Three language extension mechanisms for map calculus

Authors:
Pasquale Caianiello
Stefania Costantini
Eugenio G. Omodeo
— Università di L'Aquila
Dipartimento di Informatica

Presented by: Andrea Formisano
— Università di Perugia
Dipartimento di Matematica ed Informatica

APPIA–GULP–PRODE 2002
Madrid, 16-18 sep 2002

Research related to COST action n.274 (TARSKI) http://tarski.org
and to MURST / MIUR 40% : *Aggregate- and number-reasoning* ···

A platform for map reasoning

Map calculus is a classical form of algebraic logic, admittedly not very readable, but perhaps better suited for automatic reasoning than more advanced systems of logic

- it is *devoid of variables*, and
- merely *equational*

A *platform* supporting map calculus will include, among others:

- Proof methods (current experimentation relies on *Otter*)
- Translators of man-oriented formalisms into map calculus

Translation platform component: *MetaMorpho*

Two approaches — two sucomponents

- Rewriting rules — *KataMorpho*

  This approach was exploited, e.g., to translate 1st-order sentences in 3 variables into map calculus

- Definitional extensions — *AnaMorpho*

  Three such mechanisms will be illustrated through examples in what follows

Roughly speaking, KataMorpho proceeds downwards, and AnaMorpho proceeds upwards

Both components are *generic*, because we are aware that map calculus is the first, not necessarily the best language for algebraic logic

(Cf. Haskell-based platform for heterogeneous relation algebras by Kahl and Schmidt)

An analog of map calculus — to convey the idea simply

-- **primitive operators for regular expressions:**

P^*	$=:$	P^*	-- Kleene star-operator
$P \cup Q$	$=:$	$P \cup Q$	-- union
$P \circ Q$	$=:$	$P \circ Q$	-- concatenation
0	$=:$	0	-- void language

-- **aliases and derived operators / relators:**

$\emptyset$	$=:$	0
ι	$=:$	$\emptyset^*$
P^+	$=:$	$P \circ P^*$
$P \subseteq Q$	$\leftrightarrow:$	$P \cup Q = Q$
$\bigcup([])$	$=:$	$\emptyset$
$\bigcup([P \mid Q])$	$=:$	$P \cup \bigcup(Q)$

-- **useful templates:**

semiGroup(P) **Θ:** [
$P(P(Q,R),S)=P(Q,P(R,S))$]
-- **: associative law**

leftMonoid(P,Q) **Θ:** [
semiGroup(P),
$P(Q,R)=R$]
-- **: left monoid**

monoid(P,Q) **Θ:** [
leftMonoid(P,Q),
$P(R,Q)=R$]
-- **: bilateral monoid**

commMonoid(P,Q) **Θ:** [
leftMonoid(P,Q),
$P(R,S)=P(S,R)$]
-- **: commutative monoid**

leftDistributes(P,Q) **Θ:** [
$P(R,Q(S,T))=Q(P(R,S),P(R,T))$]
-- **: left distributive law**

etc.

-- **logical axioms ...**

$$\mathsf{commMonoid}(\cup, \emptyset)$$

$$P \cup P = P$$

-- : idempotence

$$\mathsf{monoid}(\circ, \boldsymbol{\iota})$$

$$\_\circ\emptyset = \emptyset$$

$$\emptyset\circ\_ = \emptyset$$

$$\mathsf{leftDistributes}(\circ, \cup)$$

$$\mathsf{rightDistributes}(\circ, \cup)$$

$$\boldsymbol{\iota} \cup P \circ P^* = P^*$$

$$(\boldsymbol{\iota} \cup P)^* = P^*$$

-- **... a proper axiom ...**

$$P \subseteq (\bigcup([\mathsf{s}_1, \mathsf{s}_2, \mathsf{s}_2]))^*$$

-- **... and an inference rule**

$$[P \cup Q \circ R = Q] \Rightarrow P \circ R^* = Q$$

-- logical axioms as internally expanded:

$$P \cup Q \cup R = P \cup (Q \cup R)$$

$$\emptyset \cup P = P$$

$$P \cup Q = Q \cup P$$

$$P \cup P = P$$

$$P \circ Q \circ R = P \circ (Q \circ R)$$

$$\boldsymbol{\iota} \circ P = P$$

$$P \circ \boldsymbol{\iota} = P$$

$$\_ \circ \emptyset = \emptyset$$

$$\emptyset \circ \_ = \emptyset$$

$$P \circ (Q \cup R) = P \circ Q \cup P \circ R$$

$$(P \cup Q) \circ R = P \circ R \cup Q \circ R$$

$$\boldsymbol{\iota} \cup P \circ P^* = P^*$$

$$(\boldsymbol{\iota} \cup P)^* = P^*$$

$$[P \cup Q \circ R = Q] \Rightarrow P \circ R^* = Q$$

Map calculus – syntax

-- **Primitive constructors of map expressions:**

$\boldsymbol{\iota}$	$=:$	$\boldsymbol{\iota}$	-- diagonal map
$\overline{P}$	$=:$	$\overline{P}$	-- Boolean complementation
$P^{\smile}$	$=:$	$P^{\smile}$	-- Peircean operation of forming the converse
$P \cup Q$	$=:$	$P \cup Q$	-- Boolean join
$P \circ Q$	$=:$	$P \circ Q$	-- Peircean map-composition

-- **Secondary constructors of map expressions:**

$\boldsymbol{\delta}$	$=:$	$\overline{\boldsymbol{\iota}}$	-- difference map
$\mathbb{1}$	$=:$	$\boldsymbol{\iota} \cup \boldsymbol{\delta}$	-- top
$\emptyset$	$=:$	$\overline{\mathbb{1}}$	-- bottom
$P \cap Q$	$=:$	$\overline{\overline{P} \cup \overline{Q}}$	-- Boolean meet

-- **Further Boolean operators:**

$P - Q$	$=:$	$P \cap \overline{Q}$	-- map difference
$P \triangle Q$	$=:$	$\overline{P \cup \overline{Q}} \cup \overline{Q \cup \overline{P}}$	-- symmetric map difference

Map calculus – logical axioms

-- **Boolean axioms (Huntington–Robbins, 1933/1934):**

$$P \cup Q = Q \cup P$$

$$\mathsf{semiGroup}(\cup)$$

$$(P \cup Q) \cap (P \cup \overline{Q}) = P$$

-- **associative law, and unit element, for map composition:**

$$\mathsf{rightMonoid}(\circ, \boldsymbol{\iota})$$

-- distributivity of composition over union:

$$\mathsf{rightDistributes}(\circ, \cup)$$

-- **convolutory laws:**

$$P^{\smile\smile} = P$$

$$(P \cup Q)^{\smile} = P^{\smile} \cup Q^{\smile}$$

$$(P \circ Q)^{\smile} = Q^{\smile} \circ P^{\smile}$$

-- **Schröder's cycle inference rule:**

$$[P \circ Q \cap R = \emptyset] \Rightarrow P^{\smile} \circ R \cap Q = \emptyset$$

Map calculus – further derived constructs

$$
\begin{array}{rcl}
P \subseteq Q & \leftrightarrow: & \emptyset = P - Q \\
\mathsf{rA}(P) & =: & P \circ 1\!\!1 \\
\mathsf{Tot}(P) & \leftrightarrow: & \mathsf{rA}(P) = 1\!\!1 \\
\mathsf{Coll}(P) & \leftrightarrow: & P \subseteq \boldsymbol{\iota} \\
\mathsf{mult}(P) & =: & P \cap P \circ \boldsymbol{\delta} \\
\mathsf{isFunc}(P) & \leftrightarrow: & \mathsf{mult}(P) = \emptyset \\
\mathsf{Skolem}(P, Q, R) & \Theta: & [\ R =: Q, \quad R \subseteq P, \\
 & & \ \ \mathsf{isFunc}(R), \quad \mathsf{rA}(R) = \mathsf{rA}(P)] \\
\mathsf{areQProj}(L, R) & \Theta: & [\ \mathsf{isFunc}(R), \quad \mathsf{isFunc}(L), \\
 & & \ \ L^{\smile} \circ R = 1\!\!1]
\end{array}
$$

Predicates λ, ϱ which meet the abstract properties $\mathsf{areQProj}(\lambda, \varrho)$ are called *conjugated (quasi-) projections* and are the key for translating each sentence of ‘strong’ first-order theories into an equivalent equation of map calculus

Maddux' embedding of quantificational logic into map calculus

$$
\begin{array}{lll}
\mathsf{th}(L, R \,\|\, 1) & =: & L \\
\mathsf{th}(L, R \,\|\, i + 1) & =: & R \circ \mathsf{th}(L, R, i) \\
 & & \\
\mathsf{sibs}(L, R \,\|\, [\,]) & =: & \mathbb{1} \\
\mathsf{sibs}(L, R \,\|\, [\mathsf{v}_i | \overrightarrow{V}\,]) & =: & \mathsf{th}(L, R, i) \circ \mathsf{th}^{\smile}(L, R, i) \cap \mathsf{sibs}(L, R, \overrightarrow{V}\,) \\
 & & \\
\mathsf{mXpr}(L, R \,\|\, p(\mathsf{v}_i, \mathsf{v}_j)) & =: & (\mathsf{th}(L, R, i) \circ p \cap \mathsf{th}(L, R, j)) \circ \mathbb{1} \\
\mathsf{mXpr}(L, R \,\|\, \neg\varphi) & =: & \overline{\mathsf{mXpr}}(L, R, \varphi) \\
\mathsf{mXpr}(L, R \,\|\, \varphi \,\&\, \psi) & =: & \mathsf{mXpr}(L, R, \varphi) \cap \mathsf{mXpr}(L, R, \psi) \\
\mathsf{mXpr}(L, R \,\|\, \exists \overrightarrow{V} \; \varphi) & =: & \mathsf{sibs}(L, R, \mathsf{freeVars}(\exists \overrightarrow{V} \; \varphi)) \circ \mathsf{mXpr}(L, R, \varphi) \\
 & & \\
\mathsf{Maddux}(L, R \,\|\, \chi) & \Theta: & [\quad Tr(\chi) \;\leftrightarrow:\; \mathsf{Maddux}(L, R, \chi) \quad]
\end{array}
$$

$i, j = 1, 2, \ldots$, $\quad$ p map letter,

$\overrightarrow{V}$ variable-list, $\quad$ φ, ψ, χ first-order formulas

We assume $\mathsf{areQProj}(L, R)$, $\mathsf{Tot}(L)$, and $\mathsf{Tot}(L)$

Case-study: A rather weak set theory (first-order spec.)

(E) $\forall v\,(v \in X \leftrightarrow v \in Y) \;\rightarrow\; X = Y$

(N) $\exists z\,\forall v\,\neg\, v \in z$

(W) $\exists w\,\forall v\,(v \in w \;\leftrightarrow\; (v \in X \;\vee\; v = Y\,))$

(L) $\exists \ell\,\forall v\,(v \in \ell \;\leftrightarrow\; (v \in X \,\&\, \neg\, v = Y\,))$

(R) $\exists r\,((r \in X \;\vee\; r = X)\,\&\,\neg\exists v\,(v \in r \,\&\, v \in X))$

Case-study: A rather weak set theory (spec. in map calculus)

$\in \; =: \; \mathsf{p}_1 \qquad\qquad \ni \; =: \; \in^{\smile}$

$\mathfrak{F}(P) \; =: \; \overline{P \circ \overline{\in}} \cap \overline{\overline{P}^{\smile} \circ \in}$ -- **set-formation construct**

$\mathsf{Coll}(\mathfrak{F}(\ni))$ -- **extensionality axiom (E)**

$\in \boldsymbol{=} \mathbb{1} \circ (\in - \ni \circ \in)$ -- **regularity axiom (R)**

$\mathsf{Skolem}((\ni \cup \boldsymbol{\iota}) - \ni \circ \in, \mathsf{p}_2, \mathsf{arb})$ -- conservative extension of language

$\mathsf{funcPart}(P) \; =: \; P - P \circ \boldsymbol{\delta}$

$\mathsf{tot}(P) \; =: \; P \triangle (\boldsymbol{\iota} - \mathsf{rA}(P)) \qquad\qquad \mathsf{mix} \; =: \; \ni \circ \ni \cap \ni \circ \overline{\ni}$

$\lambda \; =: \; \mathsf{funcPart}(\mathsf{mix}) \qquad\qquad \rho \; =: \; (\ni \cap \overline{\mathsf{mix} \circ \boldsymbol{\delta}} \circ \in) \circ \lambda$

$\in \circ \lambda \boldsymbol{=} \mathbb{1}$ -- **axiom (NWL) of elementary sets**

$\mathsf{Maddux}(\mathsf{tot}(\lambda), \mathsf{tot}(\rho), \mathsf{setMaddux})$ -- this brings in full first-order notation

Another case-study: Entity-Relationship modeling

The same definitional machinery which enabled us to express first-order set theory within map calculus, was exploited to obtain a

translator of ER-models into map calculus

(collaboration with Ernst-Erich Doberkat, Un. Dortmund)