Stronger Compositions for Retrenchments

R. Banach, C. Jeske
School of Computer Science, Manchester University, Manchester, M13 9PL, U.K.
banach@cs.man.ac.uk, jeskec@cs.man.ac.uk

Abstract. Noting that the usual 'propositionally' based way of composing retrenchments can yield many 'junk' cases, alternative approaches to composition are introduced (via notions of tidy, neat, and fastidious retrenchments) that behave better in this regard. These alternatives do however make other issues such as associativity harder. The technical details are presented for vertical composition of retrenchments (i.e. the composition of successive retrenchment steps).
Keywords. Retrenchment, Composition, Associativity.

1 Introduction

In [Banach et al. (2007)] the authors gave a comprehensive and broadly based overview of the motivations for introducing retrenchment — background and context were extensively discussed, and some key issues that arise with retrenchment were described. Briefly, retrenchment attempts to provide a level of rigour, broadly comparable to that which is found in notions of model based refinement [de Roever and Engelhardt (1998), Derrick and Boiten (2001)], but in situations such as model evolution, in which model based refinement is simply too demanding. It does so by judiciously weakening typical refinement-style proof obligations (POs) by incorporating additional relations (the retrenchment data) to add expressivity.[1] In [Banach et al. (2008)], various kinds of composition for retrenchment were studied, and these were shown to be both associative individually, and mutually coherent.

Composition mechanisms are not simply God-given, but are a matter for definition. One posits a definition for a law of composition (in a given algebraic structure), and then shows that it is sound. In many algebraic structures there are usually few 'sensible' candidates for a composition of a particular type; often there is only one. Viewing retrenchment as a particular kind of algebraic structure (an instance of the algebraic structure consisting of the retrenchment data for some specific retrenchment), the composition mechanisms of [Banach et al. (2008)] are certainly the ones that most obviously come to mind. These mechanisms are based on straightforward 'propositional' reasoning. By 'propositional' we mean that although retrenchment data are (relations described by) predicates, the reasoning needed is almost the same as if the data consisted just of propositions, and the use of predicate calculus hardly goes beyond the movement of existential scopes across disjunctions.

However, while being perfectly sound, these mechanisms do have a tendency to proliferate 'junk' cases in the highest level of the retrenchment conclusion when used in specific application contexts. This is because retrenchment offers a disjunction of a number of cases in its conclusion, only one of which needs to be true at any time. Under composition, the distributive law wastes no time in multiplying the possibilities,

1. A broad view of the current state of development of retrenchment at any time may be gained from the Retrenchment Homepage [RET].

and when a case that is false is combined with a case that is true (at a given point), the result is a(nother) case that is false. The number of such false cases can grow exponentially in the number of retrenchments that are being composed, potentially interfering with the usefulness of retrenchment in the applications sphere. It is not hard to see that the simple 'propositional' reasoning referred to above does nothing to alleviate this situation due to its obliviousness to the details of the underlying relations.

In this paper we attempt to curtail the proliferation of junk cases by exploiting 'semantic' insights of varying depth to yield alternative, stronger composition laws. By 'semantic' we mean that these stronger composition laws perforce take greater note of the detailed properties of the relations that comprise the retrenchment data, and of the fact that the POs reason about the steps of transition systems. Using this approach we can successfully limit the junk proliferation in varying degrees, but the price we have to pay is that various considerations, notably closure of the constructions and associativity, become technically more troublesome.

Our investigations in this paper are confined exclusively to vertical composition, which is the composition of successive retrenchment-described phases of development, or of model evolution steps. Our starting point is the 'propositional' version of vertical composition in [Banach et al. (2008)], which is recapitulated. This makes the present paper technically self-contained. Moreover, the relationship between the results derived here and the 'propositional' case, makes it clear how things would go for analogous stronger versions of the other types of composition studied in [Banach et al. (2008)], since those other types of composition typically just differ in the variables in the retrenchment data which are being matched up, whereas the subsequent manipulation of the data usually follows very similar lines to the vertical case.

In more detail, the rest of the paper is as follows. Section 2 recalls the basic definitions for retrenchment, reviews the usual 'propositionally' based vertical composition, and recalls default retrenchments. Section 3 introduces some special cases of retrenchment, the tidy, neat and fastidious retrenchments. We give some motivating commentary about these special forms, and we see that default retrenchments are naturally fastidious, and under additional constraints, also neat or tidy. In Section 4 we consider stronger ways of composing a pair of retrenchments than the usual 'propositional' technique, relying on assumptions about the transition relations of the systems involved. Though showing that the stronger techniques are sound is not problematic, no attempt is made yet to show that the new compositions preserve the stronger properties assumed in their hypotheses. Beginning with some counterexamples to illustrate why the issue is nontrivial, Section 5 explores the closure and associativity properties of the stronger composition methods. After some protracted calculations which are relegated to the appendix of the online version of this paper available from [RET] (likewise some of the longer derivations from Section 4), sufficient conditions are established which guarantee that the needed closure and associativity properties hold. Section 6 concludes, and indicates briefly the reasons why there are in fact a number of alternative approaches to the issue of stronger compositions for retrenchments —quite aside from the strategy pursued here— arising from the rich nature of retrenchment data and of its equally rich relationship to the underlying transition systems of which it speaks.

2 Retrenchment

In this section we give our notational conventions and present our basic definitions. Retrenchment offers a variation on the usual kind of model based refinement step. Accordingly, we suppose that there is an abstract system *Abs* and a concrete one *Conc*. The abstract system has a set of operation names $\mathsf{Ops}_{\mathrm{A}}$, with typical element Op_{A}. An operation Op_{A} will work on the abstract state space U having typical element u (the before-state), and an input space $\mathsf{I}_{Op_{\mathrm{A}}}$ with typical element i. Op_{A} will produce an after-state typically written u', once more in U, and an output o drawn from an output space $\mathsf{O}_{Op_{\mathrm{A}}}$. Initial states are those that satisfy the property $Init_{\mathrm{A}}(u')$. In this paper we work exclusively in a transition system framework, so an operation Op_{A} is given by its transition or step relation consisting of steps u -(i, Op_{A}, o)-› u'. The set of such steps is written $stp_{Op_{\mathrm{A}}}(u, i, u', o)$. At the concrete level we have a similar setup. The operation names are $Op_{\mathrm{C}} \in \mathsf{Ops}_{\mathrm{C}}$. States are $v \in \mathsf{V}$, inputs are $j \in \mathsf{J}_{Op_{\mathrm{C}}}$, outputs are $p \in \mathsf{P}_{Op_{\mathrm{C}}}$. Initial states satisfy $Init_{\mathrm{C}}(v')$. Typical transitions are v -(j, Op_{C}, p)-› v', elements of the concrete step relation $stp_{Op_{\mathrm{C}}}(v, j, v', p)$.

2.1 The Retrenchment POs and Simulation Relation

Given the preceding, retrenchment is defined by three facts. Firstly $\mathsf{Ops}_{\mathrm{A}} \subseteq \mathsf{Ops}_{\mathrm{C}}$, i.e. to each abstract operation there corresponds a concrete operation assumed to have the same name.[2] The inclusion can be proper so the converse need not hold.[3] Secondly we have relations as follows: a retrieve relation $G(u, v)$ between abstract and concrete state spaces; and for each operation $Op_{\mathrm{A}} \in \mathsf{Ops}_{\mathrm{A}}$, within, output and concedes relations: $P_{Op}(i, j, u, v)$, $O_{Op}(o, p; u', v', i, j, u, v)$ and $C_{Op}(u', v', o, p; i, j, u, v)$ respectively.[4] The within and concedes relations are over the variables shown, i.e. the within relations involve the inputs and before-states, while the concedes relations involve predominantly the outputs and after-states, though inputs and before-states can also feature if required. We suppress the 'A' and 'C' subscripts on Op in these relations since they concern both levels of abstraction equally. Thirdly a collection of properties (the proof obligations or POs) must hold. The initial states must satisfy:

$$Init_{\mathrm{C}}(v') \;\Rightarrow\; (\exists\, u' \bullet Init_{\mathrm{A}}(u') \wedge G(u', v')) \tag{2.1}$$

and for every corresponding operation pair Op_{A} and Op_{C}, the abstract and concrete step relations must satisfy the operation PO:

$$\begin{aligned} & G(u, v) \wedge P_{Op}(i, j, u, v) \wedge stp_{Op_{\mathrm{C}}}(v, j, v', p) \;\Rightarrow \\ & \quad (\exists\, u', o \bullet stp_{Op_{\mathrm{A}}}(u, i, u', o) \wedge ((G(u', v') \wedge O_{Op}(o, p; u', v', i, j, u, v)) \vee \\ & \qquad C_{Op}(u', v', o, p; i, j, u, v))) \end{aligned} \tag{2.2}$$

The retrenchment POs above give a good idea of what the various components of the retrenchment data do. Thus the retrieve relation plays a conventional role, relating the two state spaces. The within relation acts as a constraint, limiting the scope of what the retrenchment is able to claim. The output relation strengthens the retrieve

2. This confirms that the 'A' and 'C' subscripts on operation names are meta level tags.
3. Operations with names outside $\mathsf{Ops}_{\mathrm{A}} \cap \mathsf{Ops}_{\mathrm{C}}$ play no role here, so the relationship between $\mathsf{Ops}_{\mathrm{A}}$ and $\mathsf{Ops}_{\mathrm{C}}$ could easily be more symmetrical. The stated inclusion is the usual case.
4. We recall that the semicolons in O_{Op} and C_{Op} are purely cosmetic, separating the variables 'of most interest' from others which are permitted, if seldom needed in practice.

relation in the conclusion when the latter is re-established by the PO, allowing more incisive statements to be made. Finally, the concedes relation permits a description of the state of affairs when re-establishing the retrieve relation in the conclusion fails. It is this last aspect which is most characteristic of retrenchment, and which most differentiates it from various flavours of model based refinement.

Associated with the operation PO is the retrenchment simulation relation given by removing the quantification, and changing the implication to a conjunction:

$$G(u, v) \wedge P_{Op}(i, j, u, v) \wedge stp_{Op_C}(v, j, v', p) \wedge stp_{Op_A}(u, i, u', o) \wedge \\ ((G(u', v') \wedge O_{Op}(o, p; u', v', i, j, u, v)) \vee C_{Op}(u', v', o, p; i, j, u, v)) \tag{2.3}$$

The simulation relation is what we get by excising the 'don't care' interpretation of the implication in (2.2). As we will see below, much of the technical manipulation in this paper is concerned with establishing the simulation relation by means other than via the operation PO.

2.2 Vertical Composition

The usual, 'propositional' vertical composition for retrenchments, gives the retrenchment data for the composition of two retrenchments steps, in terms of the retrenchment data of the components. In outline this is as follows:

$$G_{(1,2)} \equiv G_1 \fatsemi G_2 \tag{2.4}$$

$$P_{Op,(1,2)} \equiv (G_1 \wedge P_{Op,1}) \fatsemi (G_2 \wedge P_{Op,2}) \tag{2.5}$$

$$O_{Op,(1,2)} \equiv O_{Op,1} \fatsemi O_{Op,2} \tag{2.6}$$

$$C_{Op,(1,2)} \equiv (G'_1 \wedge O_{Op,1} \fatsemi C_{Op,2}) \vee (C_{Op,1} \fatsemi G'_2 \wedge O_{Op,2}) \vee (C_{Op,1} \fatsemi C_{Op,2}) \tag{2.7}$$

In (2.4)-(2.7) the forward relational composition $\fatsemi$ is via the relevant variables of the intermediate system (i.e. the system that is the target of the first retrenchment and the source of the second). The structure is relatively straightforward, aside from the concedes relation, which states that: either the first system behaves well and the second displays some 'exceptional' behaviour; or vice versa; or both are exceptional.

In the sequel we need many instances of formulae with a similar structure, but often displaying small variations in precise values of intermediate variables. For precision, we must descend to writing these out explicitly, so in Definition 2.1 we give the precise version of (2.4)-(2.7) for reference. In [Banach et al. (2008)] there is a proof that this definition is sound —i.e. that the relations given are indeed the data of a retrenchment— which can also be discerned from the proof of Theorem 4.1 below by erasing some of the details there.

Definition 2.1 Let Sys_0 (with system variables u_0, i_0, u'_0, o_0) be retrenched to Sys_1 (with system variables u_1, i_1, u'_1, o_1) using G_1, $\{P_{Op,1}, O_{Op,1}, C_{Op,1} \mid Op \in \mathsf{Ops}_0\}$, and Sys_1 be retrenched to Sys_2 (with system variables u_2, i_2, u'_2, o_2) using G_2, $\{P_{Op,2}, O_{Op,2}, C_{Op,2} \mid Op \in \mathsf{Ops}_1\}$. Then Sys_0 is retrenched to Sys_2 using retrieve, within, output, and concedes relations $G_{(1,2)}$, $\{P_{Op,(1,2)}, O_{Op,(1,2)}, C_{Op,(1,2)} \mid Op \in \mathsf{Ops}_0\}$, defined by:

$$G_{(1,2)}(u_0, u_2) \equiv [\exists\, u_1 \bullet G_1(u_0, u_1) \wedge G_2(u_1, u_2)] \tag{2.8}$$

$$P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \equiv$$
$$[\exists\, u_1, i_1 \bullet G_1(u_0, u_1) \wedge G_2(u_1, u_2) \wedge$$
$$P_{Op,1}(i_0, i_1, u_0, u_1) \wedge P_{Op,2}(i_1, i_2, u_1, u_2)] \tag{2.9}$$

$$O_{Op,(1,2)}(o_0, o_2; u'_0, u'_2, i_0, i_2, u_0, u_2) \equiv$$
$$[\exists\, u'_1, o_1, u_1, i_1 \bullet O_{Op,1}(o_0, o_1; \dots) \wedge O_{Op,2}(o_1, o_2; \dots)] \tag{2.10}$$

$$C_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2) \equiv$$
$$[\exists\, u'_1, o_1, u_1, i_1 \bullet$$
$$(G_1(u'_0, u'_1) \wedge O_{Op,1}(o_0, o_1; u'_0, u'_1, i_0, i_1, u_0, u_1) \wedge$$
$$C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2)) \vee$$
$$(C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge$$
$$G_2(u'_1, u'_2) \wedge O_{Op,2}(o_1, o_2; u'_1, u'_2, i_1, i_2, u_1, u_2)) \vee$$
$$(C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge$$
$$C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2))] \tag{2.11}$$

2.3 Default Retrenchments

Default retrenchments make precise the intuition that 'an arbitrary pair of systems' can be related by retrenchment — a retrenchment moreover, that lies at the opposite extreme to the '$P_{Op} \equiv \mathsf{false}$' retrenchment which can obviously also relate an arbitrary pair of systems (modulo remarks about initialisation). Since default retrenchments arise in a generic manner, they can be used to give generic treatments of many situations via retrenchment. For example in [Banach and Jeske (2009b)], we give a treatment of simple feature engineering based on default retrenchments. We recall the following from [Banach et al. (2007)].

Proposition 2.2 Suppose we are given two systems *Abs* and *Conc*, with $\mathsf{Ops}_\mathrm{A} \subseteq \mathsf{Ops}_\mathrm{C}$. Let $G(u, v)$ and $\{P_{Op}(i, j, u, v), O_{Op}(o, p; u', v', i, j, u, v) \mid Op \in \mathsf{Ops}_\mathrm{A}\}$ be arbitrary relations in the variables stated. Let default within and concedes relations $\{P^{\mathrm{Def}}{}_{Op} \mid Op \in \mathsf{Ops}_\mathrm{A}\}$ and $\{C^{\mathrm{Def}}{}_{Op} \mid Op \in \mathsf{Ops}_\mathrm{A}\}$ be given by:

$$P^{\mathrm{Def}}{}_{Op}(i, j, u, v) \equiv$$
$$(G(u, v) \wedge P_{Op}(i, j, u, v) \wedge$$
$$(\exists\, u', o, v', p \bullet stp_{Op_\mathrm{A}}(u, i, u', o) \wedge stp_{Op_\mathrm{C}}(v, j, v', p))) \tag{2.12}$$

$$C^{\mathrm{Def}}{}_{Op}(u', v', o, p; i, j, u, v) \equiv$$
$$(G(u, v) \wedge P_{Op}(i, j, u, v) \wedge stp_{Op_\mathrm{A}}(u, i, u', o) \wedge stp_{Op_\mathrm{C}}(v, j, v', p) \wedge$$
$$\neg\, (G(u', v') \wedge O_{Op}(o, p; u', v', i, j, u, v))) \tag{2.13}$$

Then G and $\{P^{\mathrm{Def}}{}_{Op}, O_{Op}, C^{\mathrm{Def}}{}_{Op} \mid Op \in \mathsf{Ops}_\mathrm{A}\}$ define a retrenchment from *Abs* to *Conc* called the default retrenchment from *Abs* to *Conc*.

It is clear that default retrenchments are parameterised by the assumed retrieve, within, and output relations. They stand in contrast to bespoke retrenchments, ones specifically crafted by designers to express the goals of their design step. Normally, we would expect a bespoke retrenchment to have a concession 'weaker' than $C^{\mathrm{Def}}{}_{Op}$ in order to express the intended design goal of the development step without cluttering the concession with all possible facts that one could include in it. A more precise statement is made at the end of Section 3. The gap between what a bespoke concession allows, and what the two systems involved can realise, is one source of the junk cases we attack in this paper.

3 Closures, Tidiness, Neatness, Fastidiousness

Suppose we have a retrenchment from *Abs* to *Conc* as previously.

Definition 3.1 We define the retrieve closure of an abstract operation Op of the retrenchment by:

$$\begin{aligned}&\overline{G}_{Op}(u', v', o, p; i, j, u, v) \equiv \\ &\quad (G(u, v) \land P_{Op}(i, j, u, v) \land stp_{Op_{\mathrm{A}}}(u, i, u', o) \land stp_{Op_{\mathrm{C}}}(v, j, v', p) \land \\ &\qquad G(u', v') \land O_{Op}(o, p; u', v', i, j, u, v))\end{aligned} \tag{3.1}$$

and the concedes closure of Op by:

$$\begin{aligned}&\overline{C}_{Op}(u', v', o, p; i, j, u, v) \equiv \\ &\quad (G(u, v) \land P_{Op}(i, j, u, v) \land stp_{Op_{\mathrm{A}}}(u, i, u', o) \land stp_{Op_{\mathrm{C}}}(v, j, v', p) \land \\ &\qquad C_{Op}(u', v', o, p; i, j, u, v))\end{aligned} \tag{3.2}$$

It is clear that the retrieve and concedes closures of Op isolate the 'refining' and 'non-refining' parts of the retrenchment simulation relation (2.3). The following two results are unsurprising.

Proposition 3.2 Let a retrenchment be defined in the usual manner. Then the operation PO (2.2) is satisfied iff the (modified) PO (3.3) is satisfied:

$$\begin{aligned}&G(u, v) \land P_{Op}(i, j, u, v) \land stp_{Op_{\mathrm{C}}}(v, j, v', p) \Rightarrow \\ &\quad (\exists\, u', o \bullet \overline{G}_{Op}(u', v', o, p; i, j, u, v) \lor \overline{C}_{Op}(u', v', o, p; i, j, u, v))\end{aligned} \tag{3.3}$$

Proof. Straightforward: in the forward direction we just note that the facts beyond $stp_{Op_{\mathrm{A}}} \land G' \land O_{Op}$ asserted in $\overline{G}_{Op}$ are present in the hypotheses (similarly for $\overline{C}_{Op}$); in the backward direction we are simply weakening the conclusion. ☺

Proposition 3.3 Let a retrenchment be defined in the usual manner. Then the operation PO (2.2) is satisfied iff the original output and concedes relations O_{Op} and C_{Op} are replaced in (2.2) by $\overline{G}_{Op}$ and $\overline{C}_{Op}$ respectively.

Proof. Straightforward. ☺

Thus $\overline{G}_{Op}$ and $\overline{C}_{Op}$ constrain the originally given $G \land O_{Op}$ and C_{Op} of the retrenchment conclusion to the maximum extent possible, while still keeping the operation PO (or an analogue of it) provable. In particular, $\overline{G}_{Op}$ and $\overline{C}_{Op}$ are never true without there being abstract and concrete transitions (which also satisfy G and P_{Op}) that witness that truth, something that need not hold for the original $G \land O_{Op}$ or C_{Op} in isolation. In a sense, as noted for bespoke retrenchments in Section 2.3, the original $G \land O_{Op}$ or C_{Op} will typically contain just those facts that the designer deems important to capture in the development step (typically some statements about the after-states and outputs only), without including everything that can possibly be said (such as delineating the before-states and inputs that lead to those after-states and outputs), and to that extent they can be viewed as a shorthand for $\overline{G}_{Op}$ and $\overline{C}_{Op}$ respectively.

Since $G \land O_{Op}$ and C_{Op} merely express some properties of interest to the development step, while $\overline{G}_{Op}$ and $\overline{C}_{Op}$ actually contain simulating pairs of transitions, we can view the latter as the transitions of a joint (*Abs*/*Conc*) system, and the passage from $G \land O_{Op}$ and C_{Op} to $\overline{G}_{Op}$ and $\overline{C}_{Op}$ as the addition of suitable 'guards' to the former, to constrain the wider selection of before-state/after-state pairs permitted by $G \land O_{Op}$

and C_{Op} (not forgetting I/O) to those that can be realised by *Abs* and *Conc*.[5] We pursue the guards idea further in the next definition.

Definition 3.4 We define the following relations for an abstract operation Op:

$$\mathrm{pre}^{\mathrm{Ret}}{}_{Op}(u, i, v, j) \;\equiv\; (\exists\, u', o, v', p \bullet \overline{G}_{Op}(u', v', o, p; i, j, u, v)) \tag{3.4}$$

$$\mathrm{pre}^{\mathrm{Con}}{}_{Op}(u, i, v, j) \;\equiv\; (\exists\, u', o, v', p \bullet \overline{C}_{Op}(u', v', o, p; i, j, u, v)) \tag{3.5}$$

$$\mathrm{pre}^{\mathrm{RetA}}{}_{Op}(u, i) \;\equiv\; (\exists\, v, j \bullet \mathrm{pre}^{\mathrm{Ret}}{}_{Op}(u, i, v, j)) \tag{3.6}$$

$$\mathrm{pre}^{\mathrm{RetC}}{}_{Op}(v, j) \;\equiv\; (\exists\, u, i \bullet \mathrm{pre}^{\mathrm{Ret}}{}_{Op}(u, i, v, j)) \tag{3.7}$$

$$\mathrm{pre}^{\mathrm{ConA}}{}_{Op}(u, i) \;\equiv\; (\exists\, v, j \bullet \mathrm{pre}^{\mathrm{Con}}{}_{Op}(u, i, v, j)) \tag{3.8}$$

$$\mathrm{pre}^{\mathrm{ConC}}{}_{Op}(v, j) \;\equiv\; (\exists\, u, i \bullet \mathrm{pre}^{\mathrm{Con}}{}_{Op}(u, i, v, j)) \tag{3.9}$$

These various relations lend themselves to a natural interpretation as guards. Thus $\mathrm{pre}^{\mathrm{Ret}}{}_{Op}$ and $\mathrm{pre}^{\mathrm{Con}}{}_{Op}$ may be seen as guards for the refining and non-refining parts of the joint *Abs*/*Conc* system. The $\mathrm{pre}^{\mathrm{RetA}}{}_{Op}(u, i)$ and $\mathrm{pre}^{\mathrm{ConA}}{}_{Op}(u, i)$ relations guard the abstract transitions alone, albeit with a 'secret' awareness of the concrete system through the relationship with the concrete system which has been quantified away. The $\mathrm{pre}^{\mathrm{RetC}}{}_{Op}(v, j)$ and $\mathrm{pre}^{\mathrm{ConC}}{}_{Op}(v, j)$ relations perform the analogous service for the concrete system.

Proposition 3.5 For any retrenchment the following holds:

$$(\mathrm{pre}^{\mathrm{Ret}}{}_{Op}(u, i, v, j) \lor \mathrm{pre}^{\mathrm{Con}}{}_{Op}(u, i, v, j)) \;\equiv\; P^{\mathrm{Def}}{}_{Op}(i, j, u, v) \tag{3.10}$$

where $P^{\mathrm{Def}}{}_{Op}(i, j, u, v)$ is defined by (2.12).

Proof. If either $\mathrm{pre}^{\mathrm{Ret}}{}_{Op}(u, i, v, j)$ or $\mathrm{pre}^{\mathrm{Con}}{}_{Op}(u, i, v, j)$ is true, then $P^{\mathrm{Def}}{}_{Op}(i, j, u, v)$ follows by weakening. Conversely, suppose $P^{\mathrm{Def}}{}_{Op}(i, j, u, v)$ holds as witnessed by some $\underline{u}'$, $\underline{o}$, v', p. Then for u, i, v, j, v', p, we have $(G \land P_{Op} \land stp_{Op_{\mathrm{C}}})$, so we can use Proposition 2.2 to deduce $\overline{G}_{Op}(u', v', o, p; i, j, u, v) \lor \overline{C}_{Op}(u', v', o, p; i, j, u, v)$ for some u', o (not necessarily related to $\underline{u}'$, $\underline{o}$). After this, we get $\mathrm{pre}^{\mathrm{Ret}}{}_{Op}(u, i, v, j) \lor \mathrm{pre}^{\mathrm{Con}}{}_{Op}(u, i, v, j)$ by quantification. ☺

Proposition 3.6 Let a retrenchment be defined in the usual manner. Then the operation PO (2.2) is satisfied iff (3.11) is satisfied:

$$\begin{aligned}
&G(u, v) \land P_{Op}(i, j, u, v) \land stp_{Op_{\mathrm{C}}}(v, j, v', p) \;\Rightarrow\\
&\quad (\exists\, u', o \bullet stp_{Op_{\mathrm{A}}}(u, i, u', o) \land\\
&\qquad ((\mathrm{pre}^{\mathrm{Ret}}{}_{Op}(u, i, v, j) \land G(u', v') \land O_{Op}(o, p; u', v', i, j, u, v)) \lor\\
&\qquad (\mathrm{pre}^{\mathrm{Con}}{}_{Op}(u, i, v, j) \land C_{Op}(u', v', o, p; i, j, u, v))))
\end{aligned} \tag{3.11}$$

Proof. Similar to Proposition 3.2. ☺

5. A note of caution about the word 'guard'. Normally it is used in the context of a specific notion of refinement, where it bears a precise technical meaning definable in terms of the technical apparatus of the refinement notion. Here there is no refinement, so the word is being used more informally. This is further underlined by the fact that if there *were* a notion of refinement within the current discourse, possessing a precise definition of guard, *grd* say, we would additionally demand compatibility conditions such as $G \land P_{Op} \Rightarrow grd$ etc. See [Banach et al. (2007), Banach (2009)] for a discussion of the relevant issues. However, since there is no refinement here, there are no conditions either.

Thus far we do not seem to have accomplished much besides relatively trivial reformulations of the retrenchment operation PO, whereas our stated goal in this paper is to curtail the proliferation of junk cases (particularly in composed concessions) insofar as we can. We work generically in this paper, so we must approach our goal by generic means. Given that presumption, about the only generic means at our disposal are the guards we have just been manipulating, since these are about the only generic things we can soundly introduce into the operation PO conclusion which will (typically) not be there already.

In this regard, a formulation like (3.11) is very appealing, since it separates the strengthening of the conclusion of the PO from the data that is already present there. Noting that we are working in a relational framework, which will be the semantic domain for some (unstated) concrete syntax for defining systems, refinements and retrenchments, the separation in (3.11) might be conveniently reflected at (and also generated from) the syntactic level.

Moreover, while the relatively trivial computations so far do not achieve anything new in themselves, when one starts to compose the structures introduced, the varying scopes of the existential quantifiers lead to varying and non-trivial effects in the compositions, due to the inability to identify existential witnesses across conjunctions.

Thus our strategy in this paper comes down to exploring formulations of the retrenchment data, strengthened along the lines illustrated above, in tandem with additional assumptions as appropriate, and elucidating the costs and benefits for composition. We start by defining three special cases of retrenchment, defined in terms of the guards already introduced.

Definition 3.7 A retrenchment is tidy iff for all abstract operations *Op*:

$$\mathrm{pre}^{\mathrm{RetA}}{}_{Op}(u, i) \wedge \mathrm{pre}^{\mathrm{ConA}}{}_{Op}(u, i) \equiv \mathsf{false} \tag{3.12}$$

and

$$\mathrm{pre}^{\mathrm{RetC}}{}_{Op}(v, j) \wedge \mathrm{pre}^{\mathrm{ConC}}{}_{Op}(v, j) \equiv \mathsf{false} \tag{3.13}$$

Thus for a tidy retrenchment, assuming one knows the various pre- sets that figure in (3.12) and (3.13), choosing u, i in the abstract system is sufficient to determine whether any abstract transition emerging from u, i will be refining or non-refining. Furthermore, these options are obviously mutually exclusive. Similar remarks apply for v, j in the concrete system.

Lemma 3.8 Assume we are given a tidy retrenchment and a quadruple of values u, i, v, j. Then in the expression $\mathrm{pre}^{\mathrm{RetA}}{}_{Op}(u, i) \wedge \mathrm{pre}^{\mathrm{ConC}}{}_{Op}(v, j)$, the values v, j cannot instantiate the quantified variables of $\mathrm{pre}^{\mathrm{RetA}}{}_{Op}(u, i)$, and u, i cannot instantiate the quantified variables of $\mathrm{pre}^{\mathrm{ConC}}{}_{Op}(v, j)$. Similar remarks hold regarding the quantified variables in the expression $\mathrm{pre}^{\mathrm{ConA}}{}_{Op}(u, i) \wedge \mathrm{pre}^{\mathrm{RetC}}{}_{Op}(v, j)$.

Proof. Obvious. ☺

Definition 3.9 A retrenchment is neat iff for all abstract operations *Op*:

$$\mathrm{pre}^{\mathrm{Ret}}{}_{Op}(u, i, v, j) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op}(u, i, v, j) \equiv \mathsf{false} \tag{3.14}$$

This is a variation on the tidy condition. This time, the entire quadruple of values u, i, v, j is needed before we can be sure of separating refining from non-refining behaviour for any pair of abstract and concrete transitions that emerge from u, i in the abstract syatem and from v, j in the concrete one. It is therefore (obviously) a weaker condition than tidiness, as the next proposition shows.

Proposition 3.10 A tidy retrenchment is neat.

Proof. Arguing by contraposition, from the denial of neatness, i.e. $\mathrm{pre}^{\mathrm{Ret}}{}_{Op}(u, i, v, j) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op}(u, i, v, j)$, we infer $(\exists\, v, j \bullet \mathrm{pre}^{\mathrm{Ret}}{}_{Op}(u, i, v, j)) \wedge (\exists\, v, j \bullet \mathrm{pre}^{\mathrm{Con}}{}_{Op}(u, i, v, j)) \equiv \mathrm{pre}^{\mathrm{RetA}}{}_{Op}(u, i) \wedge \mathrm{pre}^{\mathrm{ConA}}{}_{Op}(u, i)$, so that tidiness is contradicted. ☺

Definition 3.11 A retrenchment is fastidious iff for all abstract operations Op:

$$\overline{G}_{Op}(u', v', o, p; i, j, u, v) \wedge \overline{C}_{Op}(u', v', o, p; i, j, u, v) \;\equiv\; \mathsf{false} \tag{3.15}$$

The fastidious condition is a further weakening, since in order to separate refining from non-refining behaviour, we now not only have to be aware of the entire quadruple of before-values u, i, v, j, but also the entire quadruple of after-values u', v', o, p too. Thus, for the same u, i, v, j, one pair of abstract and concrete transitions that emerge from u, i, v, j may be refining, and arrive at $u_1{}'$, $v_1{}'$, o_1, p_1 say, while another pair that also emerges from u, i, v, j may be non-refining, arriving at $u_2{}', v_2{}', o_2, p_2$ say.

Proposition 3.12 A neat retrenchment is fastidious.

Proof. Similar to the preceding. ☺

Proposition 3.13 For any tidy or neat retrenchment we have:

$$\mathrm{pre}^{\mathrm{Ret}}{}_{Op}(u, i, v, j) \oplus \mathrm{pre}^{\mathrm{Con}}{}_{Op}(u, i, v, j) \;\equiv\; P^{\mathrm{Def}}{}_{Op}(i, j, u, v) \tag{3.16}$$

where $\oplus$ is exclusive or.

Proof. For a neat retrenchment we have (3.14). Yet for any retrenchment we have (3.10), so the 'or' must be exclusive. Since tidy retrenchments are neat, the result follows for them too. ☺

Proposition 3.14 A default retrenchment is fastidious.

Proof. We calculate for a default retrenchment:

$$\overline{G}_{Op} \wedge \overline{C}_{Op}$$

$$\equiv$$

$$(G \wedge P_{Op} \wedge stp_{Op_{\mathrm{A}}} \wedge stp_{Op_{\mathrm{C}}} \wedge G' \wedge O_{Op}) \wedge (G \wedge P_{Op} \wedge stp_{Op_{\mathrm{A}}} \wedge stp_{Op_{\mathrm{C}}} \wedge C^{\mathrm{Def}}{}_{Op})$$

$$\equiv$$

$$(G \wedge P_{Op} \wedge stp_{Op_{\mathrm{A}}} \wedge stp_{Op_{\mathrm{C}}} \wedge G' \wedge O_{Op} \wedge \neg\, (G' \wedge O_{Op}))$$

$$\equiv$$

$$\mathsf{false} \tag{3.17}$$

☺

However there is no reason to presume that an arbitrary default retrenchment will satisfy the stronger neatness or tidiness conditions.

We recall now that a deterministic system is one for which for every operation Op, given an input i and a before-state u, there is at most one output o and after-state u' for which $stp_{Op}(u, i, u', o)$ holds. This yields the following.

Proposition 3.15 A default retrenchment between two deterministic systems is neat.

Proof. We calculate:

$$
\begin{aligned}
&\mathrm{pre}^{\mathrm{Ret}}{}_{Op}(u, i, v, j) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op}(u, i, v, j) \\
&\equiv \text{(definition)} \\
&(\exists\, u'_{\mathrm{a}}, o_{\mathrm{a}}, v'_{\mathrm{a}}, p_{\mathrm{a}} \bullet \overline{G}_{Op}(u'_{\mathrm{a}}, v'_{\mathrm{a}}, o_{\mathrm{a}}, p_{\mathrm{a}}; i, j, u, v)) \wedge \\
&\qquad (\exists\, u'_{\mathrm{b}}, o_{\mathrm{b}}, v'_{\mathrm{b}}, p_{\mathrm{b}} \bullet \overline{C}^{\mathrm{Def}}{}_{Op}(u'_{\mathrm{b}}, v'_{\mathrm{b}}, o_{\mathrm{b}}, p_{\mathrm{b}}; i, j, u, v)) \\
&\equiv \text{(instantiating } u'_{\mathrm{a}}, o_{\mathrm{a}}, v'_{\mathrm{a}}, p_{\mathrm{a}}, u'_{\mathrm{b}}, o_{\mathrm{b}}, v'_{\mathrm{b}}, p_{\mathrm{b}}) \\
&(G(u, v) \wedge P_{Op}(i, j, u, v) \wedge stp_{Op_{\mathrm{A}}}(u, i, u'_{\mathrm{a}}, o_{\mathrm{a}}) \wedge stp_{Op_{\mathrm{C}}}(v, j, v'_{\mathrm{a}}, p_{\mathrm{a}}) \wedge \\
&G(u'_{\mathrm{a}}, v'_{\mathrm{a}}) \wedge O_{Op}(o_{\mathrm{a}}, p_{\mathrm{a}}; u'_{\mathrm{a}}, v'_{\mathrm{a}}, i, j, u, v)) \wedge \\
&(G(u, v) \wedge P_{Op}(i, j, u, v) \wedge stp_{Op_{\mathrm{A}}}(u, i, u'_{\mathrm{b}}, o_{\mathrm{b}}) \wedge stp_{Op_{\mathrm{C}}}(v, j, v'_{\mathrm{b}}, p_{\mathrm{b}}) \wedge \\
&\qquad C^{\mathrm{Def}}{}_{Op}(u'_{\mathrm{b}}, v'_{\mathrm{b}}, o_{\mathrm{b}}, p_{\mathrm{b}}; i, j, u, v)) \\
&\Rightarrow \text{(determinism: } u'_{\mathrm{a}} = u'_{\mathrm{b}} = u', o_{\mathrm{a}} = o_{\mathrm{b}} = o, v'_{\mathrm{a}} = v'_{\mathrm{b}} = v', p_{\mathrm{a}} = p_{\mathrm{b}} = p) \\
&(G(u, v) \wedge P_{Op}(i, j, u, v) \wedge stp_{Op_{\mathrm{A}}}(u, i, u', o) \wedge stp_{Op_{\mathrm{C}}}(v, j, v', p) \wedge \\
&\qquad G(u', v') \wedge O_{Op}(o, p; u', v', i, j, u, v) \wedge \\
&\qquad C^{\mathrm{Def}}{}_{Op}(u', v', o, p; i, j, u, v)) \\
&\equiv \text{(definition)} \\
&(G(u, v) \wedge P_{Op}(i, j, u, v) \wedge stp_{Op_{\mathrm{A}}}(u, i, u', o) \wedge stp_{Op_{\mathrm{C}}}(v, j, v', p) \wedge \\
&\qquad G(u', v') \wedge O_{Op}(o, p; u', v', i, j, u, v) \wedge \\
&\qquad \neg\,(G(u', v') \wedge O_{Op}(o, p; u', v', i, j, u, v))) \\
&\equiv \\
&\mathsf{false}
\end{aligned}
\tag{3.18}
$$

☺

Next we recall that a relation $R : \mathsf{X} \leftrightarrow \mathsf{Y}$ is regular iff $R \mathbin{;} R^{\mathrm{T}} \mathbin{;} R = R$, where $;$ is forward relational composition and R^{T} denotes the transpose of relation R; see [Schmidt and Ströhlhein (1993), Banach (1994)]. In [Banach (1995)] it is shown that regular relations often arise in practice, which makes their properties of interest in the present context. Regular relations are also often called difunctional because any regular relation R can be equivalently characterised by the property that there are two partial functions $f : \mathsf{X} \to \mathsf{T}$ and $g : \mathsf{Y} \to \mathsf{T}$ (where T is some set) such that $f \mathbin{;} g^{-1} = R$. As an easy consequence of this, a regular relation can also be characterised by the property that its domain $\mathrm{dom}(R)$ and range $\mathrm{rng}(R)$ are partitioned into an equal number of equivalence classes, such that for any two classes $[x] \subseteq \mathrm{dom}(R)$ and $[y] \subseteq \mathrm{rng}(R)$, R is either empty from $[x]$ to $[y]$, or universal from $[x]$ to $[y]$, where the universal cases correspond to $f^{-1}(t) \times g^{-1}(t)$ when $t \in \mathsf{T}$ is in the range of both f and g. These points of T consequently set up a bijection between the equivalence classes of the domain and those of the range. Adding the complement of the domain and range respectively to the collections of equivalence classes extends this bijection by one more pair (provided both complements are nonempty, otherwise we don't get a pair), and makes every point of X and Y belong to some class or other in the relevant collection. We call these extended collections of subsets of X and Y the partitions of the domain and range types.

Regarding the regularity of any of the relations G, P_{Op}, O_{Op}, C_{Op}, of a retrenchment (or any relations formed from these), we mean regularity when these relations are viewed as relations from the relevant cartesian product of abstract data spaces to the corresponding cartesian product of concrete data spaces.

Definition 3.16 A retrenchment has regular data iff for all operations Op, the relation given by $G(u, v) \wedge P_{Op}(i, j, u, v)$, the relation given by $G(u', v') \wedge O_{Op}(o, p; u', v', i, j, u, v)$, and the relation given by $C_{Op}(u', v', o, p; i, j, u, v)$, are all regular in the sense just mentioned (where in the case of $G \wedge P_{Op}$ and of $G' \wedge O_{Op}$, we implicitly assume that G and G' are extended by appropriate universal relations on the other variables involved, in order that the overall relation has the correct signature). We write the equivalence classes of the domain and range types of these relations using the notation $[u, i]_{G\wedge P}$, $[v, j]_{G\wedge P}$, $[u', o, i, u]_{G'\wedge O}$, $[v', p, j, v]_{G'\wedge O}$, $[u', o, i, u]_C$, $[v', p, j, v]_C$.

Definition 3.17 A retrenchment respects its regular data, iff it has regular data, and the following all hold. For every abstract transition u -(i, Op_{A}, o)-› u', $[u, i]_{G\wedge P}$, $[u', o, i, u]_{G'\wedge O}$, $[u', o, i, u]_C$ all exist, and:

(1) If $(\underline{u}, \underline{i}) \in [u, i]_{G\wedge P}$ and $\underline{u}$ -($\underline{i}$, Op_{A}, $\underline{o}$)-› $\underline{u}'$ is an abstract transition, then for some (u', o), $(\underline{u}', \underline{o}, \underline{i}, \underline{u}) \in [u', o, i, u]_{G'\wedge O}$, and $(\underline{u}', \underline{o}, \underline{i}, \underline{u}) \in [u', o, i, u]_C$.

(2) If $(\underline{u}', \underline{o}, \underline{i}, \underline{u}) \in [u', o, i, u]_{G'\wedge O}$ and $\underline{u}$ -($\underline{i}$, Op_{A}, $\underline{o}$)-› $\underline{u}'$ is an abstract transition, then $(\underline{u}, \underline{i}) \in [u, i]_{G\wedge P}$.

(3) If $(\underline{u}', \underline{o}, \underline{i}, \underline{u}) \in [u', o, i, u]_C$ and $\underline{u}$ -($\underline{i}$, Op_{A}, $\underline{o}$)-› $\underline{u}'$ is an abstract transition, then $(\underline{u}, \underline{i}) \in [u, i]_{G\wedge P}$.

For every concrete transition v -(j, Op_{C}, p)-› v', $[v, j]_{G\wedge P}$, $[v', p, j, v]_{G'\wedge O}$, $[v', p, j, v]_C$ all exist, and:

(4) If $(\underline{v}, \underline{j}) \in [v, j]_{G\wedge P}$ and $\underline{v}$ -($\underline{j}$, Op_{C}, $\underline{p}$)-› $\underline{v}'$ is a concrete transition, then for some (v', p), $(\underline{v}', \underline{p}, \underline{j}, \underline{v}) \in [v', p, j, v]_{G'\wedge O}$, and $(\underline{v}', \underline{p}, \underline{j}, \underline{v}) \in [v', p, j, v]_C$.

(5) If $(\underline{v}', \underline{p}, \underline{j}, \underline{v}) \in [v', p, j, v]_{G'\wedge O}$ and $\underline{v}$ -($\underline{j}$, Op_{C}, $\underline{p}$)-› $\underline{v}'$ is a concrete transition, then $(\underline{v}, \underline{j}) \in [v, j]_{G\wedge P}$.

(6) If $(\underline{v}', \underline{p}, \underline{j}, \underline{v}) \in [v', p, j, v]_C$ and $\underline{v}$ -($\underline{j}$, Op_{C}, $\underline{p}$)-› $\underline{v}'$ is a concrete transition, then $(\underline{v}, \underline{j}) \in [v, j]_{G\wedge P}$.

Proposition 3.18 In a retrenchment which respects its regular data, the abstract and concrete transitions are related by a regular relation.

Proof. As the regular relation relating the abstract and concrete transitions we can take $G(u', v') \wedge O_{Op}(o, p; u', v', i, j, u, v) \wedge C_{Op}(u', v', o, p; i, j, u, v)$. (N.B. We do not claim that every transition is in the domain or range of this relation, even though every transition is in one of the relevant equivalence classes.) ☺

Proposition 3.19 A default retrenchment which respects its regular data is tidy.

Proof. We confirm that $\mathrm{pre}^{\mathrm{RetA}}{}_{Op}(u, i) \wedge \mathrm{pre}^{\mathrm{ConA}}{}_{Op}(u, i)$ reduces to false as required by (3.12). Instantiating the existentially quantified variables we get:

$$(G(u, v_a) \wedge P_{Op}(i, j_a, u, v_a) \wedge stp_{Op_A}(u, i, u'_a, o_a) \wedge stp_{Op_C}(v_a, j_a, v'_a, p_a) \wedge$$
$$G(u'_a, v'_a) \wedge O_{Op}(o_a, p_a; u'_a, v'_a, i, j_a, u, v_a)) \wedge$$
$$(G(u, v_b) \wedge P_{Op}(i, j_b, u, v_b) \wedge stp_{Op_A}(u, i, u'_b, o_b) \wedge stp_{Op_C}(v_b, j_b, v'_b, p_b) \wedge$$
$$C^{Def}{}_{Op}(u'_b, v'_b, o_b, p_b; i, j_b, u, v_b)) \tag{3.19}$$

Now since $stp_{Op_A}(u, i, u'_a, o_a)$ and $stp_{Op_A}(u, i, u'_b, o_b)$ are both true, and the retrenchment respects its regular data, since $C^{Def}{}_{Op}(u'_b, v'_b, o_b, p_b; i, j_b, u, v_b)$ holds, we deduce $C^{Def}{}_{Op}(u'_a, v'_b, o_a, p_b; i, j_b, u, v_b)$. Since $G \wedge P_{Op}$ is regular and $G(u, v_a) \wedge P_{Op}(i, j_a, u, v_a)$ and $G(u, v_b) \wedge P_{Op}(i, j_b, u, v_b)$ are both true, $[v_a, j_a]_{G\wedge P} = [v_b, j_b]_{G\wedge P}$. So since the retrenchment respects its regular data, since $stp_{Op_C}(v_a, j_a, v'_a, p_a)$ and $stp_{Op_C}(v_b, j_b, v'_b, p_b)$ both hold, $C^{Def}{}_{Op}(u'_a, v'_b, o_a, p_b; i, j_b, u, v_b)$ implies $C^{Def}{}_{Op}(u'_a, v'_a, o_a, p_a; i, j_a, u, v_a)$. But the latter implies $\neg (G(u'_a, v'_a) \wedge O_{Op}(o_a, p_a; u'_a, v'_a, i, j_a, u, v_a))$ which contradicts the $G(u'_a, v'_a) \wedge O_{Op}(o_a, p_a; u'_a, v'_a, i, j_a, u, v_a)$ in (3.19), giving false. The calculation for (3.13) is entirely analogous. ☺

Since any tidy retrenchment is neat (Proposition 3.10), we get:

Corollary 3.20 A default retrenchment which respects its regular data, is neat.

We close this section by applying the material just developed to the comparison of default retrenchments with arbitrary bespoke retrenchments.

Suppose, for a given application with retrieve relation $G(u, v)$, that $P^{\circ}{}_{Op}(i, j, u, v)$ is a 'minimal' within relation. A minimal within relation will typically express no more than how the abstract and concrete input spaces are related (but allowing for the possibility that this relationship may depend on the state spaces). Note that although the same collection of data spaces may support a variety of different relationships, allowing for more than one possible 'minimal' within relation, in the context of a given application, it is unlikely that more than one of them will be perceived as 'natural'. Thus the choice of $P^{\circ}{}_{Op}$ is a meta level issue.[6] Let $O^{\circ}{}_{Op}(o, p; u', v', i, j, u, v)$ be a correspondingly minimal output relation. Let $P^{Def}{}_{Op}(i, j, u, v)$ be the default within relation manufactured from $P^{\circ}{}_{Op}$ by using $P^{\circ}{}_{Op}$ instead of P in (2.12), and let $C^{Def}{}_{Op}(u', v', o, p; i, j, u, v)$ be the corresponding default concedes relation.

Assuming the same retrieve relation $G(u, v)$, suppose that we also have a bespoke retrenchment characterised by data $\{P^{Bes}{}_{Op}, O^{Bes}{}_{Op}, C^{Bes}{}_{Op} \mid Op \in \mathsf{Ops}_A\}$, and let $P^{BC}{}_{Op}(i, j, u, v)$ be given by:

$$P^{BC}{}_{Op}(i, j, u, v) \equiv$$
$$(G(u, v) \wedge P^{Bes}{}_{Op}(i, j, u, v) \wedge$$
$$(\exists\, u', o, v', p \bullet stp_{Op_A}(u, i, u', o) \wedge stp_{Op_C}(v, j, v', p))) \tag{3.20}$$

i.e. the analogous construction to $P^{Def}{}_{Op}$. Then we may make the meta level assumption that for any such $P^{Bes}{}_{Op}$:

$$P^{BC}{}_{Op}(i, j, u, v) \ \Rightarrow\ P^{Def}{}_{Op}(i, j, u, v) \tag{3.21}$$

Note that $P^{BC}{}_{Op}$ and $P^{Def}{}_{Op}$, which include the guards discussed above, provide a better basis for comparison (among possible within relations) than $P^{\circ}{}_{Op}$ and $P^{Bes}{}_{Op}$ alone, since as noted already above, the application developer is liable to choose the

6. The universal relation given by true is always available and is certainly minimal (in the sense of being the weakest possible) but is usually unhelpful.

simplest form for bespoke retrenchment data, focusing only on what is considered most pertinent to the development step.

Proposition 3.21 Let $\overline{C}^{\mathrm{Bes}}{}_{Op}$ be the concedes closure for a bespoke retrenchment, and let $\overline{G}^{\mathrm{Def}}{}_{Op}$ and $\overline{C}^{\mathrm{Def}}{}_{Op}$ be the retrieve and concedes closures for the default retrenchment. Then assuming (3.21):

$$\overline{C}^{\mathrm{Bes}}{}_{Op}(u', v', o, p; i, j, u, v) \wedge \neg\overline{G}^{\mathrm{Def}}{}_{Op}(u', v', o, p; i, j, u, v) \Rightarrow \overline{C}^{\mathrm{Def}}{}_{Op}(u', v', o, p; i, j, u, v) \tag{3.22}$$

Proof. Suppressing the variable names we calculate as follows:

$$\overline{C}^{\mathrm{Bes}}{}_{Op} \wedge \neg\overline{G}^{\mathrm{Def}}{}_{Op}$$

$$\equiv$$

$$G \wedge P^{\mathrm{Bes}}{}_{Op} \wedge stp_{Op_{\mathrm{A}}} \wedge stp_{Op_{\mathrm{C}}} \wedge C^{\mathrm{Bes}}{}_{Op} \wedge \neg(G \wedge P^{\circ}{}_{Op} \wedge stp_{Op_{\mathrm{A}}} \wedge stp_{Op_{\mathrm{C}}} \wedge G' \wedge O^{\circ}{}_{Op})$$

$$\Rightarrow \text{ ((3.21), weakening)}$$

$$G \wedge P^{\circ}{}_{Op} \wedge stp_{Op_{\mathrm{A}}} \wedge stp_{Op_{\mathrm{C}}} \wedge \neg(G \wedge P^{\circ}{}_{Op} \wedge stp_{Op_{\mathrm{A}}} \wedge stp_{Op_{\mathrm{C}}} \wedge G' \wedge O^{\circ}{}_{Op})$$

$$\equiv$$

$$G \wedge P^{\circ}{}_{Op} \wedge stp_{Op_{\mathrm{A}}} \wedge stp_{Op_{\mathrm{C}}} \wedge \neg(G' \wedge O^{\circ}{}_{Op})$$

$$\equiv$$

$$\overline{C}^{\mathrm{Def}}{}_{Op} \tag{3.23}$$

☺

Thus, excluding that which can be subsumed by $\neg\overline{G}^{\mathrm{Def}}{}_{Op}$ in the hypotheses, we see that the default concedes relation is weaker than a bespoke one. In Section 2.3 we intuited that the relationship was the other way round. The truth is that while a bespoke concession will typically not include all the possible guards that the concedes closure contains, and so will be weaker in that sense, the default concession includes the guards, tending to make it stronger, but also includes the negation of the 'minimal' output relation. Since the latter is typically weak itself, this tends to make the default concession stronger. This prevents the relationship between default and bespoke concessions being completely straightforward.

The following corollary shows us that the weakest of our special classes of retrenchment permits us to illuminate the relationship between default and bespoke concessions another way.

Corollary 3.22 Under the assumptions of Proposition 3.21, if the bespoke retrenchment is fastidious (or neat or tidy), we have:

$$\overline{C}^{\mathrm{Bes}}{}_{Op}(u', v', o, p; i, j, u, v) \Rightarrow \overline{C}^{\mathrm{Def}}{}_{Op}(u', v', o, p; i, j, u, v) \vee \neg\overline{G}^{\mathrm{Bes}}{}_{Op}(u', v', o, p; i, j, u, v) \tag{3.24}$$

Proof. Suppressing the variable names we calculate as follows:

$$\overline{C}^{\mathrm{Bes}}{}_{Op}$$

$$\Rightarrow \text{ (fastidiousness)}$$

$$\overline{C}^{\text{Bes}}{}_{Op} \wedge \neg \overline{G}^{\text{Bes}}{}_{Op}$$
$$\Rightarrow \text{ (tautology)}$$
$$\overline{C}^{\text{Def}}{}_{Op} \vee \neg \overline{G}^{\text{Bes}}{}_{Op} \tag{3.25}$$

☺

4 Stronger Compositions of Retrenchments

Suppose that we are given three systems, a top level system with data u_0, i_0, u'_0, o_0, and transition relation $stp_{Op,0}$, an intermediate system with data u_1, i_1, u'_1, o_1, and transition relation $stp_{Op,1}$, and a lowest level system with data u_2, i_2, u'_2, o_2, and transition relation $stp_{Op,2}$. Let there be a retrenchment from top level to intermediate system characterised by relations $G_1(u_0, u_1)$, $P_{Op,1}(i_0, i_1, u_0, u_1)$, $O_{Op,1}(o_0, o_1; u'_0, u'_1, i_0, i_1, u_0, u_1)$, $C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1)$, and a retrenchment from intermediate to lowest level system characterised by relations $G_2(u_1, u_2)$, $P_{Op,2}(i_1, i_2, u_1, u_2)$, $O_{Op,2}(o_1, o_2; u'_1, u'_2, i_1, i_2, u_1, u_2)$, $C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2)$.

In a similar manner we define '1' subscripted and '2' subscripted versions of the relations introduced in Section 3, i.e. $\overline{G}_{Op,1}$, $\overline{C}_{Op,1}$, $\text{pre}^{\text{Ret}}{}_{Op,1}$, $\text{pre}^{\text{Con}}{}_{Op,1}$, $\text{pre}^{\text{RetA}}{}_{Op,1}$, $\text{pre}^{\text{RetC}}{}_{Op,1}$, $\text{pre}^{\text{ConA}}{}_{Op,1}$, $\text{pre}^{\text{ConC}}{}_{Op,1}$; $\overline{G}_{Op,2}$, $\overline{C}_{Op,2}$, $\text{pre}^{\text{Ret}}{}_{Op,2}$, $\text{pre}^{\text{Con}}{}_{Op,2}$, $\text{pre}^{\text{RetA}}{}_{Op,2}$, $\text{pre}^{\text{RetC}}{}_{Op,2}$, $\text{pre}^{\text{ConA}}{}_{Op,2}$, $\text{pre}^{\text{ConC}}{}_{Op,2}$.

With these in place we can derive strengthenings of the composition of retrenchments that follows from Proposition 3.2 and Proposition 3.6. The first theorem tackles this in terms of the retrieve and concedes closures. We include a detailed proof since it establishes a pattern used extensively in many similar results below.

Theorem 4.1 Two retrenchments compose to give a single retrenchment which validates the (modified) operation PO:

$$\begin{aligned}&G_{(1,2)}(u_0, u_2) \wedge P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \;\Rightarrow\\ &\quad(\exists\, u'_0, o_0 \bullet \overline{G}_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2) \vee\\ &\qquad \overline{C}_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2))\end{aligned} \tag{4.1}$$

where:

$$\begin{aligned}&\overline{G}_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2) \;\equiv\\ &\quad(G_{(1,2)}(u_0, u_2) \wedge P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \wedge\\ &\qquad stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge\\ &\qquad G_{(1,2)}(u'_0, u'_2) \wedge O_{Op,(1,2)}(o_0, o_2; u'_0, u'_2, i_0, i_2, u_0, u_2))\end{aligned} \tag{4.2}$$

$$\begin{aligned}&\overline{C}_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2) \;\equiv\\ &\quad(G_{(1,2)}(u_0, u_2) \wedge P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \wedge\\ &\qquad stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge\\ &\qquad C_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2))\end{aligned} \tag{4.3}$$

in which $G_{(1,2)}$, $P_{Op,(1,2)}$, $O_{Op,(1,2)}$, $C_{Op,(1,2)}$ are given by (2.8)-(2.11).

Proof. To show that we have a retrenchment, we must show that the POs for the composed retrenchment follow from the POs for the individual ones. The initialisation PO follows by composing the individual initialisation POs. Thus given a u'_2 satisfying $Init_2(u'_2)$, from $Init_2(u'_2) \Rightarrow (\exists\, u'_1 \bullet Init_1(u'_1) \wedge G_2(u'_1, u'_2))$ we deduce a u'_1 satisfying $Init_1(u'_1)$ (and $G_2(u'_1, u'_2)$). Repeating the argument for this u'_1, we deduce a u'_0 satisfying $Init_0(u'_0)$ and $G_1(u'_0, u'_1)$. So altogether we get $Init_2(u'_2) \Rightarrow (\exists\, u'_0 \bullet Init_0(u'_0) \wedge G_{(1,2)}(u'_0, u'_2))$ when we existentially quantify over u'_1.

For the operation PO, we are required to establish (4.1) with the component data defined above. We assume the antecedents, so that we have $G_{(1,2)} \wedge P_{Op,(1,2)}$. This gives us existential witnesses u_1 and i_1 for (2.8) and (2.9), taking the u_1 witness to be common. Since we have $G_2 \wedge P_{Op,2} \wedge stp_{Op,2}$ we use the operation PO for the intermediate to lowest level retrenchment to infer for the intermediate system $(\exists\, u'_1, o_1 \bullet stp_{Op,1} \wedge ((G_2 \wedge O_{Op,2}) \vee C_{Op,2}))$. For the u_1, i_1, u'_1, o_1 that we have now derived, and using $G_1 \wedge P_{Op,1} \wedge stp_{Op,1}$ all of which have been established, we apply the operation PO for the top level to intermediate retrenchment to deduce $(\exists\, u'_0, o_0 \bullet stp_{Op,0} \wedge ((G_1 \wedge O_{Op,1}) \vee C_{Op,1}))$ for the top level system.

Thus given $G_{(1,2)} \wedge P_{Op,(1,2)} \wedge stp_{Op,2}$ we have deduced u'_0 and o_0 such that $stp_{Op,0}$ and $((G_1 \wedge O_{Op,1}) \vee C_{Op,1}) \wedge ((G_2 \wedge O_{Op,2}) \vee C_{Op,2})$ hold, all witnessed by a common intermediate transition u_1 -(i_1, Op_{I}, o_1)-› u'_1. The distributive law now yields:

$$
\begin{aligned}
&(G'_1 \wedge O_{Op,1} \wedge G'_2 \wedge O_{Op,2}) \vee \\
&((G'_1 \wedge O_{Op,1} \wedge C_{Op,2}) \vee (C_{Op,1} \wedge G'_2 \wedge O_{Op,2}) \vee (C_{Op,1} \wedge C_{Op,2})) \qquad (4.4)
\end{aligned}
$$

all conjoined with $G_{(1,2)} \wedge P_{Op,(1,2)} \wedge stp_{Op,2} \wedge stp_{Op,0}$. When the latter is distributed into the first disjunct we obtain $G_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2)$ after pushing the existential quantification over u_1, i_1, u'_1, o_1 over the first '$\vee$' in (4.4), thus discharging (4.1). Likewise when $G_{(1,2)} \wedge P_{Op,(1,2)} \wedge stp_{Op,2} \wedge stp_{Op,0}$ is distributed into the second collection of disjuncts in (4.4) we obtain $C_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2)$ after dealing with the u_1, i_1, u'_1, o_1, quantification, also discharging (4.1). ☺

The next theorem tackles the same problem, but by keeping separate the strengthening guards from the original retrenchment data.

Theorem 4.2 Two retrenchments compose to give a single retrenchment given by the data:[7]

$$G_{(1,2)}(u_0, u_2) \equiv [\exists\, u_1 \bullet G_1(u_0, u_1) \wedge G_2(u_1, u_2)] \qquad (4.5)$$

$$
\begin{aligned}
&P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \equiv \\
&\quad [\exists\, u_1, i_1 \bullet G_1(u_0, u_1) \wedge G_2(u_1, u_2) \wedge \\
&\qquad P_{Op,1}(i_0, i_1, u_0, u_1) \wedge P_{Op,2}(i_1, i_2, u_1, u_2)] \qquad (4.6)
\end{aligned}
$$

$$
\begin{aligned}
&O_{Op,(1,2)}(o_0, o_2; u'_0, u'_2, i_0, i_2, u_0, u_2) \equiv \\
&\qquad [\exists\, u'_1, o_1, u_1, i_1 \bullet O_{Op,1}(o_0, o_1; \ldots) \wedge O_{Op,2}(o_1, o_2; \ldots) \wedge \\
&\qquad\quad \mathrm{pre^{Ret}}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre^{Ret}}_{Op,2}(u_1, i_1, u_2, i_2)] \qquad (4.7)
\end{aligned}
$$

$$
\begin{aligned}
&C_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2) \equiv \\
&\quad [\exists\, u'_1, o_1, u_1, i_1 \bullet \\
&\qquad (G_1(u'_0, u'_1) \wedge O_{Op,1}(o_0, o_1; u'_0, u'_1, i_0, i_1, u_0, u_1) \wedge \\
&\qquad\quad C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2) \wedge \\
&\qquad\quad \mathrm{pre^{Ret}}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre^{Con}}_{Op,2}(u_1, i_1, u_2, i_2)) \vee \\
&\qquad (C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge
\end{aligned}
$$

7. Note that (4.5) and (4.6) are just (2.8) and (2.9), whereas (4.7) and (4.8) strengthen (2.10) and (2.11) by the inclusion of the various 'pre-' guards.

$$G_2(u'_1, u'_2) \wedge O_{Op,2}(o_1, o_2; u'_1, u'_2, i_1, i_2, u_1, u_2) \wedge$$
$$\mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2)) \vee$$
$$(C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge$$
$$C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2) \wedge$$
$$\mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2))] \tag{4.8}$$

Proof Sketch. Initialisation is routine. For the operation PO we proceed in the usual way, establishing an intermediate transition u_1 -(i_1, Op_{I}, o_1)-› u'_1 which witnesses the standard collection of facts. These can be packaged in a slightly different way to Theorem 4.1 to get the conclusion desired here. ☺

Readers can easily convince themselves that using Proposition 3.2, a proof combining elements of both of the above theorems can establish a version of Theorem 4.1 that uses (4.7) and (4.8) instead of (2.10) and (2.11).

We also note that in Theorem 4.2, although we are able to strengthen the composed output and concedes relation in the manner expected from Proposition 3.6, a similar strengthening of the retrieve relation cannot be carried through as the retrieve relation itself does not admit all of the required variables. This is in line with the fact that the retrieve relation also appears in the antecedents of the operation PO, where the strengthening we are considering does not make sense. Thus we must distinguish carefully between strengthening what is said in the operation PO itself, as in (3.3) and (3.11), and merely strengthening the data which enter into the conventional operation PO, as in (4.7) and (4.8), and for which there are in principle fewer opportunities.

Now we turn to the tidy, neat, and fastidious retrenchments. Under suitable assumptions we will be able to compose these kinds of retrenchment in a more incisive manner than in Section 2.

Definition 4.3 We say that two adjacent retrenchments like the above, which are both tidy, are compatibly tidy iff for all abstract operations *Op*:

$$\mathrm{pre}^{\mathrm{RetA}}{}_{Op,2}(u_1, i_1) \Rightarrow \mathrm{pre}^{\mathrm{RetC}}{}_{Op,1}(u_1, i_1) \tag{4.9}$$

and

$$\mathrm{pre}^{\mathrm{ConA}}{}_{Op,2}(u_1, i_1) \Rightarrow \mathrm{pre}^{\mathrm{ConC}}{}_{Op,1}(u_1, i_1) \tag{4.10}$$

hold for the intermediate system.

Theorem 4.4 Two compatibly tidy retrenchments compose to give a single retrenchment given by the data:[8]

$$G_{(1,2)}(u_0, u_2) \equiv [\exists\, u_1 \bullet G_1(u_0, u_1) \wedge G_2(u_1, u_2)] \tag{4.11}$$

$$P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \equiv$$
$$[\exists\, u_1, i_1 \bullet G_1(u_0, u_1) \wedge G_2(u_1, u_2) \wedge$$
$$P_{Op,1}(i_0, i_1, u_0, u_1) \wedge P_{Op,2}(i_1, i_2, u_1, u_2)] \tag{4.12}$$

$$O_{Op,(1,2)}(o_0, o_2; u'_0, u'_2, i_0, i_2, u_0, u_2) \equiv$$
$$[\exists\, u'_1, o_1, u_1, i_1 \bullet O_{Op,1}(o_0, o_1; u'_0, u'_1, i_0, i_1, u_0, u_1) \wedge$$
$$O_{Op,2}(o_1, o_2; u'_1, u'_2, i_1, i_2, u_1, u_2)] \tag{4.13}$$

8. Note that (4.11)-(4.13) are just (2.8)-(2.10), whereas (4.14) strengthens (2.11) considerably.

$$C_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2) \equiv$$
$$[\exists\, u'_1, o_1, u_1, i_1 \bullet C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge$$
$$C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2)] \tag{4.14}$$

Proof Sketch. The first part of the proof runs as usual. Having established the usual menagerie of facts, we then exploit the tidiness and compatible tidiness assumptions to argue that either all the retrieves facts hold and none of the concedes facts hold, thus establishing $G_{(1,2)} \wedge O_{Op,(1,2)}$, or the converse, establishing $C_{Op,(1,2)}$. In particular, the mixed cases in (2.11) cannot arise. ☺

The structure of the above result is very appealing. The data that specifies the combined retrenchment is built in an especially simple way from the component data, and is strictly simpler than that for compositions of arbitrary retrenchments. As we weaken the separation between retrieve-relation-re-establishing behaviour and concedes-relation-establishing behaviour, this simplicity degrades, as the following results suggest.

Theorem 4.5 Two neat retrenchments compose to give a single retrenchment such that given an intermediate level before-state and input (u_1, i_1), for any intermediate transition issuing from (u_1, i_1) that witnesses the composed operation PO:

$$G_{(1,2)}(u_0, u_2) \wedge P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \Rightarrow$$
$$(\exists\, u'_0, o_0 \bullet stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge$$
$$((G_{(1,2)}(u'_0, u'_2) \wedge O_{Op,(1,2)}(o_0, o_2; u'_0, u'_2, i_0, i_2, u_0, u_2)) \vee$$
$$C_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2))) \tag{4.15}$$

with $G_{(1,2)}$, $P_{Op,(1,2)}$, $O_{Op,(1,2)}$ and $C_{Op,(1,2)}$ given by (4.5)-(4.8), at most one of:

(1) $(G'_{(1,2)} \wedge O_{Op,(1,2)})$
(2) $(G'_1 \wedge O_{Op,1} \wedge C_{Op,2})$ from $C_{Op,(1,2)}$
(3) $(C_{Op,1} \wedge G'_2 \wedge O_{Op,2})$ from $C_{Op,(1,2)}$
(4) $(C_{Op,1} \wedge C_{Op,2})$ from $C_{Op,(1,2)}$ (4.16)

is true, the choice of which is true being dependent solely on (u_1, i_1).

Proof Sketch. Starting from Theorem 4.2, assuming that more than one of (1)-(4) from (4.16) is true, leads to a contradiction of the neatness hypothesis. ☺

Corollary 4.6 Two neat retrenchments that further satisfy:

$$\mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2) \equiv \mathsf{false} \tag{4.17}$$

compose to give a single retrenchment given by (4.11)-(4.13) and:

$$C_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2) \equiv$$
$$[\exists\, u'_1, o_1, u_1, i_1 \bullet$$
$$(G_1(u'_0, u'_1) \wedge O_{Op,1}(o_0, o_1; u'_0, u'_1, i_0, i_1, u_0, u_1) \wedge$$
$$C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2) \wedge$$
$$\mathrm{pre}^{\mathrm{Ret}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2)) \vee$$
$$(C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge$$
$$G_2(u'_1, u'_2) \wedge O_{Op,2}(o_1, o_2; u'_1, u'_2, i_1, i_2, u_1, u_2) \wedge$$
$$\mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2))] \tag{4.18}$$

Of course there are similar corollaries when other $\text{pre}^{\text{Con}}{}_{Op}/\text{pre}^{\text{Ret}}{}_{Op}$ combinations reduce to false.

Theorem 4.7 Two fastidious retrenchments compose to give a single retrenchment such that for any intermediate transition that witnesses the composed operation PO:

$$\begin{aligned}
&G_{(1,2)}(u_0, u_2) \wedge P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \Rightarrow \\
&\quad (\exists\, u'_0, o_0 \bullet stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge \\
&\qquad ((G_{(1,2)}(u'_0, u'_2) \wedge O_{Op,(1,2)}(o_0, o_2; u'_0, u'_2, i_0, i_2, u_0, u_2)) \vee \\
&\qquad C_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2))) \qquad (4.19)
\end{aligned}$$

with $G_{(1,2)}$, $P_{Op,(1,2)}$, $O_{Op,(1,2)}$ and $C_{Op,(1,2)}$ given by (4.5)-(4.8), at most one of:

(1) $(G'_{(1,2)} \wedge O_{Op,(1,2)})$
(2) $(G'_1 \wedge O_{Op,1} \wedge C_{Op,2})$ from $C_{Op,(1,2)}$
(3) $(C_{Op,1} \wedge G'_2 \wedge O_{Op,2})$ from $C_{Op,(1,2)}$
(4) $(C_{Op,1} \wedge C_{Op,2})$ from $C_{Op,(1,2)}$ (4.20)

is true.

Proof. This is similar to Theorem 4.5 except that the choice between (1)-(4) depends on the individual intermediate transition, and not on a set of them issuing from a common before-state and input. ☺

Note how the increasingly delicate conditions of tidiness, neatness, and fastidiousness have decreasingly visible effects on the syntactic appearance of the composition law for concedes relations. For compatibly tidy retrenchments, we get a dramatic simplification of the composition law; for neat retrenchments, we get at best a strengthening of the individual alternatives by what are effectively additional input guards that apply anyway to any retrenchment, but that are strengthened by a mutual exclusion condition; for fastidious retrenchments the same applies but the mutual exclusion condition is more finegrained. Since the conditions weaken from tidiness onwards, it is clear that all conclusions derived for later systems are applicable to systems satisfying earlier restrictions.

5 Composition Closure and Associativity

The results of the previous section are not enough to give closure of the composition notions, let alone associativity, for all the various strengthened notions of retrenchment introduced earlier.

Counterexample 5.1 Fig. 1 shows a situation in which in all three systems, there are no inputs or outputs (thus the output relations are defined by true, and the within relations coincide with the retrieve relations on the before-state pairs). There are no other points in the state spaces other than the ones shown, and no transitions other than the ones shown either. (N.B. The diamond states and dashed transitions and relations are only present to ensure that the various retrenchment operation POs are satisfied in all necessary cases.) Both retrieve relations consist of just the pairs illustrated, and the concedes relations are focused on just the pairs of after-states indicated (being universal in the before-states). It is easy to check that the two retrenchments are both tidy; therefore they are also neat and fastidious. The composition of the two retrenchments is not fastidious though, because it is clear that the $\overline{G}_{Op,(1,2)}$ and

$\overline{C}_{Op,(1,2)}$ conditions are simultaneously verified for the pair of solid transitions shown. The composition is therefore also neither neat nor tidy.

Counterexample 5.2 Fig. 2 shows another source of trouble. With the same conventions as in Counterexample 5.1, both the upper and lower retrenchments are fastidious and neat (though not tidy). However although the intermediate after-state values referred to by the component retrieve relations differ from those referred to by the component concedes relations, when the retrenchments are composed, we find that fastidiousness fails (and therefore so does neatness and tidiness) because as in the previous case, the $\overline{G}_{Op,(1,2)}$ and $\overline{C}_{Op,(1,2)}$ conditions are simultaneously verified for the topmost and lowest transitions.

We move towards compositionality and thence to associativity by precluding situations such as these. However the conditions we come up with for compositionality will typically be sufficient rather than necessary, since there will always be situations such as the 'duelling yardbrushes' scenario depicted in Fig. 3, in which although there is scope for the 'dangling' G and C tuples to fuse to form a counterexample of the kind shown in Fig. 1 or Fig. 2, nevertheless the individual tines of the two yardbrushes never actually meet point to point in the needed way, and the composition remains problem free. Such situations remain outside the remit of conditions that can be expressed purely in terms of the intrinsic properties of the component systems, since they crucially depend on joint properties of the combination.

We tackle the various strengthenings in roughly increasing order of difficulty.

Definition 5.3 We call a retrenchment specifically closed iff the following four properties are satisfied:

$$\begin{aligned}&G(u,v) \wedge P_{Op}(i,j,u,v) \wedge G(u',v') \wedge O_{Op}(o,p;\underline{u}',\underline{v}',\underline{i},\underline{j},\underline{u},\underline{v}) \wedge \\ &stp_{Op_C}(v,j,v',p) \Rightarrow \\ &\quad (u=\underline{u}) \wedge (i=\underline{i}) \wedge (v=\underline{v}) \wedge (j=\underline{j}) \wedge (u'=\underline{u}') \wedge (v'=\underline{v}') \wedge \\ &\quad stp_{Op_A}(u,i,u',o)\end{aligned} \tag{5.1}$$

$$\begin{aligned}&G(u,v) \wedge P_{Op}(i,j,u,v) \wedge G(u',v') \wedge O_{Op}(o,p;\underline{u}',\underline{v}',\underline{i},\underline{j},\underline{u},\underline{v}) \wedge \\ &stp_{Op_A}(u,i,u',o) \Rightarrow \\ &\quad (u=\underline{u}) \wedge (i=\underline{i}) \wedge (v=\underline{v}) \wedge (j=\underline{j}) \wedge (u'=\underline{u}') \wedge (v'=\underline{v}') \wedge \\ &\quad stp_{Op_C}(v,j,v',p)\end{aligned} \tag{5.2}$$

$$\begin{aligned}&G(u,v) \wedge P_{Op}(i,j,u,v) \wedge C_{Op}(u',v',o,p;\underline{i},\underline{j},\underline{u},\underline{v}) \wedge \\ &stp_{Op_C}(v,j,v',p) \Rightarrow \\ &\quad (u=\underline{u}) \wedge (i=\underline{i}) \wedge (v=\underline{v}) \wedge (j=\underline{j}) \wedge stp_{Op_A}(u,i,u',o)\end{aligned} \tag{5.3}$$

$$\begin{aligned}&G(u,v) \wedge P_{Op}(i,j,u,v) \wedge C_{Op}(u',v',o,p;\underline{i},\underline{j},\underline{u},\underline{v}) \wedge \\ &stp_{Op_A}(u,i,u',o) \Rightarrow \\ &\quad (u=\underline{u}) \wedge (i=\underline{i}) \wedge (v=\underline{v}) \wedge (j=\underline{j}) \wedge stp_{Op_C}(v,j,v',p)\end{aligned} \tag{5.4}$$

Definition 5.4 We call a retrenchment generally closed iff the following four properties are satisfied:

$$\begin{aligned}&G(u,v) \wedge P_{Op}(i,j,u,v) \wedge G(u',v') \wedge O_{Op}(o,p;\underline{u}',\underline{v}',\underline{i},\underline{j},\underline{u},\underline{v}) \wedge \\ &stp_{Op_C}(v,j,v',p) \Rightarrow \\ &\quad stp_{Op_A}(u,i,u',o) \wedge (\forall\, \underline{u}',\underline{v}',\underline{i},\underline{j},\underline{u},\underline{v} \bullet O_{Op}(o,p;\underline{u}',\underline{v}',\underline{i},\underline{j},\underline{u},\underline{v}))\end{aligned} \tag{5.5}$$

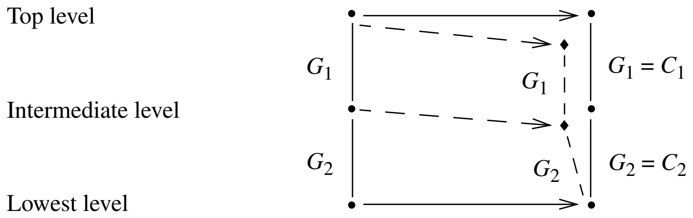

Fig. 1 A non-fastidious composition of two tidy retrenchments.

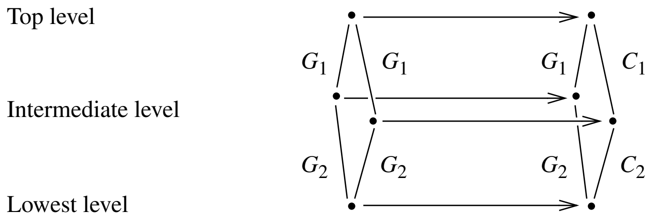

Fig. 2 A non-fastidious composition of two fastidious (and neat) retrenchments.

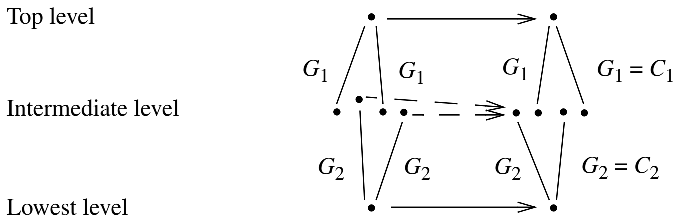

Fig. 3 A problem free composition.

$$G(u, v) \wedge P_{Op}(i, j, u, v) \wedge G(u', v') \wedge O_{Op}(o, p; \underline{u}', \underline{v}', \underline{i}, \underline{j}, \underline{u}, \underline{v}) \wedge$$
$$stp_{Op_A}(u, i, u', o) \Rightarrow$$
$$\qquad stp_{Op_C}(v, j, v', p) \wedge (\forall\ \underline{u}', \underline{v}', \underline{i}, \underline{j}, \underline{u}, \underline{v} \bullet O_{Op}(o, p; \underline{u}', \underline{v}', \underline{i}, \underline{j}, \underline{u}, \underline{v})) \tag{5.6}$$

$$G(u, v) \wedge P_{Op}(i, j, u, v) \wedge C_{Op}(u', v', o, p; \underline{i}, \underline{j}, \underline{u}, \underline{v}) \wedge$$
$$stp_{Op_C}(v, j, v', p) \Rightarrow$$
$$\qquad stp_{Op_A}(u, i, u', o) \wedge (\forall\ \underline{i}, \underline{j}, \underline{u}, \underline{v} \bullet C_{Op}(u', v', o, p; \underline{i}, \underline{j}, \underline{u}, \underline{v})) \tag{5.7}$$

$$G(u, v) \wedge P_{Op}(i, j, u, v) \wedge C_{Op}(u', v', o, p; \underline{i}, \underline{j}, \underline{u}, \underline{v}) \wedge$$
$$stp_{Op_A}(u, i, u', o) \Rightarrow$$
$$\qquad stp_{Op_C}(v, j, v', p) \wedge (\forall\ \underline{i}, \underline{j}, \underline{u}, \underline{v} \bullet C_{Op}(u', v', o, p; \underline{i}, \underline{j}, \underline{u}, \underline{v})) \tag{5.8}$$

The closedness criteria ensure that transitions exist whenever the attendant assembly of clauses leads us to hope they might do. The specific criteria ensure that the output and concedes relations cannot refer to spurious before-states and inputs, while the general criteria apply when the the output and concedes relations are independent of the before-states and inputs, as is so often the case.

These retrenchment closedness criteria compose well as is shown next.

Theorem 5.5 With the current notations, and using the standard composed retrenchment data (2.8)-(2.11), the composition of two specifically closed retrenchments is specifically closed.

Proof Sketch. A series of straightforward calculations from the hypotheses. ☺

Theorem 5.6 With the current notations, and using the standard composed retrenchment data (2.8)-(2.11), the composition of two generally closed retrenchments is generally closed.

Proof Sketch. A series of straightforward calculations from the hypotheses. ☺

Note that for these two theorems, since we obtained the desired conclusions using the standard composition of retrenchment data, they will also hold without further ado for the various stronger methods of composition that were considered in Section 4.

Theorem 5.7 With the notations of Theorem 4.4, two compatibly tidy retrenchments which are moreover either specifically or generally closed, compose to give a single tidy resp. specifically or generally closed retrenchment given by (4.11)-(4.14).

Proof Sketch. Theorem 4.4 gives us a retrenchment, and Theorem 5.5 and Theorem 5.6 tell us that it is specifically or generally closed respectively, so it remains to show tidiness. For this we deny one or other of (3.12), (3.13), instantiate the intermediate existentially quantified variables, use the closedness criteria to appropriately identify these existential witnesses, and thence derive a contradiction. ☺

Theorem 5.8 With the assumptions of Theorem 5.7, the composition of compatibly tidy specifically or generally closed retrenchments is associative.

Proof Sketch. We need first to check that in a sequence of three tidy specifically or generally closed retrenchments, in which adjacent pairs are compatibly tidy, the composition of two adjacent ones remains compatibly tidy with the third. For this we take each of the compatible tidiness criteria, for the binary composition of compatibly tidy retrenchments and the third tidy retrenchment, and show that it holds, which is a relatively straightforward exercise.

We then check that for either association order, the expressions obtained for the composed retrieve, within, output, and concedes relations are the obvious extrapolations of (4.11)-(4.14) to three components, and are symmetrical in all three of them, for example the retrieve relation:

$$
\begin{aligned}
& G_{(1,(2,3))}(u_0, u_3) \equiv (\exists\, u_1 \bullet G_1(u_0, u_1) \wedge G_{(2,3)}(u_1, u_3)) \\
& \equiv (\exists\, u_1 \bullet G_1(u_0, u_1) \wedge (\exists\, u_2 \bullet G_2(u_1, u_2) \wedge G_3(u_2, u_3))) \equiv \ldots \\
& \equiv G_{((1,2),3)}(u_0, u_3)
\end{aligned} \tag{5.9}
$$

This is sufficient. ☺

We turn our attention to neat retrenchments.

Theorem 5.9 With the notations of Theorem 4.5, two neat retrenchments which are moreover either specifically or generally closed, compose to give a single neat resp. specifically or generally closed retrenchment with data given by (4.5)-(4.8).

Proof Sketch. Theorem 4.5 tells us we have a retrenchment, and Theorem 5.5 and Theorem 5.6 tell us that the resulting retrenchment is specifically or generally closed respectively, so it remains to show neatness. For this we deny (3.14), instantiate the intermediate existentially quantified variables, and use the closedness criteria to appropriately identify these existential witnesses. The result is a conjunction, one of whose conjuncts is a disjunction. Using the distributive law on this brings together conjunctions of pre- terms that contradict the assumed neatness of the original two retrenchments, yielding a contradiction. ☺

Before going on to consider the associativity of neat retrenchments we have some results that hold without the neatness assumption, in the spirit of Theorem 4.1 and Theorem 4.2. The fact that these results do not hold without something resembling the closedness and determinism assumptions, is a reflection of precisely the kind of situations discussed in the counterexamples at the beginning of this section.

Proposition 5.10 For a composition of two specifically or generally closed retrenchments between deterministic systems we have:

$$
\begin{aligned}
& \mathrm{pre}^{\mathrm{Ret}}{}_{Op,(1,2)}(u_0, i_0, u_2, i_2) \equiv \\
& \qquad (\exists\, u_1, i_1 \bullet \mathrm{pre}^{\mathrm{Ret}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2))
\end{aligned} \tag{5.10}
$$

$$
\begin{aligned}
& \mathrm{pre}^{\mathrm{Con}}{}_{Op,(1,2)}(u_0, i_0, u_2, i_2) \equiv \\
& \qquad (\exists\, u_1, i_1 \bullet \\
& \qquad\qquad (\mathrm{pre}^{\mathrm{Ret}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2)) \vee \\
& \qquad\qquad (\mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2)) \vee \\
& \qquad\qquad (\mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2)))
\end{aligned} \tag{5.11}
$$

Proof Sketch. This requires some tedious, but otherwise undemanding calculations from the definitions. Because the scopes of various intermediate existential quantifiers are different in the left and right hand sides of (5.10) and (5.11), closedness is used in one direction to effect a reconciliation, determinism is used in the other. ☺

Corollary 5.11 For a composition of two specifically or generally closed retrenchments which both respect their regular data, we have (5.10) and (5.11).

Proof. We merely need to replace the invocations of determinism in the proof of Proposition 5.10 by an appeal to regularity and to conditions (1) and (4) of Definition 3.17, to validate the selection of a common intermediate after-state and output pair across both clauses at the relevant points in the proof. ☺

Theorem 5.12 The composition of specifically or generally closed retrenchments, with data given by (4.5)-(4.8), between deterministic systems, is associative.

Proof Sketch. Proving associativity demands that we substitute a binary composition into another binary composition, and —in order to show that the result is equivalent to the other association order— that we exhibit the symmetry of the result in the system indices. Pursued naively in the present context, the (unwieldy) result turns out to not be symmetric as required. However, closer inspection reveals a number of opportunities to apply the absorption law, after which the remainder can be manipulated into a symmetric form. ☺

Corollary 5.13 The composition of specifically or generally closed retrenchments, given by (4.5)-(4.8), which both respect their regular data, is associative.

Proof Sketch. Were it true, we would show first that the composition of two specifically or generally closed retrenchments which both respect their regular data has regular data, and moreover respects it. Unfortunately the composition of regular relations is not regular unreservedly, so this will not work. Nevertheless, the only properties of retrenchments which respect their regular data that would be needed to prove Corollary 5.11 are (a) and (b) as follows:

(a) Given two level 1 steps u_1 -(i_1, Op_1, $o_{1,\mathrm{a}}$)-› $u'_{1,\mathrm{a}}$ and u_1 -(i_1, Op_1, $o_{1,\mathrm{b}}$)-› $u'_{1,\mathrm{b}}$, whenever u_1 -(i_1, Op_1, $o_{1,\mathrm{a}}$)-› $u'_{1,\mathrm{a}}$ is related by $G_1(u'_0, u'_{1,\mathrm{a}}) \wedge O_{Op,1}(o_0, o_{1,\mathrm{a}}; u'_0, u'_{1,\mathrm{a}}, i_0, i_1, u_0, u_1)$ or $C_{Op,1}(u'_0, u'_{1,\mathrm{a}}, o_0, o_{1,\mathrm{a}}; i_0, i_1, u_0, u_1)$ to a level 0 step u_0 -(i_0, Op_0, o_0)-› u'_0, then the same can be said about u_1 -(i_1, Op_1, $o_{1,\mathrm{b}}$)-› $u'_{1,\mathrm{b}}$.

(b) Similarly for analogous relationships to a level 2 step u_2 -(i_2, Op_2, o_2)-› u'_2.

When two out of three specifically or generally closed retrenchments which all respect their regular data are composed, it is not hard to see that these properties persist for the system at the interface of the composition and the remaining retrenchment. Thus we can re-establish the analogue for three retrenchments of Corollary 5.11, and thence, following Theorem 5.12, the associativity of composition that we seek, despite the failure in general of the regular data conditions for the composites. ☺

From these facts we readily deduce the following.

Theorem 5.14 The composition of specifically or generally closed neat retrenchments, given by (4.5)-(4.8), between deterministic systems, is associative.

Corollary 5.15 The composition of specifically or generally closed neat retrenchments, given by (4.5)-(4.8), which both respect their regular data, is associative.

Now we progress to consider fastidious retrenchments.

Theorem 5.16 With the notations of Theorem 4.7, two fastidious retrenchments between deterministic systems, which are moreover either specifically or generally closed, compose to give a single fastidious resp. specifically or generally closed retrenchment given by (4.5)-(4.8).

Proof Sketch. Theorem 4.7 tells us we have a retrenchment, and Theorem 5.5 and Theorem 5.6 tell us that the resulting retrenchment is specifically or generally closed respectively, so we just have to show that it is fastidious. To do this we suppose that $\overline{G}_{Op,(1,2)} \wedge \overline{C}_{Op,(1,2)}$ is satisfiable, we instantiate the intermediate variables in the most general way, we then amalgamate these instantiations using closedness and determinism, and we then derive a disjunction, each term of which contradicts the fastidiousness of one of the original retrenchments. ☺

Predictably enough we have:

Corollary 5.17 With the notations of Theorem 4.7, two fastidious retrenchments which both respect their regular data, and which are moreover either specifically or generally closed, compose to give a single fastidious resp. specifically or generally closed retrenchment given by (4.5)-(4.8).

Since the data for a composed fastidious retrenchment is the same as that for a composed neat retrenchment, Theorem 5.14 immediately yields:

Theorem 5.18 The composition of specifically or generally closed fastidious retrenchments, given by (4.5)-(4.8), between deterministic systems, is associative.

Corollary 5.19 The composition of specifically or generally closed fastidious retrenchments, given by (4.5)-(4.8), which both respect their regular data, is associative.

6 Conclusions

In the preceding sections we have focused on introducing various strengthenings of the notion of retrenchment that subsequently lead to tighter laws of composition, helping to avoid the 'junk' that purely propositional reasoning can generate. Regarding such tighter laws, it is clear that they come at a price. When we come to consider closure of composition, and even more to the point, associativity, we find that these properties do not hold automatically for the new formulations. The calculations needed to establish the results of Section 5 turn out to be quite convoluted, and demonstrate the lengths to which we must go to recover such properties. This goes to show, that regarding the properties considered in this paper, associativity turns out to be much more like a completeness property than a soundness property. To prove associativity we must be able to decompose a composite structure into its components in a well behaved way, in order that we can subsequently reassemble *all* the pieces into the other association order. The frequent presence of conjunctions of existentially quantified expressions, in which the existential witnesses drawn from the same domain cannot be assumed to be the same across different conjuncts, causes endless trouble in this regard.

Our approach in preceding sections was to restrict where necessary the kind of retrenchments we considered in order to carry through the generic proofs we wanted in the most transparent manner possible. This meant imposing conditions on the collection of relations that expresses a retrenchment, or on the transition relations of the systems in question, or on the relationship between the two. We can call this the extrinsic approach because the conditions come from outside, and any systems etc. that do not satisfy the relevant conditions are excluded from consideration. The extrinsic approach gives an easily digestible formulation of what is needed to carry through a proof.

This extrinsic approach is not only easy to grasp, but also often proves useful, because people like to build systems using concepts that are as simple as is practicable. Consequently the ingredients of those systems can frequently satisfy simple structural conditions such as the ones we hypothesised.

However there are other options for getting the results we obtained. The conditions assumed were normally sufficient conditions to enable a particular proof fragment to be carried through. As an alternative, one could instead axiomatise the required proof fragments themselves. We can call this the weakly extrinsic approach. Such a reformulation of the material in this paper would be more widely applicable than the treatment here because we would not be insisting that a particular condition holds everywhere, but only where it will be utilised in a proof, and thus more systems would potentially satisfy the conditions demanded. (As an example, in Corollary 5.17 we used regularity to prove that from $G_1(u'_0, u'_{1,\mathrm{a}}) \wedge O_{Op,1}(o_0, o_{1,\mathrm{a}}; u'_0, u'_{1,\mathrm{a}}, \ldots) \wedge C_{Op,1}(u'_0, u'_{1,\mathrm{b}}, o_0, o_{1,\mathrm{b}}; \ldots)$ we could, amongst other things, infer $C_{Op,1}(u'_0, u'_{1,\mathrm{a}}, o_0, o_{1,\mathrm{a}}; \ldots)$. However instead of using regularity we could have assumed this implication directly as a property of the component retrenchments, and the proof would have succeeded equally well; moreover we would only have assumed just what was needed, rather than a global condition like regularity which imposes constraints even in places where the proof in question does not exploit them.) A specific case when the weakly extrinsic approach was actually unavoidable in this paper occurred in Corollary 5.13, where the simple assumptions of regularity did not compose, and we had to refer to a more finegrained condition to complete the proof.

There is yet another approach which is also available. The nature of retrenchments is that there is always scope for a tradeoff between facts stated in the output and concedes relations, and restrictions imposed in the within relations. In the present context, instead of imposing conditions on systems and retrenchments from the outside, we have the option of drafting the composed within relations so that the resulting composed retrenchments have the properties we seek to prove, given that the operation PO has the within relation as a hypothesis. In other words we create the composed retrenchments in such a manner that they avert their gaze from those parts of the two systems which do not comply with the criteria demanded for the proof of the desired property. This enables any two systems to be composed by a suitable version of any of the methods that we have introduced in this paper, at the risk that in certain cases, the composed retrenchment can turn out to be too narrowly defined (or even vacuous) if the resulting within relation turns out to be too strong (or even empty). Possibilities such as these remain to be investigated.

These technical difficulties, that arise so quickly when disjunction features so prominently at a structural level as it does in retrenchment, makes it is easy to see why there is such a strong impulse to use refinement wherever possible. The accumulation of properties, without the possibility of later needing to deny properties established earlier —so characteristic of well constructed refinement approaches— is highly appealing when compared to what we had to do above, and we would certainly not dissuade from this approach when it can achieve what is desired in a sensible way.

Nevertheless the real world is a messy place where such an accumulative strategy cannot always be carried through convincingly for realistic applications, and some-times it cannot be carried through at all. (One clear example of the latter is the capture of the transition from continuous models to discrete models, in engineering applications that require the modelling of physical phenomena in software; there, the way that engineers describe the continuous to discrete transition does not lend itself to a refinement treatment.) The intention is that once the most challenging modelling steps have been captured within suitable retrenchments, refinement, with its stronger grip on how properties evolve through the development, can control the remaining less controversial steps of the development. In other words we should apply the *Tower Pattern* [Banach et al. (2005), Banach and Jeske (2009a), Jeske (2005)] to get the best of both worlds.

References

Banach R. (1994); Regular Relations and Bicartesian Squares. Theor. Comp. Sci. **129**, 187-192.

Banach R. (1995); On Regularity in Software Design. Sci. Comp. Prog. **24**, 221-248.

Banach R. (2009); Model Based Refinement and the Design of Retrenchments. *submitted*.

Banach R., Jeske C. (2009a); Retrenchment and Refinement Interworking: the Tower Theorems. *submitted*.

Banach R., Jeske C. (2009b); Simple Feature Engineering via Retrenchment. *submitted*.

Banach R., Poppleton M., Jeske C., Stepney S. (2005); Retrenching the Purse: Finite Sequence Numbers, and the Tower Pattern. *in*: Proc. FM-05, Fitzgerald, Hayes, Tarlecki (eds.), LNCS **3582**, 382-398, Springer.

Banach R., Jeske C., Poppleton M. (2008); Composition Mechanisms for Retrenchment. J. Log. Alg. Prog., **75**, 209-229.

Banach R., Poppleton M., Jeske C., Stepney S. (2007); Engineering and Theoretical Underpinnings of Retrenchment. Sci. Comp. Prog., **67**, 301-329.

Derrick J., Boiten E. (2001); Refinement in Z and Object-Z, Foundations and Advanced Applications. FACIT, Springer.

de Roever W-P., Engelhardt K. (1998); Data Refinement: Model-Oriented Proof Methods and their Comparison. Cambridge University Press.

Jeske C. (2005); Algebraic Integration of Retrenchment and Refinement. PhD. Thesis, School of Computer Science, University of Manchester.

RET; The Retrenchment Homepage. http://www.cs.man.ac.uk/~banach/retrenchment

Schmidt G., Ströhlhein T. (1993); Relations and Graphs, Discrete Mathematics for Computer Scientists. Springer.