Multi-type universal algebra: categorical equivalence *

Yiwen Ding[1], Krishna Manoorkar[1], and Ni Wayan Switrayni[1]

1. Vrije Universiteit Amsterdam, The Netherlands

Multi-type algebras are special kinds of heterogeneous algebras [1] which can be represented as the corresponding single-type algebras and vice versa. The motivation for studying this kind of mathematical structure comes from algebraic proof theory. In [4], it is showed that the algorithm ALBA can be used to transform analytic inductive formulas into rules of display calculus which satisfy the premises of Belnap's cut-elimination theorem. However, there are many important logics whose axioms are not analytic inductive, and cut-free display calculus can not be provided for them by using ALBA directly. In order to settle this problem, the multi-type methodology is introduced which represents single-type axiomatizations as multi-type axiomatizations where axioms in the multi-type languages are all analytic inductive formulas. This methodology has been applied successfully to semi De Morgan logic [2], linear logic [5], logic of bilattices [3] and so on, whose axioms are not analytic inductive. In this paper, we introduce general definitions for both single-type and multi-type algebras and homomorphisms on them. We define a functor from the category of multi-type algebras to the category of single-type algebras and vice versa and show categorical equivalence of those two categories.

We first introduce the general definition of multi-type algebras and homomorphisms on them.

A *multi-type algebraic language* is a tuple $\mathfrak{L} = (\mathcal{L}_1, \mathcal{L}_2, h, e)$ where $\mathcal{L}_1$ and $\mathcal{L}_2$ are sets of function symbols and h and e are two unary function symbols. A *multi-type* $\mathfrak{L}$-algebra is a tuple $\mathfrak{A} = (\mathbf{A}, \mathbf{B}, h, e)$ such that: (1) $\mathbf{A}$ is an $\mathcal{L}_1$-algebra and $\mathbf{B}$ is an $\mathcal{L}_2$-algebra. (2) $h : \mathbf{A} \to \mathbf{B}$ is a unary surjective map and $e : \mathbf{B} \to \mathbf{A}$ is a unary injective map. (3) For any $\alpha_1, \cdots, \alpha_n \in \mathbf{B}$ and n-ary $f \in \mathcal{L}_1 \cap \mathcal{L}_2$, $f^{\mathbf{B}}(\alpha_1, \cdots, \alpha_n) = hf^{\mathbf{A}}(e(\alpha_1), \cdots, e(\alpha_n))$. Let $\mathfrak{A}_1 = (\mathbf{A}_1, \mathbf{B}_1, h_1, e_1)$ and $\mathfrak{A}_2 = (\mathbf{A}_2, \mathbf{B}_2, h_2, e_2)$ be any multi-type $\mathfrak{L}$-algebras. A *multi-type homomorphism* between $\mathfrak{A}_1$ and $\mathfrak{A}_2$ is a pair $F = (F^{\mathbf{A}}, F^{\mathbf{B}})$ of maps such that: (1) $F^{\mathbf{A}} : \mathbf{A}_1 \to \mathbf{A}_2$ and $F^{\mathbf{B}} : \mathbf{B}_1 \to \mathbf{B}_2$ are homomorphisms. (2) $F^{\mathbf{B}} \circ h_1 = h_2 \circ F^{\mathbf{A}}$ and $F^{\mathbf{A}} \circ e_1 = e_2 \circ F^{\mathbf{B}}$.

Now we are ready to define the single-type representation of multi-type algebras and the multi-type representation of single-type algebras.

Definition 1. *Let $\mathfrak{L} = (\mathcal{L}_1, \mathcal{L}_2, h, e)$ be a multi-type algebraic language and $\mathcal{L}_3 = (\mathcal{L}_2 - \mathcal{L}_1)$, the* single-type representation *of $\mathfrak{L}$ is $\mathfrak{L}_+ = \mathcal{L}_1 \cup \mathcal{L}_3 \cup \{e \circ h\}$. Given a multi-type $\mathfrak{L}$-algebra $\mathfrak{A} = (\mathbf{A}, \mathbf{B}, h, e)$, the* single-type representation *of $\mathfrak{A}$ is $\mathfrak{A}_+ = (\mathbf{A}, \{f^{\mathfrak{A}_+} \mid f \in \mathcal{L}_3\}, e \circ h)$ such that for any n-ary $f_i \in \mathcal{L}_3$ and $a, a_1, \cdots, a_n \in A$, $f_i^{\mathfrak{A}_+}(a_1, \cdots, a_n) := ef_i^{\mathbf{B}}(h(a_1), \cdots, h(a_1))$. Let $\mathfrak{A}_1 = (\mathbf{A}_1, \mathbf{B}_1, h, e)$ and $\mathfrak{A}_2 = (\mathbf{A}_2, \mathbf{B}_2, h, e)$ be any multi-type $\mathfrak{L}$-algebras and $F = (F^{\mathbf{A}}, F^{\mathbf{B}}) : \mathfrak{A}_1 \to \mathfrak{A}_2$ be any multi-type homomorphism, the* single-type representation *of F is $F_+ : \mathfrak{A}_{1+} \to \mathfrak{A}_{2+}$ defined by $F_+(a) := F^{\mathbf{A}}(a)$ for any $a \in A_1$.*

Given a single-type algebra, there are different ways of representing it as a multi-type algebra which depends on which operators we want to keep, to destroy or to rebuild on kernels. In order to divide our algebraic language properly, we introduce the notion of parameter.

Definition 2. *Let $\mathcal{L}$ be an algebraic language. A* parameter *on $\mathcal{L}$ is a tuple $\mathsf{P} = (\mathcal{L}_1, \mathcal{L}_2, \sigma)$ such that $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2 \cup \{\sigma\}$ and $\mathcal{L}_1$, $\mathcal{L}_2$ and $\{\sigma\}$ are pairwise disjoint. An $\mathcal{L}$-algebra $\mathbf{C}$ is called*

*Yiwen Ding is supported by the China Scholarship Council. Krishna Manoorkar is supported by the NWO grant KIVI.2019.001. Ni Wayan Switrayni is supported by the Indonesian Education Scholarship with Ref. Number: 1027/J5.2.3/BPI.LG/VIII/2022.

P-algebra *if for any* $a, a_1, \cdots, a_n \in \mathbf{C}$ *and n-ary* $f \in \mathcal{L}$, $\sigma\sigma(a) = a$ *and* $\sigma f(\sigma(a_1), \cdots, \sigma(a_n)) = f(a_1, \cdots, a_n)$.

Now we are ready to give the multi-type representation of a single-type algebra relative to a parameter.

Definition 3. *Let* $\mathcal{L}$ *be an algebraic language,* $\mathsf{P} = (\mathcal{L}_1, \mathcal{L}_2, \sigma)$ *be a parameter on* $\mathcal{L}$ *and* $\mathbf{C}$ *be any* P*-algebra. The* multi-type representation *of* $\mathcal{L}$ *relative to* P *is* $\mathcal{L}^+ = (\mathcal{L}_1, \mathcal{L}_2, h, e)$. *The* multi-type representation *of* $\mathbf{C}$ *is* $\mathbf{C}^+ = (\mathbf{A}, \mathbf{B}, h, e)$ *such that: (1)* $\mathbf{A}$ *is the* $\mathcal{L}_1$*-reduct of* $\mathbf{C}$. *(2)* $h : \mathbf{A} \to \mathbf{B}$ *and* $e : \mathbf{B} \to \mathbf{A}$ *such that for any* $a \in A$ *and* $\alpha \in \sigma(A)$, $h(a) := \sigma(a)$ *and* $e(\alpha) := \alpha$. *(3)* $\mathbf{B} = (\sigma[A], \{f^{\mathbf{B}} \mid f \in \mathcal{L}\})$ *such that for any n-ary* $f \in \mathcal{L}$ *and* $\alpha_1, ...\alpha_n \in \sigma[A]$, $f^{\mathbf{B}}(\alpha_1, ..., \alpha_n) := hf^{\mathbf{C}}(e(\alpha_1), ..., e(\alpha_n))$. *Let* $\mathbf{C}_1, \mathbf{C}_2$ *be any* $\mathcal{L}$*-algberas and* $F : \mathbf{C}_1 \to \mathbf{C}_2$ *be any homomorphism. Let* ${\mathbf{C}_i}^+ = (\mathbf{A}_i, \mathbf{B}_i, h_i, e_i)$ *for* $i \in \{1, 2\}$. The multi-type representation *of* F *is* $F^+ = (F^{\mathbf{A}}, F^{\mathbf{B}})$ *from* ${\mathbf{C}_1}^+$ *to* ${\mathbf{C}_2}^+$, *where* $F^{\mathbf{A}} : \mathbf{A}_1 \to \mathbf{A}_2$ *and* $F^{\mathbf{B}} : \mathbf{B}_1 \to \mathbf{B}_2$ *such that for any* $a \in \mathbf{A}_1$ *and* $\alpha \in \mathbf{B}_1$, $F^{\mathbf{A}}(a) := F(a)$ *and* $F^{\mathbf{B}}(\alpha) := h_2 \circ F \circ e_1(\alpha)$.

Now we are ready to establish the categorical equivalence between single-type and multi-type algebras.

Given a multi-type algebraic language $\mathfrak{L} = (\mathcal{L}_1, \mathcal{L}_2, h, e)$, $\mathsf{Cat}_{\mathfrak{L}}$ denotes the category of multi-type $\mathfrak{L}$-algebras and multi-type homomorphisms between them, and $\mathsf{Cat}_{\mathfrak{L}_+}$ denotes the category of P-algebras and homomorphisms between them, where $\mathsf{P} = (\mathcal{L}_1, \mathcal{L}_2, e \circ h)$ is the parameter on algebraic language $\mathfrak{L}_+$. $\mathsf{S}_{\mathfrak{L}} : \mathsf{Cat}_{\mathfrak{L}} \to \mathsf{Cat}_{\mathfrak{L}_+}$ is a functor such that for any $\mathfrak{A}, F \in \mathsf{Cat}_{\mathfrak{L}}$, $\mathsf{S}_{\mathfrak{L}}(\mathfrak{A}) := \mathfrak{A}_+$ and $\mathsf{S}_{\mathfrak{L}}(F) := F_+$. $\mathsf{M}_{\mathfrak{L}} : \mathsf{Cat}_{\mathfrak{L}_+} \to \mathsf{Cat}_{\mathfrak{L}}$ is a functor such that for any $\mathbf{C}, F \in \mathsf{Cat}_{\mathfrak{L}_+}$, $\mathsf{M}_{\mathfrak{L}}(\mathbf{C}) := \mathbf{C}^+$ and $\mathsf{M}_{\mathfrak{L}}(F) := F^+$.

Given an algebraic language $\mathcal{L}$ and a parameter $\mathsf{P} = (\mathcal{L}_1, \mathcal{L}_2, \sigma)$ on $\mathcal{L}$. $\mathsf{Cat}_{\mathcal{L}}$ denotes the category of P-algebras and homomorphisms between them, and $\mathsf{Cat}_{\mathcal{L}^+}$ denotes the category of multi-type $\mathcal{L}^+$-algebras and multi-type homomorphisms between them. $\mathsf{M}_{\mathcal{L}} : \mathsf{Cat}_{\mathcal{L}} \to \mathsf{Cat}_{\mathcal{L}^+}$ is a functor such that for any $\mathbf{C}, F \in \mathsf{Cat}_{\mathcal{L}}$, $\mathsf{M}_{\mathcal{L}}(\mathbf{C}) := \mathbf{C}^+$ and $\mathsf{M}_{\mathcal{L}}(F) := F^+$. $\mathsf{S}_{\mathcal{L}} : \mathsf{Cat}_{\mathcal{L}^+} \to \mathsf{Cat}_{\mathcal{L}}$ is a functor such that for any $\mathfrak{A}, F \in \mathsf{Cat}_{\mathcal{L}^+}$, $\mathsf{S}_{\mathcal{L}}(\mathfrak{A}) := \mathfrak{A}_+$ and $\mathsf{S}_{\mathcal{L}}(F) := F_+$.

According to definitions above, we can prove categorical equivalence between category of single-type algebras and category of multi-type algebras as stated the following theorem.

Theorem 1. *Let* $\mathfrak{L}$ *be a multi-type algebraic language, then* $\mathsf{S}_{\mathfrak{L}} : \mathsf{Cat}_{\mathfrak{L}} \to \mathsf{Cat}_{\mathfrak{L}_+}$ *and* $\mathsf{M}_{\mathfrak{L}} : \mathsf{Cat}_{\mathfrak{L}_+} \to \mathsf{Cat}_{\mathfrak{L}}$ *forms a categorical equivalence between them. Let* $\mathcal{L}$ *be an algebraic language and* P *be a parameter on* $\mathcal{L}$, *then* $\mathsf{M}_{\mathcal{L}} : \mathsf{Cat}_{\mathcal{L}} \to \mathsf{Cat}_{\mathcal{L}^+}$ *and* $\mathsf{S}_{\mathcal{L}} : \mathsf{Cat}_{\mathcal{L}^+} \to \mathsf{Cat}_{\mathcal{L}}$ *forms a categorical equivalence between them.*

References

[1] Garrett Birkhoff and John D Lipson. Heterogeneous algebras. *Journal of Combinatorial Theory*, 8(1):115–133, 1970.

[2] Giuseppe Greco, Fei Liang, M Andrew Moshier, and Alessandra Palmigiano. Semi de morgan logic properly displayed. *Studia logica*, 109:1–45, 2021.

[3] Giuseppe Greco, Fei Liang, Alessandra Palmigiano, and Umberto Rivieccio. Bilattice logic properly displayed. *Fuzzy Sets and Systems*, 363:138–155, 2019.

[4] Giuseppe Greco, Minghui Ma, Alessandra Palmigiano, Apostolos Tzimoulis, and Zhiguang Zhao. Unified correspondence as a proof-theoretic tool. *Journal of Logic and Computation*, 28(7), 2018.

[5] Giuseppe Greco and Alessandra Palmigiano. Linear logic properly displayed. *ACM Transactions on Computational Logic*, 24(2):1–56, 2023.