



HAL
open science

Term Charters

Alexander Knapp, María Victoria Cengarle

► **To cite this version:**

Alexander Knapp, María Victoria Cengarle. Term Charters. 24th International Workshop on Algebraic Development Techniques (WADT), Jul 2018, Egham, United Kingdom. pp.119-138, 10.1007/978-3-030-23220-7_7. hal-02364571

HAL Id: hal-02364571

<https://inria.hal.science/hal-02364571>

Submitted on 15 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Term Charters

Alexander Knapp¹ and María Victoria Cengarle²

¹ Universität Augsburg, Germany
knapp@informatik.uni-augsburg.de
² mv.cengarle@gmail.com

Abstract. Expressions in specification and programming languages often extend algebraic terms by additional term formation rules and enlarged evaluation domains. For use in different contexts, the semantics of these expressions must allow for interface modification, that is, evaluation must be compatible with signature translation and with variable renaming. This work introduces term charters as an abstract framework for terms and expressions following the slogan “evaluation is invariant under change of notation”. Several examples illustrate the use of term charters. Different term charter representations are discussed and morphisms between term charters are defined. Finally, the relation of term charters with institutions in general and with context institutions in particular is demonstrated.

1 Introduction

Algebraic signatures are customarily connected with a fixed grammar for term construction as well as a canonical accompanying procedure for term evaluation; what may vary is the term representation when using different names for function symbols. More often than not, expressions of specification or programming languages build on algebraic signatures and evaluation, but exceed the boundaries of such a tightly connected syntactical and semantical scheme. For instance, the “Object Constraint Language” (OCL [10]) is mainly based on order-sorted algebra with function symbols like $+$ and `append` giving rise to expressions like $1 + 2$ and `c1->append(c2)` as well as their usual evaluation.

The OCL grammar, however, offers a general iteration construct `c->iterate(i : s ; a : s' = e0 | e)` for applying the expression e over the successive elements of the sequence c using the iterator i and accumulating the partial results in a initialised by e_0 . Here, e has to be evaluated multiple times for different valuations of its bound variables i and a . This would be at least cumbersome to comprise into order-sorted terms using a different function symbol for every possible iteration expression. Alternatively, full higher-order algebras [14] could be employed, which yet would seem rather generous for such a small excursion. A more direct solution is to use a special evaluation rule for the iteration construct with only one local auxiliary semantic function of higher order.

Another set of language features of the OCL that stretches order-sorted algebras is `undef` representing undefinedness, its corresponding undefinedness test `e.isUndef()`, and the conditional `if e then e1 else e2 endif` that is only

strict in its first argument. A change to the setting of partial algebras [14] could be considered, but also sticking with order-sortedness and just extending the value domains of the algebras by a semantic value \dagger and providing a special evaluation rule for the conditional presents a viable alternative.

These examples suggest that it is sometimes necessary to consider special expression construction and evaluation rules added to algebraic languages. In order to be applicable in varying contexts and to allow for different representations of the underlying algebraic signature, evaluation still has to be compatible with changes to the interface of expressions, i.e., the function symbols and the free variables. In the following, we discuss a notion of evaluation for generalised terms in an abstract framework that we call *term charter*. It captures the fundamental properties of evaluation w.r.t. variables, signatures, variable renamings, and signature translations along the slogan “evaluation is invariant under change of notation”. Using this as a guideline, the present account of term charters provides a simplification and rational reconstruction of our previous rendition in [6].

We first review the motivational OCL examples and the principal evaluation properties more precisely in Sect. 2, though starting, for simplicity, from many-sorted algebras. Based on indexed categories [15,17], we then present term charters in Sect. 3. Term charters are built over *term charter domains* consisting of values and variables, structures, and the underlying values of structures, all indexed over signatures. A *term charter* itself consists of a term constructor, a variable embedding, and evaluation maps for terms over a signature and variables using valuations; the interaction of these ingredients is regulated by three axioms for variable evaluation, variable renaming, and signature translations such that indeed evaluation is invariant under change of notation. We also discuss two more compact alternative presentations of term charters using comma categories [8] and the Grothendieck construction [17] in Sect. 4. The latter lets us directly construct an *institution* [5,3] from a term charter as shown in Sect. 5; again, this new construction simplifies [6]. The generation of an institution conveniently embeds term charters into the landscape of logical systems. What is more, the induced institution is closely related to the notion of *context institutions* by Pawłowski [12], which has been introduced in a similar effort to capture open formulæ over variables in institutions. We finally discuss different notions of *morphisms* between term charter domains and term charters in Sect. 6. We conclude in Sect. 7 with some application scenarios of term charters to OCL. Mainly for fixing notation, some categorical prerequisites are briefly summarised in App. A.

2 Extending Algebraic Term Evaluation

The evaluation of terms over a many-sorted *signature* $\Sigma = (S, F)$ of sorts and function symbols directly follows term formation. Given S -sorted *variables* $X = (X_s)_{s \in S}$, the family of terms $\mathcal{T}_\Sigma^m(X) = (\mathcal{T}_\Sigma^m(X)_s)_{s \in S}$ over Σ and X is inductively defined by

- $x \in \mathcal{T}_\Sigma^m(X)_s$ if $x \in X_s$;

– $f(t_1, \dots, t_n) \in \mathcal{T}_\Sigma^m(X)_s$ if $f \in F_{s_1 \dots s_n s}$ and $t_i \in \mathcal{T}_\Sigma^m(X)_{s_i}$ for $1 \leq i \leq n$.

For a Σ -structure $M = (S^M, F^M)$ and a valuation $\beta = (\beta_s : X_s \rightarrow s^M)_{s \in S} : X \rightarrow U_\Sigma(M)$ from the variables X to the underlying sets $U_\Sigma(M) = S^M$ of the structure, term evaluation $\llbracket - \rrbracket_{(\Sigma, X)}^m(M, \beta) = (\llbracket - \rrbracket_{(\Sigma, X)}^m(M, \beta)_s : \mathcal{T}_\Sigma^m(X)_s \rightarrow s^M)_{s \in S} : \mathcal{T}_\Sigma^m(X) \rightarrow U_\Sigma(M)$ is inductively defined in correspondence with term formation of the grammar:

- $\llbracket x \rrbracket_{(\Sigma, X)}^m(M, \beta)_s = \beta_s(x)$ for $x \in X_s$, and
- $\llbracket f(t_1, \dots, t_n) \rrbracket_{(\Sigma, X)}^m(M, \beta)_s = f^M(\llbracket t_1 \rrbracket_{(\Sigma, X)}^m(M, \beta)_{s_1}, \dots, \llbracket t_n \rrbracket_{(\Sigma, X)}^m(M, \beta)_{s_n})$ for $f \in F_{s_1 \dots s_n s}$ and $t_i \in \mathcal{T}_\Sigma^m(X)_{s_i}$ for $1 \leq i \leq n$.

If, however, other term formation rules are considered that transcend the boundaries of the compliance of the function symbols f with the interpreting functions f^M or that require values besides the sorted universes s^M of the structure M for evaluation, recourse to an extended notion of structures has to be taken.

Example 1. Consider the many-sorted signature $\Sigma^\circ = (S, F)$ showing as sorts S , **Bool** for the booleans, **Int** for the integers, and, for every sort $s \in S$, **Seq**(s) for the (finite) lists, together with some function symbols F like $0 \in F_{\text{Int}}$ for zero and $+ \in F_{\text{Int Int Int}}$ for addition; and the Σ° -structure $M^\circ = (S^{M^\circ}, F^{M^\circ})$ interpreting these sorts and function symbols of Σ° in the standard way, such that, in particular, $\text{Bool}^{M^\circ} = \{\text{ff}, \text{tt}\}$, $\text{Int}^{M^\circ} = \mathbb{Z}$, $\text{Seq}(s)^{M^\circ} = (s^{M^\circ})^*$, $0^{M^\circ} = 0$, and $+^{M^\circ} = \lambda v_1, v_2. v_1 + v_2$. Besides the usual term formation rules from the variables and the function symbols above, we want to include an OCL-like iteration construct

`c->iterate(i : s; a : s' = e0 | e)`

in the grammar of iteration terms $\mathcal{T}_\Sigma^{\text{it}}(X)$: Starting with the accumulator variable a of sort s' initialised with the evaluated term $e_0 \in \mathcal{T}_\Sigma^{\text{it}}(X)_{s'}$, each entry in the evaluated term $c \in \mathcal{T}_\Sigma^{\text{it}}(X)_{\text{Seq}(s)}$ is successively assigned to the iterator variable i of sort s updating the accumulator a with the result of evaluating the term $e \in \mathcal{T}_\Sigma^{\text{it}}(X \uplus \{i : s, a : s'\})_{s'}$, which thus may contain i and a as fresh local variables, and finally, the last value of a is returned; such that, e.g., evaluation of `Seq{1, 2, 3}->iterate(i : Int; a : Int = 0 | i+a)` yields 6. Evaluation of iterations could rely on an extended Σ° with function symbols $it_{\lambda i:s, a:s'. e} \in F_{\text{Seq}(s) s'}$ for every possible iteration expression (and a correspondingly extended M°), or an embedding of the whole language into higher-order algebra. As a more moderate alternative, the evaluation of iterations can also be directly expressed using M° based on an intermediate, ad hoc higher-order function it^{M° :

$$\begin{aligned} \llbracket c \text{-> iterate}(i : s; a : s' = e_0 \mid e) \rrbracket_{(\Sigma^\circ, X)}^{\text{it}}(M^\circ, \beta)_{s'} = \\ it^{M^\circ}(\llbracket c \rrbracket_{(\Sigma^\circ, X)}^{\text{it}}(M^\circ, \beta)_{\text{Seq}(s)}, \llbracket e_0 \rrbracket_{(\Sigma^\circ, X)}^{\text{it}}(M^\circ, \beta)_{s'}, \\ \{(v_i, v_a) \mapsto \llbracket e \rrbracket_{(\Sigma^\circ, X \uplus \{i:s, a:s'\})}^{\text{it}}(M^\circ, \beta \{i : s \mapsto v_i, a : s' \mapsto v_a\})_{s'}\}, \end{aligned}$$

where $it^{M^\circ} : \text{Seq}(s)^{M^\circ} \times s'^{M^\circ} \times (s^{M^\circ} \times s'^{M^\circ} \rightarrow s'^{M^\circ}) \rightarrow s'^{M^\circ}$, like Haskell's `foldr`, is defined by $it^{M^\circ}(\varepsilon, v_a, h) = v_a$ and $it^{M^\circ}(v_i :: \ell, v_a, h) = it^{M^\circ}(\ell, h(v_i, v_a), h)$.

In order to obtain a proper evaluation, the basic properties of algebraic term evaluation w.r.t. the interface of terms must be retained also for extended term formation rules and structures, i.e., compatibility with variable renamings and signature translations has to be ensured. For the many-sorted case a *variable renaming* $\xi = (\xi_s : X_{1,s} \rightarrow X_{2,s})_{s \in S} : X_1 \rightarrow X_2$ induces a *term renaming* $\mathcal{T}_\Sigma^m(\xi) = (\mathcal{T}_\Sigma^m(\xi)_s : \mathcal{T}_\Sigma^m(X_1)_s \rightarrow \mathcal{T}_\Sigma^m(X_2)_s)_{s \in S} : \mathcal{T}_\Sigma^m(X_1) \rightarrow \mathcal{T}_\Sigma^m(X_2)$, which usually is denoted again by ξ ; then for each $t \in \mathcal{T}_\Sigma^m(X_1)_s$ a “renaming lemma” holds:

$$(R_m) \quad \llbracket \xi_s(t) \rrbracket_{(\Sigma, X_2)}^m(M, \beta)_s = \llbracket t \rrbracket_{(\Sigma, X_1)}^m(M, \beta \circ \xi)_s .$$

Moreover, a *signature translation* $\sigma = (\sigma_S : S \rightarrow S', \sigma_F : F \rightarrow F') : \Sigma = (S, F) \rightarrow \Sigma' = (S', F')$ and the corresponding *reducts* of variables $X'|\sigma = (X'_{\sigma_S(s)})_{s \in S}$ and terms $\mathcal{T}_{\Sigma'}^m(X')|\sigma = (\mathcal{T}_{\Sigma'}^m(X')_{\sigma_S(s)})_{s \in S}$ induce a *term translation* $\mathcal{T}_\sigma^m(X') : \mathcal{T}_\Sigma^m(X'|\sigma) \rightarrow \mathcal{T}_{\Sigma'}^m(X')|\sigma$ inductively defined by

- $\mathcal{T}_\sigma^m(X')_s(x') = x'$ for $x' \in (X'|\sigma)_s$;
- $\mathcal{T}_\sigma^m(X')_s(f(t_1, \dots, t_n)) = \sigma_F(f)(\mathcal{T}_\sigma^m(X')_{s_1}(t_1), \dots, \mathcal{T}_\sigma^m(X')_{s_n}(t_n))$
for $f \in F_{s_1 \dots s_n}$ and $t_i \in \mathcal{T}_\Sigma^m(X'|\sigma)_{s_i}$ for $1 \leq i \leq n$.

Then, for the σ -reduct of structures $M'|\sigma = ((\sigma_S(s)^{M'})_{s \in S}, (F'^{M'}_{\sigma_S(\bar{s})})_{\bar{s} \in S'})$ and valuations $\beta'|\sigma = (\beta'_{\sigma_S(s)})_{s \in S}$, and abbreviating $\mathcal{T}_\sigma^m(X')$ by just σ , a “translation lemma” for each $t \in \mathcal{T}_\Sigma^m(X'|\sigma)_s$ holds:

$$(T_m) \quad \llbracket \sigma_s(t) \rrbracket_{(\Sigma', X')}^m(M', \beta')_{\sigma_S(s)} = \llbracket t \rrbracket_{(\Sigma, X'|\sigma)}^m(M'|\sigma, \beta'|\sigma)_s .$$

These properties (R_m) , (T_m) can be straightforwardly transferred to properties (R_{it}) , (T_{it}) for the evaluation of iterations in Ex. 1: The signatures and structures are taken to be restricted to the forms of Σ° and M° such that signature translations preserve **Bool**, **Int**, **Seq**, **0**, etc., and $\mathcal{T}_\Sigma^{\text{it}}(\xi)$ and $\mathcal{T}_{\sigma, X'}^{\text{it}}$ are defined to preserve the grammar also of **iterate** such that, in particular, the additional clause for variable renamings becomes

$$\begin{aligned} \mathcal{T}_\Sigma^{\text{it}}(\xi)_{s'}(c \rightarrow \text{iterate}(i : s; a : s' = e_0 \mid e)) = \\ \mathcal{T}_\Sigma^{\text{it}}(\xi)_{\text{Seq}(s)}(c \rightarrow \text{iterate}(i : s; a : s' = \mathcal{T}_\Sigma^{\text{it}}(\xi)_{s'}(e_0) \mid \\ \mathcal{T}_\Sigma^{\text{it}}(\xi)\{i : s \mapsto i : s, a : s' \mapsto a : s'\})_{s'}(e)) . \end{aligned}$$

Name capturing of the bound variables i and a is avoided due to the explicit provision of fresh variables in $X \uplus \{i : s, a : s'\}$ and the corresponding extension of ξ .

Example 2. Again inspired by the OCL, consider the addition of a term former **undef** denoting *undefinedness* over the signature Σ° and the Σ° -structure M° from above. The predefined function symbols are to be interpreted strictly, i.e., if any of the argument terms of a function symbol is undefined, the overall result shall be undefined. Additionally, we want to include the non-strict term former $e.\text{isUndef}()$ for testing for undefinedness, and the conditional **if** e **then**

e_1 **else** e_2 **endif** which is only strict in its first argument. Here, a first option is to change the interpreting structure M° into a partial algebra. As an alternative to such a pervasive adaptation, we may also opt for just including a semantic value \dagger into the underlying sets of M° by defining $U^\dagger(M^\circ) = (s^{M^\circ} \uplus \{\dagger\})_{s \in S}$. The valuations then are of the form $\beta^\dagger : X \rightarrow U^\dagger(M^\circ)$ and the evaluation becomes $\llbracket - \rrbracket_{(\Sigma^\circ, X)}^u(M^\circ, \beta^\dagger) : \mathcal{T}_{\Sigma^\circ}^u(X) \rightarrow U^\dagger(M^\circ)$ defined by

$$\begin{aligned}
& - \llbracket x \rrbracket_{(\Sigma^\circ, X)}^u(M^\circ, \beta^\dagger)_s = \beta_s^\dagger(x); \\
& - \llbracket f(\vec{e}) \rrbracket_{(\Sigma^\circ, X)}^u(M^\circ, \beta^\dagger)_s = \begin{cases} f^{M^\circ}(\vec{v}) & \text{if } \forall i. \vec{v}_i = \llbracket \vec{e}_i \rrbracket_{(\Sigma^\circ, X)}^u(M^\circ, \beta^\dagger)_{s_i} \neq \dagger; \\ \dagger & \text{otherwise} \end{cases}; \\
& - \llbracket \mathbf{undef} \rrbracket_{(\Sigma^\circ, X)}^u(M^\circ, \beta^\dagger)_s = \dagger; \\
& - \llbracket e.\mathbf{isUndef}() \rrbracket_{(\Sigma^\circ, X)}^u(M^\circ, \beta^\dagger)_{\mathbf{Bool}} = \begin{cases} tt & \text{if } \llbracket e \rrbracket_{(\Sigma^\circ, X)}^u(M^\circ, \beta^\dagger)_s = \dagger; \\ ff & \text{otherwise} \end{cases}; \\
& - \llbracket \mathbf{if } e \mathbf{ then } e_{tt} \mathbf{ else } e_{ff} \mathbf{ endif} \rrbracket_{(\Sigma^\circ, X)}^u(M^\circ, \beta^\dagger)_s = \\
& \quad \begin{cases} \llbracket e_b \rrbracket_{(\Sigma^\circ, X)}^u(M^\circ, \beta^\dagger)_s & \text{if } \llbracket e \rrbracket_{(\Sigma^\circ, X)}^u(M^\circ, \beta^\dagger)_{\mathbf{Bool}} = b \in \{ff, tt\} \\ \dagger & \text{otherwise} \end{cases}.
\end{aligned}$$

The corresponding properties (R_u) and (T_u), where $\mathcal{T}_\Sigma^u(\xi)$ and $\mathcal{T}_{\sigma, X}^u$ now also preserve **undef**, **isUndef()**, and the conditional, can be proved by induction on term formation.

3 Term Charter Domains and Term Charters

We present an abstract framework capturing the basic properties of term evaluation over structures, where the terms are formed according to a grammar and the structures can consist of extended algebras. Signatures, values and variables, structures, and underlying values are comprised into a *term charter domain*; the grammar of term formation, variable renaming, variable embedding, and the evaluation proper is collected into a *term charter* over such a domain requiring conditions for variable embedding and invariance under renaming and translation. We use indexed categories [15,17] (see App. A.1) as a foundation for the abstract framework, since they provide a close match with the algebraic signatures as indexes also used in institutions [14]. We show that *term structure constructors* induce term charters. Additionally, we discuss a different form for term translation and a notion of *substitutions* in term charters. Term charters have already been introduced in [6]. The presentation there, however, employed the technical means of Grothendieck categories throughout, thereby somewhat obscuring the relation of term charters with generalised evaluation.

3.1 Term Charter Domains

A *term charter domain* $(\mathbb{S}, Val, Str, U)$ is given by a category \mathbb{S} of *signatures*, an indexed category $Val : \mathbb{S}^{\text{op}} \rightarrow \text{Cat}$ of *value variables*, an indexed category $Str : \mathbb{S}^{\text{op}} \rightarrow \text{Cat}$ of *structures*, and an *underlying* indexed functor $U : Str \rightarrow Val$.

We use the terminology of “value variables”, since Val plays the rôle of both. In fact, variables in Val can only be assigned values in Val , not, e.g., sets of values or functions.

Example 3. (1) A term charter domain $(\mathbb{S}^m, Val^m, Str^m, U^m)$ for many-sorted algebras can be obtained as follows (order-sorted or higher-order algebras can be handled similarly): The signature category \mathbb{S}^m comprises the many-sorted signatures $\Sigma = (S, F)$ and the signature morphisms $\sigma = (\sigma_S, \sigma_F) : \Sigma \rightarrow \Sigma'$. The category $Val^m(\Sigma)$ of Σ -value variables comprises the set families $X = (X_s)_{s \in S}$ and the renamings $\xi = (\xi_s : X_{1,s} \rightarrow X_{2,s})_{s \in S} : X_1 \rightarrow X_2$; the reduct functor $Val^m(\sigma) : Val^m(\Sigma') \rightarrow Val^m(\Sigma)$ yields $Val^m(\sigma)(X') = (X'_{\sigma_S(s)})_{s \in S}$ and $Val^m(\sigma)(\xi') = (\xi'_{\sigma_S(s)})_{s \in S}$. The category $Str^m(\Sigma)$ of Σ -structures comprises the many-sorted structures $(S^M = (s^M)_{s \in S}, F^M = (F_{\bar{s}}^M)_{\bar{s} \in S^+})$ and the homomorphisms $\mu = (\mu_s : s^{M_1} \rightarrow s^{M_2})_{s \in S} : M_1 \rightarrow M_2$ with $\mu(f^{M_1}(\vec{v})) = f^{M_2}(\mu(\vec{v}))$ for $f \in F$; the reduct functor $Str^m(\sigma) : Str^m(\Sigma') \rightarrow Str^m(\Sigma)$ yields $Str^m(\sigma)(M') = ((\sigma_S(s)^{M'})_{s \in S}, (\sigma_F(F_{\sigma_S(\bar{s})})^{M'})_{\bar{s} \in S^+})$ and $Str^m(\sigma)(\mu') = (\mu'_{\sigma_S(s)})_{s \in S}$. Finally, define $U_{\Sigma}^m(M) = (s^M)_{s \in S}$ and $U_{\Sigma}^m(\mu : M_1 \rightarrow M_2) = (\mu_s)_{s \in S}$ for the underlying value variables and renamings, such that indeed $U_{\Sigma}^m \circ Str^m(\sigma) = Val^m(\sigma) \circ U_{\Sigma'}^m$.

(2) For **iterate** in Ex. 1, the term charter domain $(\mathbb{S}^\circ, Val^\circ, Str^\circ, U^\circ)$ is obtained by restricting $(\mathbb{S}^m, Val^m, Str^m, U^m)$: \mathbb{S}° shows all signatures with predefined sorts **Bool**, **Int**, **Seq**(s) and function symbols 0 , $+$, etc., as well as all signature morphisms preserving these predefined symbols; Val° is only defined on \mathbb{S}° ; the structures in Str° are those with standard interpretations of the predefined symbols; and the underlying functor U° operates only on Str° .

(3) For the undefinedness extension in Ex. 2, the term charter domain $(\mathbb{S}^\circ, Val^\dagger, Str^\circ, U^\dagger)$ is defined using the indexed endo-functor $\dagger : Val^\circ \rightarrow Val^\circ$ with $\dagger_{\Sigma^\circ}(X) = (X_s \uplus \{\dagger\})_{s \in S}$ and $\dagger_{\Sigma^\circ}(\xi) = (\xi_s \{\dagger \mapsto \dagger\})_{s \in S}$ for a Kleisli construction [1]: The objects of $Val^\dagger(\Sigma^\circ)$ and $Val^\circ(\Sigma^\circ)$ coincide, but a morphism $\xi^\dagger : X_1 \rightarrow X_2$ of $Val^\dagger(\Sigma^\circ)$ is given by the $Val^\circ(\Sigma^\circ)$ -morphism $\xi : X_1 \rightarrow \dagger_{\Sigma^\circ}(X_2)$; the $Val^\dagger(\Sigma^\circ)$ -identities $1_X : X \rightarrow X$ are the inclusions $\iota_X : X \rightarrow \dagger_{\Sigma^\circ}(X)$, the composition $\xi_1^\dagger; \xi_2^\dagger$ of $\xi_1^\dagger : X_1 \rightarrow X_2$ and $\xi_2^\dagger : X_2 \rightarrow X_3$ is $(\xi_1; \xi_2 \{\dagger \mapsto \dagger\})^\dagger : X_1 \rightarrow X_3$. The underlying functor $U^\dagger : Str^\circ \rightarrow Val^\dagger$ is chosen as $U_{\Sigma^\circ}^\dagger(M) = U_{\Sigma^\circ}^\circ(M)$ and $U_{\Sigma^\circ}^\dagger(\mu : M_1 \rightarrow M_2) = (U_{\Sigma^\circ}^\circ(\mu); \iota_{U_{\Sigma^\circ}^\circ(M_2)})^\dagger$.

3.2 Term Charters

Let $(\mathbb{S}, Val, Str, U)$ be a term charter domain. Let $\mathcal{T} : Val \rightarrow Val$ be a lax indexed functor, where the functors $\mathcal{T}_\Sigma : Val(\Sigma) \rightarrow Val(\Sigma)$ for each $\Sigma \in |\mathbb{S}|$ *construct terms* and *rename terms* along value variable renamings, and the natural transformations $\mathcal{T}_\sigma : Val(\sigma)$; $\mathcal{T}_\Sigma \rightarrow \mathcal{T}_{\Sigma'}$; $Val(\sigma)$ for each $\sigma \in \mathbb{S}(\Sigma, \Sigma')$ *translate terms* along signature morphisms. Let furthermore $\nu : 1_{Val} \rightarrow \mathcal{T}$ be a lax indexed natural transformation, where the natural transformations $\nu_\Sigma : 1_{Val(\Sigma)} \rightarrow \mathcal{T}_\Sigma$ for each $\Sigma \in |\mathbb{S}|$ *embed value variables into terms*. Finally, for each $\Sigma \in |\mathbb{S}|$,

$X \in |\text{Val}(\Sigma)|$, and $M \in |\text{Str}(\Sigma)|$, let

$$(\text{ext}_\Sigma)_X^M : \text{Val}(\Sigma)(X, U_\Sigma(M)) \rightarrow \text{Val}(\Sigma)(\mathcal{T}_\Sigma(X), U_\Sigma(M))$$

be a function *extending* a value variable valuation β into a term valuation $(\text{ext}_\Sigma)_X^M(\beta)$. Then $(\mathcal{T}, \nu, \text{ext})$ is a *term charter* over $(\mathbb{S}, \text{Val}, \text{Str}, U)$ if the following *variable condition* (V), *renaming condition* (R), and *translation condition* (T) are satisfied:

$$(V) \nu_\Sigma(X); (\text{ext}_\Sigma)_X^M(\beta) = \beta$$

for all $\Sigma \in |\mathbb{S}|$, $\beta \in \text{Val}(\Sigma)(X, U_\Sigma(M))$, i.e., the term valuation over a value variable valuation indeed extends the value variable valuation;

$$(R) \mathcal{T}_\Sigma(\xi); (\text{ext}_\Sigma)_{X_2}^M(\beta) = (\text{ext}_\Sigma)_{X_1}^M(\xi; \beta)$$

for all $\Sigma \in |\mathbb{S}|$, $\xi \in \text{Val}(\Sigma)(X_1, X_2)$, $\beta \in \text{Val}(\Sigma)(X_2, U_\Sigma(M))$, i.e., the “renaming lemma” holds; and

$$(T) \mathcal{T}_\sigma(X'); \text{Val}(\sigma)((\text{ext}_{\Sigma'}^M)_{X'}^{M'}(\beta')) = (\text{ext}_\Sigma)_{\text{Val}(\sigma)(X')}^{\text{Str}(\sigma)(M')}(\text{Val}(\sigma)(\beta'))$$

for all $\sigma \in \mathbb{S}(\Sigma, \Sigma')$, $\beta' \in \text{Val}(\Sigma')(X', U_{\Sigma'}(M'))$, i.e., the “translation lemma” holds.

This notion of term charters in fact directly translates the properties of concrete evaluation as discussed in Sect. 2 into an abstract framework: Assume that for each $\Sigma \in |\mathbb{S}|$ there is a faithful functor $\mathcal{U}_\Sigma : \text{Val}(\Sigma) \rightarrow \text{Set}$, i.e., each $\text{Val}(\Sigma)$ is a concrete category. Writing X for $\mathcal{U}_\Sigma(X)$ when $X \in |\text{Val}(\Sigma)|$, ξ for $\mathcal{U}_\Sigma(\xi)$ when $\xi \in \text{Val}(\Sigma)(X_1, X_2)$, $-\sigma$ for both $\text{Str}(\sigma)(-)$ and $\text{Val}(\sigma)(-)$, $\llbracket t \rrbracket_{(\Sigma, X)}$ for $(\text{ext}_\Sigma)_X^M(\beta)(t)$, $\xi(t)$ for $\mathcal{T}_\Sigma(\xi)(t)$, and $\sigma(t)$ for $\mathcal{T}_\sigma(X')(t)$, the conditions become

$$(V) \llbracket x \rrbracket_{(\Sigma, X)}(M, \beta) = \beta(x);$$

$$(R) \llbracket \xi(t) \rrbracket_{(\Sigma, X_2)}(M, \beta) = \llbracket t \rrbracket_{(\Sigma, X_1)}(M, \beta \circ \xi);$$

$$(T) (\llbracket \sigma(t) \rrbracket_{(\Sigma', X')}^{M'}(M', \beta')) | \sigma = \llbracket t \rrbracket_{(\Sigma, X' | \sigma)}(M' | \sigma, \beta' | \sigma).$$

Example 4. (1) For many-sorted algebras as described in Sect. 2, $(\mathcal{T}^m, \nu^m, \text{ext}^m)$ with $\nu_\Sigma^m(X) = 1_X : X \hookrightarrow \mathcal{T}_\Sigma^m(X)$ and $(\text{ext}_\Sigma^m)_X^M(\beta) = \llbracket - \rrbracket_{(\Sigma, X)}^m(M, \beta)$ forms a $(\mathbb{S}^m, \text{Val}^m, \text{Str}^m, U^m)$ -term charter. Note that indeed $\mathcal{T}^m : \text{Val}^m \dashrightarrow \text{Val}^m$.

(2) For the iteration extension in Ex. 1, $(\mathcal{T}^{\text{it}}, \nu^{\text{it}}, \text{ext}^{\text{it}})$ with

$$\begin{aligned} \mathcal{T}_\sigma^{\text{it}}(X')_{s'}(c \rightarrow \text{iterate}(i : s; a : s' = e_0 \mid e)) = \\ \mathcal{T}_\sigma^{\text{it}}(X')_{\text{Seq}(s)}(c \rightarrow \text{iterate}(i : \sigma_S(s); a : \sigma_S(s') = \mathcal{T}_\sigma^{\text{it}}(X')_{s'}(e_0) \mid \\ \mathcal{T}_\sigma^{\text{it}}(X' \uplus \{i : \sigma_S(s), a : \sigma_S(s')\})_{s'}(e)) , \end{aligned}$$

$\nu_{\Sigma^\circ}^{\text{it}}(X) = 1_X : X \hookrightarrow \mathcal{T}_\Sigma^{\text{it}}(X)$, and $(\text{ext}_{\Sigma^\circ}^{\text{it}})_X^{M^\circ}(\beta) = \llbracket - \rrbracket_{(\Sigma^\circ, X)}^{\text{it}}(M^\circ, \beta)$ constitutes an $(\mathbb{S}^\circ, \text{Val}^\circ, \text{Str}^\circ, U^\circ)$ -term charter, as the instantiated properties (V_{it}) , (R_{it}) , and (T_{it}) hold by induction on term construction.

(3) Similarly, for the undefinedness extension in Ex. 2, $(\mathcal{T}^u, \nu^u, \text{ext}^u)$ with $\nu_{\Sigma^\circ}^u(X) = 1_X : X \hookrightarrow \mathcal{T}_\Sigma^u(X)$ and $(\text{ext}_{\Sigma^\circ}^u)_X^{M^\circ}(\beta^\dagger) = \llbracket - \rrbracket_{(\Sigma^\circ, X)}^u(M^\circ, \beta^\dagger)$ constitutes an $(\mathbb{S}^\circ, \text{Val}^\dagger, \text{Str}^\circ, U^\dagger)$ -term charter.

Since \mathcal{T} is a lax indexed functor, the renaming condition (R) and the translation condition (T) can be equivalently combined into a single *context condition* on term charters for expressing that “evaluation is invariant under change of notation”:

$$(C) \quad \mathcal{T}_\sigma(\xi); Val(\sigma)((ext_{\Sigma'}^M)_{X'}^{M'}(\beta')) = (ext_\Sigma)^{Str(\sigma)(M')}(\xi; Val(\sigma)(\beta'))$$

for all $\sigma \in \mathbb{S}(\Sigma, \Sigma')$, $\xi \in Val(\Sigma)(X, Val(\sigma)(X'))$, and $\beta' \in Val(\Sigma')(X', U_{\Sigma'}(M'))$, where $\mathcal{T}_\sigma(\xi) = \mathcal{T}_\Sigma(\xi)$; $\mathcal{T}_\sigma(X')$.

3.3 Obtaining a Term Charter from a Term Structure Constructor

In the many-sorted case, the terms themselves form an algebra. In particular, for each $\Sigma \in |\mathbb{S}^m|$, there is a functor $\mathcal{S}_\Sigma^m : Val^m(\Sigma) \rightarrow Str^m(\Sigma)$ that is left adjoint to the underlying functor U_Σ^m with a unit natural transformation $\eta_\Sigma : 1_{Val^m(\Sigma)} \rightarrow \mathcal{S}_\Sigma^m; U_\Sigma^m$ and a unique lifting $\beta_{\Sigma, X}^{\#m, M} : \mathcal{S}_\Sigma^m(X) \rightarrow M$ in $Str^m(\Sigma)$ for each $\beta : X \rightarrow U_\Sigma^m(M)$ in $Val^m(\Sigma)$. This adjunction, where \mathcal{S}^m is a *term structure constructor*, can equivalently be used to obtain the many-sorted term charter of Ex. 4.

Indeed, for a term charter domain $(\mathbb{S}, Val, Str, U)$, let $(\mathcal{S}, U, \eta, (-)^\#)$ be an $|\mathbb{S}|$ -wise adjunction for the family of functors $\mathcal{S} = (\mathcal{S}_\Sigma : Val(\Sigma) \rightarrow Str(\Sigma))_{\Sigma \in |\mathbb{S}|}$, the family of natural transformations $\eta = (\eta_\Sigma : 1_{Val(\Sigma)} \rightarrow \mathcal{S}_\Sigma; U_\Sigma)_{\Sigma \in |\mathbb{S}|}$, and the liftings $(-)_2^{\#M, -1} : Val(\Sigma)(-1, U_\Sigma(M)) \rightarrow Str(\Sigma)(\mathcal{S}_\Sigma(-1), M)$ for each $\Sigma \in |\mathbb{S}|$ and $M \in |Str(\Sigma)|$. Then, for each $\sigma \in \mathbb{S}(\Sigma, \Sigma')$, there is a natural transformation $\mathcal{S}_\sigma : Val(\sigma); \mathcal{S}_\Sigma \rightarrow \mathcal{S}_{\Sigma'}; Str(\sigma)$ defined by $\mathcal{S}_\sigma(X') = (Val(\sigma)(\eta_{\Sigma'}(X')))^{\#Str(\sigma)(\mathcal{S}_{\Sigma'}(X'))}_{\Sigma, Val(\sigma)(X')}$ such that $\mathcal{S} : Val \rightarrow Str$ becomes a lax indexed functor, for which

$$(*) \quad (\xi; Val(\sigma)(\beta'))^{\#Str(\sigma)(M')}_{\Sigma, X} = \mathcal{S}_\Sigma(\xi); \mathcal{S}_\sigma(X'); Str(\sigma)(\beta'^{\#M', X'})$$

is satisfied for $\xi \in Val(\Sigma)(X, Val(\sigma)(X'))$. In order to construct a term charter (\mathcal{T}, η, ext) , define the term constructor $\mathcal{T} : Val \rightarrow Val$ by $\mathcal{T}_\Sigma = \mathcal{S}_\Sigma; U_\Sigma$, $\mathcal{T}_\sigma = \mathcal{S}_\sigma; U_\Sigma$ and the evaluation morphisms as $(ext_\Sigma)^M_X(\beta) = U_\Sigma(\beta^{\#M, X}) : \mathcal{T}_\Sigma(X) \rightarrow U_\Sigma(M)$. The embedding $\eta : 1_{Val} \rightarrow \mathcal{T}$ forms a lax indexed natural transformation, since $\eta_\Sigma(Val(\sigma)(X')); U_\Sigma(\mathcal{S}_\sigma(X')) = Val(\sigma)(\eta_{\Sigma'}(X'))$. Term charter requirements (V) and (C) hold by expanding the definitions and using (*).

3.4 Term Translations and Substitutions in Term Charters

The term translation $\mathcal{T}_\sigma^m(X') : \mathcal{T}_\Sigma^m(Val^m(\sigma)(X')) \rightarrow Val^m(\sigma)(\mathcal{T}_{\Sigma'}^m(X'))$ along a signature translation $\sigma : \Sigma = (S, F) \rightarrow \Sigma' = (S', F')$ used by the many-sorted term charter operates on the terms over the σ -reduct of the value variables $X' \in |Val^m(\Sigma')|$. It may seem more natural (and is indeed assumed in context institutions, see Sect. 5.2) to have a term translation work on terms over an $X \in |Val^m(\Sigma)|$. This can be achieved via a map $\mathcal{T}_\Sigma^{m, \sigma}(X) : \mathcal{T}_\Sigma^m(X) \rightarrow$

$Val^m(\sigma)(\mathcal{T}_{\Sigma'}^m(\sigma^{Val^m}(X)))$ when defining $\sigma^{Val^m}((X_s)_{s \in S}) = (\biguplus\{X_s \mid \sigma_S(s) = s'\})_{s' \in S'}$ and setting $\mathcal{T}^{m,\sigma}(X)_s(x) = \eta_\sigma^m(X)_s(x)$ for the canonical injection $\eta_\sigma^m(X) : X \hookrightarrow Val^m(\sigma)(\sigma^{Val^m}(X))$ with $\eta_\sigma^m(X)_s(x) = x$. In particular, the functor $\sigma^{Val^m} : Val^m(\Sigma) \rightarrow Val^m(\Sigma')$ with $\sigma^{Val^m}((\xi_s)_{s \in S} : X_1 \rightarrow X_2) = (\biguplus\{\xi_s \mid \sigma_S(s) = s'\})_{s' \in S'}$ is left adjoint to the reduct functor $Val^m(\sigma)$ (cf. [2, Prop. 2.1]). Hence, both translation approaches are equivalent.

For the general case of a term charter (\mathcal{T}, ν, ext) over a term charter domain $(\mathbb{S}, Val, Str, U)$ with adjunctions $(\eta_\sigma, \kappa_\sigma) : \sigma^{Val} \dashv Val(\sigma)$ for each signature morphism σ , we obtain natural transformations $\mathcal{T}^\sigma = (\eta_\sigma^*, \mathcal{T}_\Sigma); (\sigma^{Val*}, \mathcal{T}_\sigma) : \mathcal{T}_\Sigma \dashv \sigma^{Val*}; \mathcal{T}_{\Sigma'}; Val(\sigma)$ such that the translation condition (T) becomes

$$(T_{\downarrow}) \quad \mathcal{T}^\sigma(X'); Val(\sigma)((ext_{\Sigma'}^M)_{\sigma^{Val}(X)}^{M'}(\beta')) = (ext_{\Sigma}^{Str(\sigma)(M')})_X(Val(\sigma)(\beta'); \eta_\sigma(X)) .$$

Moreover, term construction for many-sorted algebras is idempotent up to isomorphism, i.e., it holds that $\mathcal{T}_\Sigma^m(\mathcal{T}_\Sigma^m(X)) \cong \mathcal{T}_\Sigma^m(X)$; this, in fact, applies to the iteration and the undefinedness extensions as well. In particular, for a term substitution $\theta : X \rightarrow \mathcal{T}_\Sigma^m(Y)$ assigning the value variables in $X \in |Val^m(\Sigma)|$ full terms over $Y \in |Val^m(\Sigma)|$ and not only value variables, the ‘‘substitution lemma’’

$$\llbracket \theta_s(t) \rrbracket_{(\Sigma, Y)}^m(M, \beta)_s = \llbracket t \rrbracket_{(\Sigma, X)}^m(M, \llbracket - \rrbracket_{(\Sigma, Y)}^m(M, \beta) \circ \theta)_s$$

holds for all $t \in \mathcal{T}_\Sigma^m(X)_s$ and $\beta : Y \rightarrow U_\Sigma^m(M)$, where $\mathcal{T}_\Sigma^m(\theta)$ is abbreviated by θ .

The idempotency property may also be obtained for a general term charter (\mathcal{T}, ν, ext) when requiring that there is a lax natural transformation $\mu : \mathcal{T}; \mathcal{T} \dashv \mathcal{T}$ for flattening nested term constructions such that (\mathcal{T}, ν, μ) is an indexed monad [11]. A *substitution condition* on the term charter can then be expressed as

$$\mathcal{T}_\Sigma(\theta); \mu_\Sigma(Y); (ext_\Sigma)_Y^M(\beta) = (ext_\Sigma)_X^M(\theta; (ext_\Sigma)_Y^M(\beta))$$

for all $\theta \in Val(\Sigma)(X, \mathcal{T}_\Sigma(Y))$, $\beta \in Val(\Sigma)(Y, U_\Sigma(M))$.

4 Alternative Term Charter Definitions

As demonstrated in Sect. 3.3, the evaluation maps ext_Σ of a term charter (\mathcal{T}, ν, ext) resemble the liftings $(-)^{\sharp_X}$ of an adjunction $(\mathcal{S}, U, \eta, (-)^\sharp)$ with $\mathcal{S} : Val \dashv Str$; the palpable difference is that, for $\beta \in Val(\Sigma)(X, U_\Sigma(M))$, $(ext_\Sigma)_X^M(\beta) \in Val(\Sigma)(\mathcal{T}_\Sigma(X), U_\Sigma(M))$ whereas $\beta^{\sharp_{\Sigma, X}} \in Str(\Sigma)(\mathcal{S}_\Sigma(X), M)$. The situation of a term charter is thus asymmetric, and ext_Σ only affects the first parameter of the hom-functor $Val(\Sigma)(-, -)$. Lawvere [7] (see [8, Exc. IV.1.2]) observed that the symmetric constellation of an adjunction $(\eta, \kappa) : F \dashv G$ between the functors $F : C \rightarrow D$ and $G : D \rightarrow C$ can be equivalently expressed by an isomorphism $H : F \downarrow 1_D \rightarrow 1_C \downarrow G$ of comma categories such that the following

diagram commutes, with π_i the projections:

$$\begin{array}{ccc}
F \downarrow 1_D & \begin{array}{c} \xrightarrow{H} \\ \xleftarrow{H^{-1}} \end{array} & 1_C \downarrow G \\
\searrow \pi_1 \times \pi_2 & & \swarrow \pi_1 \times \pi_2 \\
& C \times D &
\end{array}$$

We transfer this observation to term charters in Sect. 4.1 where we capture the evaluation maps into a single indexed functor $ext : 1_{Val} \downarrow U \rightarrow \mathcal{T} \downarrow U$ and thus obtain a quite succinct characterisation using indexed comma categories (briefly summarised in App. A.2).

On the other hand, the interface of a term in a term charter is given by its signature Σ and its value variables X ; see condition (C). It thus seems natural to combine these two into a single entity $\langle \Sigma, X \rangle$ and to consider terms to be constructed over the combined interface. This is afforded by an application of the Grothendieck construction (briefly recapitulated in App. A.3) turning the indexed category Val into a flat category $\mathcal{G}(Val)$ with objects $\langle \Sigma, X \rangle$. The structures M and the valuations β also belong together, and can be combined into $Str^{\mathcal{G}} = \mathcal{G}(1_{Val}) \downarrow U$ with objects (M, β) . In Sect. 4.2 we thus again obtain a more compact equivalent presentation of term charters where the evaluation maps can be comprised into a single indexed functor $ext^{\mathcal{G}} : Str^{\mathcal{G}} \rightarrow \mathcal{G}(\mathcal{T})^{op}; Str^{\mathcal{G}}$.

4.1 Characterising Term Charters with Comma Categories

Let $(\mathbb{S}, Val, Str, U)$ be a term charter domain. Let $\mathcal{T} : Val \rightarrow Val$ be a lax indexed functor, $\nu : 1_{Val} \rightarrow \mathcal{T}$ a lax indexed natural transformation, and $ext : 1_{Val} \downarrow U \rightarrow \mathcal{T} \downarrow U$ an \mathbb{S} -indexed functor. Then (\mathcal{T}, ν, ext) is a term charter over $(\mathbb{S}, Val, Str, U)$ if, and only if, the following diagram commutes:

$$\begin{array}{ccc}
1_{Val} \downarrow U & \xrightarrow{ext} & \mathcal{T} \downarrow U \\
\searrow 1_{1_{Val} \downarrow U} & & \downarrow \nu \downarrow U \\
& & 1_{Val} \downarrow U
\end{array}$$

where the \mathbb{S} -indexed functor $\nu \downarrow U : \mathcal{T} \downarrow U \rightarrow 1_{Val} \downarrow U$ is given by $(\nu \downarrow U)_{\Sigma}(X, \beta^{\natural}, M) = (X, \nu_{\Sigma}(X); \beta^{\natural}, M)$ and $(\nu \downarrow U)_{\Sigma}(\xi, \mu) = (\xi, \mu)$.

Indeed, $ext; (\nu \downarrow U) = 1_{1_{Val} \downarrow U}$ yields that $\pi_1 \times \pi_2 = ext; (\pi_1 \times \pi_2)$ holds. Thus, ext induces the family of functions $(ext_{\Sigma})_{X'}^M : Val(\Sigma)(X, U_{\Sigma}(M)) \rightarrow Val(\Sigma)(\mathcal{T}_{\Sigma}(X), U_{\Sigma}(M))$ which, for each $\Sigma \in |\mathbb{S}|$ and $M \in |Str(\Sigma)|$, forms a natural transformation in $X \in |Val(\Sigma)|$, such that the renaming condition (R) holds. Furthermore, $(X, \beta, M) = (ext_{\Sigma}; (\nu \downarrow U)_{\Sigma})(X, \beta, M) = (X, \nu_{\Sigma}(X); (ext_{\Sigma})_{X'}^M(\beta), M)$ yields the variable condition (V). Finally, by the indexedness of ext it holds that $ext_{\Sigma'}; (\mathcal{T} \downarrow U)(\sigma) = (1_{Val} \downarrow U)(\sigma); ext_{\Sigma}$, which is the translation condition (T), as

$$\begin{aligned}
(\mathcal{T} \downarrow U)(\sigma)(ext_{\Sigma'}(X', \beta', M')) &= (\mathcal{T} \downarrow U)(\sigma)(X', (ext_{\Sigma'})_{X'}^{M'}(\beta'), M') = \\
&= (Val(\sigma)(X'), \mathcal{T}_{\sigma}(X'); Val(\sigma)((ext_{\Sigma'})_{X'}^{M'}(\beta')), Str(\sigma)(M')) \quad \text{and}
\end{aligned}$$

$$\begin{aligned} \text{ext}_{\Sigma}((1_{\text{Val}} \downarrow U)(\sigma)(X', \beta', M')) &= \text{ext}_{\Sigma}(\text{Val}(\sigma)(X'), \text{Val}(\sigma)(\beta'), \text{Str}(\sigma)(M')) = \\ &= (\text{Val}(\sigma)(X'), (\text{ext}_{\Sigma}^{\text{Str}(\sigma)(M')}_{\text{Val}(\sigma)(X')}(\text{Val}(\sigma)(\beta')), \text{Str}(\sigma)(M')) . \end{aligned}$$

4.2 Characterising Term Charters with Grothendieck Categories

Using the Grothendieck construction for a term charter domain $(\mathbb{S}, \text{Val}, \text{Str}, U)$ we obtain the $\mathcal{G}(\text{Val})$ -indexed category $\text{Str}^{\mathcal{G}} = \mathcal{G}(1_{\text{Val}}) \downarrow U$ such that $\mathcal{G}(\mathcal{T})^{\text{op}}; \text{Str}^{\mathcal{G}} = \mathcal{G}(\mathcal{T}) \downarrow U$ and $\mathcal{G}(\nu)^{\text{op}}; \text{Str}^{\mathcal{G}} = \mathcal{G}(\nu) \downarrow U$. A term charter $(\mathcal{T}, \nu, \text{ext})$ over the term charter domain $(\mathbb{S}, \text{Val}, \text{Str}, U)$ then yields a $\mathcal{G}(\text{Val})$ -indexed functor $\text{ext}^{\mathcal{G}} : \text{Str}^{\mathcal{G}} \rightarrow \mathcal{G}(\mathcal{T})^{\text{op}}; \text{Str}^{\mathcal{G}}$ with $\text{ext}_{\langle \Sigma, X \rangle}^{\mathcal{G}}(M, \beta) = (M, (\text{ext}_{\Sigma}^M(\beta)) \in |\text{Str}^{\mathcal{G}}(\langle \Sigma, \mathcal{T}_{\Sigma}(X) \rangle)|$ and $\text{ext}_{\langle \Sigma, X \rangle}^{\mathcal{G}}(\mu) = \mu$ such that indeed for $\langle \sigma, \xi \rangle : \langle \Sigma, X \rangle \rightarrow \langle \Sigma', X' \rangle$

$$\begin{aligned} \text{Str}^{\mathcal{G}}(\mathcal{G}(\mathcal{T})(\langle \sigma, \xi \rangle))(\text{ext}_{\langle \Sigma', X' \rangle}^{\mathcal{G}}(M', \beta')) &= \\ &= (\text{Str}(\sigma)(M'), \mathcal{T}_{\sigma}(\xi); \text{Val}(\sigma)((\text{ext}_{\Sigma'}^{M'}(\beta')))) \stackrel{(C)}{=} \\ &= (\text{Str}(\sigma)(M'), (\text{ext}_{\Sigma}^{\text{Str}(\sigma)(M')}(\xi; \text{Val}(\sigma)(\beta')))) = \\ &= \text{ext}_{\langle \Sigma, X \rangle}^{\mathcal{G}}(\text{Str}^{\mathcal{G}}(\langle \sigma, \xi \rangle)(M', \beta')) . \end{aligned}$$

Furthermore, it holds that $\text{ext}^{\mathcal{G}}; (\mathcal{G}(\nu)^{\text{op}}; \text{Str}^{\mathcal{G}}) = 1_{\text{Str}^{\mathcal{G}}}$, since

$$(\text{ext}^{\mathcal{G}}; (\mathcal{G}(\nu)^{\text{op}}; \text{Str}^{\mathcal{G}}))_{\langle \Sigma, X \rangle} = \text{ext}_{\langle \Sigma, X \rangle}^{\mathcal{G}}; (\text{Str}^{\mathcal{G}}(\mathcal{G}(\nu)(\langle \Sigma, X \rangle))) \stackrel{(V)}{=} 1_{\text{Str}^{\mathcal{G}}(\langle \Sigma, X \rangle)} .$$

Conversely, given an $\mathcal{G}(\text{Val})$ -indexed functor $\text{ext}^{\mathcal{G}} : \text{Str}^{\mathcal{G}} \rightarrow \mathcal{G}(\mathcal{T})^{\text{op}}; \text{Str}^{\mathcal{G}}$ (i.e., (C) holds), which satisfies $\text{ext}^{\mathcal{G}}; (\mathcal{G}(\nu)^{\text{op}}; \text{Str}^{\mathcal{G}}) = 1_{\text{Str}^{\mathcal{G}}}$ (i.e., (V) holds), we can reconstruct the evaluation morphisms by setting $(\text{ext}_{\Sigma}^M(\beta)) = \beta^{\natural}$ for $\text{ext}_{\langle \Sigma, X \rangle}^{\mathcal{G}}(M, \beta) = (M^{\natural}, \beta^{\natural})$. Indeed, we obtain $M = M^{\natural}$ since $(M, \beta) = \text{Str}^{\mathcal{G}}(\mathcal{G}(\nu)(\langle \Sigma, X \rangle))(M^{\natural}, \beta^{\natural}) = (M^{\natural}, \nu_{\Sigma}(X); \beta^{\natural})$.

Summarising, a term charter is equivalently given by a lax indexed functor $\mathcal{T} : \text{Val} \rightarrow \text{Val}$, a lax indexed natural transformation $\nu : 1_{\text{Val}} \rightarrow \mathcal{T}$, and a $\mathcal{G}(\text{Val})$ -indexed functor $\text{ext}^{\mathcal{G}} : \text{Str}^{\mathcal{G}} \rightarrow \mathcal{G}(\mathcal{T})^{\text{op}}; \text{Str}^{\mathcal{G}}$ such that the following diagram commutes:

$$\begin{array}{ccc} \text{Str}^{\mathcal{G}} & \xrightarrow{\text{ext}^{\mathcal{G}}} & \mathcal{G}(\mathcal{T})^{\text{op}}; \text{Str}^{\mathcal{G}} \\ & \searrow_{1_{\text{Str}^{\mathcal{G}}}} & \downarrow \mathcal{G}(\nu)^{\text{op}}; \text{Str}^{\mathcal{G}} \\ & & \text{Str}^{\mathcal{G}} \end{array}$$

We write $|\text{ext}_{\langle \Sigma, X \rangle}^{\mathcal{G}}(M, \beta)|$ for β^{\natural} when $\text{ext}_{\langle \Sigma, X \rangle}^{\mathcal{G}}(M, \beta) = (M^{\natural}, \beta^{\natural})$.

5 Constructing Institutions from a Term Charter

Institutions [5,3] provide an abstract framework for studying logical systems. Term charters concentrate on the evaluation of terms yielding values rather than

the satisfaction of formulæ. Still, as shown in Sect. 5.1, if there is a notion of truth values in the structures of a term charter domain, an institution can be derived from a term charter, thus embedding term charters into the landscape of logical systems. This construction resembles the generation of an institution from a charter [4], which has been the original reason for the choice of the name “term charters” in [6]. It turns out that also a context institution can be obtained, as introduced by Pawłowski [12] for capturing open formulæ in the framework of institutions; after summarising the definition of context institutions, we demonstrate their relationship with term charters in Sect. 5.2.

5.1 Institutions

An *institution* $(\mathbb{S}, Str, Sen, \models)$ is given by a category \mathbb{S} of *signatures*; an indexed category $Str : \mathbb{S}^{\text{op}} \rightarrow \text{Cat}$ of *structures*; a *sentence functor* $Sen : \mathbb{S} \rightarrow \text{Set}$; and a family $\models = (\models_{\Sigma} \subseteq |Str(\Sigma)| \times Sen(\Sigma))_{\Sigma \in |\mathbb{S}|}$ of *satisfaction relations*, such that for all $\sigma \in \mathbb{S}(\Sigma, \Sigma')$, $M' \in |Str(\Sigma')|$, and $\varphi \in Sen(\Sigma)$ the *satisfaction condition* holds:

$$Str(\sigma)(M') \models_{\Sigma} \varphi \iff M' \models_{\Sigma'} Sen(\sigma)(\varphi).$$

For constructing an institution from a term charter $\mathfrak{T} = (\mathcal{T}, \nu, ext)$ over a term charter domain $D = (\mathbb{S}, Val, Str, U)$ we assume that D is *logical* via a family of functors $\mathcal{F}^* = (\mathcal{F}_{\Sigma}^* : Val(\Sigma) \rightarrow \text{Set})_{\Sigma \in |\mathbb{S}|}$ yielding *formulæ* and being equipped with *truth values* $*_{\Sigma}^M \subseteq \mathcal{F}_{\Sigma}^*(U_{\Sigma}(M))$ for each $\Sigma \in |\mathbb{S}|$ and $M \in |Str(\Sigma)|$ such that $Val(\sigma); \mathcal{F}_{\Sigma}^* = \mathcal{F}_{\Sigma'}^*$, and $*_{\Sigma}^{Str(\sigma)(M')} = *_{\Sigma'}^{M'}$ for all $\sigma : \Sigma \rightarrow \Sigma'$ in \mathbb{S} and $M' \in |Str(\Sigma')|$.

Example 5. (1) For the term charter domain $(\mathbb{S}^m, Val^m, Str^m, U^m)$, we may base the formulæ on equations between value variables, representing $x_1 = x_2$ as the pair $(x_1, x_2) : \mathcal{F}_{(S,F)}^{*m}((X_s)_{s \in S}) = (X_s \times X_s)_{s \in S}$, $\mathcal{F}_{(S,F)}^{*m}((\xi_s)_{s \in S}) = (\xi_s \times \xi_s)_{s \in S}$, and $*_{(S,F)}^{m,M} = (\{(v, v) \mid (v, v) \in U_{(S,F)}^m(M)_s\})_{s \in S}$. In fact, this construction can be applied to all term charters domains with a concrete category of value variables.

(2) For the term charter domain $(\mathbb{S}^{\circ}, Val^{\circ}, Str^{\circ}, U^{\circ})$, every signature $\Sigma^{\circ} \in |\mathbb{S}^{\circ}|$ shows the predefined sort Bool that is interpreted by $\text{Bool}^{M^{\circ}} = \{\text{ff}, \text{tt}\}$ in every structure $M^{\circ} \in |Str^{\circ}(\Sigma^{\circ})|$. Thus we can choose $\mathcal{F}_{\Sigma^{\circ}}^{*\circ}(X) = X_{\text{Bool}}$ and $*_{\Sigma^{\circ}}^{\circ, M^{\circ}} = \{\text{tt}\}$.

(3) For the term charter domain $(\mathbb{S}^{\circ}, Val^{\dagger}, Str^{\circ}, U^{\dagger})$, we may again choose $\mathcal{F}_{\Sigma^{\circ}}^{*\dagger}(X) = X_{\text{Bool}}$, although now also the value \dagger for undefinedness will be contained. We may also again set $*_{\Sigma^{\circ}}^{\dagger, M^{\circ}} = \{\text{tt}\}$ since $\mathcal{F}_{\Sigma^{\circ}}^{*\dagger}(U_{\Sigma^{\circ}}^{\dagger}(M^{\circ})) = \{\dagger, \text{ff}, \text{tt}\}$.

Based on Sect. 4.2 we obtain an institution $(\mathbb{S}^{\triangleright}, Str^{\triangleright}, Sen^{\triangleright}, \models^{\triangleright})$ over \mathfrak{T} and \mathcal{F}^* (where the superscript \triangleright abbreviates $\mathfrak{T}, \mathcal{F}^*$) as follows: The category of signatures $\mathbb{S}^{\triangleright}$ is defined to be $\mathcal{G}(Val)$; the indexed category $Str^{\triangleright} : (\mathbb{S}^{\triangleright})^{\text{op}} \rightarrow \text{Cat}$ of structures is defined to be $Str^{\mathcal{G}}$; the sentence functor $Sen^{\triangleright} : \mathbb{S}^{\triangleright} \rightarrow \text{Set}$ is defined as $Sen^{\triangleright}(\langle \Sigma,$

$X\rangle) = \mathcal{F}_\Sigma^*(\mathcal{T}_\Sigma(X))$ and $Sen^\triangleright(\langle\sigma, \xi\rangle) = \mathcal{F}_\Sigma^*(\mathcal{T}_\sigma(\xi))$; the family of satisfaction relations $(\models_{\langle\Sigma, X\rangle}^\triangleright \subseteq |Str^\triangleright(\langle\Sigma, X\rangle)| \times |Sen^\triangleright(\langle\Sigma, X\rangle)|)_{\langle\Sigma, X\rangle \in |\mathbb{S}^\triangleright|}$ is defined by

$$(M, \beta) \models_{\langle\Sigma, X\rangle}^\triangleright \varphi \iff \mathcal{F}_\Sigma^*(|ext_{\langle\Sigma, X\rangle}^{\mathcal{G}}(M, \beta)|)(\varphi) \in *_{\Sigma}^M .$$

For the satisfaction condition it suffices to prove that

$$\mathcal{F}_\Sigma^*(|ext_{\langle\Sigma, X\rangle}^{\mathcal{G}}(Str^{\mathcal{G}}(\langle\sigma, \xi\rangle)(M', \beta'))|) = \mathcal{F}_\Sigma^*(\mathcal{T}_\sigma(\xi)); \mathcal{F}_{\Sigma'}^*(|ext_{\langle\Sigma', X'\rangle}^{\mathcal{G}}(M', \beta')|) ,$$

which follows from the term charter requirement (C).

5.2 Context Institutions

Context institutions [12] have been introduced to capture open formulæ in institutions. A context institution, like an ordinary institution, shows a category \mathbb{S} of *signatures* and an indexed category of *structures* $Str : \mathbb{S}^{\text{op}} \rightarrow \text{Cat}$. Instead of the sentences, however, *formulæ* are built by a family of functors $(Frm_\Sigma : Ctxt_\Sigma \rightarrow \text{Set})_{\Sigma \in |\mathbb{S}|}$ over a family of *context* categories $(Ctxt_\Sigma)_{\Sigma \in |\mathbb{S}|}$ that allow to incorporate variables. Contexts are *translated*, i.e., variables are renamed, by a family of functors $(Ctxt_\sigma : Ctxt_\Sigma \rightarrow Ctxt_{\Sigma'})_{\sigma \in \mathbb{S}(\Sigma, \Sigma')}$, and formulæ are translated along a context translation by a family of natural transformations $(Frm_\sigma : Frm_\Sigma \rightarrow Frm_{\Sigma'})_{\sigma \in \mathbb{S}(\Sigma, \Sigma')}$. *Valuations* are given by a family of functors $(Vlt_\Sigma : (Ctxt_\Sigma)^{\text{op}} \times Str(\Sigma) \rightarrow \text{Set})_{\Sigma \in |\mathbb{S}|}$ together with a family of natural transformations $(\sigma^{Vlt} : Vlt_{\Sigma'}(Ctxt_\sigma(-1), -2) \rightarrow Vlt_\Sigma(-1, Str(\sigma)(-2)))_{\sigma \in \mathbb{S}(\Sigma, \Sigma')}$ constructing *adjoint* valuations along signature morphisms, and, for each $\Sigma \in |\mathbb{S}|$, families of natural transformations $(\gamma^{Vlt} : Vlt_\Sigma(\Delta, -) \rightarrow Vlt_\Sigma(\Gamma, -))_{\gamma \in Ctxt_\Sigma(\Gamma, \Delta)}$ *translating* valuations along context translations, such that the *coherence condition* for $\sigma \in \mathbb{S}(\Sigma, \Sigma')$, $\gamma \in Ctxt_\Sigma(\Gamma, \Delta)$, and $M' \in |Str(\Sigma')|$ is satisfied:

$$(1) \quad Ctxt_\sigma(\gamma)^{Vlt}(M'); \sigma^{Vlt}(\Gamma, M') = \sigma^{Vlt}(\Delta, M'); \gamma^{Vlt}(Str(\sigma)(M')) .$$

Finally, context institutions show for all $\Sigma \in |\mathbb{S}|$, $M \in |Str(\Sigma)|$, and $\Gamma \in |Ctxt_\Sigma|$ *satisfaction relations* $M, -1 \models_{\Sigma, \Gamma} -2 \subseteq Vlt_\Sigma(\Gamma, M) \times Frm_\Sigma(\Gamma)$ for which the *substitution condition* (2) and the *satisfaction condition* (3) have to be satisfied:

$$(2) \quad M, v \models_{\Sigma, \Delta} Frm_\Sigma(\gamma)(\phi) \iff M, \gamma^{Vlt}(M)(v) \models_{\Sigma, \Gamma} \phi$$

for all $\Sigma \in |\mathbb{S}|$, $v \in Vlt_\Sigma(\Delta, M)$, $\gamma \in Ctxt_\Sigma(\Gamma, \Delta)$, $\phi \in Frm_\Sigma(\Gamma)$;

$$(3) \quad M', v' \models_{\Sigma', Ctxt_\sigma(\Gamma)} Frm_\sigma(\Gamma)(\phi) \iff Str(\sigma)(M'), \sigma^{Vlt}(\Gamma, M')(v') \models_{\Sigma, \Gamma} \phi$$

for all $\sigma \in \mathbb{S}(\Sigma, \Sigma')$, $v' \in Vlt_{\Sigma'}(Ctxt_\sigma(\Gamma), M')$, $\phi \in Frm_\Sigma(\Gamma)$.

Summarising, a context institution can be represented as the tuple $(\mathbb{S}, Str, Ctxt, Frm, Vlt, -^{Vlt}, \models)$. In fact, this account (besides calling the valuations *Vlt* instead of *Val*) omits some parts of the original definition in [12] that has been formulated somewhat more concretely: For contexts and formulæ categories with an inclusion system are used and it is required that an indexed set of *carriers* can be computed from a structure as well as a sorted set of *variables* from a

context. Then, the valuations can be represented as a concrete hom-set and the adjoint valuations can be derived.

Let $\mathfrak{T} = (\mathcal{T}, \nu, ext)$ be a term charter over a logical term charter domain $(\mathbb{S}, Val, Str, U)$ with formulæ functor \mathcal{F}^* , as above. For obtaining a context institution $(\mathbb{S}^\triangleright, Str^\triangleright, Ctxt^\triangleright, Frm^\triangleright, Vlt^\triangleright, -^{Vlt^\triangleright}, \models^\triangleright)$ from $\mathfrak{T}, \mathcal{F}^*$ it is straightforward to choose $\mathbb{S}^\triangleright = \mathbb{S}$ and $Str^\triangleright = Str$, as well as $Ctxt_\Sigma^\triangleright = Val(\Sigma)$, $Frm_\Sigma^\triangleright = \mathcal{T}_\Sigma; \mathcal{F}_\Sigma^*$, and $Vlt_\Sigma^\triangleright = Val(\Sigma)(-1, U_\Sigma(-2))$ for each $\Sigma \in |\mathbb{S}|$, as the value variables correspond to the notion of context. However, the context translation functors $Ctxt_\sigma^\triangleright : Ctxt_\Sigma^\triangleright \rightarrow Ctxt_{\Sigma'}^\triangleright$ along $\sigma : \Sigma \rightarrow \Sigma'$ have to be defined as going in the opposite direction than $Val(\sigma) : Val(\Sigma') \rightarrow Val(\Sigma)$. We thus have to require that there is an adjunction $(\eta_\sigma, \kappa_\sigma) : \sigma^{Val} \dashv Val(\sigma)$, as discussed in Sect. 3.4, and then can define $Ctxt_\sigma^\triangleright = \sigma^{Val}$ and $Frm_\sigma^\triangleright = \mathcal{T}^\sigma; \mathcal{F}_\Sigma^*$ with $\mathcal{T}^\sigma = (\eta_\sigma^*; \mathcal{T}_\Sigma); (\sigma^{Val*}; \mathcal{T}_\sigma) : \mathcal{T}_\Sigma \dashv \sigma^{Val}; \mathcal{T}_{\Sigma'}; Val(\sigma)$; for each $\sigma : \Sigma \rightarrow \Sigma'$ in $\mathbb{S}^\triangleright$ define $\sigma^{Vlt^\triangleright}(X, M')(\beta') = \eta_\sigma(X); Val(\sigma)(\beta')$, and for each $\xi : X_1 \rightarrow X_2$ in $Ctxt_\Sigma^\triangleright$ define $\xi^{Vlt^\triangleright}(M)(\beta) = \xi; \beta$. The coherence condition (1) then reads

$$\eta_\sigma(X_1); Val(\sigma)(\sigma^{Val}(\xi); \beta') = \xi; \eta_\sigma(X_2); Val(\sigma)(\beta') ,$$

which is evident by the naturality of η_σ . For the satisfaction relations we finally can define

$$M, \beta \models_{\Sigma, X}^\triangleright \phi \iff \mathcal{F}_\Sigma^*((ext_\Sigma)_X^M(\beta))(\phi) \in *_\Sigma^M ;$$

the substitution condition (2) and the satisfaction condition (3) then become

$$\begin{aligned} M, \beta \models_{\Sigma, X_2}^\triangleright \mathcal{F}_\Sigma^*(\mathcal{T}_\Sigma(\xi))(\phi) &\iff M, \xi; \beta \models_{\Sigma, X_1} \phi \quad \text{and} \\ M', \beta' \models_{\Sigma', \sigma^{Val}(X)}^\triangleright \mathcal{F}_{\Sigma'}^*(\mathcal{T}^\sigma(X))(\phi) &\iff Str(\sigma)(M'), \eta_\sigma(X); Val(\sigma)(\beta') \models_{\Sigma, X}^\triangleright \phi , \end{aligned}$$

and follow from the conditions (R) and (T₋) of term charters.

6 Morphisms for Term Charter Domains and Term Charters

For institutions there are four basic types of relationships, the institution (co-)morphisms and the institution forward (co-)morphisms [14]. For example, an *institution morphism* expresses a kind of projection from a “richer” source to a “poorer” target logic, or, conversely, how the “richer” logic is built over the “poorer” logic. Formally, an institution morphism $\mu = (\mu^{\mathbb{S}}, \mu^{Str}, \mu^{Sen}) : (\mathbb{S}, Str, Sen, \models) \rightarrow (\mathbb{S}', Str', Sen', \models')$ consists of a functor $\mu^{\mathbb{S}} : \mathbb{S} \rightarrow \mathbb{S}'$, an indexed functor $\mu^{Str} : Str \dashv (\mu^{\mathbb{S}})^{op}; Str'$, and a natural transformation $\mu^{Sen} : \mu^{\mathbb{S}}; Sen' \dashv Sen$, such that for all $\Sigma \in |\mathbb{S}|$, $M \in |Str(\Sigma)|$, and $\varphi' \in Sen'(\mu^{\mathbb{S}}(\Sigma))$ the following *satisfaction condition* holds:

$$M \models_\Sigma \mu^{Sen}(\Sigma)(\varphi') \iff \mu_\Sigma^{Str}(M) \models'_{\mu^{\mathbb{S}}(\Sigma)} \varphi' .$$

In fact, the sentence translation μ^{Sen} may be split off, yielding an *institution semi-morphism* $(\mu^{\mathbb{S}}, \mu^{Str})$; an *institution forward morphism* then just reverses the

direction of the sentence translation. Similarly, an *institution semi-co-morphism* has components $\nu^{\mathbb{S}} : \mathbb{S} \rightarrow \mathbb{S}'$ and $\nu^{Str} : (\nu^{\mathbb{S}})^{\text{op}}; Str' \rightarrow Str$; it can be extended into an *institution (forward) co-morphism* with a sentence translation going in the opposite (the same) direction as the structure translation.

For term charters and their term charter domains, a similar family of relationships arises, where the notions of semi-(co-)morphisms applies to term charter domains and the extension to (forward) (co-)morphisms to term charters. We concentrate on projecting from a more complex evaluation framework to a simpler one and show that its definition gives rise to an institution morphism on the institutions constructed in Sect. 5.1.

A *term charter domain morphism* $\mathbf{d} = (\mathbf{d}^{\mathbb{S}}, \mathbf{d}^{Val}, \mathbf{d}^{Str}) : (\mathbb{S}, Val, Str, U) \rightarrow (\mathbb{S}', Val', Str', U')$ is given by a functor $\mathbf{d}^{\mathbb{S}} : \mathbb{S} \rightarrow \mathbb{S}'$ and indexed functors $\mathbf{d}^{Val} : Val \rightarrow (\mathbf{d}^{\mathbb{S}})^{\text{op}}; Val'$ and $\mathbf{d}^{Str} : Str \rightarrow (\mathbf{d}^{\mathbb{S}})^{\text{op}}; Str'$ such that $U; \mathbf{d}^{Val} = \mathbf{d}^{Str}; ((\mathbf{d}^{\mathbb{S}})^{\text{op}}; U')$.

Example 6. There is a term charter domain morphism $\mathbf{d}^{\circ} : (\mathbb{S}^{\circ}, Val^{\circ}, Str^{\circ}, U^{\circ}) \rightarrow (\mathbb{S}^{\text{m}}, Val^{\text{m}}, Str^{\text{m}}, U^{\text{m}})$: $\mathbf{d}^{\circ \mathbb{S}}$ is the injection of \mathbb{S}° into \mathbb{S}^{m} , and also $\mathbf{d}^{\circ Val}$ and $\mathbf{d}^{\circ Str}$ just inject the value variables and structures, respectively.

Let now $\mathfrak{T} = (\mathcal{T}, \nu, ext)$ and $\mathfrak{T}' = (\mathcal{T}', \nu', ext')$ be term charters over the term charter domains $\mathbf{D} = (\mathbb{S}, Val, Str, U)$ and $\mathbf{D}' = (\mathbb{S}', Val', Str', U')$, respectively, and let $\mathbf{d} : \mathbf{D} \rightarrow \mathbf{D}'$ be a term charter domain morphism. A *d-term charter morphism* $\mathbf{m} : \mathfrak{T} \rightarrow \mathfrak{T}'$ is given by a lax natural transformation $\mathbf{m} : \mathbf{d}^{Val}; ((\mathbf{d}^{\mathbb{S}})^{\text{op}}; \mathcal{T}') \rightarrow \mathcal{T}; \mathbf{d}^{Val}$ such that for all $\Sigma \in |\mathbb{S}|$, $X \in |Val(\Sigma)|$, and $\beta \in Val(\Sigma)(X, U_{\Sigma}(M))$ it holds that

$$\begin{aligned} (\mathbf{m}_1) \quad & \nu'_{\mathbf{d}^{\mathbb{S}}(\Sigma)}(\mathbf{d}_{\Sigma}^{Val}(X)); \mathbf{m}_{\Sigma}(X) = \mathbf{d}_{\Sigma}^{Val}(\nu_{\Sigma}(X)); \\ (\mathbf{m}_2) \quad & (ext'_{\mathbf{d}^{\mathbb{S}}(\Sigma)})_{\mathbf{d}_{\Sigma}^{Val}(X)}^{\mathbf{d}_{\Sigma}^{Str}(M)}(\mathbf{d}_{\Sigma}^{Val}(\beta)) = \mathbf{m}_{\Sigma}(X); \mathbf{d}_{\Sigma}^{Val}((ext_{\Sigma})_X^M(\beta)). \end{aligned}$$

Example 7. There is a \mathbf{d}° -term charter morphism $\mathbf{m}^{\circ} : (\mathcal{T}^{\text{it}}, \nu^{\text{it}}, ext^{\text{it}}) \rightarrow (\mathcal{T}^{\text{m}}, \nu^{\text{m}}, ext^{\text{m}})$ where $\mathbf{m}_{\Sigma}^{\circ}(X) : \mathcal{T}_{\mathbf{d}^{\mathbb{S}}(\Sigma)}^{\text{m}}(\mathbf{d}_{\Sigma}^{\circ Val}(X)) \rightarrow \mathbf{d}_{\Sigma}^{\circ Val}(\mathcal{T}_{\Sigma}^{\text{it}}(X))$ is the embedding of many-sorted terms into iteration terms. The relation guarantees that both evaluations coincide on many-sorted terms.

For the Grothendieck representation of term charters in Sect. 4.2, $\mathbf{d} : \mathbf{D} \rightarrow \mathbf{D}'$ induces an indexed functor $\mathbf{d}^{Str^{\mathcal{G}}} : Str^{\mathcal{G}} \rightarrow \mathcal{G}(\mathbf{d}^{Val})^{\text{op}}; Str'^{\mathcal{G}}$ with $\mathbf{d}_{\langle \Sigma, X \rangle}^{Str^{\mathcal{G}}}(M, \beta) = (\mathbf{d}_{\Sigma}^{Str}(M), \mathbf{d}_{\Sigma}^{Val}(\beta))$. Condition (m₁) for $\mathbf{m} : \mathfrak{T} \rightarrow \mathfrak{T}'$ then is $(\mathcal{G}(\mathbf{d}^{Val}); \mathcal{G}((\mathbf{d}^{\mathbb{S}})^{\text{op}}; \nu'))$; $\mathcal{G}(\mathbf{m}) = \mathcal{G}(\nu)$; $\mathcal{G}(\mathbf{d}^{Val})$ and condition (m₂) is $\mathbf{d}^{Str^{\mathcal{G}}}; ext'^{\mathcal{G}} = ext^{\mathcal{G}}; (\mathcal{G}(\mathbf{m})^{\text{op}}; \mathbf{d}^{Str^{\mathcal{G}}})$.

In order to compare this definition of a \mathbf{d} -term charter morphism $\mathbf{m} : \mathfrak{T} \rightarrow \mathfrak{T}'$ with institution morphisms, let the term charter domains \mathbf{D} and \mathbf{D}' be logical with formulæ functors $\mathcal{F}_{\Sigma}^* : Val(\Sigma) \rightarrow \text{Set}$ and $\mathcal{F}_{\Sigma'}^{*'} : Val'(\Sigma') \rightarrow \text{Set}$ such that $\mathcal{F}_{\Sigma}^* = \mathbf{d}^{Val}; \mathcal{F}_{\mathbf{d}^{\mathbb{S}}(\Sigma)}^{*'}$ and $*_{\Sigma}^M = *_{\mathbf{d}^{\mathbb{S}}(\Sigma)}^{\mathbf{d}_{\Sigma}^{Str}(M)}$. We obtain an institution morphism $\mu^{\triangleright} : \mathfrak{I}^{\triangleright} \rightarrow \mathfrak{I}^{\triangleright'}$ for $\mathfrak{I}^{\triangleright} = (\mathbb{S}^{\triangleright}, Str^{\triangleright}, Sen^{\triangleright}, \models^{\triangleright})$ and $\mathfrak{I}^{\triangleright'} = (\mathbb{S}^{\triangleright'}, Str^{\triangleright'}, Sen^{\triangleright'}, \models^{\triangleright'})$ constructed over $\triangleright = \mathfrak{T}, \mathcal{F}^*$ and $\triangleright' = \mathfrak{T}', \mathcal{F}^{*'}$ as in Sect. 5.1 with $\mu^{\triangleright \mathbb{S}} = \mathcal{G}(\mathbf{d}^{Val})$, $\mu^{\triangleright Str} = \mathbf{d}^{Str^{\mathcal{G}}}$, and $\mu^{\triangleright Sen}(\langle \Sigma, X \rangle) = \mathcal{F}_{\mathbf{d}^{\mathbb{S}}(\Sigma)}^{*'}(\mathbf{m}_{\Sigma}(X))$: Using the abbreviations

$\Sigma' = \mathbf{d}^{\mathbb{S}}(\Sigma)$, $X' = \mathbf{d}_{\Sigma}^{\text{Val}}(X)$, $M' = \mathbf{d}_{\Sigma}^{\text{Str}}(M)$, and $\beta' = \mathbf{d}_{\Sigma}^{\text{Val}}(\beta)$, the satisfaction condition

$$(M, \beta) \models_{\langle \Sigma, X \rangle}^{\triangleright} \mathcal{F}_{\Sigma'}^*(\mathbf{m}_{\Sigma}(X))(\varphi') \iff (M', \beta') \models_{\langle \Sigma', X' \rangle}^{\triangleright'} \varphi'$$

for $\varphi' \in \mathcal{F}_{\Sigma'}^*(\mathcal{T}_{\Sigma'}(X'))$ follows from the observation that

$$\begin{aligned} \mathcal{F}_{\Sigma'}^*(\mathbf{m}_{\Sigma}(X)); \mathcal{F}_{\Sigma}^*(|\text{ext}_{\langle \Sigma, X \rangle}^{\mathcal{G}}(M, \beta)|) = \\ \mathcal{F}_{\Sigma'}^*(\mathbf{m}_{\Sigma}(X)); \mathcal{F}_{\Sigma'}^*(\mathbf{d}_{\Sigma}^{\text{Val}}(|\text{ext}_{\langle \Sigma, X \rangle}^{\mathcal{G}}(M, \beta)|)) = \mathcal{F}_{\Sigma'}^*(|\text{ext}_{\langle \Sigma', X' \rangle}^{\mathcal{G}}(M', \beta')|) . \end{aligned}$$

To achieve further accordance with institutions, a *d-term charter forward morphism* must reverse the direction of the term translation. A *term charter domain co-morphism* $\mathbf{p} : \mathbf{D} \rightarrow \mathbf{D}'$ must reverse the direction of the value variables and structures translation, and then can again be extended to a *p-term charter (forward) co-morphism* with term translation in the opposite (the same) direction as structure translation. As for institutions, these morphisms provide for embeddings, projections, and encodings of term charters.

7 Conclusions

We have presented term charter domains and term charters as a framework for general term evaluation such that evaluation is compatible with variable renamings and signature translations. This account simplifies our previous definition in [6] and puts term charters in a more general perspective. Term charter domains and term charters arrange values and variables, structures, term construction and term evaluation, all indexed over signatures, such that “evaluation is invariant under change of notation”. Term structure constructors give rise to term charters and also substitutions can be captured in term charters. We have re-presented term charters using comma and Grothendieck categories to highlight the interface character of signatures and variables. We have demonstrated a tight connection with institutions in general and context institutions in particular; along the lines of institution (semi-)morphisms we have introduced term charter (domain) morphisms.

The use of indexed categories should be complemented by fibrations [1] which may lead to another compact presentation of term charters. From the viewpoint of institutions, on the one hand substitutions in term charters should be connected with the notion of generalised substitutions [3,16] and generalised terms in 2-institutions [2]. On the other hand, an alignment with parchments [4], that focus on directly establishing the satisfaction relation rather than on term evaluations, could be based on the requirement of formulæ and truth values as above which are used for constructing an institution from a term charter. This also applies to Mayoh’s galleries [9] (similar to generalized institutions [4]) and Poigné’s foundations [13] which are based on the overall extension of functions or frames instead of values of terms.

Our original motivation for [6] has been to “institutionalise” OCL by combining small, well-understood expression language fragments, all captured as term

charters, into a bigger whole using a sequencing operator and co-limits. For this purpose, the term charter (domain) morphisms now provide a basis for term charter combinations also over heterogeneous term charter domains. What is most notably missing for an OCL-institution is the possibility to handle pre/post-condition specifications. For the evaluation of the OCL expression for the post-condition, a pre- and a post-state must be present, which necessitates a construction on term charters or a suitable extension such that two structures can be accessed simultaneously.

Acknowledgements. We have profited much from discussions with Till Mossakowski, Rolf Hennicker, and Daniel Calegari. We thank the anonymous referees for their insightful comments and suggestions.

A Categorical Terminology

We briefly summarise the categorical terminology used in the main text; for indexed categories our account is based on [15,17,3,14], for comma categories on [8], and for Grothendieck categories on [17]. We write $*$ for the horizontal composition of natural transformations when using diagrammatic notation.

A.1 Indexed Categories

An *indexed category* N over an index category I is a functor $N : I^{\text{op}} \rightarrow \text{Cat}$.

A *lax indexed functor* $F : M \rightrightarrows N$ between the I -indexed categories M and N is given by families of functors $(F_i : M(i) \rightarrow N(i))_{i \in |I|}$ and natural transformations $(F_u : M(u); F_{i_1} \rightrightarrows F_{i_2}; N(u))_{u \in I(i_1, i_2)}$ with $F_{1_i} = 1_{F_i}$ such that $F_{u_1; u_2} = (M(u_2)^* F_{u_1}); (F_{u_2}^* N(u_1))$ for all $u_1 \in I(i_1, i_2)$, $u_2 \in I(i_2, i_3)$. The composition $F; G : L \rightrightarrows N$ of $F : L \rightrightarrows M$ and $G : M \rightrightarrows N$ for the I -indexed categories L, M, N is given by $(F; G)_i = F_i; G_i$ for $i \in |I|$ and $(F; G)_u = (F_u^* G_{i_1}); (F_{i_2}^* G_u)$ for $u \in I(i_1, i_2)$. An *indexed functor* $F : M \rightarrow N$ is a lax indexed functor $F : M \rightrightarrows N$ with $1_{M(u); F_{i_1}} = F_u = 1_{F_{i_2}; N(u)}$ for all $u \in I(i_1, i_2)$, i.e., a natural transformation from M to N .

For M, N I -indexed categories and $F, G : M \rightrightarrows N$ lax indexed functors, a *lax indexed natural transformation* $\eta : F \rightrightarrows G$ is given by a family of natural transformations $(\eta_i : F_i \rightrightarrows G_i)_{i \in |I|}$ such that $(M(u)^* \eta_{i_1}); G_u = F_u; (\eta_{i_2}^* N(u))$ for $u \in I(i_1, i_2)$. If F and G are both indexed functors, $\eta : F \rightarrow G$ is also called an *indexed natural transformation*; it then has to hold that $M(u)^* \eta_{i_1} = \eta_{i_2}^* N(u)$ for all $u \in I(i_1, i_2)$.

A.2 Comma Categories

Given two functors $F_1 : A_1 \rightarrow C$ and $F_2 : A_2 \rightarrow C$, the *comma category* $F_1 \downarrow F_2$ has as objects the triples (X_1, f, X_2) with $X_1 \in |A_1|$, $X_2 \in |A_2|$, and $f : F_1(X_1) \rightarrow F_2(X_2)$ in C ; and as morphisms from (X_{11}, f_1, X_{21}) to (X_{12}, f_2, X_{22}) the pairs (x_1, x_2) with $x_1 : X_{11} \rightarrow X_{12}$ in A_1 and $x_2 : X_{21} \rightarrow X_{22}$ in A_2

such that $f_1; F_2(x_2) = F_1(x_1); f_2$. It is equipped with the *projection functors* $\pi_i : F_1 \downarrow F_2 \rightarrow A_i$ defined by $\pi_i(X_1, f, X_2) = X_i$ and $\pi_i(x_1, x_2) = x_i$ for $i \in \{1, 2\}$.

For the functors $F_1, G_1 : A_1 \rightarrow C$ and $F_2, G_2 : A_2 \rightarrow C$ consider the functor $H : F_1 \downarrow F_2 \rightarrow G_1 \downarrow G_2$ with $H(X_1, f, X_2) = (X_1, f^H, X_2)$ and $H(x_1, x_2) = (x_1, x_2)$. Then $(F_1(x_1); f_2)^H = G_1(x_1); f_2^H$ and $(f_1; F_2(x_2))^H = f_1^H; G_2(x_2)$ for all $X_i \in |A_i|$, $x_i \in A_i(X_{i1}, X_{i2})$, $f_1 : F_1(X_1) \rightarrow F_2(X_{21})$, and $f_2 : F_1(X_{12}) \rightarrow F_2(X_2)$. Thus, H induces natural transformations

$$\begin{aligned} H^{X_2} : C(F_1(-), F_2(X_2)) &\rightarrow C(G_1(-), G_2(X_2)) , \\ H_{X_1} : C(F_1(X_1), F_2(-)) &\rightarrow C(G_1(X_1), G_2(-)) . \end{aligned}$$

Conversely, a bi-natural transformation $\eta : C(F_1(-), F_2(-)) \rightarrow C(G_1(-), G_2(-))$ induces the functor $\eta : F_1 \downarrow F_2 \rightarrow G_1 \downarrow G_2$ given by $\eta(X_1, f, X_2) = (X_1, \eta(X_1, X_2)(f), X_2)$ and $\eta(x_1, x_2) = (x_1, x_2)$.

The *indexed comma category* $F \downarrow G : I^{\text{op}} \rightarrow \text{Cat}$ for a lax indexed functor $F : M_1 \rightarrow N$ and an indexed functor $G : M_2 \rightarrow N$ over I -indexed categories M_1, M_2 , and N is defined at $i \in |I|$ as the comma category $(F \downarrow G)(i) = F_i \downarrow G_i$, and at $u \in I(i_1, i_2)$ as the functor $(F \downarrow G)(u) : (F \downarrow G)(i_2) \rightarrow (F \downarrow G)(i_1)$ given by

$$\begin{aligned} (F \downarrow G)(u)(O_{21}, p_2, O_{22}) &= (M_1(u)(O_{21}), F_u(O_{21}); N(u)(p_2), M_2(u)(O_{22})) , \\ (F \downarrow G)(u)(o_{21}, o_{22}) &= (M_1(u)(o_{21}), M_2(u)(o_{22})) . \end{aligned}$$

A.3 Grothendieck Categories

The (contravariant) *Grothendieck category* $\mathcal{G}(N)$ over an I -indexed category N has as objects the pairs $\langle i, O \rangle$ with $i \in |I|$ and $O \in |N(i)|$; and as morphisms from $\langle i_1, O_1 \rangle$ to $\langle i_2, O_2 \rangle$ the pairs $\langle u, o \rangle$ with $u \in I(i_1, i_2)$ and $o \in N(i_1)(O_1, N(u)(O_2))$ where the composition of $\langle u_1, o_1 \rangle : \langle i_1, O_1 \rangle \rightarrow \langle i_2, O_2 \rangle$ and $\langle u_2, o_2 \rangle : \langle i_2, O_2 \rangle \rightarrow \langle i_3, O_3 \rangle$ is $\langle u_1, o_1 \rangle; \langle u_2, o_2 \rangle = \langle u_1; u_2, o_1; N(u_1)(o_2) \rangle$. The *projection functor* π_N from $\mathcal{G}(N)$ to I is defined by $\pi_N(\langle i, O \rangle) = i$ and $\pi_N(\langle u, o \rangle) = u$.

The *Grothendieck functor* $\mathcal{G}(F) : \mathcal{G}(M) \rightarrow \mathcal{G}(N)$ over a lax indexed functor $F : M \rightarrow N$ for I -indexed categories M and N is defined by $\mathcal{G}(F)(\langle i, O \rangle) = \langle i, F_i(O) \rangle$ for each $\langle i, O \rangle \in |\mathcal{G}(M)|$; and $\mathcal{G}(F)(\langle u, o \rangle) = \langle u, F_u(o) \rangle : \langle i_1, F_{i_1}(O_1) \rangle \rightarrow \langle i_2, F_{i_2}(O_2) \rangle$ with $F_u(o) = F_{i_1}(o); F_u(O_2) : F_{i_1}(O_1) \rightarrow N(u)(F_{i_2}(O_2))$ for each $\langle u, o \rangle : \langle i_1, O_1 \rangle \rightarrow \langle i_2, O_2 \rangle$ in $\mathcal{G}(M)$.

A lax indexed natural transformation $\eta : F \rightarrow G$ induces the *Grothendieck natural transformation* $\mathcal{G}(\eta) : \mathcal{G}(F) \rightarrow \mathcal{G}(G)$ defined by $\mathcal{G}(\eta)(\langle i, O \rangle) = \langle 1_i, \eta_i(O) \rangle$.

For a lax indexed functor $F : M_1 \rightarrow N$ and an indexed functor $G : M_2 \rightarrow N$ over I -indexed categories M_1, M_2 , and N define the indexed category $\mathcal{G}(F) \downarrow G : \mathcal{G}(M_1)^{\text{op}} \rightarrow \text{Cat}$ on objects by $(\mathcal{G}(F) \downarrow G)(\langle i, O_1 \rangle) = F_i(O_1) \downarrow G_i$, where we abbreviate an object $(F_i(O_1), p, O_2)$ of this comma category to (O_2, p) and a morphism $(1_{F_i(O_1)}, o_2)$ to o_2 , and on morphisms by $(\mathcal{G}(F) \downarrow G)(\langle u, o_1 \rangle : \langle i_1, O_{11} \rangle \rightarrow \langle i_2, O_{12} \rangle) : (\mathcal{G}(F) \downarrow G)(\langle i_2, O_{12} \rangle) \rightarrow (\mathcal{G}(F) \downarrow G)(\langle i_1, O_{11} \rangle)$ with $(\mathcal{G}(F) \downarrow G)(\langle u, o_1 \rangle)(O_{22}, p_2) = (M_2(u)(O_{22}), F_u(o_1); N(u)(p_2))$ and $(\mathcal{G}(F) \downarrow G)(\langle i, o_1 \rangle)(o_{22}) = M_2(u)(o_{22})$.

A lax indexed natural transformation $\eta : F_1 \rightarrow F_2$ then induces the indexed functor $\mathcal{G}(\eta) \downarrow G : \mathcal{G}(F_2) \downarrow G \rightarrow \mathcal{G}(F_1) \downarrow G$ defined by $(\mathcal{G}(\eta) \downarrow G)_{(i, O_1)}(O_2, p) = (O_2, \eta_i(O_1); p)$ and $(\mathcal{G}(\eta) \downarrow G)_{(i, O_1)}(o_2) = o_2$.

References

1. Borceux, F.: Handbook of Categorical Algebra. Vol. 2: Categories and Structures. Cambridge University Press (1994)
2. Climent Vidal, J., Soliveres Tur, J.: A 2-categorical Generalization of the Concept of Institution. *Studia Logica* **95**, 301–344 (2010)
3. Diaconescu, R.: Institution-independent Model Theory. *Studies in Universal Logic*, Birkhäuser (2008)
4. Goguen, J.A., Burstall, R.M.: A Study in the Foundation of Programming Methodology: Specifications, Institutions, Charters, and Parchments. In: Proc. Tut. Wsh. Category Theory and Computer Programming. *Lect. Notes Comp. Sci.*, vol. 240, pp. 313–333. Springer (1986)
5. Goguen, J.A., Burstall, R.M.: Institutions: Abstract Model Theory for Specification and Programming. *J. ACM* **39**, 95–146 (1992)
6. Knapp, A., Cengarle, M.V.: Institutions for OCL-Like Expression Languages. In: De Nicola, R., Hennicker, R. (eds.) *Software, Services, and Systems — Essays Dedicated to Martin Wirsing on the Occasion of His Retirement from the Chair of Programming and Software Engineering*, *Lect. Notes Comp. Sci.*, vol. 8950, pp. 193–214. Springer (2015)
7. Lawvere, F.W.: Functorial Semantics of Algebraic Theories. Ph.D. thesis, Columbia University (1963)
8. Mac Lane, S.: *Categories for the Working Mathematician*. Springer, 2nd edn. (1998)
9. Mayoh, B.H.: *Galleries and Institutions*. Tech. Rep. DAIMI PB – 191, Aarhus University (1985)
10. Object Management Group: Object Constraint Language. Standard formal/2014-02-03, OMG (2014), <http://www.omg.org/spec/OCL/2.4>
11. Paré, R., Schumacher, D.: Abstract Families and the Adjoint Functor Theorems. In: Johnstone, P.T., Paré, R. (eds.) *Indexed Categories and Their Applications*, *Lect. Notes Math.*, vol. 661, pp. 1–125. Springer (1978)
12. Pawłowski, W.: Context Institutions. In: *Sel. Papers 11th Ws. Specification of Abstract Data Types & 8th COMPASS Ws. Recent Trends in Data Type Specifications*, *Lect. Notes Comp. Sci.*, vol. 1130, pp. 436–457. Springer (1996)
13. Poigné, A.: Foundations are Rich Institutions, but Institutions are Poor Foundations. In: Ehrig, H., Herrlich, H., Kreowski, H.J., Preuß, G. (eds.) *Proc. Wsh. Categorical Methods in Computer Science 1988*. *Lect. Notes Comp. Sci.*, vol. 393, pp. 82–101. Springer (1989)
14. Sannella, D., Tarlecki, A.: *Foundations of Algebraic Specification and Formal Software Development*. EATCS Monographs in Theoretical Computer Science, Springer (2012)
15. Tarlecki, A., Burstall, R.M., Goguen, J.A.: Some Fundamental Algebraic Tools for the Semantics of Computation, Part 3: Indexed Categories. *Theo. Comp. Sci.* **91**, 239–264 (1991)
16. Țuțu, I., Fiadeiro, J.: From Conventional to Institution-Independent Logic Programming. *J. Logic Comp.* **27**(6) (2015)

17. Wolter, U., Martini, A.: Shedding New Light in the World of Logical Systems. In: Proc. 7th Intl. Conf. Category Theory and Computer Science (CTCS'97). Lect. Notes Comp. Sci., vol. 1290, pp. 159–176. Springer (1997)