

An Institution for UML 2.0 Static Structures

María Victoria Cengarle¹ and Alexander Knapp²

¹ Technische Universität München
cengarle@in.tum.de

² Universität Augsburg
knapp@informatik.uni-augsburg.de

Abstract. This work presents the theory of UML 2.0 static structures (or class diagrams), that is proven to define an institution.

1 Introduction

The present work is devoted to the language of UML 2.0 static structures. We develop an abstract syntax and a formal semantics, and prove this definition to be an institution [1].

Preliminaries

A *class hierarchy* is a partial order $\mathbf{C} = (C, \leq_C)$ with a set of class names C and a subclass relation $\leq_C \subseteq C \times C$. By $\mathbf{T}(\mathbf{C})$ we denote the *type extension* of \mathbf{C} by primitive types and type constructors. $\mathbf{T}(\mathbf{C})$ is likewise a class hierarchy $(T(C), \leq_{T(C)})$ with $C \subseteq T(C)$ and $\leq_C \subseteq \leq_{T(C)}$.

Given a class hierarchy $\mathbf{C} = (C, \leq_C)$, a *C-object domain* is a C -indexed family $\mathbf{O} = (O_c)_{c \in C}$ of sets with $O_c \subseteq O_{c'}$ if $c \leq_C c'$. Given moreover a type extension \mathbf{T} , the *value extension* of a \mathbf{C} -object domain $\mathbf{O} = (O_c)_{c \in C}$ by primitive values and value constructions, which is denoted by $V_C^T(\mathbf{O})$, is a $\mathbf{T}(\mathbf{C})$ -object domain $(V_c)_{c \in T(C)}$ such that $V_c = O_c$ for all $c \in C$.

These definitions can be lifted to categories, where \mathbf{T} is a functor from class hierarchies to class hierarchies, V_C^T is a functor from object domains to object domains, and V^T is a natural transformation. Moreover, class hierarchies, type extensions, object domains, and value extensions can be formalized as a Grothendieck construction [2] and also as a monad [3]; see the appendices.

2 Syntax

A *multiplicity* is a pair $m \in \mathbb{N}_0 \times (\mathbb{N}_0 \cup \{\star\})$, such that if $m = (n_1, n_2)$ and $n_2 \neq \star$ then $n_1 < n_2$. We let M denote the set of multiplicities.³

A *CL-signature* $\Sigma = (\mathbf{C}, \mathbf{P})$ declares

³ This definition is enough to cover all UML multiplicities. Indeed, a multiplicity $n \in \mathbb{N}_0$ is equivalent to $n . . n$, and a multiplicity \star is equivalent to $0 . . \star$.

1. a finite class hierarchy $\mathbf{C} = (C, \leq_C)$, and
2. an association declaration $\mathbf{P} = (R, P)$ where R is a finite set of role names and P is a finite set $(p_w)_{w \in (R \times T(C))^{++}}$ of relation names indexed over pairs of a role name and a class name of the extended class hierarchy,⁴ such that for any class name $c \in C$, the role names of the associations in which any $c' \leq_C c$ is involved are all different.⁵

If $p_w \in P$ with $w = (r_1, c_1) \cdots (r_n, c_n)$, we write $p(r_1 : c_1, \dots, r_n : c_n) \in P$.

Intuitively, a CL-signature represents a class diagram. Therein, classes with attributes are declared, as well as inheritance and association relations among classes. Structural features (and also queries) are captured by role names of associations. Declaration of methods has been disregarded, since within class diagrams methods are not given any meaning.

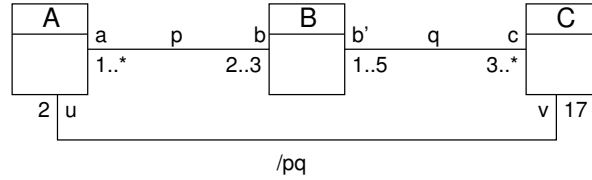


Fig. 1. A simple class diagram

Given a CL-signature $\Sigma = (\mathbf{C}, \mathbf{P})$, the Σ -formulas state the multiplicities associated with a relation name. In order to express multiplicities, the selection/partition operator \bullet , the equijoin/projection operator $*$, role name renaming $[_/_]$ and cardinality $\#$ are used. So for instance the associations declared in Fig. 1 are $p(a : A, b : B)$, $q(b' : B, c : C)$ and $pq(a : A, c : C)$, the last one derived from the former two ones. The derived association and the multiplicities associated with p , q and pq are expressed by the following formulas:

$$\begin{aligned}
2 &\leq \#a \bullet p \leq 3 \\
1 &\leq \#b \bullet p \\
3 &\leq \#b' \bullet q \\
1 &\leq \#c \bullet q \leq 5 \\
pq &= (p[b'/b] * q)[u/a, v/c] \\
\#a \bullet pq &= 17 \\
\#c \bullet pq &= 2
\end{aligned} \tag{1}$$

In the first formula, the relation p is first partitioned, grouping its pairs according to their a -value. A set of sets is then obtained, and to all of these sets the cardinality operator

⁴ The relation names represent the associations of the class diagram. Relations are at least binary, and by A^{++} we abbreviate $A \cdot A^+$.

⁵ Formally, if $p(r_1 : c_1, \dots, r_m : c_m)$ and $q(s_1 : d_1, \dots, s_n : d_n)$ are relations in P and $c_k = d_l \in C$, then $r_i \neq s_j$ for any $i \neq k$ and for any $j \neq l$ ($1 \leq i \leq m, 1 \leq j \leq n$).

is (extensionally) applied. The values thus obtained must be within the stated bounds. The second formula is similar, only that the pairs of p are grouped according to their b -value. The third and fourth formulas are similarly read.

The formula (1) renames first a role name of relation p , equijoins the obtained relation with q , and projects away the repeated value. That is, the tuples (x, y) in p and (y, z) in q are arranged to (x, y, z) . Finally the role names are adjusted.⁶ The two formulas following are read as explained above.⁷ Notice that the derived multiplicities for pq are $(1, \star)$ on the A side, which is declared to be 2, and $(6, \star)$ on the C side, which is declared to be 17. That is, the declared multiplicities do not contradict the derived ones. Hence, the diagram admits a model.

For a multiplicity constraint the selection/partition operator \bullet can be used, in which case $(n - 1)$ role names are required if the association is n -ary. Formally, the set of well-formed formulas induced by a CL-signature is defined as follows. Let $\Sigma = (\mathbf{C}, \mathbf{P})$ be a CL-signature with $\mathbf{C} = (C, \leq_C)$ and $\mathbf{P} = (R, P)$. The set Φ of Σ -formulas is defined by

$$\begin{aligned} T &::= P \mid T * T \mid T[R/R] \mid (T) \\ \Pi &::= R^+ \bullet T \\ \Phi &::= T = T \\ &\quad \mid \#T = n \mid n \leq \#T \mid \#T \leq n \\ &\quad \mid \#\Pi = n \mid n \leq \#\Pi \mid \#\Pi \leq n \end{aligned}$$

where $n \in \mathbb{N}$. Additionally, some context conditions apply. For instance, there must be exactly one pair rolename/classname coincidence in left and right operator of an equijoin/projection $*$. The words in T are called *derived relations*. A Σ -theory presentation is a finite set of Σ -formulas.

Let $\Sigma_1 = (\mathbf{C}_1, \mathbf{P}_1)$ and $\Sigma_2 = (\mathbf{C}_2, \mathbf{P}_2)$ be CL-signatures with $\mathbf{C}_i = (C_i, \leq_{C_i})$ and $\mathbf{P}_i = (R_i, P_i)$ ($i = 1, 2$). A CL-morphism $\sigma : \Sigma_1 \rightarrow \Sigma_2$ is a triple of maps $\langle \sigma_C, \sigma_R, \sigma_P \rangle$ between class names, role names, and predicate names, such that the following conditions hold:

1. $a, b \in C_1$ with $a \leq_{C_1} b$ implies $\sigma_C(a) \leq_{C_2} \sigma_C(b)$,
2. $p_w \in P_1$ implies $\sigma_P(p)_{\sigma(w)} \in P_2$,

⁶ For greater arities, the second occurrence of the repeated value is removed from the tuple. For instance, if p has arity 5 and q has arity 7, if the third component of p and the sixth of q are to be equijoined, then from tuples $(x_1, x_2, x_3, x_4, x_5)$ and $(y_1, y_2, y_3, y_4, y_5, x_3, y_7)$ the tuple $(x_1, x_2, x_3, x_4, x_5, y_1, y_2, y_3, y_4, y_5, y_7)$ is constructed for the equijoin/projection result relation.

⁷ The selection/partition operator corresponds to the clause GROUP BY of SQL; in combination with cardinality, it can be expressed using cardinality and the relational algebra operator σ_φ of selection: $2 \leq \#a \bullet p \leq 3$ is equivalent to $(\forall x \in A)(2 \leq \#\sigma_{a=x}(p) \leq 3)$ The equijoin/projection operator can be expressed using the relational algebra operators of selection, π_c of projection, \times of Cartesian product, and $\rho_{n/o}$ of renaming: $pq = (p[b'/b] * q)[u/a, v/c]$ is equivalent to $\rho_{u/a}(\rho_{v/c}(\pi_{abc}(\sigma_{b=b'}(p \times q))))$. In order to express the multiplicity constraints, however, we do not need the whole battery of possibilities as offered by the relational algebra, but just the above operators.

where the extension of σ_C to $T(C)$ leaves built-in types unchanged,⁸ $\sigma_C(w)$ is the canonical extension of σ_C to words in $T(C)^+$, and σ is the canonical extension of σ_C and σ_R to words in $(R \times T(C))^{++}$.

The extension of CL-morphisms to formulas is canonical.

It is easy to show that the composition of CL-morphisms is a CL-morphism, that composition is associative, and identities are CL-morphisms. Thus, CL-signatures and CL-morphisms define a category which we denote by CL.

3 Semantics

Given a CL-signature $\Sigma = (\mathbf{C}, \mathbf{P})$ with $\mathbf{C} = (C, \leq_C)$ and $\mathbf{P} = (R, P)$, a Σ -interpretation \mathcal{I} consists of a pair (\mathbf{O}, \mathbf{A}) where $\mathbf{O} = (O_c)_{c \in C}$ is a \mathbf{C} -object domain and \mathbf{A} contains a relation $p^{\mathcal{I}} \subseteq O_{c_1} \times \dots \times O_{c_n}$ for each relation name $p(r_1 : c_1, \dots, r_n : c_n) \in P$. The role names induce set valued functions, i.e., if $p(r_1 : c_1, \dots, r_n : c_n) \in P$, then the $(n - 1)$ -ary function $r_i^{\mathcal{I}}$ ($1 \leq i \leq n$) is defined by $r_i^{\mathcal{I}}(a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n) = \{a \in O_{c_i} \mid (a_1, \dots, a_{i-1}, a, a_{i+1}, \dots, a_n) \in p^{\mathcal{I}}\}$.⁹

Signatures do not declare methods (or operations), thus no change of the state of an object may take place, and the semantics of a signature needs not define either a notion of state or a notion of state transition.

Given a CL-signature $\Sigma = (\mathbf{C}, \mathbf{P})$ with $\mathbf{P} = (R, P)$, given a Σ -interpretation $\mathcal{I} = (\mathbf{O}, \mathbf{A})$, the interpretation evaluates derived relations as follows:

1. if $p(r_1 : c_1, \dots, r_m : c_m)$ and $q(s_1 : d_1, \dots, s_n : d_n)$ with $r_k = s_l$, $c_k = d_l$ and $r_i \neq s_j$ for any $i \neq k$ and any $j \neq l$ ($1 \leq i \leq m$, $1 \leq j \leq n$), then $(p * q)(r_1 : c_1, \dots, r_k : c_k, \dots, r_m : c_m, s_1 : d_1, \dots, s_{l-1} : d_{l-1}, s_{l+1} : d_{l+1}, \dots, s_n : d_n)$ is derived and

$$(p * q)^{\mathcal{I}} = \{ (o_1, \dots, o_k, \dots, o_m, o'_1, \dots, o'_{l-1}, o'_{l+1}, \dots, o'_n) \mid (o_1, \dots, o_k, \dots, o_m) \in p^{\mathcal{I}} \wedge (o'_1, \dots, o'_{l-1}, o_k, o'_{l+1}, \dots, o'_n) \in q^{\mathcal{I}} \}$$
2. in case of renaming, only the associated set valued function is differently denoted
3. if $p(r_1 : c_1, \dots, r_m : c_m)$ and $\{i_1, \dots, i_k\} \subseteq \{1, \dots, m\}$, then

$$(r_{i_1} \dots r_{i_k} \bullet p)^{\mathcal{I}} = \{ \{t \in p^{\mathcal{I}} \mid \pi_{i_1}(t) = o_1 \wedge \dots \wedge \pi_{i_k}(t) = o_k\} \mid o_1 \in O_{c_{i_1}}, \dots, o_k \in O_{c_{i_k}} \}$$

Given moreover a Σ -formula φ , then \mathcal{I} satisfies φ if one of the following conditions holds:

1. φ is $T_1 = T_2$ and $T_1^{\mathcal{I}} = T_2^{\mathcal{I}}$
2. φ is $\#T = n$ and $\#T^{\mathcal{I}} = n$
3. φ is $n \leq \#T$ and $n \leq \#T^{\mathcal{I}}$
4. φ is $\#T \leq n$ and $\#T^{\mathcal{I}} \leq n$

⁸ This actually is the case if \mathbf{T} is a functor; see the appendices.

⁹ Role names can be used in more than one association declaration, provided ambiguities are avoided, i.e., no role name is used more than once in the associations in which one and the same class name, or any of its subclasses, is involved. Therefore, a role name may induce more than one set valued function; these functions have different domains because of that proviso, and thus no confusion may arise from homonymy.

5. φ is $\#II = n$ and $\#S = n$ for all $S \in II^{\mathcal{I}}$
6. φ is $n \leq \#II$ and $n \leq \#S$ for all $S \in II^{\mathcal{I}}$
7. φ is $\#II \leq n$ and $\#S \leq n$ for all $S \in II^{\mathcal{I}}$

If the interpretation \mathcal{I} satisfies the formula φ , we write $\mathcal{I} \models \varphi$.¹⁰

Given a CL-signature $\Sigma = (\mathbf{C}, \mathbf{P})$ with $\mathbf{C} = (C, \leq_C)$, given Σ -interpretations $\mathcal{I} = (\mathbf{O}, \mathbf{A})$ and $\mathcal{I}' = (\mathbf{O}', \mathbf{A}')$ with $\mathbf{O} = (O_c)_{c \in C}$ and $\mathbf{O}' = (O'_c)_{c \in C}$, a Σ -homomorphism $h : \mathcal{I} \rightarrow \mathcal{I}'$ is a family of maps $(h_c)_{c \in C}$ with $h_c : O_c \rightarrow O'_c$ such that $(v_1, \dots, v_n) \in p^{\mathcal{I}}$ iff $(h_{c_1}(v_1), \dots, h_{c_n}(v_n)) \in p^{\mathcal{I}'}$ for any $v_i \in O_c$ ($i = 1, \dots, n$) for any $p(r_1 : c_1, \dots, r_n : c_n) \in P$.

It is not difficult to show the existence of homomorphic identities, that composition of Σ -homomorphisms is a Σ -homomorphism, and that composition is associative. For any CL-signature Σ , thus, Σ -interpretations and Σ -homomorphisms define a category which we denote by $\mathbb{I}(\Sigma)$.

Given CL-signatures $\Sigma_1 = (\mathbf{C}_1, \mathbf{P}_1)$ and $\Sigma_2 = (\mathbf{C}_2, \mathbf{P}_2)$ with $\mathbf{C}_i = (C_i, \leq_{C_i})$ and $\mathbf{P}_i = (R_i, P_i)$ ($i = 1, 2$), given a CL-morphism $\sigma : \Sigma_1 \rightarrow \Sigma_2$, and given a Σ_2 -interpretation $\mathcal{I}_2 = (\mathbf{O}_2, \mathbf{A}_2)$ with $\mathbf{O}_2 = (O'_c)_{c \in C_2}$ and $\mathbf{A}_2 = \{p^{\mathcal{I}_2} \mid p \in P_2\}$, the reduct $\mathcal{I}_2|_{\sigma}$ of \mathcal{I}_2 along σ is the Σ_1 -interpretation $(\mathbf{O}_1, \mathbf{A}_1)$ with $\mathbf{O}_1 = (O_{\sigma(c)})_{c \in C_1}$ and $\mathbf{A}_1 = \{\sigma_P(p)^{\mathcal{I}_2} \mid p \in P_1\}$.

Let $\Sigma_1 = (\mathbf{C}_1, \mathbf{P}_1)$ and $\Sigma_2 = (\mathbf{C}_2, \mathbf{P}_2)$ be CL-signatures with $\mathbf{C}_i = (C_i, \leq_{C_i})$ ($i = 1, 2$) and $\mathbf{P}_i = (R_i, P_i)$ ($i = 1, 2$), let $\sigma : \Sigma_1 \rightarrow \Sigma_2$ be a CL-morphism. Let $\mathcal{I}_2 = (\mathbf{O}_2, \mathbf{A}_2)$ and $\mathcal{I}'_2 = (\mathbf{O}'_2, \mathbf{A}'_2)$ be Σ_2 -interpretations with $\mathbf{O}_2 = (O_c)_{c \in C_2}$ and $\mathbf{O}'_2 = (O'_c)_{c \in C_2}$. Let moreover \mathcal{I}_1 denote $\mathcal{I}_2|_{\sigma}$ and \mathcal{I}'_1 denote $\mathcal{I}'_2|_{\sigma}$. Given a Σ_2 -homomorphism $h_2 : \mathcal{I}_2 \rightarrow \mathcal{I}'_2$ the reduct $h_2|_{\sigma}$ of h_2 along σ is the Σ_1 -homomorphism $h_1 : \mathcal{I}_1 \rightarrow \mathcal{I}'_1$ defined by $h_{1c}(v) = h_{2\sigma(c)}(v)$ for any $c \in C_1$, for any $v \in O_c$. It is trivial to check that h_1 is, indeed, a Σ_1 -homomorphism.

Given CL-signatures Σ_1 and Σ_2 , a CL-morphism $\sigma : \Sigma_1 \rightarrow \Sigma_2$ defines a functor from $\mathbb{I}(\Sigma_1)$ to $\mathbb{I}(\Sigma_2)$. Indeed, by definition domain and codomain of the reduct of an homomorphism are the reduct of domain and codomain, respectively, of the homomorphism, and it is routine to show that the reduct of a composition of two homomorphisms is the composition of the reducts of those homomorphisms and that the reduct of an identity homomorphism is likewise an identity.

Moreover, given a Σ_2 -interpretation \mathcal{I}_2 and a Σ_1 -formula φ , the *satisfaction condition* holds, i.e., $\mathcal{I}_2 \models \sigma(\varphi)$ iff $\mathcal{I}_2|_{\sigma} \models \varphi$. It suffices to show $T^{\mathcal{I}_2|_{\sigma}} = \sigma(T)^{\mathcal{I}_2}$ and $II^{\mathcal{I}_2|_{\sigma}} = \sigma(II)^{\mathcal{I}_2}$, straightforwardly checked by induction on the structure of T and II ; the base case where T is $p(c_1, \dots, c_n)$ verifies the thesis by definition of reduct.

Therefore, signatures, interpretations, reducts, formulas and the satisfaction relation define an institution [1].

4 Conclusions

This is a usual setting as in order-sorted algebras. The role names induce functions, which can be safely disregarded here, since they have no influence whatsoever on the

¹⁰ This interpretation of associations makes useless the “property string” *unique*, since by default an association end represents a set. In the presence of other properties like *sequence*, *ordered* and *nonunique*, the condition to be fulfilled by the interpretation has to be revised.

concern of this work, namely the institution of classes and in particular the satisfaction condition. Moreover, the sketched proofs are standard.

The institutional framework should be revised and correspondingly adapted if properties like *nonunique*, *sequence* or *ordered* can possibly be attached to a role name.

References

1. Goguen, J.A., Burstall, R.M.: Introducing Institutions. In Clarke, E.M., Kozen, D., eds.: Logic of Programs. Volume 164 of Lecture Notes in Computer Science., Springer (1983) 221–256
2. Grothendieck, A.: Catégories Fibrées et Descente. In: Revêtements Etales et Groupe Fondamental. Volume 224 of Lecture Notes in Mathematics., Springer (1971) 145–194
3. Lane, S.M.: Categories for the Working Mathematician. Springer-Verlag, New York (1998)
4. Tarlecki, A., Burstall, R.M., Goguen, J.A.: Some Fundamental Algebraic Tools for the Semantics of Computation – Part 3: Indexed Categories. Theoretical Computer Science **91**(2) (1991) 239–264

A Grothendieck construction

Let $\mathcal{C} : I^{\text{op}} \rightarrow \text{Cat}$ be a functor, where Cat is the category of small categories and functors, and I is a small category. The associated *Grothendieck construction* (see [2,4]) is a category $\mathcal{G}(\mathcal{C})$ with objects $\langle i, x \rangle$ where i is an object in I and x an object in $\mathcal{C}(i)$, and with arrows $\langle \sigma, \alpha \rangle : \langle i, x \rangle \rightarrow \langle j, y \rangle$ where $\sigma : i \rightarrow j$ is an arrow in \mathcal{C} and $\alpha : x \rightarrow \mathcal{C}(\sigma)(y)$ an arrow in $\mathcal{C}(i)$. Composition of arrows in $\mathcal{G}(\mathcal{C})$ is defined by $\langle \sigma, \alpha \rangle \circ \langle \sigma', \alpha' \rangle = \langle \sigma \circ \sigma', \mathcal{C}(\sigma)(\alpha') \circ \alpha \rangle$.

The functor $\mathcal{C} : I^{\text{op}} \rightarrow \text{Cat}$ can also be regarded as a category with objects $\mathcal{C}(i)$ and arrows $\mathcal{C}(\sigma) : \mathcal{C}(j) \rightarrow \mathcal{C}(i)$ if i, j are objects in I and $\sigma : i \rightarrow j$ an arrow in I . $\mathcal{C} : I^{\text{op}} \rightarrow \text{Cat}$ is called the *indexed category* over I .

The above syntax for static structures can be formalized as a Grothendieck construction. Class hierarchies are partially ordered sets or posets. Class hierarchies can thus be regarded as objects of the category Poset , whose arrows are monotone functions. We define an interpretation functor $\text{CStr} : \text{Poset}^{\text{op}} \rightarrow \text{Cat}$ that maps class names to sets of object identifiers such that the subclass relation is interpreted as the subset relation. CStr moreover maps monotone functions to the reduct. More precisely, given a class hierarchy $\mathbf{C} = (C, \leq_C)$ with $\leq_C \subseteq C \times C$, $\text{CStr}(\mathbf{C})$ is the category whose objects are \mathbf{C} -object domains and whose arrows $h : \mathbf{O}^1 \rightarrow \mathbf{O}^2$ for $\mathbf{O}^1 = (O_c^1)_{c \in C}$ and $\mathbf{O}^2 = (O_c^2)_{c \in C}$ consist of individual functions $h_c : O_c^1 \rightarrow O_c^2$ for each $c \in C$ and are called \mathbf{C} -homomorphisms; arrow composition and identity arrows are straightforwardly defined. Furthermore, given an arrow $\sigma : \mathbf{C} \rightarrow \mathbf{C}'$ in Poset with $\mathbf{C}' = (C', \leq_{C'})$, the functor $\text{CStr}(\sigma) : \text{CStr}(\mathbf{C}') \rightarrow \text{CStr}(\mathbf{C})$ maps each \mathbf{C}' -object domain $\mathbf{O}' = (O_{c'}^1)_{c' \in C'}$ to the \mathbf{C} -object domain $\mathbf{O}'|_{\sigma} = (O'_{\sigma(c)})_{c \in C}$, and maps each \mathbf{C}' -homomorphism $h = (h_{c'})_{c' \in C'}$ between \mathbf{C}' -object domains \mathbf{O}'_1 and \mathbf{O}'_2 to the \mathbf{C} -homomorphism $h|_{\sigma} = (h_{\sigma(c)})_{c \in C}$ between \mathbf{C} -object domains $\mathbf{O}'_1|_{\sigma}$ and $\mathbf{O}'_2|_{\sigma}$.

The corresponding Grothendieck construction $\mathcal{G}(\text{CStr})$ is a category whose objects are pairs (\mathbf{C}, \mathbf{O}) with \mathbf{C} a class hierarchy and \mathbf{O} a \mathbf{C} -object domain, and whose arrows

are $\langle \sigma, h \rangle : (\mathbf{C}, \mathbf{O}) \rightarrow (\mathbf{C}', \mathbf{O}')$ with $\sigma : \mathbf{C} \rightarrow \mathbf{C}'$ an arrow in Poset and $h : \mathbf{O} \rightarrow \text{CStr}(\sigma)(\mathbf{O}')$ a \mathbf{C} -homomorphism.

Let $\mathcal{C} : I^{\text{op}} \rightarrow \text{Cat}$ be an indexed category with objects $\mathcal{C}(i)$ for i an object of I and arrows $\mathcal{C}(\sigma) : \mathcal{C}(j) \rightarrow \mathcal{C}(i)$ for $\sigma : i \rightarrow j$ an arrow of I . Let $\mathcal{D} : I^{\text{op}} \rightarrow \text{Cat}$ be another indexed category over the same index category. Let $F : \mathcal{C} \rightarrow \mathcal{D}$ be an indexed functor (i.e., a natural transformation from \mathcal{C} to \mathcal{D}). Naturality of F means that $\mathcal{D}(\sigma) \circ F_j = F_i \circ \mathcal{C}(\sigma)$ for each $\sigma : i \rightarrow j$ in I (diagrammatically shown in Fig. 2).

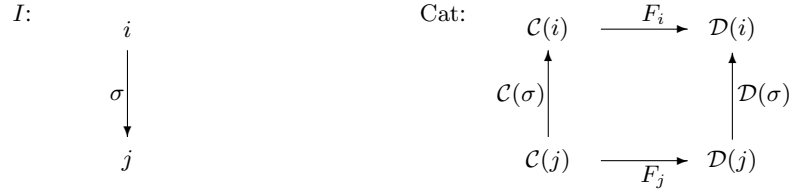


Fig. 2. Naturality diagrams

The indexed functor F induces a flattened functor $\mathcal{G}(F)$ from the flattened category $\mathcal{G}(\mathcal{C})$ to the flattened category $\mathcal{G}(\mathcal{D})$ defined, on objects $\langle i, x \rangle$, by $\mathcal{G}(F)(\langle i, x \rangle) = \langle i, F_i(x) \rangle$ and, on arrows $\langle \sigma, \alpha \rangle : \langle i, x \rangle \rightarrow \langle j, y \rangle$, by $\mathcal{G}(F)(\langle \sigma, \alpha \rangle) = \langle \sigma, F_i(\alpha) \rangle : \langle i, F_i(x) \rangle \rightarrow \langle j, F_j(y) \rangle$.

On the one hand, CStr is an indexed category over Poset and $\mathcal{G}(\text{CStr})$ is the corresponding flattened category; see [4]. On the other, $T : \text{Poset} \rightarrow \text{Poset}$ is a functor and, moreover, $V_C^T : \text{CStr}(\mathbf{C}) \rightarrow \text{CStr}(T(\mathbf{C}))$ is a functor for any class hierarchy \mathbf{C} , i.e., $V^T : \text{CStr} \rightarrow \text{CStr} \circ T$ is a natural transformation. Notice that $\text{CStr} \circ T$ is likewise an indexed category over Poset.

For the category Poset as index, CStr as the indexed category \mathcal{C} , $\text{CStr} \circ T$ as the indexed category \mathcal{D} , and V^T as the natural transformation F , the compatibility of V^T and T is given by the naturality of V^T . Indeed, let $\mathbf{C} = (\mathbf{C}, \leq_{\mathbf{C}})$ and $\mathbf{C}' = (\mathbf{C}', \leq_{\mathbf{C}'})$ be class hierarchies, i.e., objects in Poset. Let moreover $\sigma : \mathbf{C} \rightarrow \mathbf{C}'$ be a monotone function, i.e., an arrow in Poset. Finally, let \mathbf{O}' be an object domain for \mathbf{C}' . Thus, on the one hand, $\text{CStr}(\sigma)(\mathbf{O}')$ is an object domain for \mathbf{C} , and, on the other, $V_{\mathbf{C}'}^T(\mathbf{O}')$ is an object domain for $T(\mathbf{C}')$. The functor T and the natural transformation V^T satisfy the following property:

$$V_{\mathbf{C}}^T(\text{CStr}(\sigma)(\mathbf{O}')) = \text{CStr}(T(\sigma))(V_{\mathbf{C}'}^T(\mathbf{O}'))$$

or, in words, the extension of the reduct and the reduct of the extension coincide. This is due to the fact that V^T is a natural transformation; see Fig. 2 with \mathbf{C} for i , \mathbf{C}' for j , CStr for \mathcal{C} , $\text{CStr} \circ T$ for \mathcal{D} , and V^T for F .

Informally, the flattened version of the functor V_C^T transforms a pair $\langle \mathbf{C}, \mathbf{O} \rangle$ of a class hierarchy \mathbf{C} and a \mathbf{C} -object domain into a pair $\langle \mathbf{C}, V_{\mathbf{C}}^T(\mathbf{O}) \rangle$ of the same class hierarchy \mathbf{C} and the $T(\mathbf{C})$ -object domain $V_{\mathbf{C}}^T(\mathbf{O})$ that extends the \mathbf{C} -object domain \mathbf{O} according to the type extension T .

B Monad construction

We assume a monad on Poset consisting of the functor $T : \text{Poset} \rightarrow \text{Poset}$ and two natural transformations $\eta : 1_{\text{Poset}} \rightarrow T$, where 1_{Poset} denotes the identity functor on Poset , and $\mu : T^2 \rightarrow T$, where T^2 is the functor $T \circ T$ from Poset to Poset . That is, given a class hierarchy (a poset) C ,

$$\begin{aligned} \mu_C &: T(T(C)) \rightarrow T(C), \text{ and} \\ \eta_C &: C \rightarrow T(C) \text{ since } 1_{\text{Poset}}(C) = C \end{aligned}$$

are arrows in Poset , i.e., monotone functions between class hierarchies.

Example 1: T is “set”

Let B be a set of *basic* types. For any class hierarchy $C = (C, \leq_C)$ in Poset , we define the class hierarchy $B(C) = (B(C), \leq_{B(C)})$ by $B(C) = C \uplus B$ and $\leq_{B(C)} = \leq_C \cup \text{id}_B$.¹¹ For any monotone function $\sigma : C \rightarrow C'$ in Poset with $C = (C, \leq_C)$ and $C' = (C', \leq_{C'})$, we define the function $B(\sigma) : B(C) \rightarrow B(C')$ by $B(\sigma)(c) = \sigma(c)$ if $c \in C$ and $B(\sigma)(b) = b$ if $b \in B$. $B(\sigma)$ obviously is monotone, and thus B defines a functor from Poset to Poset . We denote by BPoset the category with objects $B(C)$ and arrows $B(\sigma)$, where C is an object and σ an arrow in Poset .

For any class hierarchy $C = (C, \leq_C)$ in BPoset , we define the class hierarchy $T(C) = (T(C), \leq_{T(C)})$ by $T(C) = C \cup \{\text{set}(c) : c \in C\}$ and $\leq_{T(C)} = \leq_C \cup \{(\text{set}(c_1), \text{set}(c_2)) : c_1 \leq_C c_2\}$. For any monotone function $\sigma : C \rightarrow C'$ in Poset with $C = (C, \leq_C)$ and $C' = (C', \leq_{C'})$, we define the function $T(\sigma) : T(C) \rightarrow T(C')$ by $T(\sigma)(c) = \sigma(c)$ and $T(\sigma)(\text{set}(c)) = \text{set}(\sigma(c))$ for any $c \in C$, which evidently is monotone. Therefore, T defines a functor from BPoset to BPoset .

For any class hierarchy $C = (C, \leq_C)$ in BPoset , we define the following two arrows in Poset (i.e., monotone functions):

- $\text{embed}_C : C \rightarrow T(C)$
 $\text{embed}_C(c) = c$
- $\text{flatten}_C : T(T(C)) \rightarrow T(C)$
 $\text{flatten}_C(t) = t$ if $t \in C \cup T(C)$
 $\text{flatten}_C(t) = \text{set}(c)$ if $t \notin C \cup T(C), t = \text{set}(\text{set}(c))$

Let $O = (O_c)_{c \in C}$ be an object domain for C , let its value extension be $V_C^T(O) = (O'_c)_{c \in T(C)}$. The C -object domains $\text{CStr}(\text{embed}_C)(V_C^T(O))$ and O coincide since, on the one hand and by definition, the value extension V_C^T associates the set $O'_c = O_c$ with any $c \in C$ and, on the other hand, $\text{embed}_C(c) = c$ for any $c \in C$. For any class name c in C , we let the value extension V_C^T associate with $\text{set}(c)$ the singleton $O'_{\text{set}(c)} = \{O_c\}$ (i.e., the value of $\text{set}(c)$ is the singleton whose only element is the set of all values associated with c).

$V_C^T(O)$ is a $T(C)$ -object domain and thus both $\text{CStr}(\text{flatten}_C)(V_C^T(O))$ as well as $V_{T(C)}^T(V_C^T(O))$ are $T(T(C))$ -object domains. These two $T(T(C))$ -object domains

¹¹ \uplus denotes disjoint union. Equivalently, one may think the basic types to be absent of any class hierarchy.

are isomorph, as can be easily shown. To illustrate this by means of a simple example, let C be $\{\text{person}\}$, let $O_{\text{person}} = \{\text{Joe}, \text{Mary}\}$, let $\text{CStr}(\text{flatten}_C)(\mathbf{V}_C^T(\mathbf{O})) = (O'_c)_{c \in T(T(C))}$, and let $\mathbf{V}_{T(C)}^T(\mathbf{V}_C^T(\mathbf{O})) = (O''_c)_{c \in T(T(C))}$. Then, $T(C)$ is $\{\text{person}, \text{set}(\text{person})\}$ and $O'_{\text{set}(\text{person})} = \{\{\text{Joe}, \text{Mary}\}\}$. Furthermore, $T(T(C))$ is $\{\text{person}, \text{set}(\text{person}), \text{set}(\text{set}(\text{person}))\}$, and $O''_{\text{set}(\text{set}(\text{person}))} = \{\{\{\text{Joe}, \text{Mary}\}\}\}$ whereas $O'''_{\text{set}(\text{set}(\text{person}))} = \{\{\{\{\text{Joe}, \text{Mary}\}\}\}\}$.

We let a further value extension \mathbf{W}_C^T associate with $\text{set}(c)$ the set $\wp(O_c)$ for any class name $c \in C$ (i.e., the values associated with $\text{set}(c)$ are all the subsets of O_c). Then for the particular case of above with $C = \{\text{person}\}$ and $O_{\text{person}} = \{\text{Joe}, \text{Mary}\}$, \mathbf{W}_C^T associates with $\text{set}(\text{person})$ the set $\{\emptyset, \{\text{Joe}\}, \{\text{Mary}\}, \{\text{Joe}, \text{Mary}\}\}$. Moreover,

- $\text{CStr}(\text{flatten}_C)(\mathbf{W}_C^T(\mathbf{O}))$ associates with $\text{set}(\text{set}(\text{person}))$ the set

$$\{\emptyset, \{\text{Joe}\}, \{\text{Mary}\}, \{\text{Joe}, \text{Mary}\}\}$$

whereas

- $\mathbf{W}_{T(C)}^T(\mathbf{W}_C^T(\mathbf{O}))$ associates with $\text{set}(\text{set}(\text{person}))$ the set

$$\begin{aligned} & \{\emptyset, \{\emptyset\}, \{\{\text{Joe}\}\}, \{\{\text{Mary}\}\}, \{\{\text{Joe}, \text{Mary}\}\}, \\ & \{\emptyset, \{\text{Joe}\}\}, \{\emptyset, \{\text{Mary}\}\}, \{\emptyset, \{\text{Joe}, \text{Mary}\}\}, \\ & \{\{\text{Joe}\}, \{\text{Mary}\}\}, \{\{\text{Joe}\}, \{\text{Joe}, \text{Mary}\}\}, \{\{\text{Mary}\}, \{\text{Joe}, \text{Mary}\}\}, \\ & \{\emptyset, \{\text{Joe}\}, \{\text{Mary}\}\}, \{\emptyset, \{\text{Joe}\}, \{\text{Joe}, \text{Mary}\}\}, \{\emptyset, \{\text{Mary}\}, \{\text{Joe}, \text{Mary}\}\}, \\ & \{\{\text{Joe}\}, \{\text{Mary}\}, \{\text{Joe}, \text{Mary}\}\}, \{\emptyset, \{\text{Joe}\}, \{\text{Mary}\}, \{\text{Joe}, \text{Mary}\}\} \end{aligned}$$

that is, the second one is the powerset of the first one. □

Example 2: T is “set star”

Let B be a set of *basic* types. For any class hierarchy $\mathbf{C} = (C, \leq_C)$ in Poset, we define the class hierarchy $\mathbf{T}(\mathbf{C}) = (T(C), \leq_{T(C)})$ by $T(C) = \bigcup_{i \in \mathbb{N}} T_i$ where $T_0 = C \cup B$ and $T_{i+1} = \{\text{set}(t) : t \in T_i\}$ ($i \in \mathbb{N}$), and $\leq_{T(C)} = \bigcup_{i \in \mathbb{N}} \leq_{T_i}$ with $\leq_{T_0} = \leq_C \cup \text{id}_B$ and $\leq_{T_{i+1}} = \{(\text{set}(t_1), \text{set}(t_2)) : t_1, t_2 \in T_i \wedge t_1 \leq_{T_i} t_2\}$ ($i \in \mathbb{N}$).¹²

By abuse of notation, we write $T(C) = \{\text{set}^n(c) : c \in C \cup B\}$ with $\text{set}^0(c) \stackrel{\text{def}}{=} c$ and $\text{set}^{i+1}(c) \stackrel{\text{def}}{=} \text{set}(\text{set}^i(c))$. It is worth noting that $\text{set}^i(c_1) \leq_{T(C)} \text{set}^i(c_2)$ implies $\text{set}^{i+1}(c_1) \leq_{T(C)} \text{set}^{i+1}(c_2)$.

If $\sigma : \mathbf{C} \rightarrow \mathbf{C}'$ with $\mathbf{C} = (C, \leq_C)$ and $\mathbf{C}' = (C', \leq_{C'})$ is an arrow in Poset, we define $\mathbf{T}(\sigma) : \mathbf{T}(\mathbf{C}) \rightarrow \mathbf{T}(\mathbf{C}')$ by $\mathbf{T}(\sigma)(\text{set}^0(c)) = \sigma(c)$ and $\mathbf{T}(\sigma)(\text{set}^{i+1}(c)) = \text{set}(\mathbf{T}(\sigma)(\text{set}^i(c)))$ for any $c \in C \cup B$. Monotonicity of σ trivially implies monotonicity of $\mathbf{T}(\sigma)$. Hence, \mathbf{T} defines a functor from Poset to Poset.

Notice that $\mathbf{T}(\mathbf{T}(\mathbf{C})) = \mathbf{T}(\mathbf{C})$ for any \mathbf{C} . Hence for any class hierarchy $\mathbf{C} = (C, \leq_C)$ in Poset, the two arrows in Poset (i.e., monotone functions) embed_C and flatten_C can be defined as follows:

- $\text{embed}_C : \mathbf{C} \rightarrow \mathbf{T}(\mathbf{C})$ with $\text{embed}_C(c) = c$ for any $c \in C$;

¹² Similarly as in the previous example, we assume that the basic types are not included in any class hierarchy. Moreover, the constructor $\text{set}(\cdot)$ is likewise nonexistent in any class hierarchy.

– $\text{flatten}_{\mathbf{C}} : \mathbf{T}(\mathbf{T}(\mathbf{C})) \rightarrow \mathbf{T}(\mathbf{C})$ with $\text{flatten}_{\mathbf{C}}(t) = t$ for any $t \in \mathbf{T}(\mathbf{T}(\mathbf{C}))$.

The same as in the previous example, and by definition, $\text{CStr}(\text{embed}_{\mathbf{C}})(\mathbf{V}_{\mathbf{C}}^{\mathbf{T}}(\mathbf{O})) = \mathbf{O}$ for any \mathbf{C} -object domain \mathbf{O} . Moreover, $\text{CStr}(\text{flatten}_{\mathbf{C}})(\mathbf{V}_{\mathbf{C}}^{\mathbf{T}}(\mathbf{O})) = \mathbf{V}_{\mathbf{C}}^{\mathbf{T}}(\mathbf{O})$ since $\text{flatten}_{\mathbf{C}} : \mathbf{T}(\mathbf{T}(\mathbf{C})) \rightarrow \mathbf{T}(\mathbf{C})$ is the identity.

Let $\mathbf{O} = (O_c)_{c \in \mathbf{C}}$ and $\mathbf{V}_{\mathbf{C}}^{\mathbf{T}}(\mathbf{O}) = (O'_c)_{c \in \mathbf{C}}$. We let $\mathbf{V}_{\mathbf{C}}^{\mathbf{T}}$ define $O'_{\text{set}^0(c)} = O_c$ and $O'_{\text{set}^{i+1}(c)} = \wp(O'_{\text{set}^i(c)})$ for any class name $c \in \mathbf{C}$. \square

The monad, as such, and which by abuse of notation we denote by \mathbf{T} , satisfies the following coherence conditions:

1. $\mu \circ \mathbf{T}\mu = \mu \circ \mu\mathbf{T}$ (as natural transformations $\mathbf{T}^3 \rightarrow \mathbf{T}$), and
2. $\mu \circ \mathbf{T}\eta = \mu \circ \eta\mathbf{T} = 1_{\mathbf{T}}$ (as natural transformations $\mathbf{T} \rightarrow \mathbf{T}$; here $1_{\mathbf{T}}$ denotes the identity transformation from \mathbf{T} to \mathbf{T})

where

$$\begin{aligned} \mathbf{T}\mu : \mathbf{T}^3 &\rightarrow \mathbf{T}^2, \text{ i.e., } (\mathbf{T}\mu)_{\mathbf{C}} : \mathbf{T}(\mathbf{T}(\mathbf{T}(\mathbf{C}))) \rightarrow \mathbf{T}(\mathbf{T}(\mathbf{C})) \text{ with } (\mathbf{T}\mu)_{\mathbf{C}} \stackrel{\text{def}}{=} \mathbf{T}(\mu_{\mathbf{C}}), \\ \mu\mathbf{T} : \mathbf{T}^3 &\rightarrow \mathbf{T}^2, \text{ i.e., } (\mu\mathbf{T})_{\mathbf{C}} : \mathbf{T}(\mathbf{T}(\mathbf{T}(\mathbf{C}))) \rightarrow \mathbf{T}(\mathbf{T}(\mathbf{C})) \text{ with } (\mu\mathbf{T})_{\mathbf{C}} \stackrel{\text{def}}{=} \mu_{\mathbf{T}(\mathbf{C})}, \\ \mathbf{T}\eta : \mathbf{T} &\rightarrow \mathbf{T}^2, \text{ i.e., } (\mathbf{T}\eta)_{\mathbf{C}} : \mathbf{T}(\mathbf{C}) \rightarrow \mathbf{T}(\mathbf{T}(\mathbf{C})) \text{ with } (\mathbf{T}\eta)_{\mathbf{C}} \stackrel{\text{def}}{=} \mathbf{T}(\eta_{\mathbf{C}}), \\ \eta\mathbf{T} : \mathbf{T} &\rightarrow \mathbf{T}^2, \text{ i.e., } (\eta\mathbf{T})_{\mathbf{C}} : \mathbf{T}(\mathbf{C}) \rightarrow \mathbf{T}(\mathbf{T}(\mathbf{C})) \text{ with } (\eta\mathbf{T})_{\mathbf{C}} \stackrel{\text{def}}{=} \eta_{\mathbf{T}(\mathbf{C})}. \end{aligned}$$

Let $\mathbf{C} = (C, \leq_C)$ be any class hierarchy, let \mathbf{O} be a \mathbf{C} -object domain. On the one hand, also $\mathbf{T}(\mathbf{C})$ is a class hierarchy and $\mathbf{V}_{\mathbf{C}}^{\mathbf{T}}(\mathbf{O})$ is a $\mathbf{T}(\mathbf{C})$ -object domain. Analogously, $\mathbf{T}(\mathbf{T}(\mathbf{C}))$ is a class hierarchy and $\mathbf{V}_{\mathbf{T}(\mathbf{C})}^{\mathbf{T}}(\mathbf{V}_{\mathbf{C}}^{\mathbf{T}}(\mathbf{O}))$ is a $\mathbf{T}(\mathbf{T}(\mathbf{C}))$ -object domain. Thus, the following pairs

$$\begin{aligned} &\langle \mathbf{C}, \mathbf{O} \rangle \\ &\langle \mathbf{T}(\mathbf{C}), \mathbf{V}_{\mathbf{C}}^{\mathbf{T}}(\mathbf{O}) \rangle \\ &\langle \mathbf{T}(\mathbf{T}(\mathbf{C})), \mathbf{V}_{\mathbf{T}(\mathbf{C})}^{\mathbf{T}}(\mathbf{V}_{\mathbf{C}}^{\mathbf{T}}(\mathbf{O})) \rangle \end{aligned}$$

are objects of $\mathcal{G}(\text{CStr})$.

On the other hand, $\eta_{\mathbf{C}} : \mathbf{C} \rightarrow \mathbf{T}(\mathbf{C})$ and $\mu_{\mathbf{C}} : \mathbf{T}(\mathbf{T}(\mathbf{C})) \rightarrow \mathbf{T}(\mathbf{C})$ are arrows in Poset. Assume

$$\begin{aligned} \langle \eta_{\mathbf{C}}, \text{id}_{\text{CStr}(\mathbf{C})} \rangle : \langle \mathbf{C}, \mathbf{O} \rangle &\rightarrow \langle \mathbf{T}(\mathbf{C}), \mathbf{V}_{\mathbf{C}}^{\mathbf{T}}(\mathbf{O}) \rangle \text{ and} \\ \langle \mu_{\mathbf{C}}, \text{id}_{\text{CStr}(\mathbf{T}(\mathbf{T}(\mathbf{C})))} \rangle : \langle \mathbf{T}(\mathbf{T}(\mathbf{C})), \mathbf{V}_{\mathbf{T}(\mathbf{C})}^{\mathbf{T}}(\mathbf{V}_{\mathbf{C}}^{\mathbf{T}}(\mathbf{O})) \rangle &\rightarrow \langle \mathbf{T}(\mathbf{C}), \mathbf{V}_{\mathbf{C}}^{\mathbf{T}}(\mathbf{O}) \rangle \end{aligned}$$

are arrows in $\mathcal{G}(\text{CStr})$. Since we have paired $\eta_{\mathbf{C}}$ and $\mu_{\mathbf{C}}$ with identity arrows,

$$\begin{aligned} \text{CStr}(\eta_{\mathbf{C}})(\mathbf{V}_{\mathbf{C}}^{\mathbf{T}}(\mathbf{O})) &= \mathbf{O} \text{ and} \\ \text{CStr}(\mu_{\mathbf{C}})(\mathbf{V}_{\mathbf{C}}^{\mathbf{T}}(\mathbf{O})) &= \mathbf{V}_{\mathbf{T}(\mathbf{C})}^{\mathbf{T}}(\mathbf{V}_{\mathbf{C}}^{\mathbf{T}}(\mathbf{O})). \end{aligned}$$

In consequence,

$$\begin{aligned} \text{CStr}(\eta_{\mathbf{C}}) \circ \mathbf{V}_{\mathbf{C}}^{\mathbf{T}} &= \text{id}_{\text{CStr}(\mathbf{C})} & \text{(I)} \\ \text{CStr}(\mu_{\mathbf{C}}) &= \mathbf{V}_{\mathbf{T}(\mathbf{C})}^{\mathbf{T}}, \text{ i.e., } \text{CStr } \mu = \mathbf{V}^{\mathbf{T}} \mathbf{T} & \text{(II)} \end{aligned}$$

In particular, (I) is satisfied when $\eta_{\mathbf{C}}$ is an embedding, and (II) is satisfied when \mathbf{T} is idempotent and $\mu_{\mathbf{C}}$ the identity. That this not necessarily is the case is shown by the following example.

Example 1 (contd.): Coherence conditions for “set”

Let $\mathbf{C} = (C, \leq_C)$ be a class hierarchy in BPoset.

1. (a) $(\mu \circ \mathbf{T}\mu)_{\mathbf{C}} = \mu_{\mathbf{C}} \circ (\mathbf{T}\mu)_{\mathbf{C}} = \mu_{\mathbf{C}} \circ \mathbf{T}(\mu_{\mathbf{C}}) : \mathbf{T}(\mathbf{T}(\mathbf{T}(\mathbf{C}))) \rightarrow \mathbf{T}(\mathbf{C})$

Let $c \in C$. On the one hand,

$$\mu_{\mathbf{C}} = \text{flatten}_{\mathbf{C}} : \mathbf{T}(\mathbf{T}(\mathbf{C})) \rightarrow \mathbf{T}(\mathbf{C})$$

$$\mu_{\mathbf{C}}(t) = \begin{cases} t & \text{if } t \in C \cup T(C) \\ \text{set}(c) & \text{if } t \in T(T(C)) \setminus (C \cup T(C)), t = \text{set}(\text{set}(c)) \end{cases}$$

and on the other, for any $\sigma : \mathbf{C} \rightarrow \mathbf{C}'$,

$$\mathbf{T}(\sigma) : \mathbf{T}(\mathbf{C}) \rightarrow \mathbf{T}(\mathbf{C}')$$

$$\mathbf{T}(\sigma)(t) = \begin{cases} \sigma(t) & \text{if } t \in C \\ \text{set}(\sigma(c)) & \text{if } t \in T(C) \setminus C, t = \text{set}(c) \end{cases}$$

Thus

$$\mathbf{T}(\mu_{\mathbf{C}}) : \mathbf{T}(\mathbf{T}(\mathbf{T}(\mathbf{C}))) \rightarrow \mathbf{T}(\mathbf{T}(\mathbf{C}))$$

$$\begin{aligned} \mathbf{T}(\mu_{\mathbf{C}})(t) &= \begin{cases} \mu_{\mathbf{C}}(t) & \text{if } t \in T(\mathbf{T}(\mathbf{C})) \\ \text{set}(\mu_{\mathbf{C}}(t')) & \text{if } t \in T(T(\mathbf{T}(\mathbf{C}))) \setminus T(\mathbf{T}(\mathbf{C})), t = \text{set}(t') \end{cases} \\ &= \begin{cases} t & \text{if } t \in C \cup T(C) \\ \text{set}(c) & \text{if } t \in T(T(C)) \setminus (C \cup T(C)), t = \text{set}(\text{set}(c)) \\ \text{set}(\text{set}(c)) & \text{if } t \in T(T(T(C))) \setminus (C \cup T(C) \cup T(T(C))), \\ & t = \text{set}(\text{set}(\text{set}(c))) \end{cases} \end{aligned}$$

and therefore,

$$\mu_{\mathbf{C}} \circ \mathbf{T}(\mu_{\mathbf{C}}) : \mathbf{T}(\mathbf{T}(\mathbf{T}(\mathbf{C}))) \rightarrow \mathbf{T}(\mathbf{C})$$

$$\text{set}(\text{set}(\text{set}(c))) \mapsto \text{set}(c)$$

$$\text{set}(\text{set}(c)) \mapsto \text{set}(c)$$

$$\text{set}(c) \mapsto \text{set}(c)$$

$$c \mapsto c$$

(b) $(\mu \circ \mu\mathbf{T})_{\mathbf{C}} = \mu_{\mathbf{C}} \circ (\mu\mathbf{T})_{\mathbf{C}} = \mu_{\mathbf{C}} \circ \mu_{\mathbf{T}(\mathbf{C})} : \mathbf{T}(\mathbf{T}(\mathbf{T}(\mathbf{C}))) \rightarrow \mathbf{T}(\mathbf{C})$

Let $c \in C$. On the one hand,

$$\mu_{\mathbf{T}(\mathbf{C})} = \text{flatten}_{\mathbf{T}(\mathbf{C})} : \mathbf{T}(\mathbf{T}(\mathbf{T}(\mathbf{C}))) \rightarrow \mathbf{T}(\mathbf{T}(\mathbf{C}))$$

$$\text{set}(\text{set}(\text{set}(c))) \mapsto \text{set}(\text{set}(c))$$

$$\text{set}(\text{set}(c)) \mapsto \text{set}(\text{set}(c))$$

$$\text{set}(c) \mapsto \text{set}(c)$$

$$c \mapsto c$$

and on the other,

$$\mu_{\mathbf{C}} = \text{flatten}_{\mathbf{C}} : \mathbf{T}(\mathbf{T}(\mathbf{C})) \rightarrow \mathbf{T}(\mathbf{C})$$

$$\text{set}(\text{set}(c)) \mapsto \text{set}(c)$$

$$\text{set}(c) \mapsto \text{set}(c)$$

$$c \mapsto c$$

Therefore,

$$\begin{aligned} \mu_{\mathbf{C}} \circ \mu_{T(\mathbf{C})} &: T(T(T(\mathbf{C}))) \rightarrow T(\mathbf{C}) \\ \text{set}(\text{set}(\text{set}(c))) &\mapsto \text{set}(c) \\ \text{set}(\text{set}(c)) &\mapsto \text{set}(c) \\ \text{set}(c) &\mapsto \text{set}(c) \\ c &\mapsto c \end{aligned}$$

That is, the first coherence condition holds.

2. (a) $(\mu \circ T\eta)_{\mathbf{C}} = \mu_{\mathbf{C}} \circ (T\eta)_{\mathbf{C}} = \mu_{\mathbf{C}} \circ T(\eta_{\mathbf{C}}) : T(\mathbf{C}) \rightarrow T(\mathbf{C})$

Let $c \in C$. On the one hand,

$$\begin{aligned} T(\eta_{\mathbf{C}}) &= T(\text{embed}_{\mathbf{C}}) : T(\mathbf{C}) \rightarrow T(T(\mathbf{C})) \\ \text{set}(c) &\mapsto \text{set}(c) \\ c &\mapsto c \end{aligned}$$

and on the other,

$$\begin{aligned} \mu_{\mathbf{C}} = \text{flatten}_{\mathbf{C}} &: T(T(\mathbf{C})) \rightarrow T(\mathbf{C}) \\ \text{set}(\text{set}(c)) &\mapsto \text{set}(c) \\ \text{set}(c) &\mapsto \text{set}(c) \\ c &\mapsto c \end{aligned}$$

Therefore,

$$\begin{aligned} \mu_{\mathbf{C}} \circ T(\eta_{\mathbf{C}}) &: T(\mathbf{C}) \rightarrow T(\mathbf{C}) \\ \text{set}(c) &\mapsto \text{set}(c) \\ c &\mapsto c \end{aligned}$$

thus $\mu_{\mathbf{C}} \circ T(\eta_{\mathbf{C}})$ is the identity on $T(\mathbf{C})$.

(b) $(\mu \circ \eta T)_{\mathbf{C}} = \mu_{\mathbf{C}} \circ (\eta T)_{\mathbf{C}} = \mu_{\mathbf{C}} \circ \eta_{T(\mathbf{C})}$

Let $c \in C$. On the one hand,

$$\begin{aligned} \eta_{T(\mathbf{C})} &= \text{embed}_{T(\mathbf{C})} : T(\mathbf{C}) \rightarrow T(T(\mathbf{C})) \\ \text{set}(c) &\mapsto \text{set}(c) \\ c &\mapsto c \end{aligned}$$

and on the other,

$$\begin{aligned} \mu_{\mathbf{C}} = \text{flatten}_{\mathbf{C}} &: T(T(\mathbf{C})) \rightarrow T(\mathbf{C}) \\ \text{set}(\text{set}(c)) &\mapsto \text{set}(c) \\ \text{set}(c) &\mapsto \text{set}(c) \\ c &\mapsto c \end{aligned}$$

Therefore,

$$\begin{aligned} \mu_{\mathbf{C}} \circ \eta_{T(\mathbf{C})} &: T(\mathbf{C}) \rightarrow T(\mathbf{C}) \\ \text{set}(c) &\mapsto \text{set}(c) \\ c &\mapsto c \end{aligned}$$

thus also $\mu_{\mathbf{C}} \circ \eta_{T(\mathbf{C})}$ is the identity on $T(\mathbf{C})$.

That is, the second coherence condition likewise holds.

Consequently, T defines a monad.

By definition, $\text{CStr}(\text{embed}_{\mathbf{C}})(V_{\mathbf{C}}^T(\mathbf{O})) = \mathbf{O}$, that is, condition (I) holds for T . On the contrary, $\text{CStr}(\text{flatten}_{\mathbf{C}})(V_{\mathbf{C}}^T(\mathbf{O}))$ and $V_{T(\mathbf{C})}^T(V_{\mathbf{C}}^T(\mathbf{O}))$, both object domains for $T(T(\mathbf{C}))$, are not equal but only isomorph. That is, condition (II) does not hold for this choice of T . \square

Example 2 (contd.): Coherence conditions for “set star”

Let $\mathbf{C} = (C, \leq_C)$ be a class hierarchy in Poset.

1. (a) $(\mu \circ \mathbf{T}\mu)_C = \mu_C \circ (\mathbf{T}\mu)_C = \mu_C \circ \mathbf{T}(\mu_C) : \mathbf{T}(\mathbf{T}(\mathbf{T}(C))) \rightarrow \mathbf{T}(C)$

Let $c \in C$. On the one hand,

$$\mu_C = \text{flatten}_C : \mathbf{T}(\mathbf{T}(C)) \rightarrow \mathbf{T}(C)$$

$$t \mapsto t$$

and on the other,

$$\mathbf{T}(\mu_C) : \mathbf{T}(\mathbf{T}(\mathbf{T}(C))) \rightarrow \mathbf{T}(\mathbf{T}(C))$$

$$t \mapsto t$$

Therefore,

$$\mu_C \circ \mathbf{T}(\mu_C) : \mathbf{T}(\mathbf{T}(\mathbf{T}(C))) \rightarrow \mathbf{T}(C)$$

$$t \mapsto t$$

(b) $(\mu \circ \mu\mathbf{T})_C = \mu_C \circ (\mu\mathbf{T})_C = \mu_C \circ \mu_{\mathbf{T}(C)} : \mathbf{T}(\mathbf{T}(\mathbf{T}(C))) \rightarrow \mathbf{T}(C)$

Let $c \in C$. On the one hand,

$$\mu_{\mathbf{T}(C)} = \text{flatten}_{\mathbf{T}(C)} : \mathbf{T}(\mathbf{T}(\mathbf{T}(C))) \rightarrow \mathbf{T}(\mathbf{T}(C))$$

$$t \mapsto t$$

and on the other,

$$\mu_C = \text{flatten}_C : \mathbf{T}(\mathbf{T}(C)) \rightarrow \mathbf{T}(C)$$

$$t \mapsto t$$

Therefore,

$$\mu_C \circ \mu_{\mathbf{T}(C)} : \mathbf{T}(\mathbf{T}(\mathbf{T}(C))) \rightarrow \mathbf{T}(C)$$

$$t \mapsto t$$

That is, the first coherence condition holds.

2. (a) $(\mu \circ \mathbf{T}\eta)_C = \mu_C \circ (\mathbf{T}\eta)_C = \mu_C \circ \mathbf{T}(\eta_C) : \mathbf{T}(C) \rightarrow \mathbf{T}(C)$

Let $c \in C$. On the one hand,

$$\mathbf{T}(\eta_C) = \mathbf{T}(\text{embed}_C) : \mathbf{T}(C) \rightarrow \mathbf{T}(\mathbf{T}(C))$$

$$t \mapsto t$$

and on the other,

$$\mu_C = \text{flatten}_C : \mathbf{T}(\mathbf{T}(C)) \rightarrow \mathbf{T}(C)$$

$$t \mapsto t$$

Therefore,

$$\mu_C \circ \mathbf{T}(\eta_C) : \mathbf{T}(C) \rightarrow \mathbf{T}(C)$$

$$t \mapsto t$$

thus $\mu_C \circ \mathbf{T}(\eta_C)$ is the identity on $\mathbf{T}(C)$.

(b) $(\mu \circ \eta\mathbf{T})_C = \mu_C \circ (\eta\mathbf{T})_C = \mu_C \circ \eta_{\mathbf{T}(C)}$

Let $c \in C$. On the one hand,

$$\eta_{\mathbf{T}(C)} = \text{embed}_{\mathbf{T}(C)} : \mathbf{T}(C) \rightarrow \mathbf{T}(\mathbf{T}(C))$$

$$t \mapsto t$$

and on the other,

$$\mu_C = \text{flatten}_C : \mathbf{T}(\mathbf{T}(C)) \rightarrow \mathbf{T}(C)$$

$$t \mapsto t$$

Therefore,

$$\mu_C \circ \eta_{\mathbf{T}(C)} : \mathbf{T}(C) \rightarrow \mathbf{T}(C)$$

$$t \mapsto t$$

thus also $\mu_C \circ \eta_{\mathbf{T}(C)}$ is the identity on $\mathbf{T}(C)$.

That is, the second coherence condition likewise holds.

Consequently, T defines a monad.

By definition, $\text{CStr}(\text{embed}_{\mathcal{C}})(\mathbf{V}_{\mathcal{C}}^T(\mathbf{O})) = \mathbf{O}$, that is, condition (I) holds for T . Moreover, the $T(T(\mathcal{C}))$ -object domains $\text{CStr}(\text{flatten}_{\mathcal{C}})(\mathbf{V}_{\mathcal{C}}^T(\mathbf{O}))$ and $\mathbf{V}_{T(\mathcal{C})}^T(\mathbf{V}_{\mathcal{C}}^T(\mathbf{O}))$ coincide and are equal to $\mathbf{V}_{\mathcal{C}}^T(\mathbf{O})$. That is, also condition (II) holds for this choice of T . \square