect them to some of the roots of D. Thus, cyclic extensions modify the structure of D by connecting some of its sprouts to some of its roots. Unfortunately, identity cyclic extensions suffice for that purpose only in special cases. Identity extensions allow one to change the structure of a drag without changing its internal nodes. If the drag has a single root, it is easy to see that identity extensions are enough to predict all forms that a drag may take under composition with an extension. This is not true for multirooted drags, since identity cyclic extensions cannot reach two different roots from the same sprout.

▸ **Theorem 2.** *Let D, E be clean nonempty drags and ξ be a switchboard for them. If ξ_E is surjective on maximal roots $\mathcal{R}^\bullet(D)$, then there exist drags A, B, C and switchboards ζ, θ such that (i) $B\langle\xi_B, \xi_{D\to B}\rangle$ is a cyclic extension of D; (ii) $C\theta$ is a substitution extension of $B\,\xi\,D$; (iii) $A\zeta$ is a context extension of $(B\,\theta\,D)\,\eta\,C$; (iv) $E\,\xi\,D = A\,\zeta\,((B\,\theta\,D)\,\eta\,C)$; (v) C is empty if all internal nodes of E reach one of its sprouts.*

Can a drag D be seen as the composition of a given drag G with some context C via some switchboard ξ? In this case, we say that D *matches* G, E and ξ being the *matching* context and switchboard. The idea is that E splits into three: vertices that are accessible from some sprout of G and can access some of its roots define a drag B; those accessible from some sprout of G but which cannot access any of its roots define a drag C that is a substitution extension of G; those which are not accessible from the sprouts of G form the remaining part A, which is a context extension of G.

A *graph rewrite rule* is a pair of open clean drags written $G \to G'$ such that $|\mathcal{R}(G)| = |\mathcal{R}(G')|$ and $\mathcal{V}ar(G') \subseteq \mathcal{V}ar(G)$. We say that a nonempty clean drag D *rewrites* to a clean drag D' via this rule, and write $D \longrightarrow D'$, if $D = E\,\xi\,G$ and $D' = (E\,\xi\,G')^\natural$ for some rewriting extension $E\xi$ of G, such that ξ_G is linear if G is. We have: (1) If $D \longrightarrow D'$, then $\mathcal{V}ar(D') \subseteq$

$\mathcal{V}ar(D)$. (2) If $D \longrightarrow D'$ is a rewrite, C is an open drag, and ξ is a switchboard for C, D, then ξ (restricted to sprouts of D') is also a switchboard for C, D'. (3) If $D \longrightarrow D'$, then $D = A\,\zeta\,((B\,\xi\,L)\,\theta\,C)$ and $D' \cong A\,\zeta\,((B\,\xi\,L)\,\theta\,C)$ for some A, B, C and ζ, ξ, θ such that $A\zeta$, $B\xi$ and $C\theta$ are context, cyclic and substitution extensions, respectively.

A *graph rewrite system* is a set of graph rewrite rules, each of which can be used to rewrite.

4 Graph Path Ordering

A *reduction ordering* is a well-founded ordering $\succ$ of the set of drags that is (i) *compatible* with drag isomorphism: for all D, D', E, E' such that $D \succ E$, $D \cong D'$ and $E \cong E'$, then $D' \succ E'$; (ii) *monotonic*: for all D, E such that $D \succ E$ and for all context extensions $A\xi$ of D, then $A\,\xi\,D \succ A\,\xi\,E$; and (iii) *stable*: for all D, E such that $D \succ E$ and for all substitution extensions $C\xi$ of D, then $D\,\xi\,C \succ E\,\xi\,C$. Apart from compatibility with isomorphism, the notion of reduction ordering is the same as the usual one. Monotonicity is the usual property since a directed switchboard turns the context A into a usual context. Stability corresponds to the usual stability property, but substitution extensions can introduce sharing.

▸ **Theorem 3** (Termination). *A graph rewrite system $\mathcal{R}$ terminates iff there's a graph reduction ordering $\succ$ such that, for all rules $G \to G' \in \mathcal{R}$ and cyclic extensions $B\xi$ of G, we have $B\,\xi\,G \succ B\,\xi\,R$.*

We need the analog for drags of the precedence on function symbols used by RPO: A *head order* for a drag rewrite system $\mathcal{R}$ is a quasi-order $\succcurlyeq$ on clean cyclic drags whose strict part $\succ$ is well-founded.

▸ **Definition 4** (Graph Path Order (GPO)). Given two drags s, t and a head order $\succcurlyeq$, we define $s \succ t$ (under GPO), iff any of the following three cases hold:

$$[\nabla]\ \nabla s \succcurlyeq t \qquad [\rhd]\ \widehat{s} \succ \widehat{t} \text{ and } s \succ \nabla t \qquad [\doteq]\ \widehat{s} \doteq \widehat{t},\ s \succ \nabla t \text{ and } \nabla s \succ \nabla t.$$

GPO is defined by induction on the pair $\langle s, t\rangle$ via the lexicographic extension of the subdrag order imposed by tails. Let $t \rhd u$ if $u = \nabla t$ and let $\rhd^+$ be its transitive closure.

▸ **Lemma 5** (Subdrag Properties). *(1) The subdrag order $\rhd^+$ is well-founded on nonempty open drags. (2) The empty drag is minimal in $\succ$. (3) $\rhd \subseteq \succ$. (4) If $D \succ E$, then $\mathcal{V}ar(D) \subseteq \mathcal{V}ar(E)$. (5) GPO ($\succ$) is transitive.*

GPO is a strict ordering compatible with drag isomorphism. This justifies our way of building compatibility into a path ordering via a careful definition of subdrags instead of using a normal-form representation of congruence classes of drags. If the drag ∇t is terminating with respect to $\succ$, then the drag $t = g(\overline{t})$ is.

A head order is *compatible* if two cyclic drags that are isomorphic up to their lists of roots are equivalent in the quasi-order. It is *total* if $\succcurlyeq$ is total and its equivalence $\doteq$ is exactly cyclic drag isomorphism up to their lists of roots. It is *subcyclic* if $D \succcurlyeq A$ whenever $D = A\,\xi\,B$ for a cyclic drag B that is not isomorphic to an identity. And it is *cyclic monotonic* if for all cyclic drags D, E and for all cyclic extensions $C\xi$ of D, $D \succcurlyeq E$ implies $C\,\xi\,D \succcurlyeq C\,\xi\,E$. Cyclic monotonicity tells us that the order between cycles is preserved by growing them. Monotonicity/stability of drag orders and cyclic monotonicity of head orders are complementary: the former extends a drag by preserving its head, while the latter extends heads.

▸ **Theorem 6** (GPO is Good). *(1) GPO is a rewrite ordering: it is monotonic and it is stable (if $D \succ E$, then $D\,\xi\,C \succ E\,\xi\,C$ for all substitution extensions $C\xi$ of D). (2) GPO is well-founded, which makes it a reduction ordering. (3) GPO is total on equivalence classes of drags (modulo isomorphism) if the head order is total.*

We define two head orders, one total but not subcyclic, and another that is subcyclic but partial.

Let $\geq$ be a total precedence on Σ, whose strict part $>$ is well-founded, and $D = \langle V, R, L, X\rangle$, a clean drag from which the sprouts and their incoming edges have been removed. Represent a drag as a list of function symbols. The *interpretation* $[\![D]\!]$ of $D = \langle V, R, L, X\rangle$, is a list of symbols in Σ defined as follows: If $\mathcal{A}cc(D) = \varnothing$, then $[\![D]\!] := \varnothing$. Otherwise, $[\![D]\!] := L(r) \cup [\![W]\!]$, where $R = r \cup R'$ and $W = \langle V \setminus r, (X(r) \setminus r) \cup R', L', X'\rangle$, L', X' being the restrictions of L, X to V', respectively. In words, $[\![D]\!]$ collects the function symbols labeling the internal nodes of a drag by traversing this drag in a depth-first manner starting from R. We can now define our head order on drags: $D \geqslant D'$ iff $\langle |[\![D]\!]|, [\![D]\!]\rangle\ (\geq_{\mathbb{N}}, \geq_{lex})_{lex}\ \langle |[\![D']\!]|, [\![D']\!]\rangle$. The relation $\geqslant$ is a total head order. This head order is not subcyclic, nor cyclic-monotonic.

Alternatively, represent a drag as the multiset of function symbols labeling its accessible vertices: Define the *interpretation* as the multiset of symbols that label the vertices of $D^\sharp$. Given two drags, we define: $D \geqslant D'$ iff $[\![D]\!] \geq_{mul} [\![D']\!]$. This $\geqslant$ is a subcyclic, cyclic-monotonic head order.

▸ **Example 7.** Consider an application of a rule $G = f(x) \to a = G'$, where f and a are drag labels and x is a variable, to the cyclic graph $D = \ \to f \leftrightarrow f$ leading to the noncyclic term $D' = f(a)$. The resultant inequality is $D = \langle\{1,2\}, 1, \{1, 2 \mapsto_L f\}, \{1 \mapsto_X 2, 2 \mapsto_X 1\}\rangle \succ \langle\{1,2\}, 1, \{1 \mapsto_{L'} f, 2 \mapsto_{L'} a\}, 1 \mapsto_{X'} 2\rangle = D'$, foregoing braces around singletons. The first drag is its own head. The head of the second is $f(y) = \langle\{1,2\}, 1, \{1 \mapsto_{L'} f, 2 \mapsto_{L'} y\}, 1 \mapsto_{X'} y, 2\rangle$, which is a subdrag of the first. By the subterm property of head orders, the first is strictly larger than the second in $\geqslant$. Therefore, we are left with a subgoal: $D \succ \nabla D' = a = \langle 2, 2, 2 \mapsto_{L''} a, \varnothing\rangle = D''$. This time, the second drag is also itself a head, but it is not a subdrag of the first. Therefore, we must have $D \succcurlyeq D''$ in the head order for this constraint to pass. A head order with precedence $f > a$ does the trick. ◂

▸ **Theorem 8** (Decidability). *It is decidable whether a system $\mathcal{R}$ terminates under GPO provided the universal first-order fragment of head-order constraints is decidable.*

References

1 Guillaume Bonfante and Bruno Guillaume. Non-simplifying graph rewriting termination. In Rachid Echahed and Detlef Plump, editors, *Proceedings of the 7th International Workshop on Computing with Terms and Graphs (TERMGRAPH)*, pages 4–16, Rome, Italy, March 2013.

2 Nachum Dershowitz and Jean-Pierre Jouannaud. Drags: A simple algebraic framework for graph rewriting. In *Proceedings of the 10th International Workshop on Computing with Terms and Graphs (TERMGRAPH)*, Oxford, UK, July 2018.

3 Detlef Plump. Simplification orders for term graph rewriting. In Igor Prívara and Peter Ruzicka, editors, *Proceedings of the 22nd International Symposium on Mathematical Foundations of Computer Science (MFCS'97)*, volume 1295 of *Lecture Notes in Computer Science*, pages 458–467, Bratislava, Slovakia, August 1997. Springer.