Reconciling Retrenchments and Refinements

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

Abstract. The more obvious and well known drawbacks of using refinement as the sole means of progressing from an abstract model to a concrete implementation are reviewed. Retrenchment is presented in a simple partial correctness framework as a more flexible development concept for formally capturing the early otherwise preformal stages of development, and briefly justified. Given both a retrenchment of an abstract model, and a refinement of the same model, the problem of finding a model that is both a refinement of the retrenchment and a retrenchment of the refinement, is examined. A construction is given that solves the problem in a universal manner, giving the most abstract reconciliation of the two. The universality amounts to the fact that any similar reconciliation of the original retrenchment and refinement is refinable from the universal one, factoring through it.
Keywords: Refinement, Retrenchment.

1 Introduction

Retrenchment was introduced in [1] in order to overcome the drawbacks of relying on refinement alone as the only way of going from an abstract to a concrete model of a system in a completely formal manner. Subsequently the technique was developed in [2, 3, 4]. A broad reappraisal of the issues appears in [5], written with some of the wisdom of hindsight, while [6] contains further developments. Refinement (particularly in formulations that emphasise total correctness), imposes stringent constraints on the relationship that can hold between the related models, and this can restrict the applicability of the technique quite severely. This phenomenon is most keenly felt in application situations where the conception of the system starts with physical considerations, described using conventional applied mathematics, rather than the discrete systems ubiquitous in presentations of refinement. In such situations, it is commonly found that all except the last few steps of the development process have to be performed informally — because the proof obligations (POs) of refinement are so demanding that adjacent pairs of models higher up the development hierarchy are simply unable to satisfy them. Thus we lose all the benefits of full formalism, such as mechanical checkability, for the majority of the development effort.

A retrenchment step from a more abstract to a more concrete level of abstraction admits strengthening of the precondition and weakening of the postcondition, unlike refinement in its usual forward simulation variant (see [9] for a comprehensive review of refinement from both forward and backward simulation perspectives). Retrenchment also permits the mixing of state and I/O information between the levels of abstraction in question. These activities are managed by having two extra predicates per retrenched operation, the WITHIN and CONCEDES clauses. (This is what they were called in the B-Method [10, 11], within which most of the early work on retrenchment was done.) The former expresses the precondition strengthening, and the latter expresses the postcondition weakening. The key feature of retrenchment is that non-refinement-like behaviour can be accommodated within the framework via the weakened postcondition. This permits inconvenient low level detail of the true sys-

tem from interfering with an idealised model at a high level of abstraction, giving hopefully cleaner, more understandable, earlier formalizable development routes.

Since we are permitted to postpone the introduction of low level detail in retrenchment, it follows that retrenchment permits the gradual incorporation of requirements into a final specification. Thus retrenchment can be seen as providing a flexible *specification constructor*, that we can combine with other specification combination techniques to enrich the palette of techniques we have for building specifications out of smaller simpler pieces. The principal such techniques include coproduct and similar categorical constructions, parameterisation mechanisms, and refinement itself. Given the plethora of preliminary and partial models that may arise through enthusiastically embracing the capability of building system specifications piecemeal, it is important to be clear about which model of the collection actually captures all the system requirements. We can single it out and call it the contracted model.

Since retrenchment is not equivalent to any of the earlier techniques, it properly enlarges the collection of available development routes: a more concrete model that was not derivable from one or more desired abstract precursors using previously available techniques, may become derivable when retrenchment is added to the armoury of development methods. This is a positive thing as the relative value of different development routes ought to be judged on domain-specific engineering grounds rather than being hamstrung by the limitations of available specification construction mechanisms. This is particularly the case in areas of engineering where there are already well accepted development routes for the systems of interest. Formal techniques should support and strengthen these instead of advocating their disruption.

Having suggested that retrenchment ought to coexist with other specification constructors, the interplay between the members of the now enlarged specification constructor family is clearly of interest and boils down to a family of algebraic problems. The interaction with refinement is a key issue, and in [7, 8] we have explored canonical factorisations of an arbitrary retrenchment into a refinement and a 'retrenchment which preserves the level of abstraction'. In the current paper we focus on a different algebraic problem, the 'pushout'-like problem of completing a square in which a system *Abs* is refined to a system *Ref*, and also retrenched to a system *Ret*; whereupon we must find a system *Univ* such that *Univ* is simultaneously a retrenchment of *Ref* and a refinement of *Ret* (see Fig. 1). Moreover we seek to characterise the construction in a suitably universal manner. We want it to be the case that any other system *Xtra* which achieves the same reconciliation is refinable from *Univ,* in characteristic pushout-like manner.

The interest in doing the reconciliation of retrenchment and refinement is not only algebraic; it has a tangible software engineering payoff. Suppose a formal development via refinement of some software already exists, and later, or perhaps even during the latter stages of the development itself, a proposal to change the system is adopted. Let this change be expressed as an alteration of the top level model. Frequently the nature of such an alteration makes the new top level model non-derivable from its predecessor using previously available techniques. However the more flexible nature of retrenchment admits a much higher probability that the new system arises as a retrenchment of its predecessor, exploiting the ability of retrenchment to deny previously asserted properties. Assuming this to be the case, then the reconciliation of retrenchment and refinement described here becomes an important tool for helping to mechanise the reengineering of the system that is needed.

The final benefit of the reconciliation presented in this paper, is that it helps to assimilate retrenchment, a relatively recent and unfamiliar technique, into the fold of acceptable and trusted formal development tools. Reconciling retrenchment and refinement reassures practitioners that in using retrenchment, they do not risk fracturing the development process into incompatible and irreconcilable paths.

The rest of this paper is as follows. In Section 2 we review some of the reasons why refinement alone is too stringent a technique to capture all the development steps we might consider to be desirable. Section 3 introduces retrenchment in a simple partial correctness form that makes for a transparent theory. Section 4 presents I/O-filtered refinements, the kind of refinements needed later. Section 5 presents the reconciliation result and its universal property. Section 6 gives a small example. Section 7 concludes.

Notation. In the sequel we will view systems mainly from a set theoretic and relational viewpoint, which we discuss using a logical meta-notation. Thus a predicate *is* just a notation for a set etc.

2 Some Drawbacks of Refinement

In this section we recall some of the familiar drawbacks of using refinement as a sole development technique, using a running example that we pick up at various subsequent points. Inevitably, due to lack of space, the example will be too small to be convincing. We will concentrate on forward simulation in this paper since almost all applications of refinement are of this variant of the concept.

Consider a system whose state u is a set of NATs, and which has an operation *AddEl*(n) to add an element n to the set. Its description will be our abstract model (assuming a suitable syntactic framework). At a more concrete level, we will model sets of NATs by injective sequences of NATs, so $u = \{1, 2, 3\}$ corresponds to $v = <1, 2, 3>$, or $v = <2, 1, 3>$, or to any of four other possibilities. This correspondence between the two levels forms a retrieve relation relating sets and all their possible serialisations. For pragmatic reasons, we accept that the length of any representative sequence is 10 at most, while no such restriction applies to the cardinality of the set. So not all sets have concrete representations.

At the concrete level *AddEl*(n) must test the length of the sequence (and for the prior presence of n). If n is new and the length is already 10, the modelling of set union (for example by appending the extra element to the end of the sequence) is forbidden: it would break the bound on $|v|$. Whatever the concrete *AddEl*(n) operation does in this situation, we claim it cannot be a refinement of the abstract *AddEl*(n). For to be a refinement, it would have to satisfy

$$G(u, v) \wedge stp_{AddEl_C}(v, n, v') \Rightarrow (\exists\, u' \bullet stp_{AddEl_A}(u, n, u') \wedge G(u', v')) \qquad (2.1)$$

where G is the retrieve relation, $stp_{AddEl_A}(u, n, u')$ and $stp_{AddEl_C}(w, n, w')$ are the transition or step relations of the abstract and concrete *AddEl* operations, and the primes refer to the after-state of a transition.

Let us consider various alternatives for the concrete operation. If it did nothing, i.e. did a *skip*, (2.1) would fail as *skip* (a state preserving step) is not allowed under the circumstances at the abstract level. Alternatively, if the concrete operation output some error message, (2.1) would fail as a change in signature is not allowed in conventional refinement. These two proposals exhaust the relatively sensible options for

the concrete operation in this situation. Even if the concrete operation did something else within its constraints, it could not tie up with the abstract operation, which would produce a set of cardinality 11, outside the reach of the given retrieve relation.[1]

The above describes a very simple scenario in which refinement fails to adequately describe a desired development step. Many situations involving a finite computable subdomain of a mathematically ideal and infinite one, follow the same pattern, and a number of techniques have been described in the literature, designed to address the problem. One technique is Neilson's thesis [12], which describes the concept of acceptably inadequate refinements. These tackle the problem by observing that the infinite ideal domains usually arise as well behaved limits of corresponding finite ones, and thus refinement in the idealised case can be understood as the limit of a finite version (in essence an inversion of meta level quantifiers is involved). A different technique is to be found in [13, 14], where Owe proposes a logical approach, based on a careful analysis of the effects of ill-definedness on a programming logic.

A separate issue hinted at in our example is the desirability of changing I/O signatures during the course of a development step. The desire for this ranges from wanting to make the key aspects of a specification clearer with a different I/O signature, to simply wanting to incorporate change of I/O signature during refinement. The literature contains a number of works on this question, for example [15, 16, 17, 18].

The above provides just a glimpse of the literature relating to the desire to be able to do more than conventional refinement permits in a formal development step. Retrenchment, introduced below, brings all of these aspects together into a single generic and flexible development step, quantified by a pair of predicates per operation.

3 Retrenchment

We adhere to a partial correctness framework for system description. A good comparison between partial and total correctness can be found in [9]. For us therefore, a system will be given by a state space, a set of operation names each with its own I/O signature, and a transition relation for each operation. To discuss retrenchment we need two systems, an abstract one *Abs* and a concrete one *Ret*.

For *Abs*, the set of operation names is Ops_A, with typical element Op_A. The state space is U, having typical element u. For an $Op_\mathrm{A} \in \mathsf{Ops}_\mathrm{A}$, the input and output spaces will be $\mathsf{I}_{Op_\mathrm{A}}$ and $\mathsf{O}_{Op_\mathrm{A}}$ with typical elements i, o respectively (we suppress the anticipated subscripts on i and o that indicate which Op_A they belong to). Primes, indices and other decorations will be used to distinguish different elements of the same space. A typical system transition will be written u -(i, Op_A, o)-› u', where u and u' are the before- and after- states, i and o are the input and output values, and Op_A is the name of the operation responsible for the transition. The set of all such transitions makes up the transition or step relation for the operation Op_A, denoted $stp_{Op_\mathrm{A}}(u, i, u', o)$, which we will always assume to be non-empty.

At the concrete level we have a similar setup. The operation names are $Op_\mathrm{T} \in \mathsf{Ops}_\mathrm{T}$. States are $v \in \mathsf{V}$, inputs $j \in \mathsf{J}$, outputs $p \in \mathsf{P}$. Transitions are v -(j, Op_T, p)-› v', members of the step relation $stp_{Op_\mathrm{T}}(v, j, v', p)$. In retrenchment it is assumed that there is a distinct Op_T corresponding to each distinct Op_A, *but not necessarily vice versa*, so

1. A proposal for a retrieve relation that relates sets of cardinality 10 or more to serialisations of a cardinality 10 subset, works only as long as we do not introduce a *SubtractEl* operation.

the concrete level may contain additional operations. For convenience we will assume this correspondence is the inclusion $\mathsf{Ops}_A \subseteq \mathsf{Ops}_T$.

The relationship between abstract and concrete state spaces is given by the retrieve relation $H(u, v)$. We assume there are initialisation operations $Init_A$ and $Init_T$ at abstract and concrete levels that establish H in corresponding after-states such that

$$Init_T(v') \Rightarrow (\exists\, u' \bullet Init_A(u') \wedge H(u', v')) \tag{3.1}$$

Retrenchment differs from refinement in that H alone is not enough to fix the relationship between the two levels. We also have for all operation names in Ops_A, the within relation $Q_{Op}(i, j, u, v)$ and concedes relation $D_{Op}(u', v', o, p; i, j, u, v)$, where, exploiting the inclusion $\mathsf{Ops}_A \subseteq \mathsf{Ops}_T$, the A/T subscripts on Op have been suppressed because these relations are relevant to both the abstract and concrete levels. The punctuation in D_{Op} is intended to emphasise that it is mainly concerned with after-values, but may refer to the before-values if required. These relations are combined into the retrenchment operation PO for the Ops_A operations which says

$$\begin{aligned} &H(u, v) \wedge Q_{Op}(i, j, u, v) \wedge stp_{Op_T}(v, j, v', p) \Rightarrow \\ &\quad (\exists\, u', o \bullet stp_{Op_A}(u, i, u', o) \wedge (H(u', v') \vee D_{Op}(u', v', o, p; i, j, u, v))) \end{aligned} \tag{3.2}$$

This means the following. We assert the consequent of the implication, but only provided both H and Q hold. This enables us to restrict via the within relation Q, the applicability of the relationship between the abstract and concrete systems, permitting the bringing together of models that would otherwise fail to support a refinement. The consequent itself asserts that for every concrete step, there is an abstract step that either re-establishes the retrieve relation H, or failing that, satisfies the concedes relation D. Again the additional flexibility allowed by D permits us to relate models that would not otherwise be capable of being formally related.

Thus the within relation strengthens the retrieve relation in before-states, and most importantly, the concedes relation weakens the retrieve relation in after-states. Beyond the ability to restrict the relationship between abstract and concrete levels, the within relation captures any non-trivial relationship between inputs and before-states. Likewise the concedes relation captures non-refinement-like properties, and non-trivial relationships between outputs and after-states[2] (and also before-entities if appropriate).

We now reconsider our running example as a retrenchment. Predictably $\mathsf{Ops}_A = \mathsf{Ops}_T = \{Init, AddEl\}$, $\mathsf{U} = \mathbb{P}(\mathsf{NAT})$, $\mathsf{V} = iseq(\mathsf{NAT})$, $\mathsf{I}_{AddEl} = \mathsf{J}_{AddEl} = \mathsf{NAT}$, $\mathsf{O}_{AddEl} = \varnothing$, and P_{AddEl} would depend on how we defined the concrete *AddEl* operation. For the simple case in which it is *skip* for boundary steps, $\mathsf{P}_{AddEl} = \varnothing$, otherwise it contains an overflow message; we concentrate on the former. To complete the *skip* case, we need to define the within and concedes relations. We let $Q_{AddEl} = (i = j)$ and $D_{AddEl} = (|u| = 10 \wedge i \notin u \wedge u' = u \cup \{i\} \wedge rng(v') = u)$. Notice the concession relates *Abs* after-states to which a new element has been added to *Ret* states which have remained the same. These are not the only possibilities. For instance we could replace

2. Relations between outputs and states ought to hold universally, and not just when H fails. The truth is thus typically stronger than (3.2). Sharp retrenchment and output retrenchment address this, establishing in the consequent of the PO, in the former case ($(H \vee D) \wedge V$), and in the latter case ($(H \wedge O) \vee D$), where V, the nevertheless relation, or O the output relation, allow extra conjunctive properties to be expressed. See [4] and [6]. We will ignore these relations in this paper, aside from the use of a V in I/O-filtered refinements, in which concedes relations in their turn do not appear.

Q_{AddEl} with ($|u| \leq 10 \wedge i = j$) and for D_{AddEl} we could omit the $i \notin u$ clause, but retaining it is more informative.

4 I/O-Filtered Refinements

In this section we make precise the notion of refinement we need to make our subsequent results go through. Since I/O signatures can be changed during retrenchment, it is useful to absorb this capability into refinement to get a smooth result. We assume we are working with an abstract system *Abs* and a concrete system *Ref*.

The notation set up already will do for *Abs*. For *Ref*, with operation names Ops_F, where $\mathsf{Ops}_\mathrm{F} = \mathsf{Ops}_\mathrm{A}$, the state space will be W with elements w. The input and output spaces for Op_F are given by $k \in \mathsf{K}_{Op_\mathrm{F}}$ and $q \in \mathsf{Q}_{Op_\mathrm{F}}$. The abstract and concrete state spaces are related by a retrieve relation $K(u, w)$. For an I/O-filtered refinement, we further have for each $Op \in \mathsf{Ops}_\mathrm{F}$, a within relation $R_{Op}(i, k)$, and a nevertheless relation $V_{Op}(o, q)$. We also have two POs.

First there is the initialisation PO (Init PO)

$$Init_\mathrm{F}(w') \;\Rightarrow\; (\exists\, u' \bullet Init_\mathrm{A}(u') \wedge K(u', w')) \tag{4.1}$$

This is the same as (3.1). There is also the operation PO (Op PO) which for a typical Op reads:

$$\begin{aligned} &K(u, w) \wedge R_{Op}(i, k) \wedge stp_{Op_\mathrm{F}}(w, k, w', q) \;\Rightarrow \\ &\quad (\exists\, u', o \bullet stp_{Op_\mathrm{A}}(u, i, u', o) \wedge (K(u', w') \wedge V_{Op}(o, q))) \end{aligned} \tag{4.2}$$

Note that V_{Op} enters the consequent of (4.2) conjunctively (which is why we call it a refinement), in contrast to the retrenchment case.

We return to our running example and introduce a simple refinement. The states of the concrete level are pairs of disjoint sets of NATs of similar cardinality, specifically $w = (w^a, w^b)$, $w^a, w^b \subseteq \mathsf{NAT}$, $w^a \cap w^b = \varnothing$, $\|w^a| - |w^b\| \leq 2$. The retrieve relation equates $w^a \cup w^b$ with u. Let $\underline{n}$ be the binary encoding of the natural number n. Then the $AddEl_\mathrm{F}(\underline{n})$ operation adds n to whichever of w^a, w^b causes the constraints to be maintained (or either if applicable). The within relation of this refinement simply relates n and $\underline{n}$, and the nevertheless relation is trivial. With the obvious initialisation, it is evident that (4.2), suitably instantiated, holds for this arrangement.

5 The Reconciliation

In this section we take the retrenchment from *Abs* to *Ret* and the refinement from *Abs* to *Ref*, and build a fourth universal system *Univ*, that both refines *Ret* and retrenches *Ref*. See Fig. 1. We begin by defining some equivalence relations. Let

$$HD(u, v) = H(u, v) \vee \bigvee_{Op} (\exists\, \bar{o}, \bar{p}, \dot{i}, \dot{j}, \bar{u}, \bar{v} \bullet D_{Op}(u, v, \bar{o}, \bar{p}\,;\, \dot{i}, \dot{j}, \bar{u}, \bar{v})) \tag{5.1}$$

$$QI_{Op}(i, j) = (\exists\, \bar{u}, \bar{v} \bullet Q_{Op}(i, j, \bar{u}, \bar{v})) \tag{5.2}$$

$$DO_{Op}(o, p) = (\exists\, \bar{u}', \bar{v}', \dot{i}, \dot{j}, \bar{u}, \bar{v} \bullet D_{Op}(\bar{u}', \bar{v}', o, p\,;\, \dot{i}, \dot{j}, \bar{u}, \bar{v})) \tag{5.3}$$

Then

$$\sim_\mathrm{V} = ((K^\mathrm{T};HD)^\mathrm{T};(K^\mathrm{T};HD))^* \tag{5.4}$$

$$\sim_\mathrm{W} = ((HD^\mathrm{T};K)^\mathrm{T};(HD^\mathrm{T};K))^* \tag{5.5}$$

$$\sim_{\mathrm{J}_{Op}} = ((R_{Op}^\mathrm{T};QI_{Op})^\mathrm{T};(R_{Op}^\mathrm{T};QI_{Op}))^* \tag{5.6}$$

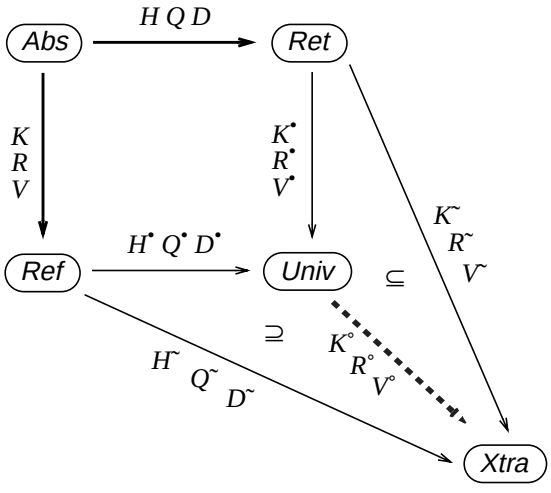

Fig. 1 All arrows labelled *H*, *Q*, *D* are retrenchments, all arrows labelled *K*, *R*, *V* are refinements.

$$\sim_{\mathrm{K}Op} = ((QI_{Op}^{\mathrm{T}};R_{Op})^{\mathrm{T}};(QI_{Op}^{\mathrm{T}};R_{Op}))^* \tag{5.7}$$

$$\sim_{\mathrm{P}Op} = ((V_{Op}^{\mathrm{T}};DO_{Op})^{\mathrm{T}};(V_{Op}^{\mathrm{T}};DO_{Op}))^* \tag{5.8}$$

$$\sim_{\mathrm{Q}Op} = ((DO_{Op}^{\mathrm{T}};V_{Op})^{\mathrm{T}};(DO_{Op}^{\mathrm{T}};V_{Op}))^* \tag{5.9}$$

The operation names set of *Univ* is Ops_U with elements Op_U. The state space is T with elements *t*, inputs are $h \in \mathsf{H}$, outputs $s \in \mathsf{S}$. These are all constructed from the systems *Ret* and *Ref* as follows. Firstly $\mathsf{Ops}_\mathrm{U} = \mathsf{Ops}_\mathrm{T} = \mathsf{Ops}_\mathrm{A} \cup (\mathsf{Ops}_\mathrm{U} - \mathsf{Ops}_\mathrm{A})$. So each Op_U is either an Op_A or an $Op_\mathrm{U} \in (\mathsf{Ops}_\mathrm{U} - \mathsf{Ops}_\mathrm{A})$. Next $\mathsf{T} = \mathsf{V}/\!\sim_\mathrm{V} \times \mathsf{W}/\!\sim_\mathrm{W}$. For $Op_\mathrm{U} \in \mathsf{Ops}_\mathrm{A}$ the input and output spaces are $\mathsf{H}_{Op} = \mathsf{J}_{Op}/\!\sim_{\mathrm{J}Op} \times \mathsf{K}_{Op}/\!\sim_{\mathrm{K}Op}$ and $\mathsf{S}_{Op} = \mathsf{P}_{Op}/\!\sim_{\mathrm{P}Op} \times \mathsf{Q}_{Op}/\!\sim_{\mathrm{Q}Op}$, while for $Op_\mathrm{U} \notin \mathsf{Ops}_\mathrm{A}$, $\mathsf{H}_{Op} = \mathsf{J}_{Op}$ and $\mathsf{S}_{Op} = \mathsf{P}_{Op}$.

Given the above we can now define the following.

$$KH(\underline{w}, [v]) \;=\; (\forall\, u \bullet K(u, \underline{w}) \Rightarrow (\exists\, \underline{v} \bullet \underline{v} \in [v] \wedge H(u, \underline{v}))) \tag{5.10}$$

$$\begin{aligned} HK([v], [w]) \;=\; & (\forall\, u, \underline{v} \bullet \underline{v} \in [v] \wedge H(u, \underline{v}) \Rightarrow \\ & (\exists\, \underline{w} \bullet \underline{w} \in [w] \wedge K(u, \underline{w}) \wedge KH(\underline{w}, [v]))) \end{aligned} \tag{5.11}$$

$$\begin{aligned} KD_{Op}(\underline{w}, [v]) \;=\; & (\forall\, u \bullet K(u, \underline{w}) \Rightarrow \\ & (\exists\, \underline{v} \bullet \underline{v} \in [v] \wedge (\exists\, \bar{o}, \bar{p}, \dot{i}, \bar{j}, \bar{u}, \bar{v} \bullet D_{Op}(u, \underline{v}, \bar{o}, \bar{p}\,;\, \dot{i}, \bar{j}, \bar{u}, \bar{v})))) \end{aligned} \tag{5.12}$$

$$\begin{aligned} DK_{Op}([v], [w]) \;=\; & \\ & (\forall\, u, \underline{v} \bullet \underline{v} \in [v] \wedge (\exists\, \bar{o}, \bar{p}, \dot{i}, \bar{j}, \bar{u}, \bar{v} \bullet D_{Op}(u, \underline{v}, \bar{o}, \bar{p}\,;\, \dot{i}, \bar{j}, \bar{u}, \bar{v})) \Rightarrow \\ & \quad (\exists\, \underline{w} \bullet \underline{w} \in [w] \wedge K(u, \underline{w}) \wedge KD_{Op}(\underline{w}, [v]))) \end{aligned} \tag{5.13}$$

$$\begin{aligned} RQ_{Op}(\underline{k}, \underline{w}, [j], [v], [w]) \;=\; & (\forall\, i, u \bullet R_{Op}(i, \underline{k}) \wedge K(u, \underline{w}) \Rightarrow \\ & (\exists\, \underline{j}, \underline{v} \bullet \underline{j} \in [j] \wedge \underline{v} \in [v] \wedge H(u, \underline{v}) \wedge Q_{Op}(i, \underline{j}, u, \underline{v}) \wedge K^\bullet(\underline{v}, ([v], [w])))) \end{aligned} \tag{5.14}$$

$$QR_{Op}([j], [k]) =$$
$$(\forall\ i, u, \underline{j}, \underline{v}, v, w \bullet \underline{j} \in [j] \wedge H(u, \underline{v}) \wedge Q_{Op}(i, \underline{j}, u, \underline{v}) \wedge K^{\bullet}(\underline{v}, ([v], [w])) \Rightarrow$$
$$(\exists\ \underline{k}, \underline{w} \bullet \underline{k} \in [k] \wedge \underline{w} \in [w] \wedge R_{Op}(i, \underline{k}) \wedge K(u, \underline{w}) \wedge$$
$$RQ_{Op}(\underline{k}, \underline{w}, [j], [v], [w])))) \tag{5.15}$$

$$VD_{Op}(\underline{w}', \underline{q}, \underline{k}, \underline{w}, [p], [v'], [w'], [j], [k], [v], [w]) =$$
$$(\forall\ u', o, i, u \bullet K(u', \underline{w}') \wedge V_{Op}(o, \underline{q}) \wedge R_{Op}(i, \underline{k}) \wedge K(u, \underline{w}) \Rightarrow$$
$$(\exists\ \underline{v}', \underline{p}, \underline{j}, \underline{v} \bullet \underline{v}' \in [v'] \wedge \underline{p} \in [p] \wedge \underline{j} \in [j] \wedge \underline{v} \in [v] \wedge$$
$$H(u, \underline{v}) \wedge Q_{Op}(i, \underline{j}, u, \underline{v}) \wedge D_{Op}(u', \underline{v}', o, \underline{p}; i, \underline{j}, u, \underline{v}) \wedge$$
$$K^{\bullet}(\underline{v}', ([v'], [w'])) \wedge R^{\bullet}{}_{Op}(\underline{j}, ([j], [k])) \wedge K^{\bullet}(\underline{v}, ([v], [w])))) \tag{5.16}$$

$$DV_{Op}([p], [q]) =$$
$$(\forall\ u', o, i, u, \underline{v}', \underline{p}, \underline{j}, \underline{v}, v', w', j, k, v, w \bullet$$
$$\underline{p} \in [p] \wedge H(u, \underline{v}) \wedge Q_{Op}(i, \underline{j}, u, \underline{v}) \wedge D_{Op}(u', \underline{v}', o, \underline{p}; i, \underline{j}, u, \underline{v}) \wedge$$
$$K^{\bullet}(\underline{v}', ([v'], [w'])) \wedge R^{\bullet}{}_{Op}(\underline{j}, ([j], [k])) \wedge K^{\bullet}(\underline{v}, ([v], [w])) \Rightarrow$$
$$(\exists\ \underline{w}', \underline{q}, \underline{k}, \underline{w} \bullet \underline{w}' \in [w'] \wedge \underline{q} \in [q] \wedge \underline{k} \in [k] \wedge \underline{w} \in [w] \wedge$$
$$K(u', \underline{w}') \wedge V_{Op}(o, \underline{q}) \wedge R_{Op}(i, \underline{k}) \wedge K(u, \underline{w}) \wedge$$
$$VD_{Op}(\underline{w}', \underline{q}, \underline{k}, \underline{w}, [p], [v'], [w'], [j], [k], [v], [w]))) \tag{5.17}$$

We turn our attention to the retrieve and other relations of the retrenchment and refinement we are going to construct; see Fig. 1 again. Let

$$K^{\bullet}(\underline{v}, ([v], [w])) \ = \ \underline{v} \in [v] \wedge HK([v], [w]) \wedge \bigwedge_{Op} DK_{Op}([v], [w]) \tag{5.18}$$

$$H^{\bullet}(\underline{w}, ([v], [w])) \ = \ \underline{w} \in [w] \wedge (\exists\ u \bullet K(u, \underline{w})) \wedge KH(\underline{w}, [v]) \wedge$$
$$HK([v], [w]) \wedge \bigwedge_{Op} DK_{Op}([v], [w]) \tag{5.19}$$

$$R^{\bullet}{}_{Op}(\underline{j}, ([j], [k])) \ = \ \underline{j} \in [j] \wedge QR_{Op}([j], [k]) \tag{5.20}$$

$$Q^{\bullet}{}_{Op}(\underline{k}, ([j], [k]), \underline{w}, ([v], [w])) \ = \ \underline{k} \in [k] \wedge \underline{w} \in [w] \wedge (\exists\ i, u \bullet R_{Op}(i, \underline{k}) \wedge$$
$$K(u, \underline{w})) \wedge RQ_{Op}(\underline{k}, \underline{w}, [j], [v], [w]) \wedge QR_{Op}([j], [k]) \tag{5.21}$$

$$V^{\bullet}{}_{Op}(\underline{p}, ([p], [q])) \ = \ \underline{p} \in [p] \wedge DV_{Op}([p], [q]) \tag{5.22}$$

$$D^{\bullet}{}_{Op}(\underline{w}', ([v'], [w']), \underline{q}, ([p], [q]); \underline{k}, ([j], [k]), \underline{w}, ([v], [w])) \ =$$
$$\underline{w}' \in [w'] \wedge \underline{q} \in [q] \wedge \underline{k} \in [k] \wedge \underline{w} \in [w] \wedge$$
$$(\exists\ u', o, i, u \bullet K(u', \underline{w}') \wedge V_{Op}(o, \underline{q}) \wedge R_{Op}(i, \underline{k}) \wedge K(u, \underline{w})) \wedge$$
$$VD_{Op}(\underline{w}', \underline{q}, \underline{k}, \underline{w}, [p], [v'], [w'], [j], [k], [v], [w]) \wedge DV_{Op}([p], [q]) \tag{5.23}$$

In the above, Op ranges over $\mathsf{Ops}_{\mathrm{A}}$. For operations not in $\mathsf{Ops}_{\mathrm{A}}$, we define $R^{\bullet}{}_{Op}$ and $V^{\bullet}{}_{Op}$ as identities on inputs and outputs respectively.

Given the above, Fig. 1 commutes in the following sense. Firstly $H(u, \underline{v});K^{\bullet}(\underline{v}, ([v], [w])) = K(u, \underline{w});H^{\bullet}(\underline{w}, ([v], [w])) = G(u, ([v], [w]))$. We write this for short as

$$H;K^{\bullet} = K;H^{\bullet} \equiv G \tag{5.24}$$

Secondly $(H(u, \underline{v}) \wedge Q_{Op}(i, \underline{j}, u, \underline{v}));(R^{\bullet}{}_{Op}(\underline{j}, ([j], [k])) \wedge K^{\bullet}(\underline{v}, ([v], [w]))) = (R_{Op}(i, \underline{k}) \wedge K(u, \underline{w}));Q^{\bullet}{}_{Op}(\underline{k}, ([j], [k]), \underline{w}, ([v], [w]))$, or more briefly

$$(H \wedge Q);(R^{\bullet} \wedge K^{\bullet}) = (R \wedge K);Q^{\bullet} \equiv P_{Op} \tag{5.25}$$

Thirdly, quoting the shorter form only, and using a prime to indicate that after-states are being referred to, we have

$$(H \wedge Q \wedge D);(K^{\bullet\prime} \wedge V^{\bullet} \wedge R^{\bullet} \wedge K^{\bullet}) = (K' \wedge V \wedge R \wedge K);D^{\bullet} \equiv C_{Op} \tag{5.26}$$

Proof. We prove (5.24) and (5.25). The proof for (5.26) is similar. Take (5.24) first and assume only one *Op* for simplicity. Expanding $H(u, \underline{v});K^\bullet(\underline{v}, ([v], [w]))$ gives

$$
\begin{aligned}
&(\exists\, \underline{v} \bullet H(u, \underline{v}) \wedge K^\bullet(\underline{v}, ([v], [w])))\\
&= (\exists\, \underline{v} \bullet H(u, \underline{v}) \wedge \underline{v} \in [v] \wedge HK([v], [w]) \wedge DK_{Op}([v], [w]))\\
&= (\exists\, \underline{v} \bullet H(u, \underline{v}) \wedge \underline{v} \in [v] \wedge DK_{Op}([v], [w]) \wedge\\
&\quad (\forall\, u, \underline{v} \bullet \underline{v} \in [v] \wedge H(u, \underline{v}) \Rightarrow (\exists\, \underline{w} \bullet \underline{w} \in [w] \wedge K(u, \underline{w}) \wedge KH(\underline{w}, [v]))))\\
&= (\exists\, \underline{v} \bullet H(u, \underline{v}) \wedge \underline{v} \in [v] \wedge DK_{Op}([v], [w]) \wedge\\
&\quad (\underline{v} \in [v] \wedge H(u, \underline{v}) \Rightarrow (\exists\, \underline{w} \bullet \underline{w} \in [w] \wedge K(u, \underline{w}) \wedge KH(\underline{w}, [v]))) \wedge\\
&\quad (\forall\, u, \underline{v} \bullet \underline{v} \in [v] \wedge H(u, \underline{v}) \Rightarrow (\exists\, \underline{w} \bullet \underline{w} \in [w] \wedge K(u, \underline{w}) \wedge KH(\underline{w}, [v]))))\\
&= (\exists\, \underline{v} \bullet H(u, \underline{v}) \wedge \underline{v} \in [v] \wedge DK_{Op}([v], [w]) \wedge\\
&\quad (\exists\, \underline{w} \bullet \underline{w} \in [w] \wedge K(u, \underline{w}) \wedge KH(\underline{w}, [v])) \wedge\\
&\quad (\forall\, u, \underline{v} \bullet \underline{v} \in [v] \wedge H(u, \underline{v}) \Rightarrow (\exists\, \underline{w} \bullet \underline{w} \in [w] \wedge K(u, \underline{w}) \wedge KH(\underline{w}, [v]))))\\
&= (\exists\, \underline{v} \bullet H(u, \underline{v}) \wedge \underline{v} \in [v] \wedge DK_{Op}([v], [w]) \wedge\\
&\quad (\exists\, \underline{w} \bullet \underline{w} \in [w] \wedge K(u, \underline{w}) \wedge KH(\underline{w}, [v])) \wedge HK([v], [w]))\\
&= (\exists\, \underline{v}, \underline{w} \bullet H(u, \underline{v}) \wedge \underline{v} \in [v] \wedge DK_{Op}([v], [w]) \wedge \underline{w} \in [w] \wedge K(u, \underline{w}) \wedge\\
&\quad KH(\underline{w}, [v]) \wedge HK([v], [w]))
\end{aligned}
$$

Expanding $K(u, \underline{w});H^\bullet(\underline{w}, ([v], [w]))$ gives

$$
\begin{aligned}
&(\exists\, \underline{w} \bullet K(u, \underline{w}) \wedge H^\bullet(\underline{w}, ([v], [w])))\\
&= (\exists\, \underline{w} \bullet K(u, \underline{w}) \wedge \underline{w} \in [w] \wedge (\exists\, u \bullet K(u, \underline{w})) \wedge KH(\underline{w}, [v]) \wedge HK([v], [w]) \wedge\\
&\quad DK_{Op}([v], [w]))\\
&= (\exists\, \underline{w} \bullet K(u, \underline{w}) \wedge \underline{w} \in [w] \wedge KH(\underline{w}, [v]) \wedge HK([v], [w]) \wedge DK_{Op}([v], [w]))\\
&= (\exists\, \underline{w} \bullet K(u, \underline{w}) \wedge \underline{w} \in [w] \wedge HK([v], [w]) \wedge DK_{Op}([v], [w]) \wedge\\
&\quad (\forall\, u \bullet K(u, \underline{w}) \Rightarrow (\exists\, \underline{v} \bullet \underline{v} \in [v] \wedge H(u, \underline{v}))))\\
&= (\exists\, \underline{w} \bullet K(u, \underline{w}) \wedge \underline{w} \in [w] \wedge HK([v], [w]) \wedge DK_{Op}([v], [w]) \wedge\\
&\quad (\exists\, \underline{v} \bullet \underline{v} \in [v] \wedge H(u, \underline{v})) \wedge KH(\underline{w}, [v]))\\
&= (\exists\, \underline{w}, \underline{v} \bullet K(u, \underline{w}) \wedge \underline{w} \in [w] \wedge HK([v], [w]) \wedge DK_{Op}([v], [w]) \wedge \underline{v} \in [v] \wedge\\
&\quad H(u, \underline{v}) \wedge KH(\underline{w}, [v]))
\end{aligned}
$$

Thus $H;K^\bullet = K;H^\bullet$. Now take (5.25). First consider $(H \wedge Q);(R^\bullet \wedge K^\bullet)$.

$$
\begin{aligned}
&(H(u, \underline{v}) \wedge Q_{Op}(i, \underline{j}, u, \underline{v}));(R^\bullet{}_{Op}(\underline{j}, ([j], [k])) \wedge K^\bullet(\underline{v}, ([v], [w])))\\
&= (\exists\, \underline{j}, \underline{v} \bullet H(u, \underline{v}) \wedge Q_{Op}(i, \underline{j}, u, \underline{v}) \wedge R^\bullet{}_{Op}(\underline{j}, ([j], [k])) \wedge K^\bullet(\underline{v}, ([v], [w])))\\
&= (\exists\, \underline{j}, \underline{v} \bullet H(u, \underline{v}) \wedge Q_{Op}(i, \underline{j}, u, \underline{v}) \wedge R^\bullet{}_{Op}(\underline{j}, ([j], [k])) \wedge \underline{v} \in [v] \wedge K^\bullet(\underline{v}, ([v], [w])))\\
&= (\exists\, \underline{j}, \underline{v} \bullet H(u, \underline{v}) \wedge Q_{Op}(i, \underline{j}, u, \underline{v}) \wedge \underline{v} \in [v] \wedge K^\bullet(\underline{v}, ([v], [w])) \wedge \underline{j} \in [j] \wedge\\
&\quad QR_{Op}([j], [k]))\\
&= (\exists\, \underline{j}, \underline{v} \bullet H(u, \underline{v}) \wedge Q_{Op}(i, \underline{j}, u, \underline{v}) \wedge \underline{v} \in [v] \wedge K^\bullet(\underline{v}, ([v], [w])) \wedge \underline{j} \in [j] \wedge\\
&\quad (\forall\, i, u, \underline{j}, \underline{v}, v, w \bullet \underline{j} \in [j] \wedge H(u, \underline{v}) \wedge Q_{Op}(i, \underline{j}, u, \underline{v}) \wedge K^\bullet(\underline{v}, ([v], [w])) \Rightarrow\\
&\qquad (\exists\, \underline{k}, \underline{w} \bullet \underline{k} \in [k] \wedge \underline{w} \in [w] \wedge R_{Op}(i, \underline{k}) \wedge K(u, \underline{w}) \wedge\\
&\qquad RQ_{Op}(\underline{k}, \underline{w}, [j], [v], [w]))))\\
&= (\exists\, \underline{j}, \underline{v} \bullet H(u, \underline{v}) \wedge Q_{Op}(i, \underline{j}, u, \underline{v}) \wedge \underline{v} \in [v] \wedge K^\bullet(\underline{v}, ([v], [w])) \wedge \underline{j} \in [j] \wedge\\
&\quad (\exists\, \underline{k}, \underline{w} \bullet \underline{k} \in [k] \wedge \underline{w} \in [w] \wedge R_{Op}(i, \underline{k}) \wedge K(u, \underline{w}) \wedge RQ_{Op}(\underline{k}, \underline{w}, [j], [v], [w])) \wedge\\
&\quad QR_{Op}([j], [k]))\\
&= (\exists\, \underline{j}, \underline{v}, \underline{k}, \underline{w} \bullet H(u, \underline{v}) \wedge Q_{Op}(i, \underline{j}, u, \underline{v}) \wedge \underline{v} \in [v] \wedge K^\bullet(\underline{v}, ([v], [w])) \wedge \underline{j} \in [j] \wedge\\
&\quad \underline{k} \in [k] \wedge \underline{w} \in [w] \wedge R_{Op}(i, \underline{k}) \wedge K(u, \underline{w}) \wedge RQ_{Op}(\underline{k}, \underline{w}, [j], [v], [w]) \wedge\\
&\quad QR_{Op}([j], [k]))
\end{aligned}
$$

The clause $(R_{Op}(i, \underline{k}) \wedge K(u, \underline{w}));Q^\bullet{}_{Op}(\underline{k}, ([j], [k]), \underline{w}, ([v], [w]))$ equals

$$
\begin{aligned}
&(\exists\, \underline{k}, \underline{w} \bullet R_{Op}(i, \underline{k}) \wedge K(u, \underline{w}) \wedge \underline{k} \in [k] \wedge \underline{w} \in [w] \wedge (\exists\, i, u \bullet R_{Op}(i, \underline{k}) \wedge K(u, \underline{w})) \wedge \\
&\quad RQ_{Op}(\underline{k}, \underline{w}, [j], [v], [w]) \wedge QR_{Op}([j], [k])))\\
&= (\exists\, \underline{k}, \underline{w} \bullet R_{Op}(i, \underline{k}) \wedge K(u, \underline{w}) \wedge \underline{k} \in [k] \wedge \underline{w} \in [w] \wedge QR_{Op}([j], [k])) \wedge \\
&\quad (\forall\, i, u \bullet R_{Op}(i, \underline{k}) \wedge K(u, \underline{w}) \Rightarrow \\
&\qquad (\exists\, \underline{j}, \underline{v} \bullet \underline{j} \in [j] \wedge \underline{v} \in [v] \wedge H(u, \underline{v}) \wedge Q_{Op}(i, \underline{j}, u, \underline{v}) \wedge K^{\bullet}(\underline{v}, ([v], [w])))))\\
&= (\exists\, \underline{k}, \underline{w} \bullet R_{Op}(i, \underline{k}) \wedge K(u, \underline{w}) \wedge \underline{k} \in [k] \wedge \underline{w} \in [w] \wedge QR_{Op}([j], [k])) \wedge \\
&\quad (\exists\, \underline{j}, \underline{v} \bullet \underline{j} \in [j] \wedge \underline{v} \in [v] \wedge H(u, \underline{v}) \wedge Q_{Op}(i, \underline{j}, u, \underline{v}) \wedge K^{\bullet}(\underline{v}, ([v], [w]))) \wedge \\
&\quad RQ_{Op}(\underline{k}, \underline{w}, [j], [v], [w]))\\
&= (\exists\, \underline{k}, \underline{w}, \underline{j}, \underline{v} \bullet R_{Op}(i, \underline{k}) \wedge K(u, \underline{w}) \wedge \underline{k} \in [k] \wedge \underline{w} \in [w] \wedge QR_{Op}([j], [k])) \wedge \\
&\quad \underline{j} \in [j] \wedge \underline{v} \in [v] \wedge H(u, \underline{v}) \wedge Q_{Op}(i, \underline{j}, u, \underline{v}) \wedge K^{\bullet}(\underline{v}, ([v], [w])) \wedge \\
&\quad RQ_{Op}(\underline{k}, \underline{w}, [j], [v], [w])))
\end{aligned}
$$

So $(H \wedge Q);(R^{\bullet} \wedge K^{\bullet}) = (R \wedge K);Q^{\bullet}$. We are done.

Now we can give the transitions of *Univ*, firstly remembering that the operation names set Ops_U decomposes as $\mathsf{Ops}_U = \mathsf{Ops}_A \cup (\mathsf{Ops}_U - \mathsf{Ops}_A)$.

For $Op_U \in (\mathsf{Ops}_U - \mathsf{Ops}_A)$ we have a transition t -(h, Op_U, s)-› t' or more explicitly

$$([v], [w])\ \text{-}(\underline{j}, Op_U, \underline{p})\text{-›}\ ([v'], [w']) \tag{5.27}$$

iff $[v]$, $[w]$, $\underline{j}$, $\underline{p}$, $[v']$, $[w']$ satisfy

$$\begin{aligned}
&\mathbf{(}\forall\, \underline{v} \bullet \underline{v} \in [v] \Rightarrow\\
&\quad (\exists\, \underline{v}' \bullet stp_{Op_T}(\underline{v}, \underline{j}, \underline{v}', \underline{p}) \wedge K^{\bullet}(\underline{v}', ([v'], [w'])))\mathbf{)}
\end{aligned} \tag{5.28}$$

For $Op_A \in \mathsf{Ops}_A$, we have t -(h, Op_A, s)-› t' or

$$([v], [w])\ \text{-}(([j], [k]), Op_A, ([p], [q]))\text{-›}\ ([v'], [w']) \tag{5.29}$$

iff $[v]$, $[w]$, $[j]$, $[k]$, $[p]$, $[q]$, $[v']$, $[w']$ satisfy

$$\begin{aligned}
&\mathbf{(}\forall\, \underline{v}, \underline{j} \bullet \underline{v} \in [v] \wedge \underline{j} \in [j] \Rightarrow (\exists\, \underline{v}', \underline{p} \bullet stp_{Op_T}(\underline{v}, \underline{j}, \underline{v}', \underline{p}) \wedge\\
&\quad K^{\bullet}(\underline{v}', ([v'], [w'])) \wedge V^{\bullet}{}_{Op}(\underline{p}, ([p], [q])))\mathbf{)} \qquad \text{(a)}\\
&\wedge\\
&\mathbf{(}\forall\, \underline{w}, \underline{k} \bullet H^{\bullet}(\underline{w}, ([v], [w])) \wedge Q^{\bullet}{}_{Op}(\underline{k}, ([j], [k]), \underline{w}, ([v], [w])) \Rightarrow\\
&\quad (\exists\, \underline{w}', \underline{k} \bullet stp_{Op_F}(\underline{w}, \underline{k}, \underline{w}', \underline{q}) \wedge (H^{\bullet}(\underline{w}', ([v'], [w'])) \vee\\
&\qquad D^{\bullet}{}_{Op}(\underline{w}', ([v'], [w']), \underline{q}, ([p], [q]); \underline{k}, ([j], [k]), \underline{w}, ([v], [w])))))\mathbf{)} \quad \text{(b)}
\end{aligned} \tag{5.30}$$

Note that the implication in (5.30b) gives rise to junk transitions when (b) holds trivially. Part (a) can never hold trivially since v is always in $[v]$ and j is always in $[j]$.

Finally, $Init_U(t')$ sets t' to any $([v'], [w'])$ such that the following is true.

$$\begin{aligned}
&(\exists\, \underline{v}' \bullet Init_T(\underline{v}') \wedge K^{\bullet}(\underline{v}', ([v'], [w'])))\ \ \wedge\\
&(\exists\, \underline{w}' \bullet Init_F(\underline{w}') \wedge H^{\bullet}(\underline{w}', ([v'], [w'])))
\end{aligned} \tag{5.31}$$

We now establish that the components introduced define a retrenchment from *Ref* to *Univ* and an I/O-filtered refinement from *Ret* to *Univ*, by showing that the appropriate POs are satisfied.

Take *Ret* to *Univ* first. The Init PO has the form

$$Init_U(t') \ \Rightarrow\ (\exists\, \underline{v}' \bullet Init_T(\underline{v}') \wedge K^{\bullet}(\underline{v}', t')) \tag{5.32}$$

Assume $Init_U(t')$ with $t' = ([v'], [w'])$. Then by (5.31) the consequent is immediate.

Now consider the Op PO

$$K^\bullet(\underline{v}, t) \wedge R^\bullet_{Op}(\underline{j}, h) \wedge stp_{Op_U}(t, h, t', s) \Rightarrow$$
$$(\exists\, \underline{v}', \underline{p} \bullet stp_{Op_T}(\underline{v}, \underline{j}, \underline{v}', \underline{p}) \wedge K^\bullet(\underline{v}', t') \wedge V^\bullet_{Op}(\underline{p}, s)) \quad (5.33)$$

Since Ops_U decomposes as $\mathsf{Ops}_A \cup (\mathsf{Ops}_U - \mathsf{Ops}_A)$, there are two cases to consider. Case $Op_U \in \mathsf{Ops}_A$. Assume the antecedents with $t = ([v], [w])$, $h = ([j], [k])$, $t' = ([v'], [w'])$ and $s = ([p], [q])$. Then $K^\bullet$ together with (5.18) says $\underline{v} \in [v]$, $R^\bullet_{Op}$ together with (5.20) says $\underline{j} \in [j]$. The consequent now follows from (5.30a). Case $Op_U \notin \mathsf{Ops}_A$. Assume the antecedents with $t = ([v], [w])$, $h = \underline{j}$, $t' = ([v'], [w'])$ and $s = \underline{p}$. As before $\underline{v} \in [v]$. So from $stp_{Op_U}(t, h, t', s)$, by (5.28), $stp_{Op_T}(\underline{v}, \underline{j}, \underline{v}', \underline{p})$ and $K^\bullet(\underline{v}', t')$ hold, and since $s = \underline{p}$, $V^\bullet_{Op}(\underline{p}, s)$ also holds. We are done.

Now for *Ref* to *Univ*. The Init PO for this retrenchment reads

$$Init_U(t') \Rightarrow (\exists\, \underline{w}' \bullet Init_F(\underline{w}') \wedge H^\bullet(\underline{w}', t')) \quad (5.34)$$

Assume $Init_U(t')$ with $t' = ([v'], [w'])$. Then by (5.31) the consequent is immediate.

Turning to the Op PO we have to demonstrate that

$$H^\bullet(\underline{w}, t) \wedge Q^\bullet_{Op}(\underline{k}, h, \underline{w}, t) \wedge stp_{Op_U}(t, h, t', s) \Rightarrow$$
$$(\exists\, \underline{w}', \underline{q} \bullet stp_{Op_F}(\underline{w}, \underline{k}, \underline{w}', \underline{q}) \wedge$$
$$(H^\bullet(\underline{w}', t') \vee D^\bullet_{Op}(\underline{w}', t', \underline{q}, s; \underline{k}, h, \underline{w}, t))) \quad (5.35)$$

For this relationship Op_U only ranges over Ops_A. Choose values $t = ([v], [w])$, $h = ([j], [k])$, $t' = ([v'], [w'])$ and $s = ([p], [q])$ for which the antecedent holds. Then the consequent follows immediately from (5.30b). Done.

The final piece of the construction is to show that either the composition of the *Abs* to *Ret* retrenchment and the *Ret* to *Univ* refinement on the one hand, or the *Abs* to *Ref* refinement and the *Ref* to *Univ* retrenchment on the other, do indeed yield a retrenchment from *Abs* to *Univ*. For if so then (5.24)-(5.26) show that they both give the *same* retrenchment, with retrieves, within, and concedes relations given respectively by G, P_{Op}, C_{Op}.

The Init PO demands that

$$Init_U(t') \Rightarrow (\exists\, u' \bullet Init_A(u') \wedge G(u', t')) \quad (5.36)$$

Assume $Init_U(t')$ with $t' = ([v'], [w'])$. Then by (5.31) there is a value, $\underline{w}'$ say, for which $Init_F(\underline{w}')$ and $H^\bullet(\underline{w}', t')$ hold. Given $Init_F(\underline{w}')$, we can use (4.1) to get $Init_A(u')$, which we require, and also $K(u', \underline{w}')$. Finally, since we now have $K(u', \underline{w}')$ and $H^\bullet(\underline{w}', t')$, then by composition we also have $G(u', t')$. We are done.

Next, consider the Op PO

$$G(u, t) \wedge P_{Op}(i, h, u, t) \wedge stp_{Op_U}(t, h, t', s) \Rightarrow$$
$$(\exists\, u', o \bullet stp_{Op_A}(u, i, u', o) \wedge (G(u', t') \vee C_{Op}(u', t', o, s; i, h, u, t))) \quad (5.37)$$

For this Op_U only ranges over Ops_A. Assume the antecedents with $t = ([v], [w])$, $h = ([j], [k])$, $t' = ([v'], [w'])$ and $s = ([p], [q])$. Now $P_{Op}(i, ([j], [k]), u, ([v], [w])) = (H(u, \underline{v}) \wedge Q_{Op}(i, \underline{j}, u, \underline{v}));(R^\bullet_{Op}(\underline{j}, ([j], [k])) \wedge K^\bullet(\underline{v}, ([v], [w])))$. Thus we have $K^\bullet(\underline{v}, ([v], [w]))$, $R^\bullet_{Op}(\underline{j}, ([j], [k]))$ and $stp_{Op_U}(t, h, t', s)$. We can therefore use (5.33) to derive $stp_{Op_T}(\underline{v}, \underline{j}, \underline{v}', \underline{p})$, $K^\bullet(\underline{v}', t')$ and $V^\bullet_{Op}(\underline{p}, s)$. Next, $H(u, \underline{v})$, $Q_{Op}(i, \underline{j}, u, \underline{v})$ and $stp_{Op_T}(\underline{v}, \underline{j}, \underline{v}', \underline{p})$ make up the antecedent of (3.2), so $stp_{Op_A}(u, i, u', o)$ and $H(u', \underline{v}') \vee D_{Op}(u', \underline{v}', o, \underline{p}; i, \underline{j}, u, \underline{v})$ are also true. So we have shown that stp_{Op_A} holds. All we need is $G' \vee C_{Op}$, which we derive from $H' \vee D_{Op}$ as follows. Assume $H(u', \underline{v}')$. Then as $K^\bullet(\underline{v}', t')$ holds, we have $H(u', \underline{v}');K^\bullet(\underline{v}', t')$ and thus $G(u', t')$. Now assume $D_{Op}(u', \underline{v}',$

$o, \underline{p}; i, \underline{j}, u, \underline{v}$). As H, Q_{Op}, $K^{\bullet\prime}$, $V^{\bullet}_{Op}$, $R^{\bullet}_{Op}$ and $K^{\bullet}$ all hold, we have ($H(u, \underline{v}) \wedge Q_{Op}(i, \underline{j}, u, \underline{v}) \wedge D_{Op}(u', \underline{v}', o, \underline{p}; i, \underline{j}, u, \underline{v})$);($K^{\bullet}(\underline{v}', t') \wedge V^{\bullet}_{Op}(\underline{p}, s) \wedge R^{\bullet}_{Op}(\underline{j}, h) \wedge K^{\bullet}(\underline{v}, t)$) and thus $C_{Op}(u', t', o, s; i, h, u, t)$. Hence $G' \vee C_{Op}$ holds and we are done.

Theorem 5.1 Let there be a retrenchment from *Abs* to *Ret*, and a refinement from *Abs* to *Ref*. Then (see Fig. 1)

(1) There is a universal system *Univ* such that there is a retrenchment from *Ref* to *Univ* and an I/O-filtered refinement from *Ret* to *Univ* whose compositions with the original refinement and retrenchment respectively are equal as retrenchments from *Abs* to *Univ*, and which satisfies (U1)-(U10) below.

(2) Whenever there is a system *Xtra* and a retrenchment from *Ref* to *Xtra* and an I/O-filtered refinement from *Ret* to *Xtra* whose compositions with the original refinement and retrenchment respectively are equal as retrenchments from *Abs* to *Xtra*, and which satisfies (X1)-(X10) below, then there is an I/O-filtered refinement from *Univ* to *Xtra* such that $H^{\bullet};K^{\circ} \Rightarrow H^{\sim}$, $(H^{\bullet} \wedge Q^{\bullet});(R^{\circ} \wedge K^{\circ}) \Rightarrow Q^{\sim}$, $(H^{\bullet} \wedge Q^{\bullet} \wedge D^{\bullet});(K^{\circ\prime} \wedge V^{\circ} \wedge R^{\circ} \wedge K^{\circ}) \Rightarrow D^{\sim}$, and such that $K^{\bullet};K^{\circ} \Rightarrow K^{\sim}$, $R^{\bullet};R^{\circ} \Rightarrow R^{\sim}$, $V^{\bullet};V^{\circ} \Rightarrow V^{\sim}$.

(3) Whenever a system *Univ** has properties (1) and (2) above of *Univ*, then *Univ* and *Univ** are mutually I/O-filtered interrefinable.

Proof. We have proved part (1) of Theorem 5.1 already except for properties (U1) to (U10) which we postpone for just a moment. Now suppose there is an I/O-filtered refinement from *Ret* to *Xtra* with retrieve relation $K^{\sim}$, within relations $R^{\sim}_{Op}$, and nevertheless relations $V^{\sim}_{Op}$; and a retrenchment from *Ref* to *Xtra* with retrieve relation $H^{\sim}$, within relations $Q^{\sim}_{Op}$, and concedes relations $D^{\sim}_{Op}$, and such that the stated compositions hold. Let the state, input and output spaces of *Xtra* be given by $t^{\sim} \in \mathsf{T}^{\sim}$, $h^{\sim} \in \mathsf{H}^{\sim}_{Op}$, $s^{\sim} \in \mathsf{S}^{\sim}_{Op}$. Let the initialisation and step predicates of *Xtra* be $Init_X$ and $st\text{-}p_{Op_X}$. Suppose also that (X1) to (X10) below hold.

$$K^{\sim}(v', t^{\sim\prime}) \wedge K^{\sim}(\underline{v}', t^{\sim\prime}) \Rightarrow v' \sim \underline{v}' \tag{X1}$$

$$(H^{\sim}(w', t^{\sim\prime}) \vee D^{\sim}_{Op}(w', t^{\sim\prime}, ...)) \wedge (H^{\sim}(\underline{w}', t^{\sim\prime}) \vee D^{\sim}_{Op}(\underline{w}', t^{\sim\prime}, ...)) \Rightarrow w' \sim \underline{w}' \tag{X2}$$

$$V^{\sim}_{Op}(p, s^{\sim}) \wedge V^{\sim}_{Op}(\underline{p}, s^{\sim}) \Rightarrow p \sim \underline{p} \tag{X3}$$

$$D^{\sim}_{Op}(w', t^{\sim\prime}, q, s^{\sim}; k, h^{\sim}, w, t^{\sim}) \wedge D^{\sim}_{Op}(\underline{w}', t^{\sim\prime}, \underline{q}, s^{\sim}; \underline{k}, h^{\sim}, \underline{w}, t^{\sim}) \Rightarrow q \sim \underline{q} \tag{X4}$$

$$K^{\sim}(v', t^{\sim\prime}) \wedge \underline{v}' \in [v'] \Rightarrow K^{\sim}(\underline{v}', t^{\sim\prime}) \tag{X5}$$

$$V^{\sim}_{Op}(p, s^{\sim}) \wedge \underline{p} \in [p] \Rightarrow V^{\sim}_{Op}(\underline{p}, s^{\sim}) \tag{X6}$$

$$K^{\sim}(v', t^{\sim\prime}) \Rightarrow (H^{\sim}(w', t^{\sim\prime}) \Leftrightarrow H^{\bullet}(w', ([v'], [w']))) \tag{X7}$$

$$\begin{aligned} &K^{\sim}(v', t^{\sim\prime}) \wedge V^{\sim}_{Op}(p, s^{\sim}) \wedge R^{\sim}_{Op}(j, h^{\sim}) \wedge K^{\sim}(v, t^{\sim}) \Rightarrow \\ &\quad (D^{\sim}_{Op}(w', t^{\sim\prime}, q, s^{\sim}; k, h^{\sim}, w, t^{\sim}) \Leftrightarrow \\ &\qquad D^{\bullet}_{Op}(w', ([v'], [w']), q, ([p], [q]); k, ([j], [k]), w, ([v], [w]))) \end{aligned} \tag{X8}$$

$$K^{\sim}(v', t^{\sim\prime}) \Rightarrow (\exists\, w' \bullet K^{\bullet}(v', ([v'], [w']))) \tag{X9}$$

$$V^{\sim}_{Op}(p, s^{\sim}) \Rightarrow (\exists\, q \bullet V^{\bullet}_{Op}(p, ([p], [q]))) \tag{X10}$$

Since *Univ*, like *Xtra*, belongs to the class of systems completing the square, it must also possess these properties. We now list the corresponding properties for *Univ*.

$$K^{\bullet}(v', t') \wedge K^{\bullet}(\underline{v}', t') \Rightarrow v' \sim \underline{v}' \tag{U1}$$

$$(H^{\bullet}(w', t') \vee D^{\bullet}_{Op}(w', t', ...)) \wedge (H^{\bullet}(\underline{w}', t') \vee D^{\bullet}_{Op}(\underline{w}', t', ...)) \Rightarrow\ w' \sim \underline{w}' \qquad \text{(U2)}$$

$$V^{\bullet}_{Op}(p, s) \wedge V^{\bullet}_{Op}(\underline{p}, s) \Rightarrow\ p \sim \underline{p} \qquad \text{(U3)}$$

$$D^{\bullet}_{Op}(w', t', q, s; k, h, w, t) \wedge D^{\bullet}_{Op}(\underline{w}', t', \underline{q}, s; \underline{k}, h, \underline{w}, t) \ \Rightarrow\ q \sim \underline{q} \qquad \text{(U4)}$$

$$K^{\bullet}(v', t') \wedge \underline{v}' \in [v'] \Rightarrow\ K^{\bullet}(\underline{v}', t') \qquad \text{(U5)}$$

$$V^{\bullet}_{Op}(p, s) \wedge \underline{p} \in [p] \Rightarrow V^{\bullet}_{Op}(\underline{p}, s) \qquad \text{(U6)}$$

$$K^{\bullet}(v', t') \ \Rightarrow\ (H^{\bullet}(w', t') \Leftrightarrow H^{\bullet}(w', ([v'], [w'])) \qquad \text{(U7)}$$

$$K^{\bullet}(v', t') \wedge V^{\bullet}_{Op}(p, s) \wedge R^{\bullet}_{Op}(j, h) \wedge K^{\bullet}(v, t) \Rightarrow$$
$$(D^{\bullet}_{Op}(w', t', q, s; k, h, w, t) \Leftrightarrow$$
$$D^{\bullet}_{Op}(w', ([v'], [w']), q, ([p], [q]); k, ([j], [k]), w, ([v], [w]))) \qquad \text{(U8)}$$

$$K^{\bullet}(v', t') \ \Rightarrow\ (\exists\, w' \bullet K^{\bullet}(v', ([v'], [w']))) \qquad \text{(U9)}$$

$$V^{\bullet}_{Op}(p, s\tilde{}) \ \Rightarrow\ (\exists\, q \bullet V^{\bullet}_{Op}(p, ([p], [q]))) \qquad \text{(U10)}$$

(U1) to (U6), (U9) and (U10) are easy to verify and so we leave these to the reader. Consider (U8). First we assume $K^{\bullet}(v', t')$, $V^{\bullet}_{Op}(p, s)$, $R^{\bullet}_{Op}(j, h)$, $K^{\bullet}(v, t)$ and $D^{\bullet}_{Op}(w', t', q, s; k, h, w, t)$ and prove $D^{\bullet}_{Op}(w', ([v'], [w']), q, ([p], [q]); k, ([j], [k]), w, ([v], [w]))$. Let $t' = ([\underline{v}'], [\underline{w}'])$, $s = ([\underline{p}], [\underline{q}])$, $h = ([\underline{j}], [\underline{k}])$ and $t = ([\underline{v}], [\underline{w}])$. From the assumed $D^{\bullet}_{Op}$ and (5.23) we have $(\exists\, u', o, i, u \bullet K(u', w') \wedge V_{Op}(o, q) \wedge R_{Op}(i, k) \wedge K(u, w))$, $VD_{Op}(w', q, k, w, [\underline{p}], [\underline{v}'], [\underline{w}'], [\underline{j}], [\underline{k}], [\underline{v}], [\underline{w}])$, $DV_{Op}([\underline{p}], [\underline{q}])$ and also $w' \sim \underline{w}'$, $q \sim \underline{q}$, $k \sim \underline{k}$ and $w \sim \underline{w}$. Furthermore $K^{\bullet}{}'$, $V^{\bullet}_{Op}$, $R^{\bullet}_{Op}$, $K^{\bullet}$, (5.18), (5.20) and (5.22) give $v' \sim \underline{v}'$, $p \sim \underline{p}$, $j \sim \underline{j}$ and $v \sim \underline{v}$. Hence $VD_{Op}(w', q, k, w, [p], [v'], [w'], [j], [k], [v], [w])$ and $DV_{Op}([p], [q])$ follow. We now have enough to obtain $D^{\bullet}_{Op}(w', ([v'], [w']), q, ([p], [q]); k, ([j], [k]), w, ([v], [w]))$.

Now we assume $K^{\bullet}(v', t')$, $V^{\bullet}_{Op}(p, s)$, $R^{\bullet}_{Op}(j, h)$, $K^{\bullet}(v, t)$ and $D^{\bullet}_{Op}(w', ([v'], [w']), q, ([p], [q]); k, ([j], [k]), w, ([v], [w]))$ and prove $D^{\bullet}_{Op}(w', t', q, s; k, h, w, t)$. Let $t' = ([\underline{v}'], [\underline{w}'])$, $s = ([\underline{p}], [\underline{q}])$, $h = ([\underline{j}], [\underline{k}])$ and $t = ([\underline{v}], [\underline{w}])$. $D^{\bullet}_{Op}(w', ([v'], [w']), q, ([p], [q]); k, ([j], [k]), w, ([v], [w]))$ and (5.23) assert values u', o, i, u exist such that $K(u', w')$, $V_{Op}(o, q)$, $R_{Op}(i, k)$ and $K(u, w)$ hold. Then as $(K(u', w') \wedge V_{Op}(o, q) \wedge R_{Op}(i, k) \wedge K(u, w));D^{\bullet}_{Op}(w', ([v'], [w']), q, ([p], [q]); k, ([j], [k]), w, ([v], [w]))$ holds, $(H(u, \bar{\underline{v}}) \wedge Q_{Op}(i, \bar{\underline{j}}, u, \bar{\underline{v}}) \wedge D_{Op}(u', \bar{\underline{v}}', o, \bar{\underline{p}}; i, \bar{\underline{j}}, u, \bar{\underline{v}}));(K^{\bullet}(\bar{\underline{v}}', ([v'], [w'])) \wedge V^{\bullet}_{Op}(\bar{\underline{p}}, ([p], [q])) \wedge R^{\bullet}_{Op}(\bar{\underline{j}}, ([j], [k])) \wedge K^{\bullet}(\bar{\underline{v}}, ([v], [w])))$ also holds with $\bar{\underline{v}}' \in [v']$, $\bar{\underline{p}} \in [p]$, $\bar{\underline{j}} \in [j]$ and $\bar{\underline{v}} \in [v]$.

From $K^{\bullet}(v', ([\underline{v}'], [\underline{w}']))$ we get $v' \sim \underline{v}'$ and $DK_{Op}([\underline{v}'], [\underline{w}'])$, and thus $DK_{Op}([v'], [\underline{w}'])$. Then $D_{Op}(u', \bar{\underline{v}}', o, \bar{\underline{p}}; i, \bar{\underline{j}}, u, \bar{\underline{v}})$ and $DK_{Op}([v'], [\underline{w}'])$ give $K(u', \bar{w}')$ with $\bar{w}' \sim \underline{w}'$. Now notice we have $K(u', w')$, $K(u', \bar{w}')$ and $D_{Op}(u', \bar{\underline{v}}', o, \bar{\underline{p}}; i, \bar{\underline{j}}, u, \bar{\underline{v}})$. So by (5.5) $w' \sim \bar{w}'$. But as $\bar{w}' \sim \underline{w}'$ then $w' \sim \underline{w}'$. Next, $V^{\bullet}_{Op}(p, ([\underline{p}], [\underline{q}]))$ gives $p \sim \underline{p}$ and $DV_{Op}([\underline{p}], [\underline{q}])$, from which $DV_{Op}([p], [\underline{q}])$ follows. Then $H(u, \bar{\underline{v}})$, $Q_{Op}(i, \bar{\underline{j}}, u, \bar{\underline{v}})$, $D_{Op}(u', \bar{\underline{v}}', o, \bar{\underline{p}}; i, \bar{\underline{j}}, u, \bar{\underline{v}})$, $K^{\bullet}(\bar{\underline{v}}', ([v'], [w']))$, $R^{\bullet}_{Op}(\bar{\underline{j}}, ([j], [k]))$, $K^{\bullet}(\bar{\underline{v}}, ([v], [w]))$ and $DV_{Op}([p], [\underline{q}])$ give $V_{Op}(o, \bar{q})$ with $\bar{q} \sim \underline{q}$, by (5.17). But $V_{Op}(o, q)$, $V_{Op}(o, \bar{q})$ and $D_{Op}(u', \bar{\underline{v}}', o, \bar{\underline{p}}; i, \bar{\underline{j}}, u, \bar{\underline{v}})$ hold. So by (5.9) $q \sim \bar{q}$, and as $\bar{q} \sim \underline{q}$ then $q \sim \underline{q}$. Similarly, $R^{\bullet}_{Op}(j, ([\underline{j}], [\underline{k}]))$ gives $j \sim \underline{j}$ and $QR_{Op}([\underline{j}], [\underline{k}])$, from which $QR_{Op}([j], [\underline{k}])$ follows. Then $H(u, \bar{\underline{v}})$, $Q_{Op}(i, \bar{\underline{j}}, u, \bar{\underline{v}})$, $K^{\bullet}(\bar{\underline{v}}, ([v], [w]))$ and $QR_{Op}([j], [\underline{k}])$ give $R_{Op}(i, \bar{k})$ with $\bar{k} \sim \underline{k}$, by (5.15). Seeing that we have $R_{Op}(i, k)$, $R_{Op}(i, \bar{k})$ and $Q_{Op}(i, \bar{\underline{j}}, u, \bar{\underline{v}})$, by (5.7) $k \sim \bar{k}$, and moreover as $\bar{k} \sim \underline{k}$ then $k \sim \underline{k}$. Last, $K^{\bullet}(v, ([\underline{v}], [\underline{w}]))$ gives $v \sim \underline{v}$ and $HK([\underline{v}], [\underline{w}])$, from which $HK([v], [\underline{w}])$ follows. Then $H(u, \bar{\underline{v}})$ and $HK([v], [\underline{w}])$ give $K(u, \bar{w})$ with $\bar{w} \sim \underline{w}$, by (5.11). Furthermore, as $K(u, w)$ and $K(u, \bar{w})$ hold, by (5.5) $w \sim \bar{w}$. But $\bar{w}$

$\sim \underline{w}$, so $w \sim \underline{w}$. Altogether we now have $v' \sim \underline{v}'$, $p \sim \underline{p}$, $j \sim \underline{j}$, $v \sim \underline{v}$, $w' \sim \underline{w}'$, $q \sim \underline{q}$, $k \sim \underline{k}$ and $w \sim \underline{w}$. Therefore as $D^\bullet_{Op}(w', ([v'], [w']), q, ([p], [q]); k, ([j], [k]), w, ([v], [w]))$ holds, $D^\bullet_{Op}(w', ([\underline{v}'], [\underline{w}']), q, ([\underline{p}], [\underline{q}]); k, ([\underline{j}], [\underline{k}]), w, ([\underline{v}], [\underline{w}]))$ holds too. The proof for (U7) is similar. Done. This completes part (1) of the theorem.

To prove part (2) we must show there is an I/O-filtered refinement from *Univ* to *Xtra*. To this end we now define relations K°, R°_{Op}, V°_{Op}, and prove that they are the retrieve, within and nevertheless relations of the desired refinement. Let

$$K^\circ(t, t^\sim) \;=\; K^\circ(([v], [w]), t^\sim) \;=\; \mathbf{(}(\forall\, \underline{v} \bullet \underline{v} \in [v] \Rightarrow K^\sim(\underline{v}, t^\sim)) \wedge (\forall\, \underline{w} \bullet H^\bullet(\underline{w}, ([v], [w])) \Rightarrow H^\sim(\underline{w}, t^\sim))\mathbf{)} \tag{5.38}$$

For $Op \in \mathsf{Ops}_\mathrm{A}$ we have

$$R^\circ_{Op}(h, h^\sim) \;=\; R^\circ_{Op}(([j], [k]), h^\sim) \;=\; \mathbf{(}(\forall\, \underline{j} \bullet \underline{j} \in [j] \Rightarrow R^\sim_{Op}(\underline{j}, h^\sim)) \wedge (\forall\, \underline{k}, \underline{w}, t, t^\sim \bullet Q^\bullet_{Op}(\underline{k}, ([j], [k]), \underline{w}, t) \wedge K^\circ(t, t^\sim) \Rightarrow Q^\sim_{Op}(\underline{k}, h^\sim, \underline{w}, t^\sim))\mathbf{)} \tag{5.39}$$

$$V^\circ_{Op}(s, s^\sim) \;=\; V^\circ_{Op}(([p], [q]), s^\sim) \;=\; \mathbf{(}(\forall\, \underline{p} \bullet \underline{p} \in [p] \Rightarrow V^\sim_{Op}(\underline{p}, s^\sim)) \wedge (\forall\, \underline{w}', \underline{q}, \underline{k}, \underline{w}, t', t^{\sim\prime}, h, h^\sim, t, t^\sim \bullet D^\bullet_{Op}(\underline{w}', t', \underline{q}, ([p], [q]); \underline{k}, h, \underline{w}, t) \wedge K^\circ(t', t^{\sim\prime}) \wedge R^\circ_{Op}(h, h^\sim) \wedge K^\circ(t, t^\sim) \Rightarrow D^\sim_{Op}(\underline{w}', t^{\sim\prime}, \underline{q}, s^\sim; \underline{k}, h^\sim, \underline{w}, t^\sim))\mathbf{)} \tag{5.40}$$

For $Op \notin \mathsf{Ops}_\mathrm{A}$

$$R^\circ_{Op}(h, h^\sim) \;=\; (h = j \wedge R^\sim_{Op}(j, h^\sim)) \tag{5.41}$$

$$V^\circ_{Op}(s, s^\sim) \;=\; (s = p \wedge V^\sim_{Op}(p, s^\sim)) \tag{5.42}$$

To show that the above satisfy the inclusions stated in part (2) of the theorem is straightforward and we leave this to the reader.

The final task is to discharge the POs of the *Univ* to *Xtra* refinement. As usual, take the Init PO first. This says

$$Init_\mathrm{X}(t^{\sim\prime}) \;\Rightarrow\; (\exists\, t' \bullet Init_\mathrm{U}(t') \wedge K^\circ(t', t^{\sim\prime})) \tag{5.43}$$

Assume the antecedent $Init_\mathrm{X}(t^{\sim\prime})$. In addition, we know we have the refinement from *Ret* to *Xtra* and the retrenchment from *Ref* to *Xtra* for which the Init POs are

$$Init_\mathrm{X}(t^{\sim\prime}) \;\Rightarrow\; (\exists\, v' \bullet Init_\mathrm{T}(v') \wedge K^\sim(v', t^{\sim\prime})) \tag{5.44}$$

$$Init_\mathrm{X}(t^{\sim\prime}) \;\Rightarrow\; (\exists\, w' \bullet Init_\mathrm{F}(w') \wedge H^\sim(w', t^{\sim\prime})) \tag{5.45}$$

respectively. These say that for the initial $t^{\sim\prime}$, there is a v' for which $Init_\mathrm{T}(v')$ and a w' for which $Init_\mathrm{F}(w')$ are true. Furthermore, we also get $K^\sim(v', t^{\sim\prime})$ and $H^\sim(w', t^{\sim\prime})$, from which, by (X7), we get $H^\bullet(w', ([v'], [w'])$. From this we can derive $K^\bullet(v', ([v'], [w']))$ by using (5.18) and (5.19). Now let $t' = ([v'], [w'])$. Then by (5.31) $Init_\mathrm{U}(t')$ is true. It remains to show $K^{\circ\prime}$ holds for this value of t', i.e. that (5.38) holds. To show the first conjunct of (5.38), suppose $\underline{v}' \in [v']$. Then as $K^\sim(v', t^{\sim\prime})$ holds, (X5) asserts that $K^\sim(\underline{v}', t^{\sim\prime})$ holds as required. To establish the second conjunct, assume $H^\bullet(\underline{w}', t')$, i.e. $H^\bullet(\underline{w}', ([v'], [w']))$. But then $H^\bullet(\underline{w}', ([v'], [\underline{w}']))$ holds, and as $K^\sim(v', t^{\sim\prime})$ also holds, $H^\sim(\underline{w}', t^{\sim\prime})$ holds by (X7). We are done.

Now to the business of validating the Op PO. This states

$$K^\circ(t, t\tilde{}) \wedge R^\circ_{Op}(h, h\tilde{}) \wedge stp_{Op_X}(t\tilde{}, h\tilde{}, t\tilde{}', s\tilde{}) \Rightarrow$$
$$(\exists\, t', s \bullet stp_{Op_U}(t, h, t', s) \wedge K^\circ(t', t\tilde{}') \wedge V^\circ_{Op}(s, s\tilde{})) \quad (5.46)$$

We note that Ops_X partitions into Ops_A and non-Ops_A operations. Take the case $Op_X \in \mathsf{Ops}_A$ first. We will need the following lemmas.

Lemma 5.2 Suppose the antecedents of (5.46) hold with $t = ([v], [w])$ and $h = ([j], [k])$. Furthermore suppose $\underline{v} \in [v]$ and $\underline{j} \in [j]$. Then $K\tilde{}(\underline{v}, t\tilde{})$ and $R\tilde{}_{Op}(\underline{j}, h\tilde{})$ hold and moreover there are values, which we fix as $\underline{v}'$ and $\underline{p}$, for which $stp_{Op_T}(\underline{v}, \underline{j}, \underline{v}', \underline{p})$, $K\tilde{}(\underline{v}', t\tilde{}')$ and $V\tilde{}_{Op}(\underline{p}, s\tilde{})$ hold.

Proof. From $K^\circ(t, t\tilde{})$ and $\underline{v} \in [v]$ we get $K\tilde{}(\underline{v}, t\tilde{})$ by (5.38); from $R^\circ_{Op}(h, h\tilde{})$ and $\underline{j} \in [j]$ we get $R\tilde{}_{Op}(\underline{j}, h\tilde{})$ by (5.39). $K\tilde{}$, $R\tilde{}_{Op}$ and stp_{Op_X} are the antecedents of the Op PO for the refinement from *Ret* to *Xtra* which says

$$K\tilde{}(\underline{v}, t\tilde{}) \wedge R\tilde{}_{Op}(\underline{j}, h\tilde{}) \wedge stp_{Op_X}(t\tilde{}, h\tilde{}, t\tilde{}', s\tilde{}) \Rightarrow$$
$$(\exists\, \underline{v}', \underline{p} \bullet stp_{Op_T}(\underline{v}, \underline{j}, \underline{v}', \underline{p}) \wedge K\tilde{}(\underline{v}', t\tilde{}') \wedge V\tilde{}_{Op}(\underline{p}, s\tilde{})) \quad (5.47)$$

Thus we can pick values, which we fix as $\underline{v}'$ and $\underline{p}$, for which $stp_{Op_T}(\underline{v}, \underline{j}, \underline{v}', \underline{p})$, $K\tilde{}(\underline{v}', t\tilde{}')$ and $V\tilde{}_{Op}(\underline{p}, s\tilde{})$ hold. Done.

Lemma 5.3 Suppose the antecedents of (5.46) hold with $t = ([v], [w])$ and $h = ([j], [k])$. Furthermore suppose $H^\bullet(\underline{w}, t)$ and $Q^\bullet_{Op}(\underline{k}, h, \underline{w}, t)$ hold. Then there are values, which we fix as $\underline{w}'$ and $\underline{q}$, for which $stp_{Op_F}(\underline{w}, \underline{k}, \underline{w}', \underline{q})$ and $H\tilde{}(\underline{w}', t\tilde{}') \vee D\tilde{}_{Op}(\underline{w}', t\tilde{}', \underline{q}, s\tilde{}; \underline{k}, h\tilde{}, \underline{w}, t\tilde{})$ hold.

Proof. From $K^\circ(t, t\tilde{})$ and $H^\bullet(\underline{w}, t)$ we get $H\tilde{}(\underline{w}, t\tilde{})$ by (5.38); from $R^\circ_{Op}(h, h\tilde{})$, $Q^\bullet_{Op}(\underline{k}, h, \underline{w}, t)$ and $K^\circ(t, t\tilde{})$ we get $Q\tilde{}_{Op}(\underline{k}, h\tilde{}, \underline{w}, t\tilde{})$ by (5.39). $H\tilde{}$, $Q\tilde{}_{Op}$ and stp_{Op_X} are the antecedents of the Op PO for the retrenchment from *Ref* to *Xtra* which says

$$H\tilde{}(w, t\tilde{}) \wedge Q\tilde{}_{Op}(k, h\tilde{}, w, t\tilde{}) \wedge stp_{Op_X}(t\tilde{}, h\tilde{}, t\tilde{}', s\tilde{}) \Rightarrow$$
$$(\exists\, \underline{w}', \underline{q} \bullet stp_{Op_F}(\underline{w}, \underline{k}, \underline{w}', \underline{q}) \wedge$$
$$(H\tilde{}(\underline{w}', t\tilde{}') \vee D\tilde{}_{Op}(\underline{w}', t\tilde{}', \underline{q}, s\tilde{}; \underline{k}, h\tilde{}, \underline{w}, t\tilde{}))) \quad (5.48)$$

Thus we can pick values, which we fix as $\underline{w}'$ and $\underline{q}$, for which $stp_{Op_F}(\underline{w}, \underline{k}, \underline{w}', \underline{q})$ and $H\tilde{}(\underline{w}', t\tilde{}') \vee D\tilde{}_{Op}(\underline{w}', t\tilde{}', \underline{q}, s\tilde{}; \underline{k}, h\tilde{}, \underline{w}, t\tilde{})$ hold. Done.

To prove (5.46) assume the antecedents with $t = ([v], [w])$ and $h = ([j], [k])$. First we will establish that there are t' and s for which $stp_{Op_U}(t, h, t', s)$ and hence (5.30) holds. Consider (5.30b). For all values $\underline{k}$ and $\underline{w}$ for which the antecedents hold, Lemma 5.3 asserts we have values $\underline{w}'$ and $\underline{q}$ for which $H\tilde{}'$ or $D\tilde{}_{Op}$ and stp_{Op_F} hold. Thus we have three possibilities: (i) there are no values for which the antecedent of (b) holds; (ii) for all values for which the antecedent holds, $H\tilde{}'$ always holds; and (iii) there is at least one pair of values for which the antecedent holds for which $D\tilde{}_{Op}$ holds.

To begin with we establish the following. Since $v \in [v]$ and $j \in [j]$, by Lemma 5.2, $K\tilde{}(v, t\tilde{})$ and $R\tilde{}_{Op}(j, h\tilde{})$ hold, and furthermore there are values, which we fix as v' and p, for which $stp_{Op_T}(v, j, v', p)$, $K\tilde{}(v', t\tilde{}')$ and $V\tilde{}_{Op}(p, s\tilde{})$ hold.

Case (i). (5.30b) holds trivially. We move on to (5.30a). Its antecedent holds for values v and j, so from the previous paragraph we know $stp_{Op_T}(v, j, v', p)$, $K\tilde{}(v', t\tilde{}')$ and $V\tilde{}_{Op}(p, s\tilde{})$ hold. As we have $K\tilde{}'$, (X9) states we have a value, w' say, for which $K^\bullet(v', ([v'], [w']))$ holds. Similarly, $V\tilde{}_{Op}$ and (X10) give $V^\bullet_{Op}(p, ([p], [q]))$. Let $t' = ([v'], [w'])$ and $s = ([p], [q])$. Then the consequent of (a) holds. Having fixed t' and s, we now need to show that for any other choice of values for which the antecedent of (a) holds the consequent does too. So suppose $\underline{v} \in [v]$ and $\underline{j} \in [j]$. Then by Lemma 5.2

we can derive $stp_{Op_T}(\underline{v}, \underline{j}, \underline{v}', \underline{p})$, $K\tilde{}(\underline{v}', t\tilde{}')$ and $V\tilde{}_{Op}(\underline{p}, s\tilde{})$. Since we now have both $K\tilde{}(v', t\tilde{}')$ and $K\tilde{}(\underline{v}', t\tilde{}')$, by (X1) $v' \sim \underline{v}'$ and thus $\underline{v}' \in [v']$. Therefore, as $K^{\bullet}(v', ([v'], [w']))$ holds, by (5.18), $K^{\bullet}(\underline{v}', ([v'], [w']))$ and hence $K^{\bullet}(\underline{v}', t')$ holds. Likewise, from $V\tilde{}_{Op}(p, s\tilde{})$ and $V\tilde{}_{Op}(\underline{p}, s\tilde{})$, by (X3) and (5.22) we establish $V^{\bullet}_{Op}(\underline{p}, s)$ holds. Done.

Case (ii). Let the antecedent of (5.30b) hold for $H^{\bullet}(\underline{w}, t)$ and $Q^{\bullet}_{Op}(\underline{k}, h, \underline{w}, t)$. Then by Lemma 5.3 $stp_{Op_F}(\underline{w}, \underline{k}, \underline{w}', \underline{q})$ and $H\tilde{}(\underline{w}', t\tilde{}')$ hold ($D\tilde{}_{Op}$ is always false for this case). Recalling that we also have $K\tilde{}(v', t\tilde{}')$, we can thus derive $H^{\bullet}(\underline{w}', ([v'], [\underline{w}']))$ by (X7). Now let $t' = ([v'], [\underline{w}'])$. Then the consequent of (b) holds. Furthermore, for any other choice, say $H^{\bullet}(\bar{\underline{w}}, t)$ and $Q^{\bullet}_{Op}(\bar{\underline{k}}, h, \bar{\underline{w}}, t)$, for which the antecedent of (b) holds, by Lemma 5.3, $stp_{Op_F}(\bar{\underline{w}}, \bar{\underline{k}}, \bar{\underline{w}}', \bar{\underline{q}})$ and $H\tilde{}(\bar{\underline{w}}', t\tilde{}')$ hold ($D\tilde{}_{Op}$ is false). Now since we have $H\tilde{}(\bar{\underline{w}}', t\tilde{}')$, $K\tilde{}(v', t\tilde{}')$ and (X7) allow us to derive $H^{\bullet}(\bar{\underline{w}}', ([v'], [\bar{\underline{w}}'])$. Moreover, because we also have $H\tilde{}(\underline{w}', t\tilde{}')$, (X2) says $\underline{w}' \sim \bar{\underline{w}}'$. But this means $H^{\bullet}(\bar{\underline{w}}', ([v'], [\underline{w}'])$ and thus $H^{\bullet}(\bar{\underline{w}}', t')$ holds. Hence (b) always holds for our choice of t'. It remains to set the value of s. We already know $V\tilde{}_{Op}(p, s\tilde{})$ is true. Therefore by (X10) we can pick a value, q say, such that $V^{\bullet}_{Op}(p, ([p], [q]))$ holds. We let $s = ([p], [q])$. We now need to show that (a) always holds for these values of t' and s. Assume the antecedent of (a) holds for $\underline{v} \in [v]$ and $\underline{j} \in [j]$. By Lemma 5.2 $stp_{Op_T}(\underline{v}, \underline{j}, \underline{v}', \underline{p})$, $K\tilde{}(\underline{v}', t\tilde{}')$ and $V\tilde{}_{Op}(\underline{p}, s\tilde{})$ hold. Now $K\tilde{}(v', t\tilde{}')$, $K\tilde{}(\underline{v}', t\tilde{}')$ and (X1) imply $v' \sim \underline{v}'$. So since $H^{\bullet}(\underline{w}', t')$ holds, by (5.19) and (5.18) we can show that $K^{\bullet}(\underline{v}', t')$ holds as well. Finally note that we have $V\tilde{}_{Op}(p, s\tilde{})$ in addition to $V\tilde{}_{Op}(\underline{p}, s\tilde{})$. Therefore (X3) states $p \sim \underline{p}$, and so as $V^{\bullet}_{Op}(p, s)$ holds, by (5.22), $V^{\bullet}_{Op}(\underline{p}, s)$ does too. We are done.

Case (iii). Let the antecedent of (5.30b) hold for $H^{\bullet}(\underline{w}, t)$ and $Q^{\bullet}_{Op}(\underline{k}, h, \underline{w}, t)$. By Lemma 5.3 $stp_{Op_F}(\underline{w}, \underline{k}, \underline{w}', \underline{q})$ and $H\tilde{}(\underline{w}', t\tilde{}')$ or $D\tilde{}_{Op}(\underline{w}', t\tilde{}', \underline{q}, s\tilde{}; \underline{k}, h\tilde{}, \underline{w}, t\tilde{})$ hold, but we will further assume that for this choice of antecedent $D\tilde{}_{Op}$ holds. From Lemma 5.2 we also have $K\tilde{}(v, t\tilde{})$, $R\tilde{}_{Op}(j, h\tilde{})$, $K\tilde{}(v', t\tilde{}')$ and $V\tilde{}_{Op}(p, s\tilde{})$. Applying (X8) we thus get $D^{\bullet}_{Op}(\underline{w}', ([v'], [\underline{w}']), \underline{q}, ([p], [\underline{q}]); \underline{k}, ([j], [\underline{k}]), \underline{w}, ([v], [\underline{w}]))$. Now as $Q^{\bullet}_{Op}(\underline{k}, h, \underline{w}, t)$ holds, because $t = ([v], [w])$ and $h = ([j], [k])$, by (5.21), $\underline{k} \sim k$ and $\underline{w} \sim w$. Thus $D^{\bullet}_{Op}(\underline{w}', ([v'], [\underline{w}']), \underline{q}, ([p], [\underline{q}]); \underline{k}, ([j], [k]), \underline{w}, ([v], [w]))$ holds. Let $t' = ([v'], [\underline{w}'])$ and $s = ([p], [\underline{q}])$. Then the consequent of (b) holds. We now show that (b) holds for any other choice of antecedent. So assume $H^{\bullet}(\bar{\underline{w}}, t)$ and $Q^{\bullet}_{Op}(\bar{\underline{k}}, h, \bar{\underline{w}}, t)$. By Lemma 5.3 $stp_{Op_F}(\bar{\underline{w}}, \bar{\underline{k}}, \bar{\underline{w}}', \bar{\underline{q}})$ and $H\tilde{}(\bar{\underline{w}}', t\tilde{}') \vee D\tilde{}_{Op}(\bar{\underline{w}}', t\tilde{}', \bar{\underline{q}}, s\tilde{}; \bar{\underline{k}}, h\tilde{}, \bar{\underline{w}}, t\tilde{})$ hold. First assume $H\tilde{}(\bar{\underline{w}}', t\tilde{}')$. Then because we have $K\tilde{}(v', t\tilde{}')$, by (X7) we also have $H^{\bullet}(\bar{\underline{w}}', ([v'], [\bar{\underline{w}}'])$. However, by (X2), $H\tilde{}(\bar{\underline{w}}', t\tilde{}')$ and $D\tilde{}_{Op}(\underline{w}', t\tilde{}', ...)$ imply $\bar{\underline{w}}' \sim \underline{w}'$. Hence $H^{\bullet}(\bar{\underline{w}}', ([v'], [\underline{w}'])$ and thus $H^{\bullet}(\bar{\underline{w}}', t')$ holds, and therefore so does the consequent of (b). On the other hand assume $D\tilde{}_{Op}(\bar{\underline{w}}', t\tilde{}', \bar{\underline{q}}, s\tilde{}; \bar{\underline{k}}, h\tilde{}, \bar{\underline{w}}, t\tilde{})$. Because $K\tilde{}(v, t\tilde{})$, $R\tilde{}_{Op}(j, h\tilde{})$, $K\tilde{}(v', t\tilde{}')$, and $V\tilde{}_{Op}(p, s\tilde{})$ all hold, $D^{\bullet}_{Op}(\bar{\underline{w}}', ([v'], [\bar{\underline{w}}']), \bar{\underline{q}}, ([p], [\bar{\underline{q}}]); \bar{\underline{k}}, ([j], [\bar{\underline{k}}]), \bar{\underline{w}}, ([v], [\bar{\underline{w}}]))$ holds by (X8). But $D\tilde{}_{Op}(\underline{w}', t\tilde{}', \underline{q}, s\tilde{}; ...)$ and $D\tilde{}_{Op}(\bar{\underline{w}}', t\tilde{}', \bar{\underline{q}}, s\tilde{}; ...)$ state $\bar{\underline{w}}' \sim \underline{w}'$ by (X2) and $\bar{\underline{q}} \sim \underline{q}$ by (X4). Moreover $Q^{\bullet}_{Op}(\bar{\underline{k}}, h, \bar{\underline{w}}, t)$ and (5.21) state $\bar{\underline{k}} \sim k$ and $\bar{\underline{w}} \sim w$. As a result $D^{\bullet}_{Op}(\bar{\underline{w}}', ([v'], [\underline{w}']), \bar{\underline{q}}, ([p], [\underline{q}]); \bar{\underline{k}}, ([j], [k]), \bar{\underline{w}}, ([v], [w]))$ or equivalently $D^{\bullet}_{Op}(\bar{\underline{w}}', t', \bar{\underline{q}}, s; \bar{\underline{k}}, h, \bar{\underline{w}}, t)$ holds. Hence the consequent of (b) holds for $H^{\bullet}(\bar{\underline{w}}, t)$ and $Q^{\bullet}_{Op}(\bar{\underline{k}}, h, \bar{\underline{w}}, t)$.

We now show that (5.30a) always holds for $t' = ([v'], [\underline{w}'])$ and $s = ([p], [\underline{q}])$. Assume the antecedent holds for $\underline{v} \in [v]$ and $\underline{j} \in [j]$. By Lemma 5.2 $stp_{Op_T}(\underline{v}, \underline{j}, \underline{v}', \underline{p})$, $K\tilde{}(\underline{v}', t\tilde{}')$ and $V\tilde{}_{Op}(\underline{p}, s\tilde{})$ hold. Then $K\tilde{}(v', t\tilde{}')$, $K\tilde{}(\underline{v}', t\tilde{}')$ and (X1) say $v' \sim \underline{v}'$; $V\tilde{}_{Op}(p, s\tilde{})$, $V\tilde{}_{Op}(\underline{p}, s\tilde{})$ and (X3) say $p \sim \underline{p}$. From above we know $D^{\bullet}_{Op}(\underline{w}', ([v'], [\underline{w}']), \underline{q}, ([p], [\underline{q}]); \underline{k}, ([j], [k]), \underline{w}, ([v], [w]))$ is true. Hence (5.23) says $K(u', \underline{w}')$, $V_{Op}(o, \underline{q})$, $R_{Op}(i, \underline{k})$, $K(u, \underline{w})$ and $VD_{Op}(\underline{w}', \underline{q}, \underline{k}, \underline{w}, [p], [v'], [\underline{w}'], [j], [k], [v], [w])$ are true, from which by (5.16) we conclude $K^{\bullet}(\underline{v}\,', ([v'], [\underline{w}']))$ is true, with $\underline{v}\,' \sim v'$. But $v' \sim \underline{v}'$, so $\underline{v}\,' \sim \underline{v}'$, and hence

$K^{\bullet}(\underline{v}', ([v'], [\underline{w}']))$ or equivalently $K^{\bullet}(\underline{v}', t')$ holds. In addition (5.23) says $DV_{Op}([p], [\underline{q}])$ is true and since $p \sim \underline{p}$, $V^{\bullet}_{Op}(\underline{p}, ([p], [\underline{q}]))$ or equivalently $V^{\bullet}_{Op}(\underline{p}, s)$ follows by (5.22). Ergo (a) holds for t' and s.

All that remains is to show $K^{\circ}(t', \tilde{t}')$ and $V^{\circ}_{Op}(s, \tilde{s})$ for $t' = ([v'], [\underline{w}'])$ and $s = ([p], [\underline{q}])$. To show $K^{\circ\prime}$ we must demonstrate that both conjuncts of (5.38) hold. Take the first conjunct and assume $\underline{v}' \in [v']$. Then because $\tilde{K}(v', \tilde{t}')$ is true, (X5) states $\tilde{K}(\underline{v}', \tilde{t}')$ is true. Hence the first conjunct holds. We move on to the second. Assume $H^{\bullet}(\overline{\underline{w}}', ([v'], [\underline{w}']))$. (5.19) says $\overline{\underline{w}}' \in [\underline{w}']$, which means $H^{\bullet}(\overline{\underline{w}}', ([v'], [\overline{\underline{w}}']))$ is also true. From this, by (X7), $\tilde{H}(\overline{\underline{w}}', \tilde{t}')$ holds, and as a result $K^{\circ}(t', \tilde{t}')$ holds. To show V°_{Op} we must verify both conjuncts of (5.40). To prove the first conjunct assume $\underline{p} \in [p]$. Then because $\tilde{V}_{Op}(p, \tilde{s})$ is true, (X6) states that $\tilde{V}_{Op}(\underline{p}, \tilde{s})$ is true. To establish the second conjunct, we assume $\tilde{D}^{\bullet}_{Op}(\overline{\underline{w}}', \underline{t}', \bar{\underline{q}}, ([p], [\underline{q}]); \bar{\underline{k}}, \underline{h}, \overline{\underline{w}}, \underline{t})$, $K^{\circ}(\underline{t}', \tilde{\underline{t}}')$, $R^{\circ}_{Op}(\underline{h}, \tilde{\underline{h}})$ and $K^{\circ}(\underline{t}, \tilde{\underline{t}})$ where $\underline{t}' = ([\bar{\underline{v}}'], [\overline{\underline{w}}'])$, $\underline{h} = ([\bar{\underline{j}}], [\bar{\underline{k}}])$, $\underline{t} = ([\bar{\underline{v}}], [\overline{\underline{w}}])$ and $([p], [\underline{q}]) = s = ([p], [\bar{\underline{q}}])$. Now, $\bar{\underline{v}}' \in [\bar{\underline{v}}']$, $K^{\circ}(\underline{t}', \tilde{\underline{t}}')$ and (5.38) give $\tilde{K}(\bar{\underline{v}}', \tilde{\underline{t}}')$; $\bar{\underline{j}} \in [\bar{\underline{j}}]$, $R^{\circ}_{Op}(\underline{h}, \tilde{\underline{h}})$ and (5.39) give $\tilde{R}_{Op}(\bar{\underline{j}}, \tilde{\underline{h}})$; and $\bar{\underline{v}} \in [\bar{\underline{v}}]$, $K^{\circ}(\underline{t}, \tilde{\underline{t}})$ and (5.38) give $\tilde{K}(\bar{\underline{v}}, \tilde{\underline{t}})$. Then because we also have $\tilde{V}_{Op}(p, \tilde{s})$ (X8) gives $\tilde{D}_{Op}(\overline{\underline{w}}', \tilde{\underline{t}}', \bar{\underline{q}}, \tilde{s}; \bar{\underline{k}}, \tilde{\underline{h}}, \overline{\underline{w}}, \tilde{\underline{t}})$. Therefore the second conjunct holds and we are done.

Case $Op_X \notin \mathsf{Ops}_A$. We will use the following lemma.

Lemma 5.4 Suppose the antecedents of (5.46) hold with $t = ([v], [w])$ and $h = \underline{j}$. Furthermore suppose $\underline{v} \in [v]$. Then there are values, which we fix as $\underline{v}'$ and $\underline{p}$, for which $stp_{Op_T}(\underline{v}, \underline{j}, \underline{v}', \underline{p})$, $\tilde{K}(\underline{v}', \tilde{t}')$ and $\tilde{V}_{Op}(\underline{p}, \tilde{s})$ hold.

Proof. From $K^{\circ}(t, \tilde{t})$ and $\underline{v} \in [v]$ we get $\tilde{K}(\underline{v}, \tilde{t})$ by (5.38); from $R^{\circ}_{Op}(h, \tilde{h})$ and $h = \underline{j}$ we get $\tilde{R}_{Op}(\underline{j}, \tilde{h})$ by (5.41). $\tilde{K}$, $\tilde{R}_{Op}$ and stp_{Op_X} are the antecedents of PO (5.47), so we can pick values, which we fix as $\underline{v}'$ and $\underline{p}$, for which $stp_{Op_T}(\underline{v}, \underline{j}, \underline{v}', \underline{p})$, $\tilde{K}(\underline{v}', \tilde{t}')$ and $\tilde{V}_{Op}(\underline{p}, \tilde{s})$ hold. Done.

Now assume the antecedents of (5.46) and let $t = ([v], [w])$ and $h = j$. Since $v \in [v]$, by Lemma 5.4, there are values, which we fix as v' and p, such that $stp_{Op_T}(v, j, v', p)$, $\tilde{K}(v', \tilde{t}')$ and $\tilde{V}_{Op}(p, \tilde{s})$ hold. Then $\tilde{K}(v', \tilde{t}')$ and (X9) allow us to pick a value, say $\underline{w}'$, for which $K^{\bullet}(v', ([v'], [\underline{w}']))$ holds. Let $t' = ([v'], [\underline{w}'])$ and $s = p$. We now show that for these values of t' and s, (5.28) and thus stp_{Op_U} holds. Suppose $\underline{v} \in [v]$. Then using Lemma 5.4 we can derive $stp_{Op_T}(\underline{v}, j, \underline{v}', \underline{p})$, $\tilde{K}(\underline{v}', \tilde{t}')$ and $\tilde{V}_{Op}(\underline{p}, \tilde{s})$. Now since $\tilde{K}(v', \tilde{t}')$ and $\tilde{K}(\underline{v}', \tilde{t}')$ are true, by (X1), $v' \sim \underline{v}'$. Therefore, because $K^{\bullet}(v', ([v'], [\underline{w}']))$ is true, $K^{\bullet}(\underline{v}', ([v'], [\underline{w}']))$ is also true. Furthermore, $\tilde{V}_{Op}(p, \tilde{s})$, $\tilde{V}_{Op}(\underline{p}, \tilde{s})$ and (X3) imply $\underline{p} = p$ (because the equivalence class is a singleton). Hence $stp_{Op_U}(t, h, t', s)$ holds for values t' and s. Finally we show that $K^{\circ}(t', \tilde{t}')$ and $V^{\circ}_{Op}(s, \tilde{s})$ hold. The proof for $K^{\circ}(t', \tilde{t}')$ is identical to the one for the case $Op_X \in \mathsf{Ops}_A$. We transfer our attention to $V^{\circ}_{Op}(s, \tilde{s})$. For this we must verify (5.42) which we do easily as we already have $\tilde{V}_{Op}(p, \tilde{s})$ and $s = p$. Done. This completes part (2) of the theorem.

Part (3) follows readily by observing that for a system *Univ** having the same properties as *Univ*, there will be an I/O-filtered refinement from *Univ* to *Univ** and an I/O-filtered refinement from *Univ** to *Univ*. We are done. ☺

6 An Example

Let us return to our running example and see what the universal construction means in this setting. The states t of the *Univ* are pairs of equivalence classes $([v], [w])$ where the elements of $[v]$ are injective sequences of NATs of length ≤ 10, and the el-

ements of [w] are pairs (w^a, w^b) of disjoint sets of NATs differing in cardinality by no more than 2. So for example ([<57, 217325>], [({1, 2}, {4, 5, 7})]) is a valid state.

The non-junk steps of *Univ* in non-boundary cases look like

([<3, 4>], [({3}, {4})])
-(([1], [$\underline{1}$]), $AddEl_U$)-›
([<1, 3, 4>], [({1, 3}, {4})]) (6.1)

We can verify this step satisfies (5.30). First note the following. The *Abs* state {3, 4} is linked to the *Ret* states <3, 4> and <4, 3> by H, and to the *Ref* states ({3, 4}, ∅), (∅, {3, 4}), ({3} ,{4}) and ({4}, {3}) by K. Thus using (5.4) the two *Ret* states are the only members of [<3, 4>]. Similarly, by (5.5), the four *Ref* states are the only members of [({3}, {4})]. Given this, it is not too difficult to convince ourselves that HK([<3, 4>], [({3}, {4})]) holds. Moreover, we can see that DK([<3, 4>], [({3}, {4})]) holds trivially, since there is no D. A similar analysis establishes HK and DK for the pair ([<1, 3, 4>], [({1, 3}, {4})]). Note also that [1] and [$\underline{1}$] are both singletons.

Given the above, we can now persuade ourselves that both parts of (5.30) hold. Consider (5.30a) and let $\underline{v}$ = <3, 4> and $\underline{j}$ = 1. We have to demonstrate that there is a $\underline{v}'$ such that stp_{AddEl_T} and $K^{\bullet\prime}$ hold (we ignore $V^{\bullet}$ because there are no outputs). As $\underline{v}$ = <3, 4>, the only possible transition is <3, 4>-(1, stp_{AddEl_T})-›<3, 4, 1>, so $\underline{v}'$ has to be <3, 4, 1>. Hence, stp_{AddEl_T}(<3, 4>, 1, <3, 4, 1>) holds, and as <3, 4, 1> ∈ [<1, 3, 4>], $K^{\bullet\prime}$ does too. It should now be clear that (a) also holds for $\underline{v}$ = <4, 3>. Turning to (5.30b) the process is much the same. Take the case $\underline{w}$ = ({3, 4}, ∅) and $\underline{k}$ = $\underline{1}$. Assume $H^{\bullet}$ and $Q^{\bullet}$ hold. This time we want a $\underline{w}'$ such that stp_{AddEl_F} holds. Given the behaviour of $AddEl_F$, we know stp_{AddEl_F}(({3, 4}, ∅), $\underline{1}$, ({3, 4}, {1})) is true. Therefore let $\underline{w}'$ = ({3, 4}, {1}). It is now easy to satisfy ourselves that KH(({3, 4}, {1}), ([<1, 3, 4>], [({1, 3}, {4})])) holds. It follows that $H^{\bullet\prime}$ does too. Therefore the consequent of part (b) holds as desired. A similar process establishes (b) for the remaining three choices for $\underline{w}$. Thus (6.1) is valid. From the above we thus see that the equivalence classes have bunched together all the *Ret* and *Ref* steps which correspond to {3, 4}-(1, stp_{AddEl_A})-›{1, 3, 4} in *Abs*.

Turning to the boundary cases, a typical step looks like

([<1…10>], [({1…5}, {6…10})])
-(([11], [$\underline{11}$]), $AddEl_U$)-›
([<1…10>], [({1…5}, {6…10})]) (6.2)

To verify (5.30) for this step we start by observing the following. The elements of [<1...10>] are all the possible serialisations of the *Abs* state {1...10}. The elements of [({1…5}, {6…10})] are all the valid pairs (w^a, w^b) where $w^a \cup w^b$ is in Z = {x ∈ NAT • {1...10} ∪ {x}}, this being the set of all *Abs* states that are of interest. The relationship between the *Abs*, *Ret* and *Ref* states is shown in Fig. 2. For clarity, we have included only two of the *Ret* states that are in [<1…10>], only two of the possible *Ref* states for each *Abs* state, and only two of the *Abs* states in Z that are linked to the sequences in [<1…10>] by D. Given this arrangement it is not too difficult to assure ourselves that the members of the classes are as we claim, and to see that both HK and DK hold for ([<1…10>], [({1…5}, {6…10})]). Note also that [11] and [$\underline{11}$] are both singletons.

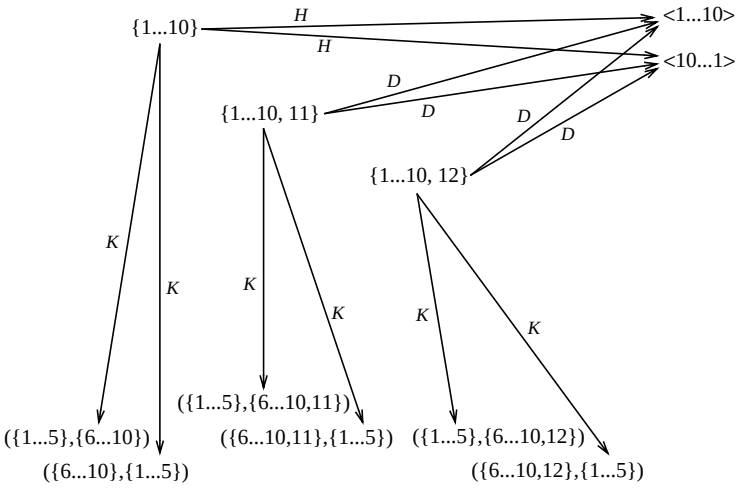

Fig. 2

It is now easy to confirm (5.30a). We have $\underline{j} = 11$, and for any choice of $\underline{v}$ in [<1…10>], since stp_{AddEl_T} does a *skip*, $\underline{v}' = \underline{v}$. Therefore, seeing that we have HK and DK, $K^{\bullet\prime}$ must hold. This completes part (a). We move on to (b). Let $\underline{w} = (\{1\ldots5\}, \{6\ldots10\})$, $\underline{k} = \underline{11}$ and assume $H^{\bullet}$ and $Q^{\bullet}$ hold. From $Q^{\bullet}$ we also know $QR([11], [\underline{11}])$ holds. As $\underline{k} = \underline{11}$, $\underline{w}' = (\{1\ldots5\}, \{6\ldots11\})$ is one choice which satisfies stp_{AddEl_F}. We will now satisfy ourselves that $D^{\bullet}$ holds for this choice of $\underline{w}'$. Examining (5.23) we see that for $u' = \{1...11\}$, $i = 11$ and $u = \{1...10\}$, K', R and K hold. As there are no outputs, all that remains is to show $VD_{Op}(\underline{w}', \underline{k}, \underline{w}, [v'], [w'], [j], [k], [v], [w])$. To do this we must establish (5.16). Notice that its antecedent only holds for the values of u', i and u just given. Furthermore, if we let $\underline{v}' = <1...10>$, $\underline{j} = 11$ and $\underline{v} = \underline{v}'$, then for these values the consequent of (5.16) holds. Similar remarks apply for any other suitable choice for $\underline{w}$.

7 Conclusion

In the preceding sections we showed how to take a retrenchment and a refinement of the same system, and build a system which reconciles the two development steps. The construction we gave was universal within a class of similar reconciliations defined by the list of properties (X1)-(X10). We illustrated our result on a small example whose size was dictated more by space considerations than by its ability to convince the reader of its usefulness. Nevertheless the situation it embodies typifies many of the cases in which the retrenchment technique has its *raison d'etre*, namely where there are details of the real world application whose incorporation one wishes to postpone till a later stage of the development, but for which this would be precluded by the exigencies of the refinement POs.

The pushout-like construction of this paper is one key result in the full algebraic integration of retrenchment and refinement. Other results appear in the already cited works [6, 7, 8], and yet others are work in progress. The complete suite of results

will ensure that retrenchment adds a truly fresh dimension to possible development routes, complementing refinement in situations where it struggles to give a convincing account, rather than being a stand-alone technique that fragments development routes into incompatible possibilities. Ultimately the developer gains by having a richer integrated palette of tools with which to organise the building of systems.

References

1. Banach R., Poppleton M. Retrenchment: An Engineering Variation on Refinement. *in*: Proc. B-98, Bert (ed.), LNCS **1393**, 129-147, Springer (1998). *See also*: UMCS Technical Report UMCS-99-3-2, `http://www.cs.man.ac.uk/cstechrep`
2. Banach R., Poppleton M. Retrenchment and Punctured Simulation. *in*: Proc. IFM-99, Taguchi, Galloway (eds.), 457-476, Springer (1999).
3. Banach R., Poppleton M. Retrenchment, Refinement and Simulation. *in*: Proc. ZB-00, Bowen, Dunne, Galloway, King (eds.), LNCS **1878**, 304-323, Springer (2000).
4. Banach R., Poppleton M. Sharp Retrenchment, Modulated Refinement and Simulation. Form. Asp. Comp. **11** (1999), 498-540.
5. Banach R., Poppleton M. Engineering and Theoretical Underpinnings of Retrenchment. `http://www.cs.man.ac.uk/~banach/some.pubs/Retrench.Underpin.ps.gz` *submitted*.
6. Banach R., Jeske C. Output Retrenchments, Defaults, Stronger Compositions, Feature Engineering. `http://www.cs.man.ac.uk/~banach/some.pubs/Retrench.Def.Out.ps.gz` *submitted*.
7. Banach R. Maximally Abstract Retrenchments. *in*: Proc. IEEE ICFEM-00, 133-142, (2000).
8. Jeske C., Banach R. Minimally and Maximally Abstract Retrenchments. *in*: Proc. IFM-02, LNCS **2335**, 380-399, Springer (2002).
9. de Roever W.-P., Engelhardt K. Data Refinement: Model-Oriented Proof Methods and their Comparison. C.U.P. (1998).
10. Abrial J. R. The B-Book. C.U.P. (1996).
11. Schneider S. The B-Method: An Introduction. Palgrave (2001).
12. Neilson D. S. From Z to C: Illustration of a Rigorous Development Method. PhD. Thesis, Oxford University Computing Laboratory Programming Research Group, Technical Monograph PRG-101 (1990).
13. Owe O. An Approach to Program Reasoning Based on a First Order Logic for Partial Functions. University of Oslo Institute of Informatics Research Report No. 89. ISBN 82-90230-88-5 (1985).
14. Owe O. Partial Logics Reconsidered: A Conservative Approach. Form. Asp. Comp. **3** (1993), 1-16.
15. Hayes I. J., Sanders J. W. Specification by Interface Separation. Form. Asp. Comp. **7** (1995), 430-439.
16. Mikhajlova A, Sekerinski E. Class Refinement and Interface Refinement in Object-Oriented Programs. *in*: Proc. FME-97, Fitzgerald, Jones, Lucas (eds.), LNCS **1313**, 82-101, Springer (1997).
17. Stepney S., Cooper D., Woodcock J. More Powerful Z Data Refinement: Pushing the State of the Art in Industrial Refinement. *in*: Proc. ZUM-98, Bowen, Fett, Hinchey (eds.), LNCS **1493**, 284-307, Springer (1998).
18. Derrick J., Boiten E. Refinement in Z and Object-Z: Foundations and Advanced Applications. Springer (2001).