(Universal Algebra and)
Category Theory
in Foundations of Computer Science

Andrzej Tarlecki

Institute of Informatics
Faculty of Mathematics, Informatics and Mechanics
Warsaw University

http://www.mimuw.edu.pl/~tarlecki office: 4750
tarlecki@mimuw.edu.pl phone: (48)(22)(55) 44475, 44214

Universal algebra and category theory: basic ideas, notions and some results

- Algebras, homomorphisms, equations: basic definitions and results
- Categories; examples and simple cateogrical definitions
- Limits and colimits
- Functors and natural transformations
- Adjunctions
- Cartesian closed categories
- Institutions (abstract model theory, abstract specification theory)

BUT: *Tell me what you want to learn!*

Literature

Plenty of standard textbooks

But this will be roughly based on:

- D.T. Sannella, A. Tarlecki.
  Foundations of Algebraic Specifications and Formal Program Development.
  Springer, forthcoming.
  - Chap. 1: *Universal algebra*
  - Chap. 2: *Simple equational specifications*
  - Chap. 3: *Category theory*

One motivation

Software systems (modules, programs, databases. . .):

sets of data with operations on them

- Disregarding: code, efficiency, robustness, reliability, . . .
- Focusing on: CORRECTNESS

Universal algebra from rough analogy

module interface $\rightsquigarrow$ signature

module $\rightsquigarrow$ algebra

module specification $\rightsquigarrow$ class of algebras

Category theory

A language to further abstract away from the standard notions of universal algebra, to deal with their numerous variants needed in foundations of computer science.

Signatures

Algebraic signature:

$$\boxed{\Sigma = (S, \Omega)}$$

- *sort names*: S
- *operation names, classified by arities and result sorts*: $\Omega = \langle \Omega_{w,s} \rangle_{w \in S^*, s \in S}$

Alternatively:

$$\boxed{\Sigma = (S, \Omega, arity, sort)}$$

with *sort names* S, *operation names* Ω, and *arity and result sort functions*

$arity : \Omega \to S^*$ and $sort : \Omega \to S$.

- $f : s_1 \times \ldots \times s_n \to s$ stands for $s_1, \ldots, s_n, s \in S$ and $f \in \Omega_{s_1 \ldots s_n, s}$

Compare the two notions

Fix a signature $\Sigma = (S, \Omega)$ for a while.

Algebras

- Σ-*algebra*:

$$\boxed{A = (|A|, \langle f_A \rangle_{f \in \Omega})}$$

  - *carrier sets:* $|A| = \langle |A|_s \rangle_{s \in S}$
  - *operations*: $f_A : |A|_{s_1} \times \ldots \times |A|_{s_n} \to |A|_s$, for $f : s_1 \times \ldots \times s_n \to s$
- the class of all Σ-algebras:

$$\boxed{\mathbf{Alg}(\Sigma)}$$

Can $\mathbf{Alg}(\Sigma)$ be empty? Finite?
Can $A \in \mathbf{Alg}(\Sigma)$ have empty carriers?

Subalgebras

- for $A \in \mathbf{Alg}(\Sigma)$, a Σ-*subalgebra* $A_{sub} \subseteq A$ is given by subset $|A_{sub}| \subseteq |A|$ closed under the operations:
  - for $f\colon s_1 \times \ldots \times s_n \to s$ and $a_1 \in |A_{sub}|_{s_1}, \ldots, a_n \in |A_{sub}|_{s_n}$,
  $$f_{A_{sub}}(a_1, \ldots, a_n) = f_A(a_1, \ldots, a_n)$$
- for $A \in \mathbf{Alg}(\Sigma)$ and $X \subseteq |A|$, the *subalgebra of A genereted by X*, $\langle A \rangle_X$, is the least subalgebra of A that contains X.
- $A \in \mathbf{Alg}(\Sigma)$ is *reachable* if $\langle A \rangle_\emptyset$ coincides with A.

Fact: *For any $A \in \mathbf{Alg}(\Sigma)$ and $X \subseteq |A|$, $\langle A \rangle_X$ exists.*

Proof (idea):

- generate the generated subalgebra from X by closing it under operations in A; or
- the intersection of any family of subalgebras of A is a subalgebra of A.

Homomorphisms

- for $A, B \in \mathbf{Alg}(\Sigma)$, a Σ-*homomorphism* $h\colon A \to B$ is a function $h\colon |A| \to |B|$ that preserves the operations:
  - for $f\colon s_1 \times \ldots \times s_n \to s$ and $a_1 \in |A|_{s_1}, \ldots, a_n \in |A|_{s_n}$,
$$h_s(f_A(a_1, \ldots, a_n)) = f_B(h_{s_1}(a_1), \ldots, h_{s_n}(a_n))$$

Fact: *Given a homomorphism $h : A \to B$ and subalgebras A_{sub} of A and B_{sub} of B, the image of A_{sub} under h, $h(A_{sub})$, is a subalgebra of B, and the coimage of B_{sub} under h, $h^{-1}(B_{sub})$, is a subalgebra of A.*

Fact: *Given a homomorphism $h : A \to B$ and $X \subseteq |A|$, $h(\langle A\rangle_X) = \langle B\rangle_{h(X)}$.*

Fact: *Identity function on the carrier of $A \in \mathbf{Alg}(\Sigma)$ is a homomorphism $id_A : A \to A$. Composition of homomorphisms $h : A \to B$ and $g : B \to C$ is a homomorphism $h;g : A \to C$.*

Isomorphisms

- for $A, B \in \mathbf{Alg}(\Sigma)$, a Σ-*isomorphism* is any Σ-homomorphism $i\colon A \to B$ that has an *inverse*, i.e., a Σ-homomorphism $i^{-1}\colon B \to A$ such that $i;i^{-1} = id_A$ and $i^{-1};i = id_B$.
- Σ-algebras are *isomorphic* if there exists an isomorphism between them.

Fact: *A Σ-homomorphism is a Σ-isomorphism iff it is bijective ("1-1" and "onto").*

Fact: *Identities are isomorphisms, and any composition of isomorphisms is an isomorphism.*

Congruences

- for $A \in \mathbf{Alg}(\Sigma)$, a Σ-*congruence on* A is an equivalence $\equiv \subseteq |A| \times |A|$ that is closed under the operations:
  - for $f : s_1 \times \ldots \times s_n \to s$ and $a_1, a_1' \in |A|_{s_1}, \ldots, a_n, a_n' \in |A|_{s_n}$,
    if $a_1 \equiv_{s_1} a_1', \ldots, a_n \equiv_{s_n} a_n'$ then $f_A(a_1, \ldots, a_n) \equiv_s f_A(a_1', \ldots, a_n')$.

Fact: *For any relation $R \subseteq |A| \times |A|$ on the carrier of a Σ-algebra A, there exists the least congruence on A that conatins R.*

Fact: *For any Σ-homomorphism $h : A \to B$, the kernel of h, $K(h) \subseteq |A| \times |A|$, where $a\ K(h)\ a'$ iff $h(a) = h(a')$, is a Σ-congruence on A.*

Quotients

- for $A \in \mathbf{Alg}(\Sigma)$ and Σ-congruence $\equiv \subseteq |A| \times |A|$ on A, the *quotient algebra* $A/{\equiv}$ is built in the natural way on the equivalence classes of $\equiv$:
  - for $s \in S$, $|A/{\equiv}|_s = \{[a]_\equiv \mid a \in |A|_s\}$, with $[a]_\equiv = \{a' \in |A|_s \mid a \equiv a'\}$
  - for $f\colon s_1 \times \ldots \times s_n \to s$ and $a_1 \in |A|_{s_1}, \ldots, a_n \in |A|_{s_n}$,
    $$f_{A/\equiv}([a_1]_\equiv, \ldots, [a_n]_\equiv) = [f_A(a_1, \ldots, a_n)]_\equiv$$

Fact: *The above is well-defined; moreover, the natural map that assigns to every element its equivalence class is a Σ-homomorphisms $[\_]_\equiv : A \to A/{\equiv}$.*

Fact: *Given two Σ-congruences $\equiv$ and $\equiv'$ on A, $\equiv \subseteq \equiv'$ iff there exists a Σ-homomorphism $h : A/{\equiv} \to A/{\equiv'}$ such that $[\_]_\equiv ;h = [\_]_{\equiv'}$.*

Fact: *For any Σ-homomorphism $h : A \to B$, $A/K(h)$ is isomorphic with $h(A)$.*

Products

- for $A_i \in \mathbf{Alg}(\Sigma)$, $i \in \mathcal{I}$, the *product of* $\langle A_i \rangle_{i \in \mathcal{I}}$, $\prod_{i \in \mathcal{I}} A_i$ is built in the natural way on the Cartesian product of the carriers of A_i, $i \in \mathcal{I}$:
  - for $s \in S$, $|\prod_{i \in \mathcal{I}} A_i|_s = \prod_{i \in \mathcal{I}} |A_i|_s$
  - for $f \colon s_1 \times \ldots \times s_n \to s$ and $a_1 \in |\prod_{i \in \mathcal{I}} A_i|_{s_1}, \ldots, a_n \in |\prod_{i \in \mathcal{I}} A_i|_{s_n}$, for $i \in \mathcal{I}$, $f_{\prod_{i \in \mathcal{I}} A_i}(a_1, \ldots, a_n)(i) = f_{A_i}(a_1(i), \ldots, a_n(i))$

Fact: *For any family $\langle A_i \rangle_{i \in \mathcal{I}}$ of Σ-algebras, projections $\pi_i(a) = a(i)$, where $i \in \mathcal{I}$ and $a \in \prod_{i \in \mathcal{I}} |A_i|$, are Σ-homomorphisms $\pi_i : \prod_{i \in \mathcal{I}} A_i \to A_i$.*

> Define the product of the empty family of Σ-algebras.
>
> When the projection π_i is an isomorphism?

Terms

Consider an S-sorted set X of variables.

- *terms* $t \in |T_\Sigma(X)|$ are built using variables X, constants and operations from Ω in the usual way: $|T_\Sigma(X)|$ is the least set such that
  - $X \subseteq |T_\Sigma(X)|$
  - for $f\colon s_1 \times \ldots \times s_n \to s$ and $t_1 \in |T_\Sigma(X)|_{s_1}, \ldots, t_n \in |T_\Sigma(X)|_{s_n}$, $f(t_1, \ldots, t_n) \in |T_\Sigma(X)|_s$
- for any Σ-algebra A and valuation $v : X \to |A|$, *the value $t_A[v]$ of a term $t \in |T_\Sigma(X)|$ in A under v* is determined inductively:
  - $x_A[v] = v_s(x)$, for $x \in X_s$, $s \in S$
  - $(f(t_1, \ldots, t_n))_A[v] = f_A((t_1)_A[v], \ldots, (t_n)_A[v])$, for $f\colon s_1 \times \ldots \times s_n \to s$ and $t_1 \in |T_\Sigma(X)|_{s_1}, \ldots, t_n \in |T_\Sigma(X)|_{s_n}$

Above and in the following: assuming unambiguous "parsing" of terms!

Term algebras

Consider an S-sorted set X of variables.

- The *term algebra* $T_\Sigma(X)$ has the set of terms as the carrier and operations defined "syntactically":
  - for $f : s_1 \times \ldots \times s_n \to s$ and $t_1 \in |T_\Sigma(X)|_{s_1}, \ldots, t_n \in |T_\Sigma(X)|_{s_n}$, $f_{T_\Sigma(X)}(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$.

Fact: *For any S-sorted set X of variables, Σ-algebra A and valuation $v : X \to |A|$, there is a unique Σ-homomorphism $v^\# : T_\Sigma(X) \to A$ that extends v. Moreover, for $t \in |T_\Sigma(X)|$, $v^\#(t) = t_A[v]$.*

$$\begin{tikzcd}
 & X \arrow[rr, "id_{X \hookrightarrow |T_\Sigma(X)|}"] \arrow[drr, "v"'] & & {|T_\Sigma(X)|} \arrow[d, "{|v^\#|}"] & & T_\Sigma(X) \arrow[d, "\exists!\, v^\#"] & \\
\mathbf{Set}^S & & & {|A|} & & A & \mathbf{Alg}(\Sigma)
\end{tikzcd}$$

Equations

- *Equation*:

$$\boxed{\forall X.t = t'}$$

where:

– X is a set of variables, and

– $t, t' \in |T_\Sigma(X)|_s$ are terms of a common sort.

- *Satisfaction relation*: Σ-algebra A *satisfies* $\forall X.t = t'$

$$\boxed{A \models \forall X.t = t'}$$

when for all $v \colon X \to |A|$, $t_A[v] = t'_A[v]$.

Semantic entailment

$$\boxed{\Phi \models_\Sigma \varphi}$$

Σ-equation φ is a semantic consequence of a set of Σ-equations Φ if φ holds in every Σ-algebra that satisfies Φ.

BTW:

- *Models* of a set of equations: $Mod(\Phi) = \{A \in \mathbf{Alg}(\Sigma) \mid A \models \Phi\}$
- *Theory* of a class of algebras: $Th(\mathcal{C}) = \{\varphi \mid \mathcal{C} \models \varphi\}$
- $\Phi \models \varphi \iff \varphi \in Th(Mod(\Phi))$
- Mod and Th form a *Galois connection*

Equational specifications

$$\boxed{\langle \Sigma, \Phi \rangle}$$

- signature Σ, to determine the static module interface
- axioms (Σ-equations), to determine required module properties

BUT:

Fact: *A class of Σ-algebras is equationally definable iff it is closed under subalgebras, products and homomorphic images.*

Equational specifications typically admit a lot of undesirable "modules"

Example

$$\begin{aligned}
\textbf{spec } \textsc{NaiveNat} = \textbf{sort } & Nat \\
\textbf{ops } & 0 : Nat; \\
& \quad succ : Nat \to Nat; \\
& \quad _ + _ : Nat \times Nat \to Nat \\
\textbf{axioms } & \forall n{:}Nat \bullet n + 0 = n; \\
& \qquad \forall n, m{:}Nat \bullet n + succ(m) = succ(n + m)
\end{aligned}$$

Now:

$$\textsc{NaiveNat} \not\models \forall n, m{:}Nat \bullet n + m = m + n$$

How to fix this

- Other (stronger) *logical systems*: conditional equations, first-order logic, higher-order logics, other bells-and-whistles
  - more about this elsewhere. . .

Institutions!

- *Constraints*:
  - *reachability* (and generation): "no junk"
  - *initiality* (and freeness): "no junk" & "no confusion"

  Constraints can be thought of as special (higher-order) formulae.

There has been a population explosion among logical systems. . .

Initial models

Fact: *Every equational specification* $\langle \Sigma, \Phi \rangle$ *has an initial model: there exists a* Σ*-algebra* $I \in Mod(\Phi)$ *such that for every* Σ*-algebra* $M \in Mod(\Phi)$ *there exists a unique* Σ*-homomorphism from* I *to* M*.*

Proof (idea):

- I is the quotient of the algebra of ground Σ-terms by the congruence that glues together all ground terms t, t' such that $\Phi \models \forall \emptyset . t = t'$.
- I is the reachable subalgebra of the product of "all" (up to isomorphism) reachable algebras in $Mod(\Phi)$.

BTW: This can be generalised to the existence of a free model of $\langle \Sigma, \Phi \rangle$ over any (many-sorted) set of data.

Example

$$
\begin{array}{l}
\textbf{spec } \mathrm{N{\scriptsize AT}} = \textbf{free } \{ \ \textbf{sort } Nat \\
\qquad \textbf{ops } 0 : Nat; \\
\qquad\qquad succ : Nat \to Nat; \\
\qquad\qquad \_ + \_ : Nat \times Nat \to Nat \\
\qquad \textbf{axioms } \forall n{:}Nat \bullet n + 0 = n; \\
\qquad\qquad \forall n, m{:}Nat \bullet n + succ(m) = succ(n + m) \\
\quad \}
\end{array}
$$

Now:

$$\mathrm{N{\scriptsize AT}} \models \forall n, m{:}Nat \bullet n + m = m + n$$

Example′

$$\begin{aligned}
&\textbf{spec } \textsc{Nat}' = \textbf{free type } Nat ::= 0 \mid succ(Nat) \\
&\qquad\qquad \textbf{op } _+_ : Nat \times Nat \to Nat \\
&\qquad\qquad \textbf{axioms } \forall n{:}Nat \bullet n + 0 = n; \\
&\qquad\qquad\qquad\quad \forall n, m{:}Nat \bullet n + succ(m) = succ(n + m)
\end{aligned}$$

$$\boxed{\textsc{Nat} \equiv \textsc{Nat}'}$$

Another example

$$
\begin{array}{l}
\textbf{spec } \textsc{String} = \\
\quad \textbf{generated } \{ \textbf{ sort } \mathit{String} \\
\qquad\qquad \textbf{ops } \mathit{nil} : \mathit{String}; \\
\qquad\qquad\quad a, \ldots, z : \mathit{String}; \\
\qquad\qquad\quad \_ \frown \_ : \mathit{String} \times \mathit{String} \to \mathit{String} \; \} \\
\qquad\qquad \textbf{axioms } \forall s{:}\mathit{String} \bullet s \frown \mathit{nil} = s; \\
\qquad\qquad\qquad \forall s{:}\mathit{String} \bullet \mathit{nil} \frown s = s; \\
\qquad\qquad\qquad \forall s, t, v{:}\mathit{String} \bullet s \frown (t \frown v) = (s \frown t) \frown v \\
\qquad \}
\end{array}
$$

Equational calculus

$$\frac{}{\forall X.t = t} \qquad \frac{\forall X.t = t'}{\forall X.t' = t} \qquad \frac{\forall X.t = t' \quad \forall X.t' = t''}{\forall X.t = t''}$$

$$\frac{\forall X.t_1 = t'_1 \quad \ldots \quad \forall X.t_n = t'_n}{\forall X.f(t_1 \ldots t_n) = f(t'_1 \ldots t'_n)} \qquad \frac{\forall X.t = t'}{\forall Y.t[\theta] = t'[\theta]} \text{ for } \theta\colon X \to |T_\Sigma(Y)|$$

Mind the variables!

$a = b$ does *not* follow from $a = f(x)$ and $f(x) = b$, unless...

Proof-theoretic entailment

$$\boxed{\Phi \vdash_\Sigma \varphi}$$

Σ-equation φ is a proof-theoretic consequence of a set of Σ-equations Φ if φ can be derived from Φ by the rules.

How to justify this?

Semantics!

Soundness & completeness

Fact: *The equational calculus is sound and complete:*

$$\boxed{\Phi \models \varphi \iff \Phi \vdash \varphi}$$

- soundness: "all that can be proved, is true" ($\Phi \models \varphi \Longleftarrow \Phi \vdash \varphi$)
- completeness: "all that is true, can be proved" ($\Phi \models \varphi \Longrightarrow \Phi \vdash \varphi$)

Proof (idea):

- soundness: easy!
- completeness: not so easy!

Moving between signatures

Let $\Sigma = (S, \Omega)$ and $\Sigma' = (S', \Omega')$

$$\boxed{\sigma : \Sigma \to \Sigma'}$$

- *Signature morphism* maps:
  - sorts to sorts: $\sigma : S \to S'$
  - operation names to operation names, preserving their profiles:
    $\sigma : \Omega_{w,s} \to \Omega'_{\sigma(w),\sigma(s)}$, for $w \in S^*$, $s \in S$, that is: for $f : s_1 \times \ldots \times s_n \to s$,
    $\sigma(f) : \sigma(s_1) \times \ldots \times \sigma(s_n) \to \sigma(s)$,

Let $\sigma\colon \Sigma \to \Sigma'$

Translating syntax

- *translation of variables*: $X \mapsto X'$, where $X'_{s'} = \biguplus_{\sigma(s)=s'} X_s$
- *translation of terms*: $\sigma\colon |T_\Sigma(X)|_s \to |T_{\Sigma'}(X')|_{\sigma(s)}$, for $s \in S$
- *translation of equations*: $\sigma(\forall X.t_1 = t_2)$ yields $\forall X'.\sigma(t_1) = \sigma(t_2)$

. . . and semantics

- σ-*reduct*: $\_|_\sigma\colon \mathbf{Alg}(\Sigma') \to \mathbf{Alg}(\Sigma)$, where for $A' \in \mathbf{Alg}(\Sigma')$
  - $|A'|_\sigma|_s = |A'|_{\sigma(s)}$, for $s \in S$
  - $f_{A'|_\sigma} = \sigma(f)_{A'}$ for $f \in \Omega$

Note the contravariancy!

Satisfaction condition

Fact: *For all signature morphisms $\sigma : \Sigma \to \Sigma'$, Σ'-algebras A' and Σ-equations φ:*

$$\boxed{A'|_\sigma \models_\Sigma \varphi \iff A' \models_{\Sigma'} \sigma(\varphi)}$$

Proof (idea): for $t \in |T_\Sigma(X)|$ and $v : X \to |A'|_\sigma|$, $t_{A'|_\sigma}[v] = \sigma(t)_{A'}[v']$, where $v' : X' \to |A'|$ is given by $v'_{\sigma(s)}(x) = v_s(x)$ for $s \in S$, $x \in X_s$.

TRUTH is preserved (at least) under:

- *change of notation*
- *restriction/extension of irrelevant context*

Preservation of consequence

Given any signature morphism $\sigma : \Sigma \to \Sigma'$, set of Σ-equations Φ and Σ-equation φ:

$$\boxed{\Phi \models_\Sigma \varphi \implies \sigma(\Phi) \models_{\Sigma'} \sigma(\varphi)}$$

Moreover, if $\_|_\sigma : \mathbf{Alg}(\Sigma') \to \mathbf{Alg}(\Sigma)$ is surjective then:

$$\boxed{\Phi \models_\Sigma \varphi \iff \sigma(\Phi) \models_{\Sigma'} \sigma(\varphi)}$$

In general, the equivalence does not hold!

Specification morphisms

Specification morphism:

$$\boxed{\sigma : \langle \Sigma, \Phi \rangle \to \langle \Sigma', \Phi' \rangle}$$

is a signature morphism $\sigma : \Sigma \to \Sigma'$ such that for all $M' \in \mathbf{Alg}(\Sigma')$:

$$\boxed{M' \in Mod(\Phi') \implies M'|_\sigma \in Mod(\Phi)}$$

Then $\_|_\sigma : Mod(\Phi') \to Mod(\Phi)$

Fact: *A signature morphism* $\sigma : \Sigma \to \Sigma'$ *is a specification morphism* $\sigma : \langle \Sigma, \Phi \rangle \to \langle \Sigma', \Phi' \rangle$ *if and only if* $\Phi' \models \sigma(\Phi)$.

Conservativity

A specification morphism:

$$\boxed{\sigma : \langle \Sigma, \Phi \rangle \to \langle \Sigma', \Phi' \rangle}$$

is *conservative* if for all Σ-equations φ: $\boxed{\Phi' \models_{\Sigma'} \sigma(\varphi) \implies \Phi \models_{\Sigma} \varphi}$

BTW: for all specification morphisms
$$\Phi \models_{\Sigma} \varphi \implies \Phi' \models_{\Sigma'} \sigma(\varphi)$$

A specification morphism $\sigma : \langle \Sigma, \Phi \rangle \to \langle \Sigma', \Phi' \rangle$ *admits model expansion* if for each $M \in Mod(\Phi)$ there exists $M' \in Mod(\Phi')$ such that $M'|_{\sigma} = M$

(i.e., $\_|_{\sigma} : Mod(\Phi') \to Mod(\Phi)$ is surjective).

Fact: *If $\sigma : \langle \Sigma, \Phi \rangle \to \langle \Sigma', \Phi' \rangle$ admits model expansion then it is conservative.*

In general, the equivalence does not hold!

More general signature morphisms

Let $\Sigma = (S, \Omega)$ and $\Sigma' = (S', \Omega')$

$$\boxed{\delta\colon \Sigma \to \Sigma'}$$

- *Derived signature morphism* maps sorts to sorts: $\delta\colon S \to S'$, and operation names to terms, preserving their profiles: for $f\colon s_1 \times \ldots \times s_n \to s$,

$$\boxed{\delta(f) \in |T_{\Sigma'}(\{x_1{:}\delta(s_1), \ldots, x_n{:}\delta(s_n)\})|_{\delta(s)}}$$

- Translation of syntax, reducts of algebras, satisfaction condition, and many other notions and results: similarly as before.

not quite all though...

Partial algebras

- *Algebraic signature* Σ: as before
- *Partial* Σ*-algebra*:

$$\boxed{A = (|A|, \langle f_A \rangle_{f \in \Omega})}$$

as before, but operations $f_A\colon |A|_{s_1} \times \ldots \times |A|_{s_n} \rightharpoonup |A|_s$, for $f\colon s_1 \times \ldots \times s_n \to s$, may now be *partial functions*.

BTW: Constants may be undefined as well.

- $\mathbf{PAlg}(\Sigma)$ stands for the class of all partial Σ-algebras.

Fix a signature $\Sigma = (S, \Omega)$ for a while.

Few further notions

- *subalgebra* $A_{sub} \subseteq A$: given by subset $|A_{sub}| \subseteq |A|$ closed under the operations; (BTW: at least two other natural notions are possible)
- *homomorphism* $h\colon A \to B$: map $h\colon |A| \to |B|$ that preserves definedness and results of operations; it is *strong* if in addition it reflects definedness of operations; (strong) homomorphisms are closed under composition; (BTW: very interesting alternative: *partial* map $h\colon |A| \rightharpoonup |B|$ that preserves results of operations)
- *congruence* $\equiv$ on A: equivalence $\equiv \subseteq |A| \times |A|$ closed under the operations whenever they are defined; it is *strong* if in addition it reflects definedness of operations; (strong) congruences are kernels of (strong) homomorphisms;
- *quotient algebra* $A/{\equiv}$: built in the natural way on the equivalence classes of $\equiv$; the natural homomorphism from A to $A/{\equiv}$ is strong if the congruence is strong.

Formulae

(Strong) equation: $$\boxed{\forall X.t \stackrel{s}{=} t'}$$ as before	*Definedness formula*: $$\boxed{\forall X.def\ t}$$ where X is a set of variables, and $t \in |T_\Sigma(X)|_s$ is a term

Satisfaction relation

partial Σ-algebra A *satisfies* $\forall X.t \stackrel{s}{=} t'$ $$\boxed{A \models \forall X.t \stackrel{s}{=} t'}$$ when for all $v\colon X \to |A|$, $t_A[v]$ is defined iff $t'_A[v]$ is defined, and then $t_A[v] = t'_A[v]$	partial Σ-algebra A *satisfies* $\forall X.def\ t$ $$\boxed{A \models \forall X.def\ t}$$ when for all $v\colon X \to |A|$, $t_A[v]$ is defined

An alternative

- *(Existence) equation*:

$$\boxed{\forall X.t \stackrel{e}{=} t'}$$

  where:

  – X is a set of variables, and

  – $t, t' \in |T_\Sigma(X)|_s$ are terms of a common sort.

- *Satisfaction relation*: Σ-algebra A *satisfies* $\forall X.t \stackrel{e}{=} t'$

$$\boxed{A \models \forall X.t \stackrel{e}{=} t'}$$

  when for all $v\colon X \to |A|$, $t_A[v] = t'_A[v]$ — both sides are defined and equal.

BTW:

- $\forall X.t \stackrel{e}{=} t'$ iff $\forall X.(t \stackrel{s}{=} t' \wedge \mathit{def}\ t)$
- $\forall X.t \stackrel{s}{=} t'$ iff $\forall X.(\mathit{def}\ t \iff \mathit{def}\ t') \wedge (\mathit{def}\ t \implies t \stackrel{e}{=} t')$

Further notions and results

To introduce and/or check:

- partial equational specifications (trivial)
- characterization of definable classes of partial algebras (difficult!)
- existence of initial models for partial equational specifications (non-trivial for existence equations; difficult for strong equations and definedness formulae)
- proof systems for partial equational logic (*ditto*)
- signature morphisms, translation of formulae, reducts of partial algebras, satisfaction condition; specification morphisms, conservativity, etc. (easy)
- even more general signature morphisms: $\delta : \Sigma \to \Sigma'$ maps sort names to sort names, and operation names $f : s_1 \times \ldots s_n \to s$ to sequences $\langle \varphi_i, t_i \rangle_{i \geq 0}$, where φ_i is a Σ'-formula and t_i is a Σ'-term of sort $\delta(s)$, both with variables among $x_1{:}\delta(s_1), \ldots, x_n{:}\delta(s_n)$; syntax does not quite translate, but reducts are well defined...

Example

$$
\begin{aligned}
\textbf{spec } \textsc{NatPred} = \textbf{free } \{ \; & \textbf{sort } Nat \\
& \textbf{ops } 0 : Nat; \\
& \qquad succ : Nat \to Nat; \\
& \qquad \_\_+\_\_ : Nat \times Nat \to Nat \\
& \qquad pred : Nat \to ?\; Nat \\
& \textbf{axioms } \forall n{:}Nat \bullet n + 0 = n; \\
& \qquad\qquad \forall n, m{:}Nat \bullet n + succ(m) = succ(n+m) \\
& \qquad\qquad \forall n{:}Nat \bullet pred(succ(n)) \stackrel{s}{=} n; \\
\}
\end{aligned}
$$

Example′

$$\begin{aligned}
&\textbf{spec } \textsc{NatPred}' = \textbf{free type } \mathit{Nat} ::= 0 \mid \mathit{succ}(\mathit{pred} :?\ \mathit{Nat}) \\
&\qquad\qquad\qquad\quad \textbf{op } _+_ : \mathit{Nat} \times \mathit{Nat} \to \mathit{Nat} \\
&\qquad\qquad\qquad\quad \textbf{axioms } \forall n{:}\mathit{Nat} \bullet n + 0 = n; \\
&\qquad\qquad\qquad\qquad\qquad\quad \forall n, m{:}\mathit{Nat} \bullet n + \mathit{succ}(m) = \mathit{succ}(n + m)
\end{aligned}$$

$$\boxed{\textsc{NatPred} \equiv \textsc{NatPred}'}$$