ISTITUTO DI ANALISI DEI SISTEMI ED INFORMATICA
CONSIGLIO NAZIONALE DELLE RICERCHE

F. Aureli, A. Formisano, E. Omodeo, M. Temperini

MAP CALCULUS: INITIAL APPLICATION SCENARIOS AND EXPERIMENTS BASED ON OTTER

R. 466 June 1998

Fabiola Aureli – Università di L'Aquila, Dipartimento di Matematica Pura ed Applicata, Via Vetoio – Loc. Coppito, I-67010, L'Aquila, Italia.

Andrea Formisano – Università di L'Aquila, Dipartimento di Matematica Pura ed Applicata, Via Vetoio – Loc. Coppito, I-67010, L'Aquila, Italia.

Eugenio Omodeo – Università di L'Aquila, Dipartimento di Matematica Pura ed Applicata, Via Vetoio – Loc. Coppito, I-67010, L'Aquila, Italia.

Marco Temperini – Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Via Salaria, 113, I-00198, Roma, Italia
and Istituto di Analisi dei Sistemi ed Informatica, CNR, Viale Manzoni, 30, I-00185, Roma, Italia.

Work partially supported by IASI-CNR, by CNR under the coordinated project SETA and by MURST under the project "Tecniche speciali per la specifica, l'analisi, la verifica, la sintesi e la trasformazione di programmi."

ISSN: 1128–3378

Collana dei Rapporti dell'Istituto di Analisi dei Sistemi ed Informatica, CNR

viale Manzoni 30, 00185 ROMA, Italy

tel. ++39-06-77161
fax ++39-06-7716461
email: `iasi@iasi.rm.cnr.it`
URL: `http://www.iasi.rm.cnr.it`

Abstract

Properties of a few familiar structures (natural numbers, nested lists, lattices) are formally specified in Tarski-Givant's map calculus, with the aim of bringing to light new translation techniques that may bridge the gap between first-order predicate calculus and the map calculus. It is also highlighted to what extent a state-of-the-art theorem-prover for first-order logic, namely Otter, can be exploited not only to emulate, but also to reason about, map calculus.

1. Introduction

Everybody remembers that Boole's *Laws of thought* (1854), Frege's *Begriffsschrift* (1879), and the Whitehead-Russell's *Principia Mathematica* (1910) have been three major milestones in the development of contemporary logic (cf. [?, ?, ?, ?]). Only a few people are aware that very important pre-*Principia* milestones were laid down by C.S. Peirce and E. Schröder and culminated in the monumental work [?, ?] on the *Algebra der Logik*.

The "rather capricious line of historical development" of the algebraic form of logic —the MAP CALCULUS, as we will call it— that Peirce and Schröder had contributed to create and which short after the appearance of the *Principia* fell into general oblivion, was already signaled as an anomaly by Tarski in 1941 (cf. [?]). Even more anomalous it should be considered today, since the subsequent work of Tarski and others (cf. [?]) made it clear that map calculus had no inner weaknesses preventing it from becoming the frame for an omni-comprehensive deductive system such as set theory. The rehabilitation of map calculus from its disrepute (whose historical causes are skillfully investigated in [?]) has reopened the opportunity, dismissed for decades, to put together complementary virtues of the map calculus and of first-order predicate logic.

This paper gives a contribution in this direction. A few scenarios of use of Tarski-Givant's map calculus are developed. Properties of familiar structures (natural numbers, nested lists, lattices) endowed with operations and relations are formally specified (see Sec.**??** and following). In particular, the formalization of a classical example (a line-editor, cf. Sec.**??**) enlightens the connection between map calculus and algebraic specifications (cf. [?, ?]). Exercise of this nature, based on paper and pencil for the time being, is aimed at bringing to light translation techniques that may effectively bridge the gap between first-order predicate calculus and the map calculus. The latter, from our point of view, is a promising candidate for the realization of a machine-level inference engine, exploiting the advantages offered by a (ground) algebraic approach.

It is also discussed in what way a state-of-the-art theorem-prover for first-order logic can be employed to emulate, and reason about, map calculus. It will be clear how, moving from first-order logic to map logic and back, permits to exploit the first-order capabilities of such a theorem prover making it possible to express properties commonly requiring higher-order formulations (cf. Sec.**??**). The specification of lattice theory that will be proposed below (cf. Sec.**??**), far from being a straightforward adaptation of any of its preexisting first-order formulations, will reveal particularly challenging due to limited expressive power of the map calculus. As is well-known, in fact, the latter has the same strength (and weakness) as a first-order calculus involving three individual variables altogether. This alternative approach can be seen as a further step in setting up the backbone of a translator between the two languages.

2. Syntactic and semantic background

$\mathcal{L}^{\times}$ is an equational language devoid of individual variables and quantifiers, where one can state properties of binary relations (*maps* for short) over an unspecified, yet fixed, domain $\mathcal{U}$ of discourse. The basic ingredients of this language are:

- three CONSTANTS: $\emptyset$, $\mathbb{1}$, ι;
- infinitely many MAP LETTERS: $\mathsf{p}_1, \mathsf{p}_2, \mathsf{p}_3, \dots$ (whose typographic form can widely vary);
- binary constructs $\cap$, $\triangle$, $\circ$ of map INTERSECTION, map SYMMETRIC DIFFERENCE, and map COMPOSITION;
- the unary construct $^{-1}$ of map INVERSION.

Map expressions are obtained through repeated use of $\cap$, $\triangle$, $\circ$, and $^{-1}$, starting from the map letters p_i, which are interpreted as binary relations over $\mathcal{U}$, and from the mentioned constants. Map equalities have the form $Q{=}R$, where Q and R are map expressions.

Once a nonempty $\mathcal{U}$ has been fixed and subsets $\mathsf{p}_1^{\Im}, \mathsf{p}_2^{\Im}, \mathsf{p}_3^{\Im}, \dots$ of $\mathcal{U}^2$ have been put in correspondence with the p_is , each map expression P comes to designate a specific map $P^{\Im}$, on the basis of the following evaluation rules:

$$\begin{gathered}
\emptyset^{\Im} =_{\mathrm{Def}} \emptyset, \quad \mathbb{1}^{\Im} =_{\mathrm{Def}} \mathcal{U}^2, \quad \iota^{\Im} =_{\mathrm{Def}} \{[a,a] \,:\, a \text{ in } \mathcal{U}\}; \\
(Q \cap R)^{\Im} =_{\mathrm{Def}} \{\, [a,b] \in Q^{\Im} \,:\, [a,b] \in R^{\Im} \,\}; \\
(Q \triangle R)^{\Im} =_{\mathrm{Def}} \{\, [a,b] \in \mathcal{U}^2 \,:\, [a,b] \in Q^{\Im} \text{ iff } [a,b] \notin R^{\Im} \,\}; \\
(Q \circ R)^{\Im} =_{\mathrm{Def}} \{\, [a,b] \in \mathcal{U}^2 \,:\, \exists\, c \text{ in } \mathcal{U} \text{ for which } [a,c] \in Q^{\Im} \text{ and } [c,b] \in R^{\Im} \,\}; \\
(Q^{-1})^{\Im} =_{\mathrm{Def}} \{\, [b,a] \,:\, [a,b] \in Q^{\Im} \,\}.
\end{gathered}$$

Accordingly, an equality $Q{=}R$ is *true* in an interpretation $\Im$ if $Q^{\Im} = R^{\Im}$, and is *false* otherwise. One often strives to specify the collection $\mathcal{C}$ of interpretations that are of interest in some application through a set of equalities that must be true in every $\Im$ of $\mathcal{C}$.[1] Requiring, for instance, that $\iota{=}\mathbb{1}$ leads in essence to propositional logic, because it forces $\mathcal{U}$ to be a singleton, and hence makes $\emptyset$ and $\mathcal{U}^2$ the only possible values for each map expression P. Figures **??** and **??**, to be commented later on, show more sophisticated examples.

We postpone to Sec.**??** the definition of a deductive machinery for $\mathcal{L}^{\times}$.

$\mathcal{L}^{+}$ is a variant version of a *first-order dyadic predicate language*: an atomic formula of $\mathcal{L}^{+}$ has either the form xQy or the form $Q{=}R$, where x, y stand for individual variables (ranging over $\mathcal{U}$) and Q, R stand for map expressions of $\mathcal{L}^{\times}$. Here propositional connectives, and existential/universal quantifiers are employed as usual.

It is shown in [**?**] that in $\mathcal{L}^{+}$ the map constructs $\emptyset, \mathbb{1}, \cap, \triangle, \circ, ^{-1}, =$ can be made to dissolve into connectives and quantifiers: this elimination (whose feasibility was already clear in [**?**]) leads to a (today) far more conventional first-order language, where ι generally takes the typographic form $=$. A remarkable fact about the elimination technique is the following: when one applies it to a sentence α of $\mathcal{L}^{+}$ involving no more than three distinct individual variables, the resulting sentence β will also involve three or fewer variables. This is what happens, e.g., when α is an equality $Q{=}R$ of $\mathcal{L}^{\times}$.

3. Basic extensions of the map language

To enrich $\mathcal{L}^{\times}$ and improve the readability of its map expressions, we can use several pieces of shorthand notation, such as:

$$\begin{aligned}
\overline{P} \equiv_{\mathrm{Def}} \not{P} \quad &\equiv_{\mathrm{Def}} \quad P \triangle \mathbb{1}, \\
P \dagger Q \quad &\equiv_{\mathrm{Def}} \quad \overline{\overline{P} \circ \overline{Q}}, \\
P \setminus Q \quad &\equiv_{\mathrm{Def}} \quad P \cap \overline{Q}, \\
P \cup Q \quad &\equiv_{\mathrm{Def}} \quad (\,P \triangle Q\,) \triangle (\,P \cap Q\,), \\
\Diamond P \quad &\equiv_{\mathrm{Def}} \quad \mathbb{1} \circ P \circ \mathbb{1}.
\end{aligned}$$

Through similar rewriting rules we can extend $\mathcal{L}^{\times}$ in order to emulate familiar constructs such as inclusion and the propositional connectives $\neg$, $\wedge$, $\vee$:

$$\begin{array}{rcllcccl}
 & & P{\subseteq}Q & \equiv_{\mathrm{Def}} & & & P \setminus Q & = \emptyset, \\
\neg P{=}Q & \equiv_{\mathrm{Def}} & P \neq Q & \equiv_{\mathrm{Def}} & & \Diamond & (\,P \triangle Q\,) & = \mathbb{1}, \\
P{=}Q & \wedge & R{=}S & \equiv_{\mathrm{Def}} & (\,P \triangle Q\,) & \cup & (\,R \triangle S\,) & = \emptyset, \\
P{=}Q & \vee & R{=}S & \equiv_{\mathrm{Def}} & (\,P \triangle Q\,) & \circ\mathbb{1}\circ & (\,R \triangle S\,) & = \emptyset.
\end{array}$$

[1]This is typical of algebraic specifications—see, e.g., [**?**].

Axioms

$> \; \equiv_{\text{Def}} <^{-1}$	$\leq \equiv_{\text{Def}} < \triangle \, \iota$
$\mathsf{Const}(\,\mathsf{zero}\,)$	$\mathsf{TotFun}(\,\mathsf{suc}\,)$
$<\subseteq\overline{1\!\!1 \circ \mathsf{zero}}$	$\overline{1\!\!1 \circ \mathsf{zero}}\subseteq 1\!\!1 \circ \mathsf{suc}$
$< \circ \, \mathsf{suc}^{-1} = \leq$	$\leq \triangle > = 1\!\!1$
$< \cap \, \iota = \emptyset$	$< \circ < \subseteq <$

Lemmas

$< \subseteq \not>$	$\mathsf{suc}\subseteq<$
$\iota \cap \not< = \iota$	$\not< = (\, < \triangle \, \iota \,)^{-1}$
$< = \mathsf{suc} \circ < \circ \, \mathsf{suc}^{-1}$	$\mathsf{suc} \circ \mathsf{zero} = \emptyset$
$\mathsf{Fun}(\,\mathsf{suc}^{-1}\,)$	$\mathsf{suc} \underbrace{\circ \mathsf{suc} \cdots \circ \mathsf{suc}}_{n \text{ times}} \subseteq \overline{\iota}$

Figure 1: Theory of natural numbers with successor function and ordering relations

It is also possible to overcome the limitation of not having constants or function symbols available in $\mathcal{L}^{\times}$, because these can be represented by map symbols P subject to the respective conditions that

$$P^{\Im} = \{[e, e]\} \text{ for some } e \in \mathcal{U},$$

and that

$$\text{for all } a \in \mathcal{U}, \text{ there is exactly one } b \in \mathcal{U} \text{ for which } [a, b] \in P^{\Im}.$$

In the map language, these two conditions can be rendered as follows:

$$\mathsf{Const}(\,P\,) \quad \equiv_{\text{Def}} \quad P \circ 1\!\!1 \circ P \subseteq \iota \;\wedge\; P \neq \emptyset,$$
$$\mathsf{TotFun}(\,P\,) \quad \equiv_{\text{Def}} \quad P^{-1} \circ P \subseteq \iota \;\wedge\; P \circ 1\!\!1 = 1\!\!1.$$

This explains the first few axioms of Figure **??**. An abbreviation of the same flavor, namely

$$\mathsf{Fun}(\,P\,) \equiv_{\text{Def}} P^{-1} \circ P \subseteq \iota,$$

which characterizes partial functions, is exploited in Figure **??**.

4. A deductive apparatus for map reasoning

We will now slightly adjust the derivability notion for $\mathcal{L}^{\times}$ formalized in [**?**] to our context. Admittedly, there will be map equalities $P = 1\!\!1$ not derivable from an empty set of premisses but nonetheless *valid*, in the sense that $P^{\Im} = \mathcal{U}^2$ is true in every interpretation $\Im$ of $\mathcal{L}^{\times}$. This is due to an intrinsic limitation: there seems to be no way out of this lack of semantic completeness of the derivability notion for $\mathcal{L}^{\times}$.[2] In spite of this shortcoming, $\mathcal{L}^{\times}$ proves adequate as a support for number theories (cf. [**?**]) as well as for full-blown theories of sets; in these contexts, it can fairly compete with predicate calculus. In short (as we will see through an example in Sec.**??** and will discuss again at the beginning of Sec.**??**), $\mathcal{L}^{\times}$ demonstrates the same power as first-order predicate calculus when the aim is to formalize —and to reason inside— a strong theory (i.e. a theory where the operation of pairing and conjugated projections are definable).

We start with recording onto the following list of schemes an infinite collection $\Lambda^{\times}$ of valid

[2] In [**?**], pp.55–64, the problem of the completability of $\mathcal{L}^{\times}$ is discussed in depth, but left open for the finitary versions of the formalism. Note, however, that since in this paper we are dealing with infinitely many map letters, the incompletability follows from a classical result due to Monk (cf. *ibid.*).

map equalities, to be regarded as the LOGICAL AXIOMS of $\mathcal{L}^{\times}$:

$$\boxed{\begin{array}{rrcl}
 & P \triangle P & = & Q \triangle Q \\
 & P \triangle (Q \triangle P) & = & Q \\
 & (R \cap Q) \triangle (R \cap P) & = & (P \triangle Q) \cap R \\
 & P \cap P & = & P \\
 & \mathbb{1} \cap P & = & P \\
(\star \in \{\triangle, \cap, \circ\}) & (P \star Q) \star R & = & P \star (Q \star R) \\
 & \iota \circ P & = & P \\
 & (P \cup Q) \circ R & = & (Q \circ R) \cup (P \circ R) \\
 & P^{-1^{-1}} & = & P \\
(\star \in \{\cap, \circ\}) & (P \star Q)^{-1} & = & Q^{-1} \star P^{-1} \\
\multicolumn{2}{r}{\big(P^{-1} \circ \big(R \cap \big((P \circ Q) \triangle R\big)\big)\big) \cap Q} & = & \emptyset
\end{array}}$$

Given a collection $\mathtt{E}$ of map equalities, we will denote as $\Theta^{\times}(\mathtt{E})$ the smallest collection of map equalities which both fulfills the inclusion

$$\Lambda^{\times} \cup \mathtt{E} \cup \{ P{=}P \ : \text{P is a map expression} \} \subseteq \Theta^{\times}(\mathtt{E})$$

and enjoys the following closure property: *When $P{=}Q$ and $R{=}S$ both belong to $\Theta^{\times}(\mathtt{E})$, and R occurs in Q, in P, or in both of them, then any equality obtainable from $P{=}Q$ by replacement of some occurrence of R by an occurrence of S belongs to $\Theta^{\times}(\mathtt{E})$ too.*

The notation $\mathtt{E} \vdash^{\times} Q{=}R$ is employed to indicate that $Q{=}R$ belongs to $\Theta^{\times}(\mathtt{E})$. We take an analogous (but semantically complete!) definition of $\vdash^{+}$ for granted.[3]

Notice that we have been using P, Q, R, and S, as meta-variables ranging over map expressions. What would be implied by us changing perspective and regarding P, Q, R, S as individual variables (ruled by understood $\forall$-quantifiers in all logical axiom schemes)? Then each scheme in $\Lambda^{\times}$ would be regarded as a single first-order equality, and we would be dealing with an equational axiomatic first-order theory Θ_{RA} instead of with an alternative formalism. The models of Θ_{RA} are the structures traditionally known as relation algebras, on which [?], p.48, states: *Every equation which is shown to be identically satisfied in every relation algebra yields a schema of which all the particular instances (obtained by substituting predicates for variables) are sentences logically provable in $\mathcal{L}^{\times}$.*

This indicates that we can use an automated deduction tool conceived for first-order logic, Otter to be specific (cf. [?]), to experiment with $\mathcal{L}^{\times}$. Although Otter cannot directly produce derivations of $\mathcal{L}^{\times}$, once the equalities that form $\Lambda^{\times}$ are loaded into Otter, whatever chain of inference steps can be drawn from them witnesses the existence of corresponding lines of reasoning in $\mathcal{L}^{\times}$.

One can moreover load into Otter, along with $\Lambda^{\times}$, a set $\mathtt{E}$ of map equalities, e.g. those in the upper part of Figure **??** (cf. [?], p.184 fol.), and derive theorems of $\Theta^{\times}(\mathtt{E})$, such as those in the lower part of the same figure.

The very shape of the equalities in $\Lambda^{\times}$ is the result of us having carried out a number of experiments of this nature. Other formulations of the logical axioms of $\mathcal{L}^{\times}$ would possibly drive Otter better; anyway, the one we have adopted above is the outcome of a series of ameliorations carried out on an initial version, until we succeeded in getting an automatic proof of various propositions of [?], pp.49-50, that we had chosen as our benchmarks (cf. Figure **??**).

Otter is not specifically oriented to equational logic, and it is conceivable that a system based on term rewriting might fit our needs better. However, we have in mind to perform extensive experimentation with theories of numbers and sets specified in $\mathcal{L}^{\times}$, and we are eager to compare

[3]Any derivability notion for first-order logic can be exploited for $\mathcal{L}^{+}$, cf. e.g. [?].

the results of our experiments with the work of others. Otter is attractive in this respect, because it has been the system underlying experiments of the kind we have in mind, as reported in [?, ?]. Moreover, the fact that Otter encompasses full first-order logic paves the way to combined reasoning tactics that, e.g., perform resolution of $\mathbb{1} \circ P \circ \mathbb{1}=\mathbb{1}$ against $P=\emptyset$.

a.	$P \circ \iota=P$	right unit for $\circ$
b.	$(P \triangle \mathbb{1}) \triangle \mathbb{1}=P$	double complementation law
c.	$P \triangle P=\emptyset$	periodicity of $\triangle$
d.	$P \triangle Q=Q \triangle P$	commutativity of $\triangle$
e.1.	$(P \triangle Q) \triangle Q=P$	
e.2.	$\neg(P \subseteq Q \wedge Q \subseteq R) \vee P \subseteq R$	
f.1.	$P \cup Q=Q \cup P$	
f.2.	$(P \cup Q) \cup R=P \cup (Q \cup R)$	
f.3.	$P \cup Q=\overline{\overline{P} \cap \overline{Q}}$	
g.	$\overline{\overline{P \cap Q} \cap \overline{P \cap \overline{Q}}}=P$	dual of Robbins' law
h.1.	$(P \cap Q) \triangle (P \cap \overline{Q})=P$	variants of Robbins' law
h.2.	$\big(P \cap (Q \triangle R)\big) \triangle \big(P \cap Q\big)=P \cap R$	
i.1.	$(\overline{P} \cap Q) \cup (Q \cap P)=Q$	Huntington's laws
i.2.	$(P \cup Q) \cap (Q \cup \overline{P})=Q$	
j.	$\emptyset \circ P=\emptyset \quad \wedge \quad P \circ \emptyset=\emptyset \quad \wedge \quad P \triangle \emptyset=P \quad \wedge \quad P \cap \emptyset=\emptyset$	
k.	$\big((P \circ Q) \cap R\big)=\emptyset \quad \rightarrow \quad \big((P^{-1} \circ R) \cap Q\big)=\emptyset$	cycle law
l.	$\big((P^{-1} \circ R) \cap Q\big)=\emptyset \quad \rightarrow \quad \big((P \circ Q) \cap R\big)=\emptyset$	cycle law
m.	$(P \cap \iota)=P \quad \rightarrow \quad (P^{-1} \cap \iota)=P \quad \wedge \quad P^{-1}=P$	
n.	$P \subseteq P \circ \mathbb{1} \circ P$	
	$\neg\mathsf{Const}(P) \vee P \subseteq \iota$	
o.	$(\forall x, u, v)\big((x \cap u) \triangle v=\emptyset \quad \longleftrightarrow$ $v \cap (u \triangle \mathbb{1})=\emptyset \wedge$ $\wedge \exists y \; x=v \triangle \big(y \cap (y \triangle \mathbb{1})\big)\big)$	key for Boolean unification

Figure 2: Some benchmark theses, with Otter serving $\mathcal{L}^{\times}$ and $\mathcal{L}^{+}$

Theories of fundamental mathematical importance, namely Peano arithmetic, Zermelo-Fraenkel set theory, and elementary geometry suffer, in their conventional first-order formulation, of a serious drawback: they do not have or admit a finite axiomatization, and hence they present an immediate obstacle to first-order proof methods. This difficulty stems from the induction scheme in arithmetic, from the separation and replacement axiom schemes in set theory, and from the continuity axiom scheme in geometry. Each one of the axiom schemes just mentioned can be expressed by a single meta-sentence where schematic variables standing for arbitrary first-order formulae occur. These schematic variables can be regarded as universally quantified second-order variables, and as such can be treated by higher-order proof assistants, Isabelle to mention one (cf., e.g., [?, ?, ?]).

A similar advantage can be gained by using first-order variables to represent predicate expressions of $\mathcal{L}^{\times}$: virtually any theorem prover or proof assistant can then be used to deal with theories of numbers and sets, because the latter become finitely axiomatized in this fashion. Though not equivalent to the conventional formulation, the new formulation is equi-consistent with it. The explanation is that individual variables do no longer range over the entities $\mathcal{U}$ of the domain of discourse; instead, they are meant to range over the subclasses of $\mathcal{U}^2$. A similar idea enabled von Neumann, Gödel, and Bernays to discover a finite axiomatization for set theory (which, by the way, is sometimes preferred to Zermelo-Fraenkel as a basis for experimentation with theorem provers, cf. [?, ?, ?]). In their formulation of set theory, variables range over the subclasses of the universe of sets, some of which are sets whereas others are not.

We will exploit these ideas in the next section, to provide a finite and equational first-order axiomatization of the theory of lists (which, when viewed as an axiomatization in $\mathcal{L}^{\times}$, consists of infinitely many axioms, though!). Along similar lines we could have formulated a theory of sets which, while being finitely axiomatized like the von Neumann-Gödel-Bernays theory, would bear a closer kinship to Zermelo-Fraenkel.

5. A theory of lists, within map calculus

As will be explained below, the axioms in Figure ?? describe a somewhat more challenging scenario than the ones in Figure ??. Intuitively speaking, here the domain $\mathcal{U}$ of discourse consists of entities of three separate kinds: an infinity of atoms, nested lists constructed out of them, and an individual NL. We are now beginning to experiment with axioms of this kind in Otter.

a.1.	$\mathsf{Const}(\,\mathsf{nl}\,)$
a.2.	$\mathsf{at} \cap \mathsf{nl} = \emptyset$
b.1.	$\mathsf{Fun}(\,\mathsf{hd}\,)$
b.2.	$\mathsf{Fun}(\,\mathsf{tl}\,)$
c.1.	$\mathsf{hd} \circ 1\!\!1 = \mathsf{tl} \circ 1\!\!1$
c.2.	$\mathsf{at} \bigtriangleup (\,\mathsf{nl} \circ 1\!\!1\,) = \overline{\mathsf{tl} \circ 1\!\!1}$
d.1.	$\overline{\mathsf{at}} \subseteq \mathsf{tl}^{-1} \circ \mathsf{hd}$
d.2.	$(\,\mathsf{hd} \circ \mathsf{hd}^{-1}\,) \cap (\,\mathsf{tl} \circ \mathsf{tl}^{-1}\,) \subseteq \iota$
e.1.	$1\!\!1 \circ (\,\mathsf{at} \setminus \mathsf{ocl}\,) = 1\!\!1$
e.2.	$\mathsf{ocl} = \big(\,(\,\iota \cup \mathsf{ocl}\,) \circ \mathsf{hd}^{-1}\,\big) \cup \big(\,\mathsf{ocl} \circ \mathsf{tl}^{-1}\,\big)$
f.1.	$(\,\mathsf{tl} \cup \mathsf{ocl}\,) \cap \iota = \emptyset$
f.2.	$\Diamond\Big(\Big(\,\mathsf{at} \cup (\,\mathsf{nl} \circ 1\!\!1\,) \cup \Big(\,(\,\mathsf{hd} \circ \boldsymbol{P}\,)\ \cap\ (\,\mathsf{tl} \circ \boldsymbol{P}\,)\Big)\Big) \setminus \boldsymbol{P}\,\Big) \dagger \boldsymbol{P} = 1\!\!1$

Figure 3: Axioms on nested lists

The 'nil' predicate nl, according to a., represents the distinguished individual NL, not to be counted among atoms. By b. and c., the 'head' and 'tail' predicates (hd and tl respectively) are partial functions defined for the same entities (to be regarded as the 'lists') of $\mathcal{U}$: the complement of their common domain consists of all atoms together with NL. Notice that, by c.2, the 'is atom' predicate at does not truly depend on its second argument.

The axiom d.1 states that for any given pair a, b in $\mathcal{U}$ with b a list or $b = \mathsf{NL}$, one can find a list whose head is a and whose tail is b; moreover, by d.2, lists that differ from one another cannot have the same head and the same tail. By axiom e., for any given c in $\mathcal{U}$ there is an atom that does not occur in c (as a consequence, the number of atoms will turn out to be infinite as desired). In this connection a recursive characterization of the 'occurs in list' predicate ocl applies: it can be read as "an entity occurs in a list b iff it either coincides with, or occurs in, the head of b, or it occurs in the tail of b". By the acyclicity requirement f.1, a c in $\mathcal{U}$ can neither occur within itself nor be its own tail.

Last comes the somewhat intriguing axiom f.2, which is an induction principle for lists. When its outer map-constructs get eliminated in $\mathcal{L}^{+}$, it becomes

$$(\forall x, y)\Big(\,x \;\mathsf{at}\; y \vee x = \mathsf{NL} \vee \big(\,\exists v(\,x \;\mathsf{hd}\; v \wedge v\,P\,y\,) \wedge \exists w(\,x \;\mathsf{tl}\; w \wedge w\,P\,y\,)\,\big) \rightarrow x\,P\,y\,\Big) \longrightarrow (\forall x, y) x\,P\,y\,,$$

g.1.	$\mathsf{rot}{=}((\mathsf{tl}\circ\mathsf{nl}\circ\mathbb{1})\cap\iota)\triangle$
	$((((\mathsf{hd}\circ\mathsf{hd}^{-1})\cap(\mathsf{tl}\circ\mathsf{tl}\circ\mathsf{tl}^{-1}))\circ\mathsf{rot}\circ\mathsf{tl}^{-1})\cap(\mathsf{tl}\circ\mathsf{hd}\circ\mathsf{hd}^{-1}))$
g.2.	$\mathsf{cat}{=}((\mathsf{hd}\circ\mathsf{nl}\circ\mathbb{1})\cap\mathsf{tl})\triangle$
	$((((\mathsf{tl}\circ\mathsf{tl}^{-1})\cap(\mathsf{hd}\circ\mathsf{tl}\circ\mathsf{hd}^{-1}))\circ\mathsf{cat}\circ\mathsf{tl}^{-1})\cap(\mathsf{hd}\circ\mathsf{hd}\circ\mathsf{hd}^{-1}))$
h.1.	$\mathsf{char}\subseteq\mathsf{at}\cap\iota$
h.2.	$\mathsf{string}{=}\mathsf{nl}\triangle((\mathsf{hd}\circ\mathsf{char}\circ\mathbb{1})\cap(\mathsf{tl}\circ\mathsf{string}\circ\mathbb{1})\cap\iota)$
h.3.	$\mathsf{line}{=}((\mathsf{hd}\circ\mathsf{string}\circ\mathbb{1})\cap(\mathsf{tl}\circ\mathsf{string}\circ\mathbb{1}))\cap\iota$
i.1.	$\mathsf{clear}{=}\mathsf{line}\circ\mathbb{1}\circ((\mathsf{hd}\circ\mathsf{nl})\cap\mathsf{tl})^{-1}$
i.2.	$\mathsf{createLine}{=}\mathsf{clear}\cap\iota$
i.3.	$\mathsf{toBeginning}{=}\mathsf{line}\circ\mathsf{cat}\circ((\mathsf{hd}\circ\mathsf{nl}\circ\mathbb{1})\cap\mathsf{tl})^{-1}$
i.4.	$\mathsf{toEnd}{=}\mathsf{line}\circ\mathsf{cat}\circ((\mathsf{tl}\circ\mathsf{nl}\circ\mathbb{1})\cap\mathsf{hd})^{-1}$
i.5.	$\mathsf{delChar}{=}((\mathsf{line}\circ\mathsf{tl}\circ\mathsf{nl}\circ\mathbb{1})\cap\iota)\triangle$
	$((\mathsf{line}\circ\mathsf{tl}\circ\mathsf{tl}\circ\mathsf{tl}^{-1})\cap(\mathsf{hd}\circ\mathsf{hd}^{-1}))$
i.6.	$\mathsf{shift}{=}(\mathsf{line}\circ\mathsf{tl}\circ\mathsf{tl}\circ\mathsf{tl}^{-1})\cap$
	$(((\mathsf{hd}\circ\mathsf{tl}^{-1})\cap(\mathsf{tl}\circ\mathsf{hd}\circ\mathsf{hd}^{-1}))\circ\mathsf{rot}\circ\mathsf{hd}^{-1})$
i.7.	$\mathsf{moveRight}{=}((\mathsf{line}\circ\mathsf{tl}\circ\mathsf{nl}\circ\mathbb{1})\cap\iota)\triangle\mathsf{shift}$
i.8.	$\mathsf{moveLeft}{=}((\mathsf{line}\circ\mathsf{hd}\circ\mathsf{nl}\circ\mathbb{1})\cap\iota)\triangle\mathsf{shift}^{-1}$
i.9.	$\mathsf{addChar}{=}((((\mathsf{hd}\circ\mathsf{hd}^{-1})\cap(\mathsf{tl}\circ\mathsf{tl}\circ\mathsf{tl}^{-1}))\circ\mathsf{tl}^{-1})\cap(\mathsf{tl}\circ\mathsf{hd}\circ\mathsf{hd}^{-1}))\circ\mathsf{shift}$

Figure 4: Specification of a simple line-editor

where P can be any map expression. One could likewise express in map-theoretic terms the induction principle of Peano arithmetic:

$$(P\cup(\mathsf{suc}^{-1}\circ\mathbb{1}))\cap(\overline{P}\cup(\mathsf{suc}\circ P))\neq\mathbb{1}\vee P{=}\mathbb{1},$$

where the intended meaning of suc is the same as in Figure **??**.

6. A classical example revisited: algebraic specification of a line-editor

For the sake of comparison, we emulate in Figure **??** the abstract data-type specification of a line-editor given in [?], p.58. The specification in [?] exploits a preceding one of the type *string*; likewise, we exploit our previous specification of nested lists.

Axioms g.1 and g.2 in Figure **??** characterize two basic operations on lists: *rotation* and *concatenation*. Rotating a list means removing its head from the beginning to place it at the end of the list. Concatenating two lists means, as usual, inserting the elements of the second of them at the end of the first; notice that we cannot define concatenation as a binary function (since this would be a ternary predicate), and hence we are defining cat as a unary function which concatenates together head and tail of its argument.

Axioms h.1 and h.2 state that a *string* is either NL or a flat list of *characters* drawn from an alphabet consisting of atoms. Axiom h.3 defines a *line* to be a list whose head and tail both are strings; virtually, a *cursor* separates these two strings. Then axioms i.1–i.9 define operations on lines: clear empties both the string that precedes the cursor and the one that follows it, createLine initializes a line as a pair of empty strings; toBeginning and toEnd bring the cursor at the two extreme positions of a line; delChar erases the character following the cursor, if any, while addChar inserts a character after the cursor and shifts the cursor to the immediate right of the new character; moveRight and moveLeft shift the cursor to the right or to the left by one character, when this is possible.[4]

On p.3, [**?**] says: *It was early discovered that simple algebraic specifications, which consist of listings of sort symbols, operation symbols, and equations, are in their pure form not appropriate for writing down specifications of larger software systems. Roughly speaking, in this regard they correspond to assembly code and not to structured programs of high level languages.*

Admittedly, the use of $\mathcal{L}^{\times}$ leads to equational specifications of an even lower level than those [**?**] refers to; indeed, in our context it would be difficult to draw a precise distinction between *sorts* and predicates of other kinds. We intend to propose $\mathcal{L}^{\times}$ as a machine-oriented language rather than as a a man-oriented language. The language $\mathcal{L}^{\times}$ may look distasteful to reading, hence it ought to be clear that techniques for moving back and forth between first-order logic and map logic exist and are partly implemented (cf. [**?**, **?**]); we are striving to ameliorate and extend them, so as to have support for a more comfortable, still logic-based, style of specification. Thanks to these techniques, we hope, the automatic crunching of map-based axioms of the kind discussed in this paper can be hidden inside the back-end of an automated reasoner.

7. A tough exercise in map specification: lattice theory

It is well-known that the map calculus has the same expressive power as a first-order language employing at most three distinct individual variables. As mentioned, techniques for translating formulae of such a language to the map calculus, and vice-versa, have been studied in depth [**?**]. Starting with this section and through the sequel of this paper, we expound an interesting case-study: the encoding of the lattice theory in the map calculus. The main aim of this study is to show how an alternative approach can be successful, even when the underlying theory is too weak to be treated by the techniques standardized by [**?**]. In a context like ours, where the map calculus is seen as a "machine language" for first-order theorem provers, providing an alternative approach to carry out the translation between the two reasoning levels has its own significance and seems worth the effort.

Let us recall a definition from [**?**], p.62:
Definition. *A sentence α of $\mathcal{L}^{+}$ is said to be* EXPRESSIBLE *in $\mathcal{L}^{\times}$ if there is a map equality β of $\mathcal{L}^{\times}$ for which $\alpha \dashv\models^{+} \beta$, i.e., $\alpha^{\Im} = \beta^{\Im}$ in every interpretation $\Im$.*

Among sentences expressible in this sense, one finds all sentences in three variables, defined as follows:
Definition. *A sentence α of $\mathcal{L}^{+}$ is said to be* IN k VARIABLES *(k a natural number) if no subformula φ of α involves more than k distinct free variables.*

As for expressibility, [**?**] and [**?**] classify many sentences as shown in Figure **??**. Of these, the ones in the first group are provable in the most varied theories of sets; those in the second group can be interpreted in LATTICE THEORY (which deals with the ordering relations where every finite nonempty set of the domain has inf[imum] and sup[remum]). In contrast with [**?**], which claims that **(3)** is inexpressible, [**?**] says that it is not known whether or not **(3)** is expressible in $\mathcal{L}^{\times}$.

The collection of all sentences expressible in $\mathcal{L}^{\times}$ was shown to be undecidable (!) in [**?**].

[4]Thanks to the richness of the nested list structure underlying our current application scenario, one could extend it to specify a file or hypertext editor.

(1)	$(\forall x,y)\ \exists w\,\forall v(v\in w\leftrightarrow v=x\vee v=y)$	Yes
(1a)	$(\forall x,y)\ \big(\exists u\,(x\in u\wedge y\in u)\rightarrow\exists w\,\forall v(v\in w\leftrightarrow v=x\vee v=y)\big)$	No
(1b)	$(\forall x,y)\ \big(\exists u\,x\in u\wedge\exists v\,y\in v\rightarrow\exists w\,\forall v(v\in w\leftrightarrow v=x\vee v=y)\big)$	No
(1c)	$(\forall x,y)\ \big(\exists u\,x\in u\wedge\exists v\,y\in v\rightarrow\exists w\,\big(\forall v(v\in w\leftrightarrow v=x\vee v=y)\wedge\exists u\,w\in u\big)\big)$	Yes
(2)	$(\forall x,y)\ \exists u\,\forall v(v\in u\leftrightarrow v\in x\vee v\in y)$	No
(2a)	$(\forall x,y)\ \exists u\,\forall v(v\in u\leftrightarrow v\in x\vee v\in y)\wedge\forall x\,\exists y\,\forall v(v\in y\leftrightarrow v=x)$	Yes
(3)	$(\forall x,y)\ \exists w\,\forall u(u\in w\leftrightarrow u=x\vee u\in y)$	No (?)
(3a)	$(\forall x,y)\ \exists w\,\forall u(u\in w\leftrightarrow u=x\vee u\in y)\wedge\exists z\forall x\,\neg x\in z$	Yes
(4)	$(\forall x,y)\ \exists v\,\forall u(x\leq u\wedge y\leq u\leftrightarrow v\leq u)$	No
(5)	$(\forall x,y)\ \exists v\,\forall u(u\leq x\wedge u\leq y\leftrightarrow u\leq v)$	No
(4a)	$(\forall x,y)\ \exists v\big(x\leq v\wedge y\leq v\wedge\forall u(x\leq u\wedge y\leq u\rightarrow v\leq u)\big)$	No
(5a)	$(\forall x,y)\ \exists v\big(v\leq x\wedge v\leq y\wedge\forall u(u\leq x\wedge u\leq y\rightarrow u\leq v)\big)$	No
(6)	$(\mathbf{4})\wedge(\mathbf{5})\wedge\forall x\,x\leq x\wedge(\forall x,y,z)(x\leq y\wedge y\leq z\rightarrow x\leq z)$ $\wedge(\forall x,y)(x\leq y\wedge y\leq x\rightarrow x=y)\wedge\exists x\,\forall y\,x\leq y\wedge\exists y\,\forall x\,x\leq y$	No

Figure 5: Expressibility in set theory and lattice theory

It often turns out that a sentence α, even though inexpressible when taken in isolation, becomes expressible within the context of a theory. This is to say, when one adopts a decidable collection T of sentences as axioms, it may well be the case that $T\vdash^{+}\alpha\leftrightarrow\beta$, where β is in three variables (or, which amounts to the same thing, β belongs to $\mathcal{L}^{\times}$).[5]

This new, more generous, meaning of the word "expressibility", trivializes the entire question of expressibility in STRONG THEORIES, such as are number teories (e.g. the Peano arithmetic or the additive-multiplicative theory of real numbers) and set theories (Zermelo-Fraenkel, Gödel-Bernays, etc.). We are referring to theories where by a sentence in three variables one can state that two specific relations —let us denote them here as ℓ (for "left") and r (for "right")— are CONJUGATED QUASI-PROJECTIONS, in the sense that

- ℓ and r are functions (at least partial) on the domain $\mathcal{U}$ of discourse,
- for any pair a,b of entities in $\mathcal{U}$ there is a c in $\mathcal{U}$ such that $\ell(c)=a$ and $r(c)=b$.

Indeed, there is a general technique that enables one to reduce to three the number of variables in any sentence α of the language of a strong theory, by suitably exploiting the map expressions that describe conjugated quasi-projections. A simple illustration of this can be found in [**?**], which provides as an example an $\mathcal{L}^{\times}$-specification of sum and product over natural numbers.

What shall we do if a theory is not strong? Succeeding in expressing in $\mathcal{L}^{\times}$ its axioms, if nothing else, would itself be a gratification; however, the axioms of lattice theory (cf. sentences **(4a)**, **(5a)**, **(6)** in Figure **??**) appear to be already beyond the expressive boundaries of $\mathcal{L}^{\times}$.

Our goal, in the next four sections, is to encode in $\mathcal{L}^{\times}$ a satisfactory surrogate of lattice theory by resorting to a device of a semantic, rather than of a syntactic, nature. This device consists in enlarging the domain of discourse. The idea is simply to avoid referring to the domain D, partially ordered and equipped with infimum and supremum operations; this will be superseded

[5] [**?**] makes the following example: resorting to four variables may seem essential to express the existence of four distinct entities in the domain $\mathcal{U}$ of discourse, but in the theory of strict total orderings $<$, the circumstance can be stated as follows: $(\exists x,y)\big(x<y\wedge\exists x(y<x\wedge\exists y\,x<y)\big)$.

by the domain $\{\emptyset\} \cup \{\{a, b\}\ :\ a, b\ \text{ in }\ D\}$, whose singletons $\{a\}$ can be identified with the entities a in the original domain D.[6] Unordered pairs are in a sense extraneous to the structure $D, \leq, \mathsf{inf}, \mathsf{sup}$ in which we are truly interested, very much like singletons are replacements for the true entities of discourse; nevertheless, the formal properties to be stated in the following will enable one to reason, by 'metaphor', on the usual lattices. The operations inf e sup shall be treated as unary operations, hence binary relations, over the (enlarged) domain of discourse: an essential condition, this, for their properties to be formalizable in the map calculus.

The characterization of lattices will be carried out so as to get along the way weak but useful deductive contexts.[7] The technique to be exploited can be adapted to any context where each $(n+1)$-ary operation $\star$ (like inf and sup) meets the condition

$$\star(x_0, \ldots, x_n) = \star(x_{\pi_0}, \ldots, x_{\pi_n}) \text{ for any permutation } \pi.$$

8. Inclusion theory, within map calculus

Can one instruct a theory of inclusion which does not presuppose a more fundamental theory of membership? By performing this task, one should rediscover something strictly akin to Aristotle's assertory syllogistic.

Let us recall, to start with, that the inclusion relation $\subseteq$ is a partial ordering relation:

$$\subseteq \circ \subseteq\ \subseteq\ \subseteq, \qquad \subseteq \cap \supseteq = \iota.$$

Clearly, the former of these condition expresses the transitivity law while the latter combines the reflexivity law $\iota \subseteq\ \subseteq$ and the antisymmetry law $\subseteq \cap \supseteq\ \subseteq \iota$.

Preliminary to introducing a counterfeit membership relation (which will, in fact, be a subrelation of inclusion), we now introduce new forms of abbreviations.[8] By putting[9]

$$\mathsf{void}\ \equiv_{\text{Def}}\ \subseteq \dagger \emptyset, \qquad \text{i.e.} \qquad V\ \mathsf{void}\ \_\ \equiv_{\text{Def}} \forall x\, V \subseteq x,$$

and

$$\mathsf{snglORvoid}\ \equiv_{\text{Def}} (\,\iota \cup \not\supseteq\,) \dagger \mathsf{void}, \qquad \mathsf{sngl}\ \equiv_{\text{Def}} \overline{\mathsf{void}} \cap \mathsf{snglORvoid},$$

i.e. $$S\ \mathsf{sngl}\ \_\ \equiv_{\text{Def}} \neg S\ \mathsf{void}\ \_\ \wedge\ \forall x\,(\,x \subseteq S \rightarrow x = S \vee x\ \mathsf{void}\ \_\,),$$

one easily verifies in $\mathcal{L}^{\times}$ that

$$\mathsf{void} \circ \mathbb{1} = \mathsf{void} \dagger \emptyset, \qquad \mathsf{sngl} \circ \mathbb{1} = \mathsf{sngl} \dagger \emptyset;$$

these state that the predicates 'is empty' and 'is singleton' do not depend on their second argument. The existence of a sole void is easily proved along the following line: if V and W are void, then either of them is included in the other, hence the two are equal, thanks to antisymmetry. (Of course this argument cannot be mimicked directly in $\mathcal{L}^{\times}$ where variables for V, W are missing.)

Here follows the definition of $\in$ that we set up:

$$\in\ \equiv_{\text{Def}} \mathsf{sngl} \cap \subseteq, \qquad \text{i.e.} \qquad S \in X\ \equiv_{\text{Def}} S\ \mathsf{sngl}\ \_ \wedge S \subseteq X.$$

The addition of the following EXTENSIONALITY axiom appears to be mandatory:

$$\supseteq \dagger \notin\ \subseteq\ \supseteq, \qquad \text{i.e.} \qquad \forall w(\,w \in X \rightarrow w \subseteq Y\,) \rightarrow X \subseteq Y.$$

One easily sees (at least in first-order logic) that if the entities belonging to X and to Y are the same (i.e. $\forall w\,(\,w \in X \leftrightarrow w \in Y\,)$), then X and Y are, by antisymmetry, equal; analogously, if X and Y include the same entities (i.e. $\forall w\,(\,w \subseteq X \leftrightarrow w \subseteq Y\,)$), then extensionality and

[6]A domain of sets richer than this, for example the family $\mathcal{P}(D)$ of all subsets of D, would also do the job. This is an idea to which we might resort in the future, in order to cope with the theories of *complete* lattices, and of Boolean algebras.

[7]For example, the inclusion theory to be seen in the sequel can be seen as a proto-theory of types.

[8]Forms of abbreviation understood in what precedes have been $\supseteq\ \equiv_{\text{Def}}\ \subseteq^{-1}$ and $xPyQz\ \equiv_{\text{Def}} xPy \wedge yQz$.

[9]Following Prolog, we represent by $\_$ a variable that occurs only once.

antisymmetry, taken together, force them to be equal.

It seems convenient to end by postulating the EXISTENCE OF VOID:

$$\mathsf{void} \neq \emptyset, \qquad \text{i.e.} \qquad \exists\, v\, \forall\, y\; v \subseteq y.$$

Analogously we could —but we refrain from this— add postulates like

$$\supseteq \dagger\emptyset \neq \emptyset \qquad (\text{i.e.} \qquad \exists\, t\, \forall\, y\; y \subseteq t);$$

however this assumption that an omni-comprehensive set exists, or similar ones (e.g., assuming that the family of sets to be dealt with enjoys particular closure properties), would be out of scale w.r.t. the limited goals of the present work.[10]

9. Selectors and choice functions, within map calculus

We say that σ is a SELECTOR if it enjoys the following properties:

$$\sigma \subseteqq\ \subseteq \cap\, \mathsf{snglORvoid}, \qquad \text{i.e.} \qquad X\sigma Y \to X \subseteq Y \wedge (\,\forall\, v \subseteq X\,)(\,v = X \vee v\; \mathsf{void}\ \_\,).$$

We then say that a selector is GLOBAL when

$$\overline{\mathsf{void}} \subseteqq \sigma^{-1} \circ \mathbb{1} \qquad \text{and moreover} \qquad \mathsf{void} \cap \sigma \subseteqq \mathsf{void}^{-1},$$

i.e.

$$\neg Y\; \mathsf{void}\ \_ \to \exists\, x\; x\sigma Y \wedge (\,\forall\, x\sigma Y\,) \neg x\; \mathsf{void}\ \_\,.$$

An example of global selector is membership. In a sense this is the greatest of all global selectors σ, each one of which is in fact bound to fulfill the condition

$$\sigma \subseteqq\ \in \cup (\,\mathsf{void}\ \cap\ \mathsf{void}^{-1}\,).$$

To switch to the opposite extreme, let us consider the global selectors that are in a sense minimal: these are the so-called GLOBAL CHOICE FUNCTIONS. Any such η is characterized by

- being a global selector;
- being a function in its second argument, and, as such
- being total (for this, it suffices to require that $\mathsf{void} \cap \mathsf{void}^{-1} \subseteqq \eta$).

One easily sees what is the task of η (cf. Figure **??**), namely to extract from any nonempty set one and only one singleton contained in it—furthermore, for the sake of definiteness, we want that η associates to the empty set the empty set itself.

In order to base a lattice theory on $\mathcal{L}^{\times}$, we need *two* global choice functions, η_1 and η_2, (for our purposes what really counts is that they be defined for all sets with one or two members; however, while singletons have been characterized already, pairs will be defined just through η_1 and η_2). In addition to being global choice functions, η_1 and η_2 shall disagree whenever possible. DISAGREEMENT postulate:

$$\eta_1 \cap \eta_2 \subseteqq \iota, \qquad \text{i.e.} \qquad Y\; \eta_1\; X \wedge Y\; \eta_2\; X \to X = Y.$$

If we were to treat more than two global choice functions —which would be useless here, as we need to consider sets with two members at most—, the disagreement postulate would take the form shown at the bottom of Figure **??**.

10. Unordered pair theory, within map calculus

With the conceptual devices made ready so far, we can instruct a theory of unordered pairs. To the postulates on inclusion we add the following PAIRING postulate:

$$\mathsf{sngl} \cap \mathsf{sngl}^{-1} \subseteqq \eta_1 \circ \eta_2^{-1} \cup \eta_2 \circ \eta_1^{-1},$$

[10]On the other hand, something like the following UNION postulate might turn out useful in order to move from lattice theory to the theory of *complete* lattices:

$$\subseteq \circ \supseteq = \mathbb{1}, \qquad \text{i.e.} \qquad \exists\, u(\,X \subseteq u \wedge Y \subseteq u).$$

$$\boxed{\begin{array}{rcll}
\eta_i & \subseteq & \in \cup(\,\mathsf{void} \cap \mathsf{void}^{-1}\,)\,, & \text{i.e.,} \\
Y\eta_i X & \rightarrow & Y \in X \vee (\,Y\ \mathsf{void}\ \_ \wedge X\ \mathsf{void}\ \_\,)\,; & \\
\hline
\eta_i \circ \eta_i^{-1} & \subseteq & \iota\,, & \text{i.e.,} \\
Y\eta_i X \wedge Z\eta_i X & \rightarrow & Z = Y\,; & \\
\hline
\iota & \subseteq & \eta_i^{-1} \circ \eta_i\,, & \text{i.e.,} \\
 & \exists\, x & x\eta_i Y\,; & \\
\hline
\mathbb{1} \circ (\,\eta_i \cap \eta_{i+1}\,) & = & \emptyset \dagger (\, \notin \cup \eta_1 \cup \cdots \cup \eta_i\,)\,, & \text{i.e.,} \\
\exists\, y(\,y\eta_i X \wedge y\eta_{i+1} X\,) & \leftrightarrow & \forall\, y(\,y \notin X \vee y\eta_1 X \vee \cdots \vee y\eta_i X\,)\,. &
\end{array}}$$

Figure 6: Properties of global choice functions $\eta_1, \eta_2, \eta_3, \ldots$

i.e.

$$X\ \mathsf{sngl}\ \_ \wedge Y\ \mathsf{sngl}\ \_ \rightarrow \exists\, w(\,(\,X\ \eta_1\ w \wedge Y\ \eta_2\ w\,) \vee (\,X\ \eta_2\ w \wedge Y\ \eta_1\ w\,)\,).$$

(With this, η_1^{-1} and η_2^{-1} come close to being conjugated quasi-projections, relative to the sub-domain of $\mathcal{U}$ made of all singletons.)

A set devoid of members, or with only one or two, can be characterized as follows in the theory of inclusion enriched with the axioms already seen about η_1 and η_2:

$$\mathsf{dbl}\ \equiv_{\text{Def}} (\ \not\supseteq \cup \iota\,) \dagger (\,\mathsf{void} \cup \eta_1 \cup \eta_2\,)$$

i.e.

$$W\ \mathsf{dbl}\ X\ \equiv_{\text{Def}} \forall\, z(\,z \subseteq W \neq z \rightarrow z\ \mathsf{void}\ \_ \vee z\ \eta_1\ X \vee z\ \eta_2\ X\,).$$

At this point we add a weak postulate of UNION, which regards singletons:

$$\eta_1 \cap \overline{\mathsf{void}} \subseteq\ \subseteq \circ (\,(\ \supseteq \circ\, \eta_2\,) \cap \mathsf{dbl}\,),$$

i.e.

$$L\ \eta_1\ X \wedge \neg L\ \mathsf{void}\ \_ \rightarrow (\,\exists\, w, u\,)(\,L \subseteq w\ \mathsf{dbl}\ X \wedge w \supseteq u\ \eta_2\ X\,).$$

11. Lattice theory: an inexpressible outflanked

In sight of a theory of lattices, we introduce now the predicate letters inf and sup, for which we demand in the first place that

- they be (partial) functions in their second argument:
$$\mathsf{inf} \circ \mathsf{inf}^{-1} \subseteq \iota, \qquad \mathsf{sup} \circ \mathsf{sup}^{-1} \subseteq \iota;$$
- they have in their domain all singletons and pairs:
$$\mathsf{dbl} \setminus \mathsf{void} \subseteq \left\{ \begin{array}{c} \mathsf{inf}^{-1} \\ \mathsf{sup}^{-1} \end{array} \right\} \circ \mathbb{1}, \quad \text{i.e.} \quad \neg X\ \mathsf{void}\ \_ \wedge X\ \mathsf{dbl}\ \_ \rightarrow \exists\, y\ y \left\{ \begin{array}{c} \mathsf{inf} \\ \mathsf{sup} \end{array} \right\} X;$$
- every image of theirs be a singleton:
$$\mathsf{inf} \cup \mathsf{sup} \subseteq \mathsf{sngl}, \quad \text{i.e.} \quad Y\ \mathsf{inf}\ X \vee Y\ \mathsf{sup}\ X \rightarrow Y\ \mathsf{sngl}\ \_;$$
- they disagree with each other:
$$\mathsf{inf} \cap \mathsf{sup} \subseteq \iota.$$

Nothing very engaging so far: we could take inf and sup to be $\eta_1 \setminus \mathsf{void}$ and $\eta_2 \setminus \mathsf{void}$. However, here we arrive at the partial ordering relation of a lattice:

$$\leq\ \equiv_{\text{Def}} (\ \in \cap\, \mathsf{inf}\,) \circ (\ \in \cap\, \mathsf{sup}\,)^{-1},$$

i.e.

$$X \leq Y\ \equiv_{\text{Def}} \exists\, u\,(\,X \in u \ni Y \wedge X\ \mathsf{inf}\ u\ \mathsf{sup}^{-1}\ Y\,).$$

We impose on this relation the partial ordering laws, which we state in the following form:

$$\leq \circ \leq\ \subseteq\ \leq, \qquad \leq \cap \geq\ = \mathsf{sngl} \cap \iota.$$

(Therefore $X \,\mathsf{sngl}\, \_{\rightarrow} X \leq X$, whence, by exploiting the definition of $\leq$ and the disagreement between inf and sup, we will obtain that

$$X \,\mathsf{sngl}\, \_{\rightarrow} X \,\mathsf{inf}\, X \,\mathsf{sup}\, X\,,$$

to wit that

$$\mathsf{sngl} \cap \iota \subseteq \mathsf{inf} \cap \mathsf{sup},$$

i.e.: “every singleton is a fixpoint of both inf and sup”.)

We conclude with more intricate conditions on inf and sup, which are the ones of MONOTONICITY and of MIN-/MAX-IMALITY. Let us begin with the former two:

$$X \subseteq Y \wedge L \,\mathsf{inf}\, X \wedge M \,\mathsf{inf}\, Y \wedge L \neq M$$
$$\rightarrow (\forall\, w, u\,)(\, w \,\mathsf{dbl}\, u \wedge L \in w \ni M \rightarrow M \,\mathsf{inf}\, w\,),$$

which is to say

$$\delta \cap (\,\mathsf{inf} \circ \supseteq \circ\, \mathsf{inf}^{-1}\,) \subseteq (\,\notin \cup\, \mathsf{inf}\,) \dagger (\,(\,\overline{\mathsf{dbl}} \dagger \emptyset\,) \cup \not\ni\,),$$

and an analogous condition on sup. From these one gets that the inf of a set is a lower bound for its members ($\mathsf{inf} \circ \ni\, \subseteq\, \leq$), while its sup is an upper bound.

As for the latter two conditions, one is:

$$(\,\leq \dagger \notin\,) \circ \mathsf{inf}^{-1} \subseteq\, \leq, \qquad \text{i.e.} \qquad Y \,\mathsf{inf}\, W \wedge (\,\forall\, z \in W\,) X \leq z \rightarrow X \leq Y\,.$$

This states that when the operation inf is applied to a pair or singleton W, or when it anyway associates a result to W, it produces the greatest lower bound of the members of W as it should. Analogously one characterizes the minimality of the upper bound produced by sup.

Let us now put together the various pieces of lattice theory developed so far. The map letters that we are regarding as PRIMITIVE are $\subseteq$, η_1, η_2, inf, sup; the DERIVED ones are introduced via the abbreviations of Figure **??**.

8./1.	$\geq$	$\equiv_{\mathrm{Def}}$	$\leq^{-1}$
	$\supseteq$	$\equiv_{\mathrm{Def}}$	$\subseteq^{-1}$
2.	$\not\subseteq$	$\equiv_{\mathrm{Def}}$	$\overline{\subseteq}$
	$\not\supseteq$	$\equiv_{\mathrm{Def}}$	$\overline{\supseteq}$
3.	void	$\equiv_{\mathrm{Def}}$	$\subseteq \dagger \emptyset$
	$\mathsf{snglORvoid}$	$\equiv_{\mathrm{Def}}$	$(\,\iota \cup \not\supseteq\,) \dagger \mathsf{void}$
4.	sngl	$\equiv_{\mathrm{Def}}$	$\mathsf{snglORvoid} \setminus \mathsf{void}$
5.	$\in$	$\equiv_{\mathrm{Def}}$	$\mathsf{sngl} \cap \subseteq$
	$\ni$	$\equiv_{\mathrm{Def}}$	$\in^{-1}$
6.	$\notin$	$\equiv_{\mathrm{Def}}$	$\overline{\in}$
	$\not\ni$	$\equiv_{\mathrm{Def}}$	$\overline{\ni}$
7.	dbl	$\equiv_{\mathrm{Def}}$	$(\,\not\supseteq \cup\, \iota\,) \dagger (\,\mathsf{void} \cup \eta_1 \cup \eta_2\,)$
	$\leq$	$\equiv_{\mathrm{Def}}$	$(\,\in \cap\, \mathsf{inf}\,) \circ (\,\in \cap\, \mathsf{sup}\,)^{-1}$

Figure 7: derived map letters.

In Figure **??** we show the AXIOMS, through which we have meant to characterize the LATTICE structure. In the figure, η stands in turn for η_1 and for η_2.

12. Conclusions

First-order predicate logic undoubtedly deserves the primacy, with respect to the map calculus, of user-friendliness and expressive manageability. One cannot, however, discard beforehand the

a.	$\subseteq \circ \subseteq$	$\subseteq$	$\subseteq$
	$\subseteq \cap \supseteq$	$=$	ι
b.	$\not\ni \dagger \subseteq$	$\subseteq$	$\subseteq$
	void	$\neq$	$\emptyset$
c.	η	$\subseteq$	$\subseteq \cap \mathsf{snglORvoid}$
d.	$\overline{\mathsf{void}}$	$\subseteq$	$\eta^{-1} \circ \mathbb{1}$
	$\mathsf{void} \cap \eta$	$\subseteq$	void^{-1}
e.	$\eta \circ \eta^{-1}$	$\subseteq$	ι
	$\mathsf{void} \cap \mathsf{void}^{-1}$	$\subseteq$	η
f.	$\eta_1 \cap \eta_2$	$\subseteq$	ι
g.	$\mathsf{sngl} \cap \mathsf{sngl}^{-1}$	$\subseteq$	$\eta_1 \circ \eta_2^{-1} \cup \eta_2 \circ \eta_1^{-1}$
	$\subseteq \circ \big((\supseteq \circ \eta_2) \cap \mathsf{dbl} \big)$	$\supseteq$	$\eta_1 \cap \overline{\mathsf{void}}$
h.	$\mathsf{inf} \circ \mathsf{inf}^{-1}$	$\subseteq$	ι
	$\mathsf{sup} \circ \mathsf{sup}^{-1}$	$\subseteq$	ι
i.	$\mathsf{dbl} \setminus \mathsf{void}$	$\subseteq$	$\mathsf{inf}^{-1} \circ \mathbb{1}$
	$\mathsf{dbl} \setminus \mathsf{void}$	$\subseteq$	$\mathsf{sup}^{-1} \circ \mathbb{1}$
j.	$\mathsf{inf} \cup \mathsf{sup}$	$\subseteq$	sngl
	$\mathsf{inf} \cap \mathsf{sup}$	$\subseteq$	ι
k.	$\leq \circ \leq$	$\subseteq$	$\leq$
	$\leq \cap \geq$	$=$	$\mathsf{sngl} \cap \iota$
l.	$\delta \cap (\mathsf{inf} \circ \supseteq \circ \mathsf{inf}^{-1})$	$\subseteq$	$(\notin \cup \mathsf{inf}) \dagger \big((\overline{\mathsf{dbl}} \dagger \emptyset) \cup \not\ni \big)$
	$\delta \cap (\mathsf{sup} \circ \supseteq \circ \mathsf{sup}^{-1})$	$\subseteq$	$(\notin \cup \mathsf{sup}) \dagger \big((\overline{\mathsf{dbl}} \dagger \emptyset) \cup \not\ni \big)$
m.	$(\leq \dagger \notin) \circ \mathsf{inf}^{-1}$	$\subseteq$	$\leq$
	$\mathsf{sup} \circ (\not\ni \dagger \leq)$	$\subseteq$	$\leq$

Figure 8: lattice theory

idea that the map calculus may perform better in the rôle of basic machine-reasoning layer. This expectation deserves, in our opinion, a serious and twofold experimentation effort. On the one hand, it calls for

- development of effective theorem-proving techniques directly rooted on the map calculus; on the other hand, it requires
- sophisticated techniques for translating sentences —or even entire sets of axioms— from first-order logic into the map language.

In essence the latter techniques (of which an example can be found in [**?**]) are to translate formal specifications, phrased in first-order logic as is nowadays more common, into a specialized area of algebra. How complex this translation task can be in a concrete situation, can be fully grasped through the lattice scenario—and to a lesser extent also through the other examples in this paper.

It is a stimulating fact of mathematics that one cannot decide the precise extent to which this translation of logic into algebra is possible (cf. [**?**]); as a consequence, this is an issue to be tackled pragmatically and conservatively. Powerful ideas have emerged from a protracted stream of research initiated in the fourties and finally blossomed in the Tarski-Givant monograph, which indicates how any theory regarding either sets or arithmetics can be phrased in map-theoretic terms. We aim at converting these ideas into tools inside a platform for computational logic.

Evidently, a good first-order theorem prover such as Otter, or simply a theorem prover for pure equational logic, provides adequate support to symbolic manipulations in the map calculus. Even more importantly, the first-order predicate formalism offers a basis for schematizing meta-theorems of the map calculus, as well as for proving them. In some cases, it enables one to compress into a single quantified sentence an infinite axiom scheme of a theory based on map calculus (to stress this fact, we have printed in boldface an induction principle referring to lists in Figure **??**). Even though we are eagerly following this approach in order to play, and experiment with, specifications written in the map language, we have in mind to invert the approach in the long run. Like others (cf. [**?**, **?**]), we believe that the map calculus deserves an autonomous and effective instrumentation to be put to the service of first-order reasoning, and of automated reasoning in general.

Acknowledgements

We are indebted to IASI (*Istituto di Analisi dei Sistemi ed Informatica*, CNR) for promoting exchanges of ideas between authors, to Vuokko-Helena Caseiro for careful revision of the manuscript, to Paul Broome for first bringing [**?**] to our attention during the ICLP conference held in Budapest in 1993, to Emidio Silvestri for paving the way to this research with a Prolog implementation of a logic-to-mapalgebra translator, to Domenico Cantone and Alessandra Cavarra for very pleasant summer conversations which led to discoveries on how to improve the translator. Thanks are due to an anonymous referee for helpful comments.

References

[1] I. H. Anellis and N. R. Houser. Nineteenth century roots of algebraic logic and universal algebra. In H. Andréka, J. D. Monk, and I. Németi, editors, *Algebraic Logic*, volume 54 of *Colloquia Mathematica societatis János Bolyai*, pages 1–36. North-Holland Publishing Co., 1991.

[2] R. Behnke, R. Berghammer, and P. Schneider. Machine Support of Relational Computations: The Kiel RELVIEW System. Bericht No. 9711. Institüt für Informatik und Praktische Mathematik, Christian-Albrechts-Universität Kiel. 1997.

[3] I. M. Bocheński. *A history of formal logic.* Chelsea, 1970. Thomas, I. editor and translator.

[4] G. Boole. *An investigation of the laws of thought* on which are founded the mathematical theories of logic and probabilities. Dover books in Advanced Mathematics. 1854.

[5] R. Boyer, E. Lusk, W. McCune, R. Overbeek, M. Stickel, and L. Wos. Set theory in first-order logic: Clauses for Gödel's axioms. *J. Automated Reasoning*, 2(3):287–327, 1986.

[6] D. Cantone, A. Cavarra, and E. G. Omodeo. On existentially quantified conjunctions of atomic formulae of $\mathcal{L}^+$. In M. P. Bonacina and U. Furbach, editors, *Proceedings of the FTP97 International workshop on first-order theorem proving*, pages 45–52, 1997. RISC-Linz Report Series No.97-50.

[7] B. Dwyer. LIBRA: A Lazy Interpreter of Binary Relational Algebra. Technical Report 95-10, Department of Computer Science. University of Adelaide. 1995.

[8] H. Ehrig and B. Mahr. *Fundamentals of algebraic specification 1* – Equations and initial semantics. Springer-Verlag, Monographs on Theoretical Computer Science, 1985.

[9] H. B. Enderton. *A Mathematical Introduction to Logic.* Academic Press, New York and London, 1972.

[10] I. Guessarian. *Algebraic semantics.* Springer-Verlag, Lecture Notes in Computer Science, 99, 1981.

[11] E. Grädel, Ph. G. Kolaitis, and M. Y. Vardi. On the decision problem for two-variable first-order logic. Technical report, 1997.

[12] J. van Heijenoort. *From Frege to Gödel - A source book in mathematical logic, 1879–1931.* Source books in the history of the sciences. Harvard University Press, 3^{rd} printing edition, 1977.

[13] M. K. Kwatinetz. *Problems of expressibility in finite languages.* PhD thesis, University of California, Berkeley, 1981.

[14] W. W. McCune. Otter 3.0 Reference manual and guide. Technical Report ANL-94/6, Argonne National Laboratory, 1994. (Revision A, august 1995).

[15] Ph. A. J. Noël. Experimenting with Isabelle in ZF set theory. *J. Automated Reasoning*, 10(1):15–58, 1993.

[16] E. G. Omodeo. Specifiche formali di proprietà di relazioni: esempi. Technical Report 36-97, Dip. Informatica e Sistemistica, Università *La Sapienza* di Roma, 1997.

[17] L. C. Paulson. Set Theory for verification: I. From foundations to functions. *J. Automated Reasoning*, 11(3):353–389, 1993.

[18] L. C. Paulson. Set Theory for verification. II: Induction and recursion. *J. Automated Reasoning*, 15(2):167–215, 1995.

[19] A. Quaife. Automated deduction in von Neumann-Bernays-Gödel set theory. *J. of Automated Reasoning*, 17(3):291-323, 1992.

[20] A. Quaife. *Automated development of fundamental mathematical theories.* Kluwer Academic Publishers, 1992.

[21] E. Schröder. *Vorlesungen über die Algebra der Logik (exakte Logik)*, volume 2, part 1. B. Teubner, Leipzig, 1891. [Reprinted by Chelsea Publishing Co., New York, 1966.].

[22] E. Schröder. *Vorlesungen über die Algebra der Logik (exakte Logik)*, volume 3, Algebra und Logik der Relative, part 1. B. Teubner, Leipzig, 1895. [Reprinted by Chelsea Publishing Co., New York, 1966.].

[23] A. Tarski. On the calculus of relations. *J. of Symbolic Logic*, 6(3):73–89, 1941.

[24] A. Tarski and S. Givant. *A formalization of Set Theory without variables*, volume 41 of *Colloquium Publications.* American Mathematical Society, 1987.

[25] A. N. Whitehead and B. Russell. *Principia Mathematica.* Cambridge University Press, 1910. Reprinted 1980.

[26] M. Wirsing. *Algebraic Specification..* In Handbook of Theoretical Computer Science, Vol. B: Formal Models and Semantics, pp. 675-788 The MIT Press, 1990.

[27] L. Wos. *Automated reasoning. 33 basic research problems.* Prentice Hall, 1988.