ISSN 1361 - 6161

THE UNIVERSITY
of MANCHESTER

Computer Science
University of Manchester

# DPO Rewriting and Abstract Semantics via Opfibrations

R. Banach

Department of Computer Science
University of Manchester
Technical Report Series
UMCS-96-8-1

# DPO Rewriting and Abstract Semantics via Opfibrations*

R. Banach

Department of Computer Science
University of Manchester
Oxford Road, Manchester, U.K.
banach@cs.man.ac.uk

31 August 1996



---

* Refereed by: Andrea Maggiolo-Schettini.

# DPO Rewriting and Abstract Semantics via Opfibrations

R. Banach
Computer Science Dept., Manchester University, Manchester, M13 9PL, U.K.
banach@cs.man.ac.uk

**Abstract:** The classical DPO graph rewriting construction is re-expressed using the split opfibration approach introduced originally for term graph rewriting. Using a skeleton category of graphs, a base of canonical graphs-in-context, with DPO rules as arrows, and with categories of redexes over each object in the base, yields a category of rewrites via the discrete Grothendieck construction. The various possible ways of combining rules and rewrites leads to a variety of functors amongst the various categories formed. Categories whose arrows are rewriting sequences have counterparts where the arrows are elementary event structures, and an event structure semantics for arbitrary graph grammars emerges naturally.

**Key Words:** Graph grammar, DPO rewriting, split opfibration, event semantics, event structure.

## 1 INTRODUCTION

Since the (split) opfibration construction for describing a species of term graph rewriting was introduced in Banach (1993, 1994), the author has wondered whether something similar would work for the classical DPO rewriting approach (Ehrig (1979, 1986)). (For example it is clear that the single pushout approach of Lowe (1991), Lowe and Ehrig (1991) can be accomodated in an opfibration construction, since opfibrations are a natural generalisation of pushouts.) The obvious thing to try, is to set up as base for a Grothendieck construction, a category whose objects are graphs, and whose arrows are DPO rules $(L \leftarrow K : l,\ r : K \rightarrow R) : L \rightarrow R$, with rules $(l,\ l)$ as identities, and forming the composition of arrows $(l_1,\ r_1)$ and $(l_2,\ r_2)$ by pulling back $r_1$ and $l_2$. This strategy is similar to what appears in Corradini et al. (1994d). However for opfibrations, such an attempt fails miserably because the application conditions mix the properties of arrows $(l,\ r) : L \rightarrow R$ with properties of the redexes $g : L \rightarrow G$ that one would find in the category above the objects $L$ in the base. Such a mixing spells disaster for the Grothendieck construction. This paper reports that a slightly different tack on the question leads to success. In fact the mixing referred to, mixes only properties of $l$ and $g$, so taking as objects in the base, arrows $l : K \rightarrow L$, sidesteps the problem and allows the Grothendieck construction to be achieved smoothly. This is done in section 2. In section 3 we consider various categories of rewriting sequences and the functors between them. Section 4 shows that the arrows in these rewriting categories have a natural correspondence with elementary event structures, and section 5 treats the preceding considerations with regard to a particular graph grammar. Section 6 contains a discussion.

## 2 THE DPO APPROACH AS OPFIBRATION

In this section we review the classic double pushout approach to graph rewriting and show how it can be placed in a split opfibration framework.

**Definition 2.1** A graph $G$ is a tuple $(E, V, s, t, l_E, l_V)$ where $E$ and $V$ are (finite) sets of edges and vertices respectively, $s, t : E \to V$ map each edge to its source and target respectively, and $l_E : E \to \Omega_E$, $l_V : V \to \Omega_V$ map edges and vertices to their labels drawn from the edge and vertex label alphabets $\Omega_E$, $\Omega_V$.

**Definition 2.2** A graph morphism $f : G \to G'$ is a pair of *injective* functions $f_E : E \to E'$, $f_V : V \to V'$ such that the diagram below commutes in the obvious way

$$\begin{tikzcd}
 & E \arrow[dl, "l_E"'] \arrow[r, shift left, "s"] \arrow[r, shift right, "t"'] \arrow[d, "f_E"] & V \arrow[d, "f_V"'] \arrow[dr, "l_V"] & \\
\Omega_E & & & \Omega_V \\
 & E' \arrow[ul, "l_E'"] \arrow[r, shift left, "s'"] \arrow[r, shift right, "t'"'] & V' \arrow[ur, "l_V'"'] &
\end{tikzcd}$$

As above, we will systematically use primes, or subscripts identifying the graph (eg. $l_{V,G}$) on the various components, when several graphs are being discussed at once, to disambiguate. This gives us the category $\mathcal{CG}$ of concrete graphs and morphisms.

**Definition 2.3** A production is a pair of arrows $l : K \to L$, $r : K \to R$ in $\mathcal{CG}$. An injective arrow $g : L \to G$ is a redex of the rule, and a rewrite (or direct derivation) of $H$ from $G$ using the given rule and redex, exists iff the diagram below can be constructed where both squares are pushouts in $\mathcal{CG}$.

$$\begin{tikzcd}
L \arrow[d, "g"'] & K \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "d"] & R \arrow[d, "h"] \\
G & D \arrow[l, "l*"] \arrow[r, "r*"'] & H
\end{tikzcd}$$

It is well known that although $\mathcal{CG}$ has all pushouts, given $l$ and g, we do not necessarily have a $D$ (the pushout complement of $l$ and $g$), or even if we do, it is not necessarily unique. We have a unique $K \to D \to G$ (up to isomorphism, and given that morphisms are injective) iff

(DANGL) $\quad e \in E_G - g(E_L)$, and $\{s(e), t(e)\} \cap g(V_L) \neq \varnothing$
$\qquad \Rightarrow \{s(e), t(e)\} \cap g(V_L) \subseteq g(l(V_K))$

If this holds we can write down a canonical expression for $D$

$$D = G - g(L - l(K))$$

which is shorthand for two expressions, one for edges and one for vertices.

Let us further agree on some canonical construction (see eg. the appendix) for the true pushout used in the right hand square of the rewrite. It will not matter that we might choose independent canonical constructions for pushout complement and pushout.

With such choices, rewriting becomes deterministic, a key ingredient for making an opfibration-based rewriting construction. Besides this it is easy to see that if a rewrite exists at all, then the arrow $h : R \to H$ satisfies (an appropriately instantiated version of) (DANGL). This has been known from the earliest days of the DPO approach.

Consider now the category $\mathcal{CP}$ (of "concrete patterns") where objects are arrows $l : K \to L$ of $\mathcal{CG}$ and whose arrows are just coinitial ordered pairs of objects, which we can write as $(L \leftarrow K : l,\, r : K \to R)$ or just $(l, r)$ where no confusion arises, and where identities are $(L \leftarrow K : l,\, l : K \to L)$, and composition of $(L_1 \leftarrow K : l_1,\, l_2 : K \to L_2)$ and $(L_2 \leftarrow K : l_2,\, l_3 : K \to L_3)$ is just $(L_1 \leftarrow K : l_1,\, l_3 : K \to L_3)$.

Above each $l : K \to L$ in $\mathcal{CP}$ we erect the discrete category $\mathcal{CR}^l$ (of "concrete redexes of $l$") whose objects are arrows $g : L \to G$ of $\mathcal{CG}$ such that in $g \circ l : K \to L \to G$, $g$ and $l$ satisfy the application condition (DANGL).

**Theorem 2.4** Let $(l, r)$ be an arrow of $\mathcal{CP}$. Then there is a functor $CRew^{(l,\, r)} : \mathcal{CR}^l \to \mathcal{CR}^r$ whose action on redexes is to send $g \circ l : K \to L \to G$ to $h \circ r : K \to R \to H$ where the latter is the result of a direct derivation using $(l, r)$.

*Proof.* Since $\mathcal{CR}^l$ is discrete and rewriting is deterministic, we just need to check that the result is an object of $\mathcal{CR}^r$, i.e. that (DANGL) applies; but we observed that this was so above. ☺

The next step is to attempt to glue the concrete rewriting functors together to get a functor from $\mathcal{CP} \to \mathcal{C}at$. Unfortunately this does not work properly as $CRew^{(l,\, l)}$ is not the identity functor on $\mathcal{CR}^l$ and neither is $CRew^{(l_2,\, l_3)} \circ CRew^{(l_1,\, l_2)}$ equal to $CRew^{(l_1,\, l_3)}$ due to the explicit nature of the rewriting construction. Nevertheless there are canonical isomorphisms from

$$CRew^{(l,\, l)}(g : L \to G) \quad \text{to} \quad (g : L \to G)$$

and from

$$CRew^{(l_2,\, l_3)} \circ CRew^{(l_1,\, l_2)}(g : L \to G) \quad \text{to} \quad CRew^{(l_1,\, l_3)}(g : L \to G).$$

These we can exploit in moving away from concrete rewriting, and going to a more abstract formulation. Specifically we can choose a skeleton category of $\mathcal{CG}$ in which these canonically isomorphic pairs of concrete redexes are represented by the same skeleton redex. Because of the importance of this point, the appendix gives some details of one specific construction which leads to the required properties. Thence or otherwise we get the following situation.

We have a skeleton category of graphs and injective morphisms $\mathcal{SG}$. From this we build the pattern category $\mathcal{P}$ of canonical arrows and coinitial pairs which we write as $\langle l, r\rangle$. Above each $l : K \to L$ in $\mathcal{P}$ we erect the discrete category of canonical redexes $\mathcal{R}^l$ for which the objects are $g : L \to G$ satisfying the application condition and where $G$ is a skeleton graph.

Theorem 2.4 now translates to a functor

$$Rew^{\langle l,\, r\rangle} : \mathcal{R}^l \to \mathcal{R}^r$$

In addition, these functors now compose together nicely and we have:

**Theorem 2.5** There is a functor $Rew : \mathcal{P} \to \mathcal{C}at$ such that

$$Rew(l) = \mathcal{R}^l$$

$$Rew(\langle l,\, r\rangle) = Rew^{\langle l,\, r\rangle} : \mathcal{R}^l \to \mathcal{R}^r$$

Thus $Rew$ maps an object of $\mathcal{P}$, $l : K \to L$, to its category of canonical redexes like $g : L \to G$ satisfying the application condition; and $Rew$ maps an arrow of $\mathcal{P}$, i.e. a double pushout rule $(L \leftarrow K : l,\, r : K \to R)$ to the skeletonised rewriting construction that turns objects $g : L \to G$ in $\mathcal{R}^l$ into the corresponding rewritten objects $h : R \to H$ in $\mathcal{R}^r$.

Given the above, we can apply the discrete Grothendieck construction to the functor $Rew$ to obtain the Grothendieck category $\mathcal{G}(\mathcal{P}, Rew)$ where objects are pairs formed from an object of the base $\mathcal{P}$, $l : K \to L$ say, and an object in the category above it $\mathcal{R}^l$, $g : L \to G$ say. We will write these pairs as $\langle g \circ l : K \to L \to G\rangle$. The arrows of $\mathcal{G}(\mathcal{P}$, Rew) are canonical rewrites; thus

$$\langle l,\, r\rangle : \langle g \circ l : K \to L \to G\rangle \to \langle h \circ r : K \to R \to H\rangle$$

is an arrow iff the $\mathcal{P}$ arrow (rewrite rule), transforms $g \circ l$ into $h \circ r$ in the expected way. The projection functor $F : \mathcal{G}(\mathcal{P}, Rew) \to \mathcal{P}$ given by

$$F(\langle g \circ l : K \to L \to G\rangle) = (l : K \to L)$$

$$\begin{aligned} F(\langle l,\, r\rangle &: \langle g \circ l : K \to L \to G\rangle \to \langle h \circ r : K \to R \to H\rangle) \\ &= \langle l,\, r\rangle = (L \leftarrow K : l,\, r : K \to R) \end{aligned}$$

is a split opfibration (Barr and Wells (1990)).

## 3 REWRITING CATEGORIES

From now on we only ever deal with skeleton graphs and canonical rewrites. In this section we construct three categories related to $\mathcal{G}(\mathcal{P}, Rew)$, and explore functorial relationships between them. They enable us to view rewriting in the DPO approach from a number of angles.

Consider the following construction. Let $\mathcal{D}$ be a small category, and let $\approx$ be an equivalence relation on objects of $\mathcal{D}$. Construct the directed graph $D^*$ whose vertices are equivalence classes $[d]$ of objects of $\mathcal{D}$ modulo $\approx$, and whose arcs are all arrows of $\mathcal{D}$, $\langle x : a \to b\rangle : [d] \to [e]$, where $a \in [d]$ and $b \in [e]$. Let $\mathcal{E}^\infty$ be the free category on $D^*$. As is well known, its objects are the $[d]$, and its arrows are sequences $[\alpha_n ; \alpha_{n-1} ; \ldots ; \alpha_1] : [d_0] \to [d_n]$, where $\langle \alpha_i : a_{i-1} \to b_i\rangle : [d_{i-1}] \to [d_i]$ is an arc of $D^*$; with the empty sequence $[\,] : [d] \to [d]$ as identity on a given object $[d]$. Composition of arrows is just composition of sequences, so that arrows of $\mathcal{E}^\infty$ have a natural notion of length. In general, $\mathcal{E}^\infty$ is not a functorial image of $\mathcal{D}$ — apart from anything else, identities do not map to identities.

Applying this to $\mathcal{G}(\mathcal{P}, Rew)$ with the equivalence

$$\langle g \circ l : K \to L \to G\rangle \approx \langle h \circ r : K \to R \to H\rangle \quad \text{iff} \quad G = H$$

we obtain (a category isomorphic to) the category $\mathcal{G}^\infty$. The objects of $\mathcal{G}^\infty$ are graphs $G$, and its arrows are sequences of rewrites:

$$[\varepsilon_n\,;\,\varepsilon_{n-1}\,;\,\ldots\,;\,\varepsilon_1] : G_0 \to G_n$$

where each $\varepsilon_i$ is of the form

$$\langle g_{i-1} \circ l_{i-1} : K_i \to L_{i-1} \to G_{i-1},\, g_i \circ l_i : K_i \to L_i \to G_i\rangle$$

which is the data for an arrow of $\mathcal{G}(\mathcal{P}, Rew)$ rearranged a bit.

Returning to the general construction on $\mathcal{D}$, we observe the following. Given that arrows of $\mathcal{D}$ are composable, we define a functor $B : \mathcal{E}^\infty \to \mathcal{E}$ exploiting this as follows.

$$\begin{aligned}
&B([u]) = [u]\\
&B([\,] : [u] \to [u]) = [\mathrm{id}_{[u]}] : [u] \to [u]\\
&B([\langle \mathrm{id}_a : a \to a\rangle] : [u] \to [u]) = [\mathrm{id}_{[u]}] : [u] \to [u] \text{ if } a \in [u]\\
&B(([\langle \gamma : c \to d\rangle] : [v] \to [w])\,;\,([\langle \alpha : a \to b\rangle] : [u] \to [v])) =\\
&\qquad [\mathrm{id}_{[u]}] : [u] \to [u] \text{ if } a = d,\ b = c,\ a \in [u], \text{ and}\\
&\qquad\qquad \langle \gamma : c \to d\rangle \circ \langle \alpha : a \to b\rangle = \mathrm{id}_a : a \to a, \text{ else}\\
&\qquad [\langle \gamma \circ \alpha : a \to d\rangle] : [u] \to [w] \text{ if } b = c, \text{ else}\\
&\qquad [(\langle \gamma\rangle : [v] \to [w])\,;\,(\langle \alpha\rangle : [u] \to [v])] : [u] \to [w] \text{ otherwise}
\end{aligned}$$

and we extend the above recursively to arrows of $\mathcal{E}^\infty$ of length $> 2$ in the only possible way that makes $B$ into a functor. Thus $\mathcal{E}$ has the objects of $\mathcal{E}^\infty$, formal identities, and arrows which are sequences as in $\mathcal{E}^\infty$, but where no adjacent pair of elements is composable in $\mathcal{D}$.

Composition in $\mathcal{E}$ for non-identity arrows can be succinctly described by

$$[\varepsilon_n\,;\,\ldots\,;\,\varepsilon_1] \circ [\delta_m\,;\,\ldots\,;\,\delta_1] = B([\varepsilon_n\,;\,\ldots\,;\,\varepsilon_1] \circ [\delta_m\,;\,\ldots\,;\,\delta_1])$$

where on the LHS, $\circ$ is the composition being defined, and on the RHS, we regard the two arrows as members of $\mathcal{E}^\infty$ for convenience.

There is now a functor

$$T : \mathcal{D} \to \mathcal{E}$$

acting by

$$\begin{aligned}
&T(d) = [d]\\
&T(\alpha : a \to b) \;=\; [\mathrm{id}_{[a]}] : [a] \to [a] \text{ if } \alpha = \mathrm{id}_a\\
&\qquad\qquad\qquad\quad [\langle \alpha : a \to b\rangle] : [a] \to [b] \text{ otherwise.}
\end{aligned}$$

We can apply the above generality to $\mathcal{G}(\mathcal{P}, Rew)$ and $\mathcal{G}^\infty$, and we get a category (isomorphic to the category) whose objects are graphs $G$, and whose non-identity arrows are sequences of rewrites

$$[\varepsilon_n\,;\,\ldots\,;\,\varepsilon_1] : G_0 \to G_n$$

where each $\varepsilon_i$ is a rewrite $\langle\langle g \circ l\rangle, \langle h \circ r\rangle\rangle$ as previously, except that $l \neq r$ or $g \neq h$ (or both), i.e. $\varepsilon_i$ is not an identity; and for each adjacent pair $(\varepsilon_i\,;\,\varepsilon_{i-1})$, $\varepsilon_i$ and $\varepsilon_{i-1}$ are not composable in $\mathcal{G}(\mathcal{P}, Rew)$.

The preceding deliberations give us functors

$$T : \mathcal{G}(\mathcal{P}, Rew) \to \mathcal{G}$$

and

$$B : \mathcal{G}^{\infty} \to \mathcal{G}$$

We also find a functor

$$S : \mathcal{G} \to \mathcal{G}(\mathcal{P}, Rew)$$

which takes objects $G$ of $\mathcal{G}$ to objects $\langle \mathrm{id}_G \circ \varnothing : \varnothing \to G \to G \rangle$ of $\mathcal{G}(\mathcal{P}, Rew)$, and acts on arrows by

$$\begin{aligned} &S([\varepsilon_n ; \ldots ; \varepsilon_1] : G \to H) = \\ &\qquad \langle \varnothing, \varnothing \rangle : \langle \mathrm{id}_G \circ \varnothing : \varnothing \to G \to G \rangle \to \langle \mathrm{id}_H \circ \varnothing : \varnothing \to H \to H \rangle \end{aligned}$$

The existence of $S$ is attributable to the extremely flexible garbage collection facilities of the DPO approach. One is free to regard the whole of the start graph of a rewrite as garbage, discard it, and replace it with the RHS graph of a suitable rule if one wishes. As a consequence, $S$ discards all of the interesting information in an arrow of $\mathcal{G}$. Pulling back via $B$ we find

$$S^{\infty} : \mathcal{G}^{\infty} \to \mathcal{G}(\mathcal{P}, Rew)$$

The situation so far is depicted below, where we leave out $T$ because $B \neq T \circ S^{\infty}$.

$$\begin{tikzcd} \mathcal{G}^{\infty} \arrow[r, "B"] \arrow[d, "S^{\infty}"'] & \mathcal{G} \arrow[dl, "S"] \\ \mathcal{G}(\mathcal{P}, Rew) & \end{tikzcd}$$

It is easy to check that both $T \circ S$ and $S \circ T$ are idempotent endofunctors. That $B \neq T \circ S^{\infty}$ is connected to the fact that there are no monads on $\mathcal{G}(\mathcal{P}, Rew)$ and $\mathcal{G}$ resolved by $T$ and $S$. In fact the identities on the prospective monads both fail. In the end what $T$ and $S$ express, is the fact that there is a canonical preorder subcategory $\mathcal{G}_0$ of $\mathcal{G}$ including an arrow $\langle\langle \mathrm{id}_G \circ \varnothing \rangle, \langle \mathrm{id}_H \circ \varnothing \rangle\rangle : G \to H$ between every pair of objects of $\mathcal{G}$, and that $\mathcal{G}(\mathcal{P}, Rew)$ contains a subcategory $\mathcal{G}(\mathcal{P}, Rew)_0$ which is isomorphic to $\mathcal{G}_0$.

Now in appendix B, we describe the R-composition of a pair of rewrites $\langle\langle g_1 \circ l_1 : K_1 \to L_1 \to G_1 \rangle, \langle j_1 \circ r_1 : K_1 \to R_1 \to J \rangle\rangle$ and $\langle\langle j_2 \circ l_2 : K_2 \to L_2 \to J \rangle, \langle h_2 \circ r_2 : K_2 \to R_2 \to H \rangle\rangle$ to produce a single rewrite $\langle\langle g \circ l : K \to L \to G \rangle, \langle h \circ r : K \to R \to H \rangle\rangle$. The abstract version is associative as discussed there, and so leads to the category $\mathcal{C}$. The objects of $\mathcal{C}$ are graphs $G$ and the arrows are single rewrites $\langle\langle g \circ l \rangle, \langle h \circ r \rangle\rangle$ under R-composition. The identity on a graph $G$ is easily seen to be $\langle\langle \mathrm{id}_G \circ \mathrm{id}_G : G \to G \to G \rangle, \langle \mathrm{id}_G \circ \mathrm{id}_G : G \to G \to G \rangle\rangle$.

This observation leads to a flurry of fresh functors (well three actually).

First

$$E : \mathcal{G}(\mathcal{P}, Rew) \to \mathcal{C}$$

acts by

$$\begin{aligned}
&E(\langle g \circ l : K \to L \to G\rangle) = G \\
&E(\langle l, r\rangle : \langle g \circ l : K \to L \to G\rangle \to \langle h \circ r : K \to R \to H\rangle) = \\
&\qquad\qquad \langle\langle g \circ l\rangle, \langle h \circ r\rangle\rangle
\end{aligned}$$

Second

$$C : \mathcal{G} \to \mathcal{C}$$

acts by

$$\begin{aligned}
&C(G) = G \\
&C([\mathrm{id}_G] : G \to G) = \langle\langle \mathrm{id}_G \circ \mathrm{id}_G : G \to G \to G\rangle, \langle \mathrm{id}_G \circ \mathrm{id}_G : G \to G \to G\rangle\rangle \\
&C([\varepsilon_n ; \ldots ; \varepsilon_1] : G_0 \to G_n) = \varepsilon_n \circ \ldots \circ \varepsilon_1 : G_0 \to G_n
\end{aligned}$$

where the composition on the RHS is in $\mathcal{C}$. Pulling back by $B$ gives us the third,

$$C^\infty : \mathcal{G}^\infty \to \mathcal{C}$$

defined by

$$\begin{aligned}
&C^\infty(G) = G \\
&C^\infty([\,] : G \to G) = \langle\langle \mathrm{id}_G \circ \mathrm{id}_G : G \to G \to G\rangle, \langle \mathrm{id}_G \circ \mathrm{id}_G : G \to G \to G\rangle\rangle \\
&C^\infty([\varepsilon_n ; \ldots ; \varepsilon_1] : G_0 \to G_n) = \varepsilon_n \circ \ldots \circ \varepsilon_1 : G_0 \to G_n
\end{aligned}$$

which is almost the same formula as for $C$, modified by the obvious action for identities of $\mathcal{G}^\infty$. We summarise the new situation in the diagram below.

$$\begin{tikzcd}
\mathcal{G}^\infty \arrow[r, "B"] \arrow[dr, "C^\infty"'] & \mathcal{G} \arrow[d, "C"] \\
\mathcal{G}(\mathcal{P}, Rew) \arrow[ur, "T"] \arrow[r, "E"'] & \mathcal{C}
\end{tikzcd}$$

Note that we cannot "unify" this diagram with the preceding one both because of the previously pointed out properties of $T$, and also because $E \circ S \neq C$ in general. We may regard the category $\mathcal{C}$ as giving a result semantics for rewriting sequences of $\mathcal{G}^\infty$ or $\mathcal{G}$, as in a single arrow, we get the result of all the rewrites. Ironically, if a rewriting sequence $[\varepsilon_n ; \ldots ; \varepsilon_1] : G_0 \to G_n$ transforms $G_0$ sufficiently drastically, so that nothing is left of $G_0$ at the end, then $\varepsilon_n \circ \ldots \circ \varepsilon_1$ in $\mathcal{C}$ will agree with $E \circ S([\varepsilon_n ; \ldots ; \varepsilon_1])$ after all.

## 4 EVENT SEMANTICS

In this section, we look at the consequences of sequential independence for rewriting sequences in the categorical framework we have set up. We consider $\mathcal{G}^\infty$ first.

Let $[\varepsilon_n ; \ldots ; \varepsilon_{i+1} ; \varepsilon_i ; \ldots ; \varepsilon_1] : G_0 \to G_n$ be an arrow of $\mathcal{G}^\infty$ and suppose $\varepsilon_{i+1} ; \varepsilon_i$ is

$$\langle\langle g_i \circ l_i : K_{i+1} \to L_i \to G_i\rangle, \langle h_{i+1} \circ r_{i+1} : K_{i+1} \to R_{i+1} \to G_{i+1}\rangle\rangle \ ; \\ \langle\langle g_{i-1} \circ l_{i-1} : K_i \to L_{i-1} \to G_{i-1}\rangle, \langle h_i \circ r_i : K_i \to R_i \to G_i\rangle\rangle$$

Now the second rewrite, $\langle\langle g_i \circ l_i\rangle, \langle h_{i+1} \circ r_{i+1}\rangle\rangle$ determines arrows $l_i{}^* : D_{i+1} \to G_i$ and $r_{i+1}{}^* : D_{i+1} \to G_{i+1}$ as in appendix A, and thus a partial isomorphism, $\tau_{Gi,Gi+1} = r_{i+1}{}^* \circ l_i{}^{*-1} : G_i \nrightarrow G_{i+1}$ as in appendix B. If there is a canonical redex of the LHS of the first rule $\langle l_{i-1}\ , r_i\rangle$ into $D_{i+1}$, i.e. a $\langle q \circ l_{i-1} : K_i \to L_{i-1} \to D_{i+1}\rangle$, then there will be a canonical redex of $\langle l_{i-1}\ , r_i\rangle$ into $G_i$ and a canonical redex of $\langle l_i\ , r_{i+1}\rangle$ into $G_{i-1}$. We then say that $\varepsilon_{i+1}\ ; \varepsilon_i$ are sequentially independent, and rewriting the new redexes instead of $\varepsilon_{i+1}\ ; \varepsilon_i$ gives us $[\varepsilon'_i\ ; \varepsilon'_{i+1}] : G_{i-1} \to G_{i+1}$ and hence a new arrow $[\varepsilon_n\ ; \dots ; \varepsilon'_i\ ; \varepsilon'_{i+1}\ ; \dots ; \varepsilon_1] : G_0 \to G_n$. This forms the basis of our event semantics.

**Lemma 4.1** Let $[\varepsilon'_1\ ; \varepsilon'_2] : G_0 \to G_2$ be obtained from $[\varepsilon_2\ ; \varepsilon_1] : G_0 \to G_2$ by swapping sequentially independent rewrites. Then (using the notation above), $[\varepsilon'_1\ ; \varepsilon'_2] = [\varepsilon_2\ ; \varepsilon_1]$ iff $l_0 = l_1$ and $r_1 = r_2$.

*Proof.* If $[\varepsilon'_1\ ; \varepsilon'_2] = [\varepsilon_2\ ; \varepsilon_1]$ then $l_0 = l_1$ and $r_1 = r_2$ by inspection. The converse is trivial on observing that given a rule and a skeleton graph, rewriting if possible, is of the unique canonical redex and gives a unique result. ☺

**Definition 4.2** Let $[\varepsilon_2\ ; \varepsilon_1] : G_0 \to G_2$ be an arrow of $\mathcal{G}^\infty$. If $[\varepsilon'_1\ ; \varepsilon'_2] : G_0 \to G_2$ is obtained from $[\varepsilon_2\ ; \varepsilon_1] : G_0 \to G_2$ by swapping sequentially independent rewrites, and $[\varepsilon'_1\ ; \varepsilon'_2] \neq [\varepsilon_2\ ; \varepsilon_1]$, then we say that $\varepsilon_1$ and $\varepsilon_2$ are independent. Otherwise they are dependent. Similarly if $\varepsilon_1$ and $\varepsilon_2$ are adjacent rewrites in a longer arrow of $\mathcal{G}^\infty$.

Note that our definition forces the following situations to be regarded as dependent: (a), two rewrites by an autoconcurrent non-consumptive rule that does not damage its redex, (thus in a concrete rewriting model, allowing both rewrites of the same redex to be done simultaneously); (b), two rewrites where in a concrete rewriting model, there are two disjoint concrete redexes for the same rule, differing by an automorphism of the rewritten graph; (c), two rewrites by the same rule where the RHS of the rule creates a fresh redex for itself, (i.e. where the rule supports a non-trivial morphism $y : L \to R$). In some of these cases, one would prefer to regard the situation as concurrent, in others not. But due to our using a deterministic rewriting model with canonical redexes, all must be deemed dependent, otherwise a rewriting event could become the same as a predecessor or successor, and the identity of individual events would be lost.

**Definition 4.3** Let $\theta = [\varepsilon_n\ ; \dots ; \varepsilon_1] : G_0 \to G_n$ be an arrow of $\mathcal{G}^\infty$ and let $\Theta$ be the set of arrows of $\mathcal{G}^\infty$ generated by permuting independent rewrites in $\theta$, and in arrows generated thereby. An event on $\Theta$ is a function $e : \Theta \to [1\dots n]$ such that if $\theta$ and $\psi$ are related by swapping independent $\varepsilon_{i+1}\ ; \varepsilon_i$ in $\theta$, and $e(\theta) = i+1$ and $f(\theta) = i$, then $e(\psi) = i$ and $f(\psi) = i+1$, and $c(\theta) = c(\psi)$ for all other events $c$ on $\Theta$. (Clearly there are exactly $n$ events on $\Theta$.) Let the set of events on $\Theta$ be $E$. We write $e \leq f$ iff $e(\varphi) \leq f(\varphi)$ for all $\varphi \in \Theta$.

Let $\mathrm{E} = (E, \leq)$. Then $\mathrm{E}$ is an elementary event structure (Nielsen et al. (1981)), the relation $\leq$ is causal dependency, and the various sequences $\theta \in \Theta$ correspond to monotonically increasing sequences of left-closed subsets (configurations) of $E$. (Note that because we deal with finite sequences throughout, our elementary event structures, and later our prime event structures, have the expected finitary properties.) In this way we get the functor

$$M^\infty : \mathcal{G}^\infty \to \mathcal{T}^\infty$$

Here, $\mathcal{T}^\infty$ has as objects graphs $G$, and as arrows elementary event structures. $M^\infty$ takes an arrow of $\mathcal{G}^\infty$ first to its equivalence class $\Theta$, and then to the elementary event structure of definition 4.3. Composition of arrows of $\mathcal{G}^\infty$ leads to a law of composition in $\mathcal{T}^\infty$ which is associative because concatenation of sequences is. Identity arrows of $\mathcal{T}^\infty$ are empty elementary event structures.

The situation with the category $\mathcal{G}$ is a little more interesting. Suppose $[\varepsilon_3 ; \varepsilon_2 ; \varepsilon_1] : G_0 \to G_3$ is an arrow of $\mathcal{G}$ and that $\varepsilon_2 ; \varepsilon_1$ are independent. Swapping them, getting say $[\varepsilon_3 ; \varepsilon'_1 ; \varepsilon'_2]$, we may find that $\varepsilon_3 ; \varepsilon'_1$ are internally composable, so that the swap in reality produces a shorter arrow $[\varepsilon_4 ; \varepsilon'_2]$. In such a case, $\varepsilon_3 ; \varepsilon_2$ must also be independent, and doing that swap instead, produces another arrow in the same equivalence class as $[\varepsilon_4 ; \varepsilon'_2]$. We now get the functor

$$M : \mathcal{G} \to \mathcal{T}$$

Again, $\mathcal{T}$ has as objects graphs, and as arrows elementary event structures. In general, composable pairs of rewrites scattered through an arrow $\theta$ of $\mathcal{G}$, will correspond to causally related pairs with no intervening event caused by the first and causing the second. Such pairs will also be independent of the same sets of other events, and lead to a well defined notion of smallest elementary event structure corresponding to $\theta$, obtained when all possible such internal compositions are performed after appropriate permutations of rewrites in $\theta$. This smallest elementary event structure is the arrow of $\mathcal{T}$ corresponding to $\theta$. There is also a functor

$$B_M : \mathcal{T}^\infty \to \mathcal{T}$$

that does the composition at the event level. In effect it consists of abstraction morphisms on individual elementary event structures, where chains of composable events in arrows of $\mathcal{T}^\infty$ are mapped onto single events in arrows of $\mathcal{T}$. By ensuring that even non-adjacent composable rewrites in arrows of $\mathcal{G}$ are composed in $\mathcal{T}$, $\mathcal{T}$ provides a particularly good notion of abstract semantics for $\mathcal{G}$. The situation is described by the following diagram,

$$\begin{tikzcd}
\mathcal{T}^\infty \arrow[r, "B_M"] & \mathcal{T} \arrow[r, "C_M"] & \mathcal{T}_C \\
\mathcal{G}^\infty \arrow[u, "M^\infty"] \arrow[r, "B"'] & \mathcal{G} \arrow[u, "M"'] \arrow[r, "C"'] & \mathcal{C} \arrow[u, "M_C"']
\end{tikzcd}$$

where in the RHS square, the arrows of $\mathcal{T}_C$ are one-event elementary event structures corresponding to the single rewrites of arrows of $\mathcal{C}$, and $C_M$ and $M_C$ are the obvious functors.

So far our event semantics has had no need for notions of conflict, which is appropriate for considering a single arrow and its equivalence class. Now starting first with $\mathcal{G}^\infty$ as before, let $\Theta : G \to H_1$ and $\Psi : G \to H_2$ be equivalence classes of arrows of $\mathcal{G}^\infty$, with corresponding arrows $\mathrm{E} = (E, \leq_E) : G \to H_1$ and $\mathrm{F} = (F, \leq_F) : G \to H_2$ of $\mathcal{T}^\infty$. We define the relation $\approx$ on events in E and F by

$$e \approx f \quad \text{iff} \quad \begin{array}{l}\text{there is a } \theta \in \Theta \text{ and } \psi \in \Psi \text{ such that } \theta \text{ and } \psi \text{ have a} \\ \text{common prefix } \varphi \text{, and } e(\theta) = f(\psi) \leq \text{length}(\varphi)\end{array}$$

Clearly $\approx$ extends to an isomorphism of left-closed substructures of E and F. What we really want is the "colimit" of E and F along this common part. Now the events in E and F are all distinct due to the quantity of information packed within them, so $E \cap F = \emptyset$ unless $\Theta = \Psi$. We exploit this in order to make a construction without the usual tags. Let

$$X = (E \cup F)/\approx$$

$$x \leq_X y \ \text{ iff } \ \exists\, e \in x, f \in y, \text{ such that } e \leq_E f \text{ or } e \leq_F f$$

$$x \,\#_X\, y \ \text{ iff } \ x \subseteq E \text{ and } y \subseteq F, \text{ or } x \subseteq F \text{ and } y \subseteq E$$

Thus members of $X$ are equivalence classes of $E \cup F$ under $\approx$ ; $x \leq_X y$ (for $\{x, y\} \subseteq X$) whenever members of $x$ and $y$ are causally related via E or F; and $x \,\#_X\, y$ whenever $x$ and $y$ are singletons unrelated via $\approx$. $\mathrm{X} = (X, \leq_X, \#_X)$ is now a prime event structure, and the various sequences in $\Theta \cup \Psi$ correspond to monotonically increasing sequences of coherent left-closed subsets (configurations) of $X$.

For larger sets of equivalence classes of coinitial arrows of $\mathcal{G}^\infty$, $\{\Theta_\lambda : G \to H_\lambda\}_{\lambda \in \Lambda}$ with corresponding elementary event structures $\{(E_\lambda, \leq_\lambda)\}_{\lambda \in \Lambda}$ we just generalise the above. The definition of $\approx$ is as before (with $\Theta$, $\Psi$ ranging over arbitrary $\Theta_\lambda$). Let $X = \bigcup E_\lambda/\approx$, and define the support of $x \in X$ as $\text{supp}(x) = \{E_\lambda \mid \exists\, e \in x \text{ such that } e \in E_\lambda\}$. Now $x \leq_X y$ iff $e \leq_\lambda f$ for some $e \in x, f \in y$ with $\{e, f\} \subseteq E_\lambda$ for some $E_\lambda \in \text{supp}(x) \cap \text{supp}(y)$. Finally we set $x \,\#_X\, y$ iff $\text{supp}(x) - \text{supp}(y) \neq \emptyset$ and $\text{supp}(y) - \text{supp}(x) \neq \emptyset$. Again $\mathrm{X} = (X, \leq_X, \#_X)$ is a prime event structure. Regardless of the cardinality of $\Lambda$, we still have the same finitary properties as before since $\approx$ is based on finite prefixes. (Eg. we have finite causes (Winskel (1986, 1988)) since every event $e$ is causally determined by at most $\min\{e(\theta) \mid \theta \in \Theta\}$ predecessors.)

We can abuse language by using E for both an elementary event structure and for the prime event structure generated by E alone, confusing an event $e$ and its singleton $\{e\}$. We also get an associative operation for combining the prime event structures for collections of (equivalence classes of) coinitial arrows, (we just unpack the original $\approx$ equivalence classes, and then perform the construction above for the entire set of events thus exposed).

For $\mathcal{G}$, a similar construction works, and is most easily obtained by restricting consideration to those executions that are interleavings of precisely the minimal elementary event structures that are the arrows of $\mathcal{T}$. We note that when in $\Theta : G \to H_1$ we have (the $\mathcal{G}^\infty$ composition) $\varepsilon_3 \circ \varepsilon_2 \circ \varepsilon_1$, and in $\Psi : G \to H_2$ we have $\phi \,;\, \varepsilon_2 \circ \varepsilon_1$ (with $\phi$ non-composable with $\varepsilon_2$), then in the event structure colimit construction, the events corresponding to $\varepsilon_2 \circ \varepsilon_1$ and to $\varepsilon_3 \circ \varepsilon_2 \circ \varepsilon_1$ are in conflict, even though both are compositions and (in $\mathcal{G}^\infty$), one is a prefix of the other. Also for $\mathcal{C}$, the construction is fairly trivial.

Finally we point out that because DPO rewriting is symmetrical, a dual construction for cofinal collections of arrows is possible; but is generally regarded as less interesting.

# 5 GRAPH GRAMMARS AND GOOD GRANULARITY

A graph grammar $\mathcal{R}$ is a pair $(R, G_0)$ where $R$ is a (finite) collection of DPO rules, and $G_0$ is an initial graph. We can consider our preceding constructions in the light of a specific grammar $\mathcal{R}$. By $\mathcal{P}_R$ we denote the subcategory of $\mathcal{P}$ consisting of (canonical rules corresponding to) rules of $R$, and whatever compositions and additional identities are required. Similarly we get $\mathcal{G}(\mathcal{P}, Rew)_R$ consisting of those rewrites generated by $\mathcal{P}_R$ on arbitrary canonical redexes. We can now construct ${\mathcal{G}^{\infty}}^*_R$ consisting of those rewriting sequences where all rules used are from $\mathcal{P}_R$. This will contain a subcategory ${\mathcal{G}^{\infty}}^*_{\mathcal{R}}$ consisting of graphs derivable from (the skeletal graph of) $G_0$ using rules from $\mathcal{P}_R$, and ${\mathcal{G}^{\infty}}^*_{\mathcal{R}}$ will contain a subcategory $\mathcal{G}^{\infty}_{\mathcal{R}}$ where all rules used in arrows are just those from $R$, excluding any extraneous compositions or identities. There will be a subcategory $\mathcal{G}(\mathcal{P}, Rew)_{\mathcal{R}}$ of $\mathcal{G}(\mathcal{P}, Rew)_R$ consisting of just those rewrites that contribute to ${\mathcal{G}^{\infty}}^*_{\mathcal{R}}$. The categories $\mathcal{G}_R$ and $\mathcal{G}_{\mathcal{R}}$ are got from ${\mathcal{G}^{\infty}}^*_R$ and ${\mathcal{G}^{\infty}}^*_{\mathcal{R}}$ by doing all internal compositions; while the categories $\mathcal{C}_R$ and $\mathcal{C}_{\mathcal{R}}$ contain the single rewrite equivalent of all derivations using only rules in $\mathcal{P}_R$ or $R$ (the distinction now disappears), and all derivations of $\mathcal{R}$ respectively. The various functors considered in section 3 restrict in a predictable way to the subcategories we have mentioned.

The event semantics of section 4 now supplies us with natural abstract semantics for a graph grammar $\mathcal{R}$. We consider all the arrows of ${\mathcal{G}^{\infty}}^*_{\mathcal{R}}$, $\mathcal{G}^{\infty}_{\mathcal{R}}$ and $\mathcal{G}_{\mathcal{R}}$ with domain $G_0$, and thence construct the associated event structures ${\mathrm{E}^{\infty}}^*_{\mathcal{R}} = ({E^{\infty}}^*_{\mathcal{R}}, {\le^{\infty}}^*_{\mathcal{R}}, {\#^{\infty}}^*_{\mathcal{R}})$ , $\mathrm{E}^{\infty}_{\mathcal{R}} = (E^{\infty}_{\mathcal{R}}, \le^{\infty}_{\mathcal{R}}, \#^{\infty}_{\mathcal{R}})$ and $\mathrm{E}_{\mathcal{R}} = (E_{\mathcal{R}}, \le_{\mathcal{R}}, \#_{\mathcal{R}})$, the configuration spaces of which will be finitary prime algebraic domains.

**Definition 5.1** A graph grammar $\mathcal{R} = (R, G_0)$ has good granularity, iff no rule of $R$ is composable with itself or with any other rule of $R$.

It is clear that the overwhelming majority of graph grammars used in practice will have good granularity. The main consequence of good granularity for a graph grammar $\mathcal{R}$, is that the restriction of the functor $B$ which it induces, namely $B_{\mathcal{R}} : \mathcal{G}^{\infty}_{\mathcal{R}} \to \mathcal{G}_{\mathcal{R}}$, is an isomorphism between $\mathcal{G}^{\infty}_{\mathcal{R}}$ and $\mathcal{G}_{\mathcal{R}}$, because there are no internal compositions. This in turn forces an isomorphism between $\mathrm{E}^{\infty}_{\mathcal{R}}$ and $\mathrm{E}_{\mathcal{R}}$.

# 6 DISCUSSION

In the preceding sections, we have reappraised the classical DPO construction within the opfibration framework. Subsequently we constructed various rewriting categories on the basis of a number of ways of combining rules and/or rewrites. Then we reinterpreted (equivalence classes under independence of) the arrows of these categories as elementatry event structures, leading to the further interpretation of suitable collections of coinitial arrows as event structures with conflict. At the end we focused on specific graph grammars to derive an abstract semantics for graph grammars via this route. Some comments are now in order.

To construct a split opfibration, we need both determinism and associativity. These properties are usually in conflict: determinism comes most naturally with a concrete nonassociative rewriting construction, while associativity comes most naturally when one considers rewriting up to isomorphism. We overcame this by using canonical redexes in a specific skeleton category. This is reasonable, even when graphs have nontrivial automorphisms, since then, up to isomorphism, the choice of one result for a rewrite is no better or worse than that of another which differs from the first by an automorphism. In this way we obtained the best of both worlds, in the process making some later constructions simpler. Nevertheless such a choice of strategy has specific consequences later on, particularly as regards the treatment of autoconcurrency and similar phenomena. For instance, given an autoconcurrent rule, and two successive rewrites using it, it is easy enough to distinguish the effects of the two individual rewrites in a concrete nonassociative rewriting construction, but when rewriting is via canonical redexes, there is no way of telling them apart, and one is forced to regard them as causally dependent if the individuality of events is not to be undermined, despite the fact that this might be felt to be counterintuitive. In order to analyse this carefully, we had to be quite precise about what an event was in our general framework, and under what circumstances two events could be regarded as distinct.

Our work in sections 3 and 4 bears comparison with other papers, notably Corradini et al. (1993a,b, 1994a,b,c,d) and Schied (1993). Unlike Corradini et al. (1993a,b, and 1994a,b,c), we use a skeleton category rather than equivalence classes of graphs/derivations. Of course, due to the equivalence between skeleton categories and ones constructed using suitable equivalences (see eg. Corradini et al. (1993a,b)), all that we have done has its counterpart in the alternative approach. The comparison with Schied's work is also instructive. Schied's concrete rewriting construction has aspects of both our concrete and abstract constructions. It is deterministic and carries some history in the structure of nodes while also being associative. The price one pays for this is the need to be confined to the graphs generated from a given start graph by a given grammar in order for the construction to satisfy Schied's definiteness condition. We overcame the same hurdle in a general setting by explicit choice of canonical entities. More generally, being based upon a "global" construction, our later constructions were not limited by the need for restriction to any particular class of graphs or of grammars, eg. the safe or the consumptive grammars.

## Appendix A

In this appendix we outline how to construct a model of labelled graphs from which we can extract a skeleton of finite graphs, such that for suitable collections of skeletal redexes, the compositionality of rewriting required by the Grothendieck construction holds.

We need supplies vertices and of edges, that are at least countable. We can take $\mathbf{N}$ for both of them. From these we can construct (say by a generate-and-test algorithm) a canonical representative for each possible edge and vertex labelled graph with labels coming from (the fixed countable) $\Omega_E$, $\Omega_V$. We call this collection of graphs (with injective morphisms as arrows) our skeleton $\mathcal{SG}$. We need to do two things. First to describe abstract rewriting of skeletal redexes in $\mathcal{SG}$, and then because $\mathcal{SG}$ is not closed under concrete rewriting, to show how its closure under that operation can be constructed so that suitable isomorphisms take concrete rewrites of skeletal redexes to abstract rewrites in a consistent way.

First part. In $\mathcal{SG}$, whenever there are arrows $l : K \to L$, $l' : K \to L$ such that $l$ and $l'$ differ by a non-trivial automorphism of $L$, choose a canonical such arrow, (otherwise the only arrow $l : K \to L$ up to isomorphisms of $L$ is canonical). With such a choice given, pushout squares in $\mathcal{SG}$ become unique if three out of the four arrows appearing there are canonical. An abstract DPO rewrite in $\mathcal{SG}$ is now one in which the arrows $l : K \to L$, $g : L \to G$, $r : K \to R$, $h : R \to H$, and $d : K \to D$ are canonical, and $l^* : D \to G$ and $r^* : D \to H$ are forced to be the only ones possible.

The rewrites now organise themselves into a Grothendieck construction fairly easily. $\mathcal{P}$ is the category with canonical arrows $l : K \to L$ as objects, and ordered pairs of canonical arrows $\langle L \leftarrow K : l, r : K \to R\rangle$ as arrows. Above each $l : K \to L$ of $\mathcal{P}$ we erect $\mathcal{R}^l$, the discrete category whose objects are canonical arrows $g : L \to G$ satisfying the application condition (DANGL). Rewriting such a $g$ using $\langle l, r\rangle$ yields a unique result, the canonical arrow $h : R \to H$ in $\mathcal{R}^r$ given by an abstract DPO rewrite, and so rewriting is a functor $Rew^{\langle l, r\rangle} : \mathcal{R}^l \to \mathcal{R}^r$. It is now clear that these functors glue together to give a functor $Rew : \mathcal{P} \to \mathcal{C}at$ as required.

Second part. Below we will be quite explicit about disjoint union: the disjoint union of $A$ and $B$ is

$$\{\langle 1, a\rangle \mid a \in A\} \cup \{\langle 2, b\rangle \mid a \in B\}$$

i.e. each element is tagged $\langle 1, -\rangle$ or $\langle 2, -\rangle$ depending on its orgin.

**Definition A.1** Let $f' : G \to G'$ and $f'' : G \to G''$ be morphisms of graphs. Their concrete pushout is the graph $C$ given by

$E_C = E' \uplus E'' / \approx$ where $\uplus$ is disjoint union and $\approx$ is the smallest equivalence relation such that $\langle 1, e'\rangle \approx \langle 2, e''\rangle$ whenever there is an $e \in E_G$ such that $f'_E(e) = e'$ and $f''_E(e) = e''$.

$V_C = V' \uplus V'' / \approx$ where $\uplus$ is disjoint union and $\approx$ is the smallest equivalence relation such that $\langle 1, v'\rangle \approx \langle 2, v''\rangle$ whenever there is an $v \in V_G$ such that $f'_V(v) = v'$ and $f''_V(v) = v''$.

$s_C$ is given by $s_C(\{\langle 1, e'\rangle \dots\}) = \{\langle 1, s'(e')\rangle \dots\}$
$s_C(\{\langle 2, e''\rangle \dots\}) = \{\langle 2, s''(e'')\rangle \dots\}$

$t_C$ is given by $t_C(\{\langle 1, e'\rangle \dots\}) = \{\langle 1, t'(e')\rangle \dots\}$
$t_C(\{\langle 2, e''\rangle \dots\}) = \{\langle 2, t''(e'')\rangle \dots\}$

$l_{E,C}$ is given by $l_{E,C}(\{\langle 1, e'\rangle \dots\}) = l'_E(e')$
$l_{E,C}(\{\langle 2, e''\rangle \dots\}) = l''_E(e'')$

$l_{V,C}$ is given by $l_{V,C}(\{\langle 1, e'\rangle \dots\}) = l'_V(e')$
$l_{V,C}(\{\langle 2, e''\rangle \dots\}) = l''_V(e'')$

In the above we use ellipsis to indicate that the equivalence class concerned needn't be necessarily a singleton, and where the various cases overlap, they are consistent.

With the above, a concrete rewrite of the redex $g : L \to G$ using rule $(L \leftarrow K : l, r : K \to R)$ produces the graph $H$ given by

$E_H = (E_G - g(E_L - l(E_K))) \uplus E_R / \approx$ where $\uplus$ is disjoint union and
$\approx$ is the smallest equivalence relation such that $\langle 1, x\rangle \approx \langle 2, y\rangle$ whenever there is an $e \in E_K$ such that $d_E(e) = x$ and $r_E(e) = y$, where $d : K \to (G - g(L - l(K)))$ is the usual map of $K$ into the pushout complement.

$V_H =$ a similar expression.

etc.

Now let $(L \leftarrow K : l, l : K \to L)$ be an identity rule. Applying it to a redex $g : L \to G$ produces $h : L \to H$ where $H$ is given by (we only quote edge expressions in future)

$$
\begin{aligned}
E_H = \ & \{\langle 1, e_G\rangle\} \text{ for } e_G \in E_G - g(E_L) \\
& \{\langle 2, e_L\rangle\} \text{ for } e_L \in E_L - g(E_K) \\
& \{\langle 1, e_G\rangle, \langle 2, e_L\rangle\} \text{ where there is an } e_K \in E_K \text{ such that} \\
& \qquad g \circ l(e_K) = e_G \text{ and } l(e_K) = e_L
\end{aligned}
$$

From this we get a canonical isomorphism $\phi^l{}_H : H \to G$ given by

$$
\begin{aligned}
& \phi^l{}_{H,E}(\{\langle 1, e_G\rangle\}) = e_G \text{ for } e_G \in E_G - g(E_L) \\
& \phi^l{}_{H,E}(\{\langle 2, e_L\rangle\}) = e_G \text{ for } e_L \in E_L - g(E_K) \text{ where } g(e_L) = e_G \\
& \phi^l{}_{H,E}(\{\langle 1, e_G\rangle, \langle 2, e_L\rangle\}) = e_G \text{ where there is an } e_K \in E_K \text{ such that} \\
& \qquad g \circ l(e_K) = e_G \text{ and } l(e_K) = e_L
\end{aligned}
$$

Note that if $g : L \to G$ was not injective, then $\phi^l{}_H$ would not be invertible.

Consider now two composable arrows $(l_1, l_2)$ and $(l_2, l_3)$ and their composition $(l_1, l_3)$. Apply these to a redex $g_1 : L_1 \to G_1$, getting in turn $g_2 : L_2 \to G_2$ and $g_3 : L_3 \to G_3$; or $h : L_3 \to H$ using $(l_1, l_3)$. We get:

$$
\begin{aligned}
E_{G_2} = \ & \{\langle 1, e_{G_1}\rangle\} \text{ for } e_{G_1} \in E_{G_1} - g_1(E_{L_1}) \\
& \{\langle 2, e_{L_2}\rangle\} \text{ for } e_{L_2} \in E_{L_2} - l_2(E_K) \\
& \{\langle 1, e_{G_1}\rangle, \langle 2, e_{L_2}\rangle\} \text{ where there is an } e_K \in E_K \text{ such that} \\
& \qquad g_1 \circ l_1(e_K) = e_{G_1} \text{ and } l_2(e_K) = e_{L_2}
\end{aligned}
$$

$$
\begin{aligned}
E_{G_3} = \ & \{\langle 1, \{\langle 1, e_{G_1}\rangle\}\rangle\} \text{ for } \{\langle 1, e_{G_1}\rangle\} \in E_{G_2} - g_2(E_{L_2}) \\
& \{\langle 2, e_{L_3}\rangle\} \text{ for } e_{L_3} \in E_{L_3} - l_3(E_K) \\
& \{\langle 1, \{\langle 1, e_{G_1}\rangle, \langle 2, e_{L_2}\rangle\}\rangle, \langle 2, e_{L_3}\rangle\} \text{ where there is an } e_K \in E_K \text{ such that} \\
& \qquad g_2 \circ l_2(e_K) = \{\langle 1, e_{G_1}\rangle, \langle 2, e_{L_2}\rangle\} \text{ and } l_3(e_K) = e_{L_3}
\end{aligned}
$$

$$
\begin{aligned}
E_H = \ & \{\langle 1, e_{G_1}\rangle\} \text{ for } e_{G_1} \in E_{G_1} - g_1(E_{L_1}) \\
& \{\langle 2, e_{L_3}\rangle\} \text{ for } e_{L_3} \in E_{L_3} - l_3(E_K)
\end{aligned}
$$

$\{\langle 1, e_{G_1}\rangle, \langle 2, e_{L_3}\rangle\}$ where there is an $e_K \in E_K$ such that
$g_1 \circ l_1(e_K) = e_{G_1}$ and $l_3(e_K) = e_{L_3}$

Once again there is a canonical isomorphism $\psi^{l_1,l_2,l_3}{}_{G_3} : G_3 \to H$ given by

$$\psi^{l_1,l_2,l_3}{}_{G_3,E}(\{\langle 1, \{\langle 1, e_{G_1}\rangle\}\rangle\}) = \{\langle 1, e_{G_1}\rangle\} \text{ for } \{\langle 1, e_{G_1}\rangle\} \in E_{G_2} - g_2(E_{L_2})$$
$$\psi^{l_1,l_2,l_3}{}_{G_3,E}(\{\langle 2, e_{L_3}\rangle\}) = \{\langle 2, e_{L_3}\rangle\} \text{ for } e_{L_3} \in E_{L_3} - l_3(E_K)$$
$$\psi^{l_1,l_2,l_3}{}_{G_3,E}(\{\langle 1, \{\langle 1, e_{G_1}\rangle, \langle 2, e_{L_2}\rangle\}\rangle, \langle 2, e_{L_3}\rangle\}) = \{\langle 1, e_{G_1}\rangle, \langle 2, e_{L_3}\rangle\} \text{ where}$$
$$\exists\, e_K \in E_K \text{ with } l_3(e_K) = e_{L_3} \text{ and } g_2 \circ l_2(e_K) = \{\langle 1, e_{G_1}\rangle, \langle 2, e_{L_2}\rangle\}$$

Now for the construction. Make $\mathcal{P}$ from $\mathcal{SG}$ as indicated above.

To deal with closure under rewriting, we construct categories $\mathcal{CR}^l{}_*$ in stages. Each stage is created by performing rewrites on the graphs created in the previous stage. Mostly this makes fresh objects by increasing nesting of the "tag and equivalence" operation of rewriting, but not always. (For consider a rewrite according to a rule $(L \leftarrow \varnothing : \varnothing, \varnothing : \varnothing \to R)$ with $g : L \to G$ onto. Regardless of the level of nesting in $G$, all of it is discarded in the pushout complement, and in the resulting $H$, everything is at level 1.) Along with the graph, we retain various other information to aid the construction.

<u>Stage 0</u>: For each object $l : K \to L$ of $\mathcal{P}$, we construct $\mathcal{CR}^l{}_0$, consisting of pairs $(g : L \to G, \theta_G)$ where $g$ is an arrow of $\mathcal{SG}$ satisfying the application condition for $l$, and $\theta_G$ is the unique automorphism on $G$ that takes $g : L \to G$ to the canonical redex $g_S : L \to G$.

<u>Stage 1</u>: For each arrow $(L \leftarrow K : l, r : K \to R)$ of $\mathcal{P}$ and each $(g : L \to G, \theta_G)$ in $\mathcal{CR}^l{}_0$, we (concretely) rewrite $g : L \to G$ using $(l, r)$ yielding $h : R \to H$.

Now for each object $m : J \to M$ of $\mathcal{P}$ such that there is a redex $t \circ m : J \to M \to H$ satisfying the application condition, we place in $\mathcal{CR}^m{}_1$, the pair $(t : M \to H, \theta_H)$ where $\theta_H$ is the unique isomorphism from $H$ to its skeletal representative $H_S$, that takes $t : M \to H$ to the canonical redex $t_S : M \to H_S$.

<u>Stage $k$+1</u>: For each arrow $(L \leftarrow K : l, r : K \to R)$ of $\mathcal{P}$ and each object $(g : L \to G, \theta_G)$ in $\mathcal{CR}^l{}_k$, we (concretely) rewrite $g : L \to G$ using $(l, r)$ yielding $h : R \to H$.

If $H$ has already been produced at an earlier stage, we take no further action with it. Otherwise, for each object $m : J \to M$ of $\mathcal{P}$ such that there is a redex $t \circ m : J \to M \to H$ satisfying the application condition, we place in $\mathcal{CR}^m{}_{k+1}$, the pair $(t : M \to H, \theta_H)$ where $\theta_H$ is the unique isomorphism from $H$ to its skeletal representative $H_S$, that takes $t : M \to H$ to the canonical redex $t_S : M \to H_S$.

Once the $\mathcal{CR}^l{}_k$ are all constructed, we get the $\mathcal{CR}^l{}_*$ as the union of $\mathcal{CR}^l{}_k$ over all $0 \le k < \infty$. We get $\mathcal{CR}^l$ from $\mathcal{CR}^l{}_*$ by forgetting the last component of each pair, and $\mathcal{CG}$ from $\bigcup_{l \in \mathcal{SG}} \mathcal{CR}^l$ by collecting together the codomains of the arrows, and equipping them with injective morphisms as arrows of $\mathcal{CG}$. For each $H$ in $\mathcal{CG}$, we get (possibly several) isomorphisms $\theta_H$ taking $H$ to its skeletal representative, by looking up the second component of any pair $(t : M \to H, \theta_H)$, and for the skeleton construction, we just pick one of them (unless $H$ is a skeleton graph itself, when we definitely choose $\mathrm{id}_H$). The fact that there might be more than one does not matter, since it is redexes not graphs that are the objects in the rewriting construction. We also note that the isomorphisms that we established above for composition and identities among concrete rewrites, ensure that the abstract rewriting construction is indeed an abstraction of the concrete one.

It is clear that the objects in $\mathcal{CG}$ are closed under rewriting by rules of $\mathcal{P}$. This not quite the same as being closed under all rules formed from all objects in $\mathcal{CG}$ but what we have done is harmless for two reasons. Firstly we are interested ultimately in restricting to rules in the pattern category $\mathcal{P}$ anyway. And secondly, it is easy to see that if $h' \circ r' : K' \to R' \to H'$ results via $(l', r')$ from $g' \circ l' : K' \to L' \to G'$ and $\theta_{K'} : K' \to K_S$ is the skeletal isomorphism for $K$, we can find the canonical arrows $l_S : K_S \to L_S$, and $r_S : K_S \to R_S$, and then subsequently $g_S : L_S \to G_S$, $h_S : R_S \to H_S$ which give the canonical representative of the rewrite, so the general construction will be an easy modification of the restricted case.

## Appendix B

Suppose, in the large category of all graphs, but still with our deterministic concrete rewriting construction, we have two DPO rewrites as below, where the rules used are arbitrary.

$$\begin{tikzcd}
L_1 \arrow[dr, "g_1"'] & & K_1 \arrow[ll, "l_1"'] \arrow[rr, "r_1"] \arrow[dr, "d_1"'] & & R_1 \arrow[dr, "j_1"'] & & L_2 \arrow[dl, "j_2"] & & K_2 \arrow[ll, "l_2"'] \arrow[rr, "r_2"] \arrow[dl, "d_2"] & & R_2 \arrow[dl, "h_2"] \\
& G & & D_1 \arrow[ll, "l_1*"] \arrow[rr, "r_1*"'] & & J & & D_2 \arrow[ll, "l_2*"] \arrow[rr, "r_2*"'] & & H &
\end{tikzcd}$$

We will deterministically construct a single rule, such that there is a single DPO rewrite deriving the graph $H$, from $G$ as below.

$$\begin{tikzcd}
L \arrow[d, "g"'] & K \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "d"] & R \arrow[d, "h"] \\
G & D \arrow[l, "l*"] \arrow[r, "r*"'] & H
\end{tikzcd}$$

The construction is really a particular restatement of the R-related composition of rewrites of Ehrig (1979). The process is illustrated in the following “St. Edward’s Crown” diagram.

Let $\tau_{G,J} = r_1{*} \circ l_1{*}^{-1}$. Then $\tau_{G,J}$ is well defined (because $l_1*$ is injective), and is a morphism of subgraph $l_1*(D_1)$ of $G$ to $J$; i.e. a partial morphism $\tau_{G,J} : G \nrightarrow J$. Likewise the partial mrophism $\tau_{J,H} : J \nrightarrow H$. Partial morphisms are composable, in particular $\tau_{G,H} = \tau_{J,H} \circ \tau_{G,J} : G \nrightarrow H$ has as domain $D = l_1{*} \circ r_1{*}^{-1} \circ l_2{*}(D_2)$. Let $l* : D \to G$ be the inclusion morphism and let injective $r* : D \to H$ be such that $\tau_{G,H} = r{*} \circ l{*}^{-1}$.

$$\begin{tikzcd}
 & & & K_{LR} \arrow[dl, "k_L"'] \arrow[dr, "k_R"] & & & \\
L_0 \arrow[dr, "\subseteq"'] \arrow[ddr, "\subseteq"] & & K_L \arrow[ll, "\subseteq"'] \arrow[dr, "k_{r*}"] \arrow[ddl, bend right=90, "\subseteq"'] & & K_R \arrow[rr, "\subseteq"] \arrow[dl, "k_{l*}"'] \arrow[ddr, bend left=90, "\subseteq"] & & R_0 \arrow[dl, "\subseteq"] \arrow[ddl, "\subseteq"'] \\
 & L \arrow[d, "\subseteq"] & & K \arrow[ll, "l"] \arrow[rr, "r"'] \arrow[d, "d"] & & R \arrow[d, "\subseteq"'] & \\
 & G \arrow[rrrr, leftrightarrow, bend right, "\tau_{G,H}"'] & & D \arrow[ll, "l*"'] \arrow[rr, "r*"] & & H &
\end{tikzcd}$$

Let $L_0$ be the “graph completion” of $G - l*(D)$ i.e.

$$E_{L_0} = E_G - E_{l*(D)}$$
$$V_{L_0} = (V_G - V_{l*(D)}) \cup s(E_{L_0}) \cup t(E_{L_0})$$

and let $K_L$ be $L_0 \cap l*(D)$ which is just the node set $V_{L_0} \cap V_{l*(D)}$ as these two subgraphs have no edges in common. Similarly we construct $R_0$ and $K_R = V_{R_0} \cap V_{r*(D)}$.

Now $\tau_{G,H}$ induces a partial bijection between $K_L$ and $K_R$. Let $K_{LR}$ be the set of related pairs, with $k_L : K_{LR} \to K_L$, $k_R : K_{LR} \to K_R$, the projections. Form the pushout of $k_L$ and $k_R$, yielding $k_{r*} : K_L \to K \leftarrow K_R : k_{l*}$, which are injective maps. Form the pushout of $K_L \subseteq L_0$ and $k_{r*}$, choosing the pushout object $L \subseteq G$, formed by identifying each element of $K$ with an element of $G$ via ${k_{l*}}^{-1}$, $K_R \subseteq H$, ${\tau_{G,H}}^{-1}$. This gives $L_0 \subseteq L \leftarrow K : l$. Likewise form $R_0 \subseteq R \leftarrow K : r$. We can now construct $d : K \to D$ uniquely and consistently.

Clearly $(L \leftarrow K : l, r : K \to R)$ is a DPO rule, the injection $L \subseteq G$ is a DPO redex satisfying the application condition, and the two squares are a DPO rewrite in the conventional sense. Evidently what we have constructed is far from being either a concrete or abstract rewrite as per appendix A, but the reader will accept that given a mild generalisation of the skeleton construction to include the graphs generated by the operations we used above, we can set what we did within a concrete rewriting framework, whence $(l, r)$ becomes a concrete rule and a concrete rewrite of $L \subseteq G$ exists, producing a graph isomorphic to $H$. This rewrite has an abstract representative, and applying this whole idea to pairs of abstract rewrites, gives us a composition for abstract rewrites, R-composition, the abstract version of R-related composition. To extend this to a composition of arrows in a category $\mathcal{C}$ as we need in section 3, requires the construction of some canonical isomorphisms, as in appendix A. This is not problematic.

Evidently composition of rewrites in $\mathcal{C}$ is a generalisation both of composition in $\mathcal{G}(\mathcal{P}, Rew)$, and of the SYNTHESIS process for sequentially independent rewrites in the DPO approach.

## References


Banach R. (1993); A Fibration Semantics for Extended Term Graph Rewriting. *in*: Term Graph Rewriting: Theory and Practice. Sleep, Plasmeijer, van Eekelen (eds.), 91-100, John Wiley, (1993).

Banach R. (1994); Term Graph Rewriting and Garbage Collection Using Opfibrations. Theor. Comp. Sci. **131** 29-94.

Barr M., Wells C. (1990); Category Theory for Computing Science. (2 ed.), Prentice-Hall.

Corradini A., Ehrig H., Lowe M., Montanari U., Rossi F. (1993a); Note on Standard Representations of Graphs and Graph Derivations. *in*: Graph Transformations in Computer Science. Schneider, Ehrig (eds.), L.N.C.S. **776** 104-118.

Corradini A., Ehrig H., Lowe M., Montanari U., Rossi F. (1993b); Abstract Graph Derivations in the Double pushout Approach. *in*: Graph Transformations in Computer Science. Schneider, Ehrig (eds.), L.N.C.S. **776** 86-103.

Corradini A., Ehrig H., Lowe M., Montanari U., Rossi F. (1994a); An Event Structure Semantics for Safe Graph Grammars. *in*: Proc. I.F.I.P. Working Conference PROCOMET-94, to appear.

Corradini A., Ehrig H., Lowe M., Montanari U., Rossi F. (1994b); An Event Structure Semantics for Consumptive graph Grammars with Parallel Productions. *in*: Proc. Fifth International Workshop on Graph Grammars and their Application to Computer Science 1994, Williamsburg U.S.A., L.N.C.S. to appear.

Corradini A., Ehrig H., Lowe M., Montanari U., Padberg J. (1994c); Functorial Semantics for Safe Graph Grammars using Prime Algebraic Domains and Event Structures. *in*: Proc. Fifth International Workshop on Graph Grammars and their Application to Computer Science 1994, Williamsburg U.S.A., L.N.C.S. to appear.

Corradini A., Ehrig H., Lowe M., Montanari U., Padberg J. (1994d); Typed Graph Grammars and their Adjunction with Categories of Derivations. *in*: Proc. Fifth International Workshop on Graph Grammars and their Application to Computer Science 1994, Williamsburg U.S.A., L.N.C.S. to appear.

Ehrig H. (1979); Introduction to the Algebraic Theory of Graph Grammars (A survey). *in*: L.N.C.S. **73**, 1-69, Springer, Berlin.

Ehrig H. (1986); A Tutorial Introduction to the Algebraic Approach of Graph Grammars. *in*: Third International Workshop on Graph Grammars, L.N.C.S. **291**, 3-14, Springer, Berlin.

Lowe M. (1991); Extended Algebraic Graph Transformation. Doctoral thesis, Fachbereich Informatik, Technical University Berlin.

Lowe M., Ehrig H. (1991); Algebraic Approach to Graph Transformation Based on Single Pushout Derivations. *in*: Proc. Graph-Theoretic Concepts in Computer Science, Mohring R.H. ed., L.N.C.S. **484** 338-353.

Nielsen M., Plotkin G., Winskel G. (1981); Petri Nets, Event Structures and Domains, Part I. Theoretical Computer Science **13** 85-108.

Schied G. (1993); On Relating Rewriting Systems and Graph Grammars to Event Structures. *in*: Graph Transformations in Computer Science, Schneider, Ehrig (eds.), L.N.C.S. **776** 326-340.

Winskel G. (1986); Event Structures. *in*: Petri Nets, An Advanced Course, L.N.C.S. **255** 325-392.

Winskel G. (1988); An Introduction to Event Structures. *in*: Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency. de Bakker, de Roever, Rozenberg (eds.), L.N.C.S. **354** 364-397.