ELECTRONIC **WORKSHOPS IN COMPUTING**

Series edited by Professor C.J. van Rijsbergen

D. J. Duke , University of York, UK and A.S. Evans, University of Bradford, UK (Eds)

2nd BCS-FACS Northern Formal Methods Workshop

Proceedings of the 2nd BCS-FACS Northern Formal Methods Workshop, Ilkley, 14-15 July 1997

Algebraic Advances for Aliasing

A.P. Hughes and A. Donnelly

Published in collaboration with the British Computer Society

BCS FACS

Springer

Algebraic Advances for Aliasing

Arthur Hughes   Alexis Donnelly
Department of Computer Science
Trinity College Dublin
Ireland
e-mail: {Arthur.P.Hughes,Alexis.Donnelly}@cs.tcd.ie

Abstract

Using algebraic structures and techniques alone we derive an intuitive result concerning updates to a system of aliases. Specifically, we use the *kernel relation* of a map to characterise the system of aliases; we express the inverse image of map override as a new operation, called "*underride*", which we define; we provide an important theorem relating map composition, override and underride in a very natural way; and finally we identify another satisfying and insightful algebraic approach to the same problem based on algebraic properties of the solution space. We thereby illustrate our contention that this use of abstract algebra extends the mathematical foundations of software engineering, provides a conveniently high level at which to reason about models (shorter intuitive proofs) and promotes increased mathematical insight on the part of practitioners

1 Introduction

> "There is an enormous wealth of basic mathematics in formal specifications just waiting to be discovered, a body of knowledge which must form the foundation of software engineering."
>
> [Mac90]

The *VDM* [Jon90] and *Z* [Spi92] notations have been used widely in the specification and development of software systems. These methods share a mathematical foundation of set theory and logic. However, it is possible to bring results from abstract algebra to bear in the development process, an option preferred by the *Irish School of Constructive Mathematics* ($\mathsf{M}^{\clubsuit}_{C}$), of which the *Irish School of the VDM* ($VDM^{\clubsuit}$) is a part. This school uses a classical engineering style of proof, in which equals are substituted for equals [GS93].

The principal contributions of this paper are the development of some new algebraic results concerning map composition, override, and a new operation, "underride". The setting used for these developments is a model of aliasing. Specifically,

- we use the *kernel relation* of a map to characterise the system of aliases;
- we express the inverse image of map override as a new operation, called "underride", which we define;
- we provide an important theorem relating map composition ($\circ$), override ($\dagger$) and underride ($\rotatebox[origin=c]{180}{$\dagger$}$) which expands the expression $(\alpha \rotatebox[origin=c]{180}{$\dagger$} \beta) \circ (\mu \dagger \nu)$ in a very natural way;
- using only algebra we prove the intuitive result that any modification to a system of aliases must be at the granularity of collections of aliases, and finally
- we identify another algebraic approach to the same problem based on algebraic properties of the solution space and suggest that it leads to a more elegant and satisfying understanding of the problem.

Our intention is to highlight the mathematics, not introduce yet another syntactic extension. We believe this mathematics could be expressed in most of the popular notations such as *Z* or *VDM-SL*.

In addition, we believe our preference for using further mathematics offers several advantages:

- the mathematical foundations of the engineering discipline are extended to another useful branch of discrete mathematics;
- the mathematics is at a higher level than logic and set theory, offering the developer more powerful (mathematical) tools (shorter proofs);
- the "matching of the mathematics to the model" referred to above offers two more subtle advantages: the model is nearer the focus of attention than with the lower level mathematics so easing the transitions between model and mathematics; and an innate awareness of (more abstract) mathematics is promoted, raising the practitioner's awareness of the mathematical nature of common models and hence of possible defects in the corresponding artifacts.

In this paper we illustrate these claims by presenting some recent work. An intuitively obvious property of a system of aliases is derived algebraically. We capture the relationship between aliases using the *kernel relation* of a map, and then proceed to derive algebraically how the system may be changed so as to preserve the alias relationships. Some useful algebraic results involving map composition, override and a new operation, "underride", are developed along the way.

Section 2 introduces our notation, and the problem is stated mathematically in Section 3. The algebraic structures used are presented in Section 4, while Section 6 develops a promising alternative approach. Finally, Section 7 discusses present and future work before concluding.

2 Notation

The $\mathsf{M}^{\clubsuit}_{C}$ uses a characteristic notation [Mac90], [Mac91], [Mac93] and proof style [But93], [Hug95]. For readers unfamiliar with this style, we briefly introduce some notations and state some important definitions.

In the $\mathsf{M}^{\clubsuit}_{C}$we begin by constructing a model in the problem domain and emphasise mathematical structures and the relationships between them. Starting with fundamental domains (which are not described in further detail), structures are built using *functors*.

The *powerset functor*, $\mathcal{P}-$, is used to construct sets ($-$ is just an argument place-holder): $\mathcal{P}A$ denotes the space of sets of elements from domain A. The empty set is denoted by $\emptyset$.

The *map functor*, $- \to -$, constructs maps: $X \to Y$ denotes the space of partial (including total) functions from X to Y. The identity map is denoted by $\mathcal{I}$ and the empty map by θ. We use exponentiation (following [Jac74]) to denote the space of total functions: Y^X denotes the space of all total mappings in $X \to Y$ and $\{y\}^X$ indicates a constant (total) function which maps all elements of X to the value y. Where no ambiguity arises, we sometimes drop the brackets.

We use *priming* in the $\mathsf{M}^{\clubsuit}_{C}$ to denote the absence of an obvious null element from a structure. Thus $\mathcal{P}'X$ denotes the space of sets of X with the empty set removed. Hence $x \in \mathcal{P}'X$ means that x is some non-empty set of elements from X. Similarly, $\rho \in (X \to \mathcal{P}'Y)'$ states that ρ is some non-empty function from X to (non-empty) sets from Y. It is also worth noting that we routinely view relations as set-valued functions.

For two functions $f : X \to A$ and $g : Y \to B$ we define a functor called a *map iterator*, written $(f \to g)$ which can be applied to a map $\rho : X \to Y$. The result is a map from A to B inheriting the associations of ρ and constructed from it by applying f to elements of the domain of ρ and applying g to corresponding elements of the range of ρ. There are some restrictions on f: it must be one-to-one and total on the domain of ρ. The structure we describe might also be written using composition: $g \circ \rho \circ f^{-1} = (f \to g)\rho$. However we prefer the $- \to -$ notation[1] as it pictorially represents the underlying construction.

[1]The *hom* functor (see [BW95], p 60) is sometimes denoted by $- \to -$, so our notation is somewhat unusual.

The *removal* of a map $\mu \in X \to Y$ with respect to a set S is denoted by $S \ntriangleleft \mu$. We also use curried forms $\ntriangleleft[S]\mu$ or $\ntriangleleft_S\mu$ especially in connection with the map iterator mentioned above. Thus, $\tau' = (\mathcal{I} \to \ntriangleleft_S)\mu$ represents the map μ with the set S systemtically removed from all its range elements.

In a similar manner we denote *restriction* of map μ with respect to a set S by the forms $S \lhd \mu$, $\lhd[S]\mu$ or $\lhd_S\mu$. Thus $\lhd_{\mathtt{dom}\,\nu}\mu$ denotes μ restricted to the domain of ν.

The characteristic function, denoted by the curried form $\chi[\![-]\!]$, tests a set for membership with respect to a particular element.

A *Monoid*, denoted $(X, \bullet, \iota)$, is an algebraic structure consisting of a base set, X, and a binary operator, $\bullet$, for which there exists a unique identity element, ι. The base set X, must be closed under $\bullet$ which must also be associative. The monoid structure is proving increasingly useful in computer science [BW95].

We use, $\circledcirc$, to denote relational union which can be expressed in terms of map override, $\dagger$, and map extend, $\sqcup$, as follows:

$$\alpha \circledcirc [x \mapsto y] = \begin{cases} \alpha \sqcup [x \mapsto y] & \neg\chi[\![x]\!]\alpha \\ \alpha \dagger [x \mapsto \alpha(x) \cup y] & \text{otherwise} \end{cases}$$

This operation is more fully defined using the algebraic structure of a direct power (see Appendix B). We have established by somewhat tedious inductive arguments [Hug96] that the two definitions are equivalent. Additional results used in the paper are given in the text.

Our proof style is equational, following Gries [GS93], where equals are substituted for equals. Each substitution is justified in a comment or hint following the equals sign between the expressions.

3 Introducing the Abstract Model

The setting for this study is a two-map model:

$$\begin{aligned} \mathsf{Env} &= \mathsf{Names} \to \mathsf{Locations} \\ \mathsf{Store} &= \mathsf{Locations} \to \mathsf{Contents}. \end{aligned}$$

Env is a space of (finite) maps from names (Names) to locations (Locations) and Store is another map from locations to their contents, (Contents). In the context of the World Wide Web, Names might be the space of hypertext links (i.e. clickable text and embedded *URL*) and Locations the space of *URL*s. Contents might denote the space of possible *HTML* file contents. In this case, Env is just a simple projection function. A more interesting model arises if Names is the space of e-mail addresses. Maps in Env might then represent the interests of various individuals (represented by e-mail addresses) in various web pages (referenced by *URL*). By manipulation of elements of Env one might construct mailing lists to support discussion, notify of changes or survey interested parties. We do not pursue the model further since aliasing is the focus of this paper. However, we note in passing, that this model is found in other areas of computing: If the triple (Names, Locations, Contents) is replaced by (Identifiers, Locations, Values) we have the classical model of an imperative programming language [BJ82]. If the replacement is (Filenames, INodes, Data) we have a `UNIX`-like filesystem [BJ82].

It is often useful to group together all the names referring to the same location of interest. In the web models outlined above, one might wish to change all hypertext links at a particular site or page when the *URL* to which they refer[2] changes (i.e. the referenced web page moves). Where Names refers to e-mail addresses, one might wish to notify interested parties of an editorial change[3], or of a change in location. In all the cases above, a collection of *aliases* is being constructed, the term being borrowed from the classical

[2]A more comprehensive list of referring pages can be generated by the *AltaVista* search engine (see use of the `link:` field in the Advanced search help pages on `http://www.altavista.digital.com`).

[3]A web service, known as the *URL-Minder* (`http://www.netmind.com`) notifies one automatically of such changes, but one has to subscribe manually.

models above. Note in the case of the web model, that aliasing can also arise in Store where, by *virtual hosting*[4], different *URLs* can refer to the same (physical) web page.

It is quite simple to generate the set of referring names in our model. If $\mu \in \mathsf{Env}$, then simply form its inverse image, μ^{-1}, and compose with μ as the following example shows:

$$\mu \;=\; \begin{bmatrix} n_1 & \mapsto & l_1 \\ n_2 & \mapsto & l_1 \\ n_3 & \mapsto & l_2 \end{bmatrix} \qquad \mu^{-1} \;=\; \begin{bmatrix} l_1 & \mapsto & \{n_1, n_2\} \\ l_2 & \mapsto & \{n_3\} \end{bmatrix}$$

$$\mu^{-1} \circ \mu \;=\; \begin{bmatrix} n_1 & \mapsto & \{n_1, n_2\} \\ n_2 & \mapsto & \{n_1, n_2\} \\ n_3 & \mapsto & \{n_3\} \end{bmatrix}$$

For this purpose, we define an operator, $\sigma : \mathsf{Env} \rightarrow (\mathsf{Names} \rightarrow \mathcal{P}'\mathsf{Names})$, on environments which yields a new map, linking each name to the set of names also referring to the same web object:

$$\sigma(\mu) \;\triangleq\; \mu^{-1} \circ \mu \qquad (1)$$

where $()^{-1}$ denotes inverse image and $\circ$ denotes composition of maps. The inverse image of a map is easily constructed by mapping each element of the range to the set of domain elements which map to it in the map itself. In this case, $\texttt{rng}\ \mu$ and $\texttt{dom}\ \mu^{-1}$ coincide so that map composition is easily constructed. For brevity in the presentation we say that $\sigma(\mu)$ maps each name in μ to the set of its *aliases*.

Effectively, σ partitions Names into equivalence classes based on μ. Thus, $\sigma(\mu)$ is the *kernel relation* of μ (see [Gol84], page 66).

Now suppose we alter the environment, μ, by overriding it with another (small) map, ν, representing a change to the environment. The altered environment is $\mu \dagger \nu$, where $\dagger$ represents map override. We name this alteration by the operator $\psi_\nu : \mathsf{Env} \rightarrow \mathsf{Env}$

$$\psi_\nu(\mu) \triangleq \mu \dagger \nu \qquad (2)$$

We are concerned with identifying which transformations, ψ_ν, leave the pattern of aliases unchanged, i.e. which ψ_ν leave $\sigma(\mu)$ unchanged:

$$\sigma(\mu) = \sigma(\psi_\nu(\mu)) \qquad (3)$$

Our task is then to solve this equation for ψ_ν — to discover the alias-preserving transformations. We note, in passing, the similarity of this equation to distance-preserving transformations in vector spaces:

$$||x|| = ||T(x)|| \qquad (4)$$

4 Algebraic Structures Used

We now outline the algebra used to solve Equation (3) above. The definitions of σ and ψ_ν may be immediately applied:

$$\begin{array}{ll} & \sigma(\psi_\nu(\mu)) \\ = & \langle \text{using equation (2)} \rangle \\ & \sigma(\mu \dagger \nu) \\ = & \langle \text{using equation (1)} \rangle \\ & (\mu \dagger \nu)^{-1} \circ (\mu \dagger \nu) \end{array}$$

[4] The Domain Name Server (DNS) can delegate to an existing server within its domain the authority to manage an additional sub-domain. In this way, a web page can be accessed using two different *URLs*, though only one of them is usually publicised.

We will now examine how inverse image interacts with override in $(\mu \dagger \nu)^{-1}$. We replace this expression by its equivalent in the inverse space using a new operation, "underride" (defined below, see Equation 5 and Theorem 1). In what follows, $(X \rightarrow Y)^{-1}$ denotes the space of inverse image maps.

Definition 1 *Given* $\alpha, \beta \in (X \rightarrow Y)^{-1}$, *define*

$$\alpha \mathbin{\rotatebox[origin=c]{180}{$\dagger$}} \beta \triangleq (\mathcal{I} \rightarrow \triangleleft\!\!\!-_{\cup/\mathbf{rng}\,\beta})'\alpha \circledcup \beta \tag{5}$$

In this equation, $\triangleleft\!\!\!-_{\cup/\mathbf{rng}\,\beta}$, which appears as the second component of a map iterator, denotes set removal with respect to the accumulated elements of the range of β. The priming denotes removal of any maplets of the form $y \mapsto \emptyset$ and $\circledcup$ denotes relational union. An illustrative example of this operator appears after the following isomorphism theorem.

Theorem 1 *The inverse image space forms a monoid* $((X \rightarrow Y)^{-1}, \mathbin{\rotatebox[origin=c]{180}{$\dagger$}}, \theta)$ *with* $\mathbin{\rotatebox[origin=c]{180}{$\dagger$}}$ *which is isomorphic to the monoid of maps,* $(X \rightarrow Y, \dagger, \theta)$. *Further, the isomorphism between these two monoids is inverse image,* $()^{-1}$, *i.e. given* $\mu, \nu \in X \rightarrow Y$,

$$(\mu \dagger \nu)^{-1} = \mu^{-1} \mathbin{\rotatebox[origin=c]{180}{$\dagger$}} \nu^{-1} \tag{6}$$

The proof of this theorem is given in Appendix A. We note in passing that the space $(Y \rightarrow \mathcal{P}'X)$ is larger than the inverse space, $(X \rightarrow Y)^{-1}$. Historically, this encouraged us to seek and discover the sought after monoid in the inverse space. An example illustrates: given $\delta \in X \rightarrow Y$ and $\eta \in Y \rightarrow \mathcal{P}'X$, say

$$\delta = \begin{bmatrix} x_1 & \mapsto & y_1 \\ x_2 & \mapsto & y_1 \\ x_3 & \mapsto & y_2 \end{bmatrix} \qquad \delta^{-1} = \begin{bmatrix} y_1 & \mapsto & \{x_1, x_2\} \\ y_2 & \mapsto & \{x_3\} \end{bmatrix} \qquad \eta = \begin{bmatrix} y_4 & \mapsto & \{x_5\} \\ y_5 & \mapsto & \{x_5\} \end{bmatrix}$$

The example of η above shows that there can be no map in $(X \rightarrow Y)$ of which it is the inverse (since then x_5 would be mapped to both y_4 and y_5). Using the counter-example of η we arrive at the result:

$$(X \rightarrow Y)^{-1} \subset (Y \rightarrow \mathcal{P}'X) \tag{7}$$

We now illustrate the definition of underride, $\mathbin{\rotatebox[origin=c]{180}{$\dagger$}}$, and Theorem 1 (equation (6), to familiarise the reader with the new operation.

$$\mu = \begin{bmatrix} x_1 & \mapsto & y_5 \\ x_2 & \mapsto & y_1 \\ x_3 & \mapsto & y_2 \\ x_5 & \mapsto & y_4 \\ x_6 & \mapsto & y_5 \end{bmatrix} \qquad \nu = \begin{bmatrix} x_1 & \mapsto & y_3 \\ x_2 & \mapsto & y_1 \\ x_3 & \mapsto & y_3 \end{bmatrix}$$

$$\mu^{-1} = \begin{bmatrix} y_1 & \mapsto & \{x_2\} \\ y_2 & \mapsto & \{x_3\} \\ y_4 & \mapsto & \{x_5\} \\ y_5 & \mapsto & \{x_1, x_6\} \end{bmatrix} \qquad \nu^{-1} = \begin{bmatrix} y_1 & \mapsto & \{x_2\} \\ y_3 & \mapsto & \{x_1, x_3\} \end{bmatrix}$$

Applying the definition from Equation (5) in two stages and inverting we obtain:

$$
\begin{aligned}
&(\mathcal{I} \rightarrow \ndres_{\cup/\mathtt{rng}\,\nu^{-1}})'\mu^{-1} = (\mathcal{I} \rightarrow \ndres_{\mathtt{dom}\,\nu})'\mu^{-1} \\
&= \begin{bmatrix} y_4 & \mapsto & \{x_5\} \\ y_5 & \mapsto & \{x_6\} \end{bmatrix}
\end{aligned}
$$

$$
\begin{aligned}
&(\mathcal{I} \rightarrow \ndres_{\cup/\mathtt{rng}\,\nu^{-1}})'\mu^{-1} \circledcup \nu^{-1} = (\mathcal{I} \rightarrow \ndres_{\mathtt{dom}\,\nu})'\mu^{-1} \circledcup \nu^{-1} \\
&= \begin{bmatrix} y_1 & \mapsto & \{x_2\} \\ y_3 & \mapsto & \{x_1, x_3\} \\ y_4 & \mapsto & \{x_5\} \\ y_5 & \mapsto & \{x_6\} \end{bmatrix}
\end{aligned}
$$

$$
\begin{aligned}
&\mu \dagger \nu \\
&= \begin{bmatrix} x_1 & \mapsto & y_3 \\ x_2 & \mapsto & y_1 \\ x_3 & \mapsto & y_3 \\ x_5 & \mapsto & y_4 \\ x_6 & \mapsto & y_5 \end{bmatrix}
\end{aligned}
$$

$$
\begin{aligned}
&(\mu \dagger \nu)^{-1} \\
&= \begin{bmatrix} y_1 & \mapsto & \{x_2\} \\ y_3 & \mapsto & \{x_1, x_3\} \\ y_4 & \mapsto & \{x_5\} \\ y_5 & \mapsto & \{x_6\} \end{bmatrix}
\end{aligned}
$$

We now resume our search for a solution to Equation (3).

$$
\begin{aligned}
& \sigma(\psi_\nu(\mu)) \\
= \quad & \langle \text{by previous work} \rangle \\
& (\mu \dagger \nu)^{-1} \circ (\mu \dagger \nu) \\
= \quad & \langle \text{by Theorem 1} \rangle \\
& (\mu^{-1} \ddagger \nu^{-1}) \circ (\mu \dagger \nu)
\end{aligned}
$$

Since map composition is defined only on the intersection, $\mathtt{rng}\,\alpha \cap \mathtt{dom}\,\beta$, in $\beta \circ \alpha$, we may write the following expression for it using a map iterator.

Definition 2 *If $\alpha \in X \rightarrow Y$ and $\beta \in Y \rightarrow Z$ then*

$$\beta \circ \alpha \triangleq (\mathcal{I} \rightarrow \beta) \rhd_{\mathtt{dom}\,\beta}\, \alpha \tag{8}$$

This expression first restricts the range of α to $\mathtt{dom}\,\beta$ before applying β to all range elements via the iterator. The interaction of composition with override and underride is expanded by Theorem 2 below.

Theorem 2 *If $\alpha, \beta \in X \rightarrow Y$ and $\mu, \nu \in (Z \rightarrow Y)^{-1}$, then*

$$
\begin{aligned}
(\mu \ddagger \nu) \circ (\alpha \dagger \beta) \quad = \quad & (\mathcal{I} \rightarrow \ndres_{\cup/\mathtt{rng}\,\nu})' \ndres_{\mathtt{dom}\,\beta}(\mu \circ \alpha) \\
& \circledcup\ (\mathcal{I} \rightarrow \ndres_{\cup/\mathtt{rng}\,\nu})'(\mu \circ \beta) \\
& \circledcup\ \ndres_{\mathtt{dom}\,\beta}(\nu \circ \alpha) \\
& \circledcup\ (\nu \circ \beta)
\end{aligned} \tag{9}
$$

This equation is somewhat reminiscent (if one ignores the map iterators and miscellaneous removals) of the arithmetic identity:

$$(u + v)(a + b) \quad = \quad ua + ub + va + vb$$

Two lemmas are required to prove [Hug97] the theorem. The first relates composition and underride, the second composition and override.

Lemma 1 *If $\alpha \in X \rightarrow Y$ and $\mu, \nu \in (Z \rightarrow Y)^{-1}$, then*

$$(\mu \ddagger \nu) \circ \alpha = (\mathcal{I} \rightarrow \ndres_{\cup/\mathtt{rng}\,\nu})'(\mu \circ \alpha) \circledcup (\nu \circ \alpha) \tag{10}$$

Lemma 2 *If $\alpha, \beta \in X \rightarrow Y$ and $\mu \in Y \rightarrow Z$ then,*

$$\mu \circ (\alpha \dagger \beta) = \ndres_{\mathtt{dom}\,\beta}(\mu \circ \alpha) \sqcup (\mu \circ \beta) \tag{11}$$

Proofs of the lemmas are offered in Appendices C and D. Note that the analogy with arithmetic continues:

$$
\begin{aligned}
(u+v)a &= ua+va \\
u(a+b) &= ua+ub
\end{aligned}
$$

We now resume our search for a solution to Equation (3).

$$
\begin{aligned}
& (\mu^{-1} \mathbin{\text{⸸}} \nu^{-1}) \circ (\mu \dagger \nu) \\
= \quad & \langle \text{by Theorem 2} \rangle \\
& (\mathcal{I} \to \mathbin{\lhd\!\!\!-}_{\cup/\mathtt{rng}\,\nu^{-1}})' \mathbin{\lhd\!\!\!-}_{\mathtt{dom}\,\nu} (\mu^{-1} \circ \mu) \;\mathbin{\bigcirc\!\!\!\!\!\cup}\; (\mathcal{I} \to \mathbin{\lhd\!\!\!-}_{\cup/\mathtt{rng}\,\nu^{-1}})'(\mu^{-1} \circ \nu) \;\mathbin{\bigcirc\!\!\!\!\!\cup}\; \mathbin{\lhd\!\!\!-}_{\mathtt{dom}\,\nu}(\nu^{-1} \circ \mu) \;\mathbin{\bigcirc\!\!\!\!\!\cup}\; (\nu^{-1} \circ \nu) \\
= \quad & \langle \text{since } {}^{\cup}/\mathtt{rng}\,\nu^{-1} = \mathtt{dom}\,\nu \rangle \\
& (\mathcal{I} \to \mathbin{\lhd\!\!\!-}_{\mathtt{dom}\,\nu})' \mathbin{\lhd\!\!\!-}_{\mathtt{dom}\,\nu} (\mu^{-1} \circ \mu) \;\mathbin{\bigcirc\!\!\!\!\!\cup}\; (\mathcal{I} \to \mathbin{\lhd\!\!\!-}_{\mathtt{dom}\,\nu})'(\mu^{-1} \circ \nu) \;\mathbin{\bigcirc\!\!\!\!\!\cup}\; \mathbin{\lhd\!\!\!-}_{\mathtt{dom}\,\nu}(\nu^{-1} \circ \mu) \;\mathbin{\bigcirc\!\!\!\!\!\cup}\; (\nu^{-1} \circ \nu) \\
= \quad & \langle \text{from the definition of } \sigma \text{ — Equation (1)} \rangle \\
& (\mathcal{I} \to \mathbin{\lhd\!\!\!-}_{\mathtt{dom}\,\nu})' \mathbin{\lhd\!\!\!-}_{\mathtt{dom}\,\nu} \sigma(\mu) \;\mathbin{\bigcirc\!\!\!\!\!\cup}\; (\mathcal{I} \to \mathbin{\lhd\!\!\!-}_{\mathtt{dom}\,\nu})'(\mu^{-1} \circ \nu) \;\mathbin{\bigcirc\!\!\!\!\!\cup}\; \mathbin{\lhd\!\!\!-}_{\mathtt{dom}\,\nu}(\nu^{-1} \circ \mu) \;\mathbin{\bigcirc\!\!\!\!\!\cup}\; \sigma(\nu)
\end{aligned}
$$

To facilitate the reader we make some abbreviations for the removal operators in what follows.

$$\mathcal{R}_\nu \triangleq (\mathcal{I} \to \mathbin{\lhd\!\!\!-}_{\mathtt{dom}\,\nu})' \tag{12}$$

$$\mathcal{F}_\nu \triangleq \mathbin{\lhd\!\!\!-}_{\mathtt{dom}\,\nu} \tag{13}$$

We must now solve the following equation for $\sigma(\nu)$ and hence for ν.

$$\sigma(\mu) = \mathcal{R}_\nu \mathcal{F}_\nu \sigma(\mu) \;\mathbin{\bigcirc\!\!\!\!\!\cup}\; \mathcal{R}_\nu(\mu^{-1} \circ \nu) \;\mathbin{\bigcirc\!\!\!\!\!\cup}\; \mathcal{F}_\nu(\nu^{-1} \circ \mu) \;\mathbin{\bigcirc\!\!\!\!\!\cup}\; \sigma(\nu) \tag{14}$$

5 Towards a Solution

We continue our search for a solution by applying the operator, $\mathcal{R}_\nu$, to both sides, knowing that $\mathcal{R}_\nu$ is idempotent since it involves removal.

$$
\begin{aligned}
& \mathcal{R}_\nu \sigma(\mu) \\
= \quad & \langle \text{by distribution of } \mathcal{R}_\nu \text{ over } \mathbin{\bigcirc\!\!\!\!\!\cup} \rangle \\
& \mathcal{R}_\nu \mathcal{R}_\nu \mathcal{F}_\nu \sigma(\mu) \;\mathbin{\bigcirc\!\!\!\!\!\cup}\; \mathcal{R}_\nu \mathcal{R}_\nu(\mu^{-1} \circ \nu) \;\mathbin{\bigcirc\!\!\!\!\!\cup}\; \mathcal{R}_\nu \mathcal{F}_\nu(\nu^{-1} \circ \mu) \;\mathbin{\bigcirc\!\!\!\!\!\cup}\; \mathcal{R}_\nu \sigma(\nu) \\
= \quad & \langle \text{by idempotency of } \mathcal{R}_\nu \rangle \\
& \mathcal{R}_\nu \mathcal{F}_\nu \sigma(\mu) \;\mathbin{\bigcirc\!\!\!\!\!\cup}\; \mathcal{R}_\nu(\mu^{-1} \circ \nu) \;\mathbin{\bigcirc\!\!\!\!\!\cup}\; \mathcal{R}_\nu \mathcal{F}_\nu(\nu^{-1} \circ \mu) \;\mathbin{\bigcirc\!\!\!\!\!\cup}\; \mathcal{R}_\nu \sigma(\nu) \\
= \quad & \langle\, \mathcal{R}_\nu \sigma(\nu) = (\mathcal{I} \to \mathbin{\lhd\!\!\!-}_{\mathtt{dom}\,\nu})'\sigma(\nu) = \theta \text{ — removal from a map of its entire domain} \rangle \\
& \mathcal{R}_\nu \mathcal{F}_\nu \sigma(\mu) \;\mathbin{\bigcirc\!\!\!\!\!\cup}\; \mathcal{R}_\nu(\mu^{-1} \circ \nu) \;\mathbin{\bigcirc\!\!\!\!\!\cup}\; \mathcal{R}_\nu \mathcal{F}_\nu(\nu^{-1} \circ \mu) \\
= \quad & \langle \text{by commutativity of operators } \mathcal{R}_\nu \text{ and } \mathcal{F}_\nu \rangle \\
& \mathcal{R}_\nu \mathcal{F}_\nu \sigma(\mu) \;\mathbin{\bigcirc\!\!\!\!\!\cup}\; \mathcal{R}_\nu(\mu^{-1} \circ \nu) \;\mathbin{\bigcirc\!\!\!\!\!\cup}\; \mathcal{F}_\nu \mathcal{R}_\nu(\nu^{-1} \circ \mu) \\
= \quad & \langle\, (\mathcal{I} \to \mathbin{\lhd\!\!\!-}_{\mathtt{dom}\,\nu})'(\nu^{-1} \circ \mu) = \theta \text{ since all range elements are systematically reduced} \\
& \quad \text{to } \emptyset \text{ by } \mathcal{R}_\nu \text{ and then removed.} \rangle \\
& \mathcal{R}_\nu \mathcal{F}_\nu \sigma(\mu) \;\mathbin{\bigcirc\!\!\!\!\!\cup}\; \mathcal{R}_\nu(\mu^{-1} \circ \nu)
\end{aligned}
$$

We now make a simplifying assumption: that the ranges of μ and ν are disjoint. This is quite reasonable in the WWW system being modelled since when a web page is moved, it is usually given a new *URL* and then the ranges of μ and ν will be disjoint. We note that the case where locations (to which a set of aliases might refer) are simply swapped is not covered (in the latter case the alias structure will be preserved).

$$\mathtt{rng}\,\mu \cap \mathtt{rng}\,\nu = \emptyset \tag{15}$$

Under this condition two terms (one already removed) reduce to the null map.

$$\mu^{-1} \circ \nu = \theta \tag{16}$$

$$\nu^{-1} \circ \mu = \theta \tag{17}$$

Our equation can be simplified.

$$\mathcal{R}_\nu \sigma(\mu) = \mathcal{R}_\nu \mathcal{F}_\nu \sigma(\mu) = \mathcal{F}_\nu \mathcal{R}_\nu \sigma(\mu) \text{ by commutativity of } \mathcal{R}_\nu \text{ and } \mathcal{F}_\nu \tag{18}$$

That is, removal with respect to $\mathtt{dom}\,\nu$ (i.e. $\mathcal{F}_\nu$) has no effect on $\mathcal{R}_\nu \sigma(\nu)$. Their domains therefore do not intersect:

$$\mathtt{dom}\,\nu \cap \mathtt{dom}\,(\mathcal{I} \rightarrow \lhd_{\mathtt{dom}\,\nu})' \sigma(\mu) = \emptyset \tag{19}$$

Thus the effect of $(\mathcal{I} \rightarrow \lhd_{\mathtt{dom}\,\nu})'$ (or $\mathcal{R}_\nu$) will be to remove all elements of the domain of ν from $\sigma(\mu)$. This means

$$(\mathcal{I} \rightarrow \lhd_{\mathtt{dom}\,\nu})' \sigma(\mu) = \lhd_{\mathtt{dom}\,\nu} \sigma(\mu) \quad \text{or} \quad \mathcal{R}_\nu \sigma(\mu) = \mathcal{F}_\nu \sigma(\mu) \tag{20}$$

i.e., both operators have the same effect on $\sigma(\mu)$.
We already know that $\sigma(\mu)$ is a *class function* (see [Hil75], page 86), i.e. it is constant over the members of the equivalence class induced by μ (see Figure 1 below). The notation $[n]$ denotes the members of n's equivalence class (where the equivalence relation is induced by μ) and n is an element of Names. $\{[n]\}$ denotes the singleton set containing n's equivalence class. $\sigma(\mu)$ can thus be written as a series of map-extensions:

$$\sigma(\mu) = \bigsqcup_{[n] \in \mathtt{rng}\ \sigma(\mu)} \{[n]\}^{[n]} \tag{21}$$

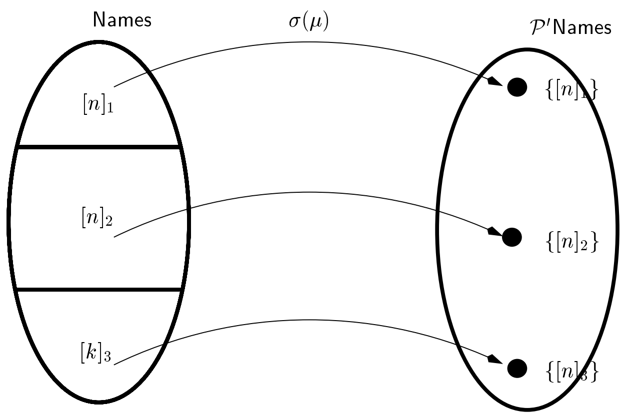

Figure 1: Class Function

In order for the effects on $\sigma(\mu)$ of the operators $\mathcal{R}_\nu$ and $\mathcal{F}_\nu$ to be the same, $\sigma(\nu)$ must act on entire equivalence classes in $\sigma(\mu)$. There can be no acting on part of an equivalence class as such actions would violate equation (20). Thus the domain of $\sigma(\nu)$ can be expressed as the union of *some* of $\sigma(\mu)$'s equivalence classes:

$$dom\ \sigma(\nu) = [n_1] \cup \ldots \cup [n_m]$$

where $[n_1] \ldots [n_m] \in rng\, \sigma(\mu)$. Furthermore, $\sigma(\nu)$, like $\sigma(\mu)$ must, by the kernel relation argument presented earlier, also be a class function. Therefore we can write it too as a sequence of map extensions:

$$\sigma(\nu) = \{[n_1]\}^{[n_1]} \sqcup \ldots \sqcup \{[n_m]\}^{[n_m]} \tag{22}$$

where $\{[n_1]\} \ldots \{[n_m]\}$ are *some* equivalence classes in $\sigma(\mu)$. Thus $\sigma(\mu)$ is also constant over entire equivalence classes.

At last we can characterise ν.

$$\nu = \{l_1\}^{[n_1]} \sqcup \ldots \sqcup \{l_m\}^{[n_m]} \tag{23}$$

where $l_i \in \mathsf{Locations}$, that is, ν is constant on those equivalence classes, $[n_i] \in rng\sigma(\mu)$ which appear in $\mathtt{dom}\,\nu$. In addition, the constant values, l_i, are unique locations different from the locations in the environment μ.

The practical implication of the solution derived algebraically above is a familiar result. Any updates, to a system of aliases must, in order to leave the pattern of aliases unchanged, operate upon entire alias classes.

We now present a useful result concerning the solution space, inspired a little by the distance-preserving equation from vector spaces (see equation (4)). The intuition here is to see if $(\nu_1 \dagger \nu_2)$ is a solution, given that ν_1 and ν_2 are both solutions. The latter facts give us, via equations (14), (16) and (17):

$$\sigma(\mu) = \sigma(\mu \dagger \nu_1) \quad = \quad \mathcal{R}_{\nu_1}\mathcal{F}_{\nu_1}\sigma(\mu) \quad ⋓ \quad \sigma(\nu_1) \tag{24}$$

$$\sigma(\mu) = \sigma(\mu \dagger \nu_2) \quad = \quad \mathcal{R}_{\nu_2}\mathcal{F}_{\nu_2}\sigma(\mu) \quad ⋓ \quad \sigma(\nu_2) \tag{25}$$

Our demonstration proceeds as follows:

$$\begin{array}{ll}
 & \sigma(\mu) \\
= & \langle \text{replacing } \sigma(\mu) \text{ in (24) from (25)} \rangle \\
 & \mathcal{R}_{\nu_1}\mathcal{F}_{\nu_1}[\mathcal{R}_{\nu_2}\mathcal{F}_{\nu_2}\sigma(\mu) \quad ⋓ \quad \sigma(\nu_2)] \quad ⋓ \quad \sigma(\nu_1) \\
= & \langle \text{by distribution of operators } \mathcal{F} \text{ and } \mathcal{R} \text{ over } ⋓ \rangle \\
 & \mathcal{R}_{\nu_1}\mathcal{F}_{\nu_1}\mathcal{R}_{\nu_2}\mathcal{F}_{\nu_2}\sigma(\mu) \quad ⋓ \quad \mathcal{R}_{\nu_1}\mathcal{F}_{\nu_1}\sigma(\nu_2) \quad ⋓ \quad \sigma(\nu_1) \\
= & \langle \text{commutativity of operators} \rangle \\
 & \mathcal{R}_{\nu_1}\mathcal{R}_{\nu_2}\mathcal{F}_{\nu_1}\mathcal{F}_{\nu_2}\sigma(\mu) \quad ⋓ \quad \mathcal{R}_{\nu_1}\mathcal{F}_{\nu_1}\sigma(\nu_2) \quad ⋓ \quad \sigma(\nu_1) \\
= & \langle \text{by definition of } \sigma(\nu_2 \dagger \nu_1) \text{ via (24)} \rangle \\
 & \mathcal{R}_{\nu_1}\mathcal{R}_{\nu_2}\mathcal{F}_{\nu_1}\mathcal{F}_{\nu_2}\sigma(\mu) \quad ⋓ \quad \sigma(\nu_2 \dagger \nu_1) \\
= & \langle \text{by } \mathcal{R}_{\nu_1}\mathcal{R}_{\nu_2} = \mathcal{R}_{\nu_2 \dagger \nu_1} \text{ and similarly for } \mathcal{F} \rangle \\
 & \mathcal{R}_{\nu_2 \dagger \nu_1}\mathcal{F}_{\nu_2 \dagger \nu_1}\sigma(\mu) \quad ⋓ \quad \sigma(\nu_2 \dagger \nu_1) \\
= & \langle \text{by (24) with } (\nu_2 \dagger \nu_1) \text{ replacing } \nu_1 \rangle \\
 & \sigma(\mu \dagger (\nu_2 \dagger \nu_1))
\end{array}$$

This result implies that the solution space of $\sigma(\mu) = \sigma(\mu \dagger \nu)$, which we denote by $\mathcal{A}_\mu$, is closed under override. Since $\dagger$ is associative and has a unique identity, θ, we can say that $(\mathcal{A}_\mu, \dagger, \theta)$ is a monoid. We strongly suspect that if the update to μ were expressed as a map composition, a group structure would exist in the solution space. Inverses could be found using inverse maps. The alternative update expression is defined by:

$$\psi'_\nu(\mu) \triangleq \nu \circ \mu \tag{26}$$

where $\nu \in \mathsf{Locations} \rightarrow \mathsf{Locations}$ and ν is one-to-one.

6 An Alternative Approach

Based on the insight behind equation (26), we demonstrate that if ν is a permutation of $\mathsf{Locations}$, then it may preserve the structure of aliases — a more comprehensive, elegant and satisfying statement of the solution. Note however, that our hypothesis rests on on Proposition 1 below, which we have not as yet proven. Nevertheless, our algebraic insight on the character of the solution space, $\mathcal{A}_\mu$, appears to suggest a fruitful avenue for exploration.

We recast the modification of the structure as a composition with a permutation of $\mathsf{Locations}$. Permutation is a group action on the set of partial maps $\mathsf{Names} \rightarrow \mathsf{Locations}$, from which we obtain [Jac74], [BW95]:

$$(h, (g, x)) \quad = \quad (hg, x) \tag{27}$$

$$(e, x) \quad = \quad x \tag{28}$$

where $g, h \in \mathsf{Locations} \rightarrow \mathsf{Locations}$(the group) and $x \in \mathsf{Names} \rightarrow \mathsf{Locations}$, e is the identity of the group.

Lemma 3 *The group of permutations of the set* $\mathsf{Locations}$*, denoted* $S_{\mathsf{Locations}}$*, acts on the set of partial maps* $\mathsf{Names} \rightarrow \mathsf{Locations}$ *by*

$$\begin{array}{ccc} S_{\mathsf{Locations}} \times (\mathsf{Names} \rightarrow \mathsf{Locations}) & \rightarrow & (\mathsf{Names} \rightarrow \mathsf{Locations}) \\ (\nu, \mu) & \mapsto & \nu \circ \mu. \end{array}$$

Proof If $\nu, \tau \in S_{\mathsf{Locations}}$ and $\mu \in \mathsf{Names} \rightarrow \mathsf{Locations}$ is $\nu(\tau\mu) = (\nu\tau)\mu$?

$$\begin{array}{ll} & \nu(\tau\mu) \\ = & \langle\text{Definition of group action.}\rangle \\ & \nu(\tau \circ \mu) \\ = & \langle\text{Definition of group action.}\rangle \\ & \nu \circ (\tau \circ \mu) \\ = & \langle\text{Composition is associative.}\rangle \\ & (\nu \circ \tau) \circ \mu \\ = & \langle\text{Definition of group product.}\rangle \\ & (\nu\tau) \circ \mu \\ = & \langle\text{Definition of group action.}\rangle \\ & (\nu\tau)\mu \end{array}$$

If $\mu \in \mathsf{Names} \rightarrow \mathsf{Locations}$ is $\mathcal{I}\mu = \mu$?

$$\begin{array}{ll} & \mathcal{I}\mu \\ = & \langle\text{Definition of group action.}\rangle \\ & \mathcal{I} \circ \mu \\ = & \langle\text{Evaluating the composition.}\rangle \\ & \mu \end{array}$$

Definition 3 *The aliases of a mapping* $\mu \in \mathsf{Names} \rightarrow \mathsf{Locations}$ *can be found by applying*

$$\begin{array}{lccc} \sigma : & (\mathsf{Names} \rightarrow \mathsf{Locations}) & \rightarrow & (\mathsf{Names} \rightarrow \mathcal{P}'\,\mathsf{Names}) \\ \sigma : & \mu & \mapsto & \mu^{-1} \circ \mu. \end{array}$$

Proposition 1 *If* $\alpha \in X \rightarrow Y$*,* $\beta \in Y \rightarrow Z$ *and* β *is one-to-one, then*

$$(\beta \circ \alpha)^{-1} = \alpha^{-1} \circ \beta^{-1}$$

where α^{-1} *denotes the inverse image of the map* α *and* β^{-1} *denotes the inverse map of* β*.*

Lemma 4 *The above action of* $S_{\mathsf{Locations}}$ *on* $\mathsf{Names} \rightarrow \mathsf{Locations}$ *induces alias-preserving transformations.*

Proof If $\mu \in \mathsf{Names} \rightarrow \mathsf{Locations}$ and $\nu \in S_{\mathsf{Locations}}$ is $\sigma(\nu\mu) = \sigma(\mu)$?

$$\begin{array}{ll} & \sigma(\nu\mu) \\ = & \langle\text{Definition of group action.}\rangle \\ & \sigma(\nu \circ \mu) \\ = & \langle\text{Definition of alias operator.}\rangle \\ & (\nu \circ \mu)^{-1} \circ (\nu \circ \mu) \\ = & \langle\text{Applying Proposition 1}\rangle \\ & (\mu^{-1} \circ \nu^{-1}) \circ (\nu \circ \mu) \\ = & \langle\text{Composition is associative.}\rangle \\ & \mu^{-1} \circ \nu^{-1} \circ \nu \circ \mu \\ = & \langle\text{The inverse element of } \nu \text{ in the group } S_{\mathsf{Locations}} \text{ is } \nu^{-1}.\rangle \\ & \mu^{-1} \circ \mathcal{I} \circ \mu \\ = & \langle\text{Evaluating composition.}\rangle \\ & \mu^{-1} \circ \mu \\ = & \langle\text{Definition of alias operator.}\rangle \\ & \sigma(\mu) \end{array}$$

7 Conclusions

In this paper we have used aspects of abstract algebra to construct a model of a system of aliases. In the process we uncovered some important results concerning composition, override and (a new operation) "underride" for maps. Our mathematics has followed the classical engineering style of substituting equals for equals. Specifically,

- we have used the *kernel relation* of a map to characterise alias relationships;
- we have discovered the inverse image of override and in the process defined a new operation, "underride" for maps;
- we have developed an important theorem relating composition, override and underride which expands the expression, $(\alpha \ddagger \beta) \circ (\mu \dagger \nu)$, in a natural way;
- we have proved a well-known result about updates to a system of aliases in a purely algebraic way, and finally
- we have discovered strong hints about possible further algebraic structure (a group) in the solution space. This lead us to re-formulate the problem and more easily derive a more elegant and satisfying solution.

Some aspects of our presentation deserve further discussion however. We assumed that $\texttt{rng}\,\mu \cap \texttt{rng}\,\nu = \emptyset$ in order to simplify the equation to be solved. We intend to explore the other cases as well. As suggested in the text, work with the composition alternative suggests that some similar reductions may be found.

We would also like to improve the final steps in our derivation of a solution — a more methodical and algebraic approach would be preferable.

An interesting occurrence was the equivalence of the operators $\mathcal{R}_\nu$ and $\mathcal{F}_\nu$ when applied to some structures. What other structures will exhibit similar properties?

Based on the group structure of permutations, we should explore whether the solution space, $\mathcal{A}_\mu$, also has a group structure. What is the group operation in this case? What morphism(s) exist between $\mathcal{A}_\mu$ and permutations? Are there some sub-groups of permutations isomorphic to $\mathcal{A}_\mu$?

Another area is of course to seek further computing applications for this algebra.

A novel feature of the work reported here has been our preference for seeking (all) transformations which preserve a structure, whereas the traditional approach has been to suggest candidate transformations and then investigate whether they preserve the structure. If the response is negative, then side (or pre-) conditions are sought and imposed.

We believe that we have illustrated our claim that use of results from abstract algebra extends the mathematical basis of software engineering, provides a convenient and usefully high level view of the system under study and promotes increased mathematical maturity among practitioners leading to improved quality of the finished artifact.

If presented with the following alternatives: the use of a little abstract algebra allowing us to produce short, insightful proofs in developing a software system or the use of some automated tools based on set theory and logic that draw us continually away from our model, we know what our choice will be.

Acknowledgements

We would like to thank our colleagues in the Foundations and Methods Group, TCD for their encouragement and feedback, particularly Dr Mícheál MacAnAirchinnigh. We would also like to express our thanks to a diligent, but anonymous, reviewer whose comments helped us improve the presentation of the paper.

References

[BJ82] Dines Bjørner and Cliff Jones. *Formal Specification and Software Development.* Prentice Hall, 1982.

[Bow95] Bowen, Jonathan P. and Hinchey, Michael G., editor. *ZUM'95: The Z Formal Specification Notation, Proceedings of the* 9^{th} *International Conference of Z Users, Limerick, Ireland, 4-9 September 1995.*, volume 967 of *Lecture Notes in Computer Science.* Springer-Verlag, September 1995.

[But93] Butterfield, Andrew. A VDM$^\clubsuit$ Study of Fault-Tolerant Stable Storage: Towards a Computer Engineering Mathematics. In Woodcock, J. and Larsen, P.G., editor, *FME'93: Industrial Strength Formal Methods, Odense, Denmark, Lecture Notes in Computer Science, Vol 670*, pages 216–223. Springer-Verlag, 1993.

[BW95] Michael Barr and Charles Wells. *Category Theory for Computing Science.* International Series in Computer Science. Prentice Hall International, 1995.

[Gol84] Robert Goldblatt. *Topoi: The Categorical Analysis of Logic.* North-Holland, 1984.

[GS93] David Gries and Fred Schneider. *A Logical Approach to Discrete Math.* Texts and Monographs in Computer Science. Springer-Verlag, 1993.

[Hil75] Victor E. Hill. *Groups, Representations and Characters.* Hafner Press, 1975.

[Hug95] Hughes, Arthur P. and Donnelly, Alexis A. An Algebraic Proof in $VDM^\clubsuit$. In Bowen, Jonathan P. and Hinchey, Michael G. [Bow95], pages 114–133.

[Hug96] Hughes, Arthur and Gallagher, Dara and Donnelly, Alexis. On the Inheritance of Indexed Monoid Properties - A Tale of Three Proofs. Technical Report TCD-CS-96-04, Department of Computer Science, Trinity College Dublin, March 1996.

[Hug97] Hughes, Arthur and Donnelly, Alexis. Composition, Underride and Override: The Development of a Useful Map Identity. Technical Report in preparation, Department of Computer Science, Trinity College Dublin, 1997.

[Jac74] Nathan Jacobson. *Basic Algebra.* Freeman, 1974.

[Jon90] Cliff Jones. *Systematic Software Deelopment using VDM.* International Series in Computer Science. Prentice Hall, second edition, 1990.

[Mac90] Mac an Airchinnigh, Mícheál. *Conceptual Models and Computing.* PhD thesis, Department of Computer Science, Trinity College Dublin, 1990.

[Mac91] Mac an Airchinnigh, Mícheál. Tutorial Lecture Notes on the Irish School of the VDM. In Prehn, S and Toetenel, W.J., editor, *VDM'91: Formal Software Development Methods, Volume 2: Tutorials*, pages 141–237. Springer-Verlag, 1991.

[Mac93] Mac an Airchinnigh, Mícheál. Formal Methods and Testing. In *Tutorial Notes:* 6^{th} *International Software Qulity Week.* Software Research Institute, 625 Third Street, San Francisco, C 94107-1997, 1993.

[Spi92] Michael Spivey. *The Z Notation: A Reference Manual.* International Series in Computer Science. Prentice Hall, second edition, 1992.

A Proof of Theorem 1

A.1 Closure

If $\alpha, \beta \in (X \to Y)^{-1}$ is $\alpha \mathbin{\rotatebox[origin=c]{180}{$\dagger$}} \beta \in (X \to Y)^{-1}$?

$$
\begin{array}{ll}
 & \alpha \mathbin{\rotatebox[origin=c]{180}{$\dagger$}} \beta \\
 & \quad < \text{Definition of underride.} > \\
= & (\mathcal{I} \to \lhd_{{}^{\cup}/rng\ \beta})'\alpha \mathbin{\bigcirc\!\!\!\!\!\cup} \beta \\
 & \quad < \text{As } \alpha = \mu^{-1}, \beta = \nu^{-1} \text{ for some } \mu, \nu \in X \to Y. > \\
= & (\mathcal{I} \to \lhd_{{}^{\cup}/rng\ \nu^{-1}})'\mu^{-1} \mathbin{\bigcirc\!\!\!\!\!\cup} \nu^{-1} \\
 & \quad < \text{As } {}^{\cup}/rng\, \nu^{-1} = dom\ \nu. > \\
= & (\mathcal{I} \to \lhd_{dom\ \nu})'\mu^{-1} \mathbin{\bigcirc\!\!\!\!\!\cup} \nu^{-1} \\
 & \quad < \text{As } (\mathcal{I} \to \lhd_{dom\ \nu})'\mu^{-1} = (\lhd_{dom\ \nu}\mu)^{-1}. > \\
= & (\lhd_{dom\ \nu}\mu)^{-1} \mathbin{\bigcirc\!\!\!\!\!\cup} \nu^{-1} \\
 & \quad < \text{Inverse image distributes over extend.} > \\
= & (\lhd_{dom\ \nu}\mu \sqcup \nu)^{-1} \\
 & \quad < \text{Definition of override in terms of extend.} > \\
= & (\mu \dagger \nu)^{-1} \in (X \to Y)^{-1}
\end{array}
$$

A.2 Associativity

If $\alpha, \beta \in (X \to Y)^{-1}$ then the proof of closure show that

$$\alpha \mathbin{\rotatebox[origin=c]{180}{$\dagger$}} \beta = (\mu \dagger \nu)^{-1}$$

for some $\mu, \nu \in X \to Y$ where

$$\alpha = \mu^{-1} \text{ and } \beta = \nu^{-1}.$$

This property is used to show that underride is associative. We wish to show for $\alpha, \beta, \gamma \in (X \to Y)^{-1}$ that

$$(\alpha \mathbin{\rotatebox[origin=c]{180}{$\dagger$}} \beta) \mathbin{\rotatebox[origin=c]{180}{$\dagger$}} \gamma = \alpha \mathbin{\rotatebox[origin=c]{180}{$\dagger$}} (\beta \mathbin{\rotatebox[origin=c]{180}{$\dagger$}} \gamma).$$

$$
\begin{array}{ll}
 & (\alpha \mathbin{\rotatebox[origin=c]{180}{$\dagger$}} \beta) \mathbin{\rotatebox[origin=c]{180}{$\dagger$}} \gamma \\
 & \quad < \text{By property where } \mu, \nu \in X \to Y \text{ such that } \alpha = \mu^{-1} \text{ and } \beta = \nu^{-1}. > \\
= & (\mu \dagger \nu)^{-1} \mathbin{\rotatebox[origin=c]{180}{$\dagger$}} \gamma \\
 & \quad < \text{By above property where } \tau \in X \to Y \text{ such that } \gamma = \tau^{-1}. > \\
= & ((\mu \dagger \nu) \dagger \tau)^{-1} \\
 & \quad < \text{Override is associative.} > \\
= & (\mu \dagger (\nu \dagger \tau))^{-1} \\
 & \quad < \text{By property.} > \\
= & \alpha \mathbin{\rotatebox[origin=c]{180}{$\dagger$}} (\nu \dagger \tau)^{-1} \\
 & \quad < \text{By property.} > \\
= & \alpha \mathbin{\rotatebox[origin=c]{180}{$\dagger$}} (\beta \mathbin{\rotatebox[origin=c]{180}{$\dagger$}} \gamma)
\end{array}
$$

A.3 Identity

If $\alpha \in (X \to Y)^{-1}$ then does $\alpha \mathbin{⸸} \theta = \alpha = \theta \mathbin{⸸} \alpha$?

$$
\begin{array}{rl}
& \alpha \mathbin{⸸} \theta \\
& \quad < \text{By property where } \mu, \theta \in X \to Y \text{ such that } \alpha = \mu^{-1} \text{ and } \theta = \theta^{-1}. > \\
= & (\mu \dagger \theta)^{-1} \\
& \quad < \text{Null map identity for override.} > \\
= & \mu^{-1} = \alpha = \mu^{-1} \\
& \quad < \text{Null map identity for override.} > \\
= & (\theta \dagger \mu)^{-1} \\
& \quad < \text{By property.} > \\
= & \theta \mathbin{⸸} \alpha
\end{array}
$$

A.4 Isomorphism

Is the monoid of maps $(X \to Y, \dagger, \theta)$ isomorphic to the monoid of inverse maps $((X \to Y)^{-1}, \mathbin{⸸}, \theta)$? The inverse image map $\mu \mapsto \mu^{-1}$ is one-to-one and onto, these facts and above property ensure that inverse image distributes over override,

$$(\mu \dagger \nu)^{-1} = \mu^{-1} \mathbin{⸸} \nu^{-1}.$$

B The Indexed Monoid

The theorem and lemmas below provide an alternative definition of relational union. The latter is a common example of an indexed operation in our work

Theorem 3 *Let $(M, *, u)$ denote an arbitrary monoid, with unit u, and $(M', *)$ the corresponding semi-group, i.e., with $M' = ⩤_u M$. Then for a space X, the structure $(X \to M', \circledast, \theta)$ is an indexed monoid which inherits its operator properties from $(M, *, u)$, where for μ in $X \to M'$,*

$$\mu \circledast [x \mapsto m] = \begin{cases} \mu \sqcup [x \mapsto m], & if \neg\chi[\![x]\!]\mu \\ \mu \dagger [x \mapsto \mu(x) * m], & otherwise \end{cases}$$

The monoid of bags $(X \to N', \oplus, \theta)$ is the monoid of natural numbers $(N, +, 0)$ indexed with respect to X. The monoid of relations $(X \to \mathcal{P}'Y, \textcircled{$\cup$}, \theta)$ is the monoid of sets $(\mathcal{P}Y, \cup, \emptyset)$ indexed with respect to X. The monoid of catalogues $(X \to (Y \to Z)', \textcircled{$\dagger$}, \theta)$ is the monoid of maps $(Y \to Z, \dagger, \theta)$ indexed with respect to X. In fact, where the base operation, $*$, is commutative so is the indexed version (see [Hug96]).

Lemma 5 *Let $(M, *, u)$ denote a monoid, with unit u. Then for a space X, let M^X denote the set of all total mappings from X to M, and let u^X denote the constant mapping from X to the unit u. For $f, g \in M^X$ define,*

$$(f * g)(x) = f(x) * g(x)$$

for all $x \in X$, then $(M^X, *, u^X)$ is a monoid, called the X-direct power of M.

This is taken from [Jac74].

The indexed monoid binary operation is redefined using the X-direct power binary operation. For mapping $\mu, \nu \in X \to M'$ define,

$$\mu \circledast \nu = ((u^X \dagger \mu) * (u^X \dagger \nu))'$$

where the priming denotes the removal of entries of the form $x \mapsto u$ and u^X denotes the constant mapping from X to u.

Lemma 6 *For an arbitrary monoid $(M, *, u)$ and a space X, the indexed monoid $(X \to M', \circledast, \theta)$ is isomorphic to the X-direct power monoid $(M^X, *, u^X)$.*

The isomorphism is override on the left by the constant mapping from X to u,

$$u^X \dagger \_ : (X \to M', \circledast, \theta) \longrightarrow (M^X, *, u^X).$$

The inverse map which removes entries over the unit u is an isomorphism also,

$$(\_)' : (M^X, *, u^X) \longrightarrow (X \to M', \circledast, \theta).$$

Using somewhat tedious inductive arguments, we have established the equivalence of the definition given in Section 2 (and repeated at the start of this Appendix) to the X-direct power definition above [Hug96].

C Proof of Lemma 1

C.1 Range Restriction & Union

Lemma 7 *If $S, R \in \mathcal{P}Y$ and $\mu \in X \to Y$, then*

$$\rhd_{(S \cup R)}\, \mu = \rhd_S\, \mu \cup \rhd_R\, \mu,$$

where $\rhd_S\, \mu \cup \rhd_R\, \mu$ denotes the glueing together of the maps $\rhd_S\, \mu$ and $\rhd_R\, \mu$.

Proof

$$\begin{array}{ll}
 & \rhd_{(S \cup R)}\, \mu \\
= & \langle \text{Definition of range restriction.} \rangle \\
 & \lhd_{\mu^{-1}(S \cup R)} \mu \\
= & \langle \text{Inverse image distributes over union.} \rangle \\
 & \lhd_{(\mu^{-1}(S) \cup \mu^{-1}(R))} \mu \\
= & \langle \text{If } S \in \mathcal{P}X \text{ and } \mu \in X \to Y \text{ then define } \lhd_\mu S = \lhd_S \mu. \rangle \\
 & \lhd_\mu (\mu^{-1}(S) \cup \mu^{-1}(R)) \\
= & \langle \text{As restriction w.r.t. a map } \lhd_\mu \text{ is a homomorphism form a monoid of sets} \\
 & \quad \text{under union } (\mathcal{P}X, \cup, \emptyset) \text{ to a monoid of maps under glueing } (\lhd_\mu(\mathcal{P}X), \cup, \theta). \rangle \\
 & \lhd_\mu \mu^{-1}(S) \cup \lhd_\mu \mu^{-1}(R) \\
= & \langle \text{Definition of restriction w.r.t. a map.} \rangle \\
 & \lhd_{\mu^{-1}(S)} \mu \cup \lhd_{\mu^{-1}(R)} \mu \\
= & \langle \text{Definition of range restriction.} \rangle \\
 & \rhd_S\, \mu \cup \rhd_R\, \mu
\end{array}$$

C.2 Relational Union & Composition

Lemma 8 *If $\alpha, \beta \in Y \to \mathcal{P}'Z$ and $\mu \in X \to Y$, then*

$$(\alpha \circledcirc \beta) \circ \mu = (\alpha \circ \mu) \circledcirc (\beta \circ \mu).$$

Proof

$$\begin{array}{ll}
 & (\alpha \circledcirc \beta) \circ \mu \\
= & \langle \text{Definition of composition.} \rangle \\
 & (\mathcal{I} \to (\alpha \circledcirc \beta)) \rhd_{\mathtt{dom}\,(\alpha \circledcirc \beta)}\, \mu \\
= & \langle \text{Domain is a homomorphism from the monoid of relations} \\
 & \quad (X \to \mathcal{P}'Y, \circledcirc, \theta) \text{ to monoid of sets under union } (\mathcal{P}X, \cup, \emptyset). \rangle \\
 & (\mathcal{I} \to (\alpha \circledcirc \beta)) \rhd_{(\mathtt{dom}\,\alpha \cup \mathtt{dom}\,\beta)}\, \mu
\end{array}$$

$$
\begin{array}{cl}
= & \langle \text{Split } \mathtt{dom}\,\alpha \cup \mathtt{dom}\,\beta \text{ into disjoint sets.}\rangle \\
 & (\mathcal{I} \to (\alpha \circledcirc \beta)) \rhd_{((\mathtt{dom}\,\alpha \backslash \mathtt{dom}\,\beta) \uplus (\mathtt{dom}\,\alpha \cap \mathtt{dom}\,\beta) \uplus (\mathtt{dom}\,\beta \backslash \mathtt{dom}\,\alpha))}\ \mu \\
= & \langle \text{Applying lemma 7 where } \uplus \text{denotes the union of two disjoint set.}\rangle \\
 & (\mathcal{I} \to (\alpha \circledcirc \beta))(\rhd_{(\mathtt{dom}\,\alpha \backslash \mathtt{dom}\,\beta)}\ \mu \sqcup \rhd_{(\mathtt{dom}\,\alpha \cap \mathtt{dom}\,\beta)}\ \mu \sqcup \rhd_{(\mathtt{dom}\,\beta \backslash \mathtt{dom}\,\alpha)}\ \mu) \\
= & \langle \text{A map iterator distributes over extend.}\rangle \\
 & (\mathcal{I} \to (\alpha \circledcirc \beta)) \rhd_{(\mathtt{dom}\,\alpha \backslash \mathtt{dom}\,\beta)}\ \mu \sqcup (\mathcal{I} \to (\alpha \circledcirc \beta))) \rhd_{(\mathtt{dom}\,\alpha \cap \mathtt{dom}\,\beta)}\ \mu \\
 & \sqcup (\mathcal{I} \to (\alpha \circledcirc \beta) \rhd_{(\mathtt{dom}\,\beta \backslash \mathtt{dom}\,\alpha)}\ \mu \\
= & \langle \text{As } (\mathtt{dom}\,\alpha \backslash \mathtt{dom}\,\beta) \subset \mathtt{dom}\,\alpha \text{ and } (\mathtt{dom}\,\beta \backslash \mathtt{dom}\,\alpha) \subset \mathtt{dom}\,\beta.\rangle \\
 & (\mathcal{I} \to \alpha) \rhd_{(\mathtt{dom}\,\alpha \backslash \mathtt{dom}\,\beta)}\ \mu \sqcup (\mathcal{I} \to (\alpha \circledcirc \beta)) \rhd_{(\mathtt{dom}\,\alpha \cap \mathtt{dom}\,\beta)}\ \mu \\
 & \sqcup (\mathcal{I} \to \beta) \rhd_{(\mathtt{dom}\,\beta \backslash \mathtt{dom}\,\alpha)}\ \mu \\
= & \langle \text{Applying the map iterator.}\rangle \\
 & (\mathcal{I} \to \alpha) \rhd_{(\mathtt{dom}\,\alpha \backslash \mathtt{dom}\,\beta)}\ \mu \sqcup (\mathcal{I} \to \alpha) \rhd_{(\mathtt{dom}\,\alpha \cap \mathtt{dom}\,\beta)}\ \mu \\
 & \circledcirc (\mathcal{I} \to \beta) \rhd_{(\mathtt{dom}\,\alpha \cap \mathtt{dom}\,\beta)}\ \mu \sqcup (\mathcal{I} \to \beta) \rhd_{(\mathtt{dom}\,\beta \backslash \mathtt{dom}\,\alpha)}\ \mu \\
= & \langle \text{Applying lemma 7.}\rangle \\
 & (\mathcal{I} \to \alpha) \rhd_{((\mathtt{dom}\,\alpha \backslash \mathtt{dom}\,\beta) \uplus (\mathtt{dom}\,\alpha \cap \mathtt{dom}\,\beta))}\ \mu \circledcirc (\mathcal{I} \to \beta) \rhd_{((\mathtt{dom}\,\alpha \cap \mathtt{dom}\,\beta) \uplus (\mathtt{dom}\,\beta \backslash \mathtt{dom}\,\alpha))}\ \mu \\
= & \langle \text{Evaluating the disjoint unions.}\rangle \\
 & (\mathcal{I} \to \alpha) \rhd_{\mathtt{dom}\,\alpha}\ \mu \circledcirc (\mathcal{I} \to \beta) \rhd_{\mathtt{dom}\,\beta}\ \mu \\
= & \langle \text{Definition of composition.}\rangle \\
 & (\alpha \circ \mu) \circledcirc (\beta \circ \mu)
\end{array}
$$

C.3 Underride & Composition

Lemma 1 *If $\alpha, \beta \in (Z \to Y)^{-1}$ and $\mu \in X \to Y$, then*

$$(\alpha \mathbin{\ddagger} \beta) \circ \mu = (\mathcal{I} \to \lhd_{\cup/\mathtt{rng}\,\beta})'(\alpha \circ \mu) \circledcirc (\beta \circ \mu).$$

Proof

$$
\begin{array}{cl}
 & (\alpha \mathbin{\ddagger} \beta) \circ \mu \\
= & \langle \text{Definition of underride.}\rangle \\
 & ((\mathcal{I} \to \lhd_{\cup/\mathtt{rng}\,\beta})'\alpha \circledcirc \beta) \circ \mu \\
= & \langle \text{Applying lemma 8.}\rangle \\
 & ((\mathcal{I} \to \lhd_{\cup/\mathtt{rng}\,\beta})'\alpha \circ \mu) \circledcirc (\beta \circ \alpha) \\
= & \langle \text{As composition is associative.}\rangle \\
 & (\mathcal{I} \to \lhd_{\cup/\mathtt{rng}\,\beta})'(\alpha \circ \mu) \circledcirc (\beta \circ \mu)
\end{array}
$$

D Proof of Lemma 2

D.1 Range Restriction & Override

Lemma 10 *If $S \in \mathcal{P}Y$ and $\alpha, \beta \in X \to Y$, then*

$$\rhd_S (\alpha \dagger \beta) = \lhd_{\mathtt{dom}\,\beta} \rhd_S \alpha \sqcup \rhd_S \beta.$$

Proof

$$
\begin{array}{cl}
 & \rhd_S (\alpha \dagger \beta) \\
= & \langle \text{If } S \in \mathcal{P}Y \text{ and } \alpha \in X \to Y \text{ then } \rhd_S \alpha = \lhd_{\alpha^{-1}(S)} \alpha.\rangle \\
 & \lhd_{(\alpha \dagger \beta)^{-1}(S)} (\alpha \dagger \beta) \\
= & \langle \text{Restriction w.r.t. a set is an endomorphism of the monoid of maps.}\rangle \\
 & \lhd_{(\alpha \dagger \beta)^{-1}(S)} \alpha \dagger \lhd_{(\alpha \dagger \beta)^{-1}(S)} \beta \\
= & \langle \text{If } \alpha, \beta \in X \to Y \text{ and } S \in \mathcal{P}Y \text{ then } (\alpha \dagger \beta)^{-1}(S) = \lhd_{\mathtt{dom}\,\beta} \alpha^{-1}(S) \cup \beta^{-1}(S).\rangle
\end{array}
$$

$$
\begin{array}{ll}
 & \lhd_{(\ntriangleleft_{\mathtt{dom}\,\beta}\alpha^{-1}(S)\cup\beta^{-1}(S))}\alpha \dagger \lhd_{(\ntriangleleft_{\mathtt{dom}\,\beta}\alpha^{-1}(S)\cup\beta^{-1}(S))}\beta \\
= & \langle \text{If } S \in \mathcal{P}X \text{ and } \alpha \in X \to Y \text{ then define } \lhd_{\mu}S = \lhd_{S}\mu. \rangle \\
 & \lhd_{\alpha}(\ntriangleleft_{\mathtt{dom}\,\beta}\alpha^{-1}(S) \cup \beta^{-1}(S)) \dagger \lhd_{\beta}(\ntriangleleft_{\mathtt{dom}\,\beta}\alpha^{-1}(S) \cup \beta^{-1}(S)) \\
= & \langle \text{As restriction w.r.t. a map } \lhd_{\alpha} \text{ is a homomorphism form a monoid of sets} \\
 & \quad \text{under union } (\mathcal{P}X, \cup, \emptyset) \text{ to a monoid of maps under glueing } (\lhd_{\alpha}(\mathcal{P}X), \cup, \theta). \rangle \\
 & (\lhd_{\alpha}\ntriangleleft_{\mathtt{dom}\,\beta}\alpha^{-1}(S) \cup \lhd_{\alpha}\beta^{-1}(S)) \dagger (\lhd_{\beta}\ntriangleleft_{\mathtt{dom}\,\beta}\alpha^{-1}(S) \cup \lhd_{\beta}\beta^{-1}(S)) \\
= & \langle \text{If } S \in \mathcal{P}X \text{ and } \alpha \in X \to Y \text{ then } \lhd_{\alpha}\ntriangleleft_{S} = \ntriangleleft_{S}\lhd_{\alpha}. \rangle \\
 & (\ntriangleleft_{\mathtt{dom}\,\beta}\lhd_{\alpha}\,\alpha^{-1}(S) \cup \lhd_{\alpha}\beta^{-1}(S)) \dagger (\ntriangleleft_{\mathtt{dom}\,\beta}\lhd_{\beta}\,\alpha^{-1}(S) \cup \lhd_{\beta}\beta^{-1}(S)) \\
= & \langle \text{As } \lhd_{\mu}S = \lhd_{S}\mu. \rangle \\
 & (\ntriangleleft_{\mathtt{dom}\,\beta}\lhd_{\alpha^{-1}(S)}\,\alpha \cup \lhd_{\beta^{-1}(S)}\alpha) \dagger (\ntriangleleft_{\mathtt{dom}\,\beta}\lhd_{\alpha^{-1}(S)}\,\beta \cup \lhd_{\beta^{-1}(S)}\beta) \\
= & \langle \text{If } S, R \in \mathcal{P}X \text{ then } \ntriangleleft_{S}\lhd_{R} = \lhd_{R}\ntriangleleft_{S}. \rangle \\
 & (\ntriangleleft_{\mathtt{dom}\,\beta}\lhd_{\alpha^{-1}(S)}\,\alpha \cup \lhd_{\beta^{-1}(S)}\alpha) \dagger (\lhd_{\alpha^{-1}(S)}\ntriangleleft_{\mathtt{dom}\,\beta}\beta \cup \lhd_{\beta^{-1}(S)}\beta) \\
= & \langle \text{As } \ntriangleleft_{\mathtt{dom}\,\beta}\beta = \theta. \rangle \\
 & (\ntriangleleft_{\mathtt{dom}\,\beta}\lhd_{\alpha^{-1}(S)}\,\alpha \cup \lhd_{\beta^{-1}(S)}\alpha) \dagger (\theta \cup \lhd_{\beta^{-1}(S)}\beta) \\
= & \langle \text{Null map is an identity for glueing.} \rangle \\
 & (\ntriangleleft_{\mathtt{dom}\,\beta}\lhd_{\alpha^{-1}(S)}\,\alpha \cup \lhd_{\beta^{-1}(S)}\alpha) \dagger \lhd_{\beta^{-1}(S)}\beta \\
= & \langle \text{If } S \in \mathcal{P}X \text{ and } \alpha, \beta \in \lhd_{S}(X \to Y) \text{ also if } \mu \in X \to Y \\
 & \quad \text{then } (\alpha \cup \beta) \dagger \mu = (\alpha \dagger \mu) \cup (\beta \dagger \mu). \rangle \\
 & (\ntriangleleft_{\mathtt{dom}\,\beta}\lhd_{\alpha^{-1}(S)}\,\alpha \dagger \lhd_{\beta^{-1}(S)}\beta) \cup (\lhd_{\beta^{-1}(S)}\alpha \dagger \lhd_{\beta^{-1}(S)}\beta) \\
= & \langle \text{If } \alpha, \beta \in X \to Y \text{ and } S \subset \mathtt{dom}\,\beta \text{ then } \lhd_{S}\alpha \dagger \lhd_{S}\beta = \lhd_{S}\beta. \rangle \\
 & (\ntriangleleft_{\mathtt{dom}\,\beta}\lhd_{\alpha^{-1}(S)}\,\alpha \dagger \lhd_{\beta^{-1}(S)}\beta) \cup \lhd_{\beta^{-1}(S)}\beta \\
= & \langle \text{If } \mu \in X \to Y \text{ then } \mu \cup \mu = \mu. \rangle \\
 & \ntriangleleft_{\mathtt{dom}\,\beta}\lhd_{\alpha^{-1}(S)}\,\alpha \dagger \lhd_{\beta^{-1}(S)}\beta \\
= & \langle \text{As } \rhd_{S}\,\alpha = \lhd_{\alpha^{-1}(S)}\alpha. \rangle \\
 & \ntriangleleft_{\mathtt{dom}\,\beta}\rhd_{S}\,\alpha \dagger \rhd_{S}\,\beta
\end{array}
$$

D.2 Composition & Override

Lemma 2 *If* $\alpha, \beta \in X \to Y$ *and* $\mu \in X \to Y$*, then*

$$\mu \circ (\alpha \dagger \beta) = \ntriangleleft_{\mathtt{dom}\,\beta}(\mu \circ \alpha) \sqcup (\mu \circ \beta).$$

Proof

$$
\begin{array}{ll}
 & \mu \circ (\alpha \dagger \beta) \\
= & \langle \text{Definition of composition.} \rangle \\
 & (\mathcal{I} \to \mu)\rhd_{\mathtt{dom}\,\mu}(\alpha \dagger \beta) \\
= & \langle \text{Applying lemma 10.} \rangle \\
 & (\mathcal{I} \to \mu)(\ntriangleleft_{\mathtt{dom}\,\beta}\rhd_{\mathtt{dom}\,\mu}\alpha \sqcup \rhd_{\mathtt{dom}\,\mu}\beta) \\
= & \langle \text{As a map iterator distributes over extend.} \rangle \\
 & (\mathcal{I} \to \mu)\ntriangleleft_{\mathtt{dom}\,\beta}\rhd_{\mathtt{dom}\,\mu}\alpha \sqcup (\mathcal{I} \to \mu)\rhd_{\mathtt{dom}\,\mu}\beta \\
= & \langle \text{As a map iterator commutes with a domain removal.} \rangle \\
 & \ntriangleleft_{\mathtt{dom}\,\beta}(\mathcal{I} \to \mu)\rhd_{\mathtt{dom}\,\mu}\alpha \sqcup (\mathcal{I} \to \mu)\rhd_{\mathtt{dom}\,\mu}\beta \\
= & \langle \text{Definition of composition.} \rangle \\
 & \ntriangleleft_{\mathtt{dom}\,\beta}(\mu \circ \alpha) \sqcup (\mu \circ \beta)
\end{array}
$$