Didier Bert and Christine Choppy and Peter Mosses (eds.) Recent Trends in Algebraic Development Techniques. LNCS 1827, pp. 182–200, Springer Verlag, 2000.

Guarded Algebras: Disguising Partiality so You Won't Know Whether its There*

Magne Haveraaen[1] and Eric G. Wagner[2]

[1] Institutt for Informatikk, Universitetet i Bergen, HiB, N-5020 Bergen, Norway
http://www.ii.uib.no/~magne
[2] Wagner Mathematics, 1058 Old Albany Post Road, Garrison, NY 10524, USA
http://www.ii.uib.no/~wagner

Abstract. Motivated by considerations from program semantics, we suggest the notion of guarded algebras. These make explicit the significant arguments to functions, and prevent involuntary capture of error values and undefined cases in specifications. Here we show that guarded reasoning disguises whether the underlying models are partial or total.

1 Introduction

When using algebraic technology to specify software we run into the problem of partiality in the operators. Partiality can be inherent in an operation itself, or be an artifact of certain models, such as the finite size of our computational models. An example of the former is division by zero. Examples of the latter are limits on the applicability of arithmetic operators due to size limitations on computer representation of numbers.

These issues are addressed by *guarded algebras* [HW95]. The carrier sets of the sorts are partitioned, using sort-guards, into "significant" and "insignificant" elements, and the arguments of the operations are partitioned, using operator-guards, into "significant" and "insignificant" arguments. Guards are ordinary operations which are being used to describe the limitations of sorts and other operations. The notion of significance is built into the semantics of the specifications by an appropriate choice for the satisfaction relation between models and specifications. These modifications have two desirable effects:

1. Guarded algebras may be used to model detectable error conditions by having error values as insignificant elements in the carriers.
2. Undetectable error situations may be represented by partiality in the guarded models.

If we distinguish models solely by their significant elements, we find that there is an isomorphism between the category of care-distinct guarded total models (all

* This work has been partially supported by the research council of Norway, by the EU through the COMPASS and CoFI networks, by the Department of Informatics at the University of Bergen, and the Department of Computer Science at the University of Wales Swansea.

errors are detectable) and the category of care-distinct guarded partial models (undetectable errors are allowed). Thus we may fully disguise whether total or partial models are being used. In this paper we restrict our attention to conditional equational logic.

Several approaches have been developed to handle error situations using total algebras. Among these are error algebras [GTW78], which identify error elements for every sort, or OK-algebras [GDLE84,HL89], which use predicates to designate the safe arguments of ordinary operators. Surveys with further references for many of these and various other approaches, also involving partiality, can be found in [Mos93] and [Mos95]. We restrict our more detailed comparison to three approaches: Reichel's equoids [Rei87], Kreowski's based algebras [Kre87,KM95], and Meseguer's membership algebras [Mes98]. These approaches have in common the feature that within a specification special sets are defined (akin to the guards in guarded specifications) which are, or can be, used to define the "significant" elements and/or arguments, or can be used to define the domains of definition of the operations in a partial algebra interpretation.

Reichel's equoids are similar to guarded algebras in that for each operator σ a set of equations, *def* σ, is used to specify the domains of the operators – the domain of $A(\sigma)$ in an algebra A being required to be exactly the set of solutions for *def* σ. This differs from the guarded framework in which the satisfaction of operator-guards is only a sufficient condition for definedness (significance). Reichel's framework does not include a counterpart to the sort-guards of the guarded framework. Equoids are specified using existential satisfaction of conditional equations with an emphasis on partiality, but a similar approach could be taken within the framework of total conditional equational logic. Another difference is that equoids have hierarchically structured signatures rather than the simple two-level partition into UF and PF employed in this paper. The use of hierarchical signatures increases expressiveness but having *def* σ tightly determine the domain of definition appears to restrict the model class.

Kreowski's based algebras provide a means for defining certain classes of partial algebras within the framework of total conditional equational logic. A based specification is a pair $BASP = (BASE, SPEC)$ of plain specifications, where $BASE$ is a subspecification of $SPEC$. A $BASP$-algebra is a triple (B, C, h) where B is a $BASE$-algebra, C is a $SPEC$-algebra and $h : B \to C|_{BASE}$ is a homomorphism from B to the $BASE$-reduct of C (a slightly more general definition is given in [KM95]). The partiality is achieved by means of the `PART` constructions which restricts and corestricts the operations of C to the image of B under h. While the emphasis in [Kre87] is on partiality, the same idea could be used to define a notion akin to that of significant arguments in guarded algebras. The results in [KM95] show that the based algebra framework is equivalent in expressiveness to total conditional equational logic. A similar result can be shown for the guarded algebra framework. However this proof (and definition) of equivalent expressiveness does not take into account the difference in how the transition is made from total to partial algebras. While based algebras can be used to produce many pleasing specifications (including one for the standard

example: stacks), the technique does not seem applicable to examples such as bounded stacks. The reason is that the sets of "significant elements" in a based algebra are always a subset of the carrier generated by a subset of the operators starting from a subset of the carriers. This is not as general as what can be done in the other approarches.

Messquer's membership algebras [Mes98], while apparently developed primarily as general framework for dealing with subsorting issues, can also be used to define partiality within a total algebra framework. Roughly speaking, specifications of membership algebras employ membership predicates to specify the sortedness of terms and the relationships between sorts. By decreeing certain sorts to be "significant" one can simulate guarded algebras, but at the cost of losing the brevity of guarded satisfaction since the guards must be explicitly stated in the conditional equations. Membership algebras are shown, in [Mes98], to have the same expressive power as Horn Clause Logic (with equality and predicates) which is a more expressively powerful framework than that used for guarded algebras. Conversely, it is possible to approximate the membership predicates within the framework of guarded algebras. We are still investigating the effects of this difference in power on actual specifications.

This paper is organized as follows: In section two we recapitulate some definitions and results about institutions and general logics. Then we define total, partial and guarded algebras. Section four defines the corresponding institutions, relates these, and contains our main result on guardedness and partiality. Finally we discuss further theoretical development of the notion of guarded algebras.

2 Institutions and Logics

The general concept of a logic [Mes89] has three facets: model theory, entailment, and proof calculus, all sharing the same notions of signature and sentence. The model-theoretic aspect is captured by the concept of an institution: signatures, sentences, models and satisfaction. An entailment system captures the syntactic notions of a logic: signatures, sentences and syntactic derivability (entailment). A proof calculus is on top of an entailment system and provides the structural aspects of how to actually prove a derivation of the entailment system.

Let **Set** be the category of sets and total functions.

Definition 1 (Institution). *An* institution *is given by a quadruple* $\mathcal{INST} = (\mathbf{Sign}, \mathit{Sen}, \mathit{Mod}, \models)$*, where*

- **Sign** *is a category of* signatures*,*
- $\mathit{Sen} : \mathbf{Sign} \to \mathbf{Set}$ *is a functor giving the* sentences
- $\mathit{Mod} : \mathbf{Sign}^{\mathsf{op}} \to \mathbf{CAT}$ *is a functor giving the category of* models
- $\models$ *is a family, indexed by* $\mathsf{Obj}(\mathbf{Sign})$*, of* satisfaction relations*,* $\models_\Theta \subseteq \mathit{Mod}(\Theta) \times \mathit{Sen}(\Theta)$ *for each* $\Theta \in \mathsf{Obj}(\mathbf{Sign})$

such that, for each morphism $\theta \in \mathbf{Sign}(\Theta, \Theta')$*, the* satisfaction condition*,*

$$M' \models_{\Theta'} \mathit{Sen}(\theta)(\varphi) \Leftrightarrow \mathit{Mod}(\theta)(M') \models_\Theta \varphi,$$

holds for each $M' \in \mathsf{Obj}(Mod(\Theta'))$ *and each* $\varphi \in Sen(\Theta)$.

An *entailment* system defines an entailment relation between sets of sentences (axioms) and sentences (facts) for any given signature. Using entailment one may derive, on a syntactic level, new facts as consequences of axioms. An entailment system coupled with an institution forms a *logic*. A logic coupled with a *proof calculus*, taking into account the structural aspects in how facts are derived from axioms, forms a *logical system*[1]. If the entailment system of a logic only derives facts satisfied by the models of the axioms, it is *sound*. The entailment system is *complete* if, for the models satisfying a set of axioms, all sentences satisfied by the models are derivable from the axioms.

There are forgetful functors from logical systems and logics to institutions. These functors are adjoint to several functors going from institutions to logics and logical systems. Given an institution, entailment systems and proof calculi may be generated using these functors. In most cases such systems will be of little use as they will not be efficient to work with. Instead one wants an independently developed proof calculus with an entailment system which is efficient, sound and, if possible, complete.

Institutions can be related by many different kinds of morphisms. Certain of these morphisms transport entailment systems and proof calculi between institutions. This allows reuse of useful entailment systems and proof calculi, and also the reuse of tools developed for one logical system for the entailment system and proof calculus of a different institution.

The following definitions and theorems are adapted from [Mes89]. See also [Cer93] which investigates their use in relating various partial algebra approaches.

Definition 2 (Theory). *Given an institution* $\mathcal{INST} = (\mathbf{Sign}, Sen, Mod, \models)$.
The axiom-preserving theory *of* $\mathcal{INST}$ *is the category* $\mathbf{Th_0}(\mathcal{INST})$ *with*

- presentations *as objects* $(\Theta, \Phi) \in \mathsf{Obj}(\mathbf{Th_0}(\mathcal{INST}))$ *where* $\Theta \in \mathsf{Obj}(\mathbf{Sign})$ *and* $\Phi \subseteq Sen(\Theta)$, *and*
- *morphisms* $\theta : (\Theta, \Phi) \rightarrow (\Theta', \Phi') \in \mathsf{Mor}(\mathbf{Th_0}(\mathcal{INST}))$ *where* $\theta \in \mathbf{Sign}(\Theta, \Theta')$ *such that* $Sen(\theta)(\Phi) \subseteq \Phi'$.

Definition 3. *Given an institution* $\mathcal{INST} = (\mathbf{Sign}, Sen, Mod, \models)$.
The varieties *of* $\mathcal{INST}$ *are the categories given by the functor* $Vmod_{\mathcal{INST}} : \mathbf{Th}_0(\mathcal{INST})^{\mathsf{op}} \rightarrow \mathbf{CAT}$, *where* $Vmod_{\mathcal{INST}}(\Theta, \Phi)$, *for* $\Phi \subseteq Sen(\Theta)$, *is the full subcategory of* $Mod(\Theta)$ *with objects* $\mathsf{Obj}(Vmod_{\mathcal{INST}}(\Theta, \Phi)) = \{M \in \mathsf{Obj}(Mod(\Theta)) \mid M \models_\Theta \Phi\}$.

Definition 4 (Simple map of institutions). *Let* $sign : \mathbf{Th}_0(\mathcal{INST}) \rightarrow \mathbf{Sign}$ *and* $sign' : \mathbf{Th}_0(\mathcal{INST}') \rightarrow \mathbf{Sign}'$ *be projection functors on theories for institutions* $\mathcal{INST} = (\mathbf{Sign}, Sen, Mod, \models)$ *and* $\mathcal{INST}' = (\mathbf{Sign}', Sen', Mod', \models')$.
A simple map of institutions *is given by* $(\zeta, \alpha, \beta) : \mathcal{INST} \rightarrow \mathcal{INST}'$, *where*

[1] Readers are referred to [Mes89] for a proper treatment of these concepts. Lack of space prohibits a more in depth presentation here, and further technical details are not needed in this paper.

– $\zeta : \mathbf{Th}_0(\mathcal{INST}) \to \mathbf{Th}_0(\mathcal{INST}')$ *is a functor with a related functor on signatures* $\zeta_1 : \mathbf{Sign} \to \mathbf{Sign}'$ *such that*
  - $sign' \circ \zeta = \zeta_1 \circ sign$
  - $\zeta(\Theta, \Phi) = \zeta(\Theta, \emptyset) \cup (\zeta_1(\Theta), \alpha_\Theta(\Phi))$ *for all* $(\Theta, \Phi) \in \mathsf{Obj}(\mathbf{Th_0}(\mathcal{INST}))$
– $\alpha : Sen \to Sen' \circ \zeta_1$ *is a natural transformation in* **Set** *indexed by the signatures in* **Sign**,
– $\beta : Vmod_{\mathcal{INST}'} \circ \zeta^{op} \to Vmod_{\mathcal{INST}}$ *is a natural transformation in* **CAT** *indexed by the presentations in* $\mathbf{Th}_0(\mathcal{INST})$,

such that for each $\Theta \in \mathsf{Obj}(\mathbf{Sign})$, $\varphi \in Sen(\Theta)$ *and* $M' \in Vmod_{\mathcal{INST}'}(\zeta(\Theta, \emptyset))$,

$$M' \models'_{\zeta_1(\Theta)} \alpha_\Theta(\varphi) \Leftrightarrow \beta_{(\Theta,\emptyset)}(M') \models_\Theta \varphi.$$

If the functors $\beta_{(\Theta,\emptyset)}$ are surjective for every $(\Theta, \emptyset) \in \mathsf{Obj}(\mathbf{Th_0}(\mathcal{INST}))$, then the simple map of institutions is said to be surjective.

Theorem 1. *Given institutions* $\mathcal{INST} = (\mathbf{Sign}, Sen, Mod, \models)$ *and* $\mathcal{INST}' = (\mathbf{Sign}', Sen', Mod', \models')$ *where* $\mathcal{INST}'$ *is part of the logical system* $\mathcal{L}'$ *with an entailment system and corresponding proof calculus.*

If there is a simple map of institutions $(\zeta, \alpha, \beta) : \mathcal{INST} \to \mathcal{INST}'$ *then we get a logical system* $\mathcal{L}$*, containing* $\mathcal{INST}$*, with entailment system and corresponding proof calculus from* $\mathcal{L}'$*.*

If $\mathcal{L}'$ *has a complete entailment system, then* $\mathcal{L}$ *will have a complete entailment system. Moreover, if* $\mathcal{L}'$ *has a sound entailment system and the simple map of institutions is surjective, then* $\mathcal{L}$ *will have a sound entailment system.*

3 Algebraic Concepts

We write $[n]$ for the set $\{1, \ldots, n\}$. We view a string of length n of elements of a set S as a mapping $u : [n] \to S$. We write ϵ for the empty string and S^* for the set of all strings over S. We write $|w|$ for the length of a string w. A partial function $f : A \xrightarrow{p} B$ consists of a set $\partial(f)$ together with total function $\underline{f} : \partial(f) \xrightarrow{t} B$.

3.1 Signatures

Definition 5 (Plain signatures). *A* plain signature Σ *is given by the data of a 4-tuple,* $\Sigma = (S, F, \mathrm{dom}, \mathrm{cod})$*, for* S *a set of sorts,* F *a set of operators, a function* $\mathrm{dom} : F \xrightarrow{t} S^*$ *giving the domain, and a function* $\mathrm{cod} : F \xrightarrow{t} S$ *giving the codomain of every operator. A* plain signature morphism $\mu : \Sigma \to \Sigma'$*, for plain signatures* $\Sigma = (S, F, \mathrm{dom}, \mathrm{cod})$ *and* $\Sigma' = (S', F', \mathrm{dom}', \mathrm{cod}')$*, is a pair of functions* $\mu_1 : S \xrightarrow{t} S'$ *and* $\mu_2 : F \xrightarrow{t} F'$ *such that* $\mathrm{dom}' \circ \mu_2 = \mu_1 \circ \mathrm{dom}$*, and* $\mathrm{cod}' \circ \mu_2 = \mu_1 \circ \mathrm{cod}$.

Plain signatures with signature morphisms form a category **Sig**. For an operator $\sigma \in F$ the pair, $\langle \mathrm{dom}(\sigma), \mathrm{cod}(\sigma) \rangle$ is called the *profile* of σ.

Example 1. The plain signature for the mathematical concept of a Group can be given using the following, slightly sugared, syntax.

$$
\begin{array}{ll}
\textbf{sig}\ \Sigma_{\mathsf{Group}} = & \\
\quad \textbf{sorts} & \mathsf{group} \\
\quad \textbf{opns} & \odot : \rightarrow \mathsf{group} \\
 & \oplus : \mathsf{group} \times \mathsf{group} \rightarrow \mathsf{group} \\
 & \ominus : \mathsf{group} \rightarrow \mathsf{group}
\end{array}
$$

Definition 6 (Guarded signatures). *A* guarded signature Γ *is given by the data of a 7-tuple,* $\Gamma = (S, UF, PF, \mathrm{dom}, \mathrm{cod}, \delta, \gamma)$*, for* S *a set of sorts;* UF *a set of unprotected operators;* PF *a set of protected operators, such that* $(UF \cap PF) = \emptyset$*; a function* $\mathrm{dom} : (UF \cup PF) \xrightarrow{t} S^*$ *giving the domain and a function* $\mathrm{cod} : (UF \cup PF) \xrightarrow{t} S$ *giving the codomain of every operator;* $\delta : S \xrightarrow{t} (S \times UF \times UF)$ *the sort-guard and* $\gamma : PF \xrightarrow{t} (S \times UF \times UF)$ *the operator-guard, with the requirement that for every* $s \in S$ *and for every* $\psi \in PF$

$$
\begin{gathered}
\delta(s)_1 = \mathrm{cod}(\delta(s)_2) = \mathrm{cod}(\delta(s)_3), \quad \mathrm{dom}(\delta(s)_2) = \epsilon, \quad \mathrm{dom}(\delta(s)_3) = s, \\
\gamma(\psi)_1 = \mathrm{cod}(\gamma(\psi)_2) = \mathrm{cod}(\gamma(\psi)_3), \quad \mathrm{dom}(\gamma(\psi)_2) = \epsilon, \quad \mathrm{dom}(\gamma(\psi)_3) = \mathrm{dom}(\psi).
\end{gathered}
$$

A closed guarded signature morphism $\mu : \Gamma \rightarrow \Gamma'$*, for guarded signatures* $\Gamma = (S, UF, PF, \mathrm{dom}, \mathrm{cod}, \delta, \gamma)$ *and* $\Gamma' = (S', UF', PF', \mathrm{dom}', \mathrm{cod}', \delta', \gamma')$*, is a triple of functions* $\mu_1 : S \xrightarrow{t} S'$*,* $\mu_2 : UF \xrightarrow{t} UF'$ *and* $\mu_3 : PF \xrightarrow{t} PF'$ *such that the profiles and guard structure are preserved, that is:*

$$
\begin{gathered}
\mathrm{dom}' \circ \mu_2 = \mu_1 \circ \mathrm{dom}, \quad \mathrm{cod}' \circ \mu_2 = \mu_1 \circ \mathrm{cod}, \\
\mathrm{dom}' \circ \mu_3 = \mu_1 \circ \mathrm{dom}, \quad \mathrm{cod}' \circ \mu_3 = \mu_1 \circ \mathrm{cod}, \\
\delta' \circ \mu_1 = (\mu_1 \times \mu_2 \times \mu_2) \circ \delta, \quad \gamma' \circ \mu_3 = (\mu_1 \times \mu_2 \times \mu_2) \circ \gamma.
\end{gathered}
$$

Guarded signatures with closed signature morphisms form a category **CGSig**.

Let PL : **CGSig**→**Sig** be the functor taking each guarded signature to the corresponding plain signature, i.e., $PL(S, UF, PF, \mathrm{dom}, \mathrm{cod}, \delta, \gamma) = (S, (UF \cup PF), \mathrm{dom}, \mathrm{cod})$ and $PL(\mu_1, \mu_2, \mu_3) = (\mu_1, \mu_2 \cup \mu_3)$.

Example 2. The guarded signature for the mathematical concept of a Group is the same as the plain signature for Group (see Example 1), but extended with the declaration of sorts and functions for the sort- and operator-guards.

$$
\begin{array}{ll}
\textbf{sig}\ \Gamma_{\mathsf{Group}} = & \\
\quad \textbf{sorts} & \mathsf{group} \\
\quad \textbf{opns} & \odot : \rightarrow \mathsf{group} \\
 & \oplus : \mathsf{group} \times \mathsf{group} \rightarrow \mathsf{group} \\
 & \ominus : \mathsf{group} \rightarrow \mathsf{group} \\
\quad \textbf{delta}(\mathsf{group}) & = (B, t, \mathsf{dgroup}) \\
\quad \textbf{delta}(B) & = (B, t, \mathsf{dB})
\end{array}
$$

Here the line "**delta**(group) = (B, t, dgroup)" signals the definition of sort-guard for sort group, and implicitly declares B as a sort name and operations $t : \rightarrow B$ and $\mathsf{dgroup} : \mathsf{group} \rightarrow B$ in accordance with the requirements on δ. The line "**delta**$(B) = (B, t, \mathsf{dB})$" declares sort-guard for the sort B, naming B as the sort, reusing the operation $t : \rightarrow B$, and introduces the operation $\mathsf{dB} : B \rightarrow B$. In [HW95] it was shown that the sort-guards may be added in a canonical way using an additional sort B, a fixed constant t and a fresh operation name $\mathsf{d}s$ for every sort s. Thus we will normally not include explicit sort-guards as part of the sugared signature declarations.

Example 3. The situation becomes more interesting if we create a guarded group with an operator-guard on the inverse-operation (omitting the sort-guards):

sig $\Gamma_{\mathsf{GGroup}} =$

sorts	group
opns	$\odot : \rightarrow \mathsf{group}$
	$\oplus : \mathsf{group} \times \mathsf{group} \rightarrow \mathsf{group}$
	$\ominus : \mathsf{group} \rightarrow \mathsf{group}$
guard$(\ominus)$	$= (\mathsf{Bool}, \mathsf{true}, \mathsf{invertible})$

The line "**guard**$(\ominus) = (\mathsf{Bool}, \mathsf{true}, \mathsf{invertible})$" defines the operator-guard for $\ominus$, and implicitly declares Bool as a sort name and operations $\mathsf{true} : \rightarrow \mathsf{Bool}$ and $\mathsf{invertible} : \mathsf{group} \rightarrow \mathsf{Bool}$ in accordance with the requirements on γ. An operation in the target of a **guard** cannot be used as the argument for a guard in order to provide the necessary partitioning of operators for guarded signatures. The declaration Γ_{GGroup} defines sorts group and Bool, and an anonymous sort B as target for δ. Unprotected operators are $\odot$, $\oplus$, true, invertible, the anonymous constant t into B and anonymous operations dgroup, dBool and dB from each sort into B, as required for δ. In this example there is only one protected operator, namely $\ominus$.

3.2 Algebras

We use the standard notions of plain total and partial algebras. For every plain signature Σ the *category of plain total algebras*, $\mathbf{TAlg}(\Sigma)$, has plain total algebras for Σ as objects and plain homomorphisms between them as morphisms. The *categories of plain partial algebras*, $\mathbf{PAlg}(\Sigma)$, has plain partial algebras for Σ as objects and weak homomorphisms between them as morphisms.

Definition 7 (Significant elements). *Let $\Sigma = PL(\Gamma)$ for a guarded signature $\Gamma = (S, UF, PF, \mathrm{dom}, \mathrm{cod}, \delta, \gamma)$.*

The set of significant elements *for a sort $s \in S$ of a plain total algebra A for Σ is given by*

$$D(A, s) = \{a \in A(s) \mid A(\delta(s)_2) = A(\delta(s)_3)(a)\}$$

The set of significant elements *for a sort $s \in S$ of a plain partial algebra A for Σ is given by*

$$D(A,s) = \begin{cases} \{a \in \partial(A(\delta(s)_3)) \mid A(\delta(s)_2) = A(\delta(s)_3)(a)\} & \text{if } \partial(A(\delta(s)_2)) \neq \emptyset \\ \emptyset & \text{if } \partial(A(\delta(s)_2)) = \emptyset \end{cases}$$

For $w \in S^*$ define $D(A,w) = D(A,w_1) \times \cdots \times D(A,w_{|w|})$.

Definition 8 (Significant arguments). *Let $\Gamma = (S, UF, PF, \mathrm{dom}, \mathrm{cod}, \delta, \gamma)$ be a guarded signature and let $\Sigma = PL(\Gamma)$.*

The set of significant arguments *for an operator $\sigma \in (UF \cup PF)$ in the plain total algebra A for Σ is given by*

$$G(A,\sigma) = \begin{cases} D(A, \mathrm{dom}(\sigma)) & \text{if } \sigma \in UF \\ \{a \in D(A,\mathrm{dom}(\sigma)) \mid A(\gamma(\sigma)_2) = A(\gamma(\sigma)_3)(a)\} & \text{if } \sigma \in PF, \end{cases}$$

The significant arguments *for an operator $\sigma \in UF \cup PF$ in the plain partial algebra A for Σ are given by*

$$G(A,\sigma) = \begin{cases} D(A,\mathrm{dom}(\sigma)) & \text{if } \sigma \in UF \\ \{a \in D(A,\mathrm{dom}(\sigma)) \mid & \\ \quad A(\gamma(\sigma)_2) = A(\gamma(\sigma)_3)(a)\} & \text{if } \sigma \in PF, \\ & \quad \partial(A(\gamma(\sigma)_2)) \neq \emptyset \text{ and} \\ & \quad D(A,\mathrm{dom}(\sigma)) \subseteq \partial(A(\gamma(\sigma)_3)) \\ \emptyset & \text{otherwise} \end{cases}$$

Definition 9 (Guarded algebras). *Let $\Sigma = PL(\Gamma)$ for guarded signature $\Gamma = (S, UF, PF, \mathrm{dom}, \mathrm{cod}, \delta, \gamma)$.*

A guarded total algebra *A for Γ is a plain total algebra A for Σ, such that for every operator $\sigma \in (UF \cup PF)$, we have $A(\sigma)\,(G(A,\sigma)) \subseteq D(A,\mathrm{cod}(\sigma))$.*

A guarded partial algebra *A for Γ is a plain partial algebra A for Σ, such that for every operator $\sigma \in (UF \cup PF)$, we have $G(A,\sigma) \subseteq \partial(A(\sigma))$ and $\underline{A(\sigma)}\,(G(A,\sigma)) \subseteq D(A,\mathrm{cod}(\sigma))$.*

For a guarded partial algebra A we have that $\partial(A(\delta(s)_2)) \neq \emptyset$ for every $s \in S$ and $\partial(A(\gamma(\sigma)_2)) \neq \emptyset$ for every $\sigma \in PF$ since $D(A,\epsilon) \neq \emptyset$. Also, $D(A,\mathrm{dom}(\sigma)) \subseteq \partial(A(\gamma(\sigma)_3))$ for every $\sigma \in PF$ since $D(A,\mathrm{dom}(\sigma)) = D(A,\mathrm{dom}(\gamma(\sigma)_3)) = G(A,\gamma(\sigma)_3)$ and $G(A,\gamma(\sigma)_3) \subseteq \partial(A(\gamma(\sigma)_3))$.

The *category of guarded total algebras* $\mathbf{GTAlg}(\Gamma)$ is the full subcategory of $\mathbf{TAlg}(\Sigma)$ with guarded total algebras as objects. The *category of guarded partial algebras* $\mathbf{GPAlg}(\Gamma)$ is the full subcategory of $\mathbf{PAlg}(\Sigma)$ with guarded partial algebras as objects.

Defining reducts (on algebras and homomorphisms) in the usual way, we may extend the notion of a category of algebras for every signature to a functor from the dual of the signature category into **CAT**. Then we get functors $\mathbf{TAlg} : \mathbf{Sig}^{op} \to \mathbf{CAT}$ and $\mathbf{PAlg} : \mathbf{Sig}^{op} \to \mathbf{CAT}$ for plain algebras and $\mathbf{GTAlg} : \mathbf{CGSig}^{op} \to \mathbf{CAT}$ and $\mathbf{GPAlg} : \mathbf{CGSig}^{op} \to \mathbf{CAT}$ for guarded algebras.

The intuition behind guarded algebras is that if A is a guarded algebra with plain signature Σ, then, for each $s \in S$, we only "really care" about those elements of the carrier $a \in A(s)$ which are in $D(A,s)$, and, for each $\sigma \in (UF \cup PF)$, we only care about the values of $A(\sigma)$ for the arguments $a \in G(A,\sigma) \subseteq A(\mathrm{dom}(\sigma))$. This suggests that if we have two guarded algebras, A and B, in which the elements and arguments about which we "really care" are the same and whose operations behave the same on those arguments, then these algebras are equivalent.

Definition 10. *Given guarded, total or partial, algebras A and B with the same guarded signature $\Gamma = (S, UF, PF, \mathrm{dom}, \mathrm{cod}, \delta, \gamma)$ we say they are* care-equivalent*, and write $A \equiv_\Gamma B$, if*

1. *$D(A,s) = D(B,s)$ for every $s \in S$.*
2. *$G(A,\sigma) = G(B,\sigma)$ for every $\sigma \in (UF \cup PF)$.*
3. *For every $\sigma \in (UF \cup PF)$ we have $A(\sigma)(a)|_{G(A,\sigma)} = B(\sigma)(a)|_{G(A,\sigma)}$.*

Algebras which are not care-equivalent are care-distinct.

3.3 Terms, Sentences and Satisfaction

Definition 11. *A* finite collection of variables *$\chi = (X, v, \overline{x})$ for a plain signature $\Sigma = (S, F, \mathrm{dom}, \mathrm{cod})$, consists of a finite set X of variable names, where $X \cap F = \emptyset$, a function $v : X \to S$ typing the variables and a bijective function $\overline{x} : [|X|] \to X$ defining a string of all the variables in X.*

The function v gives the sort of each variable, the function $\overline{x}$ orders the variables, permitting us, among other things, to speak of the i'th variable. The use of a finite collection of variables is insufficient for certain constructions, but is sufficient when considering ordinary terms and axioms composed from finite sets of terms.

The S-indexed family $T(\Sigma, \chi)$ of terms with variables χ for Σ is defined in the normal way. We extend the functions dom and cod to derived operators $\chi.t \in T(\Sigma,\chi)_s$ for $s \in S$ by taking $\mathrm{dom}(\chi.t) = v \circ \overline{x}$, and $\mathrm{cod}(\chi.t) = s$. The result is a derived function $A(\chi.t) : A(v \circ \overline{x}) \to A(s)$, for each total or partial algebra A for Σ, defined recursively in the normal way on the structure of t.

We define the semantics of a guarded total (respectively partial) function with variables $\chi = (X, v, \overline{x})$ over Γ with $(S, F, \mathrm{dom}, \mathrm{cod}) = PL(\Gamma)$ as for a total (partial) derived operation. In addition we define the significant arguments, $G(A, \chi.t) \subseteq D(A, \mathrm{dom}(\chi.t))$ by

$$G(A,\chi.t) = \begin{cases} D(A,\mathrm{dom}(\chi.t)) & \text{If } t = \overline{x}_i \\ \Big\{ a \in D(A,\mathrm{dom}(\chi.t)) \mid a \in G(A,\chi.t_i)), i \in [n], \\ \quad \langle A(\chi.t_1)(a), \ldots, A(\chi.t_n)(a)\rangle \in G(A,\sigma) \Big\} & \text{If } t = \sigma(t_1,\ldots,t_n) \end{cases}$$

Proposition 1. *Given a guarded partial algebra A and variables $\chi = (X, v, \overline{x})$ for guarded signature Γ with $\Sigma = PL(\Gamma)$, then $G(A, \chi.t) \subseteq \partial(A(\chi.t))$.*

Proof. Easily shown by structural induction on the form of t by observing it holds for t a variable and is preserved for every operation in the signature. □

The sentences, given by the functor Sen : **CGSig**→**Set**, define the kind of axioms we have. In this paper we focus on conditional equational logics, so our axiom schemata will only allow for this.

Definition 12 (Conditional equation). *Given a plain signature Σ.*

A conditional equation $\chi.(\{(t_1, u_1), \ldots, (t_k, u_k)\}, (t_{k+1}, u_{k+1}))$*, for $k \geq 0$, consists of terms $\chi.t_i, \chi.u_i \in T(\Sigma, \chi)_{s_i}$, $i \in [k+1]$, $s_i \in S$, for a finite collection of variables $\chi = (X, v, \overline{x})$ over Σ.*

The set of all conditional equations for Σ is denoted $CE(\Sigma)$. Different conditional equations in $CE(\Sigma)$ may have different collections of variables.

We may now extend CE to a functor CE : **Sig**→**Set** by defining $CE(\mu : \Sigma \to \Sigma')$ as the normal substitution of operations (and variables) generated by μ. This also gives us a functor $CE \circ PL$: **CGSig**→**Set**, which we sometimes will denote as just CE.

We can define a translation from guarded conditional equations to plain conditional equations that follows the same pattern as that of defining the significant arguments for a derived operator.

Definition 13. *Given a guarded signature $\Gamma = (S, UF, PF, \mathrm{dom}, \mathrm{cod}, \delta, \gamma)$, and a finite collection of variables $\chi = (X, v, \overline{x})$.*

The domain-set $\mathbf{d}(\Gamma, \chi, t)$*, for a term $\chi.t \in T(PL(\Gamma), \chi)_s$ for some $s \in S$, is defined by*

$$\mathbf{d}(\Gamma, \chi, t) = \begin{cases} \{(\delta(v(x))_2, \delta(v(x))_3(x)) | x \in X\} & \text{If } t \in X \\ \{(\delta(v(x))_2, \delta(v(x))_3(x)) | x \in X\} & \\ \quad \bigcup \left(\cup_{i \in [n]} (\mathbf{d}(\Gamma, \chi, t_i)) \right) & \text{If } \upsilon \in UF,\ t = \upsilon(t_1, \ldots, t_n) \\ \{(\delta(v(x))_2, \delta(v(x))_3(x)) | x \in X\} & \\ \quad \bigcup \left(\cup_{i \in [n]} (\mathbf{d}(\Gamma, \chi, t_i)) \right) & \\ \quad \bigcup \{(\gamma(\psi)_2, \gamma(\psi)_3(t_1, \ldots, t_n))\} & \text{If } \psi \in PF,\ t = \psi(t_1, \ldots, t_n) \end{cases}$$

The *generating principle* for an operation $\sigma \in UF \cup PF$ from a guarded signature $\Gamma = (S, UF, PF, \mathrm{dom}, \mathrm{cod}, \delta, \gamma)$ is the $CE(\Gamma)$ equations

$$\mathbf{d}(\Gamma, \sigma) = \chi. \left(\mathbf{d}(\Gamma, \chi, \sigma(\overline{x})), (\delta(\mathrm{cod}(\sigma))_2, \delta(\mathrm{cod}(\sigma))_3(\sigma(\overline{x}))) \right)$$

for some $\chi = (X, v, \overline{x})$ such that $v \circ \overline{x} = \mathrm{dom}(\sigma)$.

The *fully guarded-equation* $\mathbf{d}(\Gamma, \varphi)$ for a sentence $\varphi \in CE(PL(\Gamma))$, where $\varphi = \chi.(\{(t_i, u_i) | i \in [k]\}, (t_{k+1}, u_{k+1}))$ for $k \geq 0$, is defined by

$$\mathbf{d}(\Gamma, \varphi) = \chi. \left(\cup_{i \in [k+1]} (\mathbf{d}(\Gamma, \chi, t_i) \cup \mathbf{d}(\Gamma, \chi, u_i)) \bigcup \{(t_i, u_i) | i \in [k]\}, (t_{k+1}, u_{k+1}) \right)$$

Definition 14 (Plainification). *For $\Gamma = (S, UF, PF, \mathrm{dom}, \mathrm{cod}, \delta, \gamma)$, a guarded signature, and $\Phi \subseteq CE(\Gamma)$, a set of sentences, define the* plainification *of presentation (Γ, Φ) by*

$$PL(\Gamma, \Phi) = (PL(\Gamma), \{\mathbf{d}(\Gamma, \sigma) \mid \sigma \in UF \cup PF\} \cup \{\mathbf{d}(\Gamma, \varphi) \mid \varphi \in \Phi\}) .$$

The plainified specification consists of two parts: one generated from the signature, and the other generated from each of the equations of the presentation. Note the overloading of PL as a functor $PL : \mathbf{CGSig} \to \mathbf{Sig}$ and a translation on presentations (pairs consisting of a signature and a set of axioms).

A satisfaction relation relates models and sentences by defining when a sentence holds in an algebra. We will define satisfaction relations for the three cases of total, partial and guarded algebras.

Definition 15 (Plain satisfaction). *Let $\Sigma = (S, F, \mathrm{dom}, \mathrm{cod})$ be a plain signature, $\varphi = \chi.(\{(t_1, u_1), \ldots, (t_k, u_k)\}, (t_{k+1}, u_{k+1})) \in CE(\Sigma)$ a conditional equation.*

The total satisfaction relation *for Σ, $\overset{t}{\models}_\Sigma \subseteq \mathsf{Obj}(\mathbf{TAlg}(\Sigma)) \times CE(\Sigma)$, is defined by* $A \overset{t}{\models}_\Sigma \varphi \Leftrightarrow \Big(\forall \alpha \in A(v \circ \overline{x}) \cdot \bigwedge_{i \in [k]} \{A(\chi.t_i)(\alpha) = A(\chi.u_i)(\alpha)\} \Rightarrow A(\chi.t_{k+1})(\alpha) = A(\chi.u_{k+1})(\alpha)\Big)$.

The existential satisfaction relation, *$\overset{e}{\models}_\Sigma \subseteq \mathsf{Obj}(\mathbf{PAlg}(\Sigma)) \times CE(\Sigma)$, is defined by* $A \overset{e}{\models}_\Sigma \varphi \Leftrightarrow \Big(\forall \alpha \in \Big\{a \in A(v \circ \overline{x}) \mid a \in \cap_{i \in [k]}(\partial(A(\chi.t_i)) \cap \partial(A(\chi.u_i)))\Big\} \cdot \bigwedge_{i \in [k]} \{A(\chi.t_i)(\alpha) = A(\chi.u_i)(\alpha)\} \Rightarrow \quad \alpha \in \partial(A(\chi.t_{k+1})) \wedge \alpha \in \partial(A(\chi.u_{k+1})) \wedge A(\chi.t_{k+1})(\alpha) = A(\chi.u_{k+1})(\alpha)\Big)$.

The *weak satisfaction relation*, which is the same as the existential satisfaction relation, except that $\alpha \in \cap_{i \in [k+1]}(\partial(A(\chi.t_i)) \cap \partial(A(\chi.u_i)))$, is closer to the notion of guarded satisfaction, being defined below. Using weak satisfaction will not change any of the results presented here. Existential satisfaction is considered more versatile than weak satisfaction, and is the one most commonly used.

Definition 16 (Guarded satisfaction). *Let $\Gamma = (S, UF, PF, \mathrm{dom}, \mathrm{cod}, \delta, \gamma)$ be a guarded signature and let $\varphi = \chi.(\{(t_1, u_1), \ldots, (t_k, u_k)\}, (t_{k+1}, u_{k+1})) \in CE(PL(\Gamma))$ be a conditional equation.*

The guarded satisfaction relation *for Γ, $\overset{g}{\models}_\Gamma \subseteq \mathsf{Obj}(\mathbf{GAlg})(\Gamma) \times CE(\Gamma)$, where* **GAlg** *is* **GTAlg** *or* **GPAlg** *as appropriate, is defined by* $A \overset{g}{\models}_\Gamma \varphi \Leftrightarrow \Big(\forall \alpha \in \bigcap_{i \in [k+1]}(G(A, \chi.t_i) \cap G(A, \chi.u_i)) \cdot \bigwedge_{i \in [k]} \{A(\chi.t_i)(\alpha) = A(\chi.u_i)(\alpha)\} \Rightarrow A(\chi.t_{k+1})(\alpha) = A(\chi.u_{k+1})(\alpha)\Big)$.

Example 4. We may now provide the standard group axioms for the signature Σ_{Group} from Example 1.

$$
\begin{array}{ll}
\textbf{spec}\ \mathsf{Group} = & \\
\quad \textbf{sorts} & \mathsf{group} \\
\quad \textbf{opns} & \odot : \rightarrow \mathsf{group} \\
 & \oplus : \mathsf{group} \times \mathsf{group} \rightarrow \mathsf{group} \\
 & \ominus : \mathsf{group} \rightarrow \mathsf{group} \\
\quad \textbf{vars} & a, b, c : \mathsf{group} \\
\quad \textbf{axioms} & \odot \oplus a = a = a \oplus \odot \\
 & (a \oplus b) \oplus c = a \oplus (b \oplus c) \\
 & a \oplus (\ominus a) = \odot = (\ominus a) \oplus a
\end{array}
$$

This identifies $\odot$ as the unit element of the group, $\oplus$ as associative, and $\ominus$ as the inverse operation with respect to $\oplus$. The Group variety for total satisfaction includes the integers and the rationals with 0 as unit element, addition as $\oplus$ and minus for $\ominus$. It does not include the integers with 1, multiplication and division, since only 1 and -1 have multiplicative inverses among the integers. It does not include the rationals with 1, multiplication and division either, since 0 does not have a multiplicative inverse. Using existential satisfaction and partial algebras will change this, since letting $\ominus$ be undefined on the problematic elements will be inconsistent with the third axiom which implicitly ensures definedness of $\ominus$ on all arguments.

Using weak satisfaction would circumvent the forced definedness of $\ominus$ on all values, so we could then make $\ominus$ undefined on the problematic values and permit the last two models. However, weak satisfaction does not allow error recovery, as any value, even an error value, returned by $\ominus$ must obey all axioms of the specification.

Example 5. We may also provide the guarded signature Γ_{GGroup} from Example 3 with the group axioms.

$$
\begin{array}{ll}
\textbf{spec}\ \mathsf{GGroup} = & \\
\quad \textbf{sorts} & \mathsf{group} \\
\quad \textbf{opns} & \odot : \rightarrow \mathsf{group} \\
 & \oplus : \mathsf{group} \times \mathsf{group} \rightarrow \mathsf{group} \\
 & \ominus : \mathsf{group} \rightarrow \mathsf{group} \\
\quad \textbf{guard}(\ominus) & = (\mathsf{Bool}, \mathsf{true}, \mathsf{invertible}) \\
\quad \textbf{vars} & a, b, c : \mathsf{group} \\
\quad \textbf{axioms} & \odot \oplus a = a = a \oplus \odot \\
 & (a \oplus b) \oplus c = a \oplus (b \oplus c) \\
 & a \oplus (\ominus a) = \odot = (\ominus a) \oplus a
\end{array}
$$

The axioms for $\ominus$ are now protected, and may not be relevant for all elements in the carrier for group. The quantification on the variable a in the last axiom is restricted to values such that $\mathsf{invertible}(a) = \mathsf{true}$. Models of this specification includes those for the plain group for the sort group and operators $\odot$, $\oplus$ and $\ominus$,

irrespectively of the models for Bool, true and invertible. This holds, e.g., when the carrier for Bool has only one element. If the carrier for Bool has at least two elements, and invertible($\odot$) is distinct from true, then the rationals with 1, multiplication and division will be a model for GGroup. Also, if invertible only takes on the value true for 1 and -1, the integers with 1, multiplication and division will also be a model.

In this example, as in general, a sort like Bool will not have a two-valued carrier. Any value that is distinct from true will implicitly be understood as false when guarding $\ominus$. These non-true values may be significant or non-significant (error values), and the implicit sort-guards will automatically distinguish between these. This also goes for the carrier for group. For the last two models mentioned, $\ominus(0)$ will be an insignificant error value or a significant recovery value, as determined by the sort-guard, if $\ominus(0)$ is defined. In any case, this value need not obey the third axiom since $\ominus$ is protected by invertible. If the value returned by $\ominus$ is non-significant, it will in fact not have to obey any of the axioms of the specification, since the axioms are only relevant for significant values and significant arguments. Note that guarded satisfaction will behave the same way whether a total or a partial algebra is chosen as the model for GGroup.

Proposition 2. *For $z = t, e, g, g$, for* $\mathbf{Sign} = \mathbf{Sig},\ \mathbf{Sig}, \mathbf{CGSig}, \mathbf{CGSig}$ *and for* $\mathbf{Alg} = \mathbf{TAlg}, \mathbf{PAlg}, \mathbf{GTAlg}, \mathbf{GPAlg}$*, respectively:*

Given a signature morphism $\theta \in \mathbf{Sign}(\Theta, \Theta')$, a conditional equation $\varphi \in CE(\Theta)$, and an algebra $A' \in \mathsf{Obj}(\mathbf{Alg}(\Theta'))$, then $A' \overset{z}{\models}_{\Theta'} CE(\theta)(\varphi)$ if and only if $\mathbf{Alg}(\theta)(A') \overset{z}{\models}_{\Theta} \varphi$.

The proof is standard and follows the same pattern in all cases.

In each of the cases above, let $\overset{z}{\models}$ denote the family of satisfaction relations $\overset{z}{\models}_{\Theta}$ for $\Theta \in \mathsf{Obj}(\mathbf{Sign})$.

Proposition 3. *If A and B are care-equivalent guarded, total or partial, algebras over a guarded signature Γ, then $A \overset{g}{\models}_{\Gamma} \varphi$ if and only if $B \overset{g}{\models}_{\Gamma} \varphi$.*

Proof. Guarded satisfaction only relates to the significant arguments. Since care-equivalent algebras are identical on the significant arguments, they must satisfy the same axioms.

4 Institutions of Algebras

The concepts we developed in the previous section give us several institutions.

Proposition 4. *There exist institutions $\mathcal{TCEL}$ (Total Conditional Equational Logic), $\mathcal{PCEL}$ (Partial Conditional Equational Logic), $\mathcal{GTCEL}$ (Guarded Total*

Conditional Equational Logic), and $\mathcal{GPCEL}$ (Guarded Partial Conditional Equational Logic) given by

$$\begin{aligned}
\mathcal{TCEL} &= (\mathbf{Sig}, CE, \mathbf{TAlg}, \overset{t}{\models}) \\
\mathcal{PCEL} &= (\mathbf{Sig}, CE, \mathbf{PAlg}, \overset{e}{\models}) \\
\mathcal{GTCEL} &= (\mathbf{CGSig}, CE \circ PL, \mathbf{GTAlg}, \overset{g}{\models}) \\
\mathcal{GPCEL} &= (\mathbf{CGSig}, CE \circ PL, \mathbf{GPAlg}, \overset{g}{\models}).
\end{aligned}$$

Note that $\mathbf{Th}_0(\mathcal{TCEL}) = \mathbf{Th}_0(\mathcal{PCEL})$ and $\mathbf{Th}_0(\mathcal{GTCEL}) = \mathbf{Th}_0(\mathcal{GPCEL})$.

Both $\mathcal{TCEL}$ and $\mathcal{PCEL}$ have extensively studied entailment systems and proof calculi, and there also exist quite a lot of useful support tools for the corresponding logical systems.

4.1 Theories and Varieties

Given a plain signature Σ. The *inclusion functor* $\mathcal{I}_\Sigma$: $\mathbf{TAlg}(\Sigma) \to \mathbf{PAlg}(\Sigma)$ takes a total algebra to the partial algebra with the same carriers and functions. The functor $\mathcal{I}_\Sigma$ has a left adjoint $\mathcal{T}_\Sigma$: $\mathbf{PAlg}(\Sigma) \to \mathbf{TAlg}(\Sigma)$ (see [Bur86] or [HW95] for a proof). We call $\mathcal{T}_\Sigma$ the *free totalisation functor for* Σ.

The next functor relating total and partial model classes is defined on guarded signatures since it exploits the extra structure provided by the guards.

Definition 17 (Partialisation functor). *Given a guarded signature Γ.*

The partialisation functor $\mathcal{P}_\Gamma : \mathbf{GTAlg}(\Gamma) \to \mathbf{GPAlg}(\Gamma)$ *takes a guarded total algebra to the guarded partial algebra resulting from removing all insignificant elements from each carrier and restricting the domain of each operation to its significant arguments.*

It is easy to verify that $\mathcal{T} \circ \mathcal{P}(A) \equiv_\Gamma A$ for every $A \in \mathsf{Obj}(\mathbf{GTAlg}(\Gamma))$ and that $\mathcal{P} \circ \mathcal{T}(A) \equiv_\Gamma A$ for every $A \in \mathsf{Obj}(\mathbf{GPAlg}(\Gamma))$.

The inclusion, totalisation and partialisation functors all form natural transformations, indexed by the signatures, in **CAT**.

The inclusion functor restricts covariantly for plain and guarded presentations, e.g., for $(\Sigma, \Phi) \in \mathsf{Obj}(\mathbf{Th}_0(\mathcal{TCEL}))$ we have that $\mathcal{I}_\Sigma\big|_{Vmod_{\mathcal{TCEL}}(\Sigma,\Phi)}$: $Vmod_{\mathcal{TCEL}}(\Sigma, \Phi) \to Vmod_{\mathcal{PCEL}}(\Sigma, \Phi)$. The totalisation functor does not restrict nicely for plain presentations. The new elements being added to the carriers and domains of the operations will in general not obey the axioms.

Proposition 5. *Given a guarded signature Γ.*

The totalisation functor $\mathcal{T}_{PL(\Gamma)}$: $\mathbf{GPAlg}(\Gamma) \to \mathbf{GTAlg}(\Gamma)$ and the the partialisation functor $\mathcal{P}_\Gamma$: $\mathbf{GTAlg}(\Gamma) \to \mathbf{GPAlg}(\Gamma)$ both restrict covariantly for presentations.

Proof. For any $\varphi \in CE(\Gamma)$, $A \in \mathsf{Obj}(\mathbf{GTAlg}(\Gamma))$ and $M \in \mathsf{Obj}(\mathbf{GPAlg}(\Gamma))$ we have that $(A \overset{g}{\models}_\Gamma \varphi) \Leftrightarrow (\mathcal{P}(A) \overset{g}{\models}_\Gamma \varphi)$ and $(M \overset{g}{\models}_\Gamma \varphi) \Leftrightarrow (\mathcal{P}(M) \overset{g}{\models}_\Gamma \varphi)$ since

$\overset{g}{\models}_\Gamma$ only relates to the significant arguments of the operations of Γ, and these are preserved (and reflected) by both $\mathcal{P}$ and $\mathcal{T}$. □

We define the following algebra functors $\mathbf{IGTAlg} = \mathcal{T} \circ \mathcal{P} \circ \mathbf{GTAlg} : \mathbf{CGSig}^{\mathsf{op}} \to \mathbf{CAT}$ and $\mathbf{IGPAlg} = \mathcal{P} \circ \mathcal{T} \circ \mathbf{GPAlg} : \mathbf{CGSig}^{\mathsf{op}} \to \mathbf{CAT}$. Using these functors we get institutions of initial guarded total conditional equational logic $\mathcal{IGTCEL} = (\mathbf{CGSig}, CE \circ PL, \mathbf{IGTAlg}, \overset{g}{\models})$ and of initial guarded partial conditional equational logic $\mathcal{IGPCEL} = (\mathbf{CGSig}, CE \circ PL, \mathbf{IGPAlg}, \overset{g}{\models})$. The name derives from the fact that these varieties contain the initial algebra from each care-equivalence class as the unique representative of that care-equivalence class (see [HW95] for details).

All the guarded institutions share the same syntactic notions (guarded signatures and sentences), i.e., $\mathbf{Th}_0(\mathcal{GTCEL}) = \mathbf{Th}_0(\mathcal{GPCEL}) = \mathbf{Th}_0(\mathcal{IGTCEL}) = \mathbf{Th}_0(\mathcal{IGPCEL})$ all represent the same guarded theory.

We may relate the guarded theories to the plain theories using plainification. We have already defined plainification on signature categories and on presentations. Plainification on guarded theory-morphisms reduces to plainification of the underlying signature morphism. This gives us a plainification functor $PL : \mathbf{Th}_0(\mathcal{GTCEL}) \to \mathbf{Th}_0(\mathcal{TCEL})$ on theories. If we let $cgsig : \mathbf{Th}_0(\mathcal{GTCEL}) \to \mathbf{CGSig}$ and $sig : \mathbf{Th}_0(\mathcal{TCEL}) \to \mathbf{Sig}$ be the obvious first projection functors on theories, we get $sig \circ PL = PL \circ cgsig$ relating plainification of theories and plainification of signatures.

Proposition 6. *Plainification of guarded theories preserves varieties, i.e.,*

$$Vmod_{\mathcal{TCEL}} \circ PL^{\mathsf{op}} = Vmod_{\mathcal{GTCEL}} \text{ and } Vmod_{\mathcal{TCEL}} \circ PL^{\mathsf{op}} = Vmod_{\mathcal{GTCEL}}.$$

Proof. See [HW95]. □

Proposition 7.

$$\mathcal{T} \circ \mathcal{P} \circ Vmod_{\mathcal{GTCEL}} = Vmod_{\mathcal{IGTCEL}}$$
$$\mathcal{P} \circ \mathcal{T} \circ Vmod_{\mathcal{GPCEL}} = Vmod_{\mathcal{IGPCEL}}.$$

Proof. Follows from the observation that care-equivalent algebras satisfy the same axioms, and that, for every $(\Gamma, \Phi) \in \mathsf{Obj}(\mathbf{Th}_0(\mathcal{GTCEL}))$ we have that $Vmod_{\mathcal{IGTCEL}}(\Gamma, \Phi)$ is a full subcategory of $Vmod_{\mathcal{GTCEL}}(\Gamma, \Phi)$. Likewise for the varieties $Vmod_{\mathcal{IGPCEL}}$ and $Vmod_{\mathcal{GPCEL}}$. □

Proposition 8. *The categories $Vmod_{\mathcal{IGTCEL}}(\Gamma, \Phi)$ and $Vmod_{\mathcal{IGPCEL}}(\Gamma, \Phi)$ are isomorphic for every $(\Gamma, \Phi) \in \mathsf{Obj}(\mathbf{Th}_0(\mathcal{GTCEL}))$.*

Proof. Given an arbitrary guarded signature Γ. For $A \in \mathsf{Obj}(\mathbf{IGTAlg}(\Gamma))$ we easily see that $\mathcal{T} \circ \mathcal{P}(A) = A$ and that for $h \in \mathbf{IGTAlg}(\Gamma)(A, A')$ we have that $\mathcal{T} \circ \mathcal{P}(h) = h$. Likewise, $\mathcal{P} \circ \mathcal{T}(M) = M$ and $\mathcal{P} \circ \mathcal{T}(w) = w$ for $M \in \mathsf{Obj}(\mathbf{IGPAlg}(\Gamma))$ and $w \in \mathbf{IGPAlg}(\Gamma)(M, M')$. □

Obviously, **IGTAlg** is isomorphic to **IGPAlg** with $\mathcal{P}$ and $\mathcal{T}$ as the isomorphisms.

4.2 Relating the Institutions

The next series of theorems lead to the main claim of this paper, that in the context of a guarded specification, we will not be able to distinguish between total or partial models from a logical viewpoint.

Theorem 2. *There is a simple map of institutions* $(\zeta, \alpha, \beta) : \mathcal{GTCEL} \rightarrow \mathcal{TCEL}$ *which is surjective.*

Proof. We need to define the functor ζ on theories (and ζ_1 on signatures), and the natural transformations α indexed by guarded signatures and β indexed by guarded presentations, and show that they have the necessary properties.

- Define the functor $\zeta = PL : \mathbf{Th}_0(\mathcal{GTCEL}) \rightarrow \mathbf{Th}_0(\mathcal{TCEL})$ and the functor $\zeta_1 = PL : \mathbf{CGSig} \rightarrow \mathbf{Sig}$.
- Define the natural transformation $\alpha : (CE \circ PL) \rightarrow CE \circ PL$ by $\alpha_\Gamma(\varphi) = \mathbf{d}(\Gamma, \varphi)$, the "translation of axioms" part of PL on presentations.
- Define the natural transformation $\beta : Vmod_{\mathcal{TCEL}} \circ PL^{\mathsf{op}} \rightarrow Vmod_{\mathcal{GTCEL}}$ as the identity natural transformation since $Vmod_{\mathcal{TCEL}} \circ PL^{\mathsf{op}} = Vmod_{\mathcal{GTCEL}}$.

This satisfies the necessary conditions:

- We have that $sig \circ \zeta = \zeta_1 \circ cgsig$ since $sig \circ \zeta = sig \circ PL = PL \circ cgsig$ (with PL on signatures) and $\zeta_1 \circ cgsig = PL \circ cgsig$.
- $\zeta(\Gamma, \Phi) = PL(\Gamma, \Phi) = PL(\Gamma, \emptyset) \cup (PL(\Gamma), \alpha_\Gamma(\Phi)) = \zeta(\Gamma, \emptyset) \cup (\zeta_1(\Gamma), \alpha_\Gamma(\Phi))$.
- Let $sp : \mathbf{Th}_0(\mathcal{TCEL}) \rightarrow \mathbf{Set}$ be the obvious second projection. For each $\Gamma \in \mathsf{Obj}(\mathbf{CGSig})$, $\varphi \in CE(PL(\Gamma))$ and $M' \in Vmod_{\mathcal{TCEL}}(PL(\Gamma, \emptyset))$, we have that $\beta_{(\Gamma,\emptyset)}(M') \overset{g}{\models}_\Gamma \varphi \Leftrightarrow M' \overset{g}{\models}_\Gamma \varphi \Leftrightarrow M' \overset{t}{\models}_{PL(\Gamma)} sp \circ PL(\Gamma, \varphi) \Leftrightarrow M' \overset{t}{\models}_{PL(\Gamma)} sp(PL(\Gamma, \emptyset) \cup (PL(\Gamma), \alpha_\Gamma(\varphi))) \Leftrightarrow M' \overset{t}{\models}_{PL(\Gamma)} sp(PL(\Gamma, \emptyset)) \cup sp(PL(\Gamma), \alpha_\Gamma(\varphi)) \Leftrightarrow M' \overset{t}{\models}_{PL(\Gamma)} sp(PL(\Gamma), \alpha_\Gamma(\varphi)) \Leftrightarrow M' \overset{t}{\models}_{PL(\Gamma)} \alpha_\Gamma(\varphi)$ since $M' \overset{t}{\models} sp(PL(\Gamma, \emptyset))$ by default.
- Since β is the identity natural transformation it is automatically surjective. □

Theorem 3. *There is a simple map of institutions* $(\zeta, \alpha, \beta) : \mathcal{GPCEL} \rightarrow \mathcal{PCEL}$ *which is surjective.*

Proof. Let sp, ζ, ζ_1 and α be as above.

- Define the natural transformation $\beta : Vmod_{\mathcal{PCEL}} \circ PL^{\mathsf{op}} \rightarrow Vmod_{\mathcal{GPCEL}}$ as the identity natural transformation since $Vmod_{\mathcal{PCEL}} \circ PL^{\mathsf{op}} = Vmod_{\mathcal{GPCEL}}$.

This satisfies the necessary conditions:

- The constraints on syntax have been shown for $\mathcal{GTCEL}$.

– For each $\Gamma \in \mathsf{Obj}(\mathbf{CGSig})$, $\varphi \in CE(PL(\Gamma))$ and $M' \in Vmod_{\mathcal{PCEL}}(PL(\Gamma,\emptyset))$ we get: $\beta_{(\Gamma,\emptyset)}(M') \overset{g}{\models}_{\Gamma} \varphi \Leftrightarrow M' \overset{e}{\models}_{PL(\Gamma)} sp(PL(\Gamma,\emptyset)) \cup sp(PL(\Gamma), \alpha_{\Gamma}(\varphi)) \Leftrightarrow M' \overset{e}{\models}_{PL(\Gamma)} \alpha_{\Gamma}(\varphi)$ since $M' \overset{e}{\models} sp(PL(\Gamma,\emptyset))$ by default.
– Since β is the identity natural transformation it is automatically surjective.

□

Theorem 4. *There is a simple map of institutions* $(\zeta, \alpha, \beta) : \mathcal{IGTCEL} \rightarrow \mathcal{GTCEL}$ *which is surjective.*

Proof. Let *sp* be as before. Since the theories are the same, ζ, ζ_1 are identity functors and α is the identity natural transformation.

– Define $\beta = \mathcal{T} \circ \mathcal{P} : Vmod_{\mathcal{GTCEL}} \circ \zeta^{\mathsf{op}} \rightarrow Vmod_{\mathcal{IGTCEL}}$.

This satisfies the necessary conditions:

– We have that $cgsig \circ \zeta = \zeta_1 \circ cgsig$ since ζ and ζ_1 are identities.
– $\zeta(\Gamma, \Phi) = (\Gamma, \Phi) = \zeta(\Gamma, \emptyset) \cup (\zeta_1(\Gamma), \alpha_{\Gamma}(\Phi))$ since all these functions and functors are identities.
– For each $\Gamma \in \mathsf{Obj}(\mathbf{CGSig})$, $\varphi \in CE(\Gamma)$ and $M' \in Vmod_{\mathcal{GTCEL}}(PL(\Gamma,\emptyset))$, we have that $\beta_{(\Gamma,\emptyset)}(M') \overset{g}{\models}_{\Gamma} \varphi \Leftrightarrow \mathcal{T}(\mathcal{P}(M')) \overset{g}{\models}_{\Gamma} \varphi \Leftrightarrow M' \overset{g}{\models}_{\Gamma} \varphi \Leftrightarrow M' \overset{g}{\models}_{\Gamma} \alpha_{\Gamma}(\varphi)$ since $M' \equiv_{\Gamma} \mathcal{T}(\mathcal{P}(M'))$ and by Proposition 3 they will satisfy the same formulas.
– Since $\beta = \mathcal{T} \circ \mathcal{P}$ it is surjective.

□

Theorem 5. *There is a simple map of institutions* $(\zeta, \alpha, \beta) : \mathcal{IGPCEL} \rightarrow \mathcal{GPCEL}$*, which is surjective.*

Proof. Similar to the previous proof. □

Theorem 6. *There is an isomorphism from* $\mathcal{IGTCEL}$ *to* $\mathcal{IGPCEL}$ *which is a surjective, simple map of institutions* $(\zeta, \alpha, \beta) : \mathcal{IGTCEL} \rightarrow \mathcal{IGPCEL}$.

Proof. Let *cgsig* and *sp* be as before. Since the theories are the same, ζ, ζ_1 are identity functors and α is the identity natural transformation.

– Define the natural transformation $\beta = \mathcal{T} : Vmod_{\mathcal{IGPCEL}} \circ \zeta^{\mathsf{op}} \rightarrow Vmod_{\mathcal{IGTCEL}}$, which is also an isomorphism between the categories.

This satisfies the necessary conditions. For the theories this has already been proved.

– For each $\Gamma \in \mathsf{Obj}(\mathbf{CGSig})$, $\varphi \in CE(\Gamma)$ and $M' \in Vmod_{\mathcal{IGPCEL}}(PL(\Gamma,\emptyset))$, we have that $\beta_{(\Gamma,\emptyset)}(M') \overset{g}{\models}_{\Gamma} \varphi \Leftrightarrow \mathcal{T}(M') \overset{g}{\models}_{\Gamma} \varphi \Leftrightarrow M' \overset{g}{\models}_{\Gamma} \varphi \Leftrightarrow M' \overset{g}{\models}_{\Gamma} \alpha_{\Gamma}(\varphi)$ since $\mathcal{T}$ preserves and reflects satisfaction.
– Since $\beta = \mathcal{T}$ is an isomorphism it is surjective.

Since all components of (ζ, α, β) are identities or isomorphisms, the map itself is an isomorphism. □

This allows us to prove our main theorem.

Theorem 7. *We may use entailment systems and proof calculi for $\mathcal{TCEL}$ for $\mathcal{IGTCEL}$ and use entailment systems and proof calculi for $\mathcal{PCEL}$ for $\mathcal{IGPCEL}$.*

Further, in the context of $\mathcal{IGTCEL}$ and $\mathcal{IGPCEL}$ entailment systems and proof calculi for $\mathcal{TCEL}$ and $\mathcal{PCEL}$ are indistinguishable.

Proof. The first two statements follow from composing the institution maps of the previous theorems. The last statement follows from the isomorphism institution maps between the institutions $\mathcal{IGTCEL}$ and $\mathcal{IGPCEL}$. □

This result can be extended to $\mathcal{GTCEL}$ and $\mathcal{GPCEL}$, but the proof will be more involved since the maps upwards from $\mathcal{IGTCEL}$ and $\mathcal{IGPCEL}$ are not surjective on model classes. We should also be able to allow weak guarded signature morphisms, i.e., guarded signature morphisms which also may map protected operators to unprotected operators.

The model classes of $\mathcal{IGTCEL}$ and $\mathcal{IGPCEL}$ contain exactly one representative from each of the care-equivalence classes, a representative that is canonical in the sense that it is initial in the care-equivalence class [HW95]. A care-equivalence class does not contain final algebras, but there are other choices, such as error-algebras, which may be interesting for certain purposes. How to select these, and how those classes relate back to $\mathcal{GTCEL}$ and $\mathcal{GPCEL}$ is open for investigation.

5 Conclusion

We have shown that the concept of guarded algebra provides a very nice correspondence between total and partial varieties using the inclusion, totalisation and partialisation model functors. Further, it allows a logical framework insensitive to the choice of total or partial models. We may choose the entailment systems and proof calculi which are most simple to work with, irrespectively of whether total models or partial models are best suited to understand the problem domain. This may be useful when, for example, investigating program semantics with detectable and undetectable errors.

The practical use may be hampered by the number of conditions generated by plainification, the translation from guarded to plain presentations. Other translation schemes that yield plain presentations that are more efficient to work with are being considered. Such a scheme may not yield the full model classes of $\mathcal{GTCEL}$ or $\mathcal{GPCEL}$ after the translation, but it seems that as long as each care-equivalence class is represented, the translation will be adequate from the logical aspect.

Further investigation of the relationship between guarded algebras and similar specification formalisms, such as membership algebras [Mes98], may allow the

transferal of additional useful logical reasoning and rewrite tools to the guarded context.

We also want to extend the notion of guardedness beyond conditional equational specifications. First-order logic is especially interesting. A positive result here may be useful in the context of the algebraic specification language CASL [Mos97], which admits both partial and total models.

References

[Bur86] Peter Burmeister. *A Model Theoretic Oriented Approach to Partial Algebras*. Akademie-Verlag, 1986.

[Cer93] Maura Cerioli. *Relationships between Logical Formalism*. PhD thesis, Università de Pisa–Genova–Udine, 1993.

[GDLE84] M. Gogolla, K. Drosten, U. Lipeck, and H.-D. Ehrich. Algebraic and operational semantics of specifications allowing exceptions and errors. *Theoretical Computer Science*, 34:289–313, 1984.

[GTW78] Joseph A. Goguen, J. W. Thatcher, and Eric G. Wagner. An initial algebra approach to the specification, correctness, and implementation of abstract data types. In R. T. Yeh, editor, *Current Trends in Programming Methodology, IV, Data Structuring*, pages 80–149. Prentice-Hall, 1978.

[HL89] Ivo Van Horebeek and Johan Lewi. *Algebraic Specifications in Software Engineering – an introduction*. International Series of Monographs on Computer Science. Springer–Verlag, Berlin, 1989.

[HW95] Magne Haveraaen and Eric G. Wagner. Guarded algebras and data type specification. Technical Report 108, Department of Informatics, University of Bergen, P.O.Box 7800, N-5020 Bergen, Norway, October 1995.

[Kre87] Hans-Jörg-Kreowski. Partial algebras flow from algebraic specifications. In *Proc. ICALP 87*, volume 267 of *Lecture Notes in Computer Science*, pages 521–530. Springer Verlag, 1987.

[KM95] Hans-Jörg-Kreowski and Till Mossakowski. Equivalence and difference between institutions: simulating Horn Clause Logic with based algebras. *Math Struct. in Comp. Science* 5:189–215, 1995.

[Mes89] J. Meseguer. General logics. In *Proc. Logic Colloquium '87*. North-Holland, 1989.

[Mes98] José Meseguer. Membership algebra as a logical framework for equational specification. In Francesco Parisi Presicce, editor, *Recent Trends in Algebraic Development Techniques*, volume 1376 of *Lecture Notes in Computer Science*, pages 18–61. Springer Verlag, 1998.

[Mos95] Till Mossakowski. Equivalences among various logical frameworks of partial algebras. Bericht Nr, 4/95, Universität Bremen, Fachbereich Mathematik und Informatik, 1995.

[Mos93] Peter D. Mosses. The use of sorts in algebraic data type specification. In *Recent Trends in Data Type Specification*, pages 66–91. LNCS 655, 1993.

[Mos97] Peter D. Mosses. CoFI: The common framework initiative for algebraic specification and development. In Michel Bidoit and Max Dauchet, editors, *TAPSOFT'97: Theory and Practice of Software Development*, volume 1214 of *Lecture Notes in Computer Science*, pages 115–137. Springer-Verlag, 1997.

[Rei87] Horst Reichel. *Initial Computability Algebraic Specifications, and Partial Algebras*. Clarendon Press, Oxford, 1987