Stronger Compositions for Retrenchments, and Feature Engineering

R. Banach, C. Jeske
Computer Science Dept., 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 compositionality 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. This technology is used to give a retrenchment account of elementary feature engineering, the full flexibility of which, refinement can struggle to capture.
Keywords. Retrenchment, Compositionality, Associativity, Feature Engineering.

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, which we will not repeat here. In [Banach et al. (2008)], various kinds of composition for retrenchment were studied, and these were shown to be both associative individually, and associatively compatible.

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, the composition mechanisms of [Banach et al. (2008)], which are based on purely propositional reasoning, are certainly the ones that most obviously come to mind. However, while being perfectly sound, they 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, and when a case that is false is combined with a case that is true (at a given point), the result is a case that is false. The number of such false cases can grow exponentially in the number of retrenchments that are being composed, interfering with the usefulness of retrenchment in the applications sphere.

In this paper we attempt to bypass the proliferation of nonsense cases by exploiting semantic insights of varying depth to yield stronger composition laws, attacking the notion of composition from a different direction. And while we can successfully limit the nonsensical proliferation in varying degrees, the price we have to pay is that various considerations, notably associativity, become technically more troublesome. Our investigations are confined to vertical composition: the relationship of our results here to the propositional reasoning for vertical composition in [Banach et al. (2008)]

shows how things would go for other types of composition. We illustrate what this general framework can accomplish by considering a simple model for feature engineering.

In more detail, the rest of the paper is as follows. Section 2 recalls the basic definitions for retrenchment, the 'usual' propositionally based vertical composition, and default retrenchments. Section 3 considers some special cases of retrenchment, the tidy, neat and fastidious retrenchments. We see that default retrenchments are naturally fastidious, and also neat or tidy under additional constraints. 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 to show that the new compositions preserve the stronger properties assumed. Beginning with some counterexamples to illustrate why it is nontrivial, Section 5 explores the compositionality and associativity properties of the stronger methods of composition. After some protracted calculations, sufficient conditions are established which guarantee the needed compositionality and associativity.

Section 6 applies the technology just developed to give a retrenchment based account of software feature engineering. Features in software are normally introduced and manipulated without too much regard to whether or not what is being done lends itself well to being expressed via refinement. Usually 'not' prevails. We see that with the help of a few reasonable assumptions, the more liberal possibilities that retrenchment allows us, are flexible enough to give an unstressed account of at least the kind of elementary feature oriented situations satisfying the stated assumptions. Section 7 concludes, intimating that in this paper we have merely scratched the surface in terms of the possibilities for richer compositions for retrenchments, and points to future work for mapping out the territory more extensively.

2 Retrenchment

In this section we give our notational conventions and present our basic definitions. We suppose that there is an abstract system *Abs* and a concrete one *Conc*. The abstract system has a set of operation names Ops_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 any states 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 $j \in \mathsf{J}_{Op_\mathrm{C}}$, outputs $p \in \mathsf{P}_{Op_\mathrm{C}}$. 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

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; the inclusion can be proper so the converse need not hold.[1] 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.[2] 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:

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

2.2 Vertical Composition

We next record the definition of (the usual, propositionally based) vertical composition for retrenchments. See [Banach et al. (2008)] for a proof of soundness.

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 $G_{(1,2)}$, $\{P_{Op,(1,2)}, O_{Op,(1,2)}, C_{Op,(1,2)} \mid Op \in \mathsf{Ops}_0\}$, where:

$$G_{(1,2)}(u_0, u_2) \;\equiv\; [\exists\, u_1 \bullet G_1(u_0, u_1) \wedge G_2(u_1, u_2)] \tag{2.3}$$

$$\begin{array}{l} P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \;\equiv \\ \quad [\exists\, u_1, i_1 \bullet G_1(u_0, u_1) \wedge G_2(u_1, u_2) \wedge \\ \qquad P_{Op,1}(i_0, i_1, u_0, u_1) \wedge P_{Op,2}(i_1, i_2, u_1, u_2)] \end{array} \tag{2.4}$$

$$\begin{array}{l} O_{Op,(1,2)}(o_0, o_2; u'_0, u'_2, i_0, i_2, u_0, u_2) \;\equiv \\ \quad [\exists\, u'_1, o_1, u_1, i_1 \bullet O_{Op,1}(o_0, o_1; \ldots) \wedge O_{Op,2}(o_1, o_2; \ldots)] \end{array} \tag{2.5}$$

$$\begin{array}{l} C_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2) \;\equiv \\ \quad [\exists\, u'_1, o_1, u_1, i_1 \bullet \\ \qquad (G_1(u'_0, u'_1) \wedge O_{Op,1}(o_0, o_1; u'_0, u'_1, i_0, i_1, u_0, u_1) \wedge \\ \qquad\quad C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2)) \vee \\ \qquad (C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \\ \qquad\quad G_2(u'_1, u'_2) \wedge O_{Op,2}(o_1, o_2; u'_1, u'_2, i_1, i_2, u_1, u_2)) \vee \\ \qquad (C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \\ \qquad\quad C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2))] \end{array} \tag{2.6}$$

1. This confirms that the 'A' and 'C' subscripts on operation names are meta level tags.
2. 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.

2.3 Default Retrenchments

Default retrenchments make precise the intuition that 'an arbitrary pair of systems' can be related by retrenchment. Since they arise in a generic manner, they can be used to give generic treatments of many situations via retrenchment, our discussion of feature engineering in Section 7 being a case in point. We recall the following from [Banach et al. (2007)].

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

$$\begin{array}{l} P^{\mathrm{Def}}{}_{Op}(i, j, u, v) \equiv \\ \quad (G(u, v) \wedge P_{Op}(i, j, u, v) \wedge \\ \qquad (\exists\, u', o, v', p \bullet stp_{Op_A}(u, i, u', o) \wedge stp_{Op_C}(v, j, v', p))) \end{array} \tag{2.7}$$

$$\begin{array}{l} C^{\mathrm{Def}}{}_{Op}(u', v', o, p; i, j, u, v) \equiv \\ \quad (G(u, v) \wedge P_{Op}(i, j, u, v) \wedge stp_{Op_A}(u, i, u', o) \wedge stp_{Op_C}(v, j, v', p) \wedge \\ \qquad \neg\, (G(u', v') \wedge O_{Op}(o, p; u', v', i, j, u, v)) \end{array} \tag{2.8}$$

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

Default retrenchments stand in contrast to bespoke retrenchments, ones specifically crafted by designers to express the goals of their design step. Normally, a bespoke retrenchment will have a concession weaker than $C^{\mathrm{Def}}{}_{Op}$, in order to more transparently express the intended design goal. 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 an retrenchment from *Abs* to *Conc* as previously.

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

$$\begin{array}{l} \overline{G}_{Op}(u', v', o, p; i, j, u, v) \equiv \\ \quad (G(u, v) \wedge P_{Op}(i, j, u, v) \wedge stp_{Op_A}(u, i, u', o) \wedge stp_{Op_C}(v, j, v', p) \wedge \\ \qquad G(u', v') \wedge O_{Op}(o, p; u', v', i, j, u, v)) \end{array} \tag{3.1}$$

and the concedes closure of Op by:

$$\begin{array}{l} \overline{C}_{Op}(u', v', o, p; i, j, u, v) \equiv \\ \quad (G(u, v) \wedge P_{Op}(i, j, u, v) \wedge stp_{Op_A}(u, i, u', o) \wedge stp_{Op_C}(v, j, v', p) \wedge \\ \qquad C_{Op}(u', v', o, p; i, j, u, v)) \end{array} \tag{3.2}$$

Loosely speaking, the retrieve and concedes closures of Op give versions of the retrieve and output relations and the concedes relations respectively, that are validated in a particularly strong way; i.e. there exist abstract and concrete transitions that witness the validity of these closures whenever they are true, something that need not hold for either $G(u', v') \wedge O_{Op}(o, p; \ldots)$ or for $C_{Op}(u', v', o, p; \ldots)$ in isolation. In a

sense, the $G(u', v') \wedge O_{Op}(o, p; \ldots)$ and $C_{Op}(u', v', o, p; \ldots)$ that appear in the consequent of the retrenchment operation PO, and are largely determined by what the designer deems important to capture in the PO, can be viewed as a shorthand for $\overline{G}_{Op}$ and $\overline{C}_{Op}$ respectively, in a manner made precise by the next result.

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

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

Proof. Straightforward: in the forward direction we just note that the facts beyond $stp_{Op_A} \wedge G' \wedge O_{Op}$ 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. ☺

It is clear that $\overline{G}_{Op}(u', v', o, p; i, j, u, v)$ and $\overline{C}_{Op}(u', v', o, p; i, j, u, v)$ describe exactly those situations in the relationship between the two systems which are captured by the operation PO. The closures enable us to define some additional conditions on retrenchments that later lead to interesting composition properties.

Definition 3.3 We define the following relations for an abstract operation Op:

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

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

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

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

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

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

Proposition 3.4 For any retrenchment the following holds:

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

where $P^{\text{Def}}{}_{Op}(i, j, u, v)$ is defined by (3.1).

Proof. If either $\text{pre}^{\text{Ret}}{}_{Op}(u, i, v, j)$ or $\text{pre}^{\text{Con}}{}_{Op}(u, i, v, j)$ is true, then $P^{\text{Def}}{}_{Op}(i, j, u, v)$ follows by weakening. Conversely suppose $P^{\text{Def}}{}_{Op}(i, j, u, v)$ holds as witnessed by some $\underline{u}', \underline{o}, v', p$. Then for u, i, v, j, v', p, we have $(G \wedge P_{Op} \wedge stp_{Op_C})$ so that we can invoke the operation PO to infer the existence of some u', o (not necessarily related to $\underline{u}', \underline{o}$), such that the consequent of the PO holds. From this we deduce $\overline{G}_{Op}(u', v', o, p; i, j, u, v) \vee \overline{C}_{Op}(u', v', o, p; i, j, u, v)$ by (3.3), which yields $\text{pre}^{\text{Ret}}{}_{Op}(u, i, v, j) \vee \text{pre}^{\text{Con}}{}_{Op}(u, i, v, j)$ by quantification. ☺

Proposition 3.5 Let an retrenchment be defined in the usual manner. Then the operation PO (2.2) is satisfied iff (3.11) is satisfied:

$$\begin{aligned} &G(u, v) \wedge P_{Op}(i, j, u, v) \wedge stp_{Op_C}(v, j, v', p) \Rightarrow \\ &\quad (\exists\, u', o \bullet stp_{Op_A}(u, i, u', o) \wedge \\ &\qquad ((\text{pre}^{\text{Ret}}{}_{Op}(u, i, v, j) \wedge G(u', v') \wedge O_{Op}(o, p; u', v', i, j, u, v)) \vee \\ &\qquad (\text{pre}^{\text{Con}}{}_{Op}(u, i, v, j) \wedge C_{Op}(u', v', o, p; i, j, u, v)))) \end{aligned} \tag{3.11}$$

Proof. Similar to Proposition 3.2. ☺

The reformulation of the operation PO in (3.11) is very convenient among strengthenings of the operation PO that capture all of the information available in the antecedents and see that it is added to the consequent. All the elements of the original PO remain undisturbed, and the extra information is held in the pre- relations. We will return to this formulation when convenient subsequently.

Definition 3.6 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

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

This says that the combinations of before-states and inputs at both levels that characterise the transitions that can reestablish the retrieve and output relations, are disjoint from those that merely establish the concedes relation, in a particularly strong way.

Lemma 3.7 For a tidy output retrenchment we have:

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

and

$$\mathrm{pre}^{\mathrm{ConA}}{}_{Op}(u, i) \wedge \mathrm{pre}^{\mathrm{RetC}}{}_{Op}(v, j) \equiv \mathsf{false} \tag{3.15}$$

Proof. Obvious. ☺

Definition 3.8 A retrenchment is neat iff for all abstract operations Op:

$$\mathrm{pre}^{\mathrm{Ret}}{}_{Op}(u, i, v, j) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op}(u, i, v, j) \equiv \mathsf{false} \tag{3.16}$$

The neat condition keeps retrieve relation preserving behaviour apart from concedes relation establishing behaviour, as does the tidy condition, but it does it in a technically different and more finegrained way as the next Proposition shows.

Proposition 3.9 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 3.10 A retrenchment is fastidious iff for all abstract operations Op:

$$\overline{G}_{Op}(u', v', o, p; i, j, u, v) \wedge \overline{C}_{Op}(u', v', o, p; i, j, u, v) \equiv \mathsf{false} \tag{3.17}$$

The fastidious condition keeps retrieve relation preserving behaviour apart from concedes relation establishing behaviour, in an even more finegrained way than the neat condition.

Proposition 3.11 A neat retrenchment is fastidious.

Proof. Similar to the preceding. ☺

Proposition 3.12 For any tidy or neat retrenchment we have:

$$(\mathrm{pre}^{\mathrm{Ret}}{}_{Op}(u, i, v, j) \oplus \mathrm{pre}^{\mathrm{Con}}{}_{Op}(u, i, v, j)) \equiv P^{\mathrm{Def}}{}_{Op}(i, j, u, v)) \tag{3.18}$$

where $\oplus$ is exclusive or.

Proof. For a neat retrenchment we have (3.16). 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 3.13 A default retrenchment is fastidious.

Proof. We calculate for a default retrenchment:

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

$$\equiv$$

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

$$\equiv$$

$$(G \wedge P_{Op} \wedge stp_{Op_A} \wedge stp_{Op_C} \wedge G' \wedge O_{Op} \wedge \neg (G' \wedge O_{Op}))$$

$$\equiv$$

$$\mathsf{false} \tag{3.19}$$

☺

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 3.14 A default retrenchment between two deterministic systems is neat.

Proof. We calculate:

$${\mathrm{pre}^{\mathrm{Ret}}}_{Op}(u, i, v, j) \wedge {\mathrm{pre}^{\mathrm{Con}}}_{Op}(u, i, v, j)$$

$\equiv$ (definition)

$$(\exists\, {u'}_{\mathrm{a}}, o_{\mathrm{a}}, {v'}_{\mathrm{a}}, p_{\mathrm{a}} \bullet \overline{G}_{Op}({u'}_{\mathrm{a}}, {v'}_{\mathrm{a}}, o_{\mathrm{a}}, p_{\mathrm{a}}; i, j, u, v)) \wedge (\exists\, {u'}_{\mathrm{b}}, o_{\mathrm{b}}, {v'}_{\mathrm{b}}, p_{\mathrm{b}} \bullet {\overline{C}^{\mathrm{Def}}}_{Op}({u'}_{\mathrm{b}}, {v'}_{\mathrm{b}}, o_{\mathrm{b}}, p_{\mathrm{b}}; i, j, u, v))$$

$\equiv$ (instantiating ${u'}_{\mathrm{a}}, o_{\mathrm{a}}, {v'}_{\mathrm{a}}, p_{\mathrm{a}}, {u'}_{\mathrm{b}}, o_{\mathrm{b}}, {v'}_{\mathrm{b}}, p_{\mathrm{b}}$)

$$(G(u, v) \wedge P_{Op}(i, j, u, v) \wedge stp_{Op_A}(u, i, {u'}_{\mathrm{a}}, o_{\mathrm{a}}) \wedge stp_{Op_C}(v, j, {v'}_{\mathrm{a}}, p_{\mathrm{a}}) \wedge G({u'}_{\mathrm{a}}, {v'}_{\mathrm{a}}) \wedge O_{Op}(o_{\mathrm{a}}, p_{\mathrm{a}}; {u'}_{\mathrm{a}}, {v'}_{\mathrm{a}}, i, j, u, v)) \wedge (G(u, v) \wedge P_{Op}(i, j, u, v) \wedge stp_{Op_A}(u, i, {u'}_{\mathrm{b}}, o_{\mathrm{b}}) \wedge stp_{Op_C}(v, j, {v'}_{\mathrm{b}}, p_{\mathrm{b}}) \wedge {C^{\mathrm{Def}}}_{Op}({u'}_{\mathrm{b}}, {v'}_{\mathrm{b}}, o_{\mathrm{b}}, p_{\mathrm{b}}; i, j, u, v))$$

$\Rightarrow$ (determinism: ${u'}_{\mathrm{a}} = {u'}_{\mathrm{b}} = u'$, $o_{\mathrm{a}} = o_{\mathrm{b}} = o$, ${v'}_{\mathrm{a}} = {v'}_{\mathrm{b}} = v'$, $p_{\mathrm{a}} = p_{\mathrm{b}} = p$)

$$(G(u, v) \wedge P_{Op}(i, j, u, v) \wedge stp_{Op_A}(u, i, u', o) \wedge stp_{Op_C}(v, j, v', p) \wedge G(u', v') \wedge O_{Op}(o, p; u', v', i, j, u, v) \wedge {C^{\mathrm{Def}}}_{Op}(u', v', o, p; i, j, u, v))$$

$\equiv$ (definition)

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

$$\equiv$$

$$\mathsf{false} \tag{3.20}$$

☺

Next we recall that a relation $R : \mathsf{X} \leftrightarrow \mathsf{Y}$ is regular iff $R ; R^{-1} ; R = R$, where ; is forward relational composition; see [Schmidt and Ströhlhein (1993), Banach (1994), Banach (1995)]. 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 $g : \mathsf{Y} \to \mathsf{T}$ such that $f ; g^{-1} = R$. 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 spaces.

Definition 3.15 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 3.16 A retrenchment respects its regular data, iff it has regular data, and also for every abstract transition u -(i, Op_{A}, o)-› u':

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

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

and for every concrete transition v -(j, Op_{C}, p)-› v':

(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 $(\underline{v}', \underline{p}, \underline{j}, \underline{v}) \in [v', p, j, v]_{G' \wedge O}$, and $(\underline{v}', \underline{p}, \underline{j}, \underline{v}) \in [v', p, j, v]_C$, for some (v', p).

(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 3.17 A default retrenchment which respects its regular data is tidy.

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

$$\begin{array}{l}(G(u, v_a) \wedge P_{Op}(i, j_a, u, v_a) \wedge stp_{Op_A}(u, i, u'_a, o_a) \wedge stp_{Op_C}(v_a, j_a, v'_a, p_a) \wedge \\ \quad G(u'_a, v'_a) \wedge O_{Op}(o_a, p_a; u'_a, v'_a, i, j_a, u, v_a)) \wedge \\ (G(u, v_b) \wedge P_{Op}(i, j_b, u, v_b) \wedge stp_{Op_A}(u, i, u'_b, o_b) \wedge stp_{Op_C}(v_b, j_b, v'_b, p_b) \wedge \\ \quad C^{\text{Def}}{}_{Op}(u'_b, v'_b, o_b, p_b; i, j_b, u, v_b)) \end{array} \tag{3.21}$$

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 $C^{\text{Def}}{}_{Op}(u'_b, v'_b, o_b, p_b; i, j_b, u, v_b)$ holds, we deduce $C^{\text{Def}}{}_{Op}(u'_a, v'_b, o_a, p_b; i, j_b, u, v_b)$. 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, $C^{\text{Def}}{}_{Op}(u'_a, v'_b, o_a, p_b; i, j_b, u, v_b)$ implies $C^{\text{Def}}{}_{Op}(u'_a, v'_a, o_a, p_a; i, j_a, u, v_a)$. 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 (3.21), giving false. The calculation for (3.13) is entirely analogous. ☺

Since any tidy retrenchment is neat (Proposition 3.9), we get:

Corollary 3.18 A default retrenchment which respects its regular data, is neat.

We close this section by applying the techniques developed here to the comparison of defaults 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, expressing no more than how abstract and concrete inputs are related (but allowing for the possibility that this relationship may depend on the states). It is important to note that the choice of $P^{\circ}{}_{Op}$ is a meta level issue, as 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. Let $O^{\circ}{}_{Op}(o, p; u', v', i, j, u, v)$ be a correspondingly minimal output relation. Let $P^{\text{Def}}{}_{Op}(i, j, u, v)$ be the default within relation manufactured from $P^{\circ}{}_{Op}$ by using $P^{\circ}{}_{Op}$ instead of P in (2.7), and let $C^{\text{Def}}{}_{Op}(u', v', o, p; i, j, u, v)$ be the corresponding default concedes relation.

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

$$\begin{array}{l}P^{\text{BC}}{}_{Op}(i, j, u, v) \equiv \\ \quad (G(u, v) \wedge P^{\text{Bes}}{}_{Op}(i, j, u, v) \wedge \\ \quad\quad (\exists\, u', o, v', p \bullet stp_{Op_A}(u, i, u', o) \wedge stp_{Op_C}(v, j, v', p))) \end{array} \tag{3.22}$$

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

$$ {P^{\mathrm{BC}}}_{Op}(i, j, u, v) \;\Rightarrow\; {P^{\mathrm{Def}}}_{Op}(i, j, u, v) \tag{3.23} $$

Note that ${P^{\mathrm{BC}}}_{Op}$ and ${P^{\mathrm{Def}}}_{Op}$ provide a better basis for comparison than ${P^{\circ}}_{Op}$ and ${P^{\mathrm{Bes}}}_{Op}$ alone, since the application developer is prone to choose the simplest form for the latter, secure in the knowledge that they will only be used in the context of the relevant POs.

In the same vein, and motivated by the presumption that a bespoke output relation will be at worst stronger than the minimal one, we assume also at the meta level:

$$ \begin{array}{l} G(u', v') \wedge {O^{\mathrm{Bes}}}_{Op}(o, p; u', v', i, j, u, v) \;\Rightarrow \\ \quad G(u', v') \wedge {O^{\circ}}_{Op}(o, p; u', v', i, j, u, v) \end{array} \tag{3.24} $$

The absence of any existential quantification in (3.24) greatly assists the following result.

Proposition 3.19 Let ${\overline{G}^{\mathrm{Bes}}}_{Op}$ and ${\overline{C}^{\mathrm{Bes}}}_{Op}$ be the retrieve and concedes closures for a bespoke retrenchment, and let ${\overline{C}^{\mathrm{Def}}}_{Op}$ be the concedes closure for the default retrenchment. Then assuming (3.23) and (3.24):

$$ \begin{array}{l} {\overline{C}^{\mathrm{Bes}}}_{Op}(u', v', o, p; i, j, u, v) \wedge \neg {\overline{G}^{\mathrm{Bes}}}_{Op}(u', v', o, p; i, j, u, v) \;\Rightarrow \\ \quad {\overline{C}^{\mathrm{Def}}}_{Op}(u', v', o, p; i, j, u, v) \vee \neg {\overline{G}^{\mathrm{Bes}}}_{Op}(u', v', o, p; i, j, u, v) \end{array} \tag{3.25} $$

Proof. Suppressing the variable names we calculate as follows:

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

$$ \equiv $$

$$ \begin{array}{l} G \wedge {P^{\mathrm{Bes}}}_{Op} \wedge {stp}_{{Op}_{\mathrm{A}}} \wedge {stp}_{{Op}_{\mathrm{C}}} \wedge {C^{\mathrm{Bes}}}_{Op} \wedge \\ \quad \neg (G \wedge {P^{\mathrm{Bes}}}_{Op} \wedge {stp}_{{Op}_{\mathrm{A}}} \wedge {stp}_{{Op}_{\mathrm{C}}} \wedge G' \wedge {O^{\mathrm{Bes}}}_{Op}) \end{array} $$

$$ \equiv $$

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

$$ \equiv \text{ (3.24)} $$

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

$$ \Rightarrow \text{ (3.23), weakening, PC)} $$

$$ \begin{array}{l} (G \wedge {P^{\mathrm{Def}}}_{Op} \wedge {stp}_{{Op}_{\mathrm{A}}} \wedge {stp}_{{Op}_{\mathrm{C}}} \wedge \neg (G' \wedge {O^{\circ}}_{Op})) \vee \\ (G \wedge {P^{\mathrm{Bes}}}_{Op} \wedge {stp}_{{Op}_{\mathrm{A}}} \wedge {stp}_{{Op}_{\mathrm{C}}} \wedge \neg (G' \wedge {O^{\mathrm{Bes}}}_{Op})) \end{array} $$

$$ \Rightarrow $$

$$ {\overline{C}^{\mathrm{Def}}}_{Op} \vee \neg {\overline{G}^{\mathrm{Bes}}}_{Op} \tag{3.26} $$

☺

Thus under the constraint of $\neg {\overline{G}^{\mathrm{Bes}}}_{Op}$ in the hypotheses, and in the presence of $\neg {\overline{G}^{\mathrm{Bes}}}_{Op}$ in the conclusion to mop up the properties of any 'stray' transition pairs, we see that the default concedes relation is weaker than a bespoke one, according with rough intuition.

Corollary 3.20 Under the assumptions of Proposition 3.19, if the bespoke retrenchment is fastidious (or neat or tidy), we have:

$$\begin{array}{l}\overline{C}^{\mathrm{Bes}}{}_{Op}(u', v', o, p; i, j, u, v) \Rightarrow \\ \quad \overline{C}^{\mathrm{Def}}{}_{Op}(u', v', o, p; i, j, u, v) \vee \neg \overline{G}^{\mathrm{Bes}}{}_{Op}(u', v', o, p; i, j, u, v)\end{array} \tag{3.27}$$

Proof. We merely need to prefix (3.26) with $\overline{C}^{\mathrm{Bes}}{}_{Op} \Rightarrow \overline{C}^{\mathrm{Bes}}{}_{Op} \wedge \neg \overline{G}^{\mathrm{Bes}}{}_{Op}$. ☺

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. $\overline{G}_{Op,1}$, $\overline{C}_{Op,1}$, $\mathrm{pre}^{\mathrm{Ret}}{}_{Op,1}$, $\mathrm{pre}^{\mathrm{Con}}{}_{Op,1}$, $\mathrm{pre}^{\mathrm{RetA}}{}_{Op,1}$, $\mathrm{pre}^{\mathrm{RetC}}{}_{Op,1}$, $\mathrm{pre}^{\mathrm{ConA}}{}_{Op,1}$, $\mathrm{pre}^{\mathrm{ConC}}{}_{Op,1}$; $\overline{G}_{Op,2}$, $\overline{C}_{Op,2}$, $\mathrm{pre}^{\mathrm{Ret}}{}_{Op,2}$, $\mathrm{pre}^{\mathrm{Con}}{}_{Op,2}$, $\mathrm{pre}^{\mathrm{RetA}}{}_{Op,2}$, $\mathrm{pre}^{\mathrm{RetC}}{}_{Op,2}$, $\mathrm{pre}^{\mathrm{ConA}}{}_{Op,2}$, $\mathrm{pre}^{\mathrm{ConC}}{}_{Op,2}$.

With these in place we can derive strengthenings of the composition of retrenchments that follows from Proposition 3.2 and Proposition 3.5.

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

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

where:

$$G_{(1,2)}(u_0, u_2) \equiv [\exists\, u_1 \bullet G_1(u_0, u_1) \wedge G_2(u_1, u_2)] \tag{4.2}$$

$$\begin{array}{l}P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \equiv \\ \quad [\exists\, u_1, i_1 \bullet G_1(u_0, u_1) \wedge G_2(u_1, u_2) \wedge \\ \qquad P_{Op,1}(i_0, i_1, u_0, u_1) \wedge P_{Op,2}(i_1, i_2, u_1, u_2)]\end{array} \tag{4.3}$$

$$\begin{array}{l}O_{Op,(1,2)}(o_0, o_2; u'_0, u'_2, i_0, i_2, u_0, u_2) \equiv \\ \quad [\exists\, u'_1, o_1, u_1, i_1 \bullet O_{Op,1}(o_0, o_1; \ldots) \wedge O_{Op,2}(o_1, o_2; \ldots)]\end{array} \tag{4.4}$$

$$\begin{array}{l}C_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2) \equiv \\ \quad [\exists\, u'_1, o_1, u_1, i_1 \bullet \\ \qquad (G_1(u'_0, u'_1) \wedge O_{Op,1}(o_0, o_1; u'_0, u'_1, i_0, i_1, u_0, u_1) \wedge \\ \qquad\quad C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2)) \vee \\ \qquad (C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \\ \qquad\quad G_2(u'_1, u'_2) \wedge O_{Op,2}(o_1, o_2; u'_1, u'_2, i_1, i_2, u_1, u_2)) \vee \\ \qquad (C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \\ \qquad\quad C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2))]\end{array} \tag{4.5}$$

and:

$$\overline{G}_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2) \equiv$$
$$(G_{(1,2)}(u_0, u_2) \wedge P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \wedge$$
$$stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge$$
$$G_{(1,2)}(u'_0, u'_2) \wedge O_{Op,(1,2)}(o_0, o_2; u'_0, u'_2, i_0, i_2, u_0, u_2)) \quad (4.6)$$

$$\overline{C}_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2) \equiv$$
$$(G_{(1,2)}(u_0, u_2) \wedge P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \wedge$$
$$stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge$$
$$C_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2)) \quad (4.7)$$

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 (4.2) and (4.3), 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:

$$(G'_1 \wedge O_{Op,1} \wedge G'_2 \wedge O_{Op,2}) \vee$$
$$((G'_1 \wedge O_{Op,1} \wedge C_{Op,2}) \vee (C_{Op,1} \wedge G'_2 \wedge O_{Op,2}) \vee (C_{Op,1} \wedge C_{Op,2})) \quad (4.8)$$

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 $\overline{G}_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2)$ after pushing the existential quantification over u_1, i_1, u'_1, o_1 over the first '$\vee$' in (4.8), this 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 (4.8) we obtain $\overline{C}_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2)$ after dealing with the quantification over u_1, i_1, u'_1, o_1, also discharging (4.1). ☺

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

$$G_{(1,2)}(u_0, u_2) \equiv [\exists\, u_1 \bullet G_1(u_0, u_1) \wedge G_2(u_1, u_2)] \quad (4.9)$$

$$P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \equiv$$
$$[\exists\, u_1, i_1 \bullet G_1(u_0, u_1) \wedge G_2(u_1, u_2) \wedge$$
$$P_{Op,1}(i_0, i_1, u_0, u_1) \wedge P_{Op,2}(i_1, i_2, u_1, u_2)] \quad (4.10)$$

$$O_{Op,(1,2)}(o_0, o_2; u'_0, u'_2, i_0, i_2, u_0, u_2) \equiv$$
$$[\exists\, u'_1, o_1, u_1, i_1 \bullet O_{Op,1}(o_0, o_1; \dots) \wedge O_{Op,2}(o_1, o_2; \dots) \wedge$$
$$\text{pre}^{\text{Ret}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \text{pre}^{\text{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2)] \tag{4.11}$$

$$C_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2) \equiv$$
$$[\exists\, u'_1, o_1, u_1, i_1 \bullet$$
$$(G_1(u'_0, u'_1) \wedge O_{Op,1}(o_0, o_1; u'_0, u'_1, i_0, i_1, u_0, u_1) \wedge$$
$$C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2) \wedge$$
$$\text{pre}^{\text{Ret}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \text{pre}^{\text{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2)) \vee$$
$$(C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge$$
$$G_2(u'_1, u'_2) \wedge O_{Op,2}(o_1, o_2; u'_1, u'_2, i_1, i_2, u_1, u_2) \wedge$$
$$\text{pre}^{\text{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \text{pre}^{\text{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2)) \vee$$
$$(C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge$$
$$C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2) \wedge$$
$$\text{pre}^{\text{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \text{pre}^{\text{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2))] \tag{4.12}$$

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 is disposed of as in Theorem 4.1.

For the operation PO, we are required to establish:

$$G_{(1,2)}(u_0, u_2) \wedge P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \Rightarrow$$
$$(\exists\, u'_0, o_0 \bullet stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge$$
$$((G_{(1,2)}(u'_0, u'_2) \wedge O_{Op,(1,2)}(o_0, o_2; u'_0, u'_2, i_0, i_2, u_0, u_2)) \vee$$
$$C_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2))) \tag{4.13}$$

which would not entail anything unfamiliar were it not for the fact that $O_{Op,(1,2)}$ is now given by (4.11) and not (2.5) and $C_{Op,(1,2)}$ is given by (4.12) and not (2.6). We argue as usual from $G_{(1,2)} \wedge P_{Op,(1,2)}$, to get u_1 and i_1, taking u_1 to be common. From $G_2 \wedge P_{Op,2} \wedge stp_{Op,2}$ we use the operation PO to infer $(\exists\, u'_1, o_1 \bullet stp_{Op,1} \wedge ((G_2 \wedge O_{Op,2}) \vee C_{Op,2}))$. For u_1, i_1, u'_1, o_1, we use $G_1 \wedge P_{Op,1} \wedge stp_{Op,1}$ and the operation PO 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.

So for all these existential witnesses we have established:

$$stp_{Op,0} \wedge stp_{Op,1} \wedge stp_{Op,2} \wedge G_1 \wedge P_{Op,1} \wedge G_2 \wedge P_{Op,2} \wedge$$
$$((G'_1 \wedge O_{Op,1}) \vee C_{Op,1}) \wedge ((G'_2 \wedge O_{Op,2}) \vee C_{Op,2}) \tag{4.14}$$

The two disjunctions generate a disjunction of four terms by the distributive law: $(G'_1 \wedge O_{Op,1} \wedge G'_2 \wedge O_{Op,2} \dots) \vee (G'_1 \wedge O_{Op,1} \wedge C_{Op,2} \dots) \vee (C_{Op,1} \wedge G'_2 \wedge O_{Op,2} \dots) \vee (C_{Op,1} \wedge C_{Op,2} \dots)$, where the ellipsis refers to everything on the top line of (4.14). In each of these disjuncts it is now straightforward to see that the properties asserted by the various pre- clauses in (4.11) and (4.12) are easily provable, so that a composition of retrenchments utilising (4.9)-(4.12) is sound. ☺

Note the contrast between Theorem 4.1 and Theorem 4.2. The former is stated in terms of a modified operation PO, while the latter uses the standard retrenchment operation PO but just requires the composed retrenchment data to be strengthened. Furthermore readers can easily convince themselves, using Proposition 3.2, that a proof combining elements of both of these can establish a version of Theorem 4.1 that uses (4.11) and (4.12) instead of (4.4) and (4.5).

We moreover note that in Theorem 4.2, although we are able to strengthen the composed output and concedes relation in the manner expected from Proposition 3.5, 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 (4.11) and (4.12), and for which there are in principle fewer opportunities.

Now we turn to the tidy, neat, and fastidious retrenchments. Under suitable assumptions we will 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:

$$\mathrm{pre}^{\mathrm{RetA}}{}_{Op,2}(u_1, i_1) \Rightarrow \mathrm{pre}^{\mathrm{RetC}}{}_{Op,1}(u_1, i_1) \tag{4.15}$$

and

$$\mathrm{pre}^{\mathrm{ConA}}{}_{Op,2}(u_1, i_1) \Rightarrow \mathrm{pre}^{\mathrm{ConC}}{}_{Op,1}(u_1, i_1) \tag{4.16}$$

hold for the intermediate system.

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

$$G_{(1,2)}(u_0, u_2) \equiv [\exists\, u_1 \bullet G_1(u_0, u_1) \wedge G_2(u_1, u_2)] \tag{4.17}$$

$$\begin{aligned} &P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \equiv \\ &\quad [\exists\, u_1, i_1 \bullet G_1(u_0, u_1) \wedge G_2(u_1, u_2) \wedge \\ &\qquad P_{Op,1}(i_0, i_1, u_0, u_1) \wedge P_{Op,2}(i_1, i_2, u_1, u_2)] \end{aligned} \tag{4.18}$$

$$\begin{aligned} &O_{Op,(1,2)}(o_0, o_2; u'_0, u'_2, i_0, i_2, u_0, u_2) \equiv \\ &\quad [\exists\, u'_1, o_1, u_1, i_1 \bullet O_{Op,1}(o_0, o_1; u'_0, u'_1, i_0, i_1, u_0, u_1) \wedge \\ &\qquad O_{Op,2}(o_1, o_2; u'_1, u'_2, i_1, i_2, u_1, u_2)] \end{aligned} \tag{4.19}$$

$$\begin{aligned} &C_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2) \equiv \\ &\quad [\exists\, u'_1, o_1, u_1, i_1 \bullet C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \\ &\qquad C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2)] \end{aligned} \tag{4.20}$$

Proof. We can take over the first two paragraphs of the proof of Theorem 4.2 verbatim, aside from the fact that a different composition of output and concedes relations is being dealt with here.

Having established (4.14) as before, we now argue as follows. Since the intermediate to lowest level retrenchment is tidy, either $\mathrm{pre}^{\mathrm{RetC}}{}_{Op,2}(u_2, i_2)$ or $\mathrm{pre}^{\mathrm{ConC}}{}_{Op,2}(u_2, i_2)$ will hold, but not both. Suppose we have $\mathrm{pre}^{\mathrm{RetC}}{}_{Op,2}(u_2, i_2)$. Then by tidiness we can deduce $\mathrm{pre}^{\mathrm{RetA}}{}_{Op,2}(u_1, i_1)$ too. Then for $u_1, i_1, u'_1, o_1, u_2, i_2, u'_2, o_2$, from the truth of the operation PO we must have $\overline{G}_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2)$, since tidiness and our assumptions preclude $\overline{C}_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2)$. Since the two retrenchments are compatibly tidy and we have $\mathrm{pre}^{\mathrm{RetA}}{}_{Op,2}(u_1, i_1)$, we can also deduce $\mathrm{pre}^{\mathrm{RetC}}{}_{Op,1}(u_1, i_1)$ from (4.15). Since the top level to intermediate retrenchment is tidy, $\mathrm{pre}^{\mathrm{ConC}}{}_{Op,1}(u_1, i_1)$ becomes impossible, whereupon the truth of the operation PO for

the top level to intermediate output retrenchment implies for u_0, i_0, u'_0, o_0, u_1, i_1, u'_1, o_1, that $\overline{G}_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1)$ holds, and that $\overline{C}_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1)$ is precluded. So we have $\overline{G}_{Op,2} \wedge \overline{G}_{Op,1}$ which yields $G_{(1,2)}(u'_0, u'_2) \wedge O_{Op,(1,2)}(o_0, o_2; u'_0, u'_2, i_0, i_2, u_0, u_2)$, which for the stated u'_0, o_0 is enough to prove the operation PO.

Alternatively suppose that $\text{pre}^{\text{ConC}}{}_{Op,2}(u_2, i_2)$ is true. Then analogous reasoning establishes in turn $\overline{C}_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2)$, $\text{pre}^{\text{ConA}}{}_{Op,2}(u_1, i_1)$, $\text{pre}^{\text{ConC}}{}_{Op,1}(u_1, i_1)$, and $\overline{C}_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1)$. Thus $C_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2)$ (given by (4.20)) holds. The two cases together verify the operation PO for the composed retrenchment with retrieve, within, output and concedes relations given by (4.17)-(4.20). ☺

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:

$$\begin{array}{l} G_{(1,2)}(u_0, u_2) \wedge P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \Rightarrow \\ \quad (\exists\, u'_0, o_0 \bullet stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge \\ \qquad ((G_{(1,2)}(u'_0, u'_2) \wedge O_{Op,(1,2)}(o_0, o_2; u'_0, u'_2, i_0, i_2, u_0, u_2)) \vee \\ \qquad C_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2))) \end{array} \tag{4.21}$$

with $G_{(1,2)}$, $P_{Op,(1,2)}$, $O_{Op,(1,2)}$ and $C_{Op,(1,2)}$ given by (4.9)-(4.12), 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 $C_{Op,(1,2)}$ (4.22)

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

Proof. Since a neat retrenchment is a retrenchment, it is described by the data (4.9)-(4.12), by Theorem 4.2. Given (u_1, i_1), suppose an intermediate step u_1 -$(i_1, Op_I, o_{1,a})$-› $u'_{1,a}$ witnessed some option (a) from (4.22) in the proof of Theorem 4.2, and an intermediate step u_1 -$(i_1, Op_I, o_{1,b})$-› $u'_{1,b}$ witnessed option (b) from (4.22) in the proof of Theorem 4.2, where $a, b \in \{1 \ldots 4\}$ and $a \neq b$. Then we could contradict the neatness of either the top level to intermediate retrenchment, or the intermediate to lowest level retrenchment. For example, if options (1) and (2) were verified by u_1 -$(i_1, Op_I, o_{1,a})$-› $u'_{1,a}$ and u_1 -$(i_1, Op_I, o_{1,b})$-› $u'_{1,b}$ respectively, then we would also have $\text{pre}^{\text{Ret}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \text{pre}^{\text{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2)$ from (4.11) and $\text{pre}^{\text{Ret}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \text{pre}^{\text{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2)$ from (4.12). As a consequence $\text{pre}^{\text{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2)$ and $\text{pre}^{\text{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2)$ would contradict the neatness of the intermediate to lowest level retrenchment. ☺

Corollary 4.6 Two neat retrenchments that further satisfy:

$$\mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2) \equiv \mathsf{false} \tag{4.23}$$

compose to give a single retrenchment given by (4.17)-(4.19) and:

$$\begin{array}{l} C_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2) \equiv \\ \quad [\exists\, u'_1, o_1, u_1, i_1 \bullet \\ \qquad (G_1(u'_0, u'_1) \wedge O_{Op,1}(o_0, o_1; u'_0, u'_1, i_0, i_1, u_0, u_1) \wedge \\ \qquad\quad C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2) \wedge \\ \qquad\quad \mathrm{pre}^{\mathrm{Ret}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2)) \vee \\ \qquad (C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \\ \qquad\quad G_2(u'_1, u'_2) \wedge O_{Op,2}(o_1, o_2; u'_1, u'_2, i_1, i_2, u_1, u_2) \wedge \\ \qquad\quad \mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2))] \end{array} \tag{4.24}$$

Of course there are similar corollaries when other $\mathrm{pre}^{\mathrm{Con}}{}_{Op}$/$\mathrm{pre}^{\mathrm{Ret}}{}_{Op}$ 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:

$$\begin{array}{l} G_{(1,2)}(u_0, u_2) \wedge P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \Rightarrow \\ \quad (\exists\, u'_0, o_0 \bullet stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge \\ \qquad ((G_{(1,2)}(u'_0, u'_2) \wedge O_{Op,(1,2)}(o_0, o_2; u'_0, u'_2, i_0, i_2, u_0, u_2)) \vee \\ \qquad C_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2))) \end{array} \tag{4.25}$$

with $G_{(1,2)}$, $P_{Op,(1,2)}$, $O_{Op,(1,2)}$ and $C_{Op,(1,2)}$ given by (4.9)-(4.12), 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 $C_{Op,(1,2)}$ (4.26)

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 Compositionality and Associativity

The results of the previous section are not enough to give compositionality let alone associativity for all the various strengthened notions of retrenchment.

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 focussed 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 6.1, both the upper and lower retrenchments are fastidious and neat (though not tidy). However although the intermediate after-state values refered to by the component retrieve relations differ from those refered 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:

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

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

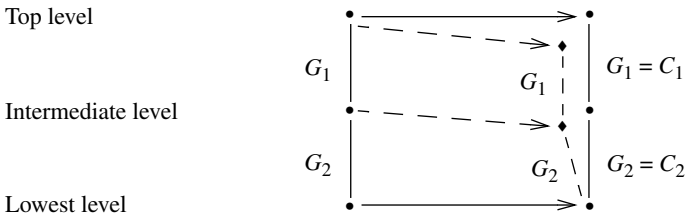

Fig. 1 A non-fastidious composition of two tidy retrenchments.

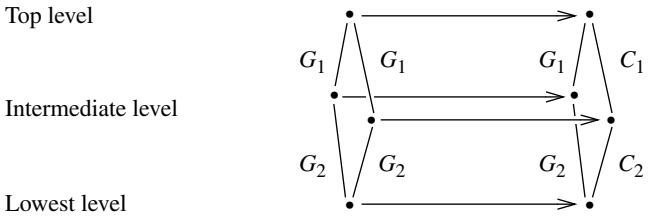

Fig. 2 A non-fastidious composition of two fastidious (and neat) retrenchments.

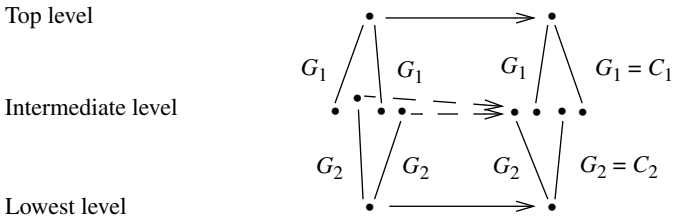

Fig. 3 A problem free composition.

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

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

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

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

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

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

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

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 closedness criteria are compositional as is shown next.

Theorem 5.5 With the current notations, the composition of two specifically closed retrenchments is specifically closed.

Proof. Consider (5.1). We calculate as follows:

$$G_{(1,2)}(u_0, u_2) \wedge P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \wedge G_{(1,2)}(u'_0, u'_2) \wedge O_{Op,(1,2)}(o_0, o_2; \underline{u}'_0, \underline{u}'_2, \underline{i}_0, \underline{i}_2, \underline{u}_0, \underline{u}_2) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2)$$

$\Rightarrow$ (instantiation, for some $u_1, i_1, u'_1, o_1, \underline{u}_1, \underline{i}_1, \underline{u}'_1$)

$$G_1(u_0, u_1) \wedge G_2(u_1, u_2) \wedge P_{Op,1}(i_0, i_1, u_0, u_1) \wedge P_{Op,2}(i_1, i_2, u_1, u_2) \wedge G_1(u'_0, u'_1) \wedge G_2(u'_1, u'_2) \wedge O_{Op,1}(o_0, o_1; \underline{u}'_0, \underline{u}'_1, \underline{i}_0, \underline{i}_1, \underline{u}_0, \underline{u}_1) \wedge O_{Op,2}(o_1, o_2; \underline{u}'_1, \underline{u}'_2, \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2)$$

$\Rightarrow$ (specific closedness, one point rule)

$$G_1(u_0, u_1) \wedge G_2(u_1, u_2) \wedge P_{Op,1}(i_0, i_1, u_0, u_1) \wedge P_{Op,2}(i_1, i_2, u_1, u_2) \wedge G_1(u'_0, u'_1) \wedge G_2(u'_1, u'_2) \wedge$$

$O_{Op,1}(o_0, o_1; \underline{u}'_0, u'_1, \underline{i}_0, i_1, \underline{u}_0, u_1) \wedge O_{Op,2}(o_1, o_2; u'_1, \underline{u}'_2, i_1, \underline{i}_2, u_1, \underline{u}_2) \wedge$
$stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge$
$(u_2 = \underline{u}_2) \wedge (i_2 = \underline{i}_2) \wedge (u'_2 = \underline{u}'_2) \wedge stp_{Op,1}(u_1, i_1, u'_1, o_1)$

$\Rightarrow$ (specific closedness)

$(u_0 = \underline{u}_0) \wedge (i_0 = \underline{i}_0) \wedge (u_2 = \underline{u}_2) \wedge (i_2 = \underline{i}_2) \wedge (u'_0 = \underline{u}'_0) \wedge (u'_2 = \underline{u}'_2) \wedge$
$stp_{Op,0}(u_0, i_0, u'_0, o_0)$ (5.9)

Property (6.2) is similar. Next we examine (5.4), (5.3) being similar. As before we calculate:

$G_{(1,2)}(u_0, u_2) \wedge P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \wedge$
$C_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; \underline{i}_0, \underline{i}_2, \underline{u}_0, \underline{u}_2) \wedge$
$stp_{Op,0}(u_0, i_0, u'_0, o_0)$

$\Rightarrow$ (instantiation, for some $u_1, i_1, u'_1, o_1, \underline{u}_1, \underline{i}_1$)

$G_1(u_0, u_1) \wedge G_2(u_1, u_2) \wedge P_{Op,1}(i_0, i_1, u_0, u_1) \wedge P_{Op,2}(i_1, i_2, u_1, u_2) \wedge$
$((G_1(u'_0, u'_1) \wedge O_{Op,1}(o_0, o_1; u'_0, u'_1, \underline{i}_0, \underline{i}_1, \underline{u}_0, \underline{u}_1) \wedge$
$\quad C_{Op,2}(u'_1, u'_2, o_1, o_2; \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2)) \vee$
$(C_{Op,1}(u'_0, u'_1, o_0, o_1; \underline{i}_0, \underline{i}_1, \underline{u}_0, \underline{u}_1) \wedge$
$\quad G_2(u'_1, u'_2) \wedge O_{Op,2}(o_1, o_2; u'_1, u'_2, \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2)) \vee$
$(C_{Op,1}(u'_0, u'_1, o_0, o_1; \underline{i}_0, \underline{i}_1, \underline{u}_0, \underline{u}_1) \wedge$
$\quad C_{Op,2}(u'_1, u'_2, o_1, o_2; \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2))) \wedge$
$stp_{Op,0}(u_0, i_0, u'_0, o_0)$

$\Rightarrow$ (specific closedness, one point rule)

$G_1(u_0, u_1) \wedge G_2(u_1, u_2) \wedge P_{Op,1}(i_0, i_1, u_0, u_1) \wedge P_{Op,2}(i_1, i_2, u_1, u_2) \wedge$
$((G_1(u'_0, u'_1) \wedge O_{Op,1}(o_0, o_1; u'_0, u'_1, \underline{i}_0, i_1, \underline{u}_0, u_1) \wedge$
$\quad C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, \underline{i}_2, u_1, \underline{u}_2)) \vee$
$(C_{Op,1}(u'_0, u'_1, o_0, o_1; \underline{i}_0, i_1, \underline{u}_0, u_1) \wedge$
$\quad G_2(u'_1, u'_2) \wedge O_{Op,2}(o_1, o_2; u'_1, u'_2, i_1, \underline{i}_2, u_1, \underline{u}_2)) \vee$
$(C_{Op,1}(u'_0, u'_1, o_0, o_1; \underline{i}_0, i_1, \underline{u}_0, u_1) \wedge$
$\quad C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, \underline{i}_2, u_1, \underline{u}_2))) \wedge$
$stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge$
$(u_0 = \underline{u}_0) \wedge (i_0 = \underline{i}_0) \wedge stp_{Op,1}(u_1, i_1, u'_1, o_1)$

$\Rightarrow$ (specific closedness, all three disjuncts)

$(u_0 = \underline{u}_0) \wedge (i_0 = \underline{i}_0) \wedge (u_2 = \underline{u}_2) \wedge (i_2 = \underline{i}_2) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2)$ (5.10)

This completes the proof. ☺

Theorem 5.6 With the current notations, the composition of two generally closed output retrenchments is generally closed.

Proof. We examine (5.6) and (5.7). For the former we have:

$G_{(1,2)}(u_0, u_2) \wedge P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \wedge G_{(1,2)}(u'_0, u'_2) \wedge$
$O_{Op,(1,2)}(o_0, o_2; \underline{u}'_0, \underline{u}'_2, \underline{i}_0, \underline{i}_2, \underline{u}_0, \underline{u}_2) \wedge$
$stp_{Op,0}(u_0, i_0, u'_0, o_0)$

$\Rightarrow$ (instantiation, for some $u_1, i_1, u'_1, o_1, \underline{u}_1, \underline{i}_1, \underline{u}'_1$)

$G_1(u_0, u_1) \wedge G_2(u_1, u_2) \wedge P_{Op,1}(i_0, i_1, u_0, u_1) \wedge P_{Op,2}(i_1, i_2, u_1, u_2) \wedge$
$G_1(u'_0, u'_1) \wedge G_2(u'_1, u'_2) \wedge$
$O_{Op,1}(o_0, o_1; \underline{u}'_0, \underline{u}'_1, \underline{i}_0, \underline{i}_1, \underline{u}_0, \underline{u}_1) \wedge O_{Op,2}(o_1, o_2; \underline{u}'_1, \underline{u}'_2, \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2) \wedge$
$stp_{Op,0}(u_0, i_0, u'_0, o_0)$

$\Rightarrow$ (general closedness)

$G_1(u_0, u_1) \wedge G_2(u_1, u_2) \wedge P_{Op,1}(i_0, i_1, u_0, u_1) \wedge P_{Op,2}(i_1, i_2, u_1, u_2) \wedge$
$G_1(u'_0, u'_1) \wedge G_2(u'_1, u'_2) \wedge$
$O_{Op,1}(o_0, o_1; \underline{u}'_0, \underline{u}'_1, \underline{i}_0, \underline{i}_1, \underline{u}_0, \underline{u}_1) \wedge O_{Op,2}(o_1, o_2; \underline{u}'_1, \underline{u}'_2, \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2) \wedge$
$stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge$
$stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge$
$(\forall\, \underline{u}'_0, \underline{u}'_1, \underline{i}_0, \underline{i}_1, \underline{u}_0, \underline{u}_1 \bullet O_{Op,1}(o_0, o_1; \underline{u}'_0, \underline{u}'_1, \underline{i}_0, \underline{i}_1, \underline{u}_0, \underline{u}_1))$

$\Rightarrow$ (general closedness)

$stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge$
$(\forall\, \underline{u}'_0, \underline{u}'_1, \underline{i}_0, \underline{i}_1, \underline{u}_0, \underline{u}_1 \bullet O_{Op,1}(o_0, o_1; \underline{u}'_0, \underline{u}'_1, \underline{i}_0, \underline{i}_1, \underline{u}_0, \underline{u}_1)) \wedge$
$(\forall\, \underline{u}'_1, \underline{u}'_2, \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2 \bullet O_{Op,2}(o_1, o_2; \underline{u}'_1, \underline{u}'_2, \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2))$

$\Rightarrow$

$stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge$
$(\forall\, \underline{u}'_0, \underline{u}'_2, \underline{i}_0, \underline{i}_2, \underline{u}_0, \underline{u}_2 \bullet O_{Op,(1,2)}(o_0, o_2; \underline{u}'_0, \underline{u}'_2, \underline{i}_0, \underline{i}_2, \underline{u}_0, \underline{u}_2))$ (5.11)

For (6.7) we have:

$G_{(1,2)}(u_0, u_2) \wedge P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \wedge$
$C_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; \underline{i}_0, \underline{i}_2, \underline{u}_0, \underline{u}_2) \wedge$
$stp_{Op,2}(u_2, i_2, u'_2, o_2)$

$\Rightarrow$ (instantiation, for some $u_1, i_1, u'_1, o_1, \underline{u}_1, \underline{i}_1$)

$G_1(u_0, u_1) \wedge G_2(u_1, u_2) \wedge P_{Op,1}(i_0, i_1, u_0, u_1) \wedge P_{Op,2}(i_1, i_2, u_1, u_2) \wedge$
$((G_1(u'_0, u'_1) \wedge O_{Op,1}(o_0, o_1; u'_0, u'_1, \underline{i}_0, \underline{i}_1, \underline{u}_0, \underline{u}_1) \wedge$
$\quad C_{Op,2}(u'_1, u'_2, o_1, o_2; \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2)) \vee$
$(C_{Op,1}(u'_0, u'_1, o_0, o_1; \underline{i}_0, \underline{i}_1, \underline{u}_0, \underline{u}_1) \wedge$
$\quad G_2(u'_1, u'_2) \wedge O_{Op,2}(o_1, o_2; u'_1, u'_2, \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2)) \vee$
$(C_{Op,1}(u'_0, u'_1, o_0, o_1; \underline{i}_0, \underline{i}_1, \underline{u}_0, \underline{u}_1) \wedge$
$\quad C_{Op,2}(u'_1, u'_2, o_1, o_2; \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2))) \wedge$
$stp_{Op,2}(u_2, i_2, u'_2, o_2)$

$\Rightarrow$ (general closedness)

$G_1(u_0, u_1) \wedge G_2(u_1, u_2) \wedge P_{Op,1}(i_0, i_1, u_0, u_1) \wedge P_{Op,2}(i_1, i_2, u_1, u_2) \wedge$
$((G_1(u'_0, u'_1) \wedge O_{Op,1}(o_0, o_1; u'_0, u'_1, \underline{i}_0, \underline{i}_1, \underline{u}_0, \underline{u}_1) \wedge$
$\quad C_{Op,2}(u'_1, u'_2, o_1, o_2; \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2) \wedge$
$\quad (\forall\, \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2 \bullet C_{Op,2}(u'_1, u'_2, o_1, o_2; \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2))) \vee$
$(C_{Op,1}(u'_0, u'_1, o_0, o_1; \underline{i}_0, \underline{i}_1, \underline{u}_0, \underline{u}_1) \wedge$
$\quad G_2(u'_1, u'_2) \wedge O_{Op,2}(o_1, o_2; u'_1, u'_2, \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2) \wedge$
$\quad (\forall\, \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2 \bullet O_{Op,2}(o_1, o_2; u'_1, u'_2, \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2))) \vee$
$(C_{Op,1}(u'_0, u'_1, o_0, o_1; \underline{i}_0, \underline{i}_1, \underline{u}_0, \underline{u}_1) \wedge$
$\quad C_{Op,2}(u'_1, u'_2, o_1, o_2; \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2) \wedge$

$$(\forall\ \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2 \bullet C_{Op,2}(u'_1, u'_2, o_1, o_2; \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2)))) \wedge$$
$$stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge stp_{Op,1}(u_1, i_1, u'_1, o_1)$$

$\Rightarrow$ (general closedness)

$$stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge$$
$$((G_1(u'_0, u'_1) \wedge O_{Op,1}(o_0, o_1; u'_0, u'_1, \underline{i}_0, \underline{i}_1, \underline{u}_0, \underline{u}_1) \wedge$$
$$(\forall\ \underline{i}_0, \underline{i}_1, \underline{u}_0, \underline{u}_1 \bullet O_{Op,1}(o_0, o_1; u'_0, u'_1, \underline{i}_0, \underline{i}_1, \underline{u}_0, \underline{u}_1)) \wedge$$
$$C_{Op,2}(u'_1, u'_2, o_1, o_2; \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2) \wedge$$
$$(\forall\ \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2 \bullet C_{Op,2}(u'_1, u'_2, o_1, o_2; \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2))) \vee$$
$$(C_{Op,1}(u'_0, u'_1, o_0, o_1; \underline{i}_0, \underline{i}_1, \underline{u}_0, \underline{u}_1) \wedge$$
$$(\forall\ \underline{i}_0, \underline{i}_1, \underline{u}_0, \underline{u}_1 \bullet C_{Op,1}(u'_0, u'_1, o_0, o_1; \underline{i}_0, \underline{i}_1, \underline{u}_0, \underline{u}_1)) \wedge$$
$$G_2(u'_1, u'_2) \wedge O_{Op,2}(o_1, o_2; u'_1, u'_2, \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2) \wedge$$
$$(\forall\ \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2 \bullet O_{Op,2}(o_1, o_2; u'_1, u'_2, \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2))) \vee$$
$$(C_{Op,1}(u'_0, u'_1, o_0, o_1; \underline{i}_0, \underline{i}_1, \underline{u}_0, \underline{u}_1) \wedge$$
$$(\forall\ \underline{i}_0, \underline{i}_1, \underline{u}_0, \underline{u}_1 \bullet C_{Op,1}(u'_0, u'_1, o_0, o_1; \underline{i}_0, \underline{i}_1, \underline{u}_0, \underline{u}_1)) \wedge$$
$$C_{Op,2}(u'_1, u'_2, o_1, o_2; \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2) \wedge$$
$$(\forall\ \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2 \bullet C_{Op,2}(u'_1, u'_2, o_1, o_2; \underline{i}_1, \underline{i}_2, \underline{u}_1, \underline{u}_2))))$$

$\Rightarrow$

$$stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge$$
$$(\forall\ \underline{i}_0, \underline{i}_2, \underline{u}_0, \underline{u}_2 \bullet C_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; \underline{i}_0, \underline{i}_2, \underline{u}_0, \underline{u}_2) \tag{5.12}$$

We are done. ☺

Note that in the calculations (5.10) and (5.12), we used the most liberal recipe for composing concedes relations, i.e. the standard formula (2.6). Since we obtained the desired conclusions in that case, 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 output retrenchments which are moreover either specifically or generally closed, compose to give a single tidy resp. specifically or generally closed output retrenchment given by (4.17)-(4.20).

Proof. Theorem 4.4 tells us we have an output retrenchment, and Theorem 5.5 and Theorem 5.6 tell us that the resulting output retrenchment is specifically or generally closed respectively, so we just have to show that it is tidy. Suppose that for the composed retrenchment we had:

$$\mathrm{pre}^{\mathrm{RetA}}{}_{Op,(1,2)}(u_0, i_0) \wedge \mathrm{pre}^{\mathrm{ConA}}{}_{Op,(1,2)}(u_0, i_0) \not\equiv \mathsf{false} \tag{5.13}$$

or

$$\mathrm{pre}^{\mathrm{RetC}}{}_{Op,(1,2)}(u_2, i_2) \wedge \mathrm{pre}^{\mathrm{ConC}}{}_{Op,(1,2)}(u_2, i_2) \not\equiv \mathsf{false} \tag{5.14}$$

Let us suppose that (5.13) is true. Then we would have some u_0, i_0, such that there would be $u_{2,\mathrm{a}}$, $i_{2,\mathrm{a}}$, $u'_{0,\mathrm{a}}$, $o_{0,\mathrm{a}}$, $u'_{2,\mathrm{a}}$, $o_{2,\mathrm{a}}$, and $u_{2,\mathrm{b}}$, $i_{2,\mathrm{b}}$, $u'_{0,\mathrm{b}}$, $o_{0,\mathrm{b}}$, $u'_{2,\mathrm{b}}$, $o_{2,\mathrm{b}}$, such that we could argue as follows:

$$G_{(1,2)}(u_0, u_{2,\mathrm{a}}) \wedge P_{Op,(1,2)}(i_0, i_{2,\mathrm{a}}, u_0, u_{2,\mathrm{a}}) \wedge G_{(1,2)}(u'_{0,\mathrm{a}}, u'_{2,\mathrm{a}}) \wedge$$
$$O_{Op,(1,2)}(o_{0,\mathrm{a}}, o_{2,\mathrm{a}}; u'_{0,\mathrm{a}}, u'_{2,\mathrm{a}}, i_0, i_{2,\mathrm{a}}, u_0, u_{2,\mathrm{a}}) \wedge$$
$$stp_{Op,0}(u_0, i_0, u'_{0,\mathrm{a}}, o_{0,\mathrm{a}}) \wedge stp_{Op,2}(u_{2,\mathrm{a}}, i_{2,\mathrm{a}}, u'_{2,\mathrm{a}}, o_{2,\mathrm{a}}) \wedge$$

$G_{(1,2)}(u_0, u_{2,b}) \wedge P_{Op,(1,2)}(i_0, i_{2,b}, u_0, u_{2,b}) \wedge$
$C_{Op,(1,2)}(u'_{0,b}, u'_{2,b}, o_{0,b}, o_{2,b}; i_0, i_{2,b}, u_0, u_{2,b}) \wedge$
$stp_{Op,0}(u_0, i_0, u'_{0,b}, o_{0,b}) \wedge stp_{Op,2}(u_{2,b}, i_{2,b}, u'_{2,b}, o_{2,b})$

$\Rightarrow$ (instantiation, for some
$u_{1,aa}, i_{1,aa}, u'_{1,aa}, u_{1,ab}, i_{1,ab}, u'_{1,ab}, o_{1,a}, u_{1,ba}, i_{1,ba}, u_{1,bb}, i_{1,bb}, u'_{1,b}, o_{1,b}$)

$G_1(u_0, u_{1,aa}) \wedge G_2(u_{1,aa}, u_{2,a}) \wedge$
$P_{Op,1}(i_0, i_{1,aa}, u_0, u_{1,aa}) \wedge P_{Op,2}(i_{1,aa}, i_{2,a}, u_{1,aa}, u_{2,a}) \wedge$
$G_1(u'_{0,a}, u'_{1,aa}) \wedge G_2(u'_{1,a}, u'_{2,a}) \wedge$
$O_{Op,1}(o_{0,a}, o_{1,a}; u'_{0,a}, u'_{1,ab}, i_0, i_{1,ab}, u_0, u_{1,ab}) \wedge$
$O_{Op,2}(o_{1,a}, o_{2,a}; u'_{1,ab}, u'_{2,a}, i_{1,ab}, i_{2,a}, u_{1,ab}, u_{2,a}) \wedge$
$stp_{Op,0}(u_0, i_0, u'_{0,a}, o_{0,a}) \wedge stp_{Op,2}(u_{2,a}, i_{2,a}, u'_{2,a}, o_{2,a}) \wedge$
$G_1(u_0, u_{1,ba}) \wedge G_2(u_{1,ba}, u_{2,b}) \wedge$
$P_{Op,1}(i_0, i_{1,ba}, u_0, u_{1,ba}) \wedge P_{Op,2}(i_{1,ba}, i_{2,b}, u_{1,ba}, u_{2,b}) \wedge$
$C_{Op,1}(u'_{0,b}, u'_{1,b}, o_{0,b}, o_{1,b}; i_0, i_{1,bb}, u_0, u_{1,bb}) \wedge$
$C_{Op,2}(u'_{1,b}, u'_{2,b}, o_{1,b}, o_{2,b}; i_{1,bb}, i_{2,b}, u_{1,bb}, u_{2,b}) \wedge$
$stp_{Op,0}(u_0, i_0, u'_{0,b}, o_{0,b}) \wedge stp_{Op,2}(u_{2,b}, i_{2,b}, u'_{2,b}, o_{2,b})$

$\Rightarrow$ (closedness, getting $u_{1,aa} = u_{1,ab} = u_{1,a}$, $i_{1,aa} = i_{1,ab} = i_{1,a}$, $u'_{1,aa} = u'_{1,ab} = u'_{1,a}$,
$u_{1,ba} = u_{1,bb} = u_{1,b}$, $i_{1,ba} = i_{1,bb} = i_{1,b}$ either directly for specific closedness,
or after some modus ponens for general closedness)

$G_1(u_0, u_{1,a}) \wedge G_2(u_{1,a}, u_{2,a}) \wedge$
$P_{Op,1}(i_0, i_{1,a}, u_0, u_{1,a}) \wedge P_{Op,2}(i_{1,a}, i_{2,a}, u_{1,a}, u_{2,a}) \wedge$
$G_1(u'_{0,a}, u'_{1,a}) \wedge G_2(u'_{1,a}, u'_{2,a}) \wedge$
$O_{Op,1}(o_{0,a}, o_{1,a}; u'_{0,a}, u'_{1,a}, i_0, i_{1,a}, u_0, u_{1,a}) \wedge$
$O_{Op,2}(o_{1,a}, o_{2,a}; u'_{1,a}, u'_{2,a}, i_{1,a}, i_{2,a}, u_{1,a}, u_{2,a}) \wedge$
$stp_{Op,0}(u_0, i_0, u'_{0,a}, o_{0,a}) \wedge stp_{Op,2}(u_{2,a}, i_{2,a}, u'_{2,a}, o_{2,a}) \wedge$
$stp_{Op,1}(u_{1,a}, i_{1,a}, u'_{1,a}, o_{1,a}) \wedge$
$G_1(u_0, u_{1,b}) \wedge G_2(u_{1,b}, u_{2,b}) \wedge$
$P_{Op,1}(i_0, i_{1,b}, u_0, u_{1,b}) \wedge P_{Op,2}(i_{1,b}, i_{2,b}, u_{1,b}, u_{2,b}) \wedge$
$C_{Op,1}(u'_{0,b}, u'_{1,b}, o_{0,b}, o_{1,b}; i_0, i_{1,b}, u_0, u_{1,b}) \wedge$
$C_{Op,2}(u'_{1,b}, u'_{2,b}, o_{1,b}, o_{2,b}; i_{1,b}, i_{2,b}, u_{1,b}, u_{2,b}) \wedge$
$stp_{Op,0}(u_0, i_0, u'_{0,b}, o_{0,b}) \wedge stp_{Op,2}(u_{2,b}, i_{2,b}, u'_{2,b}, o_{2,b}) \wedge$
$stp_{Op,1}(u_{1,b}, i_{1,b}, u'_{1,b}, o_{1,b})$ (5.15)

From the presence of the intermediate level steps, we deduce that the tidiness condition (3.12) for the top level to intermediate retrenchment is violated, a contradiction. The argument for (5.14) is similar. We are done. ☺

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

Proof. We need 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, i.e.:

$$\mathrm{pre}^{\mathrm{RetA}}{}_{Op,(2,3)}(u_1, i_1) \Rightarrow \mathrm{pre}^{\mathrm{RetC}}{}_{Op,1}(u_1, i_1) \tag{5.16}$$

$$\mathrm{pre}^{\mathrm{ConA}}{}_{Op,(2,3)}(u_1, i_1) \Rightarrow \mathrm{pre}^{\mathrm{ConC}}{}_{Op,1}(u_1, i_1) \tag{5.17}$$

and

$$\mathrm{pre}^{\mathrm{RetA}}{}_{Op,3}(u_2, i_2) \Rightarrow \mathrm{pre}^{\mathrm{RetC}}{}_{Op,(1,2)}(u_2, i_2) \tag{5.18}$$

$$\mathrm{pre}^{\mathrm{ConA}}{}_{Op,3}(u_2, i_2) \Rightarrow \mathrm{pre}^{\mathrm{ConC}}{}_{Op,(1,2)}(u_2, i_2) \tag{5.19}$$

Of these we will prove (5.16) and (5.19). For the former we get:

$\mathrm{pre}^{\mathrm{RetA}}{}_{Op,(2,3)}(u_1, i_1)$

$\equiv$

$(\exists\, u'_1, o_1, u_3, i_3, u'_3, o_3 \bullet \overline{G}_{Op,(2,3)}(u'_1, u'_3, o_1, o_3; i_1, i_3, u_1, u_3))$

$\equiv$

$(\exists\, u'_1, o_1, u_3, i_3, u'_3, o_3 \bullet$
$\quad G_{(2,3)}(u_1, u_3) \wedge P_{Op,(2,3)}(i_1, i_3, u_1, u_3) \wedge$
$\quad stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge stp_{Op,3}(u_3, i_3, u'_3, o_3) \wedge$
$\quad G_{(2,3)}(u'_1, u'_3) \wedge O_{Op,(2,3)}(o_1, o_3; u'_1, u'_3, i_1, i_3, u_1, u_3))$

$\Rightarrow$ (instantiation, for some $u_{2,\mathrm{a}}$, $i_{2,\mathrm{a}}$, $u'_{2,\mathrm{a}}$, o_2, $u'_{2,\mathrm{b}}$, $i_{2,\mathrm{b}}$, $u_{2,\mathrm{b}}$)

$(\exists\, u'_1, o_1, u_3, i_3, u'_3, o_3 \bullet$
$\quad G_2(u_1, u_{2,\mathrm{a}}) \wedge G_3(u_{2,\mathrm{a}}, u_3) \wedge$
$\quad P_{Op,2}(i_1, i_{2,\mathrm{a}}, u_1, u_{2,\mathrm{a}}) \wedge P_{Op,3}(i_{2,\mathrm{a}}, i_3, u_{2,\mathrm{a}}, u_3) \wedge$
$\quad stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge stp_{Op,3}(u_3, i_3, u'_3, o_3) \wedge$
$\quad G_2(u'_1, u'_{2,\mathrm{a}}) \wedge G_3(u'_{2,\mathrm{a}}, u'_3) \wedge$
$\quad O_{Op,2}(o_1, o_2; u'_1, u'_{2,\mathrm{b}}, i_1, i_{2,\mathrm{b}}, u_1, u_{2,\mathrm{b}}) \wedge$
$\quad O_{Op,3}(o_2, o_3; u'_{2,\mathrm{b}}, u'_3, i_{2,\mathrm{b}}, i_3, u_{2,\mathrm{b}}, u_3))$

$\Rightarrow$ (closedness, inferring $u_{2,\mathrm{a}} = u_{2,\mathrm{b}} = u_2$, $i_{2,\mathrm{a}} = i_{2,\mathrm{b}} = i_2$, $u'_{2,\mathrm{a}} = u'_{2,\mathrm{b}} = u'_2$)

$(\exists\, u'_1, o_1, u_3, i_3, u'_3, o_3 \bullet$
$\quad G_2(u_1, u_2) \wedge G_3(u_2, u_3) \wedge$
$\quad P_{Op,2}(i_1, i_2, u_1, u_2) \wedge P_{Op,3}(i_2, i_3, u_2, u_3) \wedge$
$\quad stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge stp_{Op,3}(u_3, i_3, u'_3, o_3) \wedge$
$\quad G_2(u'_1, u'_2) \wedge G_3(u'_2, u'_3) \wedge$
$\quad O_{Op,2}(o_1, o_2; u'_1, u'_2, i_1, i_2, u_1, u_2) \wedge$
$\quad O_{Op,3}(o_2, o_3; u'_2, u'_3, i_2, i_3, u_2, u_3))$

$\equiv$

$(\exists\, u'_1, o_1, u_2, i_2, u'_2, o_2, u_3, i_3, u'_3, o_3 \bullet$
$\quad G_3(u_2, u_3) \wedge P_{Op,3}(i_2, i_3, u_2, u_3) \wedge$
$\quad stp_{Op,3}(u_3, i_3, u'_3, o_3) \wedge$
$\quad G_3(u'_2, u'_3) \wedge O_{Op,3}(o_2, o_3; u'_2, u'_3, i_2, i_3, u_2, u_3) \wedge$
$\quad \overline{G}_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2))$

$\Rightarrow$

$\mathrm{pre}^{\mathrm{RetA}}{}_{Op,2}(u_1, i_1)$

$\Rightarrow$ (compatible tidiness)

$$\mathrm{pre}^{\mathrm{RetC}}{}_{Op,1}(u_1, i_1) \tag{5.20}$$

as required. The argument for (5.17) is similar. For (5.19) we get:

$\text{pre}^{\text{ConA}}{}_{Op,3}(u_2, i_2)$

$\Rightarrow$ (compatible tidiness)

$\text{pre}^{\text{ConC}}{}_{Op,2}(u_2, i_2)$

$\equiv$

$(\exists\, u_1, i_1 \bullet \text{pre}^{\text{ConC}}{}_{Op,2}(u_2, i_2) \wedge \text{pre}^{\text{ConA}}{}_{Op,2}(u_1, i_1))$

$\Rightarrow$ (compatible tidiness)

$(\exists\, u_1, i_1 \bullet \text{pre}^{\text{ConC}}{}_{Op,2}(u_2, i_2) \wedge \text{pre}^{\text{ConA}}{}_{Op,2}(u_1, i_1) \wedge \text{pre}^{\text{ConC}}{}_{Op,1}(u_1, i_1))$

$\equiv$

$(\exists\, u'_2, o_2, u_1, i_1, u'_{1,\text{a}}, o_{1,\text{a}}, u'_{1,\text{b}}, o_{1,\text{b}}, u_0, i_0, u'_{0,\text{b}}, o_{0,\text{b}} \bullet$
$\quad G_2(u_1, u_2) \wedge P_{Op,2}(i_1, i_2, u_1, u_2) \wedge$
$\quad stp_{Op,1}(u_1, i_1, u'_{1,\text{a}}, o_{1,\text{a}}) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge$
$\quad C_{Op,2}(u'_{1,\text{a}}, u'_2, o_{1,\text{a}}, o_2; i_1, i_2, u_1, u_2) \wedge$
$\quad G_1(u_{0,\text{b}}, u_1) \wedge P_{Op,1}(i_{0,\text{b}}, i_1, u_{0,\text{b}}, u_1) \wedge$
$\quad stp_{Op,0}(u_{0,\text{b}}, i_{0,\text{b}}, u'_{0,\text{b}}, o_{0,\text{b}}) \wedge stp_{Op,1}(u_1, i_1, u'_{1,\text{b}}, o_{1,\text{b}}) \wedge$
$\quad C_{Op,1}(u'_{0,\text{b}}, u'_{1,\text{b}}, o_{0,\text{b}}, o_{1,\text{b}}; i_{0,\text{b}}, i_1, u_{0,\text{b}}, u_1))$

$\Rightarrow$ (operation PO, $G_1(u'_{0,\text{a}}, u'_{1,\text{a}}) \wedge O_{Op,1}(o_{0,\text{a}}, o_{1,\text{a}}; u'_{0,\text{a}}, u'_{1,\text{a}}, i_0, i_1, u_0, u_1)$
being impossible by tidiness)

$(\exists\, u'_2, o_2, u_1, i_1, u'_{1,\text{a}}, o_{1,\text{a}}, u'_{1,\text{b}}, o_{1,\text{b}}, u_0, i_0, u'_{0,\text{a}}, o_{0,\text{a}}, u'_{0,\text{b}}, o_{0,\text{b}} \bullet$
$\quad G_2(u_1, u_2) \wedge P_{Op,2}(i_1, i_2, u_1, u_2) \wedge$
$\quad stp_{Op,1}(u_1, i_1, u'_{1,\text{a}}, o_{1,\text{a}}) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge$
$\quad C_{Op,2}(u'_{1,\text{a}}, u'_2, o_{1,\text{a}}, o_2; i_1, i_2, u_1, u_2) \wedge$
$\quad G_1(u_0, u_1) \wedge P_{Op,1}(i_0, i_1, u_0, u_1) \wedge$
$\quad stp_{Op,0}(u_0, i_0, u'_{0,\text{a}}, o_{0,\text{a}}) \wedge$
$\quad C_{Op,1}(u'_{0,\text{a}}, u'_{1,\text{a}}, o_{0,\text{a}}, o_{1,\text{a}}; i_0, i_1, u_0, u_1) \wedge$
$\quad stp_{Op,0}(u_0, i_0, u'_{0,\text{b}}, o_{0,\text{b}}) \wedge stp_{Op,1}(u_1, i_1, u'_{1,\text{b}}, o_{1,\text{b}}) \wedge$
$\quad C_{Op,1}(u'_{0,\text{b}}, u'_{1,\text{b}}, o_{0,\text{b}}, o_{1,\text{b}}; i_0, i_1, u_0, u_1))$

$\Rightarrow (C_{Op,(1,2)} = (\exists\, u_1, i_1, u'_{1,\text{a}}, o_{1,\text{a}}, \bullet\, C_{Op,1} \wedge C_{Op,2}))$

$(\exists\, u'_2, o_2, u_0, i_0, u'_{0,\text{a}}, o_{0,\text{a}} \bullet$
$\quad G_{(1,2)}(u_0, u_2) \wedge P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \wedge$
$\quad stp_{Op,0}(u_0, i_0, u'_{0,\text{a}}, o_{0,\text{a}}) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge$
$\quad C_{Op,(1,2)}(u'_{0,\text{a}}, u'_2, o_{0,\text{a}}, o_2; i_0, i_2, u_0, u_2)) \wedge$
$(\exists\, u_1, i_1, u'_{1,\text{a}}, o_{1,\text{a}}, u'_{1,\text{b}}, o_{1,\text{b}}, u_0, i_0, u'_{0,\text{b}}, o_{0,\text{b}} \bullet$
$\quad stp_{Op,1}(u_1, i_1, u'_{1,\text{a}}, o_{1,\text{a}}) \wedge$
$\quad stp_{Op,0}(u_0, i_0, u'_{0,\text{b}}, o_{0,\text{b}}) \wedge stp_{Op,1}(u_1, i_1, u'_{1,\text{b}}, o_{1,\text{b}}) \wedge$
$\quad C_{Op,1}(u'_{0,\text{b}}, u'_{1,\text{b}}, o_{0,\text{b}}, o_{1,\text{b}}; i_0, i_1, u_0, u_1))$

$\Rightarrow$

$(\exists\, u'_2, o_2, u_0, i_0, u'_{0,\text{a}}, o_{0,\text{a}} \bullet \overline{C}_{Op,(1,2)}(u'_{0,\text{a}}, u'_2, o_{0,\text{a}}, o_2; i_0, i_2, u_0, u_2))$

$\equiv$

$\text{pre}^{\text{ConC}}{}_{Op,(1,2)}(u_2, i_2)$ (5.21)

as needed. The calculation for (5.18) is similar.

This done, it is now easy to check that for either association order, the expressions obtained for the composed retrieve, within, output, and concedes relations are the obvious extrapolations of (4.17)-(4.20) to three components, and are symmetrical in all three of them, for example the retrieve relation:

$$
\begin{aligned}
&G_{(1,(2,3))}(u_0, u_3) \equiv (\exists\, u_1 \bullet G_1(u_0, u_1) \wedge G_{(2,3)}(u_1, u_3)) \\
&\equiv (\exists\, u_1 \bullet G_1(u_0, u_1) \wedge (\exists\, u_2 \bullet G_2(u_1, u_2) \wedge G_3(u_2, u_3))) \\
&\equiv (\exists\, u_1, u_2 \bullet G_1(u_0, u_1) \wedge G_2(u_1, u_2) \wedge G_3(u_2, u_3)) \\
&\equiv (\exists\, u_2 \bullet (\exists\, u_1 \bullet G_1(u_0, u_1) \wedge G_2(u_1, u_2)) \wedge G_3(u_2, u_3)) \\
&\equiv (\exists\, u_2 \bullet G_{(1,2)}(u_0, u_2) \wedge G_3(u_2, u_3)) \equiv G_{((1,2),3)}(u_0, u_3) \qquad (5.22)
\end{aligned}
$$

which 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 given by (4.9)-(4.12).

Proof. 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 we just have to show that it is neat. Suppose that for the composed retrenchment we had:

$$
\mathrm{pre}^{\mathrm{Ret}}{}_{Op,(1,2)}(u_0, i_0, u_2, i_2) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,(1,2)}(u_0, i_0, u_2, i_2) \not\equiv \mathsf{false} \qquad (5.23)
$$

Instantiating the after-variables, we argue as in Theorem 6.7, except that we need no distinction between $u_{2,\mathrm{a}}$, $i_{2,\mathrm{a}}$, and $u_{2,\mathrm{b}}$, $i_{2,\mathrm{b}}$. Thus:

$$
\begin{aligned}
&G_{(1,2)}(u_0, u_2) \wedge P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \wedge G_{(1,2)}(u'_{0,\mathrm{a}}, u'_{2,\mathrm{a}}) \wedge \\
&O_{Op,(1,2)}(o_{0,\mathrm{a}}, o_{2,\mathrm{a}}; u'_{0,\mathrm{a}}, u'_{2,\mathrm{a}}, i_0, i_2, u_0, u_2) \wedge \\
&stp_{Op,0}(u_0, i_0, u'_{0,\mathrm{a}}, o_{0,\mathrm{a}}) \wedge stp_{Op,2}(u_2, i_2, u'_{2,\mathrm{a}}, o_{2,\mathrm{a}}) \wedge \\
&G_{(1,2)}(u_0, u_2) \wedge P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \wedge \\
&C_{Op,(1,2)}(u'_{0,\mathrm{b}}, u'_{2,\mathrm{b}}, o_{0,\mathrm{b}}, o_{2,\mathrm{b}}; i_0, i_2, u_0, u_2) \wedge \\
&stp_{Op,0}(u_0, i_0, u'_{0,\mathrm{b}}, o_{0,\mathrm{b}}) \wedge stp_{Op,2}(u_2, i_2, u'_{2,\mathrm{b}}, o_{2,\mathrm{b}}) \\
&\Rightarrow (\text{instantiation, for some } u_1, i_1, u'_{1,\mathrm{a}}, u_{1,\mathrm{a}}, i_{1,\mathrm{a}}, u'_{1,\mathrm{aa}}, o_{1,\mathrm{a}}, u_{1,\mathrm{b}}, i_{1,\mathrm{b}}, u'_{1,\mathrm{b}}, o_{1,\mathrm{b}}) \\
&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) \wedge \\
&G_1(u'_{0,\mathrm{a}}, u'_{1,\mathrm{a}}) \wedge G_2(u'_{1,\mathrm{a}}, u'_{2,\mathrm{a}}) \wedge \\
&O_{Op,1}(o_{0,\mathrm{a}}, o_{1,\mathrm{a}}; u'_{0,\mathrm{a}}, u'_{1,\mathrm{aa}}, i_0, i_{1,\mathrm{a}}, u_0, u_{1,\mathrm{a}}) \wedge \\
&O_{Op,2}(o_{1,\mathrm{a}}, o_{2,\mathrm{a}}; u'_{1,\mathrm{aa}}, u'_{2,\mathrm{a}}, i_{1,\mathrm{a}}, i_2, u_{1,\mathrm{a}}, u_2) \wedge \\
&\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) \wedge \\
&stp_{Op,0}(u_0, i_0, u'_{0,\mathrm{a}}, o_{0,\mathrm{a}}) \wedge stp_{Op,2}(u_2, i_2, u'_{2,\mathrm{a}}, o_{2,\mathrm{a}}) \wedge \\
&((G_1(u'_{0,\mathrm{b}}, u'_{1,\mathrm{b}}) \wedge O_{Op,1}(o_{0,\mathrm{b}}, o_{1,\mathrm{b}}; u'_{0,\mathrm{b}}, u'_{1,\mathrm{b}}, i_0, i_{1,\mathrm{b}}, u_0, u_{1,\mathrm{b}}) \wedge \\
&\quad C_{Op,2}(u'_{1,\mathrm{b}}, u'_{2,\mathrm{b}}, o_{1,\mathrm{b}}, o_{2,\mathrm{b}}; i_{1,\mathrm{b}}, i_2, u_{1,\mathrm{b}}, u_2) \wedge \\
&\quad \mathrm{pre}^{\mathrm{Ret}}{}_{Op,1}(u_0, i_0, u_{1,\mathrm{b}}, i_{1,\mathrm{b}}) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_{1,\mathrm{b}}, i_{1,\mathrm{b}}, u_2, i_2)) \vee \\
&(C_{Op,1}(u'_{0,\mathrm{b}}, u'_{1,\mathrm{b}}, o_{0,\mathrm{b}}, o_{1,\mathrm{b}}; i_0, i_{1,\mathrm{b}}, u_0, u_{1,\mathrm{b}}) \wedge
\end{aligned}
$$

$$
\begin{array}{l}
\quad G_2(u'_{1,\mathrm{b}}, u'_{2,\mathrm{b}}) \wedge O_{Op,2}(o_{1,\mathrm{b}}, o_{2,\mathrm{b}}; u'_{1,\mathrm{b}}, u'_{2,\mathrm{b}}, i_{1,\mathrm{b}}, i_2, u_{1,\mathrm{b}}, u_2) \wedge \\
\quad \mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_{1,\mathrm{b}}, i_{1,\mathrm{b}}) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,2}(u_{1,\mathrm{b}}, i_{1,\mathrm{b}}, u_2, i_2)) \vee \\
(C_{Op,1}(u'_{0,\mathrm{b}}, u'_{1,\mathrm{b}}, o_{0,\mathrm{b}}, o_{1,\mathrm{b}}; i_0, i_{1,\mathrm{b}}, u_0, u_{1,\mathrm{b}}) \wedge \\
\quad C_{Op,2}(u'_{1,\mathrm{b}}, u'_{2,\mathrm{b}}, o_{1,\mathrm{b}}, o_{2,\mathrm{b}}; i_{1,\mathrm{b}}, i_2, u_{1,\mathrm{b}}, u_2) \wedge \\
\quad \mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_{1,\mathrm{b}}, i_{1,\mathrm{b}}) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_{1,\mathrm{b}}, i_{1,\mathrm{b}}, u_2, i_2))) \wedge \\
stp_{Op,0}(u_0, i_0, u'_{0,\mathrm{b}}, o_{0,\mathrm{b}}) \wedge stp_{Op,2}(u_2, i_2, u'_{2,\mathrm{b}}, o_{2,\mathrm{b}})
\end{array}
$$

$\Rightarrow$ (closedness, inferring $u_{1,\mathrm{a}} = u_{1,\mathrm{b}} = u_1$, $i_{1,\mathrm{a}} = i_{1,\mathrm{b}} = i_1$, $u'_{1,\mathrm{aa}} = u'_{1,\mathrm{a}}$, either directly for specific closedness, or after some modus ponens for general closedness)

$$
\begin{array}{l}
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) \wedge \\
G_1(u'_{0,\mathrm{a}}, u'_{1,\mathrm{a}}) \wedge G_2(u'_{1,\mathrm{a}}, u'_{2,\mathrm{a}}) \wedge \\
O_{Op,1}(o_{0,\mathrm{a}}, o_{1,\mathrm{a}}; u'_{0,\mathrm{a}}, u'_{1,\mathrm{a}}, i_0, i_1, u_0, u_1) \wedge \\
O_{Op,2}(o_{1,\mathrm{a}}, o_{2,\mathrm{a}}; u'_{1,\mathrm{a}}, u'_{2,\mathrm{a}}, 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{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2) \wedge \\
stp_{Op,0}(u_0, i_0, u'_{0,\mathrm{a}}, o_{0,\mathrm{a}}) \wedge stp_{Op,2}(u_2, i_2, u'_{2,\mathrm{a}}, o_{2,\mathrm{a}}) \wedge \\
stp_{Op,1}(u_1, i_1, u'_{1,\mathrm{a}}, o_{1,\mathrm{a}}) \wedge \\
((G_1(u'_{0,\mathrm{b}}, u'_{1,\mathrm{b}}) \wedge O_{Op,1}(o_{0,\mathrm{b}}, o_{1,\mathrm{b}}; u'_{0,\mathrm{b}}, u'_{1,\mathrm{b}}, i_0, i_1, u_0, u_1) \wedge \\
\quad C_{Op,2}(u'_{1,\mathrm{b}}, u'_{2,\mathrm{b}}, o_{1,\mathrm{b}}, o_{2,\mathrm{b}}; i_1, i_2, u_1, u_2) \wedge \\
\quad \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,\mathrm{b}}, u'_{1,\mathrm{b}}, o_{0,\mathrm{b}}, o_{1,\mathrm{b}}; i_0, i_1, u_0, u_1) \wedge \\
\quad G_2(u'_{1,\mathrm{b}}, u'_{2,\mathrm{b}}) \wedge O_{Op,2}(o_{1,\mathrm{b}}, o_{2,\mathrm{b}}; u'_{1,\mathrm{b}}, u'_{2,\mathrm{b}}, i_1, i_2, u_1, u_2) \wedge \\
\quad \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,\mathrm{b}}, u'_{1,\mathrm{b}}, o_{0,\mathrm{b}}, o_{1,\mathrm{b}}; i_0, i_1, u_0, u_1) \wedge \\
\quad C_{Op,2}(u'_{1,\mathrm{b}}, u'_{2,\mathrm{b}}, o_{1,\mathrm{b}}, o_{2,\mathrm{b}}; i_1, i_2, u_1, u_2) \wedge \\
\quad \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))) \wedge \\
stp_{Op,0}(u_0, i_0, u'_{0,\mathrm{b}}, o_{0,\mathrm{b}}) \wedge stp_{Op,2}(u_2, i_2, u'_{2,\mathrm{b}}, o_{2,\mathrm{b}}) \wedge \\
stp_{Op,1}(u_1, i_1, u'_{1,\mathrm{b}}, o_{1,\mathrm{b}})
\end{array}
\tag{5.24}
$$

Now we consider the three disjuncts that would be obtained from the above via the distributive law. From the first we have $\mathrm{pre}^{\mathrm{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2)$ which contradicts the neatness of the intermediate to lowest level retrenchment. From the second, we deduce $\mathrm{pre}^{\mathrm{Ret}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1)$ contradicting the neatness of the top level to intermediate retrenchment. For the third disjunct we have a choice of two such arguments. So altogether, we deduce false, and the composite is neat. ☺

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:

$$
\begin{array}{l}
\mathrm{pre}^{\mathrm{Ret}}{}_{Op,(1,2)}(u_0, i_0, u_2, i_2) \equiv \\
\quad (\exists\, u_1, i_1 \bullet \mathrm{pre}^{\mathrm{Ret}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2))
\end{array}
\tag{5.25}
$$

$$
\begin{aligned}
&\mathrm{pre}^{\mathrm{Con}}{}_{Op,(1,2)}(u_0, i_0, u_2, i_2) \equiv \\
&\quad (\exists\, u_1, i_1 \bullet \\
&\qquad (\mathrm{pre}^{\mathrm{Ret}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2)) \vee \\
&\qquad (\mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2)) \vee \\
&\qquad (\mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2)))
\end{aligned}
\tag{5.26}
$$

Proof. For $\mathrm{pre}^{\mathrm{Ret}}{}_{Op,(1,2)}(u_0, i_0, u_2, i_2)$ we calculate:

$$
\begin{aligned}
&\mathrm{pre}^{\mathrm{Ret}}{}_{Op,(1,2)}(u_0, i_0, u_2, i_2) \\
&\equiv \\
&(\exists\, u'_0, o_0, u'_2, o_2 \bullet \overline{G}_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2)) \\
&\equiv \\
&(\exists\, u'_0, o_0, u'_2, o_2 \bullet \\
&\quad G_{(1,2)}(u_0, u_2) \wedge P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \wedge \\
&\quad stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge \\
&\quad 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)) \\
&\equiv \\
&(\exists\, u'_0, o_0, u'_2, o_2 \bullet \\
&\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)) \wedge \\
&\quad stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge \\
&\quad (\exists\, u'_1 \bullet G_1(u'_0, u'_1) \wedge G_2(u'_1, u'_2)) \wedge \\
&\quad (\exists\, \underline{u}'_1, o_1, \underline{u}_1, \underline{i}_1 \bullet O_{Op,1}(o_0, o_1; u'_0, \underline{u}'_1, i_0, \underline{i}_1, u_0, \underline{u}_1) \wedge \\
&\qquad O_{Op,2}(o_1, o_2; \underline{u}'_1, u'_2, \underline{i}_1, i_2, \underline{u}_1, u_2))) \\
&\Rightarrow \text{(closedness, inferring as usual } \underline{u}_1 = u_1, \underline{i}_1 = i_1, \underline{u}'_1 = u'_1) \\
&(\exists\, u_1, i_1, u'_0, o_0, u'_1, o_1, u'_2, o_2 \bullet \\
&\quad 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) \wedge \\
&\quad stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge \\
&\quad G_1(u'_0, u'_1) \wedge G_2(u'_1, u'_2) \wedge \\
&\quad 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)) \\
&\Rightarrow \\
&(\exists\, u_1, i_1 \bullet \\
&\quad (\exists\, u'_0, o_0, u'_1, o_1 \bullet G_1(u_0, u_1) \wedge P_{Op,1}(i_0, i_1, u_0, u_1) \wedge \\
&\qquad stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge \\
&\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 \\
&\quad (\exists\, u'_1, o_1, u'_2, o_2 \bullet G_2(u_1, u_2) \wedge P_{Op,2}(i_1, i_2, u_1, u_2) \wedge \\
&\qquad stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge \\
&\qquad 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))) \\
&\equiv \\
&(\exists\, u_1, i_1 \bullet \\
&\quad (\exists\, u'_0, o_0, u'_1, o_1 \bullet \overline{G}_{Op}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1)) \wedge \\
&\quad (\exists\, u'_1, o_1, u'_2, o_2 \bullet \overline{G}_{Op}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2)))
\end{aligned}
$$

$$\equiv$$

$$(\exists\, u_1, i_1 \bullet \text{pre}^{\text{Ret}}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \text{pre}^{\text{Ret}}_{Op,2}(u_1, i_1, u_2, i_2)) \tag{5.27}$$

For the converse we have:

$$(\exists\, u_1, i_1 \bullet \text{pre}^{\text{Ret}}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \text{pre}^{\text{Ret}}_{Op,2}(u_1, i_1, u_2, i_2))$$

$$\equiv$$

$$\begin{array}{l}(\exists\, u_1, i_1 \bullet \\ \quad (\exists\, u'_0, o_0, u'_{1,\text{a}}, o_{1,\text{a}} \bullet \overline{G}_{Op}(u'_0, u'_{1,\text{a}}, o_0, o_{1,\text{a}}; i_0, i_1, u_0, u_1)) \wedge \\ \quad (\exists\, u'_{1,\text{b}}, o_{1,\text{b}}, u'_2, o_2 \bullet \overline{G}_{Op}(u'_{1\text{b},}, u'_2, o_{1,\text{b}}, o_2; i_1, i_2, u_1, u_2)))\end{array}$$

$$\equiv$$

$$\begin{array}{l}(\exists\, u_1, i_1 \bullet \\ \quad (\exists\, u'_0, o_0, u'_{1,\text{a}}, o_{1,\text{a}} \bullet G_1(u_0, u_1) \wedge P_{Op,1}(i_0, i_1, u_0, u_1) \wedge \\ \qquad stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,1}(u_1, i_1, u'_{1,\text{a}}, o_{1,\text{a}}) \wedge \\ \qquad G_1(u'_0, u'_{1,\text{a}}) \wedge O_{Op,1}(o_0, o_{1,\text{a}}; u'_0, u'_{1,\text{a}}, i_0, i_1, u_0, u_1)) \wedge \\ \quad (\exists\, u'_{1,\text{b}}, o_{1,\text{b}}, u'_2, o_2 \bullet G_2(u_1, u_2) \wedge P_{Op,2}(i_1, i_2, u_1, u_2) \wedge \\ \qquad stp_{Op,1}(u_1, i_1, u'_{1,\text{b}}, o_{1,\text{b}}) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge \\ \qquad G_2(u'_{1,\text{b}}, u'_2) \wedge O_{Op,2}(o_{1,\text{b}}, o_2; u'_{1,\text{b}}, u'_2, i_1, i_2, u_1, u_2)))\end{array}$$

$$\Rightarrow \text{(determinism: } u'_{1,\text{a}} = u'_{1,\text{b}} = u'_1,\ o_{1,\text{a}} = o_{1,\text{b}} = o_1)$$

$$\begin{array}{l}(\exists\, u_1, i_1, u'_0, o_0, u'_1, o_1, u'_2, o_2 \bullet \\ \quad G_1(u_0, u_1) \wedge P_{Op,1}(i_0, i_1, u_0, u_1) \wedge \\ \qquad stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge \\ \qquad G_1(u'_0, u'_1) \wedge O_{Op,1}(o_0, o_1; u'_0, u'_1, i_0, i_1, u_0, u_1) \wedge \\ \quad G_2(u_1, u_2) \wedge P_{Op,2}(i_1, i_2, u_1, u_2) \wedge \\ \qquad stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge \\ \qquad G_2(u'_1, u'_2) \wedge O_{Op,2}(o_1, o_2; u'_1, u'_2, i_1, i_2, u_1, u_2))\end{array}$$

$$\Rightarrow$$

$$\begin{array}{l}(\exists\, u'_0, o_0, u'_2, o_2 \bullet \\ \quad (\exists\, u_1, i_1 \bullet \\ \qquad G_1(u_0, u_1) \wedge G_2(u_1, u_2) \wedge P_{Op,1}(i_0, i_1, u_0, u_1) \wedge P_{Op,2}(i_1, i_2, u_1, u_2)) \wedge \\ \quad stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2)) \wedge \\ \quad (\exists\, u'_1 \bullet G_1(u'_0, u'_1) \wedge G_2(u'_1, u'_2)) \wedge \\ \quad (\exists\, u'_1, o_1, u_1, i_1 \bullet O_{Op,1}(o_0, o_1; u'_0, u'_1, i_0, i_1, u_0, u_1) \wedge \\ \qquad O_{Op,2}(o_1, o_2; u'_1, u'_2, i_1, i_2, u_1, u_2)))\end{array}$$

$$\equiv$$

$$\begin{array}{l}(\exists\, u'_0, o_0, u'_2, o_2 \bullet \\ \quad G_{(1,2)}(u_0, u_2) \wedge P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \wedge \\ \quad stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge \\ \quad G_{(1,2)}(u'_0, u'_2) \wedge O_{Op,(1,2)}(o_0, o_2; u'_0, u'_2, i_0, i_2, u_0, u_2))\end{array}$$

$$\equiv$$

$$(\exists\, u'_0, o_0, u'_2, o_2 \bullet \overline{G}_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2))$$

$$\equiv$$

$$\mathrm{pre}^{\mathrm{Ret}}{}_{Op,(1,2)}(u_0, i_0, u_2, i_2) \tag{5.28}$$

For $\mathrm{pre}^{\mathrm{Con}}{}_{Op,(1,2)}(u_0, i_0, u_2, i_2)$ we have a similar calculation:

$\mathrm{pre}^{\mathrm{Con}}{}_{Op,(1,2)}(u_0, i_0, u_2, i_2)$

$\equiv$

$(\exists\, u'_0, o_0, u'_2, o_2 \bullet \overline{C}_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2))$

$\equiv$

$$\begin{array}{l}(\exists\, u'_0, o_0, u'_2, o_2 \bullet \\ \quad G_{(1,2)}(u_0, u_2) \wedge P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \wedge \\ \quad stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge \\ \quad C_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2))\end{array}$$

$\equiv$

$$\begin{array}{l}(\exists\, u'_0, o_0, u'_2, o_2 \bullet \\ \quad (\exists\, u_1, i_1 \bullet G_1(u_0, u_1) \wedge G_2(u_1, u_2) \wedge \\ \qquad P_{Op,1}(i_0, i_1, u_0, u_1) \wedge P_{Op,2}(i_1, i_2, u_1, u_2)) \wedge \\ \quad stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge \\ \quad (\exists\, u'_1, o_1, \underline{u}_1, \underline{i}_1 \bullet \\ \qquad (G_1(u'_0, u'_1) \wedge O_{Op,1}(o_0, o_1; u'_0, u'_1, i_0, \underline{i}_1, u_0, \underline{u}_1) \wedge \\ \qquad\quad C_{Op,2}(u'_1, u'_2, o_1, o_2; \underline{i}_1, i_2, \underline{u}_1, u_2) \wedge \\ \qquad\quad \mathrm{pre}^{\mathrm{Ret}}{}_{Op,1}(u_0, i_0, \underline{u}_1, \underline{i}_1) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(\underline{u}_1, \underline{i}_1, u_2, i_2)) \vee \\ \qquad (C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, \underline{i}_1, u_0, \underline{u}_1) \wedge \\ \qquad\quad G_2(u'_1, u'_2) \wedge O_{Op,2}(o_1, o_2; u'_1, u'_2, \underline{i}_1, i_2, \underline{u}_1, u_2) \wedge \\ \qquad\quad \mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, \underline{u}_1, \underline{i}_1) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,2}(\underline{u}_1, \underline{i}_1, u_2, i_2)) \vee \\ \qquad (C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, \underline{i}_1, u_0, \underline{u}_1) \wedge \\ \qquad\quad C_{Op,2}(u'_1, u'_2, o_1, o_2; \underline{i}_1, i_2, \underline{u}_1, u_2) \wedge \\ \qquad\quad \mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, \underline{u}_1, \underline{i}_1) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(\underline{u}_1, \underline{i}_1, u_2, i_2))))\end{array}$$

$\Rightarrow$ (closedness, inferring $\underline{u}_1 = u_1$, $\underline{i}_1 = i_1$, in each disjunct)

$$\begin{array}{l}(\exists\, u_1, i_1, u'_0, o_0, u'_1, o_1, u'_2, o_2 \bullet \\ \quad G_1(u_0, u_1) \wedge G_2(u_1, u_2) \wedge P_{Op,1}(i_0, i_1, u_0, u_1) \wedge P_{Op,2}(i_1, i_2, u_1, u_2) \wedge \\ \quad stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge \\ \quad (G_1(u'_0, u'_1) \wedge O_{Op,1}(o_0, o_1; u'_0, u'_1, i_0, i_1, u_0, u_1) \wedge \\ \qquad C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2) \wedge \\ \qquad \mathrm{pre}^{\mathrm{Ret}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2)) \vee \\ \quad (C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \\ \qquad G_2(u'_1, u'_2) \wedge O_{Op,2}(o_1, o_2; u'_1, u'_2, i_1, i_2, u_1, u_2) \wedge \\ \qquad \mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2)) \vee \\ \quad (C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \\ \qquad C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2) \wedge \\ \qquad \mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2)))\end{array}$$

$\Rightarrow$

$$\begin{array}{l}(\exists\, u_1, i_1 \bullet \\ \quad (\exists\, u'_0, o_0, u'_1, o_1, u'_2, o_2 \bullet\end{array}$$

$$
\begin{array}{l}
\quad G_1(u_0, u_1) \wedge P_{Op,1}(i_0, i_1, u_0, u_1) \wedge \\
\quad stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge \\
\quad G_1(u'_0, u'_1) \wedge O_{Op,1}(o_0, o_1; u'_0, u'_1, i_0, i_1, u_0, u_1) \wedge \\
\quad G_2(u_1, u_2) \wedge P_{Op,2}(i_1, i_2, u_1, u_2) \wedge \\
\quad stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge \\
\quad C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2) \wedge \\
\quad \mathrm{pre}^{\mathrm{Ret}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2)) \vee \\
(\exists\, u'_0, o_0, u'_1, o_1, u'_2, o_2 \bullet \\
\quad G_1(u_0, u_1) \wedge P_{Op,1}(i_0, i_1, u_0, u_1) \wedge \\
\quad stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge \\
\quad C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \\
\quad G_2(u_1, u_2) \wedge P_{Op,2}(i_1, i_2, u_1, u_2) \wedge \\
\quad stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge \\
\quad G_2(u'_1, u'_2) \wedge O_{Op,2}(o_1, o_2; u'_1, u'_2, i_1, i_2, u_1, u_2) \wedge \\
\quad \mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2)) \vee \\
(\exists\, u'_0, o_0, u'_1, o_1, u'_2, o_2 \bullet \\
\quad G_1(u_0, u_1) \wedge P_{Op,1}(i_0, i_1, u_0, u_1) \wedge \\
\quad stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge \\
\quad C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \\
\quad G_2(u_1, u_2) \wedge P_{Op,2}(i_1, i_2, u_1, u_2) \wedge \\
\quad stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge \\
\quad C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2) \wedge \\
\quad \mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2)))
\end{array}
$$

$$\Rightarrow$$

$$
\begin{array}{l}
(\exists\, u_1, i_1 \bullet \\
\quad ((\exists\, u'_0, o_0, u'_1, o_1 \bullet \\
\quad\quad G_1(u_0, u_1) \wedge P_{Op,1}(i_0, i_1, u_0, u_1) \wedge \\
\quad\quad stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge \\
\quad\quad G_1(u'_0, u'_1) \wedge O_{Op,1}(o_0, o_1; u'_0, u'_1, i_0, i_1, u_0, u_1) \wedge \\
\quad\quad \mathrm{pre}^{\mathrm{Ret}}{}_{Op,1}(u_0, i_0, u_1, i_1)) \wedge \\
\quad (\exists\, u'_1, o_1, u'_2, o_2 \bullet \\
\quad\quad G_2(u_1, u_2) \wedge P_{Op,2}(i_1, i_2, u_1, u_2) \wedge \\
\quad\quad stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge \\
\quad\quad C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2) \wedge \\
\quad\quad \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2))) \vee \\
\quad ((\exists\, u'_0, o_0, u'_1, o_1 \bullet \\
\quad\quad G_1(u_0, u_1) \wedge P_{Op,1}(i_0, i_1, u_0, u_1) \wedge \\
\quad\quad stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge \\
\quad\quad C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \\
\quad\quad \mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1)) \wedge \\
\quad (\exists\, u'_1, o_1, u'_2, o_2 \bullet \\
\quad\quad G_2(u_1, u_2) \wedge P_{Op,2}(i_1, i_2, u_1, u_2) \wedge \\
\quad\quad stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge \\
\quad\quad G_2(u'_1, u'_2) \wedge O_{Op,2}(o_1, o_2; u'_1, u'_2, i_1, i_2, u_1, u_2) \wedge \\
\quad\quad \mathrm{pre}^{\mathrm{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2))) \vee
\end{array}
$$

$$
\begin{array}{l}
\quad ((\exists\, u'_0, o_0, u'_1, o_1 \bullet \\
\qquad G_1(u_0, u_1) \wedge P_{Op,1}(i_0, i_1, u_0, u_1) \wedge \\
\qquad stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge \\
\qquad C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \\
\qquad \mathrm{pre}^{\mathrm{Con}}_{Op,1}(u_0, i_0, u_1, i_1)) \wedge \\
\quad (\exists\, u'_1, o_1, u'_2, o_2 \bullet \\
\qquad G_2(u_1, u_2) \wedge P_{Op,2}(i_1, i_2, u_1, u_2) \wedge \\
\qquad stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge \\
\qquad C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2) \wedge \\
\qquad \mathrm{pre}^{\mathrm{Con}}_{Op,2}(u_1, i_1, u_2, i_2))))
\end{array}
$$

$$\equiv$$

$$
\begin{array}{l}
(\exists\, u_1, i_1 \bullet \\
\quad ((\exists\, u'_0, o_0, u'_1, o_1 \bullet \overline{G}_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \\
\qquad \mathrm{pre}^{\mathrm{Ret}}_{Op,1}(u_0, i_0, u_1, i_1)) \wedge \\
\quad (\exists\, u'_1, o_1, u'_2, o_2 \bullet \overline{C}_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2) \wedge \\
\qquad \mathrm{pre}^{\mathrm{Con}}_{Op,2}(u_1, i_1, u_2, i_2))) \vee \\
\quad ((\exists\, u'_0, o_0, u'_1, o_1 \bullet \overline{C}_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \\
\qquad \mathrm{pre}^{\mathrm{Con}}_{Op,1}(u_0, i_0, u_1, i_1)) \wedge \\
\quad (\exists\, u'_1, o_1, u'_2, o_2 \bullet \overline{G}_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2) \wedge \\
\qquad \mathrm{pre}^{\mathrm{Ret}}_{Op,2}(u_1, i_1, u_2, i_2))) \vee \\
\quad ((\exists\, u'_0, o_0, u'_1, o_1 \bullet \overline{C}_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \\
\qquad \mathrm{pre}^{\mathrm{Con}}_{Op,1}(u_0, i_0, u_1, i_1)) \wedge \\
\quad (\exists\, u'_1, o_1, u'_2, o_2 \bullet \overline{C}_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2) \wedge \\
\qquad \mathrm{pre}^{\mathrm{Con}}_{Op,2}(u_1, i_1, u_2, i_2))))
\end{array}
$$

$$\equiv$$

$$
\begin{array}{l}
(\exists\, u_1, i_1 \bullet \\
\quad (\mathrm{pre}^{\mathrm{Ret}}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Con}}_{Op,2}(u_1, i_1, u_2, i_2)) \vee \\
\quad (\mathrm{pre}^{\mathrm{Con}}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Ret}}_{Op,2}(u_1, i_1, u_2, i_2)) \vee \\
\quad (\mathrm{pre}^{\mathrm{Con}}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Con}}_{Op,2}(u_1, i_1, u_2, i_2)))
\end{array}
\tag{5.29}
$$

And for the converse we have:

$$
\begin{array}{l}
(\exists\, u_1, i_1 \bullet \\
\quad (\mathrm{pre}^{\mathrm{Ret}}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Con}}_{Op,2}(u_1, i_1, u_2, i_2)) \vee \\
\quad (\mathrm{pre}^{\mathrm{Con}}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Ret}}_{Op,2}(u_1, i_1, u_2, i_2)) \vee \\
\quad (\mathrm{pre}^{\mathrm{Con}}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Con}}_{Op,2}(u_1, i_1, u_2, i_2)))
\end{array}
$$

$$\equiv$$

$$
\begin{array}{l}
(\exists\, u_1, i_1 \bullet \\
\quad ((\exists\, u'_0, o_0, u'_{1,\mathrm{a}}, o_{1,\mathrm{a}} \bullet \overline{G}_{Op,1}(u'_0, u'_{1,\mathrm{a}}, o_0, o_{1,\mathrm{a}}; i_0, i_1, u_0, u_1) \wedge \\
\qquad \mathrm{pre}^{\mathrm{Ret}}_{Op,1}(u_0, i_0, u_1, i_1)) \wedge \\
\quad (\exists\, u'_{1,\mathrm{b}}, o_{1,\mathrm{b}}, u'_2, o_2 \bullet \overline{C}_{Op,2}(u'_{1,\mathrm{b}}, u'_2, o_{1,\mathrm{b}}, o_2; i_1, i_2, u_1, u_2) \wedge \\
\qquad \mathrm{pre}^{\mathrm{Con}}_{Op,2}(u_1, i_1, u_2, i_2))) \vee \\
\quad ((\exists\, u'_0, o_0, u'_{1,\mathrm{a}}, o_{1,\mathrm{a}} \bullet \overline{C}_{Op,1}(u'_0, u'_{1,\mathrm{a}}, o_0, o_{1,\mathrm{a}}; i_0, i_1, u_0, u_1) \wedge \\
\qquad \mathrm{pre}^{\mathrm{Con}}_{Op,1}(u_0, i_0, u_1, i_1)) \wedge
\end{array}
$$

$(\exists\, u'_{1,\mathrm{b}}, o_{1,\mathrm{b}}, u'_2, o_2 \bullet \overline{G}_{Op,2}(u'_{1,\mathrm{b}}, u'_2, o_{1,\mathrm{b}}, o_2; i_1, i_2, u_1, u_2) \wedge$
$\quad \mathrm{pre}^{\mathrm{Ret}}_{Op,2}(u_1, i_1, u_2, i_2))) \vee$
$((\exists\, u'_0, o_0, u'_{1,\mathrm{a}}, o_{1,\mathrm{a}} \bullet \overline{C}_{Op,1}(u'_0, u'_{1,\mathrm{a}}, o_0, o_{1,\mathrm{a}}; i_0, i_1, u_0, u_1) \wedge$
$\quad \mathrm{pre}^{\mathrm{Con}}_{Op,1}(u_0, i_0, u_1, i_1)) \wedge$
$(\exists\, u'_{1,\mathrm{b}}, o_{1,\mathrm{b}}, u'_2, o_2 \bullet \overline{C}_{Op,2}(u'_{1,\mathrm{b}}, u'_2, o_{1,\mathrm{b}}, o_2; i_1, i_2, u_1, u_2) \wedge$
$\quad \mathrm{pre}^{\mathrm{Con}}_{Op,2}(u_1, i_1, u_2, i_2))))$

$\equiv$

$(\exists\, u_1, i_1 \bullet$
$\quad ((\exists\, u'_0, o_0, u'_{1,\mathrm{a}}, o_{1,\mathrm{a}} \bullet$
$\quad\quad G_1(u_0, u_1) \wedge P_{Op,1}(i_0, i_1, u_0, u_1) \wedge$
$\quad\quad G_1(u'_0, u'_{1,\mathrm{a}}) \wedge O_{Op,1}(o_0, o_{1,\mathrm{a}}; u'_0, u'_{1,\mathrm{a}}, i_0, i_1, u_0, u_1) \wedge$
$\quad\quad stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,1}(u_1, i_1, u'_{1,\mathrm{a}}, o_{1,\mathrm{a}}) \wedge$
$\quad\quad \mathrm{pre}^{\mathrm{Ret}}_{Op,1}(u_0, i_0, u_1, i_1)) \wedge$
$\quad (\exists\, u'_{1,\mathrm{b}}, o_{1,\mathrm{b}}, u'_2, o_2 \bullet$
$\quad\quad G_2(u_1, u_2) \wedge P_{Op,2}(i_1, i_2, u_1, u_2) \wedge$
$\quad\quad C_{Op,2}(u'_{1,\mathrm{b}}, u'_2, o_{1,\mathrm{b}}, o_2; i_1, i_2, u_1, u_2) \wedge$
$\quad\quad stp_{Op,1}(u_1, i_1, u'_{1,\mathrm{b}}, o_{1,\mathrm{b}}) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge$
$\quad\quad \mathrm{pre}^{\mathrm{Con}}_{Op,2}(u_1, i_1, u_2, i_2))) \vee$
$\quad ((\exists\, u'_0, o_0, u'_{1,\mathrm{a}}, o_{1,\mathrm{a}} \bullet$
$\quad\quad G_1(u_0, u_1) \wedge P_{Op,1}(i_0, i_1, u_0, u_1) \wedge$
$\quad\quad C_{Op,1}(u'_0, u'_{1,\mathrm{a}}, o_0, o_{1,\mathrm{a}}; i_0, i_1, u_0, u_1) \wedge$
$\quad\quad stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,1}(u_1, i_1, u'_{1,\mathrm{a}}, o_{1,\mathrm{a}}) \wedge$
$\quad\quad \mathrm{pre}^{\mathrm{Con}}_{Op,1}(u_0, i_0, u_1, i_1)) \wedge$
$\quad (\exists\, u'_{1,\mathrm{b}}, o_{1,\mathrm{b}}, u'_2, o_2 \bullet$
$\quad\quad G_2(u_1, u_2) \wedge P_{Op,2}(i_1, i_2, u_1, u_2) \wedge$
$\quad\quad G_2(u'_{1,\mathrm{b}}, u'_2) \wedge O_{Op,2}(o_{1,\mathrm{b}}, o_2; u'_{1,\mathrm{b}}, u'_2, i_1, i_2, u_1, u_2) \wedge$
$\quad\quad stp_{Op,1}(u_1, i_1, u'_{1,\mathrm{b}}, o_{1,\mathrm{b}}) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge$
$\quad\quad \mathrm{pre}^{\mathrm{Ret}}_{Op,2}(u_1, i_1, u_2, i_2))) \vee$
$\quad ((\exists\, u'_0, o_0, u'_{1,\mathrm{a}}, o_{1,\mathrm{a}} \bullet$
$\quad\quad G_1(u_0, u_1) \wedge P_{Op,1}(i_0, i_1, u_0, u_1) \wedge$
$\quad\quad C_{Op,1}(u'_0, u'_{1,\mathrm{a}}, o_0, o_{1,\mathrm{a}}; i_0, i_1, u_0, u_1) \wedge$
$\quad\quad stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,1}(u_1, i_1, u'_{1,\mathrm{a}}, o_{1,\mathrm{a}}) \wedge$
$\quad\quad \mathrm{pre}^{\mathrm{Con}}_{Op,1}(u_0, i_0, u_1, i_1)) \wedge$
$\quad (\exists\, u'_{1,\mathrm{b}}, o_{1,\mathrm{b}}, u'_2, o_2 \bullet$
$\quad\quad G_2(u_1, u_2) \wedge P_{Op,2}(i_1, i_2, u_1, u_2) \wedge$
$\quad\quad C_{Op,2}(u'_{1,\mathrm{b}}, u'_2, o_{1,\mathrm{b}}, o_2; i_1, i_2, u_1, u_2) \wedge$
$\quad\quad stp_{Op,1}(u_1, i_1, u'_{1,\mathrm{b}}, o_{1,\mathrm{b}}) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge$
$\quad\quad \mathrm{pre}^{\mathrm{Con}}_{Op,2}(u_1, i_1, u_2, i_2))))$

$\Rightarrow$ (determinism: $u'_{1,\mathrm{a}} = u'_{1,\mathrm{b}} = u'_1$, $o_{1,\mathrm{a}} = o_{1,\mathrm{b}} = o_1$ in all disjuncts)

$(\exists\, u_1, i_1 \bullet$
$\quad G_1(u_0, u_1) \wedge G_2(u_1, u_2) \wedge P_{Op,1}(i_0, i_1, u_0, u_1) \wedge P_{Op,2}(i_1, i_2, u_1, u_2) \wedge$
$\quad ((\exists\, u'_0, o_0, u'_1, o_1, u'_2, o_2 \bullet$
$\quad\quad G_1(u'_0, u'_1) \wedge O_{Op,1}(o_0, o_1; u'_0, u'_1, i_0, i_1, u_0, u_1) \wedge$
$\quad\quad stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge$

$C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2) \wedge$
$stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge$
$\text{pre}^{\text{Ret}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \text{pre}^{\text{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2)) \vee$
$(\exists\, u'_0, o_0, u'_1, o_1, u'_2, o_2 \bullet$
$C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge$
$stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge$
$G_2(u'_1, u'_2) \wedge O_{Op,2}(o_1, o_2; u'_1, u'_2, i_1, i_2, u_1, u_2) \wedge$
$stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge$
$\text{pre}^{\text{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \text{pre}^{\text{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2)) \vee$
$(\exists\, u'_0, o_0, u'_1, o_1, u'_2, o_2 \bullet$
$C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge$
$stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge$
$C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2) \wedge$
$stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge$
$\text{pre}^{\text{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \text{pre}^{\text{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2))))$

$\Rightarrow$

$(\exists\, u'_0, o_0, u'_2, o_2 \bullet$
$(\exists\, u_1, i_1 \bullet G_1(u_0, u_1) \wedge G_2(u_1, u_2) \wedge$
$P_{Op,1}(i_0, i_1, u_0, u_1) \wedge P_{Op,2}(i_1, i_2, u_1, u_2)) \wedge$
$stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge$
$((\exists\, u_1, i_1, u'_1, o_1 \bullet$
$G_1(u'_0, u'_1) \wedge O_{Op,1}(o_0, o_1; u'_0, u'_1, i_0, i_1, u_0, u_1) \wedge$
$C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2) \wedge$
$\text{pre}^{\text{Ret}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \text{pre}^{\text{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2)) \vee$
$(\exists\, u_1, i_1, u'_1, o_1 \bullet$
$C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge$
$G_2(u'_1, u'_2) \wedge O_{Op,2}(o_1, o_2; u'_1, u'_2, i_1, i_2, u_1, u_2) \wedge$
$\text{pre}^{\text{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \text{pre}^{\text{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2)) \vee$
$(\exists\, u_1, i_1, u'_1, o_1 \bullet$
$C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge$
$C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2) \wedge$
$\text{pre}^{\text{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \text{pre}^{\text{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2))))$

$\equiv$

$(\exists\, u'_0, o_0, u'_2, o_2 \bullet$
$G_{(1,2)}(u_0, u_2) \wedge P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \wedge$
$stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge$
$(\exists\, u_1, i_1, u'_1, o_1 \bullet$
$(G_1(u'_0, u'_1) \wedge O_{Op,1}(o_0, o_1; u'_0, u'_1, i_0, i_1, u_0, u_1) \wedge$
$C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2) \wedge$
$\text{pre}^{\text{Ret}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \text{pre}^{\text{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2)) \vee$
$(C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge$
$G_2(u'_1, u'_2) \wedge O_{Op,2}(o_1, o_2; u'_1, u'_2, i_1, i_2, u_1, u_2) \wedge$
$\text{pre}^{\text{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \text{pre}^{\text{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2)) \vee$
$(C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge$

$$
\begin{array}{l}
\qquad C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2) \wedge \\
\qquad \mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2))))
\end{array}
$$

$$
=
$$

$$
\begin{array}{l}
(\exists\, u'_0, o_0, u'_2, o_2 \bullet \\
\qquad G_{(1,2)}(u_0, u_2) \wedge P_{Op,(1,2)}(i_0, i_2, u_0, u_2) \wedge \\
\qquad stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge \\
\qquad C_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2))
\end{array}
$$

$$
=
$$

$$
\mathrm{pre}^{\mathrm{Con}}{}_{Op,(1,2)}(u_0, i_0, u_2, i_2) \tag{5.30}
$$

We are done. ☺

Corollary 5.11 For a composition of two specifically or generally closed retrenchments which both respect their regular data, we have (5.25) and (5.26).

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 3.16, to validate the selection of either $u'_{1,a}$, $o_{1,a}$ or $u'_{1,b}$, $o_{1,b}$ across both clauses at the relevant points in the proof. ☺

Theorem 5.12 The composition of specifically or generally closed retrenchments, given by (4.9)-(4.12), between deterministic systems, is associative.

Proof. We tackle the associativity for the concedes clauses, the most nontrivial part of the proof. For this we have:

$$
\begin{array}{l}
C_{Op,(1,(2,3))}(u'_0, u'_3, o_0, o_3; i_0, i_3, u_0, u_3) \\
\equiv \\
(\exists\, u'_1, o_1, u_1, i_1 \bullet \\
\qquad (G_1(u'_0, u'_1) \wedge O_{Op,1}(o_0, o_1; u'_0, u'_1, i_0, i_1, u_0, u_1) \wedge \\
\qquad\qquad C_{Op,(2,3)}(u'_1, u'_3, o_1, o_3; i_1, i_3, u_1, u_3) \wedge \\
\qquad\qquad \mathrm{pre}^{\mathrm{Ret}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,(2,3)}(u_1, i_1, u_3, i_3)) \vee \\
\qquad (C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \\
\qquad\qquad G_{(2,3)}(u'_1, u'_3) \wedge O_{Op,(2,3)}(o_1, o_3; u'_1, u'_3, i_1, i_3, u_1, u_3) \wedge \\
\qquad\qquad \mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,(2,3)}(u_1, i_1, u_3, i_3)) \vee \\
\qquad (C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \\
\qquad\qquad C_{Op,(2,3)}(u'_1, u'_3, o_1, o_3; i_1, i_3, u_1, u_3) \wedge \\
\qquad\qquad \mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,(2,3)}(u_1, i_1, u_3, i_3)))
\end{array}
$$

$$
\equiv
$$

$$
\begin{array}{l}
(\exists\, u'_1, o_1, u_1, i_1 \bullet \\
\qquad (G_1(u'_0, u'_1) \wedge O_{Op,1}(o_0, o_1; u'_0, u'_1, i_0, i_1, u_0, u_1) \wedge \\
\qquad\qquad (\exists\, u'_2, o_2, u_2, i_2 \bullet \\
\qquad\qquad\qquad (G_2(u'_1, u'_2) \wedge O_{Op,2}(o_1, o_2; u'_1, u'_2, i_1, i_2, u_1, u_2) \wedge \\
\qquad\qquad\qquad\qquad C_{Op,3}(u'_2, u'_3, o_2, o_3; i_2, i_3, u_2, u_3) \wedge \\
\qquad\qquad\qquad\qquad \mathrm{pre}^{\mathrm{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,3}(u_2, i_2, u_3, i_3)) \vee \\
\qquad\qquad\qquad (C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2) \wedge \\
\qquad\qquad\qquad\qquad G_3(u'_2, u'_3) \wedge O_{Op,3}(o_2, o_3; u'_2, u'_3, i_2, i_3, u_2, u_3) \wedge
\end{array}
$$

$$
\begin{array}{l}
\qquad\qquad \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,3}(u_2, i_2, u_3, i_3)) \vee \\
\qquad\quad (C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2) \wedge \\
\qquad\qquad C_{Op,3}(u'_2, u'_3, o_2, o_3; i_2, i_3, u_2, u_3) \wedge \\
\qquad\qquad \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,3}(u_2, i_2, u_3, i_3))) \wedge \\
\qquad \mathrm{pre}^{\mathrm{Ret}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \\
\qquad (\exists\, \underline{u}_2, \underline{i}_2 \bullet \\
\qquad\quad (\mathrm{pre}^{\mathrm{Ret}}{}_{Op,2}(u_1, i_1, \underline{u}_2, \underline{i}_2) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,3}(\underline{u}_2, \underline{i}_2, u_3, i_3)) \vee \\
\qquad\quad (\mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, \underline{u}_2, \underline{i}_2) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,3}(\underline{u}_2, \underline{i}_2, u_3, i_3)) \vee \\
\qquad\quad (\mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, \underline{u}_2, \underline{i}_2) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,3}(\underline{u}_2, \underline{i}_2, u_3, i_3)))) \vee \\
\quad (C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \\
\qquad (\exists\, u'_2 \bullet G_2(u'_1, u'_2) \wedge G_3(u'_2, u'_3)) \wedge \\
\qquad (\exists\, \underline{u}'_2, o_2, \underline{u}_2, \underline{i}_2 \bullet O_{Op,2}(o_1, o_2; u'_1, \underline{u}'_2, i_1, \underline{i}_2, u_1, \underline{u}_2) \wedge \\
\qquad\quad O_{Op,3}(o_2, o_3; \underline{u}'_2, u'_3, \underline{i}_2, i_3, \underline{u}_2, u_3) \wedge \\
\qquad\quad \mathrm{pre}^{\mathrm{Ret}}{}_{Op,2}(u_1, i_1, \underline{u}_2, \underline{i}_2) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,3}(\underline{u}_2, \underline{i}_2, u_3, i_3)) \wedge \\
\qquad \mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \\
\qquad (\exists\, u_2, i_2 \bullet \mathrm{pre}^{\mathrm{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,3}(u_2, i_2, u_3, i_3))) \vee \\
\quad (C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \\
\qquad (\exists\, u'_2, o_2, u_2, i_2 \bullet \\
\qquad\quad (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 \\
\qquad\qquad C_{Op,3}(u'_2, u'_3, o_2, o_3; i_2, i_3, u_2, u_3) \wedge \\
\qquad\qquad \mathrm{pre}^{\mathrm{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,3}(u_2, i_2, u_3, i_3)) \vee \\
\qquad\quad (C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2) \wedge \\
\qquad\qquad G_3(u'_2, u'_3) \wedge O_{Op,3}(o_2, o_3; u'_2, u'_3, i_2, i_3, u_2, u_3) \wedge \\
\qquad\qquad \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,3}(u_2, i_2, u_3, i_3)) \vee \\
\qquad\quad (C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2) \wedge \\
\qquad\qquad C_{Op,3}(u'_2, u'_3, o_2, o_3; i_2, i_3, u_2, u_3) \wedge \\
\qquad\qquad \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,3}(u_2, i_2, u_3, i_3))) \wedge \\
\qquad \mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \\
\qquad (\exists\, \underline{u}_2, \underline{i}_2 \bullet \\
\qquad\quad (\mathrm{pre}^{\mathrm{Ret}}{}_{Op,2}(u_1, i_1, \underline{u}_2, \underline{i}_2) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,3}(\underline{u}_2, \underline{i}_2, u_3, i_3)) \vee \\
\qquad\quad (\mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, \underline{u}_2, \underline{i}_2) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,3}(\underline{u}_2, \underline{i}_2, u_3, i_3)) \vee \\
\qquad\quad (\mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, \underline{u}_2, \underline{i}_2) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,3}(\underline{u}_2, \underline{i}_2, u_3, i_3)))))
\end{array}
$$

In this expression, under the top level conjunction, we have a disjunction of three major subexpressions. Consider the first of them, which contains the structure:

$$
\begin{array}{l}
(\exists\, u'_2, o_2, u_2, i_2 \bullet \\
\qquad (\ldots \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,3}(u_2, i_2, u_3, i_3)) \vee \\
\qquad (\ldots \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,3}(u_2, i_2, u_3, i_3)) \vee \\
\qquad (\ldots \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,3}(u_2, i_2, u_3, i_3))) \wedge \\
\ldots \wedge \\
(\exists\, \underline{u}_2, \underline{i}_2 \bullet \\
\qquad (\mathrm{pre}^{\mathrm{Ret}}{}_{Op,2}(u_1, i_1, \underline{u}_2, \underline{i}_2) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,3}(\underline{u}_2, \underline{i}_2, u_3, i_3)) \vee \\
\qquad (\mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, \underline{u}_2, \underline{i}_2) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,3}(\underline{u}_2, \underline{i}_2, u_3, i_3)) \vee \\
\qquad (\mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, \underline{u}_2, \underline{i}_2) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,3}(\underline{u}_2, \underline{i}_2, u_3, i_3)))
\end{array}
$$

When the $(\exists\ \underline{u}_2, \underline{i}_2 \bullet \ldots)$ conjunct is distributed into the $(\exists\ u'_2, o_2, u_2, i_2 \bullet \ldots)$ conjunct, because the u_2, i_2 and $\underline{u}_2$, $\underline{i}_2$ quantifications range over precisely the same values, for every $\underline{u}_2$, $\underline{i}_2$ pair that makes one of its (pre-… ∧ pre-…) terms true, the same pair of values will make the corresponding (… ∧ pre-… ∧ pre-…) term in the $(\exists\ u'_2, o_2, u_2, i_2 \bullet \ldots)$ conjunct true, leaving it unaffected. Therefore every conjunction of that particular (… ∧ pre-… ∧ pre-…) term with anything else demanded by the distribution will vanish by the absorption law. In this manner the whole $(\exists\ \underline{u}_2, \underline{i}_2 \bullet \ldots)$ conjunct can be erased.

A similar argument works for the third major subexpression, which contains the same structure. This leaves the second major subexpression:

$$
\begin{aligned}
&(C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \\
&\quad (\exists\ u'_2 \bullet G_2(u'_1, u'_2) \wedge G_3(u'_2, u'_3)) \wedge \\
&\quad (\exists\ \underline{u}'_2, o_2, \underline{u}_2, \underline{i}_2 \bullet O_{Op,2}(o_1, o_2; u'_1, \underline{u}'_2, i_1, \underline{i}_2, u_1, \underline{u}_2) \wedge \\
&\qquad O_{Op,3}(o_2, o_3; \underline{u}'_2, u'_3, \underline{i}_2, i_3, \underline{u}_2, u_3) \wedge \\
&\qquad \mathrm{pre}^{\mathrm{Ret}}{}_{Op,2}(u_1, i_1, \underline{u}_2, \underline{i}_2) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,3}(\underline{u}_2, \underline{i}_2, u_3, i_3)) \wedge \\
&\quad \mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \\
&\quad (\exists\ u_2, i_2 \bullet \mathrm{pre}^{\mathrm{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,3}(u_2, i_2, u_3, i_3)))
\end{aligned}
$$

Again the $(\exists\ u_2, i_2 \bullet \ldots)$ quantification can be taken into the $(\exists\ \underline{u}'_2, o_2, \underline{u}_2, \underline{i}_2 \bullet \ldots)$ quantification and eliminated. The same applies to the $(\exists\ u'_2 \bullet \ldots)$ quantification since the $(\exists\ \underline{u}'_2, o_2, \underline{u}_2, \underline{i}_2 \bullet \ldots)$ term cannot be true unless a u'_2 exists that makes $G_2 \wedge G_3$ true. Rearranging terms, we find for the whole thing:

$$
\begin{aligned}
&\equiv \\
&(\exists\ u'_1, o_1, u_1, i_1, u'_2, o_2, u_2, i_2 \bullet \\
&\quad (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 \mathrm{pre}^{\mathrm{Ret}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \\
&\quad 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{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2) \wedge \\
&\quad C_{Op,3}(u'_2, u'_3, o_2, o_3; i_2, i_3, u_2, u_3) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,3}(u_2, i_2, u_3, i_3)) \vee \\
&\quad (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 \mathrm{pre}^{\mathrm{Ret}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \\
&\quad 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,2}(u_1, i_1, u_2, i_2) \wedge \\
&\quad G_3(u'_2, u'_3) \wedge O_{Op,3}(o_2, o_3; u'_2, u'_3, i_2, i_3, u_2, u_3) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,3}(u_2, i_2, u_3, i_3)) \vee \\
&\quad (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 \mathrm{pre}^{\mathrm{Ret}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \\
&\quad 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,2}(u_1, i_1, u_2, i_2) \wedge \\
&\quad C_{Op,3}(u'_2, u'_3, o_2, o_3; i_2, i_3, u_2, u_3) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,3}(u_2, i_2, u_3, i_3)) \vee \\
&\quad (C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \\
&\quad 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{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2) \wedge \\
&\quad G_3(u'_2, u'_3) \wedge O_{Op,3}(o_2, o_3; u'_2, u'_3, i_2, i_3, u_2, u_3) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,3}(u_2, i_2, u_3, i_3)) \vee \\
&\quad (C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \\
&\quad 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{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2) \wedge \\
&\quad C_{Op,3}(u'_2, u'_3, o_2, o_3; i_2, i_3, u_2, u_3) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,3}(u_2, i_2, u_3, i_3)) \vee \\
&\quad (C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \\
&\quad 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,2}(u_1, i_1, u_2, i_2) \wedge \\
&\quad G_3(u'_2, u'_3) \wedge O_{Op,3}(o_2, o_3; u'_2, u'_3, i_2, i_3, u_2, u_3) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,3}(u_2, i_2, u_3, i_3)) \vee \\
&\quad (C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \\
&\quad 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,2}(u_1, i_1, u_2, i_2) \wedge \\
&\quad C_{Op,3}(u'_2, u'_3, o_2, o_3; i_2, i_3, u_2, u_3) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,3}(u_2, i_2, u_3, i_3))) \qquad (5.31)
\end{aligned}
$$

This structure is now suitably symmetrical in the indices 1, 2, 3, and so it is clear that the other order of association will generate the same result, completing the proof for the concedes relations. For the output relations, we need a calculation similar to that for the second major subexpression above, and the calculations for the retrieve and within relations have been covered in the proof of Theorem 5.8. We are done. ☺

Corollary 5.13 The composition of specifically or generally closed retrenchments, given by (4.9)-(4.12), which both respect their regular data, is associative.

Proof. 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 output retrenchments which respect their regular data that were needed to prove Corollary 5.11 were (a) and (b) as follows:

(a) Given two level 1 steps u_1 -(i_1, Op_1, $o_{1,a}$)-› $u'_{1,a}$ and u_1 -(i_1, Op_1, $o_{1,b}$)-› $u'_{1,b}$, whenever u_1 -(i_1, Op_1, $o_{1,a}$)-› $u'_{1,a}$ is related by $G_1(u'_0, u'_{1,a}) \wedge O_{Op,1}(o_0, o_{1,a}; u'_0, u'_{1,a}, i_0, i_1, u_0, u_1)$ or $C_{Op,1}(u'_0, u'_{1,a}, o_0, o_{1,a}; i_0, i_1, u_0, u_1)$ 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, $o_{1,b}$)-› $u'_{1,b}$.

(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 output retrenchment. Thus we can re-establish the analogue for three retrenchments of Corollary 5.11, and thence 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 (4.9)-(4.12), between deterministic systems, is associative.

Corollary 5.15 The composition of specifically or generally closed neat output retrenchments, given by (4.9)-(4.12), 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 (4.9)-(4.12).

Proof. Theorem 4.7 tells us we have an 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. Suppose that for the composed retrenchment we had:

$$\overline{G}_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2) \wedge \overline{C}_{Op,(1,2)}(u'_0, u'_2, o_0, o_2; i_0, i_2, u_0, u_2) \not\equiv \mathsf{false} \tag{5.32}$$

Instantiating the intermediate variables, we argue as in Theorem 5.9, except that we need even fewer distinct variables.

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

$\Rightarrow$ (instantiation, for some $u_1, i_1, u'_{1,a}, u_{1,a}, i_{1,a}, u'_{1,aa}, o_{1,a}, u_{1,b}, i_{1,b}, u'_{1,b}, o_{1,b}$)

$G_1(u_0, u_1) \wedge G_2(u_1, u_2) \wedge$
$P_{Op,1}(i_0, i_1, u_0, u_1) \wedge P_{Op,2}(i_1, i_2, u_1, u_2) \wedge$
$G_1(u'_0, u'_{1,a}) \wedge G_2(u'_{1,a}, u'_2) \wedge$
$O_{Op,1}(o_0, o_{1,a}; u'_0, u'_{1,aa}, i_0, i_{1,a}, u_0, u_{1,a}) \wedge$
$O_{Op,2}(o_{1,a}, o_2; u'_{1,aa}, u'_2, i_{1,a}, i_2, u_{1,a}, u_2) \wedge$
$\mathrm{pre}^{\mathrm{Ret}}{}_{Op,1}(u_0, i_0, u_{1,a}, i_{1,a}) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,2}(u_{1,a}, i_{1,a}, u_2, i_2) \wedge$
$stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge$
$((G_1(u'_0, u'_{1,b}) \wedge O_{Op,1}(o_0, o_{1,b}; u'_0, u'_{1,b}, i_0, i_{1,b}, u_0, u_{1,b}) \wedge$
$\quad C_{Op,2}(u'_{1,b}, u'_2, o_{1,b}, o_2; i_{1,b}, i_2, u_{1,b}, u_2) \wedge$
$\quad \mathrm{pre}^{\mathrm{Ret}}{}_{Op,1}(u_0, i_0, u_{1,b}, i_{1,b}) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_{1,b}, i_{1,b}, u_2, i_2)) \vee$
$(C_{Op,1}(u'_0, u'_{1,b}, o_0, o_{1,b}; i_0, i_{1,b}, u_0, u_{1,b}) \wedge$
$\quad G_2(u'_{1,b}, u'_2) \wedge O_{Op,2}(o_{1,b}, o_2; u'_{1,b}, u'_2, i_{1,b}, i_2, u_{1,b}, u_2) \wedge$
$\quad \mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_{1,b}, i_{1,b}) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,2}(u_{1,b}, i_{1,b}, u_2, i_2)) \vee$
$(C_{Op,1}(u'_0, u'_{1,b}, o_0, o_{1,b}; i_0, i_{1,b}, u_0, u_{1,b}) \wedge$
$\quad C_{Op,2}(u'_{1,b}, u'_2, o_{1,b}, o_2; i_{1,b}, i_2, u_{1,b}, u_2) \wedge$
$\quad \mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_{1,b}, i_{1,b}) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_{1,b}, i_{1,b}, u_2, i_2))) \wedge$
$stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2)$

$\Rightarrow$ (closedness, inferring $u_{1,a} = u_{1,b} = u_1$, $i_{1,a} = i_{1,b} = i_1$, $u'_{1,aa} = u'_{1,a}$,
either directly for specific closedness, or after some modus ponens
for general closedness; determinism, inferring
$u'_{1,a} = u'_{1,b} = u'_1$, $o_{1,a} = o_{1,b} = o_1$)

$G_1(u_0, u_1) \wedge G_2(u_1, u_2) \wedge$
$P_{Op,1}(i_0, i_1, u_0, u_1) \wedge P_{Op,2}(i_1, i_2, u_1, u_2) \wedge$
$G_1(u'_0, u'_1) \wedge G_2(u'_1, u'_2) \wedge$
$O_{Op,1}(o_0, o_1; u'_0, u'_1, i_0, i_1, u_0, u_1) \wedge$
$O_{Op,2}(o_1, o_2; u'_1, u'_2, i_1, i_2, u_1, u_2) \wedge$
$\mathrm{pre}^{\mathrm{Ret}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2) \wedge$
$stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge$
$stp_{Op,1}(u_1, i_1, u'_1, o_1) \wedge$
$((G_1(u'_0, u'_1) \wedge O_{Op,1}(o_0, o_1; u'_0, u'_1, i_0, i_1, u_0, u_1) \wedge$
$\quad C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2) \wedge$
$\quad \mathrm{pre}^{\mathrm{Ret}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2)) \vee$
$(C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge$
$\quad G_2(u'_1, u'_2) \wedge O_{Op,2}(o_1, o_2; u'_1, u'_2, i_1, i_2, u_1, u_2) \wedge$
$\quad \mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Ret}}{}_{Op,2}(u_1, i_1, u_2, i_2)) \vee$

$$
\begin{array}{l}
(C_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \\
\quad C_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2) \wedge \\
\quad \mathrm{pre}^{\mathrm{Con}}{}_{Op,1}(u_0, i_0, u_1, i_1) \wedge \mathrm{pre}^{\mathrm{Con}}{}_{Op,2}(u_1, i_1, u_2, i_2))) \wedge \\
stp_{Op,0}(u_0, i_0, u'_0, o_0) \wedge stp_{Op,2}(u_2, i_2, u'_2, o_2) \wedge \\
stp_{Op,1}(u_1, i_1, u'_1, o_1) \\
\\
\Rightarrow \\
\\
\overline{G}_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \overline{G}_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2) \wedge \\
((\overline{G}_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \overline{C}_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2)) \vee \\
(\overline{C}_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \overline{G}_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2)) \vee \\
(\overline{C}_{Op,1}(u'_0, u'_1, o_0, o_1; i_0, i_1, u_0, u_1) \wedge \overline{C}_{Op,2}(u'_1, u'_2, o_1, o_2; i_1, i_2, u_1, u_2))) \quad (5.33)
\end{array}
$$

It is now easy to see that distributing the $\overline{G}_{Op,1} \wedge \overline{G}_{Op,2}$ into the disjunction yields contradictions of fastidiousness of at least one of the component retrenchments in each resulting disjunct. ☺

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 (4.9)-(4.12).

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 (4.9)-(4.12), between deterministic systems, is associative.

Corollary 5.19 The composition of specifically or generally closed fastidious retrenchments, given by (4.9)-(4.12), which both respect their regular data, is associative.

6 Case Study: Simple Abstract Feature Engineering

In this section we apply some of the preceding material to give an abstract account of software feature engineering. In [Zave (2001)] a feature of a software system is described as 'an optional or incremental unit of functionality', and the technique of developing or evolving software by taking into account successive features, is called feature engineering. If it were the case that features fell into a convenient hierarchy in which successive features built smoothly upon the facilities offered by their predecessors, we could use refinement, especially in its superposition refinement incarnation [Back (2002), Back and Sere (1996), Katz (1993), Francez and Forman (1990)], to compose the system out of its pieces. However it is manifestly not the case that features necessarily conform to such a convenient discipline. Especially in telecommunications engineering [Zave (2001)], providers invent new features that telephone systems might offer, without constraining their imaginations regarding how the new features might interact with existing functionality: that is left as a challenging and important problem for system integrators. In the face of such functional anarchy, which inevitably has to face situations where the new system being developed must contradict some properties possessed by its predecessor, refinement is rather hamstrung in

what it can offer as a development methodology, since contradicting what has already been established is anathema for any refinement technique. In this regard, the more indulgent ways of retrenchment have more scope for giving an account of the process that bears some relation to what is actually done by the engineers, as we now demonstrate.

In the following paragraphs, we define a feature and give a simple language for building operations out of features in which each feature expression has a normal form. We say how feature oriented definitions of an operation may evolve, and discuss the relationships between state, input and output spaces at successive stages of such an evolution. These turn out to be regular, and, suitably packaged, they form the retrieve, within, and output relations of a retrenchment decription of the evolution process. The concedes relations for these retrenchments are built according to the default retrenchment strategy. Furthermore, under mild assumptions, these retrenchments are shown to be neat, but in general they fail to fully enjoy the associative composition properties we worked hard to establish in the previous section.

6.1 Operations, Features, Feature Expressions, Normal Forms

From our perspective, a feature oriented development process will be concerned with developing one or more operations of some system, through a number of incarnations displaying evolving functionality. As usual we will focus on some generic operation Op whose incarnations through the development are labelled Op_k where $k = 0, 1, \ldots$, as in previous sections. We assume that at each incarnation, the operation Op_k is *model complete*, by which we mean that Op_k supplies a response to any demand that it is reasonable to make on Op_k, appropriate to the level of abstraction and to the state of the overall development at level k. Model completeness is related to the issue of the totality of the transition relation of Op_k, $stp_{Op,k}$, on its space of before-states and inputs, but is different from it. In particular, it is possible that a model complete Op_k is a partial relation in the mathematical sense, i.e. $\mathrm{dom}(stp_{Op,k})$ is not all of $\mathsf{U}_k \times \mathsf{I}_{Op,k}$, so that there are before-states and inputs pairs (u, i) in $\mathsf{U}_k \times \mathsf{I}_{Op,k}$ which are not the source of some transition of Op_k. But in such cases it must be possible to firmly justify this state of affairs on engineering grounds; eg. it might be physically impossible to invoke Op_k for those particular (u, i).

Model completeness is an important consideration from requirements and fitness for purpose perspectives, but is contingent on the understanding of how a particular formal model relates to the real world. Thus it is ultimately a meta level issue, and outside the scope of the formal model itself. Nevertheless it is something that impacts formal development methodology indirectly, because the need to give an appropriate response under all circumstances that are reasonable according to the meta level relationship of the formal model to the real world, can lead to the adoption of responses that need to be contradicted at later levels of the development, with the consequences noted above. (Whereas a cunning refinement based strategy could forego model completeness and simply omit at higher levels responses that could not be subsequently refined in the way desired.)

So much for operations. Now for features. The key difference between a feature and an operation, is that every feature is 'an optional or incremental unit of functionality'. As a consequence, features are *not* required to be model complete in the sense just discussed, since the functionality that they address need not concern the entire set of

demands that might be made on Op_k. So each Op_k will be composed of a collection of features which together must guarantee model completeness but that separately need not have the property.

In the spirit of aiming to support software engineering intuitions, we said that we will assume that our models are model complete at every level k. This corresponds nicely to the statement that the system has no capabilities that lie outside the remit of the default within relations $P^{\mathrm{Def}}{}_{Op,k+1}(i_k, i_{k+1}, u_k, u_{k+1})$ defined in (2.7). For to have such capabilities implies that when $G_{k+1}(u_k, u_{k+1}) \wedge P_{Op,k+1}(i_k, i_{k+1}, u_k, u_{k+1})$ holds, we have either the absence of an abstract transition when there is a concrete one, or vice versa. But model completeness implies just the negation of that: if some (u_k, i_k) is a point from which it is necessary for a level k transition to emerge, then its level $k+1$ counterpart (u_{k+1}, i_{k+1}), obtained via $G_{k+1} \wedge P_{Op,k+1}$, must be a point from which it is appropriate for a level $k+1$ transition to emerge, and vice versa. We avail ourselves of the properties of model completeness below.

Individual features are denoted $f_{\mathrm{d},k}$ where d refers to the feature itself and k is the level of the development, and are (normally partial) relations with the same signature as the operations to which they contribute: $\mathsf{U}_k \times \mathsf{I}_{Op,k} \leftrightarrow \mathsf{U}_k \times \mathsf{O}_{Op,k}$. We now (and at other times below) suppress the subscript k till we return to discussing relationships between different levels of abstraction/development.

Since all features that contribute to a given Op have the same signature, features may be combined using any operator on relations that yields a result with the given relational signature from suitable parameters. We fix on the following, rather oversimplified menu of combinators; one which is just rich enough to show the potential utility of retrenchment based techniques in this application area.

Definition 6.1 Feature expression combinators (illustrated working on individual features, but applicable to feature expressions in general).

(1) Union: written $(\_ \cup \_)$; eg. $(f_{\mathrm{d}} \cup f_{\mathrm{e}})$

(2) Union asserted disjoint: written $(\_ \mathbin{\dot\cup} \_)$; eg. $(f_{\mathrm{d}} \mathbin{\dot\cup} f_{\mathrm{e}})$

(3) Override: written $(\_ \lhd \_)$; eg. $(f_{\mathrm{e}} \lhd f_{\mathrm{d}})$

(4) Conditional: written (If _ then _ else _ fi) ; eg. (If $p(u, i)$ then f_{d} else f_{e} fi)
also written $(\_ \lhd_p \_)$; eg. $(f_{\mathrm{e}} \lhd_p f_{\mathrm{d}})$

(5) Case: written (Case _ of _: _ ; … ; esac) ;
eg. (Case $p(u, i)$ of v_0: f_{d_0} ; v_1: f_{d_1} ; … v_n: f_{d_n} ; else f_{e} ; esac)
also written $(\_ \bullet \_: \_ ; \ldots ;; \_)$; eg. $(p \bullet v_0{:}\, f_{\mathrm{d}_0} ; v_1{:}\, f_{\mathrm{d}_1} ; \ldots v_n{:}\, f_{\mathrm{d}_n} ;; f_{\mathrm{e}})$ (6.1)

It is clear from this that there are two kinds of subexpression of feature expressions: a subexpression can be a feature subexpression (a FSE, a subexpression of feature type), or a condition subexpression (a CSE, a subexpression of boolean type). We call a subexpression a pure FSE, iff it is not a subexpression of a CSE.

Definition 6.2 A feature expression ϕ is featurewise linear iff any individual feature f_{d} occurs at most once as a pure FSE of ϕ.

Semantically, the feature expression combinators are to be understood as follows.

Union has the meaning of conventional set theoretic union.

Union asserted disjoint is slightly unusual in that its semantics is that of conventional set theoretic union, but with the side condition that its operand relations have disjoint domains. The elements of a union asserted disjoint can each be mapped back to the set from which they came, without confusion, because of the side condition. This is in contrast to disjoint union proper, which guarantees such a 'birth certificate' property unconditionally, by employing some behind the scenes machinery typically involving the tagging of each element of the disjoint union with some label indicating where it came from. We reject the general construction for our purposes because it always introduces some additional set theoretic machinery which is not specified canonically. In our environment, set theoretic details are suposed to correspond to elements of the application that we are trying to model, and arbitrary unspecified set theoretic mechanisms can have no place. Notations employing union asserted disjoint in which the side condition is not true are undefined.

Conditional composition behaves as expected. When $p(u, i)$ evaluates to true, then the conditional (If $p(u, i)$ then f_d else f_e fi) is true iff $stp_{f_d}(u, i, u', o)$ is true. When $p(u, i)$ evaluates to false, then the conditional (If $p(u, i)$ then f_d else f_e fi) is true iff $stp_{f_e}(u, i, u', o)$ is true. And $p(u, i)$ must always evaluate to one of these, or else the whole expression is not defined. N.B. Note the differing order of the operands in the two notations.

Override is a special case of Conditional in which $p(u, i)$ is $((u, i) \in \mathrm{dom}(stp_{f_d}))$.

Case is defined in the usual way. We demand that $p(u, i)$ is a (partial) function, yielding at most one value, or else the whole expression is not defined. When $p(u, i)$ evaluates to one of the values in $\{v_0 \ldots v_n\}$, v_j say, then the case construct is true iff for the relevant j, $stp_{f_{d_j}}(u, i, u', o)$ is true. Otherwise, i.e. if $p(u, i)$ yields a value not in $\{v_0 \ldots v_n\}$, or if $p(u, i)$ is undefined, then the case construct is true iff $stp_{f_e}(u, i, u', o)$ is true.

These operators are sufficient to model others we might also want to consider. For example domain restriction and domain subtraction can be modelled by:

$$(S \lhd f_d) \equiv (\text{If } (u, i) \in (\mathrm{dom}(stp_{f_d}) \cap S) \text{ then } f_d \text{ else } \varnothing \text{ fi}) \tag{6.2}$$

$$(S \lhd\!\!\!- f_d) \equiv (\text{If } (u, i) \in (\mathrm{dom}(stp_{f_d}) - S) \text{ then } f_d \text{ else } \varnothing \text{ fi}) \tag{6.3}$$

With this collection of feature combination operators at our disposal we can regard the *stp* relation of an operator *Op* as the value of an expression built out of features using these combinators, a feature expression (FE). When an operator *Op* is defined by a FE, we call this a feature oriented definition (FOD) of *Op*.

In line with the emphasis on semantic issues in this paper, we regard FEs as identical if they differ at worst by the renaming of constituent individual features or by the permutation of the parameters of commutative combinators. Thus $f_d \cup f_e$ and $f_e \cup f_d$ are regarded as identical FEs. On the other hand $f_d \cup f_d$ and f_d are different FEs which happen to have the same value (i.e. they are equivalent FEs though not identical). We are thus regarding FEs as a fairly abstract syntax for the relations that they denote.

In future we will write $\mathrm{dom}(f_d)$ and $\mathrm{dom}(Op)$ instead of $\mathrm{dom}(stp_{f_d})$ and $\mathrm{dom}(stp_{Op})$. If ϕ is a FE, we write $\mathrm{dom}(\phi)$ for the domain of the relation that ϕ defines.

Theorem 6.3 Every FE ϕ has a normal form, $\mathrm{NF}(\phi)$:

$$\phi \;=\; \phi_1 \uplus \phi_2 \uplus \ldots \uplus \phi_n \tag{6.4}$$

such that:

(1) Each ϕ_j is nonempty, unless ϕ itself denotes the empty relation (in which case $\mathrm{NF}(\phi) \equiv \varnothing$).

(2) $\mathsf{U} \times \mathsf{I}_{Op}$ is partitioned into:

$$\mathrm{dom}(\phi_1) \uplus \mathrm{dom}(\phi_2) \uplus \ldots \uplus \mathrm{dom}(\phi_n) \uplus ((\mathsf{U} \times \mathsf{I}_{Op}) - \mathrm{dom}(\phi)) \tag{6.5}$$

where the last term is omitted if ϕ is total, and all terms except the last are omitted if ϕ is the empty relation.

(3) For each j, $\phi_j = \mathrm{dom}(\phi_j) \lhd (f_{\mathrm{a}j} \cup f_{\mathrm{b}j} \cup \ldots \cup f_{\mathrm{z}j})$ where: the $f_{\mathrm{b}j}$ are distinct individual features occuring in the FE ϕ; for each $f_{\mathrm{b}j}$, $\mathrm{dom}(\phi_j) \subseteq \mathrm{dom}(f_{\mathrm{b}j})$; and if $j_1 \neq j_2$, then $(f_{\mathrm{a}j_1} \cup f_{\mathrm{b}j_1} \cup \ldots \cup f_{\mathrm{z}j_1})$ and $(f_{\mathrm{a}j_2} \cup f_{\mathrm{b}j_2} \cup \ldots \cup f_{\mathrm{z}j_2})$ differ by at least one individual feature.

Proof. We go by induction on the structure of ϕ. If ϕ is $\varnothing$, or is an individual feature f_{b}, (7.4) is an identity and the remaining conclusions are trivial.

Suppose ϕ is $(\phi_{\mathrm{a}} \cup \phi_{\mathrm{b}})$, and suppose $\mathrm{NF}(\phi_{\mathrm{a}})$ and $\mathrm{NF}(\phi_{\mathrm{b}})$ are known. Then since:

$$\mathrm{dom}(\phi_{\mathrm{a}} \cup \phi_{\mathrm{b}}) \;=\; \mathrm{dom}(\phi_{\mathrm{a}} \cap \phi_{\mathrm{b}}) \uplus \mathrm{dom}(\phi_{\mathrm{a}} - \phi_{\mathrm{b}}) \uplus \mathrm{dom}(\phi_{\mathrm{b}} - \phi_{\mathrm{a}}) \tag{6.6}$$

we can use this decomposition of $\mathrm{dom}(\phi_{\mathrm{a}} \cup \phi_{\mathrm{b}})$ to generate a common refinement of the partitions of $\mathsf{U} \times \mathsf{I}_{Op}$ from the NFs of ϕ_{a} and ϕ_{b}, and to define the ϕ_js belonging to this partition. In a preprocessing phase, elements ϕ_j of the NF for ϕ_{a} whose domains intersect both $\mathrm{dom}(\phi_{\mathrm{a}} - \phi_{\mathrm{b}})$ and $\mathrm{dom}(\phi_{\mathrm{a}} \cap \phi_{\mathrm{b}})$ are first split into two across the boundary of $\mathrm{dom}(\phi_{\mathrm{b}})$ yielding $(\mathrm{dom}(\phi_j) - \mathrm{dom}(\phi_{\mathrm{b}})) \lhd (f_{\mathrm{a}j} \cup f_{\mathrm{b}j} \cup \ldots \cup f_{\mathrm{z}j})$ and $(\mathrm{dom}(\phi_j) \cap \mathrm{dom}(\phi_{\mathrm{b}})) \lhd (f_{\mathrm{a}j} \cup f_{\mathrm{b}j} \cup \ldots \cup f_{\mathrm{z}j})$. Likewise for elements ϕ_j of the NF for ϕ_{b} whose domains intersect both $\mathrm{dom}(\phi_{\mathrm{b}} - \phi_{\mathrm{a}})$ and $\mathrm{dom}(\phi_{\mathrm{a}} \cap \phi_{\mathrm{b}})$. It is clear that this first phase preserves all the desired properties of the NFs for ϕ_{a} and ϕ_{b}, except for the 'differ by at least one individual feature' property in (3). Each element of these NFs for ϕ_{a} and ϕ_{b} now has its domain wholly inside one of: (i) $\mathrm{dom}(\phi_{\mathrm{a}} \cap \phi_{\mathrm{b}})$, (ii) $\mathrm{dom}(\phi_{\mathrm{a}} - \phi_{\mathrm{b}})$, (iii) $\mathrm{dom}(\phi_{\mathrm{b}} - \phi_{\mathrm{a}})$. The NF for $\phi_{\mathrm{a}} \cup \phi_{\mathrm{b}}$ is now constructed as follows.

Any elements $((\mathsf{U} \times \mathsf{I}_{Op}) - \mathrm{dom}(\phi_{\mathrm{a}}))$ and $((\mathsf{U} \times \mathsf{I}_{Op}) - \mathrm{dom}(\phi_{\mathrm{b}}))$ from the original partitions of $\mathsf{U} \times \mathsf{I}_{Op}$ are both refined to an element of the new partition $((\mathsf{U} \times \mathsf{I}_{Op}) - \mathrm{dom}(\phi_{\mathrm{a}} \cup \phi_{\mathrm{b}}))$, provided it is nonempty.

If $\mathrm{dom}(\phi_{\mathrm{a}j}) \cap \mathrm{dom}(\phi_{\mathrm{b}j}) \neq \varnothing$, where $\phi_{\mathrm{a}j}$ and $\phi_{\mathrm{b}j}$ are elements of the NFs of ϕ_{a} and ϕ_{b}, then if $\phi_{\mathrm{a}j} = \mathrm{dom}(\phi_{\mathrm{a}j}) \lhd (f_{\mathrm{a}j} \cup f_{\mathrm{aa}j} \cup \ldots \cup f_{\mathrm{aaa}j})$ and $\phi_{\mathrm{b}j} = \mathrm{dom}(\phi_{\mathrm{b}j}) \lhd (f_{\mathrm{b}j} \cup f_{\mathrm{bb}j} \cup \ldots \cup f_{\mathrm{bbb}j})$, then an element of the NF for $\phi_{\mathrm{a}} \cup \phi_{\mathrm{b}}$ is defined by $(\mathrm{dom}(\phi_{\mathrm{a}j}) \cap \mathrm{dom}(\phi_{\mathrm{b}j})) \lhd (f_{\mathrm{a}j} \cup f_{\mathrm{aa}j} \cup \ldots \cup f_{\mathrm{aaa}j} \cup f_{\mathrm{b}j} \cup f_{\mathrm{bb}j} \cup \ldots \cup f_{\mathrm{bbb}j})$, where duplicate occurrences of features are removed from the union. The set $(\mathrm{dom}(\phi_{\mathrm{a}j}) \cap \mathrm{dom}(\phi_{\mathrm{b}j}))$ forms an element of the new partition of $\mathsf{U} \times \mathsf{I}_{Op}$.

Elements of the partition for ϕ_{a} whose domains lie wholly inside $\mathrm{dom}(\phi_{\mathrm{a}} - \phi_{\mathrm{b}})$ form elements of the new partition, and their ϕ_js also belong to the new normal form. Likewise for the 'entirely in $\mathrm{dom}(\phi_{\mathrm{b}} - \phi_{\mathrm{a}})$' case.

In a postprocessing phase, any elements of the partition of $\mathsf{U} \times \mathsf{I}_{Op}$ that now share the same collection of individual features, are amalgamated. This completes the construction. It is easy to see that the construction possesses the claimed properties.

Suppose ϕ is $(\phi_a \Cup \phi_b)$. Then we have a simpler version of the preceding.

Suppose ϕ is $(\phi_b \lhd_p \phi_a)$. Let $true(p)$ denote the subset of $\mathsf{U} \times \mathsf{I}_{Op}$ where p is true; let $false(p)$ denote its complement. In a preprocessing phase, elements of the NFs of both ϕ_a and ϕ_b are first split into two across the boundary between $true(p)$ and $false(p)$ in the obvious way. New partition element $((\mathsf{U} \times \mathsf{I}_{Op}) - \mathrm{dom}(\phi_b \lhd_p \phi_a))$ is generated, provided it is nonempty. The new NF then consists of elements of the NF of ϕ_a (and the corresponding partition elements) whose domains lie entirely in $true(p)$ together with elements of the NF of ϕ_b (and the corresponding partition elements) whose domains lie entirely in $false(p)$, discarding of course any empty ones, and amalgamating any partition elements that share the same individual feature collections.

Suppose ϕ is $(\phi_b \lhd \phi_a)$. This is a special case of the preceding.

Suppose ϕ is $(p \bullet v_0{:}\ \phi_{a_0}\ ;\ v_1{:}\ \phi_{a_1}\ ;\ \ldots\ v_n{:}\ \phi_{a_n}\ ;;\ \phi_b)$. This is similar to the preceding.

We are done. ☺

The normal form theorem gives us a vivid picture of what systems built out of features using the combinators of (7.1) look like. Fundamentally, the space of valid before-states and inputs partitions into a collection of sets, on each of which a well defined subset of the features defines the behaviour by nondeterministic choice amongst them; essentially this reduces any FE to a case analysis.

Theorem 6.4 Every FE ϕ is equivalent to a featurewise linear FE ϕ'.

Proof. Suppose $\mathrm{NF}(\phi) = \phi_1 \Cup \phi_2 \Cup \ldots \Cup \phi_n$, in which the individual features that occur are $\{f_a, f_b, \ldots, f_z\}$. For all $i \in \{a \ldots z\}$, let $\mathrm{dom_{fl}}(f_i) \equiv \bigcup \{\mathrm{dom}(\phi_j) \mid j \in \{1 \ldots n\}, f_i \text{ is an element of the union } \phi_j\}$. Then it is easy to see that ϕ is equivalent to:

$$\phi' \equiv (\mathrm{dom_{fl}}(f_a) \lhd f_a) \cup (\mathrm{dom_{fl}}(f_b) \lhd f_b) \cup \ldots \cup (\mathrm{dom_{fl}}(f_z) \lhd f_z) \tag{6.7}$$

which is featurewise linear. ☺

We can get another handle on where individual feature occurrences (and more general subexpressions of a FOD) determine the behaviour defined, by solving a constraint problem in the manner of attributed grammars. Thus, while the domains of individual features attempt to define the behaviour of the operation as a whole, the conditions in the Case, Conditional and Override combinators impose restrictions on what part of the domain of the operation has its behaviour defined by the controlled subexpressions. We calculate the tradeoff using an inherited attribute $in(\phi)$ and a synthesised attribute $sy(\phi)$, for each subexpression ϕ.

Definition 6.5 The active domain $\mathrm{dom_{act}}(\phi)$ for a FSE ϕ in a FE Φ is given by finding the solution to the set of constraints generated as follows over the structure of Φ:

(1) The root of ϕ has $in(\phi) = \mathsf{U}_k \times \mathsf{I}_{Op,k}$.

(2) An individual feature f_c has $sy(f_c) = \mathrm{dom}(f_c) \cap in(f_c)$.

(3) If ϕ is $(\phi_d \cup \phi_e)$ then $in(\phi_d) = in(\phi)$, $in(\phi_e) = in(\phi)$, $sy(\phi) = sy(\phi_d) \cup sy(\phi_e)$.

(4) If ϕ is $(\phi_d \Cup \phi_e)$ then $in(\phi_d) = in(\phi)$, $in(\phi_e) = in(\phi)$, $sy(\phi) = sy(\phi_d) \cup sy(\phi_e)$.

(5) If ϕ is $(\phi_e \lessdot \phi_d)$ then $in(\phi_d) = in(\phi)$, $in(\phi_e) = in(\phi) - sy(\phi_d)$,
$sy(\phi) = sy(\phi_d) \cup sy(\phi_e)$.

(6) If ϕ is $(\phi_e \lessdot_p \phi_d)$ then $in(\phi_d) = in(\phi) \cap true(p)$, $in(\phi_e) = in(\phi) \cap false(p)$,
$sy(\phi) = (sy(\phi_d) \cap true(p)) \cup (sy(\phi_e) \cap false(p))$.

(7) If ϕ is $(p \bullet v_0{:}\ \phi_{d_0}\ ;\ v_1{:}\ \phi_{d_1}\ ;\ \ldots\ v_n{:}\ \phi_{d_n}\ ;;\ \phi_e)$ then $in(\phi_{d_j}) = in(\phi) \cap p^{-1}(v_j)$,
$in(\phi_e) = in(\phi) \cap (\bigcup_k \mathsf{I}_{Op,k} - \bigcup_j (p^{-1}(v_j)))$,
$sy(\phi) = \bigcup_j (sy(\phi_{d_j}) \cap p^{-1}(v_j)) \cup (sy(\phi_e) \cap (\bigcup_k \mathsf{I}_{Op,k} - \bigcup_j (p^{-1}(v_j))))$.

(8) $\mathrm{dom}_{\mathrm{act}}(\phi) = sy(\phi)$.

Because we have no feature variables (which might lead to recursive equations) it is clear that for any FOD Φ, the above system can be solved for the $sy(\phi)$ provided we know all the $\mathrm{dom}(f_c)$ and the various conditions p. The solution is obtained by a depth first traversal of the parse tree of the FOD Φ, pushing $in(\_)$ sets down and picking up $sy(\_)$ sets on the way back up.

Specifically, the base cases of the traversal for (1) and (2) are obvious. For cases (3) and (4), we propagate the *in* sets down first, and wait for the *sy* sets to come back up. For case (5), we propagate the *in* set down the dominant branch and wait for the returning *sy* set, after which we propagate the relevant *in* set down the subordinate branch and wait for the *sy* set, after which we send the *sy* set for the whole subexpression up. For case (6), we propagate the relevant part of the *in* set down the *true* branch, similarly for the *false* branch; and when the child *sy* sets are available, we synthesise the *sy* set for the whole case. Case (7) is similar. When the process has calculated the *sy* set for ϕ, we return the desired result via (8).

If we let ϕ in the above be a feature occurrence f_d, we can calculate its active domain by the above procedure. If we aggregate all the occurrences of the same feature f_d in Φ, the union of the collection of $\mathrm{dom}_{\mathrm{act}}$ sets that we generate (one per occurrence of f_d), equals $\mathrm{dom}_{\mathrm{fl}}(f_d)$, a fact whose proof we leave to the reader. In future, when we write $\mathrm{dom}_{\mathrm{act}}(f_d)$ without making it explicit that an *occurrence* of f_d is being referred to, we will intend it to mean this aggregation of $\mathrm{dom}_{\mathrm{act}}$ sets of individual occurrences, equal to $\mathrm{dom}_{\mathrm{fl}}(f_d)$.

Definition 6.6 Let Φ be a FE and let ϕ be a FSE of Φ. The apparent active domain of ϕ is defined as the maximal subset of $\mathrm{dom}(\phi)$, $\mathrm{dom}_{\mathrm{appact}}(\phi)$, such that $(\mathrm{dom}_{\mathrm{appact}}(\phi) \lhd \phi)$ is a subrelation of (the relation defined by) Φ.

Note that $\mathrm{dom}_{\mathrm{act}}(\phi) \subseteq \mathrm{dom}_{\mathrm{appact}}(\phi)$, and the inclusion may be proper since ϕ may be overridden on part of its domain by some other part of the FE Φ that locally defines the same relation as ϕ. The FE $(\phi_d \lessdot \phi_d)$ is the obvious example.

Feature engineering consists of manipulating the features that enter into an operation in order to achieve the effects desired, where we can describe this activity either via the FOD or via the NF. Depending on the details, it might be more convenient to align an implementation with one or the other of these descriptions. We turn our attention to how such descriptions evolve.

6.2 Evolution of FODs of Operations

We will consider two kinds of development steps for FODs of operations. Neither is required to go according to the syntactic structure of the FOD, simply because there

is no reason to assume that the development activity —driven as it is by many external considerations— will meekly conform to syntactic criteria. Indeed we have remarked already that model completeness at all stages of development is a crucial consideration, and there need not be any correlation between that and syntax.

The first kind of development step simply alters the condition p in a Conditional or Case construct somewhere in the FOD; the second involves the infiltration of one or more new features into the current FOD of the operation. We say 'infiltration' to stress we are not necessarily working by recursion on the structure of the final FOD.

Despite the unruly nature of such steps as regards the syntactic structure of the FOD, the normal form theorem assures us that all such development steps can be reduced to consideration of partitions of the before-state and input space and the specification of appropriate behaviour on any new pieces generated.

We examine this in more detail below. However before we do so we must recognise that it is seldom the case that all the features we need to deal with intrinsically have the same signature $\mathsf{U}_k \times \mathsf{I}_{Op,k} \leftrightarrow \mathsf{U}_k \times \mathsf{O}_{Op,k}$. Often new features introduced during the development of a system need additional data structures to support the novel functionality, so as k increases, the various U_k, $\mathsf{I}_{Op,k}$, $\mathsf{O}_{Op,k}$ spaces cannot be assumed to stay the same. Luckily the relative independence of individual features makes this situation tractable.

Suppose that feature f_d, which is introduced in the step from level k to level $k+1$, requires additional supporting data u_d in the state, drawn from a space of values for u_d, U_d say. Then U_d will be present in U_{k+1} and not be present in U_k. It will be present in U_{k+1} as a cartesian factor, since U_d will be independent of any other data types used in the system.

Given this, we must now consider how a feature f_c which is present at level k and persists into level $k+1$, and so is well defined on states in U_k, is to be understood on a larger state space including U_d. The answer is easy. In common with programming practice in which an update of a variable leaves all other variables unaffected, we understand the relation representing f_c in the larger state space including U_d, to be the relation on U_k extended by the identity on irrelevant factors such as U_d. Specifically, if u -$(i, f_{\mathrm{c},k}, o)$-› u' is a typical transition for individual feature f_c at level k, then at level $k+1$, the transition u -$(i, f_{\mathrm{c},k}, o)$-› u' will be represented by a collection of transitions (u, u_d) -$(i, f_{\mathrm{c},k+1}, o)$-› (u', u_d), one for every $u_\mathrm{d} \in \mathsf{U}_\mathrm{d}$, where we have assumed that the adjunction of the space U_d is the only alteration in the state spaces needed in the passage from level k to level $k+1$. The same idea works for as many irrelevant factors as we need to introduce in the passage from level k to level $k+1$.

In general U_k will be a cartesian product of individual types $\mathsf{U}_{\mathrm{a},k} \times \mathsf{U}_{\mathrm{b},k} \times \ldots \times \mathsf{U}_{\mathrm{d},k}$ some of which are present because a specific feature demands them, others being common to the activity of several or all of the features at level k. Similarly for level $k+1$ where U_{k+1} will be $\mathsf{U}_{\mathrm{a}',k+1} \times \mathsf{U}_{\mathrm{b}',k+1} \times \ldots \times \mathsf{U}_{\mathrm{e}',k+1}$. Regarding the relationship between levels k and $k+1$, some of the $\mathsf{U}_{\mathrm{b},k}$ can be identified with some of the $\mathsf{U}_{\mathrm{b}',k+1}$. This will be because they are data types 'used in the same way' by features that are present at both levels; typically they will be the types of the same variables in a syntactic description of the common features. (Nonetheless we emphasise that in a purely semantic framework like ours, strictly speaking, this correspondence remains outside the formalism without recourse to some such syntactic description of features;

which is why 'used in the same way' appeared in quotes.) The remainder of the $\mathsf{U}_{c,k}$ … $\mathsf{U}_{d,k}$ will be present only at level k, and the remainder of the $\mathsf{U}_{c',k+1}$ … $\mathsf{U}_{e',k+1}$ will be present only at level $k+1$; this being because they concern features present exclusively at one level but not the other[3].

In this scenario the relationship between U_k and U_{k+1} will be a total surjective regular relation $\rho_{\mathsf{U},k,k+1}$. For let a typical value in U_k be $u_k = (u_{a,k}, \ldots, u_{c,k}, u_{d_1,k}, \ldots, u_{d_n,k})$ and a typical value in U_{k+1} be $u_{k+1} = (u_{d'_1,k+1}, \ldots, u_{d'_n,k+1}, u_{e',k+1}, \ldots, u_{g',k+1})$, where subspace $\mathsf{U}_{a,k} \times \ldots \times \mathsf{U}_{c,k}$ is removed from U_k in the passage from level k to level $k+1$, and subspace $\mathsf{U}_{e',k+1} \times \ldots \times \mathsf{U}_{g',k+1}$ is added; and where $\mathsf{U}_{d_1,k} = \mathsf{U}_{d'_1,k+1}, \ldots, \mathsf{U}_{d_n,k} = \mathsf{U}_{d'_n,k+1}$ are the common types, n of them, identified as indicated.

Then:

$$\rho_{\mathsf{U},k,k+1}(u_k, u_{k+1}) \equiv (u_{d_1,k} = u_{d'_1,k+1} \wedge \ldots \wedge u_{d_n,k} = u_{d'_n,k+1}) \tag{6.8}$$

We see that $\rho_{\mathsf{U},k,k+1}$ is the composition of the projection that discards $(u_{a,k}, \ldots, u_{c,k})$ from u_k followed by the inverse projection that adds $(u_{e',k+1}, \ldots, u_{g',k+1})$ to get u_{k+1}. Since projections are functions, this displays $\rho_{\mathsf{U},k,k+1}$ in difunctional form.

Regular relations between the various levels such as these, possess compositionality properties not shared by arbitrary chains of regular relations. For suppose U_k and U_{k+1} are related via $\rho_{\mathsf{U},k,k+1}$ as in (6.8).

For the relationship between levels $k+1$ and $k+2$, let us relabel U_{k+1} by letting a typcal value in U_{k+1} be $u_{k+1} = (u_{a',k+1}, \ldots, u_{c',k+1}, u_{e'_1,k+1}, \ldots, u_{e'_m,k+1})$, and let a typcal value in U_{k+2} be $u_{k+2} = (u_{e''_1,k+2}, \ldots, u_{e''_m,k+2}, u_{f'',k+2}, \ldots, u_{h'',k+2})$, where subspace $\mathsf{U}_{a',k+1} \times \ldots \times \mathsf{U}_{c',k+1}$ is removed from U_{k+1} in the passage from level $k+1$ to level $k+2$, and subspace $\mathsf{U}_{f'',k+2} \times \ldots \times \mathsf{U}_{h'',k+2}$ is added. Suppose $\mathsf{U}_{e'_1,k+1} = \mathsf{U}_{e''_1,k+2}, \ldots, \mathsf{U}_{e'_m,k+1} = \mathsf{U}_{e''_m,k+2}$ are the common types between levels $k+1$ and $k+2$, m of them, identified as indicated.

Then:

$$\rho_{\mathsf{U},k+1,k+2}(u_{k+1}, u_{k+2}) \equiv (u_{e'_1,k+1} = u_{e''_1,k+2} \wedge \ldots \wedge u_{e'_m,k+1} = u_{e''_m,k+2}) \tag{6.9}$$

Let θ_{k+1} be the relabelling function so that θ_{k+1} captures the bijection between the labels of the $(u_{d'_1,k+1}, \ldots, u_{d'_n,k+1}, u_{e',k+1}, \ldots, u_{g',k+1})$ decomposition of u_{k+1} and the labels of the $(u_{a',k+1}, \ldots, u_{c',k+1}, u_{e'_1,k+1}, \ldots, u_{e'_m,k+1})$ decomposition. Then we define the composition of $\rho_{\mathsf{U},k,k+1}$ and $\rho_{\mathsf{U},k+1,k+2}$ by:

$$\begin{aligned}
&(\rho_{\mathsf{U},k,k+1} \,;\, \rho_{\mathsf{U},k+1,k+2})(u_k, u_{k+2}) \equiv \\
&\quad ((u_{d_{j1},k} = u_{d'_{j1},k+1} \wedge \theta_{k+1}(d'_{j1,k}) = e'_{j1,k} \wedge u_{e'_{j1},k+1} = u_{e''_{j1},k+2}) \wedge \\
&\quad (u_{d_{j2},k} = u_{d'_{j2},k+1} \wedge \theta_{k+1}(d'_{j2,k}) = e'_{j2,k} \wedge u_{e'_{j2},k+1} = u_{e''_{j2},k+2}) \wedge \\
&\qquad\quad \ldots \qquad \ldots \qquad \ldots \\
&\quad (u_{d_{jl},k} = u_{d'_{jl},k+1} \wedge \theta_{k+1}(d'_{jl,k}) = e'_{jl,k} \wedge u_{e'_{jl},k+1} = u_{e''_{jl},k+2}))
\end{aligned} \tag{6.10}$$

3. $\mathsf{U}_{c,k}$ not present at level $k+1$ can arise when a feature is removed (by being completely overridden) at level $k+1$. Why introduce something earlier only to override it later? In a monolithic development it makes no sense. But in a long lived development process, when there might be millions of units of an earlier design out in the field, it will be impossible to pretend that a feature installed earlier can simply be erased from the development. Telephony is the obvious example. If a feature is completely overridden in this manner it need not be implemented, and so the data that it would otherwise need, i.e. $\mathsf{U}_{c,k}$, can be removed from the state space.

where $\mathsf{d}'_{j1,k}, \mathsf{d}'_{j2,k}, \ldots, \mathsf{d}'_{jl,k}$ lists all the types at level $k+1$ that are both common with types at level k, and (under a different name) with types at level $k+2$. Since this is a composition of a projection with an inverse projection just as before, it is a difunctional presentation of $\rho_{\mathsf{U},k,k+1} \,;\, \rho_{\mathsf{U},k+1,k+2}$ and thus is total, surjective and regular. Furthermore it is easy to see that the composition of these relations is associative.

We will assume that similar mechanisms hold for the input and output spaces $\mathsf{I}_{Op,k}$ and $\mathsf{O}_{Op,k}$, whose incarnations are related by total surjective regular relations $\rho_{\mathsf{I}\text{-}Op,k,k+1}$ and $\rho_{\mathsf{O}_{Op},k,k+1}$ constructed in the same manner, i.e. by discarding some component types and incorporating new ones. In particular their compositions are also total, surjective and regular[4].

One pathological situation that we must mention is when there are no common types between two levels which are adjacent, or become related as a result of one or more compositions. In this case the ρ relation becomes universal (an empty conjunction). We will assume that the developments we are considering display enough coherence that this situation does not arise. Since in practical feature engineering situations, adding new features is far more prevalent than removing them completely, this is a reasonable assumption.

The passage from level k to level $k+1$ can now be subdivided into the following steps. We first add in any types newly required at level $k+1$ to the level k FOD in the manner just described. Next we modify the FOD to incorporate any new feature(s) as we will show below, given that the state and I/O spaces are now adequate to accommodate them. Finally we can project out any types no longer needed due to their individual features having empty active domains as a consequence of their being completely overridden, using another application of the above techniques.

At this point we can reconsider the modification of a FOD on the assumption that the signatures of all FEs entering into the discourse are the same. As previously the modifications are of two kinds: the alteration of a boolean condition, and the infiltration of new features into the FOD which we now discuss in more detail.

Suppose we have an FOD $\Phi[\phi]$, which we want to change in the vicinity of ϕ. Maybe we want to introduce a new feature f_c to act alongside ϕ. Unfortunately the domains of ϕ and f_c overlap, so we cannot just move to $\Phi[(\phi \Cup f_c)]$ because the subexpression $(\phi \Cup f_c)$ is ill defined. We could adopt the possibilities $\Phi[(\phi \Lleft f_c)]$ or $\Phi[(f_c \Lleft \phi)]$ but we might have to acknowledge that the behaviour of ϕ or f_c alone in the region of overlap is no longer appropriate in the presence of the other. Instead we define a new

4. There is a subtlety with input and output spaces that is largely hidden in the case of the system state. If the output space (say) needs to acquire a new cartesian factor, going from J to $\mathsf{J} \times \mathsf{K}$, because of the demands of some new feature f_{new} in the operation, then even when f_{new} is not being used, a value from K must be output for all steps of the operation, even if it has to be a default value, because outputs are now pairs. Although not inconceivable, this is quite a drastic redesign of the operation as a whole. What is more likely in practice is that f_{new} will output some hitherto unused values from J to accomplish its task (making the construction more like a sum than a product, a technical ramification we will not pursue further). Similar remarks apply to inputs. State is different because it persists from step to step, so state components of no interest to the feature currently being invoked remain undisturbed, and do not impact on the current step. Only at system initialisation time do we see an effect as for inputs and outputs, when values have to be supplied simultaneously for all state components.

feature f_x to take care of the interaction, making sure that the domain of f_x is precisely $\mathrm{dom}(f_c) \cap \mathrm{dom}(\phi)$. Now we can design $\Phi[((\phi \cup f_c) \nleftarrow f_x)]$, avoiding the unnaturalness of the $(\phi \nleftarrow f_c)$ or $(f_c \nleftarrow \phi)$ partial solutions, and also of $\Phi[(\phi \cup f_c)]$, where the nondeterminism between ϕ and f_c in the overlapping region may be regarded as equally inappropriate.

The schema for modifications of FODs that we thus consider is the rewrite of $\Phi[\phi]$ to $\Phi[\gamma[\phi]]$. Here $\Phi[\_]$ is a FE context, i.e. a FE with a hole [_], within which we apply the rewrite rule:

$$[\_] \Rightarrow \gamma[\_] \tag{6.11}$$

Here $\gamma[\_]$ is itself also a FE context that is interposed between Φ and ϕ. We will assume that all the modifications of the second kind that we permit to be applied to a FOD are of this form.

This spells out in detail what is meant by 'infiltration of new features'. The interposition of γ induces an operator $Sy_\gamma(\_)$ on the $sy(\phi)$ set passed up into Φ and an operator $In_\gamma(\_)$ on the $in(\phi)$ set passed down into ϕ thus:

$$Sy_\gamma(sy(\phi)) \equiv sy(\gamma(\phi)) \tag{6.12}$$

$$In_\gamma(in(\phi)) \equiv in(\phi)_{\gamma[\_]} \tag{6.13}$$

where the notation on the right hand side of (6.13) reflects the fact that in reality, $in(\phi)$ depends more on the context of ϕ than on ϕ itself. More generally $Sy_\gamma(\psi)$ and $In_\gamma(\psi)$ may be understood as similarly induced for any subexpression ψ of Φ.

The equations in Definition 6.5 permit the calculation of Sy_γ and In_γ from γ in terms of the other quantities of the system. For example, if for $\gamma[\_]$ we take $(([\_] \cup f_c) \nleftarrow f_x)$ as above, then taking all the other facts about that example into account we can calculate:

$$sy(\gamma(\phi)) = sy(\phi) \cup (\mathrm{dom}(f_c) \cap in(\phi)_{\Phi[\_]}) \tag{6.14}$$

$$in(\phi)_{\gamma[\_]} = in(\phi) - \mathrm{dom}(f_x) \tag{6.15}$$

so that in this example:

$$Sy_\gamma(\_) = (\_) \cup (\mathrm{dom}(f_c) \cap in(\phi)_{\Phi[\_]}) \tag{6.16}$$

$$In_\gamma(\_) = (\_) - \mathrm{dom}(f_x) \tag{6.17}$$

The presence of $in(\phi)_{\Phi[\_]}$ in (6.16) shows that in general Sy_γ and In_γ are not independent of one another.

Definition 6.7 We call an individual feature f_d in a FOD active iff $\mathrm{dom}_{act}(f_d)$ is nonempty, and say that f_d is active at some $(u, i) \in \mathrm{dom}(f_d)$ iff $(u, i) \in \mathrm{dom}_{act}(f_d)$.

6.3 Feature Evolution via Retenchment

Having described how we can move from level to level both in terms of how FODs alter and how we can describe the relationship between the relevant state and other spaces, we now turn our attention to describing the retrenchments that capture this process.

We define the retrieve, within, and output relations between successive layers thus:

$$G_{k+1}(u_k, u_{k+1}) \equiv \rho_{\mathsf{U},k,k+1}(u_k, u_{k+1}) \tag{6.18}$$

$$P_{Op,k+1}(i_k, i_{k+1}, u_k, u_{k+1}) \equiv \rho_{\mathsf{I}_{Op},k,k+1}(i_k, i_{k+1}) \tag{6.19}$$

$$O_{Op,k+1}(o_k, o_{k+1}; u'_k, u'_{k+1}, i_k, i_{k+1}, u_k, u_{k+1}) \equiv \rho_{\mathsf{O}_{Op},k,k+1}(o_k, o_{k+1}) \tag{6.20}$$

In (6.19) there is an implicit cartesian product with a universal relation from U_k to U_{k+1} on the right hand side, and in (6.20) there is on the right hand side an implicit cartesian product with a universal relation from the values of the variables u'_k, i_k, u_k, to the values of the variables u'_{k+1}, i_{k+1}, u_{k+1}. Since the cartesian product of regular relations is regular, we conclude that the within and output relations will also be total surjective regular relations whose compositions are total, surjective and regular.

Now we form the relation $G_{k+1}(u_k, u_{k+1}) \wedge P_{Op,k+1}(i_k, i_{k+1}, u_k, u_{k+1})$ by forming the cartesian product of $G_{k+1}(u_k, u_{k+1})$ with a universal relation from the inputs at level k to those at level $k+1$, and taking the intersection of the resulting relation with $P_{Op,k+1}(i_k, i_{k+1}, u_k, u_{k+1})$. Since the intersection of regular relations is regular, we see that the relations $G_{k+1} \wedge P_{Op,k+1}$ are total surjective regular relations whose compositions are total, surjective and regular. A similar argument shows that the relations $G_{k+1}(u'_k, u'_{k+1}) \wedge O_{Op,k+1}(o_k, o_{k+1}; u'_k, u'_{k+1}, i_k, i_{k+1}, u_k, u_{k+1})$ also have this property.

We have almost shown that the retrenchments we are developing have regular data. To say something about the concedes relations, we must first discuss the transition relations for Op_k and Op_{k+1}. We do so in the context of a number of assumptions.

Assumption 6.8 Model Completeness. We have discussed this above, where we inferred that it implies that there are no situations of interest regarding models at adjacent levels in the series, whose before-states and inputs are not in the scope of $P^{\mathrm{Def}}{}_{Op,k+1}(i_k, i_{k+1}, u_k, u_{k+1})$, where $P^{\mathrm{Def}}{}_{Op,k+1}$ is built according to (2.7) using the $G_{k+1}(u_k, u_{k+1}) \wedge P_{Op,k+1}(i_k, i_{k+1}, u_k, u_{k+1})$ just constructed.

Assumption 6.9 Interfeature Independence. By this we mean that distinct features do not encroach on each other's work. There is no point in designing a new feature to duplicate the work of an old one[5], so if f_{c} and f_{d} are distinct features present in levels k and $k+j$, ($j \geq 0$) then we will always have the negation of the analogue of $\overline{G}_{Op,(k+1,k+j)}$ for them:

$$\begin{aligned}\neg\, (&G_{(k+1,k+j)}(u_k, u_{k+j}) \wedge P_{Op,(k+1,k+j)}(i_k, i_{k+j}, u_k, u_{k+j}) \wedge \\ &stp_{f_{\mathrm{c}}}(u_k, i_k, u'_k, o_k) \wedge stp_{f_{\mathrm{d}}}(u_{k+j}, i_{k+j}, u'_{k+j}, o_{k+j}) \wedge \\ &G_{(k+1,k+j)}(u'_k, u'_{k+j}) \wedge O_{Op,(k+1,k+j)}(o_k, o_{k+j}; u'_k, u'_{k+j}, i_k, i_{k+j}, u_k, u_{k+j}))\end{aligned} \tag{6.21}$$

In (6.21), $G_{(k+1,k+j)}$ is the composition G_{k+1} ; G_{k+2} ; … ; G_{k+j}, defaulting to the identity if $j = 0$, and to G_{k+1} if $j = 1$. Similarly for the other relations.

Assumption 6.10 Interfeature Determinism. By this we mean that for any (u_k, i_k), there should be at most one feature f_{d} of the operation with a transition u_k -(i_k,$f_{\mathrm{d},k}$, o_k)-› u'_k emerging from (u_k, i_k). In other words, the active domains of distinct features should not intersect. Users have a right to expect predictable behaviour for a given starting condition, and we assume such nuggets of predictable behaviour are

5. This is perhaps a bit hasty. It may well be that on occasion we want to 'clean up' a longlived development by removing (i.e. overriding) some tired old features and replacing them with shiny new ones that (at least some of the time) do the same job as the old ones. But we will ignore this possibility here for simplicity.

encapsulated within individual features. That is not to say that individual features cannot themselves be nondeterministic when there are justifiable requirements reasons for them being so, (for example seat allocation on budget airline flights), but that is a different issue.

Interfeature determinism excludes certain FODs, primarily ones containing 'naked unions' such as $(f_c \cup f_d)$ where the domains of f_c and f_d overlap and the union is not masked by some form of overriding. Not all FODs containing overlapping unions are disbarred. Referring to our previous example $\gamma(f_d) = ((f_d \cup f_c) \nleftarrow f_x)$, where $\mathrm{dom}(f_x) = \mathrm{dom}(f_c) \cap \mathrm{dom}(f_d)$, we see that despite the union, it is interfeature deterministic because of the override on the overlap. Such FODs can be rewritten to remove the unions, eg. $((f_d \nleftarrow f_c) \nleftarrow f_x)$. Although equivalent to the former expression this could be less appropriate as regards eloquence in expressing requirements, as we noted before.

Let us reflect a little on these assumptions. While model completeness can easily be understood as an uncontroversial requirement of the development methodology, the status of interfeature independence and interfeature determinism is more open to question. For example one can certainly imagine designing features that partially duplicate each other's work, as noted already. But then one could focus the analysis lower down, by introducing a notion of subfeature, such that each feature would be a union asserted disjoint of a family of subfeatures, and such that individual subfeatures capture the unique pieces of functionality shared among more than one parent feature. One could then consider the legitimacy of the idea of intersubfeature independence. Similarly one can imagine designing operations requiring features that partially overlap in a common subdomain where both are active. In such a case one could again refocus the analysis on subfeatures that capture the common behaviour and consider the legitimacy of intersubfeature determinism.

In both scenarios the crucial issue amounts to 'What is a (sub)feature?', a meta level issue that is related to the naturalness or otherwise of different subdivisions of the functionality offered by an operation. For the sake of having a relatively straightforward scenario to illustrate our retrenchment theory we will accept the assumptions stated without further comment.

We now consider a development step from level k to level $k+1$, given either by modifying some condition in the FOD $\Phi_k[\phi]$ of Op_k, or by applying a rewrite rule like [_] => γ[_] in (7.11) to some subexpression ϕ of $\Phi_k[\phi]$. The fact that neither option need act at the root of the parse tree of $\Phi_k[\phi]$ by an application of a FE constructor, blocks the analysis of what can happen via the structural induction route, at least in any straightforward way. Fortunately, the NF theorem allied with the assumptions above gives us another route towards an analysis.

Consider some (u_k, i_k) from which a level k transition emerges. By model completeness, there will be a (u_{k+1}, i_{k+1}) related to (u_k, i_k) by ${P^{\mathrm{Def}}}_{Op,k+1}(i_k, i_{k+1}, u_k, u_{k+1})$, from which a level $k+1$ transition emerges, and vice versa. We fix (u_k, i_k) and (u_{k+1}, i_{k+1}) for the next few paragraphs. By interfeature determinism, both transitions will belong to features $f_{a,k}$ and $f_{y,k+1}$, each unique within the context of its level. There are now three possibilities, which by interfeature determinism again are mutually exclusive, (P1), (P2), (P3):

(P1) $f_{y,k+1}$ is the image under G_{k+1}, $P_{Op,k+1}$, $O_{Op,k+1}$ of $f_{a,k}$.

In this case for every level $k+1$ transition u_{k+1} -$(i_{k+1}, f_{a,k+1}, o_{k+1})$-› u'_{k+1} there will be a level k transition u_k -$(i_k, f_{a,k}, o_k)$-› u'_k related to it by:

$$\begin{aligned}&(G_{k+1}(u_k, u_{k+1}) \wedge P_{Op,k+1}(i_k, i_{k+1}, u_k, u_{k+1}) \wedge \\ &\quad stp_{f_a}(u_k, i_k, u'_k, o_k) \wedge stp_{f_a}(u_{k+1}, i_{k+1}, u'_{k+1}, o_{k+1}) \wedge \\ &\quad G_{k+1}(u'_k, u'_{k+1}) \wedge O_{Op,k+1}(o_k, o_{k+1}; u'_k, u'_{k+1}, i_k, i_{k+1}, u_k, u_{k+1}))\end{aligned} \tag{6.22}$$

because all of the level $k+1$ transitions are constructed precisely by mapping all of the level k transitions through G_{k+1}, $P_{Op,k+1}$, G'_{k+1}, $O_{Op,k+1}$. In this case we reestablish the retrieve relation, and have $\text{pre}^{\text{Ret}}{}_{Op,k+1}(u_k, i_k, u_{k+1}, i_{k+1})$.

(P2) $f_{y,k+1}$ is the image under $G_{k+1}, P_{Op,k+1}, O_{Op,k+1}$ of some level k feature $f_{b,k} \neq f_{a,k}$.

This can arise because a condition somewhere in $\Phi_k[\phi]$ was modified, making $f_{b,k+1}$ active at (u_{k+1}, i_{k+1}) whereas $f_{a,k}$ was active at (u_k, i_k). In this case, interfeature independence, in the shape of (6.21), ensures that we have:

$$\begin{aligned}&\neg\,(G_{k+1}(u_k, u_{k+j}) \wedge P_{Op,k+1}(i_k, i_{k+1}, u_k, u_{k+1}) \wedge \\ &\quad stp_{f_a}(u_k, i_k, u'_k, o_k) \wedge stp_{f_b}(u_{k+1}, i_{k+1}, u'_{k+1}, o_{k+1}) \wedge \\ &\quad G_{k+1}(u'_k, u'_{k+1}) \wedge O_{Op,k+1}(o_k, o_{k+1}; u'_k, u'_{k+1}, i_k, i_{k+1}, u_k, u_{k+1}))\end{aligned} \tag{6.23}$$

which is equivalent to:

$$\begin{aligned}&G_{k+1}(u_k, u_{k+j}) \wedge P_{Op,k+1}(i_k, i_{k+1}, u_k, u_{k+1}) \wedge \\ &stp_{f_a}(u_k, i_k, u'_k, o_k) \wedge stp_{f_b}(u_{k+1}, i_{k+1}, u'_{k+1}, o_{k+1}) \;\Rightarrow \\ &\quad \neg\,(G_{k+1}(u'_k, u'_{k+1}) \wedge O_{Op,k+1}(o_k, o_{k+1}; u'_k, u'_{k+1}, i_k, i_{k+1}, u_k, u_{k+1}))\end{aligned} \tag{6.24}$$

so that the antecedents of (6.24) are sufficient to imply the whole of what would be the default concedes relation for this (u_k, i_k) and (u_{k+1}, i_{k+1}). As a result, when building the complete concedes relation, including these antecedents will be enough to express what is needed.

(P3) $f_{y,k+1}$ is *not* the $G_{k+1}, P_{Op,k+1}, O_{Op,k+1}$ image of some level k feature $f_{b,k} \neq f_{a,k}$.

In this case $f_{y,k+1}$ is a new feature in the FOD, freshly introduced via [_] => γ[_]. The same arguments as in (P2) apply, and the analogues of (6.23) and (6.24) hold.

From now on we treat (P2) and (P3) simultaneously, calling the level $k+1$ feature f_e regardless of its origins. So we have for the given (u_k, i_k) and (u_{k+1}, i_{k+1}):

$$\begin{aligned}&(G_{k+1}(u_k, u_{k+1}) \wedge P_{Op,k+1}(i_k, i_{k+1}, u_k, u_{k+1}) \wedge \\ &stp_{f_a}(u_k, i_k, u'_k, o_k) \wedge stp_{f_e}(u_{k+1}, i_{k+1}, u'_{k+1}, o_{k+1}))\end{aligned} \tag{6.25}$$

To build the complete concedes relation we must accumulate these cases taking care to delineate how the pieces fit together according to the partitions of $\mathsf{U}_k \times \mathsf{I}_{Op,k}$ and $\mathsf{U}_{k+1} \times \mathsf{I}_{Op,k+1}$ induced from the NF theorem.

Definition 6.11 We define the offdiagonal active domain of a pair of distinct features $f_{b,k}$ and $f_{g,k+1}$ with respect to the given development step by:

$$\begin{aligned}&\text{dom}_{\text{ODact}}(f_{b,k}, f_{g,k+1}) \;\equiv \\ &\quad \{(u_k, i_k, u_{k+1}, i_{k+1}) \mid (u_k, i_k) \in \text{dom}_{\text{act}}(f_{b,k}), (u_{k+1}, i_{k+1}) \in \text{dom}_{\text{act}}(f_{g,k+1}), \\ &\qquad (\text{feature } f_b \neq \text{feature } f_g) \wedge G_{k+1}(u_k, u_{k+1}) \wedge P_{Op,k+1}(i_k, i_{k+1}, u_k, u_{k+1}) \wedge \\ &\qquad (\exists\, u'_k, o_k, u'_{k+1}, o_{k+1} \bullet \\ &\qquad\quad stp_{f_b}(u_k, i_k, u'_k, o_k) \wedge stp_{f_g}(u_{k+1}, i_{k+1}, u'_{k+1}, o_{k+1}))\}\end{aligned} \tag{6.26}$$

We can now define the complete concedes relation for levels k and $k+1$ as follows. What we do is tantamount to writing out the default concedes relation that relates Op_k and Op_{k+1} except that we decompose Op_k and Op_{k+1} into their various features according to the offdiagonal active domains for various distinct feature pairs:

$$\begin{array}{l} C_{Op,k+1}(u'_k, u'_{k+1}, o_k, o_{k+1}; i_k, i_{k+1}, u_k, u_{k+1}) \equiv \\ \quad (G_{k+1}(u_k, u_{k+1}) \wedge P_{Op,k+1}(i_k, i_{k+1}, u_k, u_{k+1}) \wedge \\ \quad (\bigvee_{(f_b \neq f_g)} ((u_k, i_k, u_{k+1}, i_{k+1}) \in \mathrm{dom}_{\mathrm{ODact}}(f_{b,k}, f_{g,k+1}) \wedge \\ \qquad\qquad stp_{f_b}(u_k, i_k, u'_k, o_k) \wedge stp_{f_g}(u_{k+1}, i_{k+1}, u'_{k+1}, o_{k+1})))) \end{array} \tag{6.27}$$

From (6.27) we conclude that in (P2) and (P3), we have $\mathrm{pre}^{\mathrm{Con}}{}_{Op,k+1}(u_k, i_k, u_{k+1}, i_{k+1})$ for the concedes relation (6.27), as we would expect.

Proposition 6.12 The concedes relation (6.27) is regular.

Proof. Consider $C_{Op,k+1}$. Let $\pi_k(u'_k, o_k; i_k, u_k)$ be the projection that takes (u'_k, o_k, i_k, u_k) to (i_k, u_k); and let $\pi_{k+1}(u'_{k+1}, o_{k+1}; i_{k+1}, u_{k+1})$ be the projection that takes $(u'_{k+1}, o_{k+1}, i_{k+1}, u_{k+1})$ to (i_{k+1}, u_{k+1}). Let $\iota_k(u'_k, o_k, i_k, u_k)$ be the injection of $\mathrm{dom}(C_{Op,k+1})$ into $\mathsf{U}_k \times \mathsf{O}_{Op,k} \times \mathsf{I}_{Op,k} \times \mathsf{U}_k$, and let $\iota_{k+1}(u'_{k+1}, o_{k+1}, i_{k+1}, u_{k+1})$ be the injection of $\mathrm{rng}(C_{Op,k+1})$ into $\mathsf{U}_{k+1} \times \mathsf{O}_{Op,k+1} \times \mathsf{I}_{Op,k+1} \times \mathsf{U}_{k+1}$. Let $f; g^{-1}$ be a difunctional presentation of $G_{k+1} \wedge P_{Op,k+1}$. Then we have a difunctional presentation of $C_{Op,k+1}$ given by:

$$\begin{array}{l} C_{Op,k+1}(u'_k, u'_{k+1}, o_k, o_{k+1}; i_k, i_{k+1}, u_k, u_{k+1}) \equiv \\ \quad (\iota_k(u'_k, o_k, i_k, u_k) \,;\, \pi_k(u'_k, o_k; i_k, u_k) \,;\, f) \,; \\ \quad (\iota_{k+1}(u'_{k+1}, o_{k+1}, i_{k+1}, u_{k+1}) \,;\, \pi_{k+1}(u'_{k+1}, o_{k+1}; i_{k+1}, u_{k+1}) \,;\, g)^{-1} \end{array} \tag{6.28}$$

This shows that $C_{Op,k+1}$ is regular. ☺

So our retrenchments have regular data. The next question is whether they respect their regular data. Here the answer is no. For consider the following situation in which there is no I/O. We have at level k, a value u of the state variable u_k, from which a transition of feature $f_{a,k}$ issues. At level level $k+1$ the state variables are u_{k+1} which consist of a pair of values (u, w) where the value w is needed by feature $f_{e,k+1}$, newly introduced at level $k+1$. The retrieve relation $G_{k+1}(u_k, u_{k+1})$ is just the inverse projection that relates u at level k to (u, w) for any w at level $k+1$. Now we suppose that for a value w_1 there is a level $k+1$ transition of $f_{a,k+1}$, namely (u, w_1) -$(f_{a,k+1})$-› (u', w'_1), and for a value w_2 there is a level $k+1$ transition of $f_{e,k+1}$, namely (u, w_2) -$(f_{e,k+1})$-› (u'_2, w'_2). (This is quite reasonable since there is no requirement for any individual feature to be model complete.) In such a case, for the (u, w_1) transition, we would establish $G_{k+1}(u', (u', w'_1))$, since the level $k+1$ transition would just be a copy of a level k transition, u -$(f_{a,k})$-› u'; and for the (u, w_2) transition we would have $C_{Op,k+1}(u'_k, u'_{k+1} \ldots)$ because feature $f_{e,k+1}$ is active. Now (u, w_1) and (u, w_2) are in the same $(G_{k+1} \wedge P_{Op,k+1})$ equivalence class, since they are both $(G_{k+1} \wedge P_{Op,k+1})$-related to u at level k. But (u', w'_1) and (u'_2, w'_2) are not in the same $C_{Op,k+1}$ equivalence class, since $C_{Op,k+1}$ is only defined when the level k feature and level $k+1$ feature are different, by (6.27); in particular $C_{Op,k+1}$ is not defined for (u', w'_1). So condition (4) of Definition 3.16 is violated and the retrenchments with concedes relations (6.27) do not respect their regular data. In particular, tidiness is too strong a property to expect in feature engineering. On the other hand, we have the following.

Proposition 6.13 The retrenchments given by (6.18)-(6.20) and (6.27) are neat.

Proof. Consider some (u_k, i_k) related to some (u_{k+1}, i_{k+1}) by $G_{k+1} \wedge P_{Op,k+1}$. If we have the same feaure active at both levels, then we have $\text{pre}^{\text{Ret}}{}_{Op,k+1}(u_k, i_k, u_{k+1}, i_{k+1})$. By interfeature determinism, no other features can be active for this (u_k, i_k) and (u_{k+1}, i_{k+1}) so for the after-states and outputs etc. related to these (u_k, i_k) and (u_{k+1}, i_{k+1}), $C_{Op,k+1}$ will not be defined (because we were careful to relate only offdiagonal active domains via $C_{Op,k+1}$) and so $\text{pre}^{\text{Con}}{}_{Op,k+1}(u_k, i_k, u_{k+1}, i_{k+1})$ will not be valid. Contrarily, if for (u_k, i_k) and (u_{k+1}, i_{k+1}) two different features are active, and we have $C_{Op,k+1}$, and thus $\text{pre}^{\text{Con}}{}_{Op,k+1}(u_k, i_k, u_{k+1}, i_{k+1})$ is valid, interfeature determinism says no other features can be active for these (u_k, i_k) and (u_{k+1}, i_{k+1}), and interfeature independence allows us to conclude that $G'_{k+1} \wedge O_{Op,k+1}$ will not be valid 'fortuitously', precluding the validity of $\text{pre}^{\text{Ret}}{}_{Op,k+1}(u_k, i_k, u_{k+1}, i_{k+1})$, a possibility we must guard against since $G'_{k+1} \wedge O_{Op,k+1}$ are globally defined on $\mathsf{U}_k \times \mathsf{O}_{Op,k} \times \mathsf{I}_{Op,k} \times \mathsf{U}_k$, and $\mathsf{U}_{k+1} \times \mathsf{O}_{Op,k+1} \times \mathsf{I}_{Op,k+1} \times \mathsf{U}_{k+1}$. Neatness thus follows. ☺

Even though we have neatness of the retrenchments for individual development steps, we do not in general have strong enough properties to be able to deduce that compositions of such steps (done according to (4.9)-(4.12)) yield neat retrenchments, nor that the composition of such retrenchments, if neat, is necessarily associative. In fact it is easy to see that the output relations satisfy the general closedness conditions (5.5)-(5.6); but there is no reason to suppose that the concedes relations (6.27) are benign enough. (The composition of the neat retrenchments is of course associative when it is done in the sense of the normal composition of retrenchments, i.e. (2.3)-(2.6); rather than (4.9)-(4.12) together with Theorem 4.5.)

Counterexample 6.14 Consider an operation defined at level 0 by $(f_\text{b} \lhd_p f_\text{a})$, where we suppose that the domains of the two interfeature independent features f_a and f_b are equal. Consider the development step in which p is replaced by $\neg p$, giving at level 1, $(f_\text{b} \lhd_{\neg p} f_\text{a})$. A nontrivial concedes relation will be needed to describe the fact that one behaviour is replaced by another in the whole domain. Consider the further development step in which $\neg p$ is replaced by p, giving at level 2, $(f_\text{b} \lhd_p f_\text{a})$ again. Another nontrivial concedes relation will be needed here to undo the damage caused by the first one; in fact it will be the transpose of the first concedes relation. However the composition of these development steps is the identity development step, which needs only the empty concedes relation. This is not equal to the composition of the component concedes relations according to any of the schemes that we considered above, since the composition of a nontrivial relation and its transpose will in general include a nonempty subrelation of the identity relation. We therefore see that in this example, for a typical transition of the operation, we will validate both the $\overline{G}$ and $\overline{C}$ conditions for the composition, and the composition will not be neat. Actually, if in Fig. 2 we remove the rear intermediate level transition and the left hand G_1/G_2 pair it issues from, we get an illustration of the situation just described.

The general form of modification to FODs that we allow in the passage from one level to another makes it difficult to say anything too specific about how the partition of the $\mathsf{U}_k \times \mathsf{I}_{Op,k}$ promised by the NF theorem evolves from level to level. Of course in any particular case, the calculational strategy of Definition 6.5 et seq. will yield a particular answer. However there are some special cases, in which the partition evolves in a more systematic manner, and we illustrate one of these.

Suppose all the FODs for Op are simply sequences of overrides, such as:

$$Op_k \;=\; f_{\text{Def},k} \lhd f_{\text{q},k} \lhd f_{\text{p},k} \cdots f_{\text{b},k} \lhd f_{\text{a},k} \tag{6.29}$$

Here f_{Def} is some default feature that guarantees model completeness no matter what. (Thus we tacitly assume that at each level the limits of its domain serve to define model completeness, and that no feature is defined beyond the domain of f_{Def} at any level.) Then the progression from level k to level $k+1$ is just the insertion of some $f_{\mathrm{g},k+1}$ into the above sequence, changing $\ldots f_{\mathrm{d},k} \lessdot f_{\mathrm{c},k} \ldots$ into $\ldots f_{\mathrm{d},k+1} \lessdot f_{\mathrm{g},k+1} \lessdot f_{\mathrm{c},k+1} \cdots$.

It is now clear that the only alteration to the definition of Op will occur on $\mathrm{dom}(f_{\mathrm{g},k+1}) - \mathrm{dom}(f_{\mathrm{c},k+1} \cup \ldots \cup f_{\mathrm{a},k+1})$ at level $k+1$. (N.B. We no longer have to worry separately about offdiagonal active domains because of the simple structure of (6.29).) On this part, the concedes relation $C_{Op,k+1}$ will be defined and will equal:

$$\begin{aligned}
&C_{Op,k+1}(u'_k, u'_{k+1}, o_k, o_{k+1}; i_k, i_{k+1}, u_k, u_{k+1}) \equiv \\
&\quad (G_{k+1}(u_k, u_{k+1}) \wedge P_{Op,k+1}(i_k, i_{k+1}, u_k, u_{k+1}) \wedge \\
&\quad (((u_k, i_k) \in (\mathrm{dom}(f_{\mathrm{d},k}) - \mathrm{dom}(f_{\mathrm{c},k}) \ldots - \mathrm{dom}(f_{\mathrm{a},k})) \wedge \\
&\qquad stp_{f_{\mathrm{d}}}(u_k, i_k, u'_k, o_k)) \vee \\
&\qquad \ldots \qquad \ldots \vee \\
&\quad ((u_k, i_k) \in (\mathrm{dom}(f_{\mathrm{Def},k}) - \mathrm{dom}(f_{\mathrm{q},k}) - \ldots - \mathrm{dom}(f_{\mathrm{d},k}) \\
&\qquad - \mathrm{dom}(f_{\mathrm{c},k}) - \ldots - \mathrm{dom}(f_{\mathrm{a},k})) \wedge stp_{f_{\mathrm{Def}}}(u_k, i_k, u'_k, o_k))) \wedge \\
&\quad ((u_{k+1}, i_{k+1}) \in (\mathrm{dom}(f_{\mathrm{g},k+1}) - \mathrm{dom}(f_{\mathrm{c},k+1}) - \ldots - \mathrm{dom}(f_{\mathrm{a},k+1})) \wedge \\
&\qquad stp_{f_{\mathrm{g}}}(u_{k+1}, i_{k+1}, u'_{k+1}, o_{k+1}))
\end{aligned} \tag{6.30}$$

where we have assumed that $\mathrm{dom}(f_{\mathrm{g},k+1}) \subseteq \mathrm{dom}(f_{\mathrm{Def},k+1})$, and the dissection of the concedes relation into specific feature stp relations on the various domains is just the decomposition of $stp_{Op_k}(u_k, i_k, u'_k, o_k)$ and $stp_{Op_{k+1}}(u_{k+1}, i_{k+1}, u'_{k+1}, o_{k+1})$ into their constituents.

When we have a sequence of such modifications, the concedes relations that describe the resulting operation depend on the order in which different features are inserted into the overall FOD. Eg. suppose after inserting $f_{\mathrm{g},k+1}$ above we next insert feature $f_{\mathrm{h},k+2}$. Then we have two different outcomes depending on whether $f_{\mathrm{h},k+2}$ is overridden by $f_{\mathrm{g},k+2}$ or not, i.e. whether it occurs lower down the override hierarchy.

Suppose it is inserted next in priority after $f_{\mathrm{g},k+1}$, giving $\ldots f_{\mathrm{d},k+2} \lessdot f_{\mathrm{h},k+2} \lessdot f_{\mathrm{g},k+2} \lessdot f_{\mathrm{c},k+2} \ldots$. Then the corresponding concedes relation reads:

$$\begin{aligned}
&C_{Op,k+2}(u'_{k+1}, u'_{k+2}, o_{k+1}, o_{k+2}; i_{k+1}, i_{k+2}, u_{k+1}, u_{k+2}) \equiv \\
&\quad (G_{k+2}(u_{k+1}, u_{k+2}) \wedge P_{Op,k+2}(i_{k+1}, i_{k+2}, u_{k+1}, u_{k+2}) \wedge \\
&\quad (((u_{k+1}, i_{k+1}) \in (\mathrm{dom}(f_{\mathrm{d},k+1}) - \mathrm{dom}(f_{\mathrm{g},k+1}) - \mathrm{dom}(f_{\mathrm{c},k+1}) \ldots - \mathrm{dom}(f_{\mathrm{a},k+1})) \wedge \\
&\qquad stp_{f_{\mathrm{d}}}(u_{k+1}, i_{k+1}, u'_{k+1}, o_{k+1})) \vee \\
&\qquad \ldots \qquad \ldots \vee \\
&\quad ((u_{k+1}, i_{k+1}) \in (\mathrm{dom}(f_{\mathrm{Def},k+1}) - \mathrm{dom}(f_{\mathrm{q},k+1}) - \ldots - \mathrm{dom}(f_{\mathrm{d},k+1}) \\
&\qquad - \mathrm{dom}(f_{\mathrm{g},k+1}) - \mathrm{dom}(f_{\mathrm{c},k+1}) - \ldots - \mathrm{dom}(f_{\mathrm{a},k+1})) \wedge \\
&\qquad stp_{f_{\mathrm{Def}}}(u_{k+1}, i_{k+1}, u'_{k+1}, o_{k+1}))) \wedge \\
&\quad ((u_{k+2}, i_{k+2}) \in (\mathrm{dom}(f_{\mathrm{h},k+2}) - \mathrm{dom}(f_{\mathrm{g},k+2}) - \mathrm{dom}(f_{\mathrm{c},k+2}) - \ldots \\
&\qquad - \mathrm{dom}(f_{\mathrm{a},k+2})) \wedge stp_{f_{\mathrm{h}}}(u_{k+2}, i_{k+2}, u'_{k+2}, o_{k+2}))
\end{aligned} \tag{6.31}$$

We can see that since in (6.30), $(u_{k+1}, i_{k+1}) \in (\mathrm{dom}(f_{\mathrm{g},k+1}) - \ldots)$ guards $f_{\mathrm{g},k+1}$, and in (6.31), $(u_{k+1}, i_{k+1}) \in (\ldots - \mathrm{dom}(f_{\mathrm{g},k+1}) \ldots)$ guards $f_{\mathrm{d},k+2}$ (and similarly for the other features that contribute to the level $k+1$ values), then (6.30) and (6.31) compose to

give the empty relation, and we have a composition of output retrenchments reminiscent of the situation described in Corollary 4.6.

On the other hand, if $f_{\mathrm{h},k+2}$ overrides $f_{\mathrm{g},k+2}$ on a nonempty overlap of domains, then some of the modification captured by $C_{Op,k+1}$ earlier, will be undone by the new modification, and this will be captured by $C_{Op,k+2}$, such that in a subsequent composition, all three terms of the composed concedes relation will be valid somewhere. That this is not the same as in the previous case is not surprising since $f_{\mathrm{d},k+2} \lhd f_{\mathrm{h},k+2} \lhd f_{\mathrm{g},k+2} \lhd f_{\mathrm{c},k+2}$ is not the same as $f_{\mathrm{d},k+2} \lhd f_{\mathrm{g},k+2} \lhd f_{\mathrm{h},k+2} \lhd f_{\mathrm{c},k+2}$. However these two feature expressions have *the same shape*, so one might expect the same shape of overall concedes relation from $(\ldots f_{\mathrm{d},k+2} \lhd f_{\mathrm{c},k+2} \ldots)$ to $(\ldots f_{\mathrm{d},k+2} \lhd f_{\mathrm{h},k+2} \lhd f_{\mathrm{g},k+2} \lhd f_{\mathrm{c},k+2} \ldots)$ or $(\ldots f_{\mathrm{d},k+2} \lhd f_{\mathrm{g},k+2} \lhd f_{\mathrm{h},k+2} \lhd f_{\mathrm{c},k+2} \ldots)$ to emerge. In benign cases, appropriate manipulations of the two compositions can bring out the expected similarity.

We conclude this section by pointing out that in [Banach and Poppleton (2003)], there is a toy feature engineering case study, focused on telephone system feature interaction, and done largely along the lines of the theory above. The fact that it is very much a toy is a consequence of using a formalism similar to the one in this paper, relating a single step at one level to a single step at the next level. However such an approach has a very real drawback in that the behavioural or multistep aspects of genuine telephony applications are abstracted away. In a typical interaction with a real telephone system, one goes through a number of phases before the interaction completes, and disregarding this finer level of granularity undoubtedly undermines the credibility of any such decription. Still, the main point of [Banach and Poppleton (2003)] was to illustrate retrenchment, not to advance the state of the art in telephony. However, both that paper and this one, support the view that a development of retrenchment based ideas more accurately targeted at the needs of realistic telephone feature engineering problems would enjoy a good measure of success.

Aside from the previous point, there is a crucial difference between the theory of this paper and that of [Banach and Poppleton (2003)] since the case study there is done using primitive retrenchment rather than the output retrenchment[6] of this paper.[7] At a number of points, especially when we want to distinguish between system transitions that differ only in their outputs, the insensitivity of primitive retrenchment to this kind of situation inhibits its use in giving a fluent account of the matter. The kinds of theorem we have been able to prove in preceding sections of this paper cannot be constructed in as clean a manner without output relations. It would be an undemanding exercise to repeat the case study in [Banach and Poppleton (2003)] in the present framework, and to carry out successfully the programme discussed there, but only partially carried through.

---

6. Outputs retrenchments and primitive retrenchments, both introduced in [Banach et al. (2007)], differ in that primitive retrenchments do not have a separate output relation. Even though the two formulations are equivalent (as shown in [Banach et al. (2007)]), the convenience of having a separate output relation is considerable. The retrenchments of this paper are all (of course) output retrenchments.

7. There is another technical difference between this paper and [Banach and Poppleton (2003)]. In the latter input, output, and state spaces were assumed fixed ab initio, and large enough to accommodate all features needed at any point in the development; thus making G and P identities (there was no O of course).

7 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 compositionality, and even more to the point, associativity, we find that these properties do not hold automatically for the new formulations. The convoluted calculations of Section 5 demonstrate the lengths to which we must go to recover them. This goes to show that regarding the properties considered in this paper, associativity is 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 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. A good illustration of this occurred in 6, where with a few general assumptions, the theory we had previously developed gave a reasonable account of simple feature engineering.

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 $G_1(u'_0, u'_{1,a}) \wedge O_{Op,1}(o_0, o_{1,a}; u'_0, u'_{1,a}, \ldots) \wedge C_{Op,1}(u'_0, u'_{1,b}, o_0, o_{1,b}; \ldots)$ we could, amongst other things, infer $C_{Op,1}(u'_0, u'_{1,a}, o_0, o_{1,a}; \ldots)$. However instead of using regularity we could have assumed this implication directly as a property of the component output 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 fingrained 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 concedes relation (and in our case the output relation too), 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 to narrowly defined (or even vacuuous) 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 denying 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 sometimes 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 application of default output retrenchments to feature engineering in the preceding section, illustrates that the greater flexibility of retrenchments can give a formal account of situations such as these. 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), Jeske (2005)] to get the best of both worlds.

Having developed the theory, we confronted it with a plausible application scenario, namely the theoretical description of a fairly flexible model of feature engineering. There, we saw that given the assumptions we made about the process, certain aspects of the previously developed theoretical landscape were indeed applicable, and others were not, showing the usefulness of having a range of different results to deploy. And since, in the absence of a concrete application, it is impossible to write down specific refinement or retrenchment data, our treatment was based on default retrenchments, which enabled a generic account to be given. While default retrenchments are 'always available', and so are very useful for giving generic accounts of this kind, it is worth reiterating that the mere possibility of applying them mechanistically, is coun-

terproductive as regards possibly the main benefit of retrenchment, which is to describe non-refinement system evolutions in a manner that demands that some mathematical consistency be established between the system models involved, and human intuition about what the evolution is meant to achieve.

References

Back R. J. R. (2002); Software Construction by Stepwise Feature Introduction. *in*: Proc. ZB-02, Bert, Bowen, Henson, Robinson (eds.), LNCS **2272**, 162-183, Springer.

Back R. J. R., Sere K. (1996); Superposition Refinement of Reactive Systems. Form. Asp. Comp. **8**, 324-346.

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., 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. Sci. Comp. Prog., *submitted*.

Banach R., Poppleton M. (2003); Retrenching Partial Requirements into System Definitions: A Simple Feature Interaction Case Study. Req. Eng. J. **8**, 266-288.

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

Francez N., Forman I. R. (1990); Superimposition for Interactive Processes. *in*: Proc. CONCUR-90, Baeten, Klop (eds.), LNCS **458**, 230-245, Springer.

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

Katz S. (1993); A Superimposition Control Construct for Distributed Systems. ACM Trans. Prog. Lang. Sys. **15**, 337-356.

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

Zave P. (2001); Requirements for Evolving Systems: A Telecommunications Perspective. *in*: Proc. 5th IEEE Int. Symp. Requirements Engineering, 2-9.