Abstract derivations, equational logic and interpolation

Renardel de Lavalette, Gerard R.

Published in:
Structures and Deduction - the Quest for the Essence of Proofs

Publication date:
2005

Interpolation in equational logic via abstract derivations

14 July 2005

Gerard R. Renardel de Lavalette

Department of Mathematics and Computing Science
University of Groningen, the Netherlands

Abstract. We define abstract derivations for equational logic and use them to prove the interpolation property.

1 Introduction

In this paper, we introduce a notion of abstract derivation for equational logic and use it to prove the interpolation property. The work reported here has the character of an experiment, intended to sharpen and test our initially rather vague ideas about abstract derivations. These ideas arose from a feeling of dissatisfaction with the low level of abstraction in traditional proof theory, where derivations are trees consisting of sequents, i.e. strings of symbols, with great redundancy by repeating in every proof step the parts of a sequent that do not change. As a consequence, operations on derivations (normalization, e.g. via cut elimination, interpolant extraction) only admit a precise definition in local terms, on the level of proof steps, and global properties are left to intuition. More particular, in [4] and [16] we were able to prove interpolation for several fragments of intuitionistic propositional logic, but we admit that full understanding of what is really happening on a global level in our proofs is lacking, because of the reasons sketched above.

The choice for equational logic as a basis for the elaboration of our ideas was motivated by two reasons. Firstly, the prooftheoretical proof for interpolation in equational logic by Rodenburg in [12] is rather involved, especially when compared with the algebraic proof by the same author in [11], and constitutes a challenge for proof theory to try to fill the gap with model theory. (Recently, we discovered the unpublished note [15] by Van Oostrom with a more perspicuous proof based on rewrite theory.) Secondly, equational logic is a very general and rather strong system: it is undecidable, and virtually all propositional, modal and linear logics can be embedded in it. Moreover, proof theory for equational logic seems to be an underdeveloped area of research.

The experiment in abstract derivation design is by no means finished yet, but we think the time has come to report on the first results. After a short sketch of the ideas behind abstract derivations and a survey of interpolation in equational

logic, we present the main definitions, prove soundness and completeness, derive some relevant properties and prove the interpolation theorem. We end with some suggestions for further research.

1.1 Abstract derivations

The notion of abstract derivation we present here is based on an abstract view on the traditional notion of derivation for equational logic. We see two fundamental and general principles at work here. Firstly the idea of *matching*: e.g. $f(s) \equiv f(t)$ follows from $s \equiv t$ since the f in $f(s)$ matches the f in $f(t)$. Secondly the idea of *abstraction*, or its dual instantiation: $r(t) \equiv s(t)$ follows from $r(x) \equiv s(x)$ since the latter equation is to be read as '$r(x)$ and $s(x)$ are equal for all x'. We consider the fact that $\equiv$ is an equivalence relation to be less fundamental for the proof system: it is of course essential for a logic called equational, but another interesting logic may be obtained if we would replace $\equiv$ by $\leq$, a partial order.

Now the idea of abstract derivation can be explained as follows. Start with an abstract representation of terms: the obvious choice is trees consisting of nodes labeled by signature elements (is there an alternative?). Then abstract equations become pairs of nodes, and binary relations on nodes are sets of equations. The proof rules become operations on relations, so e.g. R^e, the least equivalence relation containing R, corresponds with application of the proof rules of reflexivity, symmetry and transitivity. For the congruence rule, we put $\mathsf{cong}(R) = M \cap \mathsf{lift}(R)$. Here $\mathsf{lift}(R)$ is the collection of pairs (k, l) such that k and l represent terms with an equal number of direct subterms, and for all k', l' representing corresponding direct subterms we have $(k', l') \in R$. M is the matching relation: we have that $(k, l) \in M$ implies that k and l are labeled by the same signature element, but the converse implication does not hold in general. The role of M is to regulate the development of the abstract derivation. In order to deal with the instantiation rule, we work with abstractions, an inverse of substitutions. Abstractions α are partial mappings from nodes to variables, and when applied to a term structure T they yield a new term structure T_α where some terms are replaced by variables. Now an abstract derivation is a tuple $D = \langle T, M, \alpha \rangle$ where T is an abstract term structure, M is a matching relation on K, and α is an abstraction. An operator $\mathsf{der}(E) = \mu R.(E \cup \mathsf{cong}(R))^e$, defined for $E \subseteq K^2$, yields the set of equalities that are derived in D from E. Moreover, the abstraction α should be justified by E. See Definition 3 for the details.

For the moment, we choose to represent the steps in the proof and its conclusion implicitly: only the premiss E is present in the representation, its consequences after n proof steps can be *computed* by applying the operator $\lambda R.(E \cup \mathsf{cong}(R))^e$ n times. For our experiment with interpolation, this representation will do, but for other purposes it may be useful to work with other, more explicit variants.

For conciseness' sake, we will often drop the epitheton *abstract*, and refer to derivations when we mean abstract derivations.

1.2 Equational logic

A *signature* SIG is a collection of constants and function symbols. The arity of a signature element $s \in \mathsf{SIG}$ is given by $\mathsf{arity}(s) \in \mathbb{N}$. VAR is an infinite collection of variable symbols, with $\mathsf{SIG} \cap \mathsf{VAR} = \emptyset$. For variables $x \in \mathsf{VAR}$, $\mathsf{arity}(x) = 0$. Terms built from signature elements and variables are defined as usual; we write $\mathsf{sig}(t)$ for the collection of signature elements that occur in term t. $[x := s]t$ denotes substitution of term s for all occurrences of variable x in term t, and $[x_i := s_i]_{i<n}t$ is used for simultaneous substitution. Equations between terms are denoted by $s \equiv t$: they are the formulas φ of equational logic. Sequents are of the form $\Gamma \vdash \varphi$ where Γ is a collection of equations. As usual, the derivability relation $\vdash$ is the least relation satisfying

assumption	$\Gamma \vdash \varphi$ if $\varphi \in \Gamma$
reflexivity	$\Gamma \vdash t \equiv t$
symmetry	$\Gamma \vdash s \equiv t \;\Rightarrow\; \Gamma \vdash t \equiv s$
transitivity	$\Gamma \vdash r \equiv s \;\&\; \Gamma \vdash s \equiv t \;\Rightarrow\; \Gamma \vdash r \equiv t$
congruence	$\Gamma \vdash s_0 \equiv t_0 \;\&\; \ldots \;\&\; \Gamma \vdash s_{n-1} \equiv t_{n-1}$ $\Rightarrow\; \Gamma \vdash f(s_0, \ldots, s_{n-1}) \equiv f(t_1, \ldots, t_{n-1})$
instantiation	$\Gamma \vdash \varphi \;\Rightarrow\; \Gamma \vdash [x := t]\varphi$

The instantiation rule can be generalized to

$$\Gamma \vdash t_0 \equiv t_1 \;\&\; \Gamma \vdash t_1 \equiv t_2 \;\&\; \ldots \;\&\; \Gamma \vdash t_{n-1} \equiv t_n \;\&\; \Gamma \vdash \varphi(x, \ldots, x) \;\Rightarrow\; \Gamma \vdash \varphi(t_0, \ldots, t_n)$$

which says that we may substitute provably equal terms for different occurrences of the same variable. Our Definition 3 of abstract derivation is based on a weaker version of this principle, viz. where the equalities $t_i \equiv t_j$ are derivable without using the rules for reflexivity, symmetry and transitivity.

1.3 Interpolation

Interpolation is the following property;

> if $\Gamma, \Delta \vdash \varphi$, then there is a collection of formulae Θ such that
> $\Gamma \vdash \theta$ for every $\theta \in \Theta$, $\Theta, \Delta \vdash \varphi$, and $\mathsf{sig}(\Theta) \subseteq \mathsf{sig}(\Gamma) \cap \mathsf{sig}(\Delta \cup \{\varphi\})$.

Θ is called the *interpolant*. The first formulation and proof are by Craig in [3], first for classical predicate logic without function symbols; the case with function symbols is reduced to the former case by replacing function symbols by predicates. Craig's proof is prooftheoretical and proceeds via proof normalization. Since then, interpolation has been shown for many logics, with either prooftheoretic or modeltheoretic means. All prooftheoretic proofs work with proof normalization, usually obtained via cut elimination. This can be seen as a disadvantage, since proof normalization may lead to an exponential increase in size.

Interpolation as defined above does not hold for equational logic, as is shown in the following simple counterexample: take

$$\Gamma = \{f(a) \equiv b, f(c) \equiv d\}, \Delta = \{a \equiv c\}, \varphi = (b \equiv d).$$

The only possible interpolant would be something like $(a \equiv c) \rightarrow (b \equiv d)$, but this is not expressible in equational logic.

However, the weaker version of interpolation with $\Delta = \emptyset$ is valid for equational logic. It was first proved by Rodenburg in [11] with a rather short and perspicuous algebraic proof, using Birkhoff's HSP theorem. (Birkhoff's theorem states that a class of algebras K is equational, i.e. characterized by a set of equations, iff it is a variety, i.e. closed under homomorphic images, subalgebras, and direct products; see e.g. [8].) Rodenburg's proof is not constructive in the sense that the proof does not contain an effective method to construct the interpolant. In [12], Rodenburg gives a prooftheoretical proof of the same theorem, which is constructive but not very perspicuous. The proof transforms a derivation in equational logic step by step into a derivation in a related system, from which an interpolant can be obtained easily. (As Rodenburg points out, an apparently different interpolant construction for equational logic is already implicit in the proof of the main theorem of [9] by Pigozzi.)

A constructive proof of interpolation for equational logic can also be extracted from the prooftheoretical proof for interpolation for predicate logic with function symbols given by Felscher in [6]. This proof (which is also rather involved) does not eliminate function symbols, but uses what is called *Takeuti's Lemma* (Felscher gives the obscure reference [14]) to eliminate spurious function symbols from a candidate interpolant. Takeuti's Lemma indicates when, in a derivation, a term t can be replaced by a variable x, so it is a kind of inverse substitution property. Part of our proof of the Interpolation theorem is related to Takeuti's Lemma.

Let us see what has to be done for the construction of an interpolant for a provable sequent $\Gamma \vdash \varphi$ in equational logic. The general situation is: there are enough candidate interpolants Θ, e.g. Γ or $\{\varphi\}$, but in general they do not satisfy the signature condition $\mathsf{sig}(\Theta) \subseteq \mathsf{sig}(\Gamma) \cap \mathsf{sig}(\varphi)$. This is caused by the proof rules, which may add signature elements to or eliminate them from the conclusion (observe that all proof rules leave the premiss Γ intact and only modify the conclusion φ). The congruence rule and the instantiation rule may add signature elements. In the transitivity rule, the term s is eliminated, and possibly some signature elements that occur in it. In that sense, the transitivity rule is comparable with the cut rule of propositional and predicate logic. There is no Transitivity Elimination Theorem, however, which would help us here. But we do have a partial result in that direction: Lemma 2, which splits an abstract derivation in two parts: in the first part no new signature elements occur, and the second part contains only congruence steps. This lemma is the first step in the proof of interpolation; we then need abstractions to eliminate signature elements that were introduced in the conclusion by the instantiation rule, in a way comparable to Felscher's use of Takeuti's Lemma as described in the previous paragraph.

2 Preliminaries

For any set X, X^* denotes the collection of finite sequences of elements of X. If $xs \in X^*$, then $\mathsf{lth}(xs)$ is the length of xs, and xs_i (where $0 \leq i < \mathsf{lth}(xs)$) denotes the i-th element of xs. We write $x \in xs$ to denote that x occurs in the sequence xs, and the empty sequence is denoted by $()$.

If $R \subseteq X \times Y$ is a relation between X and Y, then we can extend R to $R^{\otimes} \subseteq X^* \times Y^*$ by defining

$$R^{\otimes} = \{(xs, ys) \mid \mathsf{lth}(xs) = \mathsf{lth}(ys) \wedge \forall i < \mathsf{lth}(xs)\ (xs_i, ys_i) \in R\}$$

and likewise for functions, so

$$f^{\otimes}(x_0, \ldots, x_{n-1}) = (f(x_0), \ldots, f(x_{n-1}))$$

Composition of relations is defined as usual: $R \cdot S = \{(x, z) \mid \exists y(xRy \wedge ySz)\}$. If $f : X \to Y$, then $\mathsf{ker}(f) \subseteq X^2$ is the equivalence relation defined by

$$\mathsf{ker}(f) = \{(x, y) \mid f(x) = f(y)\}.$$

$\mathsf{car}(R)$, the *carrier* of R, is the least set X with $R \subseteq X^2$, so

$$\mathsf{car}(R) = \{x \mid \exists y((x, y) \in R \vee (y, x) \in R)\}$$

If f, g are partial functions, then we define $f{:}g$ (f *before* g) by

$$f{:}g = f \cup (g \restriction (\mathsf{dom}(g) - \mathsf{dom}(f)))$$

Here $\restriction$ denotes restriction: $f \restriction X = f \cap (X \times \mathsf{rg}(f))$. Observe that $:$ is associative: $f{:}(g{:}h) = (f{:}g){:}h$.

We write R^+ for the transitive closure of R, R^* for the reflexive transitive closure, and R^e for the smallest equivalence relation containing R; so $R^e = (R \cup R^{-1})^*$. Let $R \subseteq X^2$ and $Y \subseteq X$: we define

$$R \text{ respects } Y \text{ iff } R \subseteq Y^2 \cup (X - Y)^2$$

so if $(x, y) \in R$ then $x \in Y$ iff $y \in Y$. If $f : X \rightharpoonup Z$ is a partial function,

$$R \text{ respects } f \text{ iff } R \subseteq \mathsf{ker}(f) \cup (X - \mathsf{dom}(f))^2$$

Finally, we define an ordering $\sqsubseteq$ on set-valued functions f, g:

$$f \sqsubseteq g \text{ iff } \forall x \in \mathsf{dom}(f)\ f(x) \subseteq g(x)$$

3 Term structures and abstractions

3.1 Forests

The idea to represent terms by trees is straightforward. The context for an abstract derivation will be a *forest*, i.e. a collection of trees that represent terms.

A forest is a pair $F = \langle K, \mathsf{arg}\rangle$, where K is a collection of nodes, and $\mathsf{arg} : K \to K^*$; moreover, the direct subnode relation $[\mathsf{arg}]$ on K, defined by

$$[\mathsf{arg}] = \{(k, l) \mid k \in \mathsf{arg}(l)\}$$

is wellfounded. As a consequence, we have the following *subnode induction principle* for forests: for all $L \subseteq K$,

$$\forall k(\mathsf{arg}(k) \subseteq L \to k \in L) \to L = K \tag{1}$$

The subnode relation $\leq$ is defined as the transitive closure of $[\mathsf{arg}]$. The *arity* of a node k is defined as the length of its sequence of children: $\mathsf{arity}(k) = \mathsf{lth}(\mathsf{arg}(k))$. For $0 \leq i < \mathsf{arity}(k)$, we write $\mathsf{arg}_i(k)$ for $(\mathsf{arg}(k))_i$, the i-th element of $\mathsf{arg}(k)$: so $\mathsf{arg}(k) = (\mathsf{arg}_0(k), \ldots, \mathsf{arg}_{\mathsf{arity}(k)-1}(k))$. A *path* in F is a finite nonempty sequence $(k_0, \ldots, k_n) \in K^+$ satisfying $(k_{i+1}, k_i) \in [\mathsf{arg}]$ for $0 \leq i < n$, so the steps in a path go from a node to one of its direct subnodes.

Parallel to (1), we have a recursion principle for forests: if $f : (K \otimes X) \to X$, where $K \otimes X = \{(k, (x_0, \ldots, x_{n-1})) \mid n = \mathsf{arity}(k), x_0, \ldots, x_{n-1} \in X\}$, then there is a unique $g : K \to X$ with

$$g(k) = f(k)(g^{\otimes}(\mathsf{arg}(k)))$$

We define the operator lift on relations $R \subseteq K^2$ by

$$\mathsf{lift}(R) = \mathsf{arg} \cdot R^{\otimes} \cdot \mathsf{arg}^{-1}$$

As a consequence, we see that

$$(k, l) \in \mathsf{lift}(R) \Leftrightarrow \mathsf{arity}(k) = \mathsf{arity}(l) \wedge \forall i < \mathsf{arity}(k)\ (\mathsf{arg}_i(k), \mathsf{arg}_i(l)) \in R$$

lift will be used for the congruence step in derivations.

Definition 1 (Term structures). *A* term structure *over* SIG *is a triple* $T = \langle K, \mathsf{arg}, \sigma\rangle$*, where* $\langle K, \mathsf{arg}\rangle$ *is a forest, and* $\sigma : K \to \mathsf{SIG} \cup \mathsf{VAR}$ *preserves arity.*

So a term structure is a forest where every node is labeled with a signature element or a variable, in such a way that the arity of node and label correspond. It is clear that every node k represents a term: to obtain this term, we define (with recursion) the term operator term by

$$\mathsf{term}(k, T) = (\sigma(k))(\mathsf{term}^{\otimes}(\mathsf{arg}(k), T))$$

The formula operator form is defined on pairs of nodes:

$$\mathsf{form}(k, l, T) = (\mathsf{term}(k, T) \equiv \mathsf{term}(l, T))$$

The signature $\mathsf{sig}(k, T)$ of a node k is the signature of the term represented by k, so $\mathsf{sig}(k, T) = \mathsf{sig}(\mathsf{term}(k, T))$. (We trust that this overloading of sig will not confuse the reader.) Observe that sig satisfies

$$\mathsf{sig}(k, T) = (\{\sigma(k)\} \cap \mathsf{SIG}) \cup \bigcup \mathsf{sig}^{\otimes}(\mathsf{arg}(k), T)$$

term and sig are extended to sets of nodes by $\mathsf{term}(L,T) = \{\mathsf{term}(k,T) \mid k \in L\}$, and similarly for sig. form is extended to relations on nodes: $\mathsf{form}(R,T) = \{\mathsf{form}(k,l,T) \mid (k,l) \in R\}$.

We also define, for $\Sigma \subseteq \mathsf{SIG}$:

$$\begin{aligned} K_\Sigma &= \{k \in K \mid \sigma(k) \in \Sigma\} &&= \sigma^{-1}[\Sigma] \\ K_{\overline{\Sigma}} &= \{k \in K \mid \mathsf{sig}(k,T) \subseteq \Sigma\} &&= \mathsf{sig}^{-1}[\wp(\Sigma)] \end{aligned}$$

Observe that $K_{\overline{\Sigma}}$ and K_Σ are different and even incomparable. K_Σ contains all nodes that represent terms with their principal signature element in Σ, while $K_{\overline{\Sigma}}$ contains nodes representing terms that are made from variables and elements of Σ. So $k \in K_{\overline{\Sigma}} - K_\Sigma$ if $\sigma(k) \in \mathsf{VAR}$, and $k \in K_\Sigma - K_{\overline{\Sigma}}$ if $\sigma(k) \in \Sigma$, $\mathsf{arg}(k) = (l)$ and $\sigma(l) \in \mathsf{SIG} - \Sigma$.

To deal with the instantiation rule of equational logic, we introduce the notion of *abstraction*. It is a converse of substitution that we prefer for technical reasons.

Definition 2 (Abstraction). *Let $T = \langle K, \mathsf{arg}, \sigma\rangle$ be a term structure. A partial mapping $\alpha : K \stackrel{\cdot}{\to} \mathsf{VAR}$ is an* abstraction *of T if $\mathsf{rg}(\alpha) \cap \mathsf{rg}(\sigma) = \emptyset$.*
T_α, the result of applying α to T, is defined as $T_\alpha = \langle K, \mathsf{arg}_\alpha, \alpha{:}\sigma\rangle$, where arg_α is defined by

$$\begin{aligned} \mathsf{arg}_\alpha(k) &= () && \text{if } k \in \mathsf{dom}(\alpha) \\ &= \mathsf{arg}(k) && \text{if } k \notin \mathsf{dom}(\alpha) \end{aligned}$$

By the definition of $\alpha{:}\sigma$ (see section 2), we have

$$\begin{aligned} (\alpha{:}\sigma)(k) &= \alpha(k) \text{ if } k \in \mathsf{dom}(\alpha) \\ &= \sigma(k) \text{ if } k \notin \mathsf{dom}(\alpha) \end{aligned}$$

The idea behind the definition of abstraction is that, for $k \in \mathsf{dom}(\alpha)$, the term $\mathsf{term}(k)$ represented by k is replaced by the variable $\alpha(k)$. The restriction $\mathsf{rg}(\alpha) \cap \mathsf{rg}(\sigma) = \emptyset$ ensures that the variables in $\mathsf{rg}(\alpha)$ are fresh.

We give an example of a term structure T and an abstraction α in Figure 1. Nodes are indicated by their signature element (and their image under abstraction when defined), arrows correspond with arg. The terms represented by

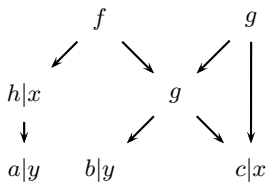

Fig. 1. A term structure with an abstraction.

T are

$$f(ha, g(b,c)),\ g(g(b,c),c),\ ha,\ g(b,c),\ a,\ b,\ c$$

and the terms represented by T_α are

$$f(x, g(y, x)),\ g(g(y, x), x),\ x \text{ (twice)},\ g(y, x),\ y \text{ (twice)}$$

4 Derivations

Now we can define derivations, the central notion of this paper.

Definition 3 (Derivation). *A* derivation D *is a triple* $\langle T, M, \alpha\rangle$ *where* $T = \langle K, \mathsf{arg}, \sigma\rangle$ *is a term structure,* $M \subseteq \mathsf{ker}(\sigma)$ *is an equivalence relation on* K *called the* matching relation *and* $\alpha : K \dot{\to} \mathsf{VAR}$ *is an abstraction of* T*. The* congruence operator cong*, the* derivability operator der *and the* weak derivability operator der^- *are defined by:*

$$\begin{aligned}\mathsf{cong}(E) &= M \cap \mathsf{lift}(E)\\ \mathsf{der}(E) &= \mu R.(E \cup \mathsf{cong}(R))^e\\ \mathsf{der}^-(E) &= \mu R.(E \cup \mathsf{cong}(R))\end{aligned}$$

We say that α *is* justified by E *if*

$$\mathsf{ker}(\alpha) \subseteq \mathsf{cong}(\mathsf{der}^-(E)) \tag{2}$$

For a formula φ *and a collection of formulas* Γ*, we define* abstract derivability *by*

$$\begin{aligned}\Gamma \vdash_{D,E} \varphi : {} & D = \langle K, \mathsf{arg}, \sigma, M, \alpha\rangle \text{ and } E \subseteq K^2 \text{ satisfy}\\ & E \text{ justifies } \alpha, \Gamma = \mathsf{form}(E, T) \text{ and } \varphi \in \mathsf{form}(\mathsf{der}(E), T)\end{aligned}$$

$$\Gamma \vdash_{\mathsf{abs}} \varphi \ : \text{there are } D, E \text{ such that } \Gamma \vdash_{D,E} \varphi$$

So a derivation is a term structure with a matching relation and an abstraction. The matching relation is a restriction on the possible pairs of terms that may be proved to be equal via congruence (i.e. via equality of corresponding direct subterms). We use M in the proof of the interpolation theorem as a locality restriction, regulating the application of congruence steps in an abstract derivation. Justification of abstraction α means that E proves $\mathsf{ker}(\alpha)$, the equalities that are implicit in α.

By definition, $\mathsf{der}(E)$ is the least equivalence relation containing E and closed under cong:

$$\mathsf{der}(E) = (E \cup \mathsf{cong}(\mathsf{der}(E)))^e \tag{3}$$

$$\text{if } (E \cup \mathsf{cong}(R))^e \subseteq R, \text{ then } \mathsf{der}(E) \subseteq R \tag{4}$$

We shall refer to these as the *defining property* and the *minimality property* of der, respectively. Similarly, we have for der^-:

$$\mathsf{der}^-(E) = E \cup \mathsf{cong}(\mathsf{der}^-(E)) \tag{5}$$

$$\text{if } E \cup \mathsf{cong}(R) \subseteq R, \text{ then } \mathsf{der}^-(E) \subseteq R \tag{6}$$

Moreover, we have

$$E \subseteq \mathsf{der}^-(E) = \mathsf{der}^-(\mathsf{der}^-(E)) \subseteq \mathsf{der}(E) = \mathsf{der}(\mathsf{der}(E)) \tag{7}$$

$E \subseteq \mathsf{der}^-(E) \subseteq \mathsf{der}(E) \subseteq \mathsf{der}(\mathsf{der}(E))$ and $\mathsf{der}^-(E) \subseteq \mathsf{der}^-(\mathsf{der}^-(E))$ are obvious. $\mathsf{der}(\mathsf{der}(E)) \subseteq \mathsf{der}(E)$ follows via the minimality property of der from $(\mathsf{der}(E) \cup \mathsf{cong}(\mathsf{der}(E)))^e \subseteq \mathsf{der}(E)$. Similarly for $\mathsf{der}^-(\mathsf{der}^-(E)) \subseteq \mathsf{der}^-(E)$.

4.1 Examples of abstract derivations

We give a simple example of an abstract derivation in Figure 2. It represents a derivation of the sequent

$$a \equiv b,\ b \equiv c,\ g_1 x \equiv g_2 x,\ g_2 h y \equiv d \ \vdash\ f g_1 h k a \equiv f d$$

As in the previous example, nodes are indicated by their signature element and their variable if any, and arrows correspond with arg. Moreover, single lines represent the matching relation M and double lines the equality relation E. Since all functions are unary, the diagram is planar.

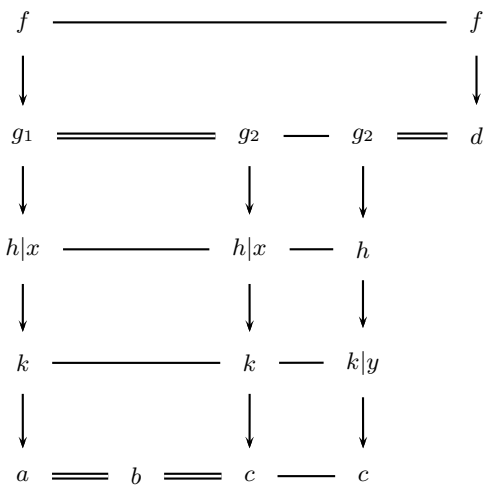

Fig. 2. A simple derivation

We give another derivation which is slightly more involved since it contains a binary function: see Figure 3. It represents

$$a \equiv f(x,x),\ y \equiv hgy,\ f(z, ghz) \equiv b \ \vdash\ a \equiv b$$

As a third example, we present in Figure 4 a derivation of the property $(a^{-1})^{-1} =$

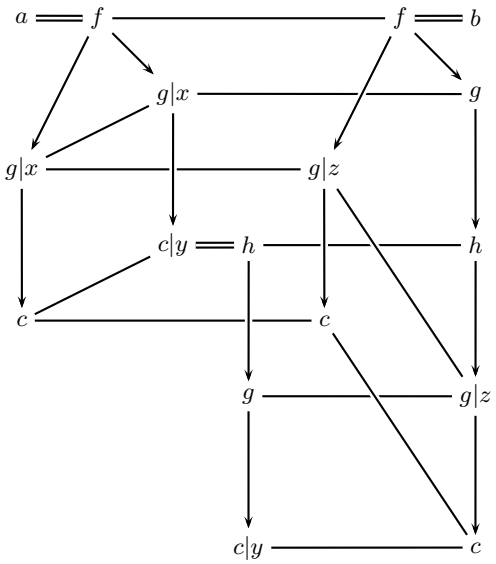

Fig. 3. A derivation with a binary function

a in group theory. In the usual algebraic notation, this reads

$$x = xe,\ e = x^{-1}x,\ x(yz) = (xy)z,\ ex = x \quad \vdash \quad (a^{-1})^{-1} = a$$

We use the following signature elements, all in prefix notation: m for multiplication, i for inverse, e for the unit element and a as a constant. Then the following sequent is derivable:

$$\begin{array}{l} x \equiv m(x,e), e \equiv m(ix,x), \\ m(x,m(y,z)) \equiv m(m(x,y),z), m(e,x) \equiv x \quad \vdash \quad iia \equiv a \end{array}$$

When we forget the abstraction and focus on the basic derivation, we have the following derivable sequent:

$$\begin{array}{ll} iia \equiv m(iia,e),\ e \equiv m(ia,a), & \\ m(iia,m(ia,a)) \equiv m(m(iia,ia),a) & \\ m(iia,ia) \equiv e,\ m(e,a) \equiv a & \vdash \quad iia \equiv a \end{array}$$

this is represented by the compact basic derivation in Figure 5:

4.2 Discussion about justification

The choice for (2), the definition of justification, is one from several alternatives. The most conservative choice is

$$\mathsf{ker}(\alpha) \subseteq \mathsf{der}^{-}(\emptyset) \tag{8}$$

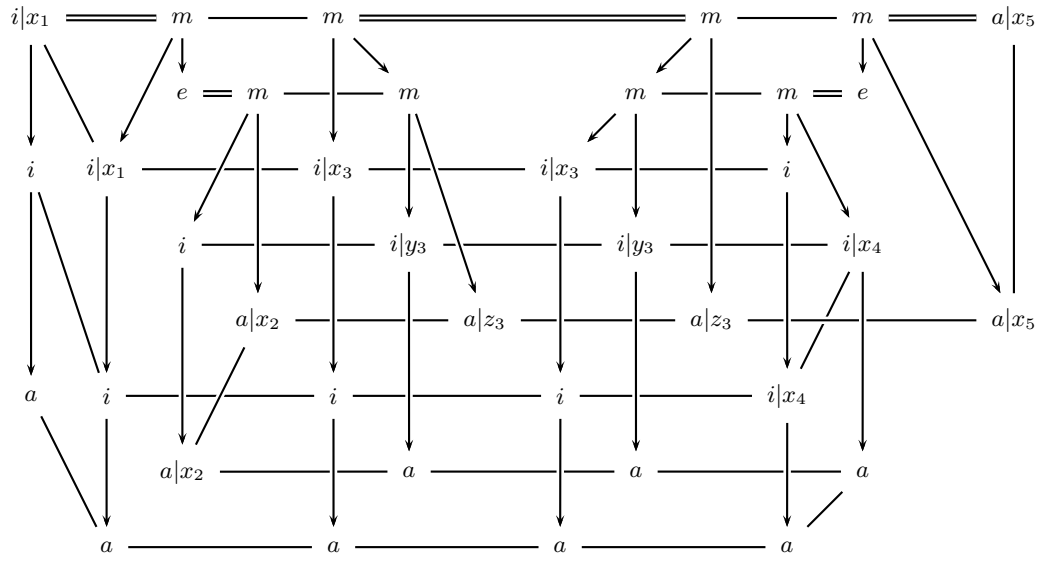

Fig. 4. Derivation for $(a^{-1})^{-1} = a$

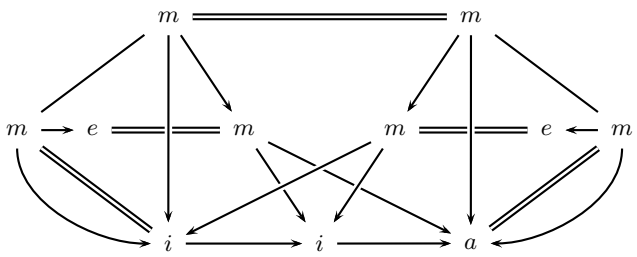

Fig. 5. Compact basic derivation for $(a^{-1})^{-1} = a$

since $\mathsf{der}^-(\emptyset) = \mathsf{der}(\emptyset) = \mathsf{ker}(\lambda k.\mathsf{term}(k, T))$ is the collection of pairs of provably identical terms, an abstraction α satisfying (8) replaces only provably identical terms by some variable x. This corresponds with the usual instantiation rule $\Gamma \vdash \varphi \;\Rightarrow\; \Gamma \vdash [x := t]\varphi$. However, this weak notion of justification would complicate the proof of the interpolation theorem, since the obvious abstraction required for the interpolant may identify non-identical terms.

A more radical condition is

$$\mathsf{ker}(\alpha) \subseteq \mathsf{der}(E) \tag{9}$$

This condition may seem appealing: α replaces provably equal terms by a variable, so it corresponds to the general instantiation rule discussed in Section 1.2. It is unsound, however. The proof of the equalities of the terms that are identified by α may depend on equations that are being affected by the same α: this may

be harmless, but only if the dependencies generated here form a wellordering. This can be violated easily, however, as the example in figure 6 illustrates.

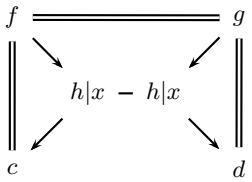

Fig. 6. Counterexample for the justification condition $\mathsf{ker}\alpha \subseteq \mathsf{cong}(\mathsf{der}(E))$

It is an example of a basic derivation $\langle T, M\rangle$ and an abstraction α with $\mathsf{ker}(\alpha) \subseteq \mathsf{cong}(\mathsf{der}(E))$, while $\mathsf{form}(E, T_\alpha) \models \mathsf{form}(\mathsf{der}(E), T)$ is false. The basic derivation represents

$$c \equiv fhc, fhc \equiv ghd, ghd \equiv d \vdash c \equiv d$$

which is correct, but when we apply the abstraction, we get

$$c \equiv fhc, fx \equiv gx, ghd \equiv d \vdash c \equiv d$$

which is not valid.

One way to impose the required wellordering on the dependencies generated by an abstraction is to extend abstract derivations with a wellordering $\prec$ on the pairs $(k, l) \in E$ that represent equations, and to require that the equality of terms that occur in the equation represented by (k, l) and that are subsumed by α, are provable using only equations $(k', l') \prec (k, l)$. This approach requires quite some overhead, however. In this paper, we decided to work with a simpler solution: use a wellordering that is already present in the notion of derivations, viz. the subnode order $\leq$ generated by arg. This leads to the original condition (2).

In [5], an earlier version of this paper, we imposed the wellordering on dependencies by requiring that $\mathsf{der}(E)\cdot <_1$ is a wellordering, where $<_1$ is the direct subnode relation. This works well in the proofs for soundness and interpolation, but it leads to a notion of derivability that is too weak. There are derivations in equational logic that cannot be brought in a form where $\mathsf{der}(E)\cdot <_1$ is well-founded: an example is the derivable sequent $a \equiv f(x, x), y \equiv hgy, f(z, ghz) \equiv b \vdash a \equiv b$ that is represented in Figure 3.

5 Correspondence with ordinary derivability

In this section, we show that the two derivability notions $\vdash$ and $\vdash_{\mathsf{abs}}$ coincide. This will be done via the completeness property of equational logic w.r.t. algebraic structures. We start with the interpretation of abstract term structures.

Definition 4. *Let a signature* SIG *and a collection* VAR *of variables be given. An interpretation in universe* U *is a mapping* $I : \mathsf{SIG} \cup \mathsf{VAR} \to \bigcup_n (U^n \to U)$ *that respects arity, i.e. if* $I(f) \in (U^{\mathsf{arity}(f)} \to U)$*. We write* $\mathsf{Int}(\mathsf{SIG}, U)$ *for the collection of all interpretations of* SIG *in* U*. Two interpretations* $I, J \in \mathsf{Int}(\mathsf{SIG}, U)$ *are similar, denoted by* $I \sim J$*, when their restriction to* SIG *is equal:*

$$I \sim J \Leftrightarrow \forall f \in \mathsf{SIG}\ I(f) = J(f)$$

Now let some term structure $T = \langle K, \mathsf{arg}, \sigma \rangle$ *be given, with* $\mathsf{rg}(\sigma) \subseteq \mathsf{SIG} \cup \mathsf{VAR}$*. A model* $\mathcal{M} = \langle U, I \rangle$ *for* T *consists of a universe* $U \neq \emptyset$ *and a signature interpretation* $I \in \mathsf{Int}(\mathsf{SIG}, U)$*. With recursion, we define* $\overline{I}_\sigma : K \to U$*:*

$$\overline{I}_\sigma(k) = I(\sigma(k))(\overline{I}^{\otimes}_\sigma(\mathsf{arg}(k)))$$

Validity for relations E *in abstract term structures* T *is defined by*

$$\begin{array}{ll} \langle U, I \rangle \models (E, T) & \textit{iff}\ \ E \subseteq \mathsf{ker}(\overline{I}_\sigma) \\ \langle U, I \rangle \models \forall(E, T) & \textit{iff}\ \ \forall J \sim I\ \langle U, J \rangle \models (E, T) \end{array}$$

For formula φ *and collection of formulas* Γ*, we define*

$$\begin{array}{l} \Gamma \models_{\mathsf{abs}} \varphi : \textit{there is an abstract term structure } T = \langle K, \mathsf{abs}, \sigma \rangle \\ \qquad \textit{and an abstraction } \alpha \textit{ with } E \subseteq K^2,\ k, l \in K \textit{ such that} \\ \qquad \mathsf{form}(E, T_\alpha) = \Gamma,\ \mathsf{form}((k, l), T) = \varphi \textit{ and} \\ \qquad \forall(E, T_\alpha) \models ((k, l), T) \end{array}$$

Observe that we have, for abstraction α:

$$\langle U, I \rangle \models (E, T_\alpha) \text{ iff } E \subseteq \mathsf{ker}(\overline{I}_{\alpha:\sigma})$$

We shall show the following correspondences between (abstract) derivability and validity:

$$\Gamma \vdash_{\mathsf{abs}} \varphi \ \Leftrightarrow\ \Gamma \vdash \varphi \ \Leftrightarrow\ \Gamma \models \varphi \ \Leftrightarrow\ \Gamma \models_{\mathsf{abs}} \varphi \tag{10}$$

The second equivalence (between $\vdash$ and $\models$) is the well known completeness of equational logic w.r.t. algebraic structures, the third equivalence (between $\models$ and $\models_{\mathsf{abs}}$) follows directly from Definition 4. We establish the first equivalence between $\vdash_{\mathsf{abs}}$ and $\vdash$ by proving firstly the soundness of $\vdash_{\mathsf{abs}}$ w.r.t. $\models_{\mathsf{abs}}$, and then the implication $\Gamma \vdash \varphi \ \Rightarrow\ \Gamma \vdash_{\mathsf{abs}} \varphi$.

Theorem 1 (Soundness). *Abstract derivations are sound: if* $\Gamma \vdash_{\mathsf{abs}} \varphi$ *then* $\Gamma \models_{\mathsf{abs}} \varphi$

Proof. Assume that $\Gamma \vdash_{\mathsf{abs}} \varphi$, i.e. there is a derivation $D = \langle T, M, \alpha \rangle$ with $E \subseteq K^2$, $\Gamma = \mathsf{term}(E, T_\alpha)$, $\varphi \in \mathsf{term}(\mathsf{der}(E), T)$ and E justifies α. By the definition of $\models_{\mathsf{abs}}$, it suffices to show, for all $\mathcal{M} = \langle U, I \rangle$:

if $\forall J \sim I\ E \subseteq \mathsf{ker}(\overline{J}_{\alpha:\sigma})$, then $\mathsf{der}(E) \subseteq \mathsf{ker}(\overline{I}_\sigma)$.

So let $\mathcal{M} = \langle U, I\rangle$ be given, and assume

$$\forall J \sim I \; E \subseteq \mathsf{ker}(\overline{J}_{\alpha:\sigma}); \tag{11}$$

we set out to show $\mathsf{der}(E) \subseteq \mathsf{ker}(\overline{I}_\sigma)$. Now $\mathsf{ker}(\overline{I}_\sigma)$ is an equivalence relation and hence closed under $\cdot^e$, and one easily verifies that $\mathsf{ker}(\overline{I}_\sigma)$ is closed under cong. So by (4) it suffices to show that $E \subseteq \mathsf{ker}(\overline{I}_\sigma)$. In order to show $E \subseteq \mathsf{ker}(\overline{I}_\sigma)$, we shall construct a J with $J \sim I$, hence $E \subseteq \mathsf{ker}(\overline{J}_{\alpha:\sigma})$ by (11); and $E \subseteq \mathsf{ker}(\overline{I}_\sigma)$ then follows if we can show

$$\forall k \in K \; \overline{I}_\sigma(k) = \overline{J}_{\alpha:\sigma}(k) \tag{12}$$

Let $\eta : \mathsf{rg}(\alpha) \to K$ be a right inverse of α, i.e. $\forall x \in \mathsf{rg}(\alpha) \; \alpha(\eta(x)) = x$. Define J by

$$\begin{array}{l} J(s) = I(s) \text{ if } s \in \mathsf{SIG} \\ J(x) = I(x) \text{ if } x \in \mathsf{VAR} - \mathsf{rg}(\alpha) \\ J(x) = \overline{I}_\sigma(\eta(x)) \text{ if } x \in \mathsf{rg}(\alpha) \end{array}$$

Then $J \sim I$, so by (11) we have $E \subseteq \mathsf{ker}(\overline{J}_{\alpha:\sigma})$. Now we define

$$\begin{array}{l} L_1 = \{k \mid \overline{I}_\sigma(k) = \overline{J}_{\alpha:\sigma}(k)\} \\ L_2 = \{k \mid \forall l((k,l) \in \mathsf{der}^-(E) \Rightarrow \overline{I}_\sigma(k) = \overline{I}_\sigma(l))\} \end{array}$$

and observe that $L_1 = K$ implies (12) and the theorem follows. We shall prove $L_1 \cap L_2 = K$ with induction over the subnode relation, where L_2 is used for induction loading. So we shall show, for all k:

$$\mathsf{arg}(k) \subseteq L_1 \cap L_2 \Rightarrow k \in L_1 \cap L_2 \tag{13}$$

This follows from

$$\mathsf{arg}(k) \subseteq L_1 \cap L_2 \Rightarrow k \in L_1 \tag{14}$$

$$k \in L_1 \;\&\; \mathsf{arg}(k) \subseteq L_2 \Rightarrow k \in L_2 \tag{15}$$

First we prove (14). Let k be given with $\mathsf{arg}(k) \subseteq L_1 \cap L_2$. We distinguish between $k \in \mathsf{dom}(\alpha)$ and $k \notin \mathsf{dom}(\alpha)$, and in both cases we prove $\overline{J}_{\alpha:\sigma}(k) = \overline{I}_\sigma(k)$. If $k \in \mathsf{dom}(\alpha)$, then

$$\begin{array}{ll} & \mathsf{arg}(k) \subseteq L_2 \\ \Rightarrow & \quad \{\text{definition of } L_2\} \\ & \forall l((k,l) \in \mathsf{cong}(\mathsf{der}^-(E)) \Rightarrow \overline{I}_\sigma(k) = \overline{I}_\sigma(l)) \\ \Rightarrow & \quad \{\alpha \text{ is justified, so } \mathsf{ker}(\alpha) \subseteq \mathsf{cong}(\mathsf{der}^-(E))\} \\ & \forall l((k,l) \in \mathsf{ker}(\alpha) \Rightarrow \overline{I}_\sigma(k) = \overline{I}_\sigma(l)) \\ \Rightarrow & \quad \{\alpha(\eta(\alpha(k))) = \alpha(k), \text{ so } (\eta(\alpha(k)), k) \in \mathsf{ker}(\alpha)\} \\ & \overline{I}_\sigma(\eta(\alpha(k))) = \overline{I}_\sigma(k) \\ \Leftrightarrow & \quad \{\text{definition of } J\} \\ & J(\alpha(k)) = \overline{I}_\sigma(k) \\ \Leftrightarrow & \quad \{k \in \mathsf{dom}(\alpha)\} \\ & \overline{J}_{\alpha:\sigma}(k) = \overline{I}_\sigma(k) \end{array}$$

If $k \notin \mathsf{dom}(\alpha)$, then

$$
\begin{array}{ll}
 & \mathsf{arg}(k) \subseteq L_1 \\
\Rightarrow & \quad \{\text{definition of } L_1\} \\
 & I(\sigma(k))\overline{J}^{\otimes}_{\alpha:\sigma}(\mathsf{arg}(k)) = \overline{I}_\sigma(k) \\
\Leftrightarrow & \quad \{I \sim J\} \\
 & J(\sigma(k))\overline{J}^{\otimes}_{\alpha:\sigma}(\mathsf{arg}(k)) = \overline{I}_\sigma(k) \\
\Leftrightarrow & \quad \{k \notin \mathsf{dom}(\alpha)\} \\
 & \overline{J}_{\alpha:\sigma}(k) = \overline{I}_\sigma(k)
\end{array}
$$

This ends the proof of (14). We turn to (15), and assume $k \in L_1$ and $\mathsf{arg}(k) \subseteq L_2$. Now

$$
\begin{array}{ll}
 & (k,l) \in \mathsf{der}^-(E) \\
\Rightarrow & \quad \{\mathsf{der}^-(E) = E \cup \mathsf{cong}(\mathsf{der}^-(E))\} \\
 & (k,l) \in E \vee (k,l) \in \mathsf{cong}(\mathsf{der}^-(E)) \\
\Rightarrow & \quad \{E \subseteq \mathsf{ker}(\overline{J}_{\alpha:\sigma}) \text{ by (11), and } k \in L_1\} \\
 & \overline{I}_\sigma(k) = \overline{I}_\sigma(l) \vee (k,l) \in \mathsf{cong}(\mathsf{der}^-(E)) \\
\Rightarrow & \quad \{\mathsf{arg}(k) \subseteq L_2\} \\
 & \overline{I}_\sigma(k) = \overline{I}_\sigma(l) \vee (\sigma(k) = \sigma(l) \;\&\; \overline{I}^{\otimes}_\sigma(\mathsf{arg}(k) = \overline{I}^{\otimes}_\sigma(\mathsf{arg}(l))) \\
\Rightarrow & \quad \{\} \\
 & \overline{I}_\sigma(k) = \overline{I}_\sigma(l)
\end{array}
$$

So $k \in L_2$ and we have proved (15). This ends the proof.

Theorem 2 (derivability implies abstract derivability). *If $\Gamma \vdash \varphi$, then $\Gamma \vdash_{\mathsf{abs}} \varphi$.*

Proof. Assume that $\Gamma \vdash \varphi$. We must show that there is a derivation $D = \langle T, M, \alpha\rangle$ and $E \subseteq K^2$ with

$$E \text{ justifies } \alpha,\ \mathsf{form}(E, T_\alpha) = \Gamma \text{ and } \varphi \in \mathsf{form}(\mathsf{der}(E), T)$$

This is done via induction over a derivation of $\Gamma \vdash_{\mathsf{EQL}} \varphi$. We apply some induction loading and claim that also $M = \mathsf{ker}(\sigma)$ and $\mathsf{ker}(\alpha) \subseteq \mathsf{der}(\emptyset)$. So justification is always OK; since M equals $\mathsf{ker}(\sigma)$, we shall drop M from the notation for derivations in this proof, and write $D = \langle T, \alpha\rangle$. When we consider two or more term structures in the same context, we assume that they are disjoint; similarly, we assume that different abstractions in the same context have disjoint ranges.

The union of two term structures $T_1 = \langle K_1, \mathsf{arg}_1, \sigma_1\rangle$, $T_2 = \langle K_2, \mathsf{arg}_2, \sigma_2\rangle$ is defined as $T = T_1 \cup T_2 = \langle K_1 \cup K_2, \mathsf{arg}_1 \cup \mathsf{arg}_2, \sigma_1 \cup \sigma_2\rangle$; since we assume that $K_1 \cap K_2 = \emptyset$, we have that $\mathsf{arg}_1 \cup \mathsf{arg}_2$ and $\sigma_1 \cup \sigma_2$ are functions and T is a term structure. Similarly, the union of two derivations $D_1 = \langle T_1, \alpha_1\rangle$, $D_2 = \langle T_2, \alpha_2\rangle$ is defined as $D = D_1 \cup D_2 = \langle T_1 \cup T_2, \alpha_1 \cup \alpha_2\rangle$. If cong, cong_1, cong_2 are the congruence operators associated with D, D_1 and D_2, respectively, then we have $\mathsf{cong}_1 \sqcup \mathsf{cong}_2 \sqsubseteq \mathsf{cong}$ because of monotonicity; similarly for derivability. Similarly for unions of more than two term structures or derivations.

Finally, before we start with the proof of the induction steps, we observe that, for every collection of formulas Γ, there is a term structure T with domain K and a relation $E \subseteq K^2$ with $\mathsf{form}(E, T) = \Gamma$.

assumption: $(s \equiv t) \in \Gamma$, so $\Gamma \vdash s \equiv t$.
Put $D = \langle T, \emptyset \rangle$, where T be a term structure with $E \subseteq K^2$ such that $\mathsf{form}(E, T) = \Gamma$.

reflexivity: $\Gamma \vdash t \equiv t$.
Put $D = \langle T, \emptyset \rangle$, where T be a term structure with $E \subseteq K^2$ such that $\mathsf{form}(E, T) = \Gamma$, and with $k \in T$ such that $\mathsf{term}(k) = t$.

symmetry: $\Gamma \vdash s \equiv t \Rightarrow \Gamma \vdash t \equiv s$.
Follows from the induction hypothesis and the symmetry of $\mathsf{der}(E)$.

transitivity: $\Gamma \vdash r \equiv s \;\&\; \Gamma \vdash s \equiv t \Rightarrow \Gamma \vdash r \equiv t$.
By induction hypothesis, we have derivations $D_1 = \langle T_1, \alpha_1 \rangle$, $D_2 = \langle T_2, \alpha_2 \rangle$, $E_1 \subseteq K_1^2$, $E_2 \subseteq K_2^2$, $k_1, l_1 \in K_1$, $k_2, l_2 \in K_2$ with $\mathsf{form}(E_1, T_{1,\alpha_1}) = \mathsf{form}(E_2, T_{2,\alpha_2}) = \Gamma$, $(l_1, k_1) \in \mathsf{der}_1(E_1)$, $(k_2, l_2) \in \mathsf{der}_2(E_2)$, $\mathsf{term}(l_1, T_1) = r$, $\mathsf{term}(k_1, T_1) = \mathsf{term}(k_2, T_2) = s$, $\mathsf{term}(l_2, T_2) = t$.

Define $D = \langle T_1 \cup T_2, \alpha_1 \cup \alpha_2 \rangle$ and $E = E_1 \cup E_2$, then $\Gamma = \mathsf{der}(E)$ and $(l_1, k_1), (k_2, l_2) \in \mathsf{der}(E_1 \cup E_2)$. Since also $(k_1, k_2) \in \mathsf{der}(\emptyset) \subseteq \mathsf{der}(E_1 \cup E_2)$, we have (by transitivity of $\mathsf{der}(E_1 \cup E_2)$) that $(l_1, l_2) \in \mathsf{der}(E_1 \cup E_2)$, so $(r \equiv t) \in \mathsf{form}(\mathsf{der}(E_1 \cup E_2), T_1 \cup T_2)$.

congruence: $\Gamma \vdash s_i \equiv t_i$ for i with $0 \leq i < n \;\Rightarrow\; \Gamma \vdash f(s_0, \ldots, s_{n-1}) \equiv f(t_0, \ldots, t_{n-1})$
For $i = 0, \ldots, n-1$ we have, by induction hypothesis, derivations $D_i = \langle T_i, \alpha_i \rangle$ with $E_i \subseteq K_i^2$, $\mathsf{form}(E_i, T_{i,\alpha_i}) = \Gamma$ and $k_i, l_i \in K_i$ with $(k_i, l_i) \in \mathsf{der}_i(E_i)$, $\mathsf{term}(k_i, T_i) = s_i$, $\mathsf{term}(l_i, T_i) = t_i$.

Let k, l be fresh nodes, i.e. not in $K_0, \ldots, K_{n-1}$. Now define $D = \langle K, \mathsf{arg}, \sigma, \alpha \rangle$ by

$$\begin{aligned} K &= K_0 \cup \cdots \cup K_{n-1} \cup \{k, l\} \\ \mathsf{arg} &= \mathsf{arg}_0 \cup \cdots \cup \mathsf{arg}_{n-1} \cup \{(k, (k_0, \ldots, k_{n-1})), (l, (l_0, \ldots, l_{n-1}))\} \\ \sigma &= \sigma_0 \cup \cdots \cup \sigma_{n-1} \cup \{(k, f), (l, f)\} \\ \alpha &= \alpha_0 \cup \cdots \cup \alpha_{n-1} \end{aligned}$$

and define $E = E_0 \cup \cdots \cup E_{n-1}$. We observe that $\mathsf{form}(E, T_\alpha) = \mathsf{form}(E_0, T_{0,\alpha_0}) \cup \cdots \cup \mathsf{form}_($ $E_{n-1}, T_{n-1,\alpha_{n-1}}) \subseteq \Gamma$. Also, $(f(s_0, \ldots, s_{n-1}) \equiv f(t_0, \ldots, t_{n-1})) = \mathsf{form}((k, l), T) \in \mathsf{form}(\mathsf{der}(E), T)$, since $(k, l) \in \mathsf{cong}(\{(k_0, l_0), \ldots, (k_{n-1}, l_{n-1})\}) \subseteq \mathsf{cong}(\mathsf{der}(E)) \subseteq \mathsf{der}(E)$.

instantiation: $\Gamma \vdash r \equiv s \;\Rightarrow\; \Gamma \vdash [x := t]r \equiv [x := t]s$
By induction hypothesis, we have a derivation $D_1 = \langle K_1, \mathsf{arg}_1, \sigma_1, \alpha_1 \rangle$ with $E_1 \subseteq K_1^2$, $k', l' \in K_1$ such that $\mathsf{form}(E_1, T_{1,\alpha_1}) = \Gamma$, $(k', l') \in \mathsf{der}_1(E_1)$, $\mathsf{term}(k', T_1) = r$, $\mathsf{term}(l', T_1) = s$. Let $t = f(t_0, \ldots, t_{n-1})$ (so n is the arity of f). Let $T_2 = \langle K_2, \mathsf{arg}_2, \sigma_2 \rangle$ be a term structure with $k_0, \ldots k_{n-1} \in K_2$, $\mathsf{term}(k_i, T_2) = t_i$ for i

with $0 \leq i < n$. Define $T = \langle K_1 \cup K_2, \mathsf{arg}, \sigma \rangle$ where arg, σ are defined by

$$\begin{array}{rll} \mathsf{arg}(k) & = (k_0, \ldots, k_{n-1}) & \text{if } k \in K_1 \text{ and } \sigma_1(k) = x \\ & = \mathsf{arg}_1(k) & \text{if } k \in K_1 \text{ and } \sigma_1(k) \neq x \\ & = \mathsf{arg}_2(k) & \text{if } k \in K_2 \\ \sigma(k) & = f & \text{if } k \in K_1 \text{ and } \sigma_1(k) = x \\ & = \sigma_1(k) & \text{if } k \in K_1 \text{ and } \sigma_1(k) \neq x \\ & = \sigma_2(k) & \text{if } k \in K_2 \end{array}$$

and define α by

$$\begin{array}{rll} \mathsf{dom}(\alpha) & = \mathsf{dom}(\alpha_1) \cup \{k \in K_1 \mid \sigma_1(k) = x\} & \\ \alpha(k) & = \alpha_1(k) & \text{if } k \in \mathsf{dom}(\alpha_1) \\ & = x & \text{if } k \notin \mathsf{dom}(\alpha_1) \text{ and } \sigma_1(k) = x \end{array}$$

$\mathsf{form}(E_1, T_\alpha) = \mathsf{form}(E_1, T_{1,\alpha_1})$ is easily verified, so we have $\mathsf{form}(E_1, T_\alpha) = \Gamma$. $([x := t]r \equiv [x := t]s) \in \mathsf{form}(\mathsf{der}(E_1), T)$ follows from $\mathsf{term}(k', T) = [x := t]r$, $\mathsf{term}(l', T) = [x := t]s$ and the fact that $\mathsf{der}_1(E_1) \subseteq \mathsf{der}(E_1) \cap K_1^2$. This inclusion is proved using the minimality property for der_1, using

$$\mathsf{cong}_1(\mathsf{der}(E_1) \cap K_1^2) \cap K_1^2 \subseteq \mathsf{cong}(\mathsf{der}(E_1) \cap K_1^2)$$

To see that this holds, observe that cong and cong_1 only diverge for (k, l) with $\sigma_1(k) = \sigma_1(l) = x$, and then $\mathsf{arg}(k) = \mathsf{arg}(l) = (k_0, \ldots, k_{n-1})$, so indeed $(\mathsf{arg}(k), \mathsf{arg}(l)) \in (\mathsf{der}(E_1) \cap K_1^2)^\otimes$.

6 Some properties of derivations

In this section, we prove some properties of abstract derivations. They provide insight in the nature of derivations, and they will be applied in the proof of the Interpolation theorem given in the next section. First we define the restriction of a term structure and a (basic) derivation.

Definition 5 (Restriction). *Let $T = \langle K, \mathsf{arg}, \sigma \rangle$ be a term structure, and $D = \langle T, M, \alpha \rangle$ a derivation. Let $L \subseteq K$ be* arg*-closed. Then the* restriction *T_L of T, and D_L of D to L are defined by*

$$\begin{array}{l} T_L = \langle L, \mathsf{arg} \restriction L, \sigma \restriction L \rangle \\ D_L = \langle T_L, M \cap L^2, \alpha \restriction L \rangle \end{array}$$

Lemma 1 (intersection and restriction). *If $L \subseteq K$ is* arg*-closed, then*

$$\mathsf{der}_L^-(E \cap L^2) = \mathsf{der}^-(E \cap L^2) \cap L^2 = \mathsf{der}^-(E) \cap L^2$$

Proof. The first inclusion follows from $\mathsf{der}_L^- \sqsubseteq \mathsf{der}^-$, the second from monotonicity of der^-, so it suffices to show $\mathsf{der}^-(E) \cap L^2 \subseteq \mathsf{der}_L^-(E \cap L^2)$. We prove this using the property

$$L \text{ } \mathsf{arg}\text{-closed } \Rightarrow \mathsf{cong}(R) \cap L^2 = \mathsf{cong}_L(R \cap L^2) \subseteq \mathsf{cong}(R \cap L^2) \tag{16}$$

which is verified easily. Now

$$
\begin{array}{ll}
 & \mathsf{der}^-(E) \cap L^2 \subseteq \mathsf{der}^-_L(E \cap L^2) \\
\Leftrightarrow & \quad \{\text{elementary set theory}\} \\
 & \mathsf{der}^-(E) \subseteq \mathsf{der}^-_L(E \cap L^2) \cup (K^2 - L^2) \\
\Leftarrow & \quad \{\text{minimality property of } \mathsf{der}^-\} \\
 & E \cup \mathsf{cong}(\mathsf{der}^-_L(E \cap L^2) \cup (K^2 - L^2)) \subseteq \mathsf{der}^-_L(E \cap L^2) \cup (K^2 - L^2) \\
\Leftrightarrow & \quad \{\text{elementary set theory}\} \\
 & (E \cap L^2) \cup (\mathsf{cong}(\mathsf{der}^-_L(E \cap L^2) \cup (K^2 - L^2)) \cap L^2) \subseteq \mathsf{der}^-_L(E \cap L^2) \\
\Leftrightarrow & \quad \{(16)\} \\
 & (E \cap L^2) \cup \mathsf{cong}(\mathsf{der}^-_L(E \cap L^2)) \subseteq \mathsf{der}^-_L(E \cap L^2) \\
\Leftrightarrow & \quad \{\text{defining property of } \mathsf{der}^-_L\} \\
 & \mathsf{true}
\end{array}
$$

The next Lemma indicates how we can reduce der to der^- without affecting the signature.

Lemma 2 (reduction of der to der^-). *Let $B = \langle T, M, \alpha \rangle$ be a derivation, and $E \subseteq K^2$. Then there is a relation $E' \subseteq \mathsf{der}(E)$ satisfying $\mathsf{der}(E) = \mathsf{der}^-(E')$ and $\mathsf{sig}(E') \subseteq \mathsf{sig}(E)$.*

Proof. Let $E' = \mathsf{der}(E) \cap K^2_{\overline{\Sigma}}$ with $\Sigma := \mathsf{sig}(E)$. $\mathsf{sig}(E') \subseteq \mathsf{sig}(E)$ is evident, and $\mathsf{der}^-(E') \subseteq \mathsf{der}(E') \subseteq \mathsf{der}(\mathsf{der}(E)) = \mathsf{der}(E)$. So we only have to show

$$\mathsf{der}(E) \subseteq \mathsf{der}^-(E') \tag{17}$$

We observe that $E \subseteq \mathsf{der}(E) \cap K_{\overline{\Sigma}} \subseteq \mathsf{der}^-(\mathsf{der}(E) \cap K_{\overline{\Sigma}}) = \mathsf{der}^-(E')$ and that $\mathsf{der}^-(E')$ is closed under cong. So for (17) it suffices to show that $\mathsf{der}^-(E')$ is an equivalence relation. Reflexivity of $\mathsf{der}^-(E')$ follows from $\forall k \in K\ (k,k) \in \mathsf{der}^-(\emptyset)$, which is proved straightforwardly with subnode induction. Symmetry of $\mathsf{der}^-(E')$ follows from the symmetry of E' and the fact that cong preserves symmetry. Transitivity is more work. First we prove

$$\mathsf{der}^-(E') \subseteq E' \cup (K - K_{\overline{\Sigma}})^2 \tag{18}$$

This is done as follows, using the property that

$$\mathsf{cong} \text{ preserves } K_{\overline{\Sigma}}\text{-respect} \tag{19}$$

(i.e. if $R \subseteq K^2_{\overline{\Sigma}} \cup (K - K_{\overline{\Sigma}})^2$, then $\mathsf{cong}(R) \subseteq K^2_{\overline{\Sigma}} \cup (K - K_{\overline{\Sigma}})^2$), which is verified easily. Now

$$
\begin{array}{ll}
 & \mathsf{der}^-(E') \subseteq E' \cup (K - K_{\overline{\Sigma}})^2 \\
\Leftrightarrow & \quad \{\text{minimality property of } \mathsf{der}^-\} \\
 & E' \cup \mathsf{cong}(E' \cup (K - K_{\overline{\Sigma}})^2) \subseteq E' \cup (K - K_{\overline{\Sigma}})^2 \\
\Leftarrow & \quad \{\text{elementary set theory}\} \\
 & \mathsf{cong}(E' \cup (K - K_{\overline{\Sigma}})^2) - (K - K_{\overline{\Sigma}})^2 \subseteq E' \\
\Leftarrow & \quad \{(19)\} \\
 & \mathsf{cong}(E' \cup (K - K_{\overline{\Sigma}})^2) \cap K^2_{\overline{\Sigma}} \subseteq E' \\
\Leftarrow & \quad \{\text{definition of } E' \text{ and } (16)\}
\end{array}
$$

$$
\begin{array}{ll}
 & \mathsf{cong}(\mathsf{der}(E) \cap K^2_{\overline{\Sigma}}) \subseteq \mathsf{der}(E) \\
\Leftarrow & \quad \{\text{property of } \mathsf{der}(E)\} \\
 & \mathsf{true}
\end{array}
$$

Now we claim

$$\forall l \in K \forall km \in K((k,l),(l,m) \in \mathsf{der}^-(E') \Rightarrow (k,m) \in \mathsf{der}^-(E')) \qquad (20)$$

It is clear that this implies that $\mathsf{der}^-(E')$ is transitive. We prove (20) with induction over the subnode ordering $\leq$. Assume $(k,l),(l,m) \in \mathsf{der}^-(E)$ and recall that $\mathsf{der}^-(E') = E' \cup \mathsf{cong}(\mathsf{der}^-(E'))$. We distinguish two cases.

1. $(k,l) \in E'$ or $(l,m) \in E'$. Then $\mathsf{sig}(l) \subseteq \Sigma$, so $(k,l),(l,m) \in \mathsf{der}^-(E') - (K - K_{\overline{\Sigma}})^2$. By (18), we get $(k,l),(l,m) \in E'$, so $(k,l) \in E' \subseteq \mathsf{der}^-(E')$ since E' is an equivalence relation.
2. $(k,l),(l,m) \in \mathsf{cong}(\mathsf{der}^-(E'))$, so for some n we have $\mathsf{lth}(k) = \mathsf{lth}(l) = \mathsf{lth}(m) = n$ and $\forall i < n\ (\mathsf{arg}_i(k),\mathsf{arg}_i(l)),(\mathsf{arg}_i(l),\mathsf{arg}_i(m)) \in \mathsf{der}^-(E')$. By the induction hypothesis, we have (since $\mathsf{arg}_i(l) < l$ for $0 \leq i < n$) $\forall i < n (\mathsf{arg}_i(k),\mathsf{arg}_i(m)) \in \mathsf{der}^-(E')$, so $(k,m) \in \mathsf{cong}(\mathsf{der}^-(E')) \subseteq \mathsf{der}^-(E')$.

This ends the proof.

In Definition 2, we defined application of an abstraction to a term structure. We extend this now to basic derivations.

Definition 6 (applicability). *Let $D = \langle T, M\rangle$ be a basic derivation, and α an abstraction of T. We call α* applicable *to D if*

$$M \text{ respects } \alpha \text{ (i.e. } M \subseteq \mathsf{ker}(\alpha) \cup (K - \mathsf{dom}(\alpha))^2\text{).}$$

The result of applying α to D is the basic derivation $D_\alpha = \langle T_\alpha, M\rangle$. The associated congruence and derivability operations are denoted cong_α, der_α and der^-_α.

Lemma 3 (applicability). *If abstraction α is applicable to basic derivation D, then D_α is a derivation. Moreover, $\mathsf{cong} \sqsubseteq \mathsf{cong}_\alpha$ and hence $\mathsf{der} \sqsubseteq \mathsf{der}_\alpha$, $\mathsf{der}^- \sqsubseteq \mathsf{der}^-_\alpha$.*

Proof. It is easy to see that if α is applicable to B, then B_α is a derivation: for $M \subseteq \mathsf{ker}(\alpha) \cup (K - \mathsf{dom}(\alpha))^2$ and $M \subseteq \mathsf{ker}(\sigma)$ imply that $M \subseteq \mathsf{ker}(\alpha : \sigma)$.

To show that $\mathsf{cong} \sqsubseteq \mathsf{cong}_\alpha$, we argue as follows. Let $(k,l) \in \mathsf{cong}(R)$, so $(k,l) \in M$ and $(\mathsf{arg}(k),\mathsf{arg}(l)) \in R^\otimes$. We want

$$(\mathsf{arg}_\alpha(k),\mathsf{arg}_\alpha(l)) \in R^\otimes \qquad (21)$$

M respects α and hence $\mathsf{dom}(\alpha)$, so either $k,l \in \mathsf{dom}(\alpha)$ or $k,l \notin \mathsf{dom}(\alpha)$. If $k,l \in \mathsf{dom}(\alpha)$ then $\mathsf{arg}_\alpha(k) = \mathsf{arg}_\alpha(l) = ()$, so (21) holds. If $k,l \notin \mathsf{dom}(\alpha)$ then $\mathsf{arg}_\alpha(k) = \mathsf{arg}(k)$, $\mathsf{arg}_\alpha(l) = \mathsf{arg}(l)$ and again we have (21).

So $\forall R \subseteq K^2\ \mathsf{cong}(R) \subseteq \mathsf{cong}_\alpha(R)$, i.e. $\mathsf{cong} \sqsubseteq \mathsf{cong}_\alpha$. $\mathsf{der} \sqsubseteq \mathsf{der}_\alpha$, $\mathsf{der}^- \sqsubseteq \mathsf{der}^-_\alpha$ follow from this by monotonicity.

Finally, we introduce the notion of parsimony.

Definition 7 (Parsimony). *Derivation* $D = \langle T, M, \alpha\rangle$ *is* parsimonious *with respect to* $E \subseteq K^2$ *(E-parsimonious) if* $M \subseteq \mathsf{lift}(\mathsf{der}(E))$*;.*

The idea is that, in an E-parsimonious derivation, the matching relation M is minimal: all pairs $(k, l) \in M$ are needed to establish $\mathsf{der}(E)$. This is made explicit in the first part of the next lemma. The second part shows that we may always assume that a derivation is parsimonious: for if it is not, we can make it parsimonious without changing $\mathsf{der}(E)$.

Lemma 4 (Parsimony). *Let* $D = \langle T, M, \alpha\rangle$ *be a derivation.*

1. *If* D *is* E*-parsimonious, then* $\mathsf{der}(E) = (E \cup M)^e$*.*
2. *There is an* E*-parsimonious basic derivation* $D' = \langle T, M'\alpha\rangle$ *with derivation operator* der' *such that* $\mathsf{der}(E) = \mathsf{der}'(E)$*.*

Proof. 1. If D is E-parsimonious, then $M \subseteq \mathsf{lift}(\mathsf{der}(E))$, so $M = M \cap \mathsf{lift}(\mathsf{der}(E)) = \mathsf{cong}(\mathsf{der}(E))$, hence $(E \cup M)^e = (E \cup \mathsf{cong}(\mathsf{der}(E)))^e = \mathsf{der}(E)$.

2. Define $M' = \mathsf{cong}(\mathsf{der}(E))$. Then $M' \subseteq M$, so by monotonicity we have $\mathsf{der}'(E) \subseteq \mathsf{der}(E)$. The other inclusion $\mathsf{der}(E) \subseteq \mathsf{der}'(E)$ follows via the minimality property of der, provided $\mathsf{der}'(E)$ is closed under cong. Since $\mathsf{der}'(E)$ is closed under cong', it suffices to show $\mathsf{cong}(\mathsf{der}'(E)) \subseteq \mathsf{cong}'(\mathsf{der}'(E))$, i.e.

$$M \cap \mathsf{lift}(\mathsf{der}'(E)) \subseteq (M \cap \mathsf{lift}(\mathsf{der}(E))) \cap \mathsf{lift}(\mathsf{der}'(E))$$

and this follows via monotonicity from $\mathsf{der}'(E) \subseteq \mathsf{der}(E)$.

7 Interpolation

The following version of interpolation holds for equational logic.

Theorem 3. *If* $\Gamma \vdash \varphi$*, then there is a collection* Θ *of formulas such that*

1. $\Gamma \vdash \Theta$ *(i.e.* $\Gamma \vdash \theta$ *for every* $\theta \in \Theta$*),*
2. $\Theta \vdash^- \varphi$*, and*
3. $\mathsf{sig}(\Theta) \subseteq \mathsf{sig}(\Gamma) \cap \mathsf{sig}(\varphi)$*.*

Proof. Assume $\Gamma \vdash \varphi$, so by Theorem 2 there is a derivation $D = \langle T, M, \alpha\rangle$ with $E \subseteq K^2$, $\mathsf{ker}(\alpha) \subseteq \mathsf{cong}(\mathsf{der}^-(E))$, $\mathsf{form}(E, T_\alpha) = \Gamma$ and $(k, l) \in \mathsf{der}(E)$ with $\mathsf{form}((k, l), T) = \varphi$. Define

$$\Sigma := \mathsf{sig}(E),\ \Sigma^- := \mathsf{sig}_\alpha(E),\ \Pi := \mathsf{sig}(\{k, l\}).$$

So $\mathsf{sig}(\Gamma) = \Sigma^- \subseteq \Sigma$ and $\mathsf{sig}(\varphi) = \Pi$. In a first attempt to find Θ, we consider $\Theta_0 = \mathsf{form}(I)$, where

$$I = \mathsf{der}(E) \cap K^2_{\Sigma \cap \Pi}$$

Then $I \subseteq \mathsf{der}(E)$, so $\Gamma \vdash_{\mathsf{abs}} \Theta_0$. Using Lemma 2 and Lemma 1, we see that

$$\mathsf{der}^-(I) \cap K^2_\Pi = \mathsf{der}(E) \cap K^2_\Pi \tag{22}$$

So $(k,l) \in \mathsf{der}^-(I)$ and we have $\Theta_0 \vdash_{\mathsf{abs}} \varphi$. However, $\mathsf{sig}(\Theta_0) = \mathsf{sig}(I) = \Sigma \cap \Pi$ and this is in general not contained in $\mathsf{sig}(\Gamma) \cap \mathsf{sig}(\varphi) = \Sigma^- \cap \Pi$. To get rid of the signature elements in $\Sigma - \Sigma^- \cap \Pi$ that occur in Θ_0, we introduce an abstraction β with $\mathsf{dom}(\beta) = K_{\Sigma - \Sigma^- \cap \Pi}$ and define $\Theta = \mathsf{form}_\beta(I)$. Now $\mathsf{sig}(\Theta) = \mathsf{sig}_\beta(I) = (\Sigma \cap \Pi) - (\Sigma - \Sigma^- \cap \Pi) = \Sigma^- \cap \Pi$, and we realized the third part of the Theorem.

We turn to the first part, $\Gamma \vdash \Theta$. To realize this, we want β to be applicable to D, so we define

$$\mathsf{ker}(\beta) = M \cap K^2_{\Sigma - \Sigma^- \cap \Pi} \tag{23}$$

Since VAR is infinite, such a β can be found. By Lemma 3, we have $\mathsf{cong} \sqsubseteq \mathsf{cong}_\beta$, $\mathsf{der}^- \sqsubseteq \mathsf{der}^-_\beta$, so $\mathsf{ker}(\alpha) \subseteq \mathsf{cong}_\beta(\mathsf{der}^-_\beta(E))$. Hence E_1 justifies α in $D_\beta = \langle T_\beta, M, \alpha \rangle$, and we have $\mathsf{form}(E, T_{\alpha:\beta}) \vdash_{\mathsf{EQL}} \mathsf{form}(I, T_\beta) = \Theta$. This leads to $\Gamma \vdash_{\mathsf{abs}} \Theta$ provided $\mathsf{form}(E, T_{\alpha:\beta}) = \mathsf{form}(E, T_\alpha)$. This last property follows from

$$\forall k \in \mathsf{car}(E)\ \mathsf{term}(k, T_{\alpha:\beta}) = \mathsf{term}(k, T_\alpha)$$

We argue towards contradiction, so assume $k' \in \mathsf{car}(E)$ be a node with $\mathsf{term}(k', T_{\alpha:\beta}) \neq \mathsf{term}(k', T_\alpha)$. Then there is a path in $K - \mathsf{dom}(\alpha)$ from k' to some $l' \in \mathsf{dom}(\beta)$. But then $\sigma(k') \in \mathsf{sig}(E, T_\alpha) = \Sigma^-$ and $\sigma(k') \in \mathsf{dom}(\beta) = \Sigma - \Sigma^- \cap \Pi$ and we have a contradiction. So we have proved $\Gamma \vdash_{\mathsf{abs}} \Theta$, and with (10) we conclude $\Gamma \vdash \Theta$.

Finally we show the second part, $\Theta \vdash \varphi$. For the first step, we assume (thanks to Lemma 4) that B is parsimonious, so $M = \mathsf{cong}(\mathsf{der}(E))$. This leaves us to verify that $\mathsf{cong}(\mathsf{der}(E)) \subseteq \mathsf{cong}(\mathsf{der}^-(I))$. This does not hold in D itself, but by (22) we can close the gap by restriction to $K_{\overline{\Pi}}$. So we define $D' = \langle T_{K_{\overline{\Pi}}}, M \cap K^2_{\overline{\Pi}}, \beta' \rangle$, where $\beta' = \beta \restriction K_{\overline{\Pi}}$. Let cong', der'^-, term' be the operators associated with D', then by Lemma 1 we have indeed $\mathsf{cong}(\mathsf{der}^-(E')) \cap K^2_{\overline{\Pi}} = \mathsf{cong}'(\mathsf{der}'^-(E' \cap K^2_{\overline{\Pi}})) = \mathsf{cong}'(\mathsf{der}'^-(I))$. So I justifies β' in D' and we have $(k,l) \in \mathsf{der}'^-(I)$. Hence $\Theta \vdash_{\mathsf{abs}} \varphi$, and $\Theta \vdash \varphi$ via (10).

We illustrate this with the abstract derivation of the sequent

$$a \equiv b,\ b \equiv c,\ g_1 x \equiv g_2 x,\ g_2 h y \equiv d \ \vdash\ f g_1 h k a \equiv f d$$

given in Figure 2. For reasons of perspicuity, we omit trivial equalities of the form $t \equiv t$, and we also omit $t \equiv s$ if $s \equiv t$ is present. Now

$$\begin{aligned} \Sigma &= \{a, b, c, d, g_1, g_2, h, k\} \\ \Sigma^- &= \{a, b, c, d, g_1, g_2, h\} = \Sigma - \{k\} \\ \Pi &= \{a, d, f, g_1, h, k\} \end{aligned}$$

and

$$\begin{aligned} \mathsf{form}(\mathsf{der}(E), T) = \{&a \equiv b, b \equiv c, a \equiv c, ka \equiv kc, hka \equiv hkc, \\ &g_1 hka \equiv g_2 hkc, g_2 hka \equiv d, g_1 hka \equiv d, f g_1 hka \equiv f d\} \end{aligned}$$

so we have $\Theta_0 = \{g_1 hka \equiv d\}$, and $\Theta = \{g_1 hy \equiv d\}$ is the interpolant.

8 Concluding remarks

We presented abstract derivations for equational logic and established some interesting properties which we applied in a proof of the Interpolation theorem. The proof is constructive in that the interpolant is given explicitly, using global operations on the relations that represent equations. We see this as an advantage over other proofs for the same theorem, which are either not constructive or proceed via incremental proof transformations.

We finish with some ideas for further research. In [15], Van Oostrom shows that, somewhat surprisingly, interpolation does not hold in the logic of partial order with monotonic functions (called rewrite logic in [15]). It is defined as equational logic with $\leq$ instead of $\equiv$ and without the symmetry rule. As a counterexample to interpolation, we have the following provable sequent (a slight simplification of the example given by Van Oostrom):

$$a \leq c,\ b \leq c,\ f(x,x) \leq d \ \vdash\ f(ha,hb) \leq d$$

The only possible interpolant would be something like $\exists x(ha \leq x \wedge hb \leq x)$, but this is not expressible in the logic. However, partial order is definable in equational logic in the presence of a binary operator $\sqcap$ which behaves like a supremum operator, by $s \leq t$ iff $s \sqcap t \equiv s$. We conjecture that a partial interpolation result can be obtained for partial order logic via an embedding of the logic in equational logic with $\sqcap$.

We chose equational logic for its general nature, in the sense that equality is a fundamental notion, and many logics can be naturally embedded in equational logic. But it remains to be investigated what happens with derivations in these embeddings. It may also be interesting to extend the notion of abstract terms and to allow for bags or sets instead of sequences as the datatype for the immediate subterms of a compound term: by doing so, properties like commutativity, associativity and idempotency can be 'hardwired' in the abstract terms. Another idea is to replace term structures by sequent structures (after all, sequents can be considered as a kind of generalized terms) so as to investigate sequent-based derivation systems. The representation of quantifiers is an open question.

Furthermore, there is conditional equational logic, where implications $(s_0 \equiv t_0 \wedge \ldots s_{n-1} \equiv t_{n-1}) \rightarrow s \equiv t$ are allowed: Rodenburg proved interpolation in [10] via an adaptation of the algebraic proof for equational logic in [11], but a prooftheoretic proof has not been given yet. Another direction for research is proof complexity. In principle, abstract derivations allow for efficient representation of proofs by sharing of subterms: the question is how efficient this representation is. Are they comparable with extended Frege systems, as defined by Cook and Reckhow in [2]? Are there general methods to reduce the size of derivations? The relation between abstract derivations and other alternative representation of proofs, like proof nets (see [7]) and deep derivations (see [1], [13]), is another open question.

Three anonymous referees and Piet Rodenburg are acknowledged for their constructive criticism on an earlier version of this paper.

References

1. Kai Brünnler. *Deep inference and symmetry in classical proofs.* Logos Verlag, Berlin, 2004.
2. Stephen A. Cook and Robert A. Reckhow. Time bounded random access machines. *Journal of Computer and System Sciences*, 7:354–375, 1973.
3. W.W. Craig. Linear reasoning. A new form of the Herbrand-Gentzen theorem. *Journal of Symbolic Logic*, 22:250–268, 1957.
4. Gerard R. Renardel de Lavalette. Interpolation in fragments of intuitionistic propositional logic. *Journal of Symbolic Logic*, 54:1419 – 1430, 1989.
5. Gerard R. Renardel de Lavalette. Abstract derivations, equational logic and interpolation (extended abstract). 2005. `http://www.cs.rug.nl/~grl/pub/lisbon2005ea.pdf`, to appear.
6. Walter Felscher. On interpolation when function symbols are present. *Archiv für mathematische Logik und Grundlagenforschung*, 17:145–158, 1976.
7. Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.
8. G. Grätzer. *Universal Algebra (2nd edition).* Springer-Verlag, New York, 1979.
9. Don Pigozzi. The join of equational theories. *Colloquium Mathematicum*, 30:15–25, 1974.
10. Piet Rodenburg. Interpolation in conditional equational logic. *Fundamenta Informaticae*, 15:80–85, 1991.
11. Piet Rodenburg. A simple algebraic proof of the equational interpolation theorem. *Algebra Universalis*, 28:48–51, 1991.
12. Piet Rodenburg. Interpolation in equational logic. Technical report, University of Amsterdam, Department of Mathematics and Computer Science, Programming Research Group, January 1992. Report P9201.
13. Charles Stewart and Phiniki Stouppa. A systematical proof theory for several modal logics (extended abstract). In Renate Schmidt, Ian Pratt-Hartmann, and Mark Reynolds, editors, *AiML2004 — Advances in Modal Logic (conference proceedings)*, pages 357–371. Department of Computer Science, University of Manchester, Technical Report Series UMCS-04-9-1, 2004.
14. Gaisi Takeuti. Lecture notes on proof theory (mimeographed). Urbana, 1971.
15. Vincent van Oostrom. A simple rewrite proof of the equational interpolation theorem. `http://www.phil.uu.nl/~oostrom/publication/pdf/interpolation.pdf`, 2003.
16. A. Visser, J. van Benthem, D. de Jongh, and G.R. Renardel de Lavalette. NNIL, a study in intuitionistic propositional logic. In A. Ponse, M. de Rijke, and Y. Venema, editors, *Modal Logic and Process Algebra*, pages 289 – 326. CSLI Publications, Stanford (USA), 1995.