Abstract Diagrams and an Opfibration Account of Typed Graph Transformation

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

A. Corradini
Dipartimento di Informatica, Università di Pisa, Corso Italia 40, Pisa, Italy.
andrea@di.unipi.it

Abstract: The "in the large" properties of typed graph transformation systems in the double pushout framework and a double pullback variation of it, are reexamined. Preceding accounts utilising a fixed choice of pullbacks (whether adopted directly or via partial morphisms) are seen to be excessively sensitive to the precise graphs involved for comfort. A theory of abstract diagrams is developed, that allows the smooth formulation of an abstract version of the theory. Graph transformation steps appear as a split opfibration over abstract type change. The category of graph grammars, the category of graph transition systems, and the category of graph derivation systems emerge as opfibrations over abstract type change. Weakening the level of abstraction to the extent used to preserve event identity in event based treatments of graph transformation phenomena, makes the transformation steps opfibration unsplit, and weakens certain adjunctions. All the properties of interest are combined in a single triple category.

Key Words: Graph grammars, typed graph transformations, DPO and DPB graph transformations, opfibrations.

1 Introduction

In Corradini et al. (1996b), a categorical account of typed double pushout (DPO) rewriting was given by constructing mappings from arrows *a* of the category of type graphs and concrete spans, to functors between categories of (in turn) grammars, transition systems, and derivation systems typed over the domain and codomain of *a*. Essentially the same idea will work for the single pushout approach (Löwe (1991), Löwe (1993)) by reinterpreting the span used in a double pushout rule (or direct derivation step) as a partial morphism (see Ribeiro (1996) who also uses partial morphism spans for retyping). In Ehrig et al. (1997), Heckel et al. (1997), a similar treatment was presented for a double pullback rewriting construction, essentially intended as a "looser" version of the double pushout construction rather than an independent construction (and in particular not to be confused with work of Bauderon on rewriting via pullbacks, see eg. Bauderon (1995)). The close relationship of Heckel et al.'s work to the traditional DPO construction relies crucially on the injectivity of the arrows in a DPO rule (or direct derivation step) in the two formulations being considered. In the former work the composition of concrete spans in the category of type graphs arises from a fixed choice of pullbacks; and the steps from categories of grammars to categories of transition systems and derivation systems are made via free constructions. In the latter the same effect for the category of type graphs is achieved by considering only partial morphisms of type graphs, a ruse which effectively forces a specific choice of pullbacks since one of the arrows of a span that is a concrete partial morphism must be a concrete inclusion; and the steps from categories of grammars to categories of transition systems and derivation systems are made via a Kleisli construction and a co-free construction. In fact partial morphism retyping and the use of the Kleisli construction in this manner arose in Heckel et al. (1996).

One notices two related things in these approaches. The first is that because a fixed choice of pullbacks is needed to enable the composition of two concrete spans to be a third concrete span, the action of the functors constructed subsequently is extremely fussy about the concrete graphs involved. Eg. when one graph grammar is the functorial image of another, changing the start graph in the target to an isomorphic graph will not do; the start graph has to be precisely the one given by the functor, and the grammar with the other start graph is not related to the source. The second thing is that the axioms imposed on the choice of pullbacks in order that the various constructions work smoothly, are just like those required for the splitting of a split opfibration.

This suggests that the functors corresponding to the arrows *a* mentioned above, indeed glue together to form an indexed category corresponding to a split opfibration. And furthermore that there is an underlying opfibration behind the constructions which is not split, so that the fussiness regarding specific graphs may well be attributable to having forced a splitting where there was no naturally arising one. The same train of thought prompts the search for a more abstract formulation of these phenomena, avoiding the irksome details mentioned. In this paper we re-engineer the central material of the key papers refered to in the first paragraph above, resisting the temptation to force a fixed choice of pullbacks (by whatever means). By developing a suitable theory of abstract diagrams, we do even better, finding that at a suitable level of abstraction, the splitness of relevant opfibrations emerges without effort. All of this is in contrast to the way split opfibrations are used for graph rewriting in Banach (1993, 1994, 1995).

In more detail, the rest of this paper is as follows. In Section 2 we recall the basic concepts we need on graphs, double pushout and double pullback rewriting, and the problems created by a naive approach to abstraction for graphs. We recall the essentials of standard isomorphisms as a means of building more appropriate equivalences on graphs. Section 3 reviews some categorical tools, including opfibrations, wreath products, feeble functors and weak adjunctions. Section 4, the technical core of the paper, develops a theory of abstract diagrams which are characterised as functor categories of concrete diagrams, and explores their properties. This includes incorporating the consequences of standard isomorphisms and leads to the notion of kinded abstract diagrams, in which the permitted isomorphisms between the concrete diagrams in an abstract diagram vary from vertex to vertex in the shape graph. Futher development of this theory culminates in the notion of interface-diagram category, a kind of category in which both objects and arrows are abstract diagrams of appropriately compatible shapes. These ideas turn out to be very close to those of internal category theory, except that pushout based composition techniques are used rather than pullback based ones. The former are more appropriate for our graph transformation applications. Interface-diagram categories are the central concepts using which the remainder of the notions in the paper are formalised.

In Section 5 these ideas are applied to spans and span morphisms. Their properties are gathered in a double interface-diagram category [*D-Gr-Sp*]. In Section 6 the notion of typing and type change is added to [*D-Gr-Sp*], culminating in the triple interface-diagram category [*D-Gr-Sp↓Gr-Sp*], which encapsulates all that is subsequently needed. The properties of [*D-Gr-Sp↓Gr-Sp*] constitute the technical apex of the paper. In particular [*D-Gr-Sp↓Gr-Sp*] is a split opfibration over type change. How these structures relate to graph transformation steps is described in Section 7.

Section 8 formalises the notion of graph grammar using these techniques. Graph grammars form an opfibration over type change. In Section 9 this state of affairs is generalised to graph transition systems, which are related to graph grammars via a fogetful functor and its left adjoint. A similar relationship pertains to graph derivation systems vis. a vis. transition systems, this forming the topic of Section 10. In Section 11 we consider various weakenings of the theory hitherto presented. We see that lowering the level of abstractness by having fewer concrete diagrams comprise an abstract one, makes the split opfibrations unsplit, and makes the left adjoints into weak left adjoints. Forgetting further the internal structure of abstract diagrams, yields a treatment in terms of equivalence classes, recovering a more conventional perspective on the situation. Section 12 concludes.

2 Graphs, Graph Transformations, and the Abstraction Problem

In this section we motivate what follows by presenting the essential elements of concrete graphs, concrete graph transformations, and the problems raised by trying to lift the level of abstraction. We also present the notion of abstract graphs.

2.1 Concrete Graphs and Concrete Graph Transformations

Definition 2.1.1 A concrete 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 Ω_E, Ω_V.

In fact the vertex and edge labels form a kind of typing system, classifying vertices and edges in a rather crude manner, and below we will be concerned with a more sophisticated kind of type system, where types are themselves graphs. All that we say subsequently will carry through unaltered irrespective of whether labels are present or not, and readers may prefer to forget about the labels altogether.

Definition 2.1.2 A concrete graph morphism $f : G \to G'$ is a pair of functions $f_E : E \to E'$, $f_V : V \to V'$ such that Fig. 1 below commutes in the obvious way.

Where necessary, 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 as above. This gives us the category *Gr* of concrete graphs and morphisms with obvious identities and composition of morphisms.

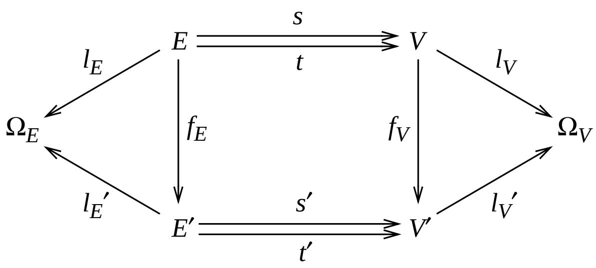

Fig. 1

In the classical double pushout (DPO) approach to graph rewriting (Ehrig (1979)) a production is defined as a concrete monic span $(L \leftarrow K \rightarrow R)$ in the category of graphs, where graphs L, R and K are called the left hand side, the right hand side and the interface, respectively. Given a graph G and an occurrence of L in G, i.e. a morphism $g : L \rightarrow G$, there is a direct derivation from G to a derived graph H if the diagram of Fig. 2 can be constructed in such a way that both squares are pushouts in Gr. This means that there is a graph D, the pushout complement of l and g, and morphisms d and $l*$ such that the left square is a pushout. (The nontrivial conditions for the existence of such a pushout complement (in Gr, which has all pushouts) are presented in Ehrig (1979).) Intuitively, the context graph D is obtained by removing from G all items that are in the image of g but not in the image of $g \circ l$. Moreover H is obtained as the pushout of r and d, which glues together the context graph and the right hand side over the common interface K.

Recently Heckel et al. (1997) have introduced a variant of the double pushout approach by considering "double pullback (DPB) transitions". Given a production as above, there is a DPB transition from G to H if a diagram like Fig. 2 can be constructed, where both squares are pullbacks. This provides a true generalization of DPO derivations, because the injectivity of productions guarantees that a DPO diagram is also a DPB. Very informally, a DPB transition using a production p can be understood as a trasformation from graph G to H where at least the effects prescribed by p have been performed, but possibly more. For a precise analsysis of the meaning of DPB transitions we refer to Heckel et al. (1997). For our purposes, we just want to stress that the formal framework we are introducing (originally conceived for the DPO approach only) can also accommodate the theory of DPB rewriting without additional effort. For the sake of uniformity, we will allow ourselves to call DPB transitions "direct derivations" as well.

Definition 2.1.3 A concrete production is a pair of monic arrows $l : K \rightarrow L$, $r : K \rightarrow R$ in Gr which we often write as $(L \leftarrow K \rightarrow R)$ when the rest can be understood.

In applications it is sometimes useful to let the right hand arrow $r : K \rightarrow R$ be not monic. However none of the theory that follows is invalidated in such a case, so we will not mention this more liberal possibility further in this paper.

Definition 2.1.4 Given a production $(L \leftarrow K \rightarrow R)$, a graph G, and an occurrence of the left hand side in G (which is just a morphism $g : L \rightarrow G$ of Gr), a direct derivation of H from G in the double pushout (DPO), respectively double pullback (DPB), approach to graph rewriting is a diagram like Fig. 2, in which both squares are pushouts, respectively pullbacks, in Gr. N.B. The application conditions of which we spoke are simply those necessary to ensure that given $l : K \to L$ and $g : L \to G$, the two pushouts or pullbacks indeed exist in Gr. See loc. cit.

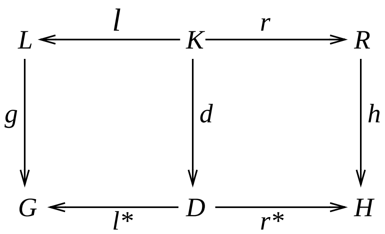

Fig. 2

Remark 2.1.5 In a commuting square which is a pushout or pullback in Gr like $LKDG$ in Fig. 2, then l is monic iff l^* is monic. In these circumstances, the only difference between the two possibilities is that for a pushout, the morphisms g and l^* are epic (i.e. onto). Thus the DPO case becomes a special case of the DPB case, and for this reason we will consider them together below. Starting with the DPB case will usually be simpler, and then we will check that the additional assumption of surjectivity behaves well in the construction in question.

A graph grammar in this classical theory is usually defined as a collection of productions plus a start graph, and a graph derivation for a grammar, is a sequence of adjacent direct derivation steps using productions of the grammar, starting from the start graph, and remembering the productions used and all the other Fig. 2 data, for each derivation step in the sequence.

One of the things we wish to do in this paper, aside from introducing the typing of graphs and their transformations, is to raise the level of abstraction from individual concrete graphs and concrete graph morphisms as in Fig. 2. A strategy which suggests itself naturally is to form equivalence classes of graphs and of morphisms and to proceed from there. Unfortunately this is easier said than done. An example due to Corradini et al. (1994a,b), taking place in Set (which we can regard as a category of unlabelled discrete graphs), illustrates the problem.

Example 2.1.6 Let $S_1 = \{1, 2\}$ and $S_2 = \{1, 2, 3\}$. Consider the maps $f : S_1 \to S_2$ and $g, g' : S_2 \to S_1$ illustrated in Fig. 3. Now in a naive construction of abstract sets and abstract maps between them, the abstract set containg a set S would be all sets equipolent to S, and the abstract map containing a map $s : S_1 \to S_2$ would be the collection of all maps $t : T_1 \to T_2$ such that there are isomorphisms $j_1 : S_1 \to T_1$ and $j_2 : S_2 \to T_2$ such that $s = j_2^{-1} \circ t \circ j_1$. In Fig. 3 we claim that g and g' would be in the same isomorphism class because if we take j_1 as the map $\{1 \mapsto 3, 2 \mapsto 1, 3 \mapsto 2\}$ and take j_2 as the map $\{1 \mapsto 2, 2 \mapsto 1\}$ then $g = j_2^{-1} \circ g' \circ j_1$. Now the composition of two abstract maps would be the abstract map containing at least all composites of respective concrete maps which are directly composable. So in the example, $g \circ f$ and $g' \circ f$ would be in the same abstract map. However, $g \circ f$ is monic while $g' \circ f$ is not, so this is impossible because monicity is invariant under isomorphism.

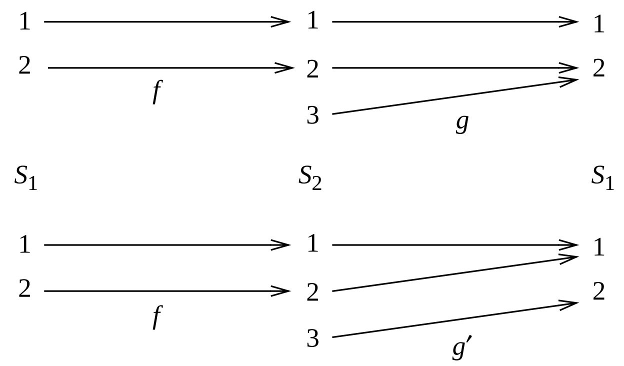

Fig. 3

The reason why we get this unpleasant phenomenon is clear. When we form the composite, we have "forgotten" that we have to relate g and g' by j_1 and j_2 in this particular instance, because the formation of equivalence classes does not remember this information. The technique of standard graphs and isomorphisms addresses this problem.

2.2 Standard Isomorphisms and Abstract Graphs and Morphisms

Definition 2.2.1 A choice of standard isomorphisms in *Gr* assigns to each pair of isomorphic graphs G_1 and G_2, a standard isomorphism $\sigma(G_1, G_2)$ such that:

(1) $\sigma(G, G) = \mathrm{id}_G$

(2) $\sigma(G_2, G_3) \circ \sigma(G_1, G_2) = \sigma(G_1, G_3)$

(3) $\sigma(G_2, G_1) = \sigma(G_1, G_2)^{-1}$

If we disallow all isomorphisms other than standard ones, the problems of Example 2.1.6 disappear because j_1 and j_2 are not standard by (1) above; hence g and g' fall into different classes.

Definition 2.2.2 We can construct a choice of standard isomorphisms in *Gr* by:

(1) Choosing one graph $\sigma(G)$ from each isomorphism class $[G]$ of graphs isomorphic to G to be standard,

(2) For each G' in $[G]$, choosing one isomorphism $\sigma(\sigma(G), G')$ to be standard (with $\sigma(\sigma(G), G')$ chosen to be $\mathrm{id}_{\sigma(G)}$ if $G' = \sigma(G)$),

(3) For all G_1, G_2 in $[G]$, setting $\sigma(G_1, G_2) = \sigma(\sigma(G), G_2) \circ \sigma(\sigma(G), G_1)^{-1}$.

For the sequel we assume fixed some choice of standard isomorphisms in Gr. The collection of standard graphs and all morphisms between them forms a skeleton category Gr^{K} of Gr. As shown in Corradini et al. (1994a), Gr^{K} is isomorphic to the category $\langle \mathit{Gr} \rangle$, whose objects are isomorphism classes of concrete graphs up to standard isomorphism called abstract graphs and written $\langle G \rangle$, and whose arrows are equivalence classes of concrete morphisms under the relation that relates $g : G \to H$ and $g' : G' \to H'$ iff $g = \sigma(G', H')^{-1} \circ g' \circ \sigma(G, H)$, called abstract morphisms and written $\langle g : G \to H \rangle$. The use of standard isomorphisms only in this relation means that there is a bijection between concrete arrows $g : G \to H$ in $\langle g : G \to H \rangle$, and ordered pairs G, H taken from $\langle G \rangle$ and $\langle H \rangle$. Identities are the equivalence classes of concrete identities, and composition of arrows $\langle g : G \to H \rangle$ and $\langle h : H \to K \rangle$ is given by composing the concrete arrows in the two respective classes in the only possible way using the standard isomorphisms, which forms another equivalence class[1].

3 Some Categorical Tools

In this section we review some categorical techniques which will be needed later.

3.1 Opfibrations

Let $P : E \to B$ be a functor from the subject category E to the base B. Suppose $P(e_0 : E_0 \to E_1) = b_0 : B_0 \to B_1$. The arrow $e_0 : E_0 \to E_1$ is opcartesian for B_0 and b_0, iff for every arrow $e_{01} : E_0 \to E_2$ and any $b_1 : B_1 \to B_2$ such that $P(e_{01} : E_0 \to E_2) = b_1 \circ b_0 : B_0 \to B_2$, we have a unique $\theta : E_1 \to E_2$ such that $e_{01} = \theta \circ e_0$ and $P(\theta) = b_1$. See Fig. 4. An opfibration is a functor $P : E \to B$, such that for every pair $(E, b : P(E) \to B)$, there is an opcartesian arrow for E and b. A particular choice of opcartesian arrow $\kappa(E, b)$ for each pair (E, b) is called an opcleavage κ of the opfibration.

In general, writing $b = b : B_0 \to B_1$, any arbitrary choice of opcleavage induces a functor $F_{\kappa(-,\, b)} : P^{-1}(B_0) \to P^{-1}(B_1)$ where $P^{-1}(B_i)$ $(i = 0, 1)$ is the subcategory of E over B_i. This works by $F_{\kappa(-,\, b)}(E_0) = \mathrm{cod}(\kappa(E, b))$ and $F_{\kappa(-,\, b)}(e : E_0 \to E_1) = \theta : \mathrm{cod}(\kappa(E_0, b)) \to \mathrm{cod}(\kappa(E_1, b) \circ e)$ where θ is the unique arrow promised by the universal property. In general, there are natural isomorphisms between $\mathrm{Id}(P^{-1}(B))$ and $F_{\kappa(-,\, \mathrm{id} : B \to B)}$, and also between $F_{\kappa(-,\, b_1 \circ b_0)}$ and $F_{\kappa(-,\, b_1)} \circ F_{\kappa(-,\, b_0)}$. If $\kappa(E, \mathrm{id} : P(E) \to P(E)) = \mathrm{id}_E$ for all

1. Note that foundationally speaking, the definition of $\langle \mathit{Gr} \rangle$ is suspect. Since the collection of concrete graphs isomorphic to any given one forms a proper class, so does the collection of concrete arrows in any isomorphism class of arrows between any two abstract graphs. Now we are in trouble since the homset between any two objects in a category (eg. two abstract graphs in $\langle \mathit{Gr} \rangle$) must be a set, and a set cannot have proper classes as members. We will not worry unduly about this, since we never use *membership* of these large collections in any way that could cause us problems — and we wish to avoid obfuscating the technical account with details that do not add materially to the essence of the algebraic story we tell; this is similar to the way that applied mathematics typically ignores the precision of rigorous analysis. Perhaps the only formulations of what we do that are truly free from foundational defects, are (1) a formulation in terms of Grothendieck's Universes, where proper classes can in effect be chosen small enough, (2) a formulation exclusively in terms of the previously selected skeleton.

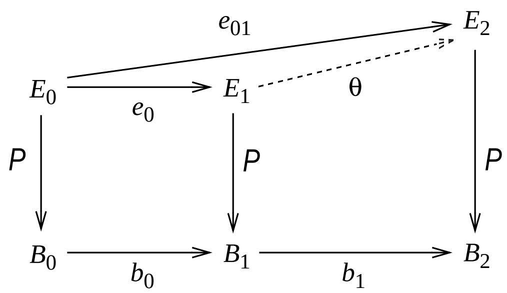

Fig. 4

E, and $\kappa(E, b_1 \circ b_0) = \kappa(\mathrm{cod}(\kappa(E, b_0)), b_1) \circ \kappa(E, b_0)$ for all relevant E, b_0, b_1, we say that the opfibration is split and the opcleavage is a splitting.

For split opfibrations (with a specified splitting) we have the theorem of Grothendieck which states that they correspond exactly to functors $\mathsf{F} : \mathsf{B} \to \mathsf{Cat}$. The functors $\mathsf{P} : \mathsf{E} \to \mathsf{B}$ and $\mathsf{F} : \mathsf{B} \to \mathsf{Cat}$ determine each other. From F we reconstruct E up to isomorphism by the Grothendieck construction, which builds $\mathsf{G}(\mathsf{B}, \mathsf{F})$, the Grothendieck category of B and F. The objects of $\mathsf{G}(\mathsf{B}, \mathsf{F})$ are (E_0, B_0) for E_0 in $\mathsf{F}(B_0)$ for B_0 an object of B, and the arrows of $\mathsf{G}(\mathsf{B}, \mathsf{F})$ are $(e_1 : \mathsf{F}(b_0)(E_0) \to E_1, b_0) : (E_0, B_0) \to (E_1, B_1)$ for $b_0 : B_0 \to B_1$ an arrow of B; with composition of $(e_1 : \mathsf{F}(b_0)(E_0) \to E_1, b_0) : (E_0, B_0) \to (E_1, B_1)$ and $(e_2 : \mathsf{F}(b_1)(E_1) \to E_2, b_1) : (E_1, B_1) \to (E_2, B_2)$ being given by $(e_2 \circ \mathsf{F}(b_1)(e_1) : \mathsf{F}(b_1 \circ b_0)(E_0) \to E_2, b_1 \circ b_0) : (E_0, B_0) \to (E_2, B_2)$; and with obvious identities.

Fig. 5 shows how the components of the Grothendieck category relate to one another, the picture on the right showing the constituent parts of an arrow of $\mathsf{G}(\mathsf{B}, \mathsf{F})$, with other related data shown dashed. Note in particular that such an arrow consists of two parts, a change of base arrow b_0 and an in-fibre morphism e_1. For an accessible introduction to opfibrations, and further key references, see Barr and Wells (1990).

We have spelled these things out here in fair detail because towards the end of the paper we will have some need for opfibrations $\mathsf{P} : \mathsf{E} \to \mathsf{B}$ which are not split, so there will not be a handy functor $\mathsf{F} : \mathsf{B} \to \mathsf{Cat}$ to conveniently visualise the inverse relationship. Nevertheless we will be dealing with subject categories E which arise most naturally by making what is in effect a nondeterministic brute force analogue of the Grothendieck construction. More specifically we identify a projection, call it $\mathsf{P}_0 : \mathsf{E}_0 \to \mathsf{B}$ say, and prove that all arrows in E_0 are opcartesian. We then typically enrich $\mathsf{P}_0 : \mathsf{E}_0 \to \mathsf{B}$ to a projection $\mathsf{P} : \mathsf{E} \to \mathsf{B}$ by "adjoining in-fibre morphisms", showing that the properties of an opfibration continue to hold — the fibres are evidently $\mathsf{P}^{-1}(b)$ for b an object of B. This amounts to showing that $\mathsf{P} : \mathsf{E} \to \mathsf{B}$ is an opfibration directly from the definition because it has "enough strong opcartesian morphisms" in the terminology of Gray (1966), Grothendieck (1961). This in turn is equivalent to constructing an opfibration via a pseudofunctor $\mathsf{F}_P : \mathsf{B} \to \mathsf{Cat}$ which chooses a cleavage that is not necessarily a splitting, by making an arbitrary choice of opcartesian arrow for each (E, b) pair. Our more direct approach avoids the distraction of making such a choice, only for it to be disregarded later, and allows proofs to be ported from the split to the nonsplit case.

3.2 Wreath Products

A wreath product is a special case of the dual construction, i.e. of a fibration. Let $\mathcal{C}$ be a category. Let $\mathcal{P}th : \mathcal{B} \to \mathcal{C}at$ be a functor, taking arrows $b_0 : B_0 \to B_1$ to functors $\mathcal{P}th(b_0) : \mathcal{P}th(B_0) \to \mathcal{P}th(B_1)$. Now let $\mathcal{F} : \mathcal{B}^{op} \to \mathcal{C}at$ be a contravariant functor, taking objects B_0 to the functor categories $[\mathcal{P}th(B_0), \mathcal{C}]$, and taking arrows $b_0 : B_0 \to B_1$ to contravariant functors $\mathcal{F}(b_0) : [\mathcal{P}th(B_1), \mathcal{C}] \to [\mathcal{P}th(B_0), \mathcal{C}]$. The functors $\mathcal{F}(b_0)$ take a functor $\mathcal{H} : \mathcal{P}th(B_1) \to \mathcal{C}$ in $[\mathcal{P}th(B_1), \mathcal{C}]$ to the functor $\mathcal{H} \circ \mathcal{F}(b_0) : \mathcal{P}th(B_0) \to \mathcal{C}$ in $[\mathcal{P}th(B_0), \mathcal{C}]$ in the expected manner.

Because $\mathcal{F}$ is a functor into $\mathcal{C}at$ we can use the (contravariant version of the) Grothendieck construction to build the Grothendieck category of $\mathcal{B}$ and $\mathcal{F}$, called the wreath product of $\mathcal{B}$ and $\mathcal{C}$ and written $\mathcal{B}\ \mathrm{wr}^{\mathcal{P}th}\ \mathcal{C}$. The objects of $\mathcal{B}\ \mathrm{wr}^{\mathcal{P}th}\ \mathcal{C}$ are pairs $(B_0, \mathcal{H}_0)$ with B_0 an object of $\mathcal{B}$ and $\mathcal{H}_0$ a functor in $[\mathcal{P}th(B_0), \mathcal{C}]$. The arrows of $\mathcal{B}\ \mathrm{wr}^{\mathcal{P}th}\ \mathcal{C}$ are pairs $(b_0, n_1) : (B_0, \mathcal{H}_0) \to (B_1, \mathcal{H}_1)$, where $b_0 : B_0 \to B_1$ is an arrow of $\mathcal{B}$ and $n_1 : \mathcal{H}_0 \to \mathcal{H}_1 \circ \mathcal{F}(b_0)$ is a natural transformation. Composition of arrows $(b_0, n_1) : (B_0, \mathcal{H}_0) \to (B_1, \mathcal{H}_1)$ and $(b_1, n_2) : (B_1, \mathcal{H}_1) \to (B_2, \mathcal{H}_2)$ is given by $(b_1, n_2) \circ (b_0, n_1) = (b_1 \circ b_0, \mathcal{F}(b_0)(n_2) \circ n_1) : (B_0, \mathcal{H}_0) \to (B_2, \mathcal{H}_2)$. There is an evident projection $\mathcal{P} : \mathcal{B}\ \mathrm{wr}^{\mathcal{P}th}\ \mathcal{C} \to \mathcal{B}$ given by forgetting the second component of objects and arrows. For more details see again Barr and Wells (1990).

3.3 Feeble Functors and Weak Adjunctions

The slightly less well known material of this subsection is adapted from Krishnan (1981), Kainen (1971). A feeble functor $\mathcal{F} : \mathcal{A} \to \mathcal{B}$ maps objects of $\mathcal{A}$ to objects of $\mathcal{B}$ as usual, but maps arrows $f : a \to b$ of $\mathcal{A}$ to nonempty sets of arrows of $\mathcal{B}$ via $\mathcal{F}(f : a \to b) \subseteq \hom_{\mathcal{B}}(\mathcal{F}(a), \mathcal{F}(b))$, such that if f and g are composable, then $\mathcal{F}(g) \circ \mathcal{F}(f) \subseteq \mathcal{F}(g \circ f)$ where the composition symbol has been overloaded in the obvious way. (We will also

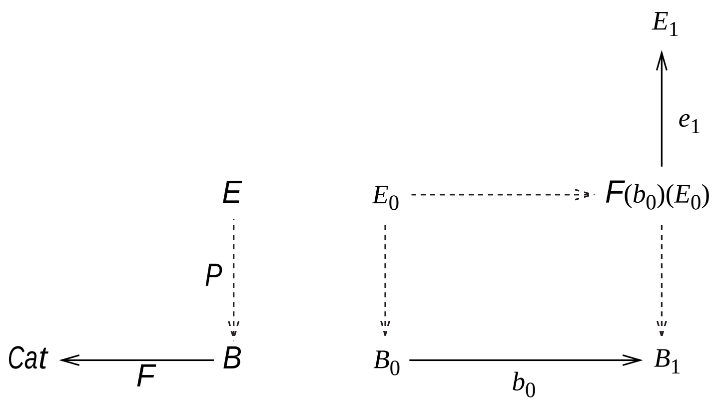

Fig. 5

need the possibility that these “hom sets” are classes, and that F maps arrows into suitable subclasses.) Given two feeble functors $F, G : A \to B$, a left-natural transformation $\eta : F \to G$ maps each object a in A to a non-empty set of arrows $\eta(a) \subseteq \hom_B(F(a), G(a))$ such that for each arrow $f : a \to b$ of A, $\eta(b) \circ F(f) \supseteq G(f) \circ \eta(a)$. If the direction of the inclusion is reversed we have a right-natural transformation. A transformation that is simultaneously left-natural and right-natural between two feeble functors is called a natural transformation.

Feeble functors are used in weak adjunctions which we now describe. Let $F : A \to B$ be a feeble functor and $G : B \to A$ be a (normal) functor. Then F is a weak left adjoint of G iff there exists a natural transformation $n : \hom_B(F \times \mathrm{Id}(B)) \to \hom_A(\mathrm{Id}(A) \times G)$ and a left-natural transformation $m : \hom_A(\mathrm{Id}(A) \times G) \to \hom_B(F \times \mathrm{Id}(B))$, such that $n \circ m = 1_{\hom_A(\mathrm{Id}(A) \times G)}$ and $m \circ n \supseteq 1_{\hom_B(F \times \mathrm{Id}(B))}$. Here both $\hom_{\_}(–, –)$ notations are being viewed as functors $A^{\mathrm{op}} \times B \to Set_{CF}$, where Set_{CF} is the category of sets and cofull relations between them. Equivalent conditions are given by the following.

Let $G : B \to A$ be a functor. Then among the conditions below we have (1) ⇒ (2) ⇒ (3) ⇒ (1) and (1*) ⇒ (2*) ⇒ (3*) ⇒ (1*).

(1) There is a feeble functor $F : A \to B$ which is a weak left adjoint to G.

(1*) In addition to (i), if (n, m) define the weak adjunction then $(\, a$ is an object of $A \wedge h \in \hom_B(F(a), F(a)) \wedge n(a, F(a)) \wedge G(h) \circ k = k \,) \Rightarrow (\, h = \mathrm{id}_{F(a)} \,)$.

(2) There is a feeble functor $F : A \to B$ and a natural transformation $\eta : \mathrm{Id}(A) \to G \circ F$, and for every object b of B a non-empty set $\nu(b) \subseteq \hom_B(F \circ G(b), b)$ such that: (a), $G(\nu(b)) \circ \eta(G(b)) = \mathrm{id}_{G(b)}$, and (b), for every object a of A $(\, f \in \hom_A(a, G(a)) \wedge h \in \nu(b) \circ F(f) \,) \Rightarrow (\, G(h) \circ \eta(a) = f \,)$.

(2*) In addition to (ii), (c), for every object a of A $(\, h \in \hom_B(F(a), F(a)) \wedge G(h) \circ \eta(a) = \eta(a) \,) \Rightarrow (\, h = \mathrm{id}_{F(a)} \,)$.

(3) Every object a of A has a universal arrow $(u : a \to G(b_a), b_a)$ in the sense that for every object b of B and every $f : a \to G(b)$ there is a (not necessarily unique) arrow $g : b_a \to b$ such that $G(g) \circ u = f$.

(3*) In addition to (iii), $(\, h : b \to b \wedge G(h) \circ u = u \,) \Rightarrow (\, h = \mathrm{id}_b \,)$.

Below, when we need to establish a weak left adjunction, we will use condition (3), which is just the normal thing one would do aside from checking uniqueness. Also it is easy to see that if we strengthen F to be a (normal) functor and the adjunction to be not weak in (1), or insist on uniqueness in (3), or the appropriate strengthening of (2), then we recover some of the conventional characterisations of adjunctions, and furthermore the provisions of the starred clauses hold automatically. Evidently there is a dual theory for weak right adjoints which we do not give in detail.

4 Concrete and Abstract Diagrams

We now build a theory of diagrams, both concrete and abstract, in a form designed for later convenience. Let B be the category of directed graphs, obtained by forgetting the labelling functions in Gr. We allow classes instead of (finite) sets of vertices and edges in a directed graph if necessary.

4.1 Concrete and Abstract Diagrams in an Arbitrary Category

Definition 4.1.1 Let $\underline{\mu}$ be a directed graph, C be a category, and $\underline{\gamma} : \underline{\mu} \to UC$ be a graph morphism from $\underline{\mu}$ to the underlying graph of C. Then $\underline{\gamma}$ is a concrete diagram of shape $\underline{\mu}$ in C. Let $Pth : B \to Cat$ be the functor that sends directed graphs to their path categories, which is left adjoint to U. Then the standard free construction extends $\underline{\gamma} : \underline{\mu} \to UC$ to a functor $\gamma : \mu \to C$ from the path category μ of $\underline{\mu}$ to C. If in addition, for all pairs of objects m_0, m_1 in μ, for all paths $(e_1, \ldots, e_k)$ from m_0 to m_1 in μ, if the internal composition $(\gamma(e_k) \circ \ldots \circ \gamma(e_1))$ in C always yields the same arrow $f : \gamma(m_0) \to \gamma(m_1)$, then the diagram is a commuting concrete diagram of shape μ.

Henceforth we will only consider commuting diagrams, and will therefore drop the adjective "commuting".

Definition 4.1.2 $[\mu, C]$ is the functor category with objects given by concrete diagrams of shape μ, and arrows given by concrete diagram morphisms, which are natural transformations $n : \gamma \to \delta$ in $[\mu, C]$.

Note that this construction characterises the (commuting) concrete diagrams in C as a wreath product, namely as $B \,\mathrm{wr}^{Pth}\, C$, with $\underline{\Theta} : B^{\mathrm{op}} \to Cat$ being the relevant contravariant change of shape functor, sending $\underline{\alpha} : \underline{\nu} \to \underline{\mu}$ to $\underline{\Theta}(\underline{\alpha}) : [\mu, C] \to [\nu, C]$. For convenience below, rather than using $\underline{\Theta}$, we will refer to Θ, where $\Theta : Pth(B)^{\mathrm{op}} \to Cat$ takes $\alpha = Pth(\underline{\alpha}) : \nu \to \mu$ to $\Theta(\alpha) : [\mu, C] \to [\nu, C]$. We clearly have that $\underline{\Theta} = \Theta \circ Pth$.

Definition 4.1.3 An abstract diagram D (of shape μ in C) is a subcategory of $[\mu, C]$ such that for any two objects γ and δ in D, there is at least one arrow $n : \gamma \to \delta$ in D, and all the C arrows that make up such a D arrow (i.e. natural transformation) n, are isomorphisms.

Definition 4.1.4 An abstract diagram D of shape μ is maximal iff **(** γ is a concrete diagram of D and $n : \gamma \to \delta$ is a concrete diagram morphism such that all the C arrows that make up n are isomorphisms **)** $\Rightarrow$ **(** $n : \gamma \to \delta$ is a concrete diagram morphism in D **)**.

Definition 4.1.5 A morphism $c : D_0 \to D_1$ of abstract diagrams (of shape μ in C) is simply a functor from D_0 to D_1 (where both D_0 and D_1 are considered simply as categories in their own right). A morphism $c : D_0 \to D_1$ is mediated by a family Ξ of arrows of C iff there is a function $\chi : (\mathrm{Vert}(\underline{\mu}) \times \mathrm{Obj}(D_0)) \to \mathrm{Arr}(C)$, whose range is Ξ, that maps pairs (m_0, γ) to arrows of C such that:

(1) For any fixed concrete diagram γ of D_0, the collection of the $\chi(m_0, \gamma)$ forms a concrete diagram morphism from γ to $c(\gamma)$.

(2) For any fixed concrete diagram morphism $n : \gamma \to \delta$ of D_0, the collection of the $\chi(m_0, \gamma)$ and $\chi(m_0, \delta)$ forms a morphism of concrete diagram morphisms from $n : \gamma \to \delta$ to $c(n : \gamma \to \delta) : c(\gamma) \to c(\delta)$, naturally.

Thus while an arbitrary morphism of abstract diagrams merely associates concrete diagrams and morphisms between them in a natural manner, a morphism of abstract diagrams mediated by a family of arrows of C must be sensitive to any internal structure of objects captured by the structure of C.

For convenience we will also allow χ to have as domain $(\mathrm{Vert}(\underline{\mu}) \times Ind(\mathrm{Obj}(D_0)))$ where Ind is an index set (or class) for the objects of D_0 below.

Clearly the change of shape action of Θ extends naturally to a change of shape action on abstract diagrams.

4.2 Operations on Concrete and Abstract Diagrams

In this section we discuss how operations that arise naturally on concrete diagrams extend equally naturally to abstract ones. We start with subdiagrams.

Definition 4.2.1 Let $\underline{\alpha} : \underline{\nu} \to \underline{\mu}$ be a monic morphism of directed graphs. This defines a particular subobject of $\underline{\mu}$. This extends naturally via the action of Pth to a particular subobject $\alpha : \nu \to \mu$. Let γ be a concrete diagram of shape μ in C. Then the natural action of Θ yields a concrete subdiagram δ of shape ν of the concrete diagram γ in C.

Definition 4.2.2 Let $\alpha : \nu \to \mu$ be a subobject of the path category object μ. Let D_0 be an abstract diagram of shape μ in C. Then the natural action of Θ yields an abstract subdiagram D_1 of shape ν of the abstract diagram D_0 in C, where the morphism from D_1 to D_0 is mediated by a family of identities. Clearly if D_0 is maximal and nonempty, then D_1 is maximal too.

Where appropriate, we can regard the process of obtaining the (concrete or abstract) subdiagram as a kind of garbage collection. Now for the pasting of diagrams, a kind of pushout.

Definition 4.2.3 Let $\underline{\alpha} : \underline{\rho} \to \underline{\mu}$ and $\underline{\beta} : \underline{\rho} \to \underline{\nu}$ be morphisms of directed graphs. Then we can form the directed graph pushout $\underline{\alpha'} : \underline{\nu} \to \underline{\mu \oplus_\rho \nu}$, $\underline{\beta'} : \underline{\mu} \to \underline{\mu \oplus_\rho \nu}$. This extends naturally via the action of Pth to the pushout $\alpha' : \nu \to \mu \oplus_\rho \nu$, $\beta' : \mu \to \mu \oplus_\rho \nu$ of path category morphisms $\alpha : \rho \to \mu$ and $\beta : \rho \to \nu$. Let γ and δ be concrete diagrams of shape μ and ν respectively in C. Suppose for all vertices m_0 and arrows $e : m_0 \to m_1$ in ρ we have that $\gamma \circ \alpha(m_0) = \delta \circ \beta(m_0)$ and $\gamma \circ \alpha(e : m_0 \to m_1) = \delta \circ \beta(e : m_0 \to m_1)$; then we say that γ and δ are compatible. We define the concrete diagram $\gamma \oplus_\rho \delta : \mu \oplus_\rho \nu \to C$, of shape $\mu \oplus_\rho \nu$, provided it is a commuting diagram, by: $\gamma \oplus_\rho \delta(m_0) = \gamma(m_0)$ if m_0 is in μ, and $\gamma \oplus_\rho \delta(m_0) = \delta(m_0)$ if m_0 is in ν; and $\gamma \oplus_\rho \delta(e : m_0 \to m_1) = \gamma(e : m_0 \to m_1)$ if e is in μ, and $\gamma \oplus_\rho \delta(e : m_0 \to m_1) = \delta(e : m_0 \to m_1)$ if e is in ν, which is consistent. We call $\gamma \oplus_\rho \delta$ the pasting of γ and δ along ρ, and say that γ and δ are a compatible consistent pair.

Note the requirement that $\gamma \oplus_\rho \delta$ commutes; the pushout of shape graphs may create new pairs of paths with the same origin and destination vertices in the result, creating in turn fresh equations that must be satisfied by the arrows of $\gamma \oplus_\rho \delta$.

Definition 4.2.4 Let $\alpha' : \nu \to \mu \oplus_\rho \nu$, $\beta' : \mu \to \mu \oplus_\rho \nu$ be the pushout of path category morphisms $\alpha : \rho \to \mu$ and $\beta : \rho \to \nu$, and D_0 and D_1 be abstract diagrams of shape μ and ν respectively in C. We define the abstract diagram $D_0 \oplus_\rho D_1$ as the family of pastings along ρ, of all compatible consistent pairs of concrete diagrams γ and δ from D_0 and D_1 respectively, these being the objects of $D_0 \oplus_\rho D_1$. If $n_1 : \gamma \to \gamma'$ and $n_2 : \delta \to \delta'$ are morphisms in D_0 and D_1 respectively, γ and δ are compatible consistent, γ' and δ' are compatible consistent, and n_1 and n_2 agree as natural transformations on ρ, i.e. $n_1(\alpha(m)) = n_2(\beta(m))$ for each vertex m in ρ, then $n_1 \oplus_\rho n_2 : \gamma \oplus_\rho \delta \to \gamma' \oplus_\rho \delta'$ defined in the obvious way, is a morphism of $D_0 \oplus_\rho D_1$. We call $D_0 \oplus_\rho D_1$ the pasting of D_0 and D_1 along ρ. Clearly if D_0 and D_1 are maximal, then $D_0 \oplus_\rho D_1$ is maximal too.

In the above, the case of most interest to us will be when α and β are monic.

The final construction that we will deal with here is the local pullback (which is *not* a dual to pasting). For this recall that in a pullback construction we start with a diagram of shape $\bullet_1 \rightarrow \bullet_0 \leftarrow \bullet_2$, and end with a commuting square (with some universal properties), except that the object at the new vertex is only fixed up to isomorphism. From our perspective, we can say that we start with a concrete diagram, and finish with a (non maximal) abstract diagram, where the only non identity isomorphisms between concrete representatives in the abstract diagram are at the new vertex. The following local pullback construction thus makes sense principally for abstract diagrams.

Definition 4.2.5 Let D_0 be an abstract diagram of shape μ in C, where we assume that C has all pullbacks. Suppose there is a monic morphism ι from $\bullet_1 \rightarrow \bullet_0 \leftarrow \bullet_2$ to μ. Let $\blacklozenge$ be a fresh vertex not occurring in μ and let ν be the shape $\bullet_1 \leftarrow \blacklozenge \rightarrow \bullet_2$. Let $\mu^{\blacklozenge} = \mu \oplus_{\rho} \nu$ be the shape obtained by pasting ν to μ, via the common subshape $\rho = (\bullet_1 \;\; \bullet_2)$ and morphisms $\alpha : \rho \rightarrow \mu$ and $\beta : \rho \rightarrow \nu$; where $\alpha(\bullet_i) = \iota(\bullet_i)$ and $\beta(\bullet_i) = \mathrm{id}(\bullet_i)$ for $i = 1$, 2. Then the abstract diagram D_1 of shape $\mu^{\blacklozenge}$ in C is given as follows. Let γ be a concrete diagram in D_0; then $\gamma \circ \iota$ is a pair of coterminal arrows over $\bullet_1 \rightarrow \bullet_0 \leftarrow \bullet_2$. Let $\xi_2 : c \rightarrow \gamma \circ \iota(\bullet_1)$ and $\xi_1 : c \rightarrow \gamma \circ \iota(\bullet_2)$ be some specific pullback of $\gamma \circ \iota$. Then we define γ_c to be the concrete diagram of shape $\mu^{\blacklozenge}$ in C given by: $\gamma_c(m_0) = \gamma(m_0)$ and $\gamma_c(e : m_0 \rightarrow m_1) = \gamma(e : m_0 \rightarrow m_1)$ for vertices m_0 and edges $e : m_0 \rightarrow m_1$ in γ; and $\gamma_c(\blacklozenge) = c$, $\gamma_c(\bullet_1 \leftarrow \blacklozenge) = \xi_2 : c \rightarrow \gamma \circ \iota(\bullet_1)$, $\gamma_c(\blacklozenge \rightarrow \bullet_2) = \xi_1 : c \rightarrow \gamma \circ \iota(\bullet_2)$ for the remainder. Then D_1 contains as objects, all such concrete diagrams γ_c, for all possible pullbacks ξ_1, ξ_2. If γ_c and δ_d are two such objects, arising from objects γ and δ in D_0, with natural transformation $n : \gamma \rightarrow \delta$, then there is a natural transformation $n_{c,d} : \gamma_c \rightarrow \delta_d$ given by extending n with the unique (by pullback properties) isomorphism from c to d which makes $n_{c,d}$ natural.

Note that D_1 above is *unique* because of the maximality inherent in its definition. This is unlike the construction of normal pullbacks where the resulting diagram is up to isomorphisms of the added object. This feature will produce tangible consequences later.

Note also that the above construction generalises to limits of larger subdiagrams than coterminal arrow pairs, and there is an obvious dual construction for colimits. However the local pullback case is the only one we need below. Note how the pasting and local limit constructions act at different levels of abstraction.

4.3 Automorphisms and Kinded Abstract Diagrams

Since we will ultimately be interested in applying our theory of abstract diagrams to the case where C is Gr, we now examine the consequences of objects in C having nontrivial automorphisms. The same problems that we have noticed already regarding equivalence classes of objects and arrows, reappear here, so we adopt the same machinery.

We thus assume chosen a skeleton subcategory C^{K} of C, leading to a choice of standard isomorphisms $\sigma(-, -)$ between objects. Also $\langle \mathsf{C} \rangle$ will be the category of abstract C objects and arrows, consisting of equivalence classes up to standard isomorphisms, of C objects and arrows.

Let *Kind* = {id, std, iso}. We will use *Kind* as a label set for shape vertices, thus for an abstract diagram of shape μ there will be a map, *kind*, from its vertices to *Kind*, and we will speak of shapes and vertices of kind such and such.

Definition 4.3.1 Let D be an abstract diagram of kinded shape μ, then D conforms to its kind iff for each vertex m_0 in μ:

(1) $kind(m_0) = \mathsf{id} \Leftrightarrow$ for each arrow $n : \gamma \to \delta$ in D, the component of the natural transformation n at the vertex m_0 is an identity in C, i.e. $n(m_0) : \gamma(m_0) \to \delta(m_0) = \mathrm{id}_{\gamma(m_0)}$,

(2) $kind(m_0) = \mathsf{std} \Leftrightarrow$ for each arrow $n : \gamma \to \delta$ in D, the component of the natural transformation n at the vertex m_0 is a standard isomorphism in C, i.e. $n(m_0) : \gamma(m_0) \to \delta(m_0) = \sigma(\gamma(m_0), \delta(m_0))$,

(3) $kind(m_0) = \mathsf{iso} \Leftrightarrow$ for each arrow $n : \gamma \to \delta$ in D, the component of the natural transformation n at the vertex m_0 is an arbitrary isomorphism in C, i.e. $n(m_0) : \gamma(m_0) \to \delta(m_0)$ is an arbitrary iso.

If D conforms to its kind then we also say that all its natural transformations $n : \gamma \to \delta$ conform to the kinds of D. From now on we assume that all abstract diagrams conform to their kinds, and we will simply speak of kinded abstract diagrams.

Definition 4.3.2 Let D be a kinded abstract diagram. Then D is maximal iff **(** γ is a concrete diagram of D and $n : \gamma \to \delta$ is a concrete diagram morphism conforming to the kinds of D **)** $\Rightarrow$ **(** $n : \gamma \to \delta$ is a concrete diagram morphism in D **)**.

Clearly, in the presence of kinds, the change of shape action of Θ extends naturally to a change of shape action on kinded abstract diagrams, provided the change of shape morphisms $\underline{\alpha} : \underline{\nu} \to \underline{\mu}$ are kind-non-increasing in the partial order $\mathsf{id} \le \mathsf{std} \le \mathsf{iso}$. Also in the presence of kinds, the subdiagram operation requires kind-non-increasingness in order to be well defined. In the presence of kinds, the pasting operation is well defined if the image of ρ in $D_0 \oplus_\rho D_1$ is kinded with the infimum of the kinds of $\alpha(\rho)$ and $\beta(\rho)$. Finally, in the local pullback construction, the fresh vertex always acquires kind iso.

The notion of maximal kinded abstract diagrams, particularly that of kinded abstract diagrams entirely of kind std, raises the question of the relationship between these and the concrete diagrams one can construct in the category $\langle C \rangle$. To answer this, and related questions, we note first that as $\langle C \rangle$ is isomorphic to C^K, standard isomorphisms in $\langle C \rangle$ are just identities; so there is no distiction between abstract diagrams in $\langle C \rangle$ entirely of kind std and those entirely of kind id. Now Fig. 6 provides a route map between the possibilities of interest. The left column depicts concrete diagrams, the middle column depicts abstract diagrams conveniently related to concrete ones, and the right column depicts general abstract diagrams. The top two rows show the situation for C, and the bottom row shows the situation for $\langle C \rangle$ (we do not bother with the situation for C^K). The connections between the various possibilities are described by families of functors $\Im_\mu^{-,-}$ relating categories of diagrams of shape μ of various kinds. For the rest of this discussion we will suppress mention of μ, and for abstract diagrams, which we will assume maximal, we will restrict to the subcategories in which all abstract diagram morphisms are mediated by families of arrows of C.

We recall that an object of $\langle C \rangle$ is an equivalence class of objects of C containing in particular a unique skeleton object from C^K, and that an arrow of $\langle C \rangle$ is an equivalence class of arrows of C in bijective correspondence with ordered pairs of representatives from its domain and codomain objects. We start with the relationship between concrete dia-

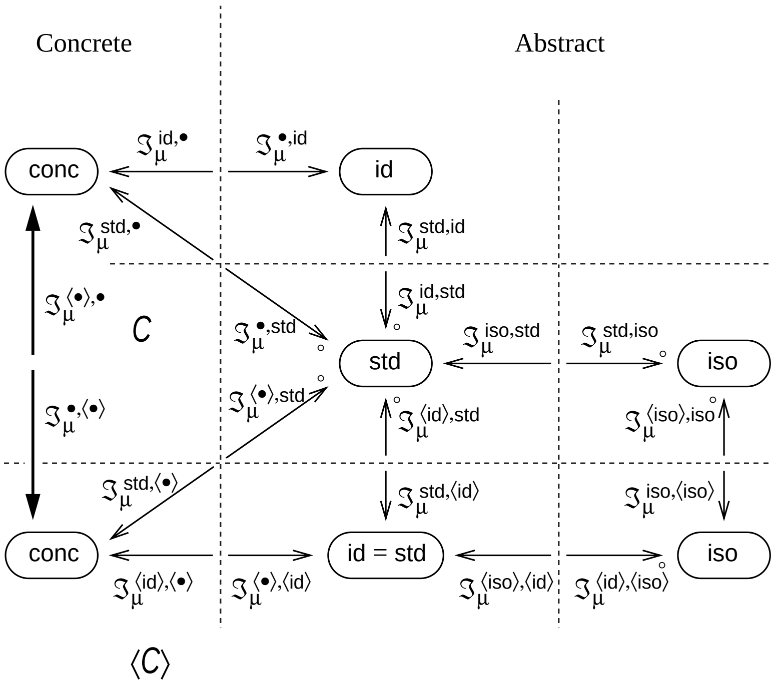

Fig. 6

grams in C, and concrete diagrams in $\langle C\rangle$. This is an easy extrapolation of the situation studied in detail for Gr in Corradini et al. (1994a). Thus $\Im_\mu^{\bullet,\langle\bullet\rangle}$ takes a concrete diagram γ in C to the concrete diagram $\gamma^{\langle\rangle}$ in $\langle C\rangle$, for which the objects and arrows of γ are members of the equivalence classes which constitute the objects and arrows of $\gamma^{\langle\rangle}$. Conversely $\Im_\mu^{\langle\bullet\rangle,\bullet}$ sends a $\langle C\rangle$ diagram $\gamma^{\langle\rangle}$ to the concrete diagram γ in C, for which the objects are the skeleton objects drawn from the equivalence class objects of $\gamma^{\langle\rangle}$, and the arrows are the unique arrows between the skeleton objects drawn from the arrow equivalence classes of $\gamma^{\langle\rangle}$. $\Im_\mu^{\bullet,\langle\bullet\rangle}$ and $\Im_\mu^{\langle\bullet\rangle,\bullet}$ constitute an equivalence of categories.

Proceeding to the top row of Fig. 6, we have the isomorphism between concrete diagrams in C and abstract diagrams entirely of kind id in C, given by functors $\Im_\mu^{\bullet,\mathrm{id}}$ and $\Im_\mu^{\mathrm{id},\bullet}$. This is essentially the correspondence between an item and the singleton class containing it. A similar situation prevails on the bottom row between concrete diagrams in $\langle C\rangle$ and abstract diagrams entirely of kind id or std in $\langle C\rangle$, given by functors $\Im_\mu^{\langle\bullet\rangle,\langle\mathrm{id}\rangle}$ and $\Im_\mu^{\langle\mathrm{id}\rangle,\langle\bullet\rangle}$. That these are isomorphisms, follows readily from the only possible action on mediated morphisms of abstract diagrams of kind id.

We next discuss the middle column of Fig. 6. The object map of the functor $\Im_\mu^{\mathrm{std},\mathrm{id}}$ takes a maximal abstract diagram D of kind std, to the singleton containing the unique concrete diagram in D consisting of skeleton objects and arrows between them. The arrow map of the functor $\Im_\mu^{\mathsf{std,id}}$ takes a morphism $c : \mathsf{D}_0 \to \mathsf{D}_1$, to the morphism $\{f\} : \{\gamma\} \to \{\delta\}$ where: γ is the unique concrete diagram in D_0 consisting of skeleton objects and arrows between them; δ is the corresponding one in D_1; and f is the natural transformation given by taking the family of C arrows that mediate c, selecting the subfamily χ that forms the natural transformation at γ, and postcomposing χ with the unique family of standard isomorphisms that takes the codomain of χ to δ.

Conversely the object map of the feeble functor $\Im_\mu^{\mathsf{id,std}}$ takes a singleton containing an individual concrete diagram γ, to the abstract diagram D_0 consisting of the class of concrete diagrams related to γ by families of standard isomorphisms. The arrow map of $\Im_\mu^{\mathsf{id,std}}$ takes a morphism $\{f\} : \{\gamma\} \to \{\delta\}$ between singletons, mediated by a single natural transformation f, and sends it to the class of mediated morphisms determined as follows. Let φ be a function that maps each χ_γ, a natural transformation of γ formed by standard isomorphisms, to a natural transformation χ_δ of δ formed by standard isomorphisms. Such a function determines a morphism $c_\varphi : \mathsf{D}_0 \to \mathsf{D}_1$ of abstract diagrams of kind std, by mapping each concrete diagram in D_0 via $\chi_\delta \circ f \circ \chi_\gamma^{-1}$. The collection of all such morphisms for all possible choices of φ, determines the arrow map of $\Im_\mu^{\mathsf{id,std}}$.

The above makes $\Im_\mu^{\mathsf{id,std}}$ and $\Im_\mu^{\mathsf{std,id}}$ into a weak equivalence of categories, weakness being in the sense that $\Im_\mu^{\mathsf{id,std}}$ is a weak left adjoint to $\Im_\mu^{\mathsf{std,id}}$. The above also fixes the properties of the pair $\Im_\mu^{\bullet,\mathsf{std}}$ and $\Im_\mu^{\mathsf{std},\bullet}$ by requiring that the upper triangle in Fig. 6 commutes in the expected way. This means that $\Im_\mu^{\bullet,\mathsf{std}}$ is feeble and a weak left adjoint to $\Im_\mu^{\mathsf{std},\bullet}$.

Moving down, the object map of the functor $\Im_\mu^{\mathsf{std},\langle\mathsf{id}\rangle}$ takes an abstract diagram D of kind std in C to the singleton containing the concrete diagram γ in $\langle\mathsf{C}\rangle$ formed by: selecting for each vertex m_0 in the shape μ, the isomorphism class of concrete objects of C up to standard isomorphisms, occurring above m_0 in the concrete diagrams of D; and selecting for each edge $e : m_0 \to m_1$ in the shape μ, the isomorphism class of concrete arrows of C up to standard isomorphisms, occurring above e in the concrete diagrams of D. The arrow map of the functor $\Im_\mu^{\mathsf{std},\langle\mathsf{id}\rangle}$ takes a morphism $c : \mathsf{D}_0 \to \mathsf{D}_1$ of abstract diagrams of kind std, and sends it to the morphism $\{[f]_\lambda\} : \{\gamma\} \to \{\delta\}$ between singletons containing γ and δ, the images of D_0 and D_1, as follows. Let $\{f_\lambda\}$ be the family of arrows mediating c at the object Γ of D_0 consisting of skeleton C objects only (and arrows between them), and let $[f]_\lambda$ be the collection of isomorphism classes up to standard isomorphisms of $\{f_\lambda\}$. These are arrows in $\langle\mathsf{C}\rangle$ forming a natural transformation of γ. We write $\{[f]_\lambda\} : \{\gamma\} \to \{\delta\}$ for the natural extension of $[f]_\lambda$ to an action on the singleton $\{\gamma\}$ containing γ.

Conversely the object map of the feeble functor $\Im_\mu^{\langle\mathsf{id}\rangle,\mathsf{std}}$ takes each singleton containing a concrete diagram γ in $\langle\mathsf{C}\rangle$ whose objects and arrows are isomorphism classes of C objects and arrows up to standard isomorphisms, and maps it to the abstract diagram D_0 consisting of the class of concrete diagrams in C which can be constructed by: selecting for each vertex m_0 in the shape μ, an element of the equivalence class which is the object of γ above it; and for each edge $e : m_0 \to m_1$ in the shape μ, selecting the unique element with appropriate domain and codomain, from the equivalence class above e in γ. The arrow map of $\Im_\mu^{\langle\mathsf{id}\rangle,\mathsf{std}}$ takes a morphism $\{[f]_\lambda\} : \{\gamma\} \to \{\delta\}$ between singletons, mediated by a single natural transformation $[f]_\lambda$ consisting of isomorphism classes of C arrows up to standard isomorphisms, containing in particular the collection $\{f_\lambda\}$ all of whose domains and codomains are skeleton objects, and maps it as follows. Let Γ and Δ be the unique concrete diagrams in D_0 and D_1 all of whose objects are skeleton objects. Let φ be a function that maps each χ_Γ, a natural transformation of Γ formed by standard isomorphisms, to χ_Δ a natural transformation of Δ formed by standard isomorphisms. Such a function determines a morphism $c_\varphi : D_0 \to D_1$ of abstract diagrams of kind std, by mapping each concrete diagram in D_0 via $\chi_\Delta \circ f \circ {\chi_\Gamma}^{-1}$. The collection of all such morphisms for all possible choices of φ, determines the arrow map of $\Im_\mu^{\langle\mathrm{id}\rangle,\mathrm{std}}$.

As above $\Im_\mu^{\langle\mathrm{id}\rangle,\mathrm{std}}$ and $\Im_\mu^{\mathrm{std},\langle\mathrm{id}\rangle}$ form a weak equivalence of categories, with $\Im_\mu^{\langle\mathrm{id}\rangle,\mathrm{std}}$ being a weak left adjoint to $\Im_\mu^{\mathrm{std},\langle\mathrm{id}\rangle}$. Requiring that the lower triangle in Fig. 6 commutes also fixes the properties of the pair $\Im_\mu^{\langle\bullet\rangle,\langle\mathrm{id}\rangle}$ and $\Im_\mu^{\langle\mathrm{id}\rangle,\langle\bullet\rangle}$, with $\Im_\mu^{\langle\bullet\rangle,\langle\mathrm{id}\rangle}$ being a weak left adjoint to $\Im_\mu^{\langle\mathrm{id}\rangle,\langle\bullet\rangle}$. We can also see that the rectangle in the left and middle columns of Fig. 6 commutes as we would expect.

We turn to the rectangle in the lower right part of Fig. 6. We observe first the following fact. Suppose in C we have arrows $f : x \to y$, $f' : x' \to y'$, and standard isomorphisms $\sigma(x, x') : x \to x'$, $\sigma(y, y') : y \to y'$, making a commuting square. Let $\tau(x, x') : x \to x'$ be any isomorphism from x to x'. In general there will not be an isomorphism $\tau(y, y') : y \to y'$ making f, f', $\tau(x, x')$, $\tau(y, y')$ commute. However we will assume subsequently that C has enough isomorphisms, in the sense that such a $\tau(y, y')$ can always be found, though it may not be unique. For example Gr has enough isomorphisms.

Now the object map of the feeble functor $\Im_\mu^{\mathrm{std},\mathrm{iso}}$ sends an abstract diagram D^{std} entirely of kind std to the abstract diagram D^{iso} having the same objects, but this time entirely of kind iso. Viewed as a category, D^{iso} has merely acquired more arrows in this process, namely the natural transformations between its concrete diagrams, incorporating at least one nonstandard isomorphism. The arrow map of the feeble functor $\Im_\mu^{\mathrm{std},\mathrm{iso}}$ sends a mediated morphism $c^{\mathrm{std}} : {D_0}^{\mathrm{std}} \to {D_1}^{\mathrm{std}}$ to the class of extensions of c^{std} which cover all the additional natural transformations too. Such extensions will exist by our observation above, but in general they will not be unique.

The object map of the functor $\Im_\mu^{\mathrm{iso},\mathrm{std}}$ likewise sends an abstract diagram D^{iso} entirely of kind iso to the abstract diagram D^{std} having the same objects, but this time entirely of kind std. As a category, D^{iso} is mapped to the subcategory D^{std} having only standard isomorphism natural transformations as arrows. The arrow map of $\Im_\mu^{\mathrm{iso},\mathrm{std}}$ sends a mediated morphism $c^{\mathrm{iso}} : {D_0}^{\mathrm{iso}} \to {D_1}^{\mathrm{iso}}$ to the mediated morphism $c^{\mathrm{std}} : {D_0}^{\mathrm{std}} \to {D_1}^{\mathrm{std}}$ determined as follows. Let Γ be the concrete diagram in ${D_0}^{\mathrm{iso}}$ consisting entirely of skeleton graphs and morphisms between them. (${D_0}^{\mathrm{iso}}$ will contain this since it is maximal.) Let χ_Γ be the collection of C arrows that mediates the morphism c^{iso} at Γ. Let $n : \Gamma \to \gamma$ be an arrow in ${D_0}^{\mathrm{std}}$, and let χ_γ be the corresponding collection of mediating arrows at γ. Suppose $n : \Gamma \to \gamma$ is mapped by c^{iso} to the concrete diagram isomorphism $c^{\mathrm{iso}}(n : \Gamma \to \gamma) : c^{\mathrm{iso}}(\Gamma) \to c^{\mathrm{iso}}(\gamma)$. Let χ_n be the collection of isomorphisms such that $\chi_n \circ c^{\mathrm{iso}}(n)$ is a concrete diagram morphism consisting entirely of standard isomorphisms. Then $\chi_n \circ c^{\mathrm{iso}}(n)$ is a morphism of ${D_1}^{\mathrm{std}}$ mediated by $\chi_n \circ \chi_\gamma$. For each γ in ${D_0}^{\mathrm{iso}}$ we replace its subfamily of mediating arrows by the subfamily $\chi_n \circ \chi_\gamma$ so determined. By the properties of standard isomorphisms, all other morphisms $n : \gamma \to \delta$ in

${D_0}^{\mathsf{std}}$ are mapped to morphisms of ${D_1}^{\mathsf{std}}$ which compose properly. This gives the morphism $c^{\mathsf{std}} : {D_0}^{\mathsf{std}} \to {D_1}^{\mathsf{std}}$.

The functors $\mathfrak{I}_\mu^{\langle \mathsf{id} \rangle, \langle \mathsf{iso} \rangle}$ and $\mathfrak{I}_\mu^{\langle \mathsf{iso} \rangle, \langle \mathsf{id} \rangle}$ are similar. The object map of the feeble functor $\mathfrak{I}_\mu^{\langle \mathsf{id} \rangle, \langle \mathsf{iso} \rangle}$ maps the objects via identities — the objects (up to $\mathsf{id} = \mathsf{std}$) being singletons containing concrete diagrams built out of objects and arrows which are equivalence classes of C objects and arrows up to standard isomorphisms. Up to $\mathsf{id} = \mathsf{std}$, abstract diagrams in $\langle C \rangle$ have only the identity automorphism; however up to iso, they in general acquire nontrivial automorphisms. The arrow map of the feeble functor $\mathfrak{I}_\mu^{\langle \mathsf{id} \rangle, \langle \mathsf{iso} \rangle}$ takes a morphism $\{[f]_\lambda\} : \{\gamma\} \to \{\delta\}$ between singletons, mediated by a single natural transformation $[f]_\lambda$ consisting of isomorphism classes of C arrows up to standard isomorphisms, containing in particular the collection $\{f_\lambda\}$ all of whose domains and codomains are skeleton objects, and maps it as follows. Let Γ and Δ be as constructed in the discussion of $\mathfrak{I}_\mu^{\langle \mathsf{id} \rangle, \mathsf{std}}$. Then each nontrivial automorphism of γ (respectively δ) has a unique representative for Γ (respectively Δ). Moreover, each nontrivial automorphism a_Γ of Γ maps via f_λ to a nontrivial automorphism a_Δ of Δ, in general in many ways. The equivalence classes up to standard isomorphisms, of the objects and arrows of a_Δ, yield an automorphism of δ which gives a possible action of $\mathfrak{I}_\mu^{\langle \mathsf{id} \rangle, \langle \mathsf{iso} \rangle}$ on the arrow $\{[f]_\lambda\}$. The collection of all such possibilities determines the arrow map of $\mathfrak{I}_\mu^{\langle \mathsf{id} \rangle, \langle \mathsf{iso} \rangle}$.

The functor $\mathfrak{I}_\mu^{\langle \mathsf{iso} \rangle, \langle \mathsf{id} \rangle}$ could not be simpler. The action on objects is the identity. On arrows, it is just the restriction to identity automorphisms only, of the action of arrows $\{[f]_\lambda\} : \{\gamma\} \to \{\delta\}$ between singletons.

As we had before, the functor pairs $\mathfrak{I}_\mu^{\mathsf{id}, \mathsf{iso}}$, $\mathfrak{I}_\mu^{\mathsf{iso}, \mathsf{id}}$ and $\mathfrak{I}_\mu^{\langle \mathsf{id} \rangle, \langle \mathsf{iso} \rangle}$, $\mathfrak{I}_\mu^{\langle \mathsf{iso} \rangle, \langle \mathsf{id} \rangle}$ give weak equivalences of categories, in the sense that $\mathfrak{I}_\mu^{\mathsf{id}, \mathsf{iso}}$ is a weak left adjoint to $\mathfrak{I}_\mu^{\mathsf{iso}, \mathsf{id}}$ and $\mathfrak{I}_\mu^{\langle \mathsf{id} \rangle, \langle \mathsf{iso} \rangle}$ is a weak left adjoint to $\mathfrak{I}_\mu^{\langle \mathsf{iso} \rangle, \langle \mathsf{id} \rangle}$.

Finally we consider $\mathfrak{I}_\mu^{\langle \mathsf{iso} \rangle, \mathsf{iso}}$ and $\mathfrak{I}_\mu^{\mathsf{iso}, \langle \mathsf{iso} \rangle}$. The functor $\mathfrak{I}_\mu^{\mathsf{iso}, \langle \mathsf{iso} \rangle}$ behaves like the functor $\mathfrak{I}_\mu^{\mathsf{std}, \langle \mathsf{id} \rangle}$ except that diagram morphisms must include also the nontrivial automorphisms. Each such nontrivial automorphism of a concrete representative of an abstract diagram of kind iso in $\langle C \rangle$, is simply mapped to the collection of equivalence classes up to standard isomorphisms in the expected way. Likewise, the feeble functor $\mathfrak{I}_\mu^{\langle \mathsf{iso} \rangle, \mathsf{iso}}$ behaves like $\mathfrak{I}_\mu^{\langle \mathsf{id} \rangle, \mathsf{std}}$ except that again nontrivial automorphisms must be taken into account. These are mapped just like all the other arrows between abstract diagrams of kind iso in $\langle C \rangle$.

Unsurprisingly the functors $\mathfrak{I}_\mu^{\langle \mathsf{iso} \rangle, \mathsf{iso}}$ and $\mathfrak{I}_\mu^{\mathsf{iso}, \langle \mathsf{iso} \rangle}$ form a weak equivalence of categories with $\mathfrak{I}_\mu^{\langle \mathsf{iso} \rangle, \mathsf{iso}}$ being a weak left adjoint to $\mathfrak{I}_\mu^{\mathsf{iso}, \langle \mathsf{iso} \rangle}$.

The small circles next to arrow heads in Fig. 6 indicate the functors in the above discussion which are feeble. It is worth summarising that the feebleness of $\mathfrak{I}_\mu^{\mathsf{std}, \mathsf{iso}}$ and of $\mathfrak{I}_\mu^{\langle \mathsf{id} \rangle, \langle \mathsf{iso} \rangle}$ is due to the nonunique way that arbitrary nonstandard isomorphisms translate along arbitrary morphisms, while the feebleness of all the other functors in the discourse is attributable to the many different mediated morphisms of abstract diagrams which map, under equivalence up to standard isomorphisms, to the same morphism of skeleton concrete diagrams (say).

The preceding discussion described the situation when all vertices in the shape of an abstract diagram are of the same kind. The facts of the matter for diagrams where the kind varies from vertex to vertex, may be determined by easy extrapolations of the above. Below we will routinely encode the kinds of the vertices of an abstract diagram by the following convention: unadorned vertices imply that the kind is id; vertices in angle brackets imply that the kind is std; and vertices in square brackets imply that the kind is iso. Thus $A \leftarrow \langle B \rangle \rightarrow [C]$ is a notation for an abstract diagram where A occurs up to identity, B occurs up to standard isomorphisms, and C occurs up to arbitrary isomorphisms. The final possibility which will be of practical interest, namely concrete diagrams over the abstract category $\langle \mathsf{C} \rangle$, we will write using superscripted angle brackets thus $A^{\Diamond} \leftarrow B^{\Diamond} \rightarrow C^{\Diamond}$.

4.4 Sufficiently Monic Kinded Abstract Diagrams

Monicity of arrows has a significant effect on the properties of abstract diagrams whose shapes have vertices of kind iso.

Definition 4.4.1 Let D be a maximal abstract diagram of shape μ in C. Suppose for each vertex m_0 in μ, of kind iso, there is a vertex m_1 of kind std or id, and an edge $e_0 : m_0 \rightarrow m_1$ such that for each concrete diagram γ in D, the arrow of γ above e_0 is monic. Then D_0 is sufficiently monic.

Clearly the quantification over γ is somewhat spurious, since if one concrete diagram in D has a monic arrow above e_0 then they all do. Note also that since we spoke of μ rather than $\underline{\mu}$, mere accessibility in $\underline{\mu}$ of a kind std or id vertex from any kind iso vertex will do.

Lemma 4.4.2 Let D be a sufficiently monic maximal abstract diagram of shape μ in C. Then between any concrete diagrams γ_0 and γ_1 in D, there is a unique natural transformation $n : \gamma_0 \rightarrow \gamma_1$.

Proof. Let $e_0 : m_0 \rightarrow m_1$ be an edge of μ with $kind(m_0) = \text{iso}$ and $kind(m_1) \in \{\text{std}, \text{id}\}$, and such that any arrow over e_0 in a concrete diagram of D is monic. Let γ_0 and γ_1 be concrete diagrams of D, $f : A_0 \rightarrow A_1$ be an arrow of γ_0 over e_0, and $g : B_0 \rightarrow B_1$ be an arrow of γ_1 over e_0. Since $kind(m_1) \in \{\text{std}, \text{id}\}$, there is a unique isomorphism of C, $\tau : A_1 \rightarrow B_1$, forming part of any natural transformation $n : \gamma_0 \rightarrow \gamma_1$ in D. Let n_0 and n_1 be two such natural transformations, and let $\upsilon_0 : A_0 \rightarrow B_0$ and $\upsilon_1 : A_0 \rightarrow B_0$ be the respective isomorphisms at A_0. Then we know that $g \circ \upsilon_0 = \tau \circ f = g \circ \upsilon_1$, and by monicity of g, $\upsilon_0 = \upsilon_1$. Because we can derive the same for every m_0 of kind iso in μ, we conclude that there is a unique $n : \gamma_0 \rightarrow \gamma_1$ in D. ☺

Let us define the standardisation μ^{std} of a shape μ, as the shape obtained by reassigning the kind of any kind iso vertices in μ to std.

Lemma 4.4.3 Let D be a sufficiently monic maximal abstract diagram of shape μ in C, and let μ^{std} be the standardisation of μ. Let $|Aut(C_{m_i})|$ be the size of the automorphism group of a C object over vertex m_i in any concrete diagram of D. Then D consists of $\prod_{m_i} |Aut(C_{m_i})|$ individual abstract diagrams of shape μ^{std}, where m_i ranges over kind iso vertices of μ.

Proof. An easy consequence of Lemma 4.4.2. ☺

4.5 Kinded Abstract Diagram Morphisms and Opfibrations

Of special interest to us are concrete and abstract diagram morphisms arising from a particular species of opfibration.

Definition 4.5.1 Let Q be a subcategory of C and $P : Q \to B$ be an opfibration. Let $b : B_0 \to B_1$ be an arrow in the base, and $\gamma_0 : \mu \to P^{-1}(B_0)$ be a concrete diagram in the fibre above B_0. Let χ be a choice of arrows $\chi(m_0) : \gamma_0(m_0) \to Q_1$, opcartesian for $(\gamma_0(m_0), b)$, one for each vertex m_0 in μ. Then the diagram $\gamma_1 : \mu \to P^{-1}(B_1)$ is given by:

(1) mapping each vertex m_0 of μ to the codomain Q_1, of the arrow $\chi(m_0) : \gamma_0(m_0) \to Q_1$ chosen for m_0 by χ; thus $Q_1 = \gamma_1(m_0)$,

(2) mapping each edge $e_0 : m_0 \to m_0'$ to the unique arrow $n_1 : Q_1 \to Q_1'$ such that $n_1 \circ \chi(m_0) = \chi(m_0') \circ n_0$, where: $n_0 : \gamma_0(m_0) \to \gamma_0(m_0')$ is the arrow that $e_0 : m_0 \to m_0'$ maps to under γ_0; $\chi(m_0) : \gamma_0(m_0) \to Q_1$ and $\chi(m_0') : \gamma_0(m_0') \to Q_1'$ are the opcartesian arrows chosen for m_0 and m_0' by χ; and uniqueness follows from the opcartesian property of $\chi(m_0')$; thus $n_1 : Q_1 \to Q_1' = n_1 : \gamma_1(m_0) \to \gamma_1(m_0')$; and the construction extends in the obvious way to paths in μ.

That γ_1 is a diagram is easy to see, as is the fact that χ induces a diagram morphism $c : \gamma_0 \to \gamma_1$.

Definition 4.5.2 Let Q be a subcategory of C containing all isomorphisms between any two of its objects, and let $P : Q \to B$ be an opfibration. Let $b : B_0 \to B_1$ be an arrow in the base and let D_0 be an abstract diagram of shape μ in the fibre $P^{-1}(B_0)$ above B_0, by which we mean that D_0 is a subcategory of $[\mu, C]$ such that all the objects and arrows of D_0 (which are concrete diagrams and diagram morphisms in C) are concrete diagrams and diagram morphisms in $P^{-1}(B_0)$. Let χ be a choice of arrows $\chi(m_0, \lambda) : \gamma_{0,\lambda}(m_0) \to Q_1$, opcartesian for $(\gamma_{0,\lambda}(m_0), b)$, one for each vertex m_0 of each concrete diagram $\gamma_{0,\lambda}$ in D_0 (where λ indexes the objects of D_0), satisfying the following conditions for each vertex m_0 in μ:

(1) $kind(m_0) = \mathsf{id} \iff$ for each concrete diagram morphism $n_{0,\lambda,\lambda'} : \gamma_{0,\lambda} \to \gamma_{0,\lambda'}$ in D_0, we have that $\chi(m_0, \lambda) = \chi(m_0, \lambda')$, i.e. $\mathrm{id}_{E_1} \circ \chi(m_0, \lambda) = \chi(m_0, \lambda') \circ \mathrm{id}_{\gamma_{0,\lambda}(m_0)}$, where $n_{0,\lambda,\lambda'}(m_0) : \gamma_{0,\lambda}(m_0) \to \gamma_{0,\lambda'}(m_0) = \mathrm{id}_{\gamma_{0,\lambda}(m_0)}$ is the identity at $\gamma_{0,\lambda}(m_0)$ which is the component of the concrete diagram morphism $n_{0,\lambda,\lambda'}(m_0) : \gamma_{0,\lambda}(m_0) \to \gamma_{0,\lambda'}(m_0)$ at m_0, and $n_{1,\lambda,\lambda'}(m_0) : Q_1 \to Q_1 = \mathrm{id}_{E_1}$ is the identity at E_1.

(2) $kind(m_0) = \mathsf{std} \iff$ for each concrete diagram morphism $n_{0,\lambda,\lambda'} : \gamma_{0,\lambda} \to \gamma_{0,\lambda'}$ in D_0, we have that $\chi(m_0, \lambda) : \gamma_{0,\lambda}(m_0) \to Q_1$ and $\chi(m_0, \lambda') : \gamma_{0,\lambda}(m_0) \to Q_1'$ are such that $\sigma(Q_1, Q_1') \circ \chi(m_0, \lambda) = \chi(m_0, \lambda') \circ \sigma(\gamma_{0,\lambda}(m_0), \gamma_{0,\lambda'}(m_0))$, where $n_{0,\lambda,\lambda'}(m_0) : \gamma_{0,\lambda}(m_0) \to \gamma_{0,\lambda'}(m_0) = \sigma(\gamma_{0,\lambda}(m_0), \gamma_{0,\lambda'}(m_0))$ is the standard isomorphism which is the component of the concrete diagram morphism $n_{0,\lambda,\lambda'}(m_0) : \gamma_{0,\lambda}(m_0) \to \gamma_{0,\lambda'}(m_0)$ at m_0, and $n_{1,\lambda,\lambda'}(m_0) : Q_1 \to Q_1' = \sigma(Q_1, Q_1')$ is the standard isomorphism from Q_1 to Q_1'.

(3) $kind(m_0) = \mathsf{iso} \iff$ for each concrete diagram morphism $n_{0,\lambda,\lambda'} : \gamma_{0,\lambda} \to \gamma_{0,\lambda'}$ in D_0, we have that $\chi(m_0, \lambda) : \gamma_{0,\lambda}(m_0) \to Q_1$ and $\chi(m_0, \lambda') : \gamma_{0,\lambda}(m_0) \to Q_1'$ are such that $\tau(Q_1, Q_1') \circ \chi(m_0, \lambda) = \chi(m_0, \lambda') \circ \tau(\gamma_{0,\lambda}(m_0), \gamma_{0,\lambda'}(m_0))$, where $n_{0,\lambda,\lambda'}(m_0) : \gamma_{0,\lambda}(m_0) \to \gamma_{0,\lambda'}(m_0) = \tau(\gamma_{0,\lambda}(m_0), \gamma_{0,\lambda'}(m_0))$ is an arbitrary isomor-

phism which is the component of the concrete diagram morphism $n_{0,\lambda,\lambda'}(m_0) : \gamma_{0,\lambda}(m_0) \to \gamma_{0,\lambda'}(m_0)$ at m_0, and $n_{1,\lambda,\lambda'}(m_0) : Q_1 \to Q_1' = \tau(Q_1, Q_1')$ is an arbitrary isomorphism from Q_1 to Q_1'.

It is clear that for any concrete diagram $\gamma_{0,\lambda}$ in D_0, the choice of $\chi(m_0, \lambda)$ ranging over the vertices m_0 of μ, provides a concrete diagram $\gamma_{1,\lambda}$ and a concrete diagram morphism $c_\lambda : \gamma_{0,\lambda} \to \gamma_{1,\lambda}$ as per Definition 4.5.1. It is equally clear that conditions (1)-(3) guarantee that natural transformations $n_{0,\lambda,\lambda'} : \gamma_{0,\lambda} \to \gamma_{0,\lambda'}$ between the concrete diagrams $\gamma_{0,\lambda}$ in D_0 are mapped to natural transformations of the same kind $n_{1,\lambda,\lambda'} : \gamma_{1,\lambda} \to \gamma_{1,\lambda'}$ between the concrete diagrams $\gamma_{1,\lambda}$, thus producing an abstract diagram D_1 which conforms to its kind, and an epic abstract diagram morphism $c : D_0 \to D_1$, which is mediated by the family of arrows $\chi(m_0, \lambda)$, with m_0 ranging over vertices of μ and λ ranging over objects of D_0.

For the above to be well defined, we should check that the conditions (1)-(3) are actually feasible. Clearly condition (1) offers no problems. Neither does condition (3), since we can choose the $\chi(m_0, \lambda)$ arbitrarily, safe in the knowledge that if $\gamma_{0,\lambda}(m_0)$ and $\gamma_{0,\lambda'}(m_0)$ are isomorphic, then for any isomorphism $\tau(\gamma_{0,\lambda}(m_0), \gamma_{0,\lambda'}(m_0))$ we will always be able to find the unique isomorphism $\tau(Q_1, Q_1')$ that solves the equation, because Q contains all isomorphisms from Q_1 to Q_1'. Condition (2) requires a little more thought. Suppose there is an m_0 in μ of kind std. Then we choose arbitrarily one object $\Gamma_{0,\lambda*}$ in D_0 to act as reference point (if D_0 is nonempty, otherwise there is nothing to prove). We know that for all $\gamma_{0,\lambda'}$ in D_0, $\sigma(\Gamma_{0,\lambda*}(m_0), \gamma_{0,\lambda'}(m_0))$ is the unique arrow between these two graphs (if D_0 indeed contains an arrow from $\Gamma_{0,\lambda*}$ to $\gamma_{0,\lambda'}$). We take $\chi(m_0, \lambda^*) : \gamma_{0,\lambda*}(m_0) \to Q_1$ as fixed. Now if $\chi(m_0, \lambda')$ satisfies the equation in condition (2) then all well and good; we set $\chi'(m_0, \lambda') = \chi(m_0, \lambda')$. If not, then we will have $\tau(Q_1, Q_1') \circ \chi(m_0, \lambda^*) = \chi(m_0, \lambda') \circ \sigma(\Gamma_{0,\lambda*}(m_0), \gamma_{0,\lambda'}(m_0))$ for some nonstandard isomorphism $\tau(Q_1, Q_1')$. In this case we will have $\sigma(Q_1, Q_1') = a(Q_1', Q_1') \circ \tau(Q_1, Q_1')$ for some unique automorphism $a(Q_1', Q_1')$ of Q_1'. We can now replace the choice of arrow $\chi(m_0, \lambda')$ by $\chi'(m_0, \lambda') = a(Q_1', Q_1') \circ \chi(m_0, \lambda')$, an equally acceptable possibility since opcartesian arrows are unique only up to isomorphism. Doing the same for all λ' gives us a choice χ' of opcartesian arrows such that condition (2) is indeed satisfied.

Let $\Phi(b, \chi_\mu)$ name the construction on (concrete and) abstract diagrams just described, where the shape is μ and D_0 is understood.

Proposition 4.5.3 Let D_0 be an abstract diagram of shape μ, and let $\underline{\alpha} : \underline{\nu} \to \underline{\mu}$ be a kind non increasing shape graph morphism. For λ an index of a concrete diagram $\gamma_{0,\lambda}$ in D_0, and n_0 a vertex of ν, let χ_ν be defined by $\chi_\nu(n_0, \lambda) = \chi_\mu(\alpha(n_0), \lambda)$ (which gives a choice of opcartesian arrows for the induced abstract diagram $E_0 = \Theta(\alpha)(D_0)$). Then $\Phi(b, \chi_\nu) \circ \Theta(\alpha) = \Theta(\alpha) \circ \Phi(b, \chi_\mu)$ as functors from the category of abstract diagrams of shape μ to the arrow category of the category of abstract diagrams of shape ν.

The proof is obvious once one notices that *exactly the same* family of opcartesian arrows is determined by both $\Phi(b, \chi_\nu) \circ \Theta(\alpha)$ and $\Theta(\alpha) \circ \Phi(b, \chi_\mu)$. Clearly the construction is natural in α, but demanding naturality in b too, amounts to splitting the opfibration $P : Q \to B$.

Note that there is a special case of the theory of Definitions 4.5.1-2 in which we impose the additional constraint on all concrete diagrams γ that $\gamma_\lambda(m_0) = \gamma_\lambda(m_1) \Rightarrow \chi(m_0, \lambda) = \chi(m_1, \lambda)$. We will not describe it in detail.

4.6 Arrow Abstract Diagrams and Interface-Diagram Categories

In this section we describe a construction for diagrams which is in many ways analogous to the arrow category construction for ordinary categories. Using it enables us to construct diagrams and categories over larger and larger shapes.

Definition 4.6.1 Let $\underline{\mu}$ be a shape graph. Then $2.\underline{\mu}$ is the shape graph given by:

Vertices: $\{(m, 0), (m, 1) \mid m \in \mathrm{Vert}(\underline{\mu})\}$

Edges: $\{(e, 0) : (m, 0) \to (m', 0) \mid e : m \to m' \in \mathrm{Edg}(\underline{\mu})\} \cup$
$\{(e, 1) : (m, 1) \to (m', 1) \mid e : m \to m' \in \mathrm{Edg}(\underline{\mu})\} \cup$
$\{(m, 01) : (m, 0) \to (m, 1) \mid m \in \mathrm{Vert}(\underline{\mu})\}$

If $\underline{\mu}$ is kinded, then $2.\underline{\mu}$ acquires kinds in the obvious way. The path category of $2.\underline{\mu}$ is $2.\mu$.

Definition 4.6.2 Let γ be a concrete diagram of shape μ in C and let $n : \gamma \to \delta$ be a concrete diagram morphism. Then n defines a concrete diagram $2.\gamma$ of shape $2.\mu$ in C as follows:

$2.\gamma((m, 0)) = \gamma(m)$
$2.\gamma((m, 1)) = \delta(m)$

$2.\gamma((e, 0) : (m, 0) \to (m', 0)) = \gamma(e : m \to m')$
$2.\gamma((e, 1) : (m, 1) \to (m', 1)) = \delta(e : m \to m')$
$2.\gamma((m, 01) : (m, 0) \to (m, 1)) = n(\gamma(m)) : \gamma(m) \to \delta(m)$

We call $2.\gamma$ the arrow concrete diagram induced by n.

Definition 4.6.3 Let $c : D_0 \to D_1$ be a morphism of maximal (kinded) abstract diagrams of shape μ, mediated by a family of arrows Ξ, with associated function χ. Let γ be a concrete diagram of D_0 and $\chi(m, \gamma) : \gamma \to c(\gamma)$, with m ranging over Vert(μ), be the concrete diagram morphism at γ induced by c. Then according to Definition 4.6.2, $\chi(m, \gamma) : \gamma \to c(\gamma)$ defines a concrete diagram $2.\chi_\gamma$, the arrow concrete diagram induced by χ, of shape $2.\mu$.

Let $[2.\chi_\gamma]$ denote the maximal abstract diagram (conforming to the kinds of $2.\mu$) containing $2.\chi_\gamma$. We call $[2.\chi_\gamma]$ the arrow abstract diagram induced by c.

Lemma 4.6.4 The definition of $[2.\chi_\gamma]$ is not affected by the specific choice of γ.

Proof. By the naturality of c and of the structure of abstract diagrams in general. ☺

The arrow concrete/abstract diagram construction gives us a versatile tool for constructing interface-diagram categories, next.

Definition 4.6.5 An interface-diagram category is given by the following data:

(1) Two (kinded) shapes, $\underline{\mu}$ and $\underline{\rho}$, and two monic shape morphisms $s, t : \underline{\rho} \to \underline{\mu}$ (that preserve kinds).

(2) A collection of concrete (resp. maximal (kinded) abstract) diagrams *Obj* of shape ρ, and a collection of nonempty concrete (resp. maximal (kinded) abstract) diagrams *Arr* of shape μ.

(3) For each A in *Arr*, two concrete (resp. maximal (kinded) abstract) subdiagrams induced by s and t, $s(A)$ and $t(A)$, both to be found in *Obj*.

(4) For each O in *Obj*, an element id_O of *Arr*.

(5) A function Ψ, which given A_0 and A_1 in *Arr* and O in *Obj*, and a pasting $A_0 \oplus_\rho A_1$ along ρ via $t(A_0) = O = s(A_1)$ of A_0 and A_1, returns an $A_1 \circ A_0$ in *Arr* with $s(A_1 \circ A_0) = s(A_0)$ and $t(A_1 \circ A_0) = t(A_1)$; and such that the usual identity and associativity laws hold namely:

   (i) For each O in *Obj*, $s(\mathrm{id}_O) = O = t(\mathrm{id}_O)$.

   (ii) For each A in *Arr*, $A \circ \mathrm{id}_{s(A)} = A = \mathrm{id}_{t(A)} \circ A$.

   (iii) For all A_1, A_2, A_3 in *Arr*, $(A_1 \circ A_2) \circ A_3 = A_1 \circ (A_2 \circ A_3)$.

If we are using concrete diagrams, we refer to a concrete interface-diagram category, while if we use abstract diagrams, we refer to an abstract interface-diagram category.

In an interface-diagram category the arrows are concrete (resp. maximal (kinded) abstract) diagrams, and the objects are subdiagrams, the interfaces, along which two arrows may be combined by Ψ, hence the name. Note that were we considering all concrete (resp. maximal (kinded) abstract) diagrams as coexisting within one categorical structure (eg. a Grothendieck category over change of base arrows), Definition 4.6.5 would be almost the definition of an internal category particularised to the case of diagrams, except that we choose to use pushouts (via pasting) rather than pullbacks as is more conventional.

Proposition 4.6.6 Let $\underline{\mu}*$ be a shape graph, and let $C_{2.\underline{\mu}*}$ be the family of all arrow concrete diagrams induced by concrete diagram morphisms between concrete diagrams of shape $\mu*$. Then $C_{2.\underline{\mu}*}$ is an interface-diagram category as follows:

(1) The $\underline{\mu}$ and $\underline{\rho}$ of the interface-diagram category are $2.\underline{\mu}*$ and $\underline{\mu}*$ respectively; $s : \underline{\rho} \to \underline{\mu}$ is the subgraph of shape $\underline{\mu}*$ of $2.\underline{\mu}*$ given by the 0-indexed vertices $(m, 0)$ and the edges between them, likewise $t : \underline{\rho} \to \underline{\mu}$ is the subgraph of shape $\underline{\mu}*$ of $2.\underline{\mu}*$ given by the 1-indexed vertices and edges.

(2) *Obj* consists of concrete diagrams of shape $\mu*$; *Arr* consists of the elements of $C_{2.\underline{\mu}*}$.

(3) For each A in *Arr*, $s(A)$ and $t(A)$ are the subdiagrams of elements of $C_{2.\underline{\mu}*}$ selected by the shape morphisms s and t via Definition 4.2.2.

(4) For each O in *Obj*, id_O in *Arr* is the concrete diagram of shape $2.\mu*$ induced by identity morphisms on concrete diagrams of shape $\mu*$.

(5) Given $A_0 = \gamma$ and $A_1 = \delta$ in *Arr* and O in *Obj*, and a pasting $A_0 \oplus_\rho A_1$ along ρ via $t(A_0) = O = s(A_1)$ of A_0 and A_1, Ψ takes the pasting $A_0 \oplus_\rho A_1$ and returns the concrete subdiagram $\gamma\delta$ of shape $2.\mu*$ constructed as follows (note that this is a subdiagram construction as per Definition 4.2.1):

$$\gamma\delta((m, 0)) = \gamma((m, 0))$$
$$\gamma\delta((m, 1)) = \delta((m, 1))$$

$$\gamma\delta((e, 0) : (m, 0) \to (m', 0)) = \gamma((e, 0) : (m, 0) \to (m', 0))$$
$$\gamma\delta((e, 1) : (m, 1) \to (m', 1)) = \delta((e, 1) : (m, 1) \to (m', 1))$$
$$\gamma\delta((m, 01) : (m, 0) \to (m, 1)) = \delta((m, 01) : (m, 0) \to (m, 1)) \circ \gamma((m, 01) : (m, 0) \to (m, 1))$$

and extending naturally to the path category $2.\mu*$.

Proof. Easy. ☺

Definition 4.6.7 $C_{2.\mu*}$ is the concrete interface-diagram category generated by $\mu*$.

Proposition 4.6.8 Let $\underline{\mu}*$ be a shape graph, and let $C_{[2.\mu*]}$ be the family of all arrow abstract diagrams induced by abstract diagram morphisms between maximal (kinded) abstract diagrams of shape $\mu*$. Then $C_{[2.\mu*]}$ is an interface-diagram category as follows:

(1) The $\underline{\mu}$ and $\underline{\rho}$ of the interface-diagram category are $2.\underline{\mu}*$ and $\underline{\mu}*$ respectively; $s : \underline{\rho} \to \underline{\mu}$ is the subgraph of shape $\underline{\mu}*$ of $2.\underline{\mu}*$ given by the 0-indexed vertices $(m, 0)$ and the edges between them, likewise $t : \underline{\rho} \to \underline{\mu}$ is the subgraph of shape $\underline{\mu}*$ of $2.\underline{\mu}*$ given by the 1-indexed vertices and edges.

(2) *Obj* consists of maximal (kinded) abstract diagrams of shape $\mu*$; *Arr* consists of the elements of $C_{[2.\mu*]}$.

(3) For each A in *Arr*, $s(A)$ and $t(A)$ are the subdiagrams of elements of $C_{[2.\mu*]}$ selected by the shape morphisms s and t via Definition 4.2.2.

(4) For each O in *Obj*, id_O in *Arr* is the maximal abstract diagram of shape $2.\mu*$ induced by identity morphisms on abstract diagrams of shape $\mu*$.

(5) Given A_0 and A_1 in *Arr* and O in *Obj*, and a pasting $A_0 \oplus_\rho A_1$ along ρ via $t(A_0) = O = s(A_1)$ of A_0 and A_1, Ψ takes the pasting $A_0 \oplus_\rho A_1$ and returns the maximal abstract subdiagram of shape $2.\mu*$ given by taking all concrete diagrams $\gamma\delta$ of shape $2.\mu*$ (and appropriate isomorphisms between them), constructed as in clause (5) of Definition 4.6.6, from pastings $\gamma \oplus_\rho \delta$ in $A_0 \oplus_\rho A_1$ of compatible consistent concrete diagrams γ and δ from A_0 and A_1 respectively. (Note that this is a subdiagram construction as per Definition 4.2.2).

Proof. Easy. ☺

Definition 4.6.9 $C_{[2.\mu*]}$ is the abstract interface-diagram category generated by $\mu*$.

The above results will provide powerful tools for the constructions we wish to make below. Note the close analogy that has arisen between operations on concrete and abstract diagrams.

5 Abstract Spans and Other Abstract Diagrams in *Gr*

We now apply the preceding to the case where C is *Gr*, and in particular to spans. Spans, i.e. coinitial pairs of arrows, arise in the algebraic theory of graph rewriting from two independent sources. Firstly, graph productions are defined in the algebraic, double pushout approach (Ehrig (1979)) as spans in the category of graphs. Secondly, in the

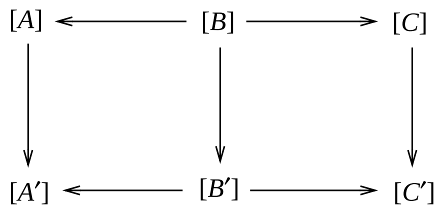

Fig. 7

definition of typed graph grammar morphisms in Corradini et al. (1996b), spans are used to relate the type graph components of grammars.

We fix the shape digraph for spans to be $\bullet_1 \leftarrow \blacklozenge \rightarrow \bullet_2$ which we call $\underline{\eta}$.

Until further notice, the kinds of all vertices will be iso.

For the rest of the paper, we will frequently suppress names of arrows when they are not crucial to the discourse or can be infered.

Definition 5.1 An abstract span is an abstract diagram of the form $[A] \leftarrow [B] \rightarrow [C]$, i.e. an abstract diagram of shape η with all kinds iso.

Definition 5.2 The category [*Sp*] is the interface-diagram category generated by η.

[*Sp*] is the category of abstract span morphisms. When we need to be explicit, we can write a morphism of [*Sp*] using the notation

$([A] \leftarrow [B] \rightarrow [C])$ -[a,b,c]-› $([A'] \leftarrow [B'] \rightarrow [C'])$

where -[a,b,c]-› is a notation for $(a : A \rightarrow A',\ b : B \rightarrow B',\ c : C \rightarrow C')$, three concrete graph morphisms representing the abstract diagram morphism from $[A] \leftarrow [B] \rightarrow [C]$ to $[A'] \leftarrow [B'] \rightarrow [C']$ which generates the arrow abstract diagram that is the morphism of [*Sp*] in question. Fig. 7 illustrates.

The local pullback construction of Definition 4.2.5 permits us to build another interface-diagram category from abstract spans.

Definition 5.3 The category [*Gr-Sp*] is the interface-diagram category given by the following data:

(1) The $\underline{\mu}$ and $\underline{\rho}$ of the[*Gr-Sp*] are $\underline{\eta}$ and $\bullet$ respectively; $s : \underline{\rho} \rightarrow \underline{\mu}$ is $\{\bullet \mapsto \bullet_1\}$ while $t : \underline{\rho} \rightarrow \underline{\mu}$ is $\{\bullet \mapsto \bullet_2\}$.

(2) *Obj* is [*Gr*], graphs up to arbitrary isomorphisms; *Arr* is the object class of [*Sp*], i.e. abstract spans.

(3) For each *A* = $[A] \leftarrow [B] \rightarrow [C]$ in *Arr*, s(*A*) and t(*A*) are the subdiagrams $[A]$ and $[C]$ respectively.

(4) For each $[A]$ in *Obj* , $\mathrm{id}_{[A]}$ in *Arr* is $[A] \leftarrow [A] \rightarrow [A]$ where the arrows are (isomorphism images of) identities on A.

(5) Given $A_0 = [A] \leftarrow [B] \rightarrow [C]$ and $A_1 = [C] \leftarrow [D] \rightarrow [E]$ in *Arr* and $[C]$ in *Obj*, and a pasting $A_0 \oplus_\rho A_1$ along ρ via $t(A_0) = [C] = s(A_1)$ of A_0 and A_1, Ψ takes the pasting $A_0 \oplus_\rho A_1$ and returns the maximal abstract subdiagram $((A_0 \oplus_\rho A_1)_{\text{lpb}})_\eta$ of shape η given by the following procedure:

(i) Form the local pullback of $[B] \rightarrow [C] \leftarrow [D]$ inside $A_0 \oplus_\rho A_1$, yielding $[B] \leftarrow [M] \rightarrow [D]$ inside a bigger abstract diagram $(A_0 \oplus_\rho A_1)_{\text{lpb}}$.

(ii) Let morphism $\underline{\alpha}$ from $\underline{\eta}$ to the shape of $(A_0 \oplus_\rho A_1)_{\text{lpb}}$ be given by:

$$\{\bullet_1 \mapsto \bullet_{1(0)}, \blacklozenge \mapsto \blacklozenge, \bullet_2 \mapsto \bullet_{2(1)}\}$$

where the (0) and (1) subscripts indicate the component of the pasting, and with the obvious extension to edges.

(iii) Let $((A_0 \oplus_\rho A_1)_{\text{lpb}})_\eta$ be the subdiagram yielded by α.

Proposition 5.4 [*Gr-Sp*] is a category.

Proof. Easy, by pullback properties. ☺

The two methods of composition involving spans can be brought together in a single structure.

Definition 5.5 A double interface-diagram category is a double category which is an interface-diagram category with respect to both horizontal and vertical composition. A double interface-diagram category is concrete or abstract according to whether the diagrams it is built out of are concrete or abstract.

For double categories see eg. Ehresmann (1963), Palmquist (1970), Bastiani and Ehresmann (1974); a tutorial treatment also appears in Gadducci and Montanari (1995).

Definition 5.6 The abstract double interface-diagram category [*D-Gr-Sp*] has as double cells abstract diagrams of the shape in Fig. 7, i.e. morphisms of [*Sp*]. As before we write such cells as $\mathbf{a} = (([A] \leftarrow [B] \rightarrow [C])$ -$[a,b,c]$-› $([A'] \leftarrow [B'] \rightarrow [C']))$. Vertical composition $*_\text{v}$ of double cells $\mathbf{a} = (([A] \leftarrow [B] \rightarrow [C])$ -$[a,b,c]$-› $([A'] \leftarrow [B'] \rightarrow [C']))$ and $\mathbf{a}' = (([A'] \leftarrow [B'] \rightarrow [C'])$ -$[a',b',c']$-› $([A''] \leftarrow [B''] \rightarrow [C'']))$ is the composition of [*Sp*], giving the double cell $(\mathbf{a} *_\text{v} \mathbf{a}') = (([A] \leftarrow [B] \rightarrow [C])$ -$[a'.a,b'.b,c'.c]$-› $([A''] \leftarrow [B''] \rightarrow [C'']))$. Horizontal composition $*_\text{h}$ of double cells $\mathbf{a} = (([A] \leftarrow [B] \rightarrow [C])$ -$[a,b,c]$-› $([A'] \leftarrow [B'] \rightarrow [C']))$ and $\mathbf{b} = (([C] \leftarrow [D] \rightarrow [E])$ -$[c,d,e]$-› $([C'] \leftarrow [D'] \rightarrow [E']))$ is given by: pasting $\mathbf{a}$ and $\mathbf{b}$ along $[C] \rightarrow [C']$, making two instances of the composition of [*Gr-Sp*] (at primed and unprimed levels respectively), pasting in the abstract morphism $[M] \rightarrow [M']$ noting that $[M] \rightarrow [M']$ is uniquely given by pullback properties and the requirements of pasting; thus obtaining the double cell $(\mathbf{a} *_\text{h} \mathbf{b}) = (([A] \leftarrow [M] \rightarrow [E])$ -$[a,m,e]$-› $([A'] \leftarrow [M'] \rightarrow [E']))$.

Proposition 5.7 [*D-Gr-Sp*] is an abstract double interface-diagram category.

Proof. Easy, if tedious, by pullback properties. ☺

It is relatively easy if tedious to see that the standard interchange law for double categories $(\mathbf{a} *_\text{v} \mathbf{a}') *_\text{h} (\mathbf{b} *_\text{v} \mathbf{b}') = (\mathbf{a} *_\text{h} \mathbf{b}) *_\text{v} (\mathbf{a}' *_\text{h} \mathbf{b}')$ holds.

The vertical arrows of [*D-Gr-Sp*] are abstract diagrams of shape $[A] \rightarrow [A']$, i.e. abstract graph morphisms, while the horizontal arrows of [*D-Gr-Sp*] are the familiar abstract spans $[A] \leftarrow [B] \rightarrow [C]$. Objects of [*D-Gr-Sp*] are graphs up to isomorphism $[A]$. The identities of horizontal arrows are double cells whose vertical arrows are isomorphisms, while identities of vertical arrows are double cells whose horizontal arrows are isomorphisms. Identities of objects are double cells with both horizontal and vertical arrows isomorphisms. These aspects will shortly prove useful.

6 The Opfibration $[P]: [Gr *\downarrow Gr\text{-}Sp] \to [Gr\text{-}Sp]$ and Others

A typed graph over a (type) graph *TG* is simply an object of the comma category $(Gr\downarrow TG)$, i.e., a graph morphism $G \to TG$. As explained in the introduction, various works (Corradini et al. (1996b), Ribeiro (1996), Heckel et. al (1997)) address, with various techniques, the issue of relating graphs typed over different graphs. By exploiting an opfibrational framework, we propose a solution that aims at the greatest generality. In this section we shall construct an opfibration which will later enable us to have for each abstract type graph $[TG]$, a fibre including all abstract graphs typed over $[TG]$, and where morphisms between abstract type graphs are abstract spans. The opfibrational framework allows us to keep the natural non-determinism of this situation. Since this section sets up technical results needed later, we will avoid refering to "typing" etc., prefering a more neutral terminology.

Definition 6.1 The category $[Gr\downarrow Gr\text{-}Sp]$ is a horizontal subcategory of $[D\text{-}Gr\text{-}Sp]$, (i.e. its objects are abstract graph morphisms $[X] \to [A]$, and its arrows are abstract span morphisms), such that two additional properties hold for every arrow

$$(([X] \leftarrow [Y] \to [Z]) \text{-}[a,b,c]\text{-}\rangle\, ([A] \leftarrow [B] \to [C])) : ([X] \to [A]) \to ([Z] \to [C])$$

of $[Gr\downarrow Gr\text{-}Sp]$ namely that:

(1) The left square *XYBA* of each concrete diagram in the arrow is a pullback.

(2) The right arrow $Y \to Z$ of the source abstract span of each concrete diagram in the arrow is an isomorphism.

We write the second property as $Y = Z$, and as a matter of convention we adopt the notation

$$(([X] \leftarrow [Y] = [Z]) \text{-}[a,b,c]\text{-}\rangle\, ([A] \leftarrow [B] \to [C]))$$

to signify that both properties hold of the abstract span morphism in question.

Lemma 6.2 $[Gr\downarrow Gr\text{-}Sp]$ is an abstract interface-diagram category.

Proof. We just have to check that both additional properties hold for the composition inherited from $[D\text{-}Gr\text{-}Sp]$. For the first consider Fig. 8.

For any concrete diagram in the abstract composite, *YZUW* and *BCDM* are pullbacks by construction, and *ZCDU* is a pullback by hypothesis. Hence *YZCDUW* is a pullback. But then *YBCDMW* is a pullback because the concrete diagram commutes. But then since *BCDM* is a pullback, so must *YBMW* be, by pullback properties. Finally since *XABY* is a pullback by hypothesis, the combination *XABMW* is a pullback, and this gives the first property.

The second property is obvious from Fig. 8; the pullback of two " $\leftarrow$ $=$ " spans is another such span. ☺

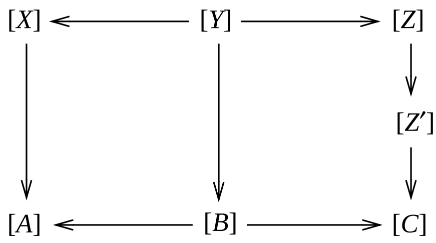

Fig. 9

Consider now the shape graph $\underline{\eta^*}$ which has vertices $\{0, 1, 2, *\}$ and edges $\{1 \mapsto 0, 1 \mapsto 2, 2 \mapsto *\}$. The related shape graph $\underline{\eta^*{\downarrow}\eta}$ has vertices $\{0, 1, 2, *, \bullet_1, \blacklozenge, \bullet_2\}$ and edges $\{1 \mapsto 0, 1 \mapsto 2, 2 \mapsto *, \blacklozenge \mapsto \bullet_1, \blacklozenge \mapsto \bullet_2, 0 \mapsto \bullet_1, 1 \mapsto \blacklozenge, * \mapsto \bullet_2\}$. The path categories are η^* and $\eta^*{\downarrow}\eta$. The graph $\underline{\eta^*{\downarrow}\eta}$ is like $2.\underline{\eta}$ with an extra vertex and different names. Fig. 9 illustrates an abstract diagram of shape $\eta^*{\downarrow}\eta$.

As above we consider a special case of abstract diagrams of shape $\eta^*{\downarrow}\eta$, in which the two properties stated in Definition 6.1 hold for the square $XYBA$ and the arrow $Y \to Z$. We will write abstract diagrams possessing these two properties as:

$$([X] \leftarrow [Y] = [Z] \to [Z']) \text{ -}[a,b,c,c']\text{-}\rightarrowtail ([A] \leftarrow [B] \to [C])$$

where $c : Z \to Z'$ and $c' : Z' \to C$ are representative arrows over $2 \mapsto *$ and $* \mapsto \bullet_2$ respectively.

As before, we will build an abstract interface-diagram category with such diagrams as arrows, but first we need a construction that will enable us to define the composition function Ψ for them.

Construction 6.3 Let $([X] \leftarrow [Y] = [Z] \to [Z'])$ -$[a,b,c,c']$-› $([A] \leftarrow [B] \to [C])$ and $([Z'] \leftarrow [U'] = [V'] \to [V''])$ -$[c',d',e',e'']$-› $([C] \leftarrow [D] \to [E])$ be two abstract diagrams of shape $\eta^*{\downarrow}\eta$ with the required properties, pasted along $[Z'] \to [C]$. The abstract dia-

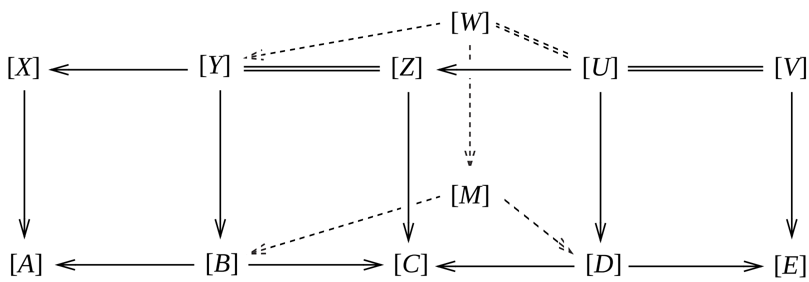

Fig. 8

gram ([X] ← [W] ═ [V] → [V'']) -[a,m,e'.e,e'']-› ([A] ← [M] → [E]) of shape η*↓η is formed as follows (see Fig. 10).

(1) Form the local pullback of [Z] → [Z'] and [Z'] ← [U'] giving [Z] ← [U] and [U] → [U'], the latter with representative $d : U \to U'$.

(2) Form the local pullback of [U] → [U'] and [U'] ═ [V'] giving [U] ═ [V] and [V] → [V'], the latter with representative $e : V \to V'$.

(3) Form the local pullback of [Y] ═ [Z] and [Z] ← [U] giving [Y] ← [W] and [W] ═ [U].

(4) Form the local pullback of [B] → [C] and [C] ← [D] giving [D] ← [M] and [B] → [M].

(5) Paste in [W] → [M], the unique (by pullback properties) abstract morphism that makes the result commute.

(6) Take the obvious subdiagram of shape η*↓η of Fig. 10 which yields the required ([X] ← [W] ═ [V] → [V'']) -[a,m,e'.e,e'']-› ([A] ← [M] → [E]).

Lemma 6.4 The abstract diagram built in Construction 6.3, ([X] ← [W] ═ [V] → [V'']) -[a,m,e'.e,e'']-› ([A] ← [M] → [E]), has the two properties of Definition 6.1.

Proof. It is sufficient to note that since $Z'CDU'$ is a pullback by hypothesis, and $ZZ'U'U$ is a pullback by construction, then $ZCDU$ is a pullback. Now the proof of Lemma 6.2 can be used unaltered. ☺

Definition 6.5 The category [*Gr* *↓*Gr-Sp*] is the abstract interface-diagram category whose objects are abstract graph morphisms [X] → [A], and whose arrows are abstract diagrams ((([X] ← [Y] ═ [Z] → [Z']) -[a,b,c,c']-› ([A] ← [B] → [C])) : ([X] → [A]) → ([Z'] → [C]) of shape η*↓η with [Y] ═ [Z] an abstract isomorphism. Composition is according to the Ψ implicit in Construction 6.2. Identities are arrows with [X] ← [Y] ═ [Z] → [Z'] all isomorphisms and [A] ← [B] → [C] also all isomorphisms.

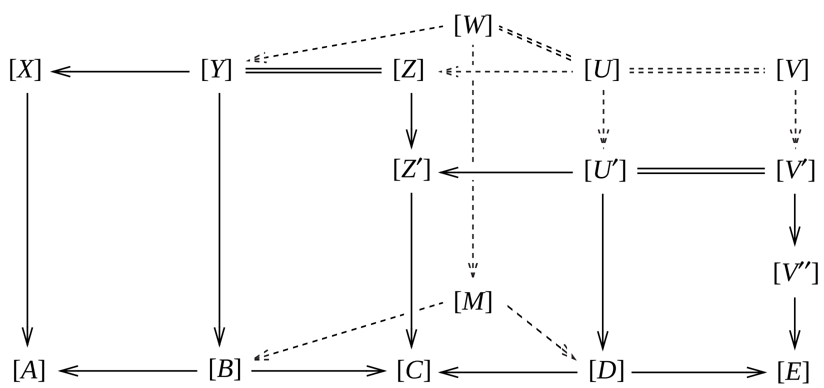

Fig. 10

Lemma 6.6 [*Gr* *↓*Gr-Sp*] is an abstract interface-diagram category.

Proof. The detailed components making up Definition 4.6.4 are easy to check. The only nontrivial part is associativity, which requires a somewhat tedious calculation using pullback properties. ☺

We now come to the first main results of this section.

Theorem 6.7 The projection [*P*] : [*Gr*↓*Gr-Sp*] → [*Gr-Sp*] that takes

(([*X*] ← [*Y*] ═ [*Z*]) -[*a*,*b*,*c*]-› ([*A*] ← [*B*] → [*C*])) : ([*X*] → [*A*]) → ([*Z*] → [*C*])

to ([*A*] ← [*B*] → [*C*]) : [*A*] → [*C*] is a split opfibration, where all arrows of [*Gr*↓*Gr-Sp*] are opcartesian and belong to the splitting.

Proof. We easily see that *P* is a functor, so we need to check the conditions for opcartesian arrows and the splitting. See Fig. 11.

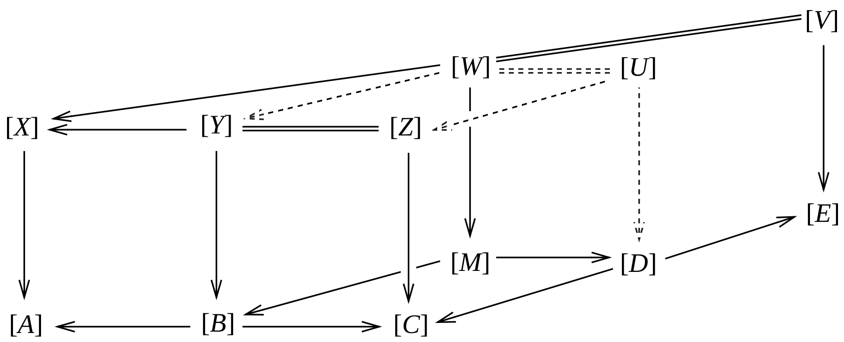

Fig. 11

To check the conditions for opcartesian arrows we paste the abstract diagrams ([*X*] ← [*Y*] ═ [*Z*]) -[*a*,*b*,*c*]-› ([*A*] ← [*B*] → [*C*]) and ([*X*] ← [*W*] ═ [*V*]) -[*a*,*m*,*e*]-› ([*A*] ← [*M*] → [*E*]) along [*X*] → [*A*], knowing that ([*A*] ← [*M*] → [*E*]) is the composition of ([*A*] ← [*B*] → [*C*]) and ([*C*] ← [*D*] → [*E*]). This gives the solid part of Fig. 11. We need to show that we can paste in the abstract diagram ([*Z*] ← [*U*] ═ [*V*]) -[*c*,*d*,*e*]-› ([*C*] ← [*D*] → [*E*]), i.e. the dashed part of Fig. 11, in the appropriate way.

Since for any concrete representative, *XABY* is a pullback by hypothesis and *W* → *X* → *A* and *W* → *M* → *B* → *A* close *X* → *A* ← *B*, we can paste in a unique [*W*] → [*Y*] by pullback properties. Next we form the local pullback of [*Z*] → [*C*] ← [*D*], giving [*Z*] ← [*U*] → [*D*]. Since for any concrete representative, *W* → *Y* → *B* → *C* and *W* → *M* → *D* → *C* close *B* → *C* ← *D*, we can paste in a unique [*W*] → [*U*] by pullback properties, though we don't yet know it is an abstract isomorphism as illustrated.

Now *XABY* is a pullback as noted previously, and *XAMW* is another, by hypothesis. So *YBMW* is a pullback by pullback properties. Combining this with the pullback *MBCD* gives *YBCDMW* as a pullback, and thus *YZCDUW* is a pullback. Since *ZCDU* is a pullback by construction, we conclude that *WYZU* is a pullback by pullback properties.

This enables us to conclude that since $[Y] = [Z]$ is an abstract isomorphism, $[W] = [U]$ must also be one. Hence $[U] = [V]$ is an abstract isomorphism as required. We see that we have pasted in $([Z] \leftarrow [U] = [V])$ -$[c,d,e]$-› $([C] \leftarrow [D] \rightarrow [E])$ as needed.

We now present the opcleavage that gives a splitting. Given an object $([X] \rightarrow [A])$ of [*Gr*↓*Gr-Sp*], projecting down to the domain of an arrow $([A] \leftarrow [B] \rightarrow [C]) : [A] \rightarrow [C]$ of [*Gr-Sp*], we define $\kappa(([X] \rightarrow [A]), ([A] \leftarrow [B] \rightarrow [C]))$ to be

$$(([X] \leftarrow [Y] = [Y]) \text{ -}[a,b,c]\text{-› } ([A] \leftarrow [B] \rightarrow [C])) : ([X] \rightarrow [A]) \rightarrow ([Y] \rightarrow [C])$$

where $[Y]$ comes from the local pullback of $[X] \rightarrow [A] \leftarrow [B]$ and $[Y] = [Y]$ is the identity abstract isomorphism. That all arrows of [*Gr*↓*Gr-Sp*] belong to the splitting follows from the uniqueness of the local pullback construction noted above, and from the fact that if $[Y] = [Z]$ is an abstract isomorphism then $[Y] = [Z]$ is just the same thing as $[Y] = [Y]$. We are done. ☺

Theorem 6.8 The projection $[P_*] : [Gr *{\downarrow}Gr\text{-}Sp] \rightarrow [Gr\text{-}Sp]$ that takes

$$(([X] \leftarrow [Y] = [Z] \rightarrow [Z']) \text{ -}[a,b,c,c']\text{-› } ([A] \leftarrow [B] \rightarrow [C])) :$$
$$([X] \rightarrow [A]) \rightarrow ([Z'] \rightarrow [C])$$

to $[A] \leftarrow [B] \rightarrow [C] : [A] \rightarrow [C]$ is a split opfibration, where all arrows of [*Gr* ∗↓*Gr-Sp*] for which $[Z] \rightarrow [Z']$ is an abstract isomorphism are opcartesian and belong to the splitting.

Proof. This a marginally more elaborate version of the preceding. Again it is clear that $[P_*]$ is a functor so we just need to check the opfibration condition and the splitting. See Fig. 12.

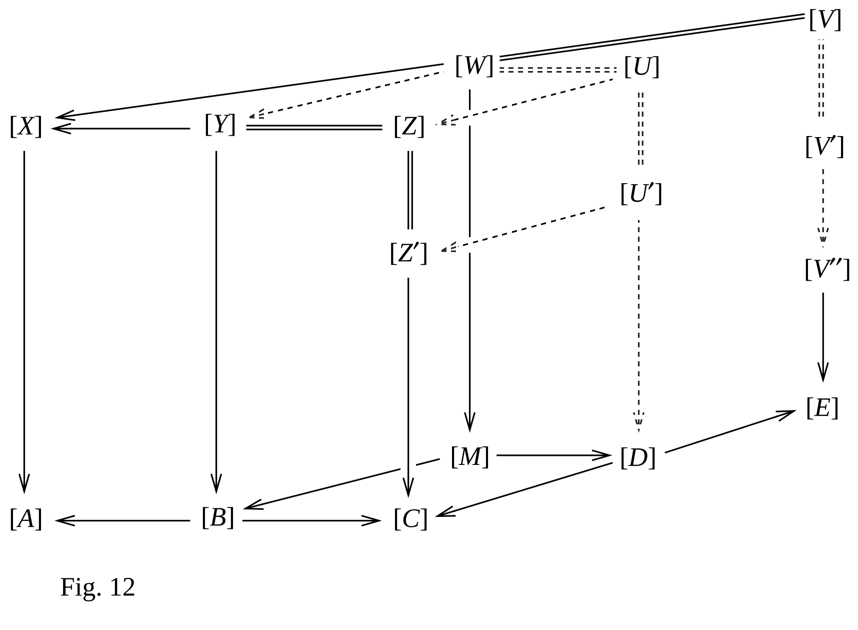

Fig. 12

For this we paste the abstract diagrams ([X] ← [Y] ═ [Z] ═ [Z']) -[a,b,c,c']-› ([A] ← [B] → [C]) and ([X] ← [W] ═ [V] → [V'']) -[a,m,$\tilde{e}$,e'']-› ([A] ← [M] → [E]) along [X] → [A], knowing that ([A] ← [M] → [E]) is the composition of ([A] ← [B] → [C]) and ([C] ← [D] → [E]), giving the solid part of Fig. 12. We need to paste in the abstract diagram ([Z'] ← [U'] ═ [V'] → [V'']) -[c,d,e,e']-› ([C] ← [D] → [E]), such that $\tilde{e} = e' \circ e$, and the relevant [*Gr* *↓*Gr-Sp*] composition properties hold.

The argument goes as per the previous theorem until the point that the local pullback of [Z] → [C] ← [D] is built. Here instead, the local pullback of [Z'] → [C] ← [D] is formed, followed by the local pullback of [Z] ═ [Z'] ← [U']. Since [Z] ═ [Z'] is an abstract isomorphism, the additional structure propagates through the rest of the proof without difficulty and we leave the details to the reader.

For the splitting, we define κ(([X] → [A]), ([A] ← [B] → [C])) to be

(([X] ← [Y] ═ [Y] ═ [Y]) -[a,b,id,c]-› ([A] ← [B] → [C])) :
([X] → [A]) → ([Y] → [C])

where [Y] is the local pullback of [X] → [A] ← [B] as before. That this works, and that all [*Gr* *↓*Gr-Sp*] arrows with [Z] → [Z'] an abstract isomorphism are included, is for the reasons quoted in the previous proof. We are done. ☺

We see that modulo an extra mention of abstract isomorphisms [Y] ═ [Y], [*Gr*↓*Gr-Sp*] is the opcartesian subcategory of [*Gr* *↓*Gr-Sp*]. So we have created the second opfibration by identifying first the opcartesian arrows, and then enhancing this to include further in-fibre arrows (the arrows [Z] → [Z'], "in-fibre" implying that there is a morphism [Z] → [Z'] → [C]), and showing that this preserved the opcartesian properties. This is an example of our nondeterministic analogue of the Grothendieck construction.

Consider the fibre in [*Gr* *↓*Gr-Sp*] above an object [A] of [*Gr-Sp*]. It consists of objects [X] → [A] and of those [*Gr* *↓*Gr-Sp*] arrows that project down to identities on [A] in [*Gr-Sp*]. Such arrows look like (([X] ← [X] ═ [X] → [X']) -[x,x,id,x']-› ([A] ← [A] → [A])) : ([X] → [A]) → ([X'] → [A]). There is clearly an isomorphism between these fibres and the abstract interface-diagram comma categories ([*Gr*]↓[A]) with objects [X] → [A] and arrows [X] → [X'] such that [X] → [X'] → [A] commutes in the expected way. Via this isomorphism, we can use the results of Section 4.5 to claim that whenever we have a concrete diagram in ([*Gr*]↓[A]), and we choose an opcartesian arrow of [*Gr* *↓*Gr-Sp*] for each object in the diagram such that there is a common arrow b of [*Gr-Sp*] to which they all project, then this yields a concrete diagram morphism that is opcartesian for all morphisms of the diagram which project to extensions of b.

We make all this more precise in the following manner. For convenience we will use ♣ as a variable that ranges over the vertices $\bullet_1$, ♦, $\bullet_2$ of $\underline{\eta}$. Let (2.$\underline{\eta}$).$\underline{\eta}$ be the shape graph given by:

Vertices: {((♣, i), j) | ♣ ∈ {$\bullet_1$, ♦, $\bullet_2$}, i ∈ {0, 1}, j ∈ {0, 1, 2}}

Edges: {(((♣, i), j), ((♣′, i'), j)) | (♣, i) ↦ (♣′, i') an edge of 2.$\underline{\eta}$, j ∈ {0, 1, 2}} ∪
{(((♣, i), j), ((♣, i), j+1)) | (♣, i) a vertex of 2.$\underline{\eta}$, j, j+1 ∈ {0, 1, 2}}

and let (2.η).η be its path category. Likewise let 2+1 = ∗, and let (2.$\underline{\eta}$).$\underline{\eta^*}$ be the shape graph given by:

Vertices: $\{((\clubsuit, i), j) \mid \clubsuit \in \{\bullet_1, \blacklozenge, \bullet_2\}, i \in \{0, 1\}, j \in \{0, 1, 2, *\}\}$

Edges: $\{(((\clubsuit, i), j), ((\clubsuit', i'), j)) \mid (\clubsuit, i) \mapsto (\clubsuit', i')$ an edge of 2.$\underline{\eta}$,
$j \in \{0, 1, 2, *\}\} \cup$
$\{(((\clubsuit, i), j), ((\clubsuit, i), j{+}1)) \mid (\clubsuit, i)$ a vertex of 2.$\underline{\eta}$,
$j, j{+}1 \in \{0, 1, 2, *\}\}$

with path category (2.η).η*. Let (2.$\underline{\eta}$).$\underline{\eta}{\downarrow}\underline{\eta}$ be the shape graph given by:

Vertices: Vert((2.$\underline{\eta}$).$\underline{\eta}$) ∪ Vert($\underline{\eta}$)

Edges: Edg((2.$\underline{\eta}$).$\underline{\eta}$) ∪ Edg($\underline{\eta}$) ∪
$\{((\clubsuit, i), j) \mapsto \clubsuit' \mid (j, \clubsuit') \in \{(0, \bullet_1), (1, \blacklozenge), (2, \bullet_2)\},$
$\clubsuit \in \{\bullet_1, \blacklozenge, \bullet_2\}, i \in \{0, 1\}\}$

with path category (2.η).η↓η. Finally let (2.$\underline{\eta}$).$\underline{\eta^*}{\downarrow}\underline{\eta}$ be the shape graph given by:

Vertices: Vert((2.$\underline{\eta}$).$\underline{\eta^*}$) ∪ Vert($\underline{\eta}$)

Edges: Edg((2.$\underline{\eta}$).$\underline{\eta}{\downarrow}\underline{\eta}$) ∪ $\{((\clubsuit, i), *) \mapsto \bullet_2 \mid \clubsuit \in \{\bullet_1, \blacklozenge, \bullet_2\}, i \in \{0, 1\}\}$

with path category (2.η).η*↓η. Fig. 13 shows (the essentials of) an abstract diagram of shape (2.η).η*↓η. We will demand that the analogues of the properties of Definition 6.1 hold for such abstract diagrams, namely that:

(1) If any square in any concrete diagram of the abstract diagram has an arrow over $\blacklozenge \mapsto \bullet_1$, then it is a pullback.

(2) Any arrow over $((\clubsuit, i), 1) \mapsto ((\clubsuit, i), 2)$ for some $\clubsuit$, i, in any concrete diagram of the abstract diagram, is an isomorphism.

We introduce the following notation for the abstract diagram in Fig. 13:

$(((([X_0] \leftarrow [Y_0] \rightarrow [Z_0])$ -$[x_0,y_0,z_0]$-› $([X_0\tilde{}] \leftarrow [Y_0\tilde{}] \rightarrow [Z_0\tilde{}])) \rightarrow [A])$ -$[ABC]$=›
$(((([X_2] \leftarrow [Y_2] \rightarrow [Z_2])$ -$[x_2,y_2,z_2]$-› $([X_2\tilde{}] \leftarrow [Y_2\tilde{}] \rightarrow [Z_2\tilde{}])) \rightarrow [C])$ -›
$(((([X_2'] \leftarrow [Y_2'] \rightarrow [Z_2'])$ -$[x_2',y_2',z_2']$-› $([X_2'\tilde{}] \leftarrow [Y_2'\tilde{}] \rightarrow [Z_2'\tilde{}])) \rightarrow [C])$

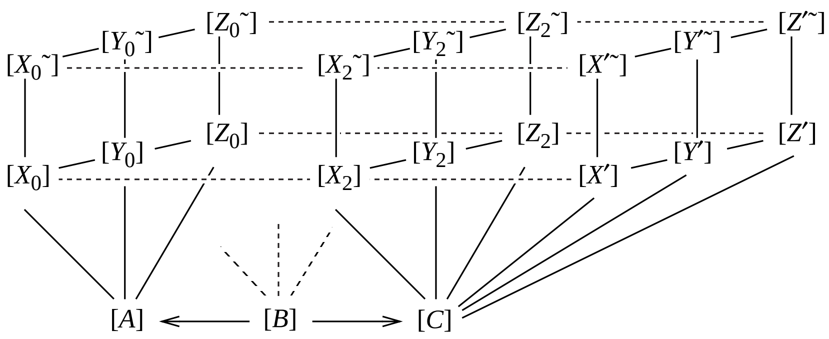

Fig. 13

where for brevity we may omit the middle row aside from "-›". For an abstract diagram of shape (2.η).η↓η we omit "-›" and the last row. Note that mention of the 2.η-shaped subdiagram projecting to $[B]$ is merely suppressed for brevity.

Definition 6.9 A triple interface-diagram category is a triple category which is an interface-diagram category with respect to horizontal, vertical and perpendicular composition. A triple interface-diagram category is concrete or abstract according to whether the diagrams it is built out of are concrete or abstract.

Now we introduce a triple category that will play a key role in the rest of the paper. The triple category adds a perpendicular dimension (i.e. the change of base via [*Gr-Sp*]) to the double category [*D-Gr-Sp*].

Definition 6.10 The triple category [*D-Gr-Sp*↓*Gr-Sp*] has as triple cells abstract diagrams of shape (2.η).η↓η. These may be combined using vertical, horizontal and perpendicular composition, $*_{\mathrm{v}}$, $*_{\mathrm{h}}$, $*_{\mathrm{p}}$ respectively. Vertical composition yields:

$\{((([X_0] \leftarrow [Y_0] \rightarrow [Z_0])$ -$[x_0,y_0,z_0]$-› $([X_0^\sim] \leftarrow [Y_0^\sim] \rightarrow [Z_0^\sim])) \rightarrow [A])$ -$[ABC]$=›
$((([X_2] \leftarrow [Y_2] \rightarrow [Z_2])$ -$[x_2,y_2,z_2]$-› $([X_2^\sim] \leftarrow [Y_2^\sim] \rightarrow [Z_2^\sim])) \rightarrow [C])\}$ $*_{\mathrm{v}}$
$\{((([X_0^\sim] \leftarrow [Y_0^\sim] \rightarrow [Z_0^\sim])$ -$[x_0^\sim,y_0^\sim,z_0^\sim]$-› $([X_0^{\sim\sim}] \leftarrow [Y_0^{\sim\sim}] \rightarrow [Z_0^{\sim\sim}])) \rightarrow [A])$ -$[ABC]$=›
$((([X_2^\sim] \leftarrow [Y_2^\sim] \rightarrow [Z_2^\sim])$ -$[x_2^\sim,y_2^\sim,z_2^\sim]$-› $([X_2^{\sim\sim}] \leftarrow [Y_2^{\sim\sim}] \rightarrow [Z_2^{\sim\sim}])) \rightarrow [C])\}$ =
$\{((([X_0] \leftarrow [Y_0] \rightarrow [Z_0])$ -$[x_0^\sim.x_0,y_0^\sim.y_0,z_0^\sim.z_0]$-› $([X_0^{\sim\sim}] \leftarrow [Y_0^{\sim\sim}] \rightarrow [Z_0^{\sim\sim}])) \rightarrow [A])$
-$[ABC]$=›
$((([X_2] \leftarrow [Y_2] \rightarrow [Z_2])$ -$[x_2^\sim.x_2,y_2^\sim.y_2,z_2^\sim.z_2]$-› $([X_2^{\sim\sim}] \leftarrow [Y_2^{\sim\sim}] \rightarrow [Z_2^{\sim\sim}])) \rightarrow [C])\}$

Horizontal composition yields:

$\{((([V_0] \leftarrow [W_0] \rightarrow [X_0])$ -$[v_0,w_0,x_0]$-› $([V_0^\sim] \leftarrow [W_0^\sim] \rightarrow [X_0^\sim])) \rightarrow [A])$ -$[ABC]$=›
$((([V_2] \leftarrow [W_2] \rightarrow [X_2])$ -$[v_2,w_2,x_2]$-› $([V_2^\sim] \leftarrow [W_2^\sim] \rightarrow [X_2^\sim])) \rightarrow [C])\}$ $*_{\mathrm{h}}$
$\{((([X_0] \leftarrow [Y_0] \rightarrow [Z_0])$ -$[x_0,y_0,z_0]$-› $([X_0^\sim] \leftarrow [Y_0^\sim] \rightarrow [Z_0^\sim])) \rightarrow [A])$ -$[ABC]$=›
$((([X_2] \leftarrow [Y_2] \rightarrow [Z_2])$ -$[x_2,y_2,z_2]$-› $([X_2^\sim] \leftarrow [Y_2^\sim] \rightarrow [Z_2^\sim])) \rightarrow [C])\}$ =
$\{((([V_0] \leftarrow [U_0] \rightarrow [Z_0])$ -$[v_0,u_0,z_0]$-› $([V_0^\sim] \leftarrow [U_0^\sim] \rightarrow [Z_0^\sim])) \rightarrow [A])$ -$[ABC]$=›
$((([V_2] \leftarrow [U_2] \rightarrow [Z_2])$ -$[v_2,u_2,z_2]$-› $([V_2^\sim] \leftarrow [U_2^\sim] \rightarrow [Z_2^\sim])) \rightarrow [C])\}$

where U_0 is a pullback of $W_0 \rightarrow X_0 \leftarrow Y_0$ etc. Perpendicular composition yields:

$\{((([X_0] \leftarrow [Y_0] \rightarrow [Z_0])$ -$[x_0,y_0,z_0]$-› $([X_0^\sim] \leftarrow [Y_0^\sim] \rightarrow [Z_0^\sim])) \rightarrow [A])$ -$[ABC]$=›
$((([X_2] \leftarrow [Y_2] \rightarrow [Z_2])$ -$[x_2,y_2,z_2]$-› $([X_2^\sim] \leftarrow [Y_2^\sim] \rightarrow [Z_2^\sim])) \rightarrow [C])\}$ $*_{\mathrm{p}}$
$\{((([X_2] \leftarrow [Y_2] \rightarrow [Z_2])$ -$[x_2,y_2,z_2]$-› $([X_2^\sim] \leftarrow [Y_2^\sim] \rightarrow [Z_2^\sim])) \rightarrow [C])$ -$[CDE]$=›
$((([X_4] \leftarrow [Y_4] \rightarrow [Z_4])$ -$[x_4,y_4,z_4]$-› $([X_4^\sim] \leftarrow [Y_4^\sim] \rightarrow [Z_4^\sim])) \rightarrow [E])\}$ =
$\{((([X_0] \leftarrow [Y_0] \rightarrow [Z_0])$ -$[x_0,y_0,z_0]$-› $([X_0^\sim] \leftarrow [Y_0^\sim] \rightarrow [Z_0^\sim])) \rightarrow [A])$ -$[AME]$=›
$((([X_4] \leftarrow [Y_4] \rightarrow [Z_4])$ -$[x_4,y_4,z_4]$-› $([X_4^\sim] \leftarrow [Y_4^\sim] \rightarrow [Z_4^\sim])) \rightarrow [E])\}$

where M is a pullback of $B \rightarrow C \leftarrow D$.

(Horizontal-vertical) double cells are double cells of [*D-Gr-Sp*] over an abstract graph, eg. $((([X_0] \leftarrow [Y_0] \rightarrow [Z_0])$ -$[x_0,y_0,z_0]$-› $([X_0^\sim] \leftarrow [Y_0^\sim] \rightarrow [Z_0^\sim])) \rightarrow [A])$; we do not describe the other two kinds of double cell here. Vertical arrows are abstract graph morphisms over an abstract graph, eg. $([X] \rightarrow [X^\sim]) : ([X] \rightarrow [A]) \rightarrow ([X^\sim] \rightarrow [A])$; horizontal arrows are abstract spans over an abstract graph, eg. $([X] \leftarrow [Y] \rightarrow [Z]) : ([X] \rightarrow [A]) \rightarrow ([Z] \rightarrow [A])$; perpendicular arrows are essentially arrows of [*Gr*↓*Gr-Sp*], i.e. abstract changes of base of abstract graphs over an abstract graph, eg. $(([X] \leftarrow [Y] = [Z])$

-$[a,b,c]$-› $([A] \leftarrow [B] \rightarrow [C])) : ([X] \rightarrow [A]) \rightarrow ([Z] \rightarrow [C])$. And by now it is clear that the objects are just abstract graphs over an abstract graph, i.e. abstract graph morphisms, eg. $([X] \rightarrow [A])$.

Proposition 6.11 [*D-Gr-Sp*↓*Gr-Sp*] is a triple interface-diagram category.

Proof. This is tedious if straightforward to show. There are three sets of identity and associativity laws, and their degenerate cases. Furthermore there are 12 terms involving different ways of assembling 8 smaller triple cells into a single large triple cell such that every distinct pair yields an interchange law (the collection of which we do not list). ☺

Definition 6.12 The triple category [*D-Gr-Sp*∗↓*Gr-Sp*] has as triple cells abstract diagrams of shape (2.η).η∗↓η. As for [*D-Gr-Sp*↓*Gr-Sp*], these may be combined using vertical, horizontal and perpendicular composition, $*_v$, $*_h$, $*_p$ respectively.

We do not go into details. These are essentially given by replacing -[…]=› by -[…]=›-› everywhere in the above.

Proposition 6.11 [*D-Gr-Sp*∗↓*Gr-Sp*] is a triple interface-diagram category.

The main results of this section are the following two theorems. These are to be understood as asserting a unique factorising triple cell (i.e. double cell morphism) θ in Fig. 4 where e_0 and e_{01} in the figure are themselves triple cells that project to b_0 and $b_1 \circ b_0$ respectively.

Theorem 6.12 The projection [$P_{D\text{-}Gr\text{-}Sp}$] : [*D-Gr-Sp*↓*Gr-Sp*] → [*Gr-Sp*] that takes

$((([X_0] \leftarrow [Y_0] \rightarrow [Z_0])$ -$[x_0,y_0,z_0]$-› $([X_0\tilde{}] \leftarrow [Y_0\tilde{}] \rightarrow [Z_0\tilde{}])) \rightarrow [A])$ -$[ABC]$=›
$((([X_2] \leftarrow [Y_2] \rightarrow [Z_2])$ -$[x_2,y_2,z_2]$-› $([X_2\tilde{}] \leftarrow [Y_2\tilde{}] \rightarrow [Z_2\tilde{}])) \rightarrow [C])$

to $([A] \leftarrow [B] \rightarrow [C]) : [A] \rightarrow [C]$ is a split opfibration, where all triple cells of [*D-Gr-Sp*↓*Gr-Sp*] are opcartesian and belong to the splitting.

Theorem 6.13 The projection [$P_{D\text{-}Gr\text{-}Sp*}$] : [*D-Gr-Sp*∗↓*Gr-Sp*] → [*Gr-Sp*] that takes

$((([X_0] \leftarrow [Y_0] \rightarrow [Z_0])$ -$[x_0,y_0,z_0]$-› $([X_0\tilde{}] \leftarrow [Y_0\tilde{}] \rightarrow [Z_0\tilde{}])) \rightarrow [A])$ -$[ABC]$=›-›
$((([X_2'] \leftarrow [Y_2'] \rightarrow [Z_2'])$ -$[x_2',y_2',z_2']$-› $([X_2'\tilde{}] \leftarrow [Y_2'\tilde{}] \rightarrow [Z_2'\tilde{}])) \rightarrow [C])$

to $([A] \leftarrow [B] \rightarrow [C]) : [A] \rightarrow [C]$ is a split opfibration, where all triple cells of [*D-Gr-Sp*∗↓*Gr-Sp*] with all arrows over $((\clubsuit, i), 2) \mapsto ((\clubsuit, i), *)$ abstract isomorphisms are opcartesian and belong to the splitting.

The proofs of these results simply adapt Theorems 6.7 and 6.8.

Remark 6.14 Note that we can think of [*D-Gr-Sp*↓*Gr-Sp*] as being presented in two ways, namely:

(1) As given, i.e as a collection of triple cells of a certain shape possessing certain properties.

(2) As a collection of triple cells given by choosing a double cell, a base span, and opcartesian arrows over the span for all objects in the double cell.

Sections 4.5 and 4.6 convince us that these amount to the same thing. However for [*D-Gr-Sp*∗↓*Gr-Sp*] we only have option (1). Given a choice of opcartesian arrows *extended by in-fibre morphisms* for each object in a double cell does not allow us to conclude that they generate a double cell morphism.

7 Abstract Graph Rewriting

At this point it behoves us to reward the patient reader with some insight as to where the preceding lengthy technical deliberations are leading us. Consider Fig. 13. It shows an abstract double square typed over an abstract graph $[A]$ which is then transported through a change of type, expressed by the abstract span $[A] \leftarrow [B] \rightarrow [C]$. The double square is intended to represent the result of an abstract graph transformation step such as an abstract typed version of Fig. 2. Only a few details prevent us from declaring this correspondence immediately.

We recall first that the productions used in graph transformation have both arrows monic. Secondly we remember that the double square ought to be two pushouts or two pullbacks. We need some lemmas.

Lemma 7.1 In Fig. 13, in the square $X_0^\sim X_0 Y_0 Y_0^\sim$ suppose $X_0 \leftarrow Y_0$, $X_0^\sim \leftarrow Y_0^\sim$ are monic. Then in $X_1^\sim X_1 Y_1 Y_1^\sim$ (not illustrated) $X_1 \leftarrow Y_1$, $X_1^\sim \leftarrow Y_1^\sim$ are monic.

Proof. Consider the cube $X_0^\sim X_0 Y_0 Y_0^\sim X_1^\sim X_1 Y_1 Y_1^\sim$. The perpendicular arrows $X_0 \leftarrow X_1$, $X_0^\sim \leftarrow X_1^\sim$, $Y_0 \leftarrow Y_1$, $Y_0^\sim \leftarrow Y_1^\sim$, (not illustrated) are constructed via the opfibration of Theorem 6.7, yielding unique arrows $X_1^\sim \leftarrow X_1$, $X_1 \leftarrow Y_1$, $Y_1^\sim \leftarrow Y_1$, $X_1^\sim \leftarrow Y_1^\sim$, such that the four squares of the cube over $A \leftarrow B$ are all easily shown to be pullbacks. Now the monicity of $X_1 \leftarrow Y_1$, $X_1^\sim \leftarrow Y_1^\sim$ is routine. ☺

Lemma 7.2 In Fig. 13, let $X_0^\sim X_0 Y_0 Y_0^\sim$ be a pullback. Then $X_1^\sim X_1 Y_1 Y_1^\sim$ is a pullback.

Proof. By remarks in the preceding proof the cube $X_0^\sim X_0 Y_0 Y_0^\sim X_1^\sim X_1 Y_1 Y_1^\sim$ commutes. Bearing in mind that all arrows in the cube are oriented towards $X_0^\sim$, we just have to show that for any $Q \rightarrow X_1$, $Q \rightarrow Y_1^\sim$ that close $X_1 \rightarrow X_1^\sim$ and $Y_1^\sim \rightarrow X_1^\sim$, there is a unique $Q \rightarrow Y_1$ that factors $Q \rightarrow X_1$, $Q \rightarrow Y_1^\sim$. But this is an easy exercise in pullback properties. ☺

Recalling now that in *Gr*, a commuting square with two monic parallel arrows is a pushout iff it is a pullback and the two arrows with the same codomain are jointly surjective, we have the following.

Lemma 7.3 In Fig. 13, let $X_0^\sim X_0 Y_0 Y_0^\sim$ be a pushout with $X_0 \leftarrow Y_0$, $X_0^\sim \leftarrow Y_0^\sim$ monic. Then $X_1^\sim X_1 Y_1 Y_1^\sim$ is a pushout with $X_1 \leftarrow Y_1$, $X_1^\sim \leftarrow Y_1^\sim$ monic.

Proof. By the preceding two lemmas we quickly deduce that $X_1^\sim X_1 Y_1 Y_1^\sim$ is a pullback with $X_1 \leftarrow Y_1$, $X_1^\sim \leftarrow Y_1^\sim$ monic. We just need to check joint surjectivity of $X_1 \rightarrow X_1^\sim \leftarrow Y_1^\sim$. Thus for a contradiction suppose there is an item (vertex or edge), $q_1^\sim$ of $X_1^\sim$, not in the ranges of $X_1 \rightarrow X_1^\sim \leftarrow Y_1^\sim$. It must map by $X_0^\sim \leftarrow X_1^\sim$ to an item $q_0^\sim$ in the ranges of $X_0 \rightarrow X_0^\sim \leftarrow Y_0^\sim$ which are jointly surjective; let us say it is in $\mathrm{rng}(X_0 \rightarrow X_0^\sim)$. So there is an item q_0 in X_0 which maps under $X_0 \rightarrow X_0^\sim$ to the same $q_0^\sim$ as item $q_1^\sim$ of $X_1^\sim$ maps to under $X_0^\sim \leftarrow X_1^\sim$. But this means that there must be at least one item q_1 in X_1 which maps to q_0 under $X_0 \leftarrow X_1$ and to $q_1^\sim$ under $X_1 \rightarrow X_1^\sim$, contradicting our supposition, otherwise $X_0^\sim X_0 X_1 X_1^\sim$ would not be a pullback. We are done. ☺

With these results to hand we can fine-tune the opfibrations constructed earlier so that they indeed act as required, secure in the knowledge that the opcartesian triple cells preserve the requisite additional properties.

Thus we have the triple category [*D-Gr-MSp*↓*Gr-Sp*], in which the double cell domain and codomain of a triple cell (over a type change morphism) are morphisms of monic spans, and its associated category [*D-Gr-MSp**↓*Gr-Sp*].

Furthermore, within [*D-Gr-MSp*↓*Gr-Sp*] we can specialise to the triple subcategory [*D-Gr-MSp*-DPB↓*Gr-Sp*], in which the double cell domain and codomain of a triple cell are morphisms of monic spans which are furthermore pairs of pullbacks. This has the associated category [*D-Gr-MSp*-DPB*↓*Gr-Sp*].

And going even further, we can identify within [*D-Gr-MSp*-DPB↓*Gr-Sp*] the triple subcategory [*D-Gr-MSp*-DPO↓*Gr-Sp*] in which the double cell domain and codomain of a triple cell are morphisms of monic spans which are pairs of pushouts, this having the associated category [*D-Gr-MSp*-DPO*↓*Gr-Sp*].

To avoid repetition, we will deal with both the pullback and pushout situations using the notations [*D-Gr-MSp*-DP#↓*Gr-Sp*] and [*D-Gr-MSp*-DP#*↓*Gr-Sp*].

Note that one convenient byproduct of phrasing derivation steps in terms of the triple categories [*D-Gr-MSp*-DP#*↓*Gr-Sp*] is that, from the interchange laws for triple categories, there immediately follow a vast number of commutativity properties relating horizontal composition, vertical composition, and change of type for derivation steps. The orthogonality of these three aspects of graph transformation in the presence of injective productions would be hard to present otherwise without compiling an exhaustive list. Furthermore we have as expected the opfibration properties.

Theorem 7.4 The projection $[P_{D\text{-}Gr\text{-}MSp\text{-}DP\#}] : [D\text{-}Gr\text{-}MSp\text{-}DP\#{\downarrow}Gr\text{-}Sp] \rightarrow [Gr\text{-}Sp]$ is a split opfibration, where all arrows of [*D-Gr-MSp*-DP#↓*Gr-Sp*] are opcartesian and belong to the splitting.

Theorem 7.5 The projection $[P_{D\text{-}Gr\text{-}MSp\text{-}DP\#*}] : [D\text{-}Gr\text{-}MSp\text{-}DP\#*{\downarrow}Gr\text{-}Sp] \rightarrow [Gr\text{-}Sp]$ is an opfibration, where all arrows of [*D-Gr-MSp*-DP#*↓*Gr-Sp*] with with all arrows over $((\clubsuit, i), 2) \mapsto ((\clubsuit, i), *)$ abstract isomorphisms are opcartesian and belong to the splitting.

With all this in hand, we can present our formal theory of type change in graph transformation.

8 The Category of Typed Graph Grammars

For the purposes of typed graph rewriting theory, we consider abstract graphs $[G]$ typed over an abstract type graph $[TG]$, or putting it another way abstract graph morphisms $[G] \rightarrow [TG]$. Changing the base type is done by means of an arbitrary abstract span eg. $([TG_0] \leftarrow [TG_1] \rightarrow [TG_2]) : [TG_0] \rightarrow [TG_2]$. Therefore the results of the previous sections are applicable, and show us how the various entities and activities involved in graph transformation, transform under such a change of typing at an abstract level.

Definition 8.1 The category of abstract typed graph grammars [*GraGra*] has the following constituents.

Objects: $([TG], [\overline{G}], P, \pi)$ where:
$[TG]$ is an abstract type graph,
$[\overline{G}]$ is an abstract start graph typed over $[TG]$,

i.e. an abstract graph morphism $[\overline{G}] \to [TG]$,
P is a set of production names,
$\pi : P \to HArr([\mathsf{D\text{-}Gr\text{-}MSp{\downarrow}Gr\text{-}Sp}])$ is a map from P to horizontal arrows of $[\mathsf{D\text{-}Gr\text{-}MSp{\downarrow}Gr\text{-}Sp}]$,
i.e. abstract typed monic spans, typed over $[TG]$.

Arrows: $([TG_0] \leftarrow [TG_1] \to [TG_2], f)$:
$([TG_0], [\overline{G}_0], P_0, \pi_0) \to ([TG_2], [\overline{G}_2], P_2, \pi_2)$
which is shorthand for a collection of arrows.
Firstly: an arrow of $[\mathsf{Gr\text{-}Sp}]$, $([TG_0] \leftarrow [TG_1] \to [TG_2]) : [TG_0] \to [TG_2]$,
Secondly: an arrow of $[\mathsf{Gr{\downarrow}Gr\text{-}Sp}]$,
$(([\overline{G}_0] \leftarrow [\overline{G}_2] = [\overline{G}_2])$ -$[g_0,g_1,g_2]$-› $([TG_0] \leftarrow [TG_1] \to [TG_2]))$:
$([\overline{G}_0] \to [TG_0]) \to ([\overline{G}_2] \to [TG_2])$,
which projects under $[\mathsf{P}]$ to the first arrow,
Thirdly: an arrow of Set, $f : P_0 \to P_2$ i.e. a map,
Fourthly: for all $p \in P_0$ a horizontal-perpendicular double cell of $[\mathsf{D\text{-}Gr\text{-}MSp{\downarrow}Gr\text{-}Sp}]$,
$([\pi_0(p)] \to [TG_0])$ -$[TG_0TG_1TG_2]$=› $([\pi_2(f(p))] \to [TG_2])$
which projects under $[\mathsf{P_{D\text{-}Gr\text{-}Sp}}]$ to the first arrow.

Composition: $([TG_2] \leftarrow [TG_3] \to [TG_4], g) \circ ([TG_0] \leftarrow [TG_1] \to [TG_2], f) = ([TG_0] \leftarrow [TG_{2'}] \to [TG_4], g \circ f)$ where $[TG_{2'}]$ arises from the composition of $[\mathsf{Gr\text{-}Sp}]$.

Identities: $([TG] \leftarrow [TG] \to [TG], \mathrm{id}_P) : ([TG], [\overline{G}], P, \pi) \to ([TG], [\overline{G}], P, \pi)$,
where the arrows in $[TG] \leftarrow [TG] \to [TG]$ are all isomorphisms.

Theorem 8.2 The projection $[\mathsf{P_{GraGra}}] : [\mathsf{GraGra}] \to [\mathsf{Gr\text{-}Sp}]$ such that

$$[\mathsf{P_{GraGra}}](([TG], [\overline{G}], P, \pi)) = [TG] \quad \text{and}$$
$$[\mathsf{P_{GraGra}}](([TG_0] \leftarrow [TG_1] \to [TG_2], f)) = ([TG_0] \leftarrow [TG_1] \to [TG_2])$$

is an opfibration, in which the arrows $([TG_0] \leftarrow [TG_1] \to [TG_2], f) : ([TG_0], [\overline{G}_0], P_0, \pi_0) \to ([TG_2], [\overline{G}_2], P_2, \pi_2)$ such that f is an iso in Set are opcartesian.

Proof. Let $([TG_0] \leftarrow [TG_1] \to [TG_2], iso) : GG_0 \to GG_2$ be a putative opcartesian arrow, and suppose we have an arrow $([TG_0] \leftarrow [TG_{2'}] \to [TG_4], g) : GG_0 \to GG_4$ whose projection is an extension of $([TG_0] \leftarrow [TG_1] \to [TG_2])$ by $([TG_2] \leftarrow [TG_3] \to [TG_4])$. We have to find a unique $([TG_2] \leftarrow [TG_3] \to [TG_4], g') : GG_2 \to GG_4$ such that $GG_0 \to GG_2 \to GG_4$ factors $GG_0 \to GG_4$.

Luckily all of the hard work has already been done in Section 6. Firstly, the component $([TG_2] \leftarrow [TG_3] \to [TG_4]) : [TG_2] \to [TG_4]$ is immediate. Secondly, the component $(([\overline{G}_2] \leftarrow [\overline{G}_4] = [\overline{G}_4])$ -$[g_2,g_3,g_4]$-› $([TG_2] \leftarrow [TG_3] \to [TG_4])) : ([\overline{G}_2] \to [TG_2]) \to ([\overline{G}_4] \to [TG_4])$ arises uniquely since $[\mathsf{P}_\varnothing] : [\mathsf{Gr{\downarrow}Gr\text{-}Sp}] \to [\mathsf{Gr\text{-}Sp}]$ is an opfibration and its opcartesian arrows agree with the start graph component of opcartesian arrows of $[\mathsf{GraGra}]$. Thirdly, the fact that $iso : P_0 \to P_2$ is an isomorphism means that the requirement $g' \circ iso = g$ fixes g' uniquely. Fourthly, for each $p \in P_2$, we use the restriction of Theorem 6.12 to horizontal arrows to find a unique $(([\pi_2(p)] \to [TG_2])$ -$[TG_2TG_3TG_4]$=› $([\pi_4(g'(p))] \to [TG_4])) : ([\pi_2(p)] \to [TG_2]) \to ([\pi_4(g'(p))] \to [TG_4])$. ☺

Note that this opfibration is not split due to the absence of any canonical isomorphisms in *Set*, but only for that reason. A choice of standard isomorphisms for *Set*, as with the corresponding construction for *Gr*, would enable a splitting to be constructed.

As for constructions in preceding sections there is another category of graph grammars [*GraGra*∗], which in its arrows, uses the morphisms of [*Gr*∗↓*Gr-Sp*] and [*D-Gr-MSp*∗↓*Gr-Sp*] in its second and fourth components respectively. The details are so similar to those for [*GraGra*] that we do not quote them in full. We merely set out the opfibration theorem for completeness.

Theorem 8.3 The projection $[P_{GraGra*}]$: [*GraGra*∗] $\rightarrow$ [*Gr-Sp*] such that

$$[P_{GraGra*}](([TG], [\overline{G}], P, \pi)) = [TG] \text{ and}$$
$$[P_{GraGra*}](([TG_0] \leftarrow [TG_1] \rightarrow [TG_2], f)) = ([TG_0] \leftarrow [TG_1] \rightarrow [TG_2])$$

is an opfibration, in which the arrows $([TG_0] \leftarrow [TG_1] \rightarrow [TG_2], f) : ([TG_0], [\overline{G}_0], P_0, \pi_0) \rightarrow ([TG_2], [\overline{G}_2], P_2, \pi_2)$ such that f is an iso in *Set* and all arrows over $((\clubsuit, i), 2) \mapsto ((\clubsuit, i), *)$ abstract isomorphisms, are opcartesian.

9 The Category of Transition Systems

In the sequel the abstract start graph $[\overline{G}]$ of a graph grammar will just be a passenger; we quietly carry it around to save having to have a fresh bunch of definitions for everything. Also we rely purely on the fact that morphisms of monic abstract spans compose well so that we use the symbol # to denote that we are dealing simultaneously with DPB and DPO rewriting.

Graph transition systems are enriched graph grammars which include all the result spans of direct derivation steps by their productions and such that the set of production names supports a partial action / by *HVDCell*([*D-Gr-MSp*-DP#∗↓*Gr-Sp*]), the (horizontal-vertical) double cells of [*D-Gr-MSp*-DP#∗↓*Gr-Sp*]. For notational compactness, we will write these double cells in future using a notation like $[d_1,d_2,d_3]$, refering to their alternative interpretation as abstract span morphisms, this in turn legitimising the use of dom and cod in the next definition.

Definition 9.1 An abstract typed graph transition system is a quintuple $([TG], [\overline{G}], P, \pi, /)$ where $([TG], [\overline{G}], P, \pi)$ is an abstract typed graph grammar, and $/ : P \times$ *HVD-Cell*([*D-Gr-MSp*-DP#∗↓*Gr-Sp*]) $\rightarrow P$ satisfies:

(1) If $\mathrm{dom}([d_1,d_2,d_3]) = [\pi(p)]$ then $p/[d_1,d_2,d_3]$ is defined and $[\pi(p/[d_1,d_2,d_3])] = \mathrm{cod}([d_1,d_2,d_3])$,

(2) $p/[\mathrm{id}_{[\pi(p)]}] = p$,

(3) $(p/[d_1,d_2,d_3])/[d'_1,d'_2,d'_3] = p/[d'_1.d_1,d'_2.d_2,d'_3.d_3]$.

Definition 9.2 The category [*GraTS*#] of abstract graph transition systems has as objects abstract graph transition systems, and as morphisms $([TG_0] \leftarrow [TG_1] \rightarrow [TG_2], f) : ([TG_0], [\overline{G}_0], P_0, \pi_0, /_0) \rightarrow ([TG_2], [\overline{G}_2], P_2, \pi_2, /_2)$, where $([TG_0] \leftarrow [TG_1] \rightarrow [TG_2], f)$ is a morphism of the underlying abstract graph grammar, and such that for each p_0 in P_0 and each $[d_{01},d_{02},d_{03}]$ with $p_0/[d_{01},d_{02},d_{03}]$ defined, we have a $[d_{21},d_{22},d_{23}]$ with $[f(p_0)/[d_{21},d_{22},d_{23}]]$ defined and $[f(p_0/[d_{01},d_{02},d_{03}])] = [f(p_0)/[d_{21},d_{22},d_{23}]]$.

Obviously there is a forgetful functor $[U\#] : [GraTS\#] \to [GraGra]$ which just ignores $/$. We now give the construction that will provide a left adjoint functor to $[U\#]$.

Definition 9.3 Let $GG = ([TG], [\overline{G}], P, \pi)$ be an abstract graph grammar. Then the abstract graph transition system $[GTS\#] = ([TG], [\overline{G}], PP, \pi\pi, /)$ is given by:

(1) $PP = \{(p, [t_1,t_2,t_3]) \mid p \in P$, and $[t_1,t_2,t_3]$ is a horizontal-vertical double cell of $[D\text{-}Gr\text{-}MSp\text{-}\mathrm{DP\#}{\downarrow}Gr\text{-}Sp]$ with $\mathrm{dom}([t_1,t_2,t_3]) = [\pi(p)]\}$,

(2) $[\pi\pi((p, [t_1,t_2,t_3]))] = \mathrm{cod}([t_1,t_2,t_3])$,

(3) If $[\pi\pi((p, [t_1,t_2,t_3]))] = \mathrm{dom}([d_1,d_2,d_3])$ then $(p, [t_1,t_2,t_3])/[d_1,d_2,d_3]$ is defined, equals $(p, [d_1.t_1,d_2.t_2,d_3.t_3])$, and thus $[\pi\pi((p, [t_1,t_2,t_3])/[d_1,d_2,d_3])] = \mathrm{cod}([d_1.t_1, d_2.t_2,d_3.t_3])$.

It is obvious that $[GTS\#]$ is an abstract graph transition system.

Theorem 9.4 The forgetful functor $[U\#] : [GraTS\#] \to [GraGra]$ has a left adjoint $[TS\#] : [GraGra] \to [GraTS\#]$ where the functor $[TS\#](GG) = GTS\#$ is given by Definition 9.3 for objects, and is given for arrows by:

$[TS\#](([TG_0] \leftarrow [TG_1] \to [TG_2], f) : GG_0 \to GG_2)$ is the unique morphism
$([TG_0] \leftarrow [TG_1] \to [TG_2], g_P) : GTS\#_0 \to GTS\#_2$ in $[GraTS\#]$ such that
for all p in P_0, $g_P((p, [\mathrm{id}_{[\pi(p)]}])) = (f(p), [\mathrm{id}_{[\pi(f(p))]}])$

Proof. Define the universal arrow $u : GG \to [U\#]([TS\#](GG) = GTS\#)$ for an abstract graph grammar $GG = ([TG], [\overline{G}], P, \pi)$ by:

$$u : GG \to [U\#](GTS\#) = ([TG] \leftarrow [TG] \to [TG], in)$$

where $([TG] \leftarrow [TG] \to [TG])$ is an identity in $[Gr\text{-}Sp]$, and $in(p) = (p, [\mathrm{id}_{[\pi(p)]}])$. Given a morphism $f : GG \to [U\#](TT) = ([TG_0] \leftarrow [TG_1] \to [TG_2], f_P)$ into the forgetful image of some abstract transition system TT, we must show that there is a unique morphism $g : [TS\#](GG) \to TT$ such that $f = [U\#](g) \circ u$. See Fig. 14.

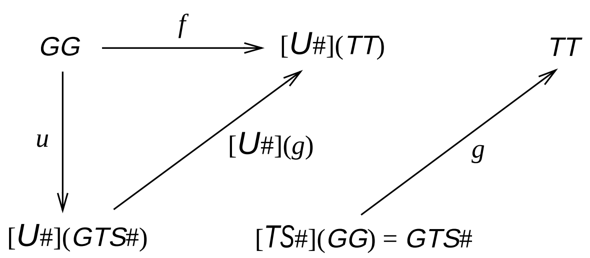

Fig. 14

Let g be the unique morphism $([TG_0] \leftarrow [TG_1] \to [TG_2], g_P) : GTS\# \to TT$ such that for all p in P, $g_P((p, [\mathrm{id}_{[\pi(p)]}])) = (f(p)/[\mathrm{id}_{[\pi(f(p))]}])$. That g exists can be seen as follows. Firstly, we can make the triangle in Fig. 14, restricting $[U\#](g)$ to the image $u(GG)$, commute. This we do just by setting $[U\#](g)((p, [\mathrm{id}_{[\pi(p)]}])) = f(p)$ for all p in P. Secondly, we extend this uniquely to all of $[U\#](GTS\#)$ as follows. For all typed abstract monic spans $\pi_{GTS\#}((p, [d_1,d_2,d_3]))$, we choose the unique typed abstract monic span $(([X] \leftarrow [Y] \rightarrow [Z]) \rightarrow [TG_2])$ over $[TG_2]$ arising from the split opfibration $[P_{D\text{-}Gr\text{-}Sp}]$, such that $(\pi_{GTS\#}((p, [d_1,d_2,d_3]))$ -$[TG_0TG_1TG_2]$=› $(([X] \leftarrow [Y] \rightarrow [Z]) \rightarrow [TG_2]))$ is a horizontal-perpendicular double cell, this being consistent with the choice already made. Now $[d_1,d_2,d_3] : \pi_{GTS\#}((p, [\mathrm{id}_{[\pi(p)]}])) \rightarrow \pi_{GTS\#}((p, [d_1,d_2,d_3]))$ is a typed abstract monic span morphism, therefore the split opfibration $[P_{D\text{-}Gr\text{-}Sp}]$ enables us to find a unique $[t_1,t_2,t_3] : \pi_{TT}(f(p)/[\mathrm{id}_{[\pi(f(p))]}]) \rightarrow (([X] \leftarrow [Y] \rightarrow [Z]) \rightarrow [TG_2])$, enabling us to simultaneously deduce that $(([X] \leftarrow [Y] \rightarrow [Z]) \rightarrow [TG_2])$ is a typed production of $[U\#](TT)$. This gives $[U\#](g) : [U\#](GTS*) \rightarrow [U\#](TT)$. We now set $g_P((p, [d_1,d_2,d_3])) = (f(p)/[\mathrm{id}_{[\pi(f(p))]}])/[t_1,t_2,t_3] = g_P((p, [\mathrm{id}_{[\pi(p)]}]))/[t_1,t_2,t_3]$. It is clear that all the required equations hold so that we have $g : [TS\#](GG) \rightarrow TT$. (So in the end we accomplish the whole thing in the reverse order to what one might expect.) ☺

The above construction of the unique g was rather drawn out — we could simply have invoked $[P_{D\text{-}Gr\text{-}Sp}]$ directly and left it at that — the slower more detailed presentation was for convenience later, in Section 11. Now Definition 9.2, which describes how the extra structure in an abstract graph transition system behaves under morphisms, makes the following expected statement evident.

Theorem 9.5 The obvious projection $[P_{GraTS\#}] : [GraTS\#] \rightarrow [Gr\text{-}Sp]$ is an opfibration, where all arrows $([TG_0] \leftarrow [TG_1] \rightarrow [TG_2], f) : ([TG_0], [\overline{G}_0], P_0, \pi_0, /_0) \rightarrow ([TG_2], [\overline{G}_2], P_2, \pi_2, /_2)$ such that f is an iso in Set are opcartesian.

At this point the reader will be expecting the usual * version of the preceding results, but he will be disappointed. Although there is an opfibration result which we quote shortly, the adjoint construction fails. The reason is the point alluded to at the end of Section 6, namely that whereas a diagram extended by a bunch of opcartesian arrows can generate a diagram morphism as in Section 4.5, extending by a bunch of opcartesian arrows themselves extended by in-fibre morphisms, in general will not do so. Viewing the structure of the productions in an abstract graph transition system as a "large diagram", shows that the construction of g in Theorem 9.4 would not go through in the more general case. This breakdown also explains why we have been careful to present separately results including and not including in-fibre aspects hitherto.

Now taking the definition of $[GraTS\#*]$, the category of abstract graph transition systems whose morphisms use the morphisms of $[Gr*{\downarrow}Gr\text{-}Sp]$ and $[D\text{-}Gr\text{-}MSp*{\downarrow}Gr\text{-}Sp]$, for granted, we have the following.

Theorem 9.6 The obvious projection $[P_{GraTS\#*}] : [GraTS\#*] \rightarrow [Gr\text{-}Sp]$ is an opfibration, where all arrows $([TG_0] \leftarrow [TG_1] \rightarrow [TG_2], f) : ([TG_0], [\overline{G}_0], P_0, \pi_0, /_0) \rightarrow ([TG_2], [\overline{G}_2], P_2, \pi_2, /_2)$ such that f is an iso in Set and all arrows over $((\clubsuit, i), 2) \mapsto ((\clubsuit, i), *)$ abstract isomorphisms, are opcartesian.

10 The Category of Derivation Systems

Now we can further enrich our transition systems, with an operation **;** of horizontal composition on production names, inherited from the corresponding property of $[D\text{-}Gr\text{-}MSp{\downarrow}Gr\text{-}Sp]$.

Definition 10.1 An abstract graph derivation system is a sextuple ($[TG]$, $[\overline{G}]$, P, π, $/$, $;$) where ($[TG]$, $[\overline{G}]$, P, π, $/$) is an abstract graph transition system, and $; : P \times P \nrightarrow P$ satisfies:

(1) If $\pi(p) = [A] \leftarrow [B] \rightarrow [C]$ and $\pi(q) = [C] \leftarrow [D] \rightarrow [E]$ then $p;q$ is defined and $\pi(p;q) = [A] \leftarrow [M] \rightarrow [E]$, where M is a pullback of $B \rightarrow C \leftarrow D$

(2) If $\pi(p) = [A] \leftarrow [B] \rightarrow [C]$ then P contains a $p_{[A]}$ with $\pi(p_{[A]}) = [A] \leftarrow [A] \rightarrow [A]$, an identity name such that $p_{[A]};p = p$, and also a similar identity name $p_{[C]}$ with $\pi(p_{[C]}) = [C] \leftarrow [C] \rightarrow [C]$ and $p;p_{[C]} = p$

(3) $;$ is associative

(4) If both $(p;q)/([s_1,s_2,s_3]*_\mathrm{h}[t_1,t_2,t_3])$ and $(p/[s_1,s_2,s_3]);(q/[t_1,t_2,t_3])$ are defined then $(p;q)/([s_1,s_2,s_3]*_\mathrm{h}[t_1,t_2,t_3]) = (p/[s_1,s_2,s_3]);(q/[t_1,t_2,t_3])$

Definition 10.2 The category [*GraDS*#] of abstract graph derivation systems has as objects abstract graph derivation systems, and as morphisms ($[TG_0] \leftarrow [TG_1] \rightarrow [TG_2]$, f) : ($[TG_0]$, $[\overline{G}_0]$, P_0, π_0, $/_0$, $;_0$) $\rightarrow$ ($[TG_2]$, $[\overline{G}_2]$, P_2, π_2, $/_2$, $;_2$), where ($[TG_0] \leftarrow [TG_1] \rightarrow [TG_2]$, f) is a morphism between the underlying abstract transition systems, such that for each identity name $p_{[A]}$ in P_0, $f(p_{[A]})$ is an identity name, and for each (p, q) pair defined for $;_0$, we have $f(p;_0q) = f(p);_2f(q)$.

As in Section 9 there is a forgetful functor [*V*#] : [*GraDS*#] $\rightarrow$ [*GraTS*#] which just ignores $;$. We now give the construction that will provide a left adjoint functor to [*V*#].

Definition 10.3 Let [*GTS*#] = ($[TG]$, $[\overline{G}]$, P, π, $/$) be an abstract graph transition system. Then the abstract graph transition system [*GDS*#] = ($[TG]$, $[\overline{G}]$, PP, $\pi\pi$, $/_{PP}$, $;_{PP}$) is given by firstly, constructing PPP, $\pi\pi\pi$, $/_{PPP}$ and $;_{PPP}$ as the smallest sets satisfying the following properties:

(1) (p) is in PPP for p in P, and $\pi\pi\pi((p)) = \pi(p)$; and whenever $p/[d_1,d_2,d_3]$ is defined, $(p)/_{PPP}[d_1,d_2,d_3] = (p/[d_1,d_2,d_3])$, (and $\pi\pi\pi((p)/_{PPP}[d_1,d_2,d_3]) = \mathrm{cod}([d_1,d_2,d_3])$)

(2) $(p_{[A]})$ and $(p_{[C]})$ are in PPP for each p in P with $\pi(p) = [A] \leftarrow [B] \rightarrow [C]$, and $\pi\pi\pi((p_{[A]})) = [A] \leftarrow [A] \rightarrow [A]$ and $\pi\pi\pi((p_{[C]})) = [C] \leftarrow [C] \rightarrow [C]$ both identity abstract spans, and $\pi\pi\pi((p_{[A]});_{PPP}(p)) = \pi\pi\pi((p)) = \pi\pi\pi((p);_{PPP}(p_{[C]}))$; and whenever $[A] = \mathrm{dom}([d])$, $(p_{[A]})/_{PPP}[d,d,d] = (p_{\mathrm{cod}(d)})$, (and $\pi\pi\pi((p_{\mathrm{cod}(d)})) = \mathrm{cod}([d,d,d])$), and similarly for $(p_{[C]})$

(3) (p, q) is in PPP for p, q in PPP such that $\pi\pi\pi(p) = [A] \leftarrow [B] \rightarrow [C]$ and $\pi\pi\pi(p) = [C] \leftarrow [D] \rightarrow [E]$; $(p, q) = p;_{PPP}q$, and $\pi\pi\pi((p, q))$ is given via the local pullback of $\pi\pi\pi(p)$ and $\pi\pi\pi(q)$; and whenever $p/_{PPP}[s_1,s_2,s_3]$, $q/_{PPP}[t_1,t_2,t_3]$ and $[s_1,s_2,s_3]*_\mathrm{h}[t_1,t_2,t_3]$ are defined, $(p, q)/_{PPP}[s_1,s_2,s_3]*_\mathrm{h}[t_1,t_2,t_3]$ is defined and $\pi\pi\pi((p, q)/_{PPP}[s_1,s_2,s_3]*_\mathrm{h}[t_1,t_2,t_3]) = \mathrm{cod}([s_1,s_2,s_3]*_\mathrm{h}[t_1,t_2,t_3])$

And then secondly, letting PP, $\pi\pi$, $/_{PP}$ and $;_{PP}$ be given by taking PPP, $\pi\pi\pi$, $/_{PPP}$ and $;_{PPP}$ modulo the composition law $(p/[d_1,d_2,d_3])/_{PPP}[d'_1,d'_2,d'_3] = p/_{PPP}[d'_1.d_1,d'_2.d_2,d'_3.d_3]$ and identity law $p/_{PPP}\mathrm{id}_{\pi(p)} = p$, and the associative law $((\mathrm{A};_{PPP}\mathrm{B});_{PPP}\mathrm{C}) = (\mathrm{A};_{PPP}(\mathrm{B};_{PPP}\mathrm{C}))$ and identity laws $(p_{[A]});_{PPP}(p) = (p) = (p);_{PPP}(p_{[C]})$.

It is clear that [*GDS*#] is an abstract graph derivation system.

Note that as our constructions are based on properties of [*D-Gr-MSp*↓*Gr-Sp*], the interchange laws of [*GraDS*#] contained within the span-transition lemma of Corradini et al. (1996b), derive directly from those of the subcategories [*D-Gr-MSp*-DP#↓ *Gr-Sp*].

Theorem 10.4 The forgetful functor [*V*#] : [*GraDS*#] → [*GraTS*#] has a left adjoint [*DS*#] : [*GraTS*#] → [*GraDS*#] where [*DS*#](*GTS*#) = *GDS*# is given by Definition 6.3 for objects, with the unique extension for arrows.

Proof. Define the universal arrow v : *GTS*# → [*V*#]([*DS*#](*GTS*#) = *GDS*#) for a concrete transition system *GTS*# = ([TG], [$\overline{G}$], P, π, /) by

$$v : DG \to [V\#](GDS\#) = ([TG] \leftarrow [TG] \to [TG], in)$$

where ([TG] ← [TG] → [TG]) is an identity in [*Gr-Sp*], and $in(p) = (p)$. Given a morphism f : *GTS*# → [*V*#](*TT*) = ([TG_0] ← [TG_1] → [TG_2], f_P) into the forgetful image of some derivation system *TT*, we must show there is a unique morphism g : [*DS*#](*GTS*#) → *TT* such that f = [*V*#](g) ○ v.

Let g be given by ([TG_0] ← [TG_1] → [TG_2], g_P) : *GDS*# → *TT* such that for all p in P, $g_P((p)) = f_P(p)$, extending by **;** to the whole of *GDS*#. It is clear that the required properties hold. ☺

Finally we have:

Theorem 10.5 The obvious projection [$P_{GraDS\#}$] : [*GraDS*#] → [*Gr-Sp*] is an opfibration, where all arrows ([TG_0] ← [TG_1] → [TG_2], f) : ([TG_0], [$\overline{G}_0$], P_0, π_0, $/_0$, $;_0$) → ([TG_2], [$\overline{G}_2$], P_2, π_2, $/_2$, $;_2$) such that f is an iso in *Set* are opcartesian.

11 Weakened Approaches

The preceding sections presented what was undeniably the simplest and most appealing treatment of its subject matter known to the authors thus far. By adhering to the greatest level of abstraction possible for graphs, spans, and morphisms of these under various operations, all the irritations of other approaches were neatly sidestepped. In particular, maximal abstraction, when properly expressed, makes various operations deterministic which in a less abstract setting would not be so, and this avoids a number of ungainly constructions designed to overcome this difficulty.

In this section we discuss the relationship between our treatment and what is obtained, both when one takes a less abstract perspective within our own technical context, and/or also when the more traditional technical route to abstraction is utilised, namely equivalence classes.

Essentially there are two areas that merit discussion, largely orthogonal to one another. The first concerns the entities to be manipulated and the purpose of the manipulation. The second is the technical strategy used for the manipulation. For the first area we have:

(A) Graphs as entities to be rewritten, and spans as productions and as transitions/ derivations.

(B) Graphs as type information, and spans as the mechanism for type change.

These are largely independent aspects. For the technical strategy area we have two essentially independent considerations to think about. The first concerns the level of abstraction at various vertices, giving:

1. Abstraction up to iso at all vertices of all diagrams; the abstract variant.
2. Abstraction up to std at selected vertices, and standard isomorphisms; the standard variant.
3. Abstraction up to id at selected vertices; the individual variant.
4. Abstraction up to std at all vertices, and standard isomorphisms; the fully standard variant.
5. Concrete diagram techniques based on abstract objects and arrows.

More or less independently we have:

(i) Abstract diagram techniques, including in particular interface-diagram categorical techniques.

(ii) Pseudoabstract diagram techniques (see below), or equivalence classes.

As is clear, we have used strategy 1.(i) to formulate both aspects (A) and (B) in this paper. In the rest of this section we will see that by forgetting various aspects of the theory, we obtain alternative accounts of the phenomena discussed, corresponding to other possible strategies. Of course where there is some forgetting going on, there is always a left adjoint lurking nearby, and we comment briefly on this at the end. To simplify what follows, we will continue to treat (A) and (B) together, and to stick to strand (i) until further notice.

Now for strategy 2.(i), let us consider using graphs up to standard isomorphisms as the entities to be rewritten and as type information. More specifically, we take abstract graphs to be abstract diagrams of kind std over a one vertex shape graph, i.e. $\langle G\rangle$. Since span composition figures at both horizontal composition level and at type change level, spans must still be abstract diagrams kinded as $(\langle A\rangle \leftarrow [B] \rightarrow \langle C\rangle)$, and span morphisms must be the evident generalisation, i.e. abstract diagrams like $(\langle A\rangle \leftarrow [B] \rightarrow \langle C\rangle)$ -⟨a,[b],c⟩-› $(\langle A'\rangle \leftarrow [B'] \rightarrow \langle C'\rangle)$ in an evident notation. In general, because certain isomorphisms are no longer available to us, two concrete diagrams, which differ only by the use of such a forbidden isomorphism at one vertex say, may no longer both be objects within the same abstract diagram. Our abstract diagrams thus become smaller.

Looking to other key elements of the theory, we must replace the category [*Gr*↓*Gr*-*Sp*] by a category ⟨*Gr*↓*Gr*-*Sp*⟩ which relates typed graphs up to std, $\langle X\rangle \rightarrow \langle A\rangle$, to other such typed graphs, $\langle Y\rangle \rightarrow \langle C\rangle$, via diagrams with kinds std and iso like $((\langle X\rangle \leftarrow [Y] = \langle Y\rangle)$ -⟨a,[b],c⟩-› $(\langle A\rangle \leftarrow [B] \rightarrow \langle C\rangle))$. Note that $[Y]$ and $\langle Y\rangle$ refer to the same collection of graphs but with different permitted internal structure. The vital category that describes how type change affects transition steps, [*D*-*Gr*-*MSp*↓*Gr*-*Sp*], is replaced by the category ⟨*D*-*Gr*-*MSp*↓*Gr*-*Sp*⟩, which relates typed abstract monic span morphism double cells like $(((\langle X\rangle \leftarrow [Y] \rightarrow \langle Z\rangle)$ -⟨x,[y],z⟩-› $(\langle \tilde{X}\rangle \leftarrow [\tilde{Y}] \rightarrow \langle \tilde{Z}\rangle)) \rightarrow \langle A\rangle)$ to similar such cells, by the evident type change operation, which we write as -⟨A[B]C⟩=› for brevity.

The details of this strategy 2.(i) variant are much like those of the 1.(i) variant with two crucial differences. The first is that the shrinkage of abstract diagrams in general, forces a loss of splitting in the analogous opfibration theorems, since there is no longer a canonical choice of opcartesian arrow, given a base category arrow and an object over its source. Specifically, Theorems 6.7, 6.8, 6.12, 6.13, 7.4, 7.5, still hold, but without the claims of a splitting. The second is that the left adjoint in Theorem 9.4 becomes a weak left adjoint. This is easy to see from the structure of the proof which exploited the split opfibration of Theorem 6.12. A modified proof can be modelled on the given proof (which is why it was presented that way); but instead of having for each $\pi_{GTS\#}((p, [d_1,d_2,d_3]))$ a canonical abstract monic span $(([X] \leftarrow [Y] \rightarrow [Z]) \rightarrow [TG_2])$ as described, we would have for each $\pi_{GTS\#}((p, \langle d_1,[d_2],d_3\rangle))$ as it would be, a number of possible $((\langle X\rangle \leftarrow [Y] \rightarrow \langle Z\rangle) \rightarrow \langle TG_2\rangle)$, obtained through different choices of opcartesian arrows for $\pi_{GTS\#}((p, \langle d_1,[d_2],d_3\rangle))$ and the $(\langle TG_0\rangle \leftarrow [TG_1] \rightarrow \langle TG_2\rangle)$ base arrow. The totality of these would make up the class of arrows for the requisite feeble functor.

Next, for strategy 3.(i), we use individual concrete graphs as as the entities to be rewritten and as type information. Since span composition figures just as much in this variant as in the others, spans must be abstract diagrams with kinds $A \leftarrow [B] \rightarrow C$, and span morphisms appear thus: $(A \leftarrow [B] \rightarrow C)$ -a,[b],c-› $(A' \leftarrow [B'] \rightarrow C')$ in a selfexplanatory notation. Our theory becomes affected by three connected things.

The first thing is a dramatic fragmentation phenomenon. For example, two graph grammars with isomorphic but non-identical start graphs, and identical in all other respects, must be considered as distinct entities. This is deeply unsatisfying in a theory of graph transformation. The same phenomenon touches other aspects of the theory. Thus the category [*Gr*↓*Gr*-*Sp*] must be replaced by *Gr*↓*Gr*-*Sp* which relates individual concrete typed graphs $X \rightarrow A$ to typed graphs $Y \rightarrow C$ via diagrams kinded as $((X \leftarrow [Y] = Y)$ -a,[b],c-› $(A \leftarrow [B] \rightarrow C))$, where $[Y]$ is the isomorphism class of graph Y. Likewise [*D*-*Gr*-*MSp*↓*Gr*-*Sp*] is replaced by the category *D*-*Gr*-*MSp*↓*Gr*-*Sp* which relates typed monic span morphism double cells $(((X \leftarrow [Y] \rightarrow Z)$ -x,[y],z-› $(X\tilde{} \leftarrow [Y\tilde{}] \rightarrow Z\tilde{}))$ $\rightarrow A)$ to similar such cells by the analogous type change operation written as -A[B]C=› for short.

The second thing is the cardinality explosion associated with this fragmentation. Since the isomorphism class of a graph is indeed a proper class, we see the consequences in the theory of graph transformation. As well as a proper class of graph grammars with isomorphic but non-identical start graphs, and identical in all other respects etc. etc., we have to recognise that the production name component of a graph grammar must itself become a proper class in general. This is forced on us by the transition system construction, which requires a separate production name $p/(d_1,[d_2],d_3)$ for each span morphism $(d_1,[d_2],d_3)$ with a given domain. Since these constitute a proper class because part of the morphism is concrete, the requirement for proper classes of production names follows immediately — even when we restrict attention to graph grammars that are otherwise completely finitistic. This too is deeply unsatisfying, and the only way of avoiding it is by some global choice mechanism, that supplies a $p/(d_1,[d_2],d_3)$ for *only some* $(d_1,[d_2],d_3)$, carefully specified.

The third thing, a consequence of the preceding two, is the nondeterminism that arises whenever the result of some operation is allowed to be any one graph out of an isomorphism class, and there is no a priori way of forcing the choice. This nondeterminism is noteworthy, but by no means as undesirable a feature as the other two. Its consequences are familiar from variant 2.(i), namely the loss of splitting in the split opfibrations of the same theorems as were quoted for variant 2.(i), and the weakening of the left adjoint in Theorem 9.4 to a weak left adjoint. Note that whereas in variant 2.(i) there was no ambiguity about the target of putative splitting arrows, there merely being more than one such arrow to consider in general, in the present case, because of the greater fragmentation, there is ambiguity even about the target, each distinct isomorphic graph providing a distinct target. So the failure of splitting is more dramatic.

Similar remarks apply to the weakening of the left adjoint. Instead of having for each $\pi_{\mathsf{GTS\#}}((p, [d_1,d_2,d_3]))$ a canonical abstract monic span $(([X] \leftarrow [Y] \rightarrow [Z]) \rightarrow [TG_2])$ as described, we would have for each $\pi_{\mathsf{GTS\#}}((p, (d_1,[d_2],d_3)))$ as it would be, a myriad possible $((X \leftarrow [Y] \rightarrow Z) \rightarrow TG_2)$, differing amongst themselves in their choice of isomorphic variants of X and Z. The totality of these would make up the class of arrows for the requisite feeble functor. This outlines the main features of the individual variant.

The fully standard variant, 4.(i), where we take all vertices of all diagrams to be of kind std, is a little different in character. We run into the difficulty that horizontal composition of spans is doggedly only up to arbitrary, not standard, isomorphisms. Therefore we cannot take a span as a single abstract diagram $\langle A\rangle \leftarrow \langle B\rangle \rightarrow \langle C\rangle$, but must take the collection of such, indexed by the group of automorphisms of B according to Lemma 4.4.3. This we can write as $\langle A\rangle \leftarrow \{\langle B\rangle\} \rightarrow \langle C\rangle$ to indicate that several abstract diagrams of kind std are being considered simultaneously. Of course this can be viewed as nothing more than a mild repackaging of the 3.(i) variant, done by discarding just those concrete diagram morphisms that use isomorphisms of B forbidden by the kind std retriction, so we do not go into detail.

Now consider the fact that throughout the whole of Sections 8-10 we never made serious use of the concrete diagram morphisms inside an abstract diagram. If we take an abstract diagram and forget all the concrete diagram morphisms between its objects, we get a discrete functor category. We call this a pseudoabstract diagram. An equivalent way of viewing a pseudoabstract diagram is as just an equivalence class of concrete diagrams, the elements of which are related by the claim of the *existence* of (one or more) suitable natural transformations (rather than these natural transformations being part of the data defining the entity, as is the case for abstract diagrams). It will not take the reader long to realise that we could redo the whole of Sections 8-10, for variants 1-4 described above, using an easily imagined theory of interface-pseudoabstract-diagram categories (or its equivalent, a theory of pasting etc. of appropriate equivalence classes). This gives us strand (ii) in the overall scheme outlined above. We note that in strand (ii) there is almost no difference between variants 3 and 4, since a variant 3 description will be based on a collection of equivalence classes, while the corresponding variant 4 description will be based on its union.

This discussion brings us to the final variant of the theory, variant 5, using concrete diagrams built out of abstract objects and arrows. There is not much to say here due to the comprehensive exposition of relationships between such concrete diagrams and their abstract counterparts in Section 4.3. The most direct route to this variant is via the 4.(i) variant, which is cast in terms of abstract diagrams completely of kind std. Such abstract diagrams (and also collections of them, as required), are readily turned into concrete diagrams built out of abstract objects and abstract arrows using the functors $\Im_{\mu}^{\mathsf{std},\bullet}$ for various shapes μ. Thus for example, typed graphs become concrete arrows $(G^{\Diamond} \to TG^{\Diamond}) : \langle G \rangle \to \langle TG \rangle$. Spans on the other hand, because of the necessity for horizontal composition, must remain collections of concrete spans which we can write by analogy with variant 4.(i) as $(A^{\Diamond} \leftarrow \{B^{\Diamond}\} \to C^{\Diamond}) : \langle A \rangle \to \langle C \rangle$. Similarly for the remaining aspects of the theory. This completes the taxonomy of alternative variants of our theory.

Of the variants discussed, the ones featuring graphs up to std in various ways, allow us connect the present theory to that in eg. Corradini et al. (1994a,b,c, 1996a,b). In particular, isomorphisms up to std are required to properly distinguish individual events through concurrent executions of graph grammars.

Finally we point out that given that we have generated all our variants by forgetting some aspects of the original theory, and these aspects (specifically, the diagram morphisms in question) were characterised by the canonical property of maximality, there will be left adjoints to the forgetful functors that simply reinstate the forgotten structure. The workings of these left adjoints will be quite straightforward, so we do not pause to elaborate the details.

12 Conclusions

In the preceding sections we set up a general framework for reasoning about how typed diagrams of different kinds behave under change of typing, where the change of typing is controlled by a span, and the crucial observation turned out to be that the behaviour of diagrams is exactly captured by an opfibration over the type change category. Different graph rewriting phenomena were then reduced almost entirely to choosing the right kind of diagram to work with to describe the situation in question. We were able to define a notion of graph grammar, and then via appropriate free constructions, to obtain notions of transition and derivation system that captured various ways of manipulating the grammar data algebraically. Much of this work can be seen as extending the double category nature of [*D-Gr-Sp*] to the triple categories [*D-Gr-MSp-* DP#↓*Gr-Sp*], and thence to the richer situation in which production names are present and must behave properly. Provided we adhered to the maximum level of abstraction, everything went smoothly. Alternative treatments, somewhat more in the spirit of existing work, emerged in the preceding section by a process of forgetting various concrete diagram morphisms. The one message that emerges clearly from this work is that in examining questions of abstractness where the subject matter is categorical, functor categories provide the most convincing approach, and treatments involving equivalence classes can be smoothly recovered from them post hoc.

13 Acknowledgements

The work reported here was initiated while the first author was on leave at the Computer Science Department of the University of Pisa. The hospitality of that department, and the financial support of the British Council, the Consiglio Nazionale delle Ricerche, and the Staff Exchange Scheme of the European Commission's ERASMUS programme are gratefully acknowledged. The second author is partly supported by the EC TMR Network GETGRATS (General Theory of Graph Transformation Systems), and by the EC Fixed Contribution Contract No. EBRFMBICT960840.

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.

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

Banach R. (1995); DPO Rewriting and Abstract Semantics via Opfibrations. *in*: Proc. SEGRAGRA-95, Corradini, Montanari (eds.), ENTCS **2**, Elsevier, `http://www.elsevier.nl/locate/entcs/volume2.html`. Extended version: University of Manchester Dept. of Computer Science Technical Report UMCS 96-8-1, `http://www.cs.man.ac.uk/csonly/cstechrep/Abstracts/UMCS-96-8-1.html`

Bastiani A., Ehresmann C. (1974); Multiple Functors I: Limits Relative to Double Categories. Cahiers de Topologie et Géométrie Differentielle, **15** 545-621.

Barr M., Wells C. (1990); Category Theory for Computing Science. (1 ed.), Prentice-Hall.

Bauderon M. (1995); Parallel Rewriting of Graphs Through the Pullback Approach. *in*: Proc. SEGRAGRA-95, Corradini, Montanari (eds.), ENTCS **2** Elsevier, `http://www.elsevier.nl/locate/entcs/volume2.html`

Corradini A., Ehrig H., Löwe M., Montanari U., Rossi F. (1994a); 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., Löwe M., Montanari U., Rossi F. (1994b); 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., Löwe M., Montanari U., Rossi F. (1994c); An Event Structure Semantics for Safe Graph Grammars. *in*: Proc. I.F.I.P. Working Conference PROCOMET-94, Olderog (ed.), 423-446.

Corradini A., Ehrig H., Löwe M., Montanari U., Rossi F. (1996a); An Event Structure Semantics for Graph Grammars with Parallel Productions. *in*: Proc. Fifth International Workshop on Graph Grammars and their Application to Computer Science 1994, Williamsburg U.S.A., Cuny, Ehrig, Engels, Rozenberg (eds.), L.N.C.S. **1073** 240-256.

Corradini A., Ehrig H., Löwe M., Montanari U., Padberg J. (1996b); The Category of Typed Graph Grammars and its Adjunctions with Categories of Derivations. *in*: Proc. Fifth International Workshop on Graph Grammars and their Application to Computer Science 1994, Williamsburg U.S.A., Cuny, Ehrig, Engels, Rozenberg (eds.), L.N.C.S. **1073** 56-74.

Corradini A., Montanari U., Rossi F., Ehrig H., Heckel R., Löwe M., (1997); Algebraic Approaches to Graph Transformation Part I: Basic Concepts and Double Pushout Approach.

in: Handbook of Graph Grammars and Computing by Graph Transformation. Rozenberg (ed.), 163-245.

Ehresmann C. (1963); Catégories Doubles et Catégories Structurées. C. R. Acad. Sci. Paris, **256** 1198-1201.

Ehrig H., Pfender M., Schneider H. J. (1973); Graph Grammars: an Algebraic Approach. *in*: Proc. I.E.E.E. Conference on Automata and Switching Theory, 167-180.

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. (1987); 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.

Ehrig H., Heckel R., Taentzer G., Engels G. (1997); A View-Oriented Approach to System Modelling Using Graph Transformation, *submitted*.

Gadducci F., Montanari U. (1995); Enriched Categories as Models of Computation. Proc. 3rd Italian Conference on Theoretical Computer Science, Ravello.

Gray J.W. (1966); Fibered and Cofibered Categories. *in*: Proc. La Jolla Conference on Categorical Algebra, Eilenberg et al. (eds.), 21-83, Springer, Berlin.

Grothendieck A. (1961); Catégories Fibrés et Descente, Seminaire de Géométrie Algébrique de l'Institut des Hautes Études Scientifiques Paris, *also in*: L.N.M. **224** 145-194, Springer, Berlin.

Heckel R., Corradini A., Ehrig H., Löwe M. (1996); Horizontal and Vertical Structuring of Typed Graph Transformation Systems. Math. Struct. Comp. Sci. **6**, 613-648.

Heckel R., Corradini A., Ehrig H., Padberg J., Wolter U. (1997); Coalgebraic Loose Semantics as Specification Frame for Graph Transformation Systems. Draft. Applied Categorical Structures special issue on the Dagstuhl Seminar 9637 on Graph Transformations in Computer Science, *submitted*.

Kainen P. (1971); Weak Adjoint Functors. Math. Z. **122**, 1-9.

Krishnan V. (1981); An Introduction to Category Theory. North-Holland.

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

Löwe M. (1993); Algebraic Approach to Single Pushout Graph Transformation. Theor. Comp. Sci. **109** 181-224.

Palmquist P. (1970); The Double Category of Adjoint Squares. *in*: Reports of the Midwest Category Seminar V, Gray, MacLane (eds.), LNM **195**, 123-153, Springer.

Ribeiro L. (1996); Parallel Composition and Unfolding Semantics of Graph Grammars. Thesis Dr.-Ing. Technical University of Berlin.