Resource combinatory algebras

A. Carraro[°†] (with T. Ehrhard[†] and A. Salibra[°])

[°]Università Ca'Foscari Venezia, Italy
[†]Université Paris-Diderot, France

Motivations

- Motivation: algebraic framework for resource λ-calculus, as was done for ordinary λ-calculus. Algebraic approach has proved fruitful so far.

- Why resource λ-calculus:
  - is connected to (differential) linear logic and differential λ-calculus (Ehrhard-Regnier)
  - can be seen as a "linear decomposition" of lambda calculus: from models of resource λ-calculus we can recover models of ordinary λ-calculus
  - we look for a definition of model of resource λ-calculus; so far just examples in a categorical setting
  - nice mix of linear algebra and combinatory logic

Overview: algebras for ordinary vs resource λ-calculi

"ordinary" case: Binary applicative structures + extra axioms/structure

- combinatory algebras (CAs, Schönfinkel & Curry),
- lambda algebras (LAs, Curry),
- lambda abstraction algebras (LAAs, Pigozzi & Salibra).

"resource" case: Bag-applicative structures + extra axioms/structure

- resource combinatory algebras (RCAs),
- resource lambda algebras (RLAs),
- resource lambda abstraction algebras (RLAAs).

PART I - THE ORDINARY WORLD

Lambda Abstraction Algebras (Pigozzi & Salibra)

- Lambda Abstraction Algebras (LAAs) = algebraic version of ordinary λ-calculus
- variables of λ-calculus $\Rightarrow$ constants in LAAs
- λ-terms $\Rightarrow$ ground terms in the signature of LAAs
- similarity type: $\{\cdot, \lambda x, x\}_{x \in V}$, where "$\cdot$" is binary operator (application) and for each $x \in V$, λx is a unary operator (abstraction) and x is a nullary operator (λ-variable)
- example: $(\lambda y.x)x$ is a term

Lambda Abstraction Algebras (Pigozzi & Salibra)

- **Lambda Abstraction Algebra** (LAA): $\mathbf{A} = (A, \cdot, \lambda x, x)_{a \in V}$

(β1) $(\lambda x.x)a = a$

(β2) $(\lambda x.y)a = y$ $\Rightarrow$ y does not depend on x, a is discarded

(β3) $(\lambda x.a)x = a$

(β4) $(\lambda x.\lambda x.a)b = \lambda x.a$ $\Rightarrow$ $\lambda x.a$ does not depend on x, b is discarded

(β5) $(\lambda x.ab)c = (\lambda x.a)c((\lambda x.b)c)$ $\Rightarrow$ c is duplicated

(β6) $(\lambda x.\lambda y.a)b = \lambda y.(\lambda x.a)b$, if $(\lambda y.b)x = b$

(α) $\lambda x.b = \lambda y.(\lambda x.b)y$, if $(\lambda y.b)x = b$

... $(\lambda y.b)x = b \Rightarrow$ algebraic way of saying that b does not depend on x

Theorem 1 *The λ-calculus is the free algebra on the empty set of generators in the variety of* LAAs.

Combinatory Algebras (Schönfinkel & Curry)

- **Combinatory Algebra** (CA): $\mathbf{A} = (A, \cdot, K, S)$

$(C1)$ $Kxy = x$

$(C2)$ $Sxyz = xz(yz)$

- corresponding equational logic: Combinatory Logic (CL)
- simulation of abstraction and classical substitution for variables:

$$(\lambda^* x.t)s = t\{x := s\}$$

- failure of weak extensionality: $CL \vdash t = s \not\Rightarrow CL \vdash \lambda^* x.t = \lambda^* x.s$
- λ-calculus is "stronger" than CL
- **Lambda Algebra** (LA): Combinatory Algebra $+$ Curry's axioms $=$ better correspondence with λ-calculus.

PART II - THE LINEAR WORLD

Bag-applicative Structures

- **Bag-applicative structure**: $\mathbf{A} = (A, \vee, 0, \cdot_n)_{n\in\mathbb{N}}$
- infinitely many $n+1$-ary applications $\_\cdot_n[\_,\ldots,\_]$, a join operation $\vee$ with neutral element 0
- each application is commutative in the last n arguments:

  example: $x \cdot_3 [y_0, y_1, y_2] = x \cdot_3 [y_1, y_2, y_0]$
- last n arguments of a $n+1$-ary application $=$ *bag*
- $x \cdot_n \underbrace{[y_0, \ldots, y_{n-1}]}_{\text{bag}}$ may be written $x[y_0, \ldots, y_{n-1}]$ or even $x\bar{y}$; $x[y]$ is abbreviated with xy
- each application is linear in every argument:

$$x[\ldots, 0, \ldots] = 0 \qquad 0[\ldots] = 0$$
$$(x \vee y)[\ldots] = x[\ldots] \vee y[\ldots] \qquad x[\ldots, y \vee z, \ldots] = x[\ldots, y, \ldots] \vee x[\ldots, z, \ldots]$$

Resource Lambda Abstraction Algebras

- Resource Lambda Abstraction Algebras (RLAAs) = algebraic version of resource λ-calculus
- variables of resource λ-calculus $\Rightarrow$ constants in RLAAs and
- resource λ-terms $\Rightarrow$ ground terms in the signature of RLAAs
- similarity type: $\{\cdot_n, \lambda x, x\}_{x \in V, n \in \mathbb{N}}$, where each "$\cdot_n$" is a $n+1$-ary operator (application) and for each $x \in V$, λx is a unary operator (abstraction) and x is a nullary operator (λ-variable)
- example: $(\lambda y.x)[x, x]$ is a term

Resource Lambda Abstraction Algebras

- **Resource Lambda Abstraction Algebra** (RLAA): $\mathbf{A} = (A, \vee, 0, \cdot_n, \lambda x, x)_{x \in V, n \in \mathbb{N}}$
- $(A, \vee, 0, \cdot_n)_{n \in \mathbb{N}}$ bag-applicative structure

$(r\beta_1)$ $(\lambda x.x)\bar{a} = \begin{cases} a_0 & \text{if } |\bar{a}| = 1 \\ 0 & \text{otherwise} \end{cases}$ $\Rightarrow$ linear identity, just one argument

$(r\beta_2)$ $(\lambda x.y)\bar{a} = \begin{cases} y & \text{if } |\bar{a}| = 0 \\ 0 & \text{otherwise} \end{cases}$ $\Rightarrow$ y does not depend on x

$(r\beta_3)$ $(\lambda x.\lambda x.a)\bar{b} = \begin{cases} \lambda x.a & \text{if } |\bar{b}| = 0 \\ 0 & \text{otherwise} \end{cases}$ $\Rightarrow$ $\lambda x.a$ does not depend on x

$(r\beta_4)$ ...

$(r\beta_5)$ $(\lambda x.a\bar{b})\bar{c} = \bigvee_{\bar{\bar{d}} \in \mathcal{Q}_{\bar{c},k+1}} (\lambda x.a)\bar{d}_0[(\lambda x.b_0)\bar{d}_1, \ldots, (\lambda x.b_{k-1})\bar{d}_k]$ $(|\bar{b}| = k)$

$\Rightarrow$ "break" $\bar{c}$ in all possible sensible ways

... no dumping nor duplication of arguments $\Rightarrow$ linearity

Resource Lambda Abstraction Algebras

$(r\alpha)$ $(\lambda x.a)x^k = a,\ (\lambda y.a)[\ ] = a \ \Rightarrow\ \lambda x.a = \lambda y.(\lambda x.a)y^k$

$(r\gamma)$ $(\lambda x.a)x^n \vee a = a$

$(r\delta)$ $(\lambda x.(\lambda x.a)x^n)x^n = (\lambda x.a)x^n$

$(r\lambda)$ $\lambda x.0 = 0; \quad \lambda x.(a \vee b) = \lambda x.a \vee \lambda x.b.$

locally finite RLAA: each element depends on a finite subset of V

Theorem 2 *The resource λ-calculus is the free algebra on the empty set of generators in the variety of* RLAA*s.*

Resource Lambda Abstraction Algebras

- $(\lambda y.a)y^n = a \Rightarrow a$ does depend on y exactly n times
- $(\lambda y.a)[\ ] = a \Rightarrow a$ does not depend on y

Recall:

$$(r\beta_3)\ \ (\lambda x\lambda x.a)\bar{b} = \begin{cases} \lambda x.a & \text{if } |\bar{b}| = 0 \\ 0 & \text{otherwise} \end{cases}$$

$$(r\beta_5)\ \ (\lambda x.a\bar{b})\bar{c} = \bigvee_{\bar{\bar{d}} \in \mathcal{Q}_{\bar{c},k+1}} (\lambda x.a)\bar{d}_0[(\lambda x.b_0)\bar{d}_1, \ldots, (\lambda x.b_{k-1})\bar{d}_k]\ \ (|\bar{b}| = k)$$

Consequence: $(\lambda y.a)y^n = a$ implies $(\lambda y.a)y^k = 0$ for all $k \neq n$.

$$\begin{aligned} (\lambda y.(\lambda y.a)y^n)y^k &= \bigvee_{p_0+\cdots+p_n=k} (\lambda y.\lambda y.a)y^{p_0}[(\lambda y.y)y^{p_1}, \ldots, (\lambda y.y)y^{p_n}] \\ &= 0 \end{aligned}$$

p_0 must be 0 $\quad$ $p_1, \ldots, p_n$ must be 1

Resource Combinatory Algebras

- **Resource Combinatory Algebra** (RCA): $\mathbf{A} = (A, \vee, 0, \cdot_n, K, S_{\bar{p}})_{n\in\mathbb{N},\bar{p}\in\mathbb{N}^*}$
- $(A, \vee, 0, \cdot_n)_{n\in\mathbb{N}}$ bag-applicative structure
- Equation scheme for K:

$$K\bar{x}\bar{y} = \begin{cases} x_0 & \text{if } |\bar{x}| = 1,\ |\bar{y}| = 0 \\ 0 & \text{otherwise} \end{cases}$$

- examples:

$$K[x, y][z] = 0 \qquad K[x, y][\,] = 0 \qquad K[K][\,] = K$$

- no erasure, K can only discard empty bags

Resource Combinatory Algebras

- Equation scheme for $S_{\bar{n}}$:

$$S_{\bar{n}}\bar{x}\bar{y}\bar{z} = \begin{cases} \bigvee_{\bar{\bar{c}}\in\mathcal{Q}_{\bar{z},\bar{n}}} x_0\bar{c}_0[y_0\bar{c}_1,\ldots,y_{k-1}\bar{c}_k] & \text{if } |\bar{x}|=1,\ k=|\bar{y}|=|\bar{n}|-1,\ |\bar{z}|=\Sigma\bar{n} \\ 0 & \text{otherwise} \end{cases}$$

- examples:

$$\overbrace{S_{(0,1,2)}}^{\text{length 3}}[x]\overbrace{[y_1,y_2]}^{\text{length 2}}[z_1,z_2,z_3] \quad = \quad x[\,][y_1[z_1],y_2[z_2,z_3]]$$
$$\vee \quad x[\,][y_1[z_2],y_2[z_1,z_3]]$$
$$\vee \quad x[\,][y_1[z_3],y_2[z_1,z_2]]$$

the first bag must be of cardinality 1, the second bag must be of cardinality $(\sharp(0,1,2))-1$, the third bag must be of cardinality $0+1+2$

$$S_{(0,1,2)}[x][y_1][z_1,z_2,z_3] = 0$$

if one of the conditions above is not satisfied, the result is 0

Monomials/polynomials

- *monomials* with indeterminates in V and parameters in A:

$$t ::= x \mid c_a \mid K \mid S_{\bar{n}} \mid t_0[t_1, \ldots t_k] \qquad (\bar{n} \in \mathbb{N}^*,\ k \in \mathbb{N},\ a \in A)$$

- *polynomials* may also contain $\vee, 0$

- every polynomial has a *canonical form* (finite join of monomials)

- Next step: use combinators to define an abstraction algorithm for simulation of linear substitution of terms and classical substitution of 0 for indeterminates in monomials

Linear substitution ...

$t\langle x := s\rangle$ is the linear substitution of s for x in t:

- $y\langle x := s\rangle = \begin{cases} s & \text{if } x \equiv y \\ 0 & \text{otherwise} \end{cases}$
- $(p[q_1, \ldots, q_n])\langle x := s\rangle = p\langle x := s\rangle[q_1, \ldots, q_n] \vee \bigvee_{i=1}^{n} p[\ldots, q_i\langle x := s\rangle, \ldots]$
- $(p \vee q)\langle x := s\rangle = p\langle x := s\rangle \vee q\langle x := s\rangle$

examples:

- $(x[y, x])\langle x := s_1\rangle = s_1[y, x] \vee x[y, s_1]$
- $(x[y, x])\langle x := s_1\rangle\langle x := s_2\rangle = s_1[y, s_2] \vee s_2[y, s_1]$
- $(x[y, x])\langle x := s_1\rangle\langle x := s_2\rangle\langle x := s_3\rangle = 0$

... and classical substitution of zero

$t\{x := 0\}$ is classical substitution of 0 for x in t

examples:

- $(x[y, x])\langle x := s_1\rangle\{x := 0\} = s_1[y, 0] \vee 0[y, s_1] = 0$, if x does not occur in s_1
- $(x[y, x])\langle x := s_1\rangle\langle x := s_2\rangle\{x := 0\} = s_1[y, s_2] \vee s_2[y, s_1]$, if x does not occur in s_1, s_2

Hence for a monomial t

$$t\langle x := s_1\rangle \cdots \langle x := s_n\rangle\{x := 0\} = \begin{cases} 0 \text{ if } x \text{ does not occur } n \text{ times in } t \\ \text{a join of polynomials resulting by substitution} \\ \text{of occurrences of } x \text{ by a permutation of the } s_i\text{'s} \end{cases}$$

Abstraction of variables

A useful combinator:

- linear identity: $I \equiv S_{(1)}K[\ ]$ satisfies

$$I\bar{x} = \begin{cases} x_0 & \text{if } |\bar{x}| = 1 \\ 0 & \text{otherwise} \end{cases}$$

Abstraction on monomials

(i) $\lambda^* x.t \equiv Kt$ if $deg_x(t) = 0$

(ii) $\lambda^* x.x \equiv I$

(iii) $\lambda^* x.t_0[t_1, \ldots, t_k] \equiv S_{\bar{n}}[\lambda^* x.t_0][\lambda^* x.t_1, \ldots, \lambda^* x.t_k]$

with $\bar{n} = (deg_x(t_0), \ldots, deg_x(t_k))$ and $\exists i.\ deg_x(t_i) \neq 0$.

(iv) ... and on polynomials: $\lambda^* x. \bigvee_{i=1}^n t_i = \bigvee_{i=1}^n \lambda^* x.t_i$.

Simulation of linear substitution

Theorem 3 *For any monomial t and polynomials $s_1, \ldots, s_n$*

$$(\lambda^* x.t)[s_1, \ldots, s_n] = t\langle x := s_1\rangle \cdots \langle x := s_n\rangle\{x := 0\}$$

- axioms of bag-applicative structures $+$
  axiom schemes for the combinators $S_{\bar{n}}, K$ $+$
  rules of equational calculus $=$
  Resource Combinatory Logic (RCL)

- weak extensionality fails in RCL:

$$RCL \vdash t = s \;\not\Rightarrow\; RCL \vdash \lambda^* x.t = \lambda^* x.s$$

- equivalently, the implication $t = s \;\Rightarrow\; \lambda^* x.t = \lambda^* x.s$ fails in some RCA: for example $I[x] = x$, but not $S_{(0,1)}[KI][I] = I$ in the free RCA over one indeterminate x

Combinatory reducts

- Recall RCA similarity type: $(\vee, 0, \cdot_n, K, S_{\bar{p}})_{n\in\mathbb{N},\bar{p}\in\mathbb{N}^*}$
- combinatory reduct $\mathrm{Cr}\,\mathbf{A}$ of a RLAA $\mathbf{A}$

$$\mathrm{Cr}\,\mathbf{A} = (A, \vee, 0, \cdot_n, \lambda xy.x, \lambda xyz.xz^{p_0}[yz^{p_1}, \ldots, yz^{p_n}])_{(p_0,\ldots,p_n)\in\mathbb{N}^*,\ n\in\mathbb{N}}$$

Theorem 4 *If* $\mathbf{A}$ *is a locally finite* RLAA, *then* $\mathrm{Cr}\,\mathbf{A}$ *is a* RCA.

- ... but $\mathrm{Cr}\,\mathbf{A}$ is something more than just an RCA
- additional properties related to weak extensionality
- there exists a set of axioms schemes which characterizes RCAs of the form $\mathrm{Cr}\,\mathbf{A}$.

Homogenizers

A useful family of combinators:

- n-homogenizer: $H_n \equiv S_{(0,n)}[KI]$ satisfies

$$H_n\bar{x}\bar{y} = \begin{cases} x_0\bar{y} & \text{if } |\bar{x}| = 1 \text{ and } |\bar{y}| = n \\ 0 & \text{otherwise} \end{cases}$$

- H_n corresponds to $\lambda xy.xy^n$ of RLAAs

Resource Lambda Algebras

- **Resource Lambda Algebra** (RLA): a RCA satisfying

(R0) $H_n[H_m x] = \begin{cases} H_m x & \text{if } n = m \\ 0 & \text{otherwise} \end{cases}$

(R1) $K = H_1 K; \quad Kx = H_0[Kx]$

(R2) $S_{\bar{n}} = H_1 S_{\bar{n}}; \quad S_{\bar{n}} x = H_{|\bar{n}|-1}[S_{\bar{n}} x]; \quad S_{\bar{n}} x \bar{y} = H_{\Sigma \bar{n}}[S_{\bar{n}} x \bar{y}]$

(R3) ...

(R4) ...

(R5) ...

(R6) ...

... axioms look complicated (we omitted the last four ones), but are related to those for LAs ...

Homogenizers

RLAs internalize the notion of degree

- the elements of the form $H_n a$ are the semantical counterpart of monomials of the form $\lambda^* x.t$, with $deg_x(t) = n$
- for any monomial t and RLA $\mathbf{A}$

$$\mathbf{A} \vDash H_n[\lambda^* x.t] = \begin{cases} \lambda^* x.t & \text{if } n = deg_x(t) \\ 0 & \text{otherwise} \end{cases}$$

- $a \in A$ is *homogeneous of degree* n iff $H_n a = a$

Resource Lambda Algebras

Theorem 5 *Let* $\mathbf{A}$ *be an* RLA. *Then*

(i) the free extension $\mathbf{A}[V]$ *of* $\mathbf{A}$ *by* V *in the variety* RCA *satisfies weak extensionality:*

$$\mathbf{A}[V] \models t = s \;\Rightarrow\; \mathbf{A}[V] \vdash \lambda^* x.t = \lambda^* x.s$$

(ii) $\mathbf{A}[V] = (A[V], \vee, 0, \cdot_n, \lambda^* x, x)_{x \in V, n \in \mathbb{N}}$ *is a locally finite* RLAA

Theorem 6 *(i) If* $\mathbf{A}$ *is a locally finite* RLAA, *then* $Cr\mathbf{A}$ *is a* RLA.

(ii) An RCA $\mathbf{A}$ *is an* RLA *iff* $\mathbf{A}$ *is embeddable into the combinatory reduct* $\mathrm{Cr}\,\mathbf{B}$ *of some* RLAA $\mathbf{B}$.

this says that the category of locally finite RLAAs is equivalent to the category of RLAs

An example of RLA

- D set with injection $\to$: $\mathcal{M}_{\mathsf{f}}(D) \times D \to D$
- $\mathbf{D} = (\mathcal{P}(D), \cup, \emptyset, \cdot_n, K, S_{\bar{k}})_{n\in\mathbb{N},\ \bar{k}\in\mathbb{N}^*}$

  $K = \{[\alpha] \to [\ ] \to \alpha : \alpha \in D\}$,

  $S_{\bar{k}} = \{[p_0 \to [\beta_1, \ldots, \beta_n] \to \beta_0] \to [p_1 \to \beta_1, \ldots, p_n \to \beta_n] \to (\uplus_{i=0}^{n} p_i) \to \beta_0 :$

  $\beta_i \in D,\ p_i \in \mathcal{M}_{\mathsf{f}}(D),\ \sharp p_i = k_i, |\bar{k}| = n+1\}$

  $$\gamma[\beta_1, \ldots, \beta_n] = \begin{cases} \alpha & \text{if } \gamma = [\beta_1, \ldots, \beta_n] \to \alpha \\ \emptyset & \text{otherwise} \end{cases}$$

(the operations are extension by linearity of the above functions defined over the singletons)

$\mathbf{D}$ is a RCA, in fact also a RLA... why? It is sufficient to embed $\mathbf{D}$ into the combinatory reduct of a suitable RLAA $\mathbf{E}$, which we are going to define.

An example of RLAA

- $\mathcal{M}_{\mathsf{f}}(D)^{(V)} = \{\rho : V \to \mathcal{M}_{\mathsf{f}}(D) : \rho(x) = [\,] \text{ for cofinitely many } x \in V\}$ (set of environments)
- $\bot \Rightarrow$ environment which maps each x to the empty multiset
- $\rho\{x := m\} \Rightarrow$ update of environment at x with multiset m

Definition 1 $\mathbf{E} = (\mathcal{P}(\mathcal{M}_{\mathsf{f}}(D)^{(V)} \times D), \cup, \emptyset, \cdot_k, \lambda x^{\mathbf{E}}, x^{\mathbf{E}})_{x \in V, k \in \mathbb{N}}$

$$x^{\mathbf{E}} = \{(\bot\{x := [\alpha]\}, \alpha) : \alpha \in D\}$$

$$\lambda x^{\mathbf{E}}.(\rho, \alpha) = (\rho\{x := [\,]\}, \rho(x) \to \alpha)$$

$$(\rho_0, \alpha_0)[(\rho_1, \alpha_1) \ldots, (\rho_n, \alpha_n)] = \begin{cases} (\uplus_{i=0}^{n} \rho_i, \alpha) & \textit{if } \alpha_0 = [\alpha_1, \ldots, \alpha_n] \to \alpha \\ \emptyset & \textit{otherwise} \end{cases}$$

$\mathbf{E}$ is a RLAA and the map $h : \mathcal{P}(D) \to \mathrm{Cr}\, E$, defined by $h(X) = \{(\bot, \alpha) : \alpha \in X\}$ is an embedding from $\mathbf{D}$ into $\mathrm{Cr}\, \mathbf{E}$.

Graded algebras over RLAs

Stratify a RLA $\mathbf{A}$ by means of homogenizers (notion of degree)

$$B_n = \{a \in A : H_n a = a\}$$

Obtain a graduation in the usual (Bourbaki) sense:

- $(B_n, \vee, 0)$ join sub-semilattice of $(A, \vee, 0)$
- $n \neq m$ implies $B_n \cap B_m = \{0\}$
- suitably define combinators, belonging to B_0
- each $k+1$-ary application is the extension by linearity of a suitably defined operation which takes $k+1$ elements of degree $d_0, \ldots, d_k$ to an element of degree $d_0 + \cdots + d_k$
- define $\mathbf{B}$ as the direct sum of the B_ns

Description of free extensions of RLAs

Lemma 1 *The $\mathbb{N}$-graded algebra $\mathbf{B}$ is the free extension of $\mathbf{A}$ by one indeterminate in the variety* RLA, *to that $\mathbf{B} \cong A[x]$.*

Lemma 2 *For all monomials t, q with at most one indeterminate x*

$$\mathbf{A}[x] \models t = s \iff \mathbf{A}[x] \models \lambda^* x.t = \lambda^* x.s$$

Proof

$$\begin{aligned} \mathbf{A}[x] \models t = s &\iff \mathbf{A}[x] \models (\lambda^* x.t)x^n = (\lambda^* x.s)x^n \\ &\iff \mathbf{A}[x] \models H_n[\lambda^* x.t] = H_n[\lambda^* x.s] \\ &\iff \mathbf{A}[x] \models \lambda^* x.t = \lambda^* x.s \end{aligned}$$

... and the result extends to polynomials too.

PART III - RECOVERING THE ORDINARY WORLD

From resource to ordinary algebras

- *Ideal* of bag-applicative structure $\mathbf{A}$: downward-closed subset of A closed under finite joins
- $\downarrow X =$ *ideal generated* by $X \subseteq A$
- $Ide(\mathbf{A}) =$ collection of all ideals of $\mathbf{A}$
- $X * Y = \downarrow \{a\overline{b} : a \in X,\ \overline{b} \in Y^*\}$

Definition 2 *Let* $\mathbf{A}$ *be a* RCA. *Define* $Ide(\mathbf{A}) = (Ide(A), *, \underline{K}, \underline{S})$ *by setting*

$$\underline{K} = \downarrow \{K\} \quad \underline{S} = \downarrow \{S_{\overline{n}} : \overline{n} \in \mathbb{N}^*\}$$

Definition 3 *Let* $\mathbf{B}$ *be a* RLAA. *Define* $Ide(\mathbf{B}) = (Ide(B), *, \underline{\lambda}x, \underline{x})_{x \in V}$ *by setting*

$$\underline{x} = \downarrow \{x\} \qquad \underline{\lambda}x.X = \downarrow \{\lambda x.a : a \in X\}$$

From resource to ordinary algebras

Theorem 7 *(i) If* $\mathbf{A}$ *is a* RCA, *then* $Ide(\mathbf{A})$ *is a combinatory algebra.*

(ii) If $\mathbf{A}$ *is a locally finite* RLAA, *then* $Ide(\mathbf{A})$ *is a* LAA.

(iii) If $\mathbf{A}$ *is a* RLA, *then* $Ide(\mathbf{A})$ *is a lambda algebra.*

Ordinary λ-terms can be interpreted in $Ide(\mathbf{A})$ ... what is the set of equations on λ-terms induced by such interpretation?

Lemma 3 *Let* $\mathbf{A}$ *be a locally finite* RLAA. *Then for all* λ*-terms* t, s

$$t, s \text{ have the same Böhm tree} \;\Rightarrow\; Ide(\mathbf{A}) \models t = s$$

Thus: ideal completions $\Rightarrow$ *sensible* λ-theories

Conclusions and future work

- we defined algebraic models of resource λ-calculus, category-theory-free setting
- we showed connections with algebraic models of ordinary λ-calculus
- extend our to resource λ-calculus with "promotion" and to differential λ-calculus (non-purely linear frameworks: duplication, cancellation re-gained)

THANK YOU