GPO: A Path Ordering for Graphs

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

1 **Tel Aviv University, Tel Aviv, Israel**
2 **École Polytechnique, Palaiseau, France**

Abstract

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

1 Introduction and Motivation

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, that is, in rooted graphs (actually, multigraphs), each vertex of which is labeled by a function symbol whose arity governs the number of vertices it relates to in the graph via an outgoing edge. Different target applications require different properties: totality is crucial for operaders, while monotonicity with respect to the graph structure is important for graph rewriters.

The recursive path ordering (RPO) is recursively generated from an order on the set of function symbols, called "precedence". RPO is compatible with term structure; it's well-founded when the precedence is; it's increasing when the precedence increases; and – last but not least – its behavior is intuitive and can be easily and efficiently implemented. One main aspect of its definition is that comparing two terms with the same head symbol reduces to a comparison of their respective subterms via a lexicographic or multiset extension of the order on their subterms organized as lists or multisets, respectively. The type of extension assigned to a head symbol is called its *status*. When statuses are lexicographic and the precedence total, RPO itself becomes total.

We propose a generalization of RPO to ordered, labeled, rooted graphs. The ordering we end up with, *GPO*, has the very same definition as 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. A key property of our decomposition is that isomorphic multigraphs have isomorphic heads and tails. This alone ensures that the order is compatible with multigraph isomorphism. 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.

2 Drags: Finite, Directed, Labeled Multigraphs

This section is a quick presentation of the algebra of drags (developed in a paper of the authors submitted to TermGraph 2018). Drags are finite directed multi-rooted graphs with labelled vertices and multiple edges. Let's assume that outgoing neighbors (vertices at the other end of outgoing edges) are ordered (from left to right, say) and that their number is fixed, depending solely on the label of the vertex: we presuppose a set of function symbols Σ, whose elements $f \in \Sigma$ used as labels are equipped with a fixed arity, and a denumerable set of variable symbols Ξ disjoint from Σ, whose arity is 0. We call these finite ***d***irected ***r***ooted l***a***belled ***g***raphs, ***drag***s. The successors of a vertex labeled f in a drag are implicitly

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

interpreted as the arguments of the function symbol f. In contrast with the standard categorical approach, this multi-graph model is very standard. The novelty is that we allow for arbitrarily many roots and arbitrarily complex cycles.

We use vertical bars $|\cdot|$ to denote various quantities, such as length of lists, size of sets or expressions, and arity of function symbols. We use $\varnothing$ for an empty list, set or multiset; $\cup$ for list concatenation as well as set and multiset union; and $\setminus$ for list, set or multiset difference. We identify a singleton list, set or multiset with its single element.

A *(closed) drag* is a tuple $\langle V, R, L, X\rangle$, where V is the finite set of *vertices*; R is a finite list of vertices, called *roots*, and $R(n)$ is the nth root in the list; $L : V \to \Sigma$ is the *labeling* function, mapping vertices to labels from some 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 $|L(v)|$. Given $b \in X(a)$, then (a, b) is an *edge* with *source* a and *target* b. We also write aXb. The transitive closure X^* of the relation X is called *accessibility*. A vertex v is said to be *accessible* if rX^*v for some root r. A drag is *clean* if all its vertices are accessible. A root r is *maximal* if $\forall r' \in R.\ r'X^*r$ implies rX^*r'. An *open drag* is a drag over the set $\Sigma \cup \Xi$, where Ξ is a set of *variables*. The vertices labeled by a function symbol in Σ are *internal*; those labeled by a variable are called *sprouts*. When non-empty, 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 *linear* if different sprouts have different labels. It is *cyclic* if $R \cap S = \varnothing$ and $\forall v \in V \setminus S.\ \exists r \in R.\ vX^*r$.

Given drag D, we use $\mathcal{A}cc(D)$ for its set of accessible vertices; $\mathcal{R}(D)$ for its set of roots; $\mathcal{R}(D)_{max}$ for the subset of *maximal* roots; $\mathcal{S}(D)$ for its set of sprouts; $\mathcal{V}ar(D)$ for the set of variables labeling its sprouts; $D^{\overline{r}}_{\overline{s}}$ to indicate its roots and sprouts; and $D{\downarrow}$ for the clean drag obtained by removing inaccessible vertices and related edges. Given a drag $U = \langle V, R, L, X, S\rangle$, and a subset W of its vertices, the *subdrag* $U_{|W}$ of U generated by W is the drag $\langle V', R', L', X', S'\rangle$ where (i) V' is the least superset of W that is closed under X; (ii) L', X', S' are the restrictions of L, X, S to V'; (iii) 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 which have an incoming edge in U that is not in $U_{|W}$. 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 $D{\downarrow}$. On the other hand, restricting any drag to an empty set of vertices, yields the empty drag $\varnothing \stackrel{\text{def}}{=} 1^{\varnothing}_{\varnothing} = (\varnothing, \varnothing, \varnothing, \varnothing, \varnothing)$.

An *isomorphism* from $D = \langle V, R, L, X, S\rangle$ to $D' = \langle V', R', L', X', S'\rangle$ is a one-to-one mapping $o : \mathcal{A}cc(V) \mapsto \mathcal{A}cc(V')$ such that (i) o restricts to bijections between the lists of roots and sets of accessible sprouts; (ii) $\forall v \in \mathcal{A}cc(V) \setminus S.\ L(v) = L'(o(v))$ and $X(v) = X'(o(v))$. We write $D =_o D'$, or simply $D = D'$ when D and D' are isomorphic drags. We write $D \simeq_o D'$, or simply $D \simeq D'$, and say that D, D' are *quasi-isomorphic* if o restricts to a bijection between the multisets (instead of lists) of roots. We write $D \equiv D'$ and $D \cong D'$ instead of $D = D'$ and $D \simeq D'$ in case o is the identity. Drag D isomorphic (with $\equiv$) to $D{\downarrow}$.

The key is that we equip drags with a binary composition operator, which connects the sprouts of each drag with the 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 subset A of its domain to subset B of its image, omitting $\to B$ when irrelevant. Both are sets. hence $\mathcal{I}m(\xi)$ may differ from $\xi(R)$, which is a list with possibly repetitions when R is a list itself. Let $D = \langle V, R, L, X, S\rangle$ and $D' = \langle V', R', L', X', S'\rangle$ be open drags. A *switchboard* ξ for (D, D') is a pair $\langle \xi_D : S \to [1 \,..\, |R'|], \xi_{D'} : S' \to [1 \,..\, |R|]\rangle$ of partial injective functions (we also say that (D', ξ) is an *extension* of D), such that (i) $\forall s, t \in S.\ s \in \mathcal{D}om(\xi_D)$ and $L(s) = L(t)$ imply $t \in \mathcal{D}om(\xi_D)$ and $D_{|R'(\xi_D(s))} \simeq D_{|R'(\xi_D(t))}$; (ii) $\forall s, t \in S'.\ s \in \mathcal{D}om(\xi_{D'})$ and $L'(s) = L'(t)$ imply $t \in \mathcal{D}om(\xi_{D'})$ and $D'_{|R(\xi_{D'}(s))} \simeq D'_{|R(\xi_{D'}(t))}$.

A switchboard ξ for (D, D') is *directed* if one of ξ_D and $\xi_{D'}$ has an empty domain. The pair $(D', \xi_{D'})$ is called a *context extension* of D when $\mathcal{D}om(\xi_D) = \varnothing$, and a *substitution extension* of D when $\mathcal{D}om(\xi_{D'}) = \varnothing$.

It is a *rewriting* switchboard if $\xi_{D'}$ is linear and surjective and ξ_D is total. Then the pair (D', ξ) is called a *rewriting extension* of D. Directed switchboards correspond to the tree case, with all connections from one drag to the other. Rewriting switchboards allow one to "encompass" drag D' in drag D.

A switchboard induces a binary composition operation on open drags, the precise definition of which we omit. The essence is that the (disjoint) union of the two drags is formed, but with sprouts in the domain of the switchboards merged with the vertices referred to in the switchboard images. This necessitates renaming the targets of edges that had pointed to the sprouts. Our definition of switchboard being (almost) symmetric, composition is itself symmetric provided all roots of the resulting drag originate from the same side. Otherwise, composition yields drags that are equal up to a cyclic permutation of their roots. Composition of drags is quasi-commutative, and becomes commutative for switchboards whose context or substitution is surjective, e.g. rewriting switchboards.

The *identity* is a linear open drag whose all vertices are its sprouts and set of edges is empty. We denote it by 1^Y_Z, where $Y \subseteq Z^*$ is its list of roots. We use $\varnothing$ for the drag $1^{\varnothing}_Z$, which is isomorphic to $1^{\varnothing}_{\varnothing}$, and is called the empty drag for that reason.

▶ **Lemma 1.** *Given a drag D and a subset of vertices $W \subseteq V$, there exists a linear drag A, called* antecedent*, and a total, surjective, and directed switchboard ζ such that $D = A \otimes_\zeta D_{|W}$.*

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. Note that ξ induces an order on the roots of the subdrag which are not roots of the whole drag.

Just as a tree can be decomposed into a head node f and subtrees, a drag has a head, which is intended to be the largest cycle containing the maximal roots of the drag – also the smallest nontrivial antecedent of the drag – and one tail, possibly a list of several connected components. The *tail* ∇D of a drag $D = \langle V, R, L, X, S\rangle$ is the subdrag generated by the set of vertices $V \setminus \{v \in V : vX^*\mathcal{R}(D)_{max}\}$. Furthermore, for drag D, there exists a linear drag $\widehat{D}$, called *head* of D, and a directed switchboard ζ such that $D = \widehat{D} \otimes_\zeta \nabla D$. The head of a drag is therefore the antecedent of its tail.

Note that ξ is a bijection between the sprouts of the head and the roots of the tail that were not already roots of the drag D. We assume that the sprouts of the head are ordered with respect to a depth first search initialized with its list of roots. This order on the sprouts of the head induces, via the switchboard ξ, an order on the roots of the associated tail that were not roots of the original drag. The tail is therefore now canonically determined. In short, 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. Moreover, the fact that tails are quasi-isomorphic does not hamper faithfulness: different orders of new roots for the tails yield different switchboards but the order of roots resulting from the composition does not depend upon those orders.

Next, we investigate further ways of decomposing a drag, possibly breaking its cycles. Can a drag D be seen as the composition of a given drag U with some context W via some switchboard ξ? In this case, we say that D *matches* U, W and ξ being the *matching* context and switchboard. The idea is that W splits into three: vertices that are accessible from some sprout of U and can access some of its roots define a drag B; those accessible from some

sprout of U but which cannot access any of its roots define a drag C that is a substitution extension of U; those which are not accessible from the sprouts of U form the remaining part A, which is a context extension of U.

An extension (B,ξ) of a clean drag U is *cyclic* if B is a linear drag generated by $\mathcal{I}m(\xi_U)$, $B\otimes_\xi U$ is a clean nonempty drag, and for all $t\in V$, there exists $s\in\mathcal{D}om(\xi_B)$ such that tX^*s. The extension is *trivial* if B is an identity drag and *total* if ξ_B is surjective. This condition resembles the one for cyclic drags, but doesn't imply that $B\otimes_\xi U$ is cyclic because not all internal vertices of U may reach a root of $B\otimes_\xi U$. Only *total* extensions are cyclic.

The conditions for being a cyclic extension impose that ξ_U is surjective on $R\setminus S$ as a set so as to generate B. *Identity cyclic extensions* of U are of the form $(1_Y^Z,\iota)$, where the variables in Y are one-to-one with those in $\mathcal{D}om(\iota_U)\subseteq\mathcal{V}ar(U)$, and $\iota_1 : Y\to[1\,..\,|R|]$ is an arbitrary map. The rôle of cyclic extensions is to modify the structure of U 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 multi-rooted drags, since identity cyclic extensions cannot reach two different roots from the same sprout.

▶ **Theorem 2.** *Let U,W be clean nonempty drags and ξ be a switchboard for (W,U). If ξ_W is surjective on $\mathcal{R}(U)_{max}$, then, there exist drags A,B,C and switchboards ζ,θ such that (i) $(B,\langle\xi_B,\xi_{U\to B}\rangle)$ is a cyclic extension of U denoted by (B,ξ); (ii) (C,θ) is a substitution extension of $B\otimes_\xi U$; (iii) (A,ζ) is a context extension of $(B\otimes_\theta U)\otimes_\eta C$; (iv) $W\otimes_\xi U = A\otimes_\zeta((B\otimes_\theta U)\otimes_\eta C)$; (v) C is empty if all internal nodes of W reach one of its sprouts.*

A *graph rewrite rule* is a pair of open clean drags written $L\to R$ such that $|\mathcal{R}(L)| = |\mathcal{R}(R)|$ and $\mathcal{V}ar(R)\subseteq\mathcal{V}ar(L)$. A graph rewrite system is a set of graph rewrite rules. Let $\mathcal{R}$ be a graph rewrite system. We say that a nonempty clean drag D *rewrites* to a clean drag D', and write $D\longrightarrow_{\mathcal{R}} D'$ iff $D = W\otimes_\xi L$ and $D' = (W\otimes_\xi R)\downarrow$ for some drag rewrite rule $L\to R\in\mathcal{R}$ and rewriting extension (W,ξ) of L, such that ξ_L is linear if L is linear. We have: (1) If $U\longrightarrow_{\mathcal{R}} V$, then $\mathcal{V}ar(V)\subseteq\mathcal{V}ar(U)$. (2) Assume U,V,W are three open drags such that the pair (U,V) is a rewrite and ξ is a switchboard for (W,U). Then, ξ is a switchboard for (W,V). (3) Assume $U\longrightarrow_{\mathcal{R}} V$. Then, $U = A\otimes_\zeta((B\otimes_\xi L)\otimes_\theta C)$ and $V\equiv A\otimes_\zeta((B\otimes_\xi L)\otimes_\theta C)$ for some A,B,C and ζ,ξ,θ such that (A,ζ), (B,ξ) and (C,θ) are context, cyclic and substitution extensions, resp.

3 GPO: Graph Path Ordering

A *rewrite ordering* is a well-founded ordering $\succ$ of the set of drags that is (i) *compatible* with drag isomorphism: for all U,U',V,V' such that $U\succ V$, $U\equiv U'$ and $V\equiv V'$, then $U'\succ V'$; (ii) *monotonic*: for all (U,V) such that $U\succ V$ and for all context extensions (A,ξ) of U, then $A\otimes_\xi U\succ A\otimes_\xi V$; (iii) *stable*: for all (U,V) such that $U\succ V$ and for all substitution extensions (C,ξ) of U, then $U\otimes_\xi C\succ V\otimes_\xi C$. Apart from compatibility with isomorphism, the notion of rewrite 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.** *A graph rewrite system $\mathcal{R}$ terminates iff there's a graph rewrite ordering $\succ$ s.t. for all rules $L\to R\in\mathcal{R}$, then $B\otimes_\xi L\succ B\otimes_\xi R$ for all cyclic extensions (B,ξ) of L.*

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 $\succeq$ on clean cyclic drags whose strict part $\succ$ is well-founded. (i) A head order is *compatible* if two cyclic drags that are isomorphic up to their lists of roots are equivalent in the quasi-order. (ii) It is *total* if $\succeq$ is total and its equivalence $\doteq$ is exactly cyclic drag isomorphism up to their lists of roots. (iii) It is *subcyclic* if $W = A \otimes_\xi B$, and B is a cyclic drag that is not isomorphic to an identity drag, then $W \succ A$. (iv) And it is *cyclic monotonic* if for all cyclic drags U, V and for all cyclic extensions (C, ξ) of U, $U \succ V$ implies $C \otimes_\xi U \succeq C \otimes_\xi V$.

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.

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

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

GPO is defined by induction on the pair $\langle s, t\rangle$ via the lexicographic extension of the subdrag order: Let $t \rhd u$ if $u = \nabla t$. (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 $U \succ V$, then $\mathcal{V}ar(U) \subseteq \mathcal{V}ar(V)$. (5) GPO $(\succ)$ is transitive.

Let U, V and $W = \langle V, R, L, X, S\rangle$ be open drags such that $\mathcal{V}ar(V) \subseteq \mathcal{V}ar(U)$. Two switchboards ξ, ξ' for (W, U) and (W, V) are *coherent* if $\forall s, t, n, p.\ \xi_U(s) = n$, $\xi'_V(t) = p$, $L(s) = x$, and $L(t) = x$ imply $\mathcal{R}(W)(n) = \mathcal{R}(W)(p)$. Let U, V, W be open drags such that $U \succ V$, and ξ, ξ' be two coherent switchboards for (W, U) and (W, V) respectively. Then, ξ is a switchboard for (W, V) such that $W \otimes_\xi V = W \otimes_{\xi'} V$.

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 $t = g(\bar{t})$ is.

▶ **Theorem 5.** *(1) GPO is well-founded. (2) If the head order is total, then GPO is total on equivalence classes of drags modulo isomorphism. (3) GPO is monotonic. (4) GPO is stable: if $U \succ V$, then $U \otimes_\xi C \succ V \otimes C$ for all substitution extensions (C, ξ) of U.*

We define a head order that is total but not subcyclic, and another, partial subcyclic one. Let $\geq$ be a total precedence on Σ, whose strict part is well-founded, and $U = \langle V, R, L, X\rangle$, 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* $[\![U]\!]$ of $U = \langle V, R, L, X\rangle$, is a list of symbols in Σ defined as follows: If $\mathcal{A}cc(U) = \varnothing$, then $[\![U]\!] \stackrel{\text{def}}{=} \varnothing$. Otherwise, $[\![U]\!] \stackrel{\text{def}}{=} L(r) \cup [\![W]\!]$, where $R = r \cup R'$ and $W = (V \setminus r, (X(r) \setminus r) \cup R', L', X')$, L', X' being the restrictions of L, X to V', respectively. In words, $[\![U]\!]$ 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: $D \succeq D'$ iff $\langle |[\![D]\!]|, [\![D]\!]\rangle\ (\geq_{\mathbb{N}}, \geq_{lex})_{lex} \langle |[\![D']\!]|, [\![D']\!]\rangle$. $\succeq$ 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: We define the *interpretation* as the multiset of symbols that label the vertices of $U{\downarrow}$. Given two drags, we define: $D \succeq D'$ iff $[\![D]\!] \geq_{mul} [\![D']\!]$. This $\succeq$ is a subcyclic, cyclic-monotonic head order.

▶ **Theorem 6.** *It is decidable whether $\mathcal{R}$ terminates with GPO provided universal head-order constraints are decidable.*