(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

tarlecki@mimuw.edu.pl

This course: http://www.mimuw.edu.pl/~tarlecki/teaching/ct/
Universal algebra and category theory: basic ideas, notions and some results

- Algebras, homomorphisms, equations: basic definitions and results
- Categories; examples and simple categorical 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!
Plenty of standard textbooks

But this will be roughly based on:

  - 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 $\mapsto$ signature
module $\mapsto$ algebra
module specification $\mapsto$ 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:**

\[ \Sigma = (S, \Omega) \]

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

Alternatively:

\[ \Sigma = (S, \Omega, \text{arity}, \text{sort}) \]

with **sort names** \( S \), **operation names** \( \Omega \), and **arity and result sort functions**

\[ \text{arity}: \Omega \rightarrow S^* \text{ and } \text{sort}: \Omega \rightarrow S. \]

- \( f: s_1 \times \ldots \times s_n \rightarrow s \) 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**

- **$\Sigma$-algebra**:

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

- **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 $\Sigma$-algebras**:

  $$\text{Alg}(\Sigma)$$

Can $\text{Alg}(\Sigma)$ be empty? Finite?

Can $A \in \text{Alg}(\Sigma)$ have empty carriers?
for $A \in \text{Alg}(\Sigma)$, a $\Sigma$-subalgebra $A_{sub} \subseteq A$ is given by subset $|A_{sub}| \subseteq |A|$ closed under the operations:

- for $f : s_1 \times \ldots \times s_n \rightarrow s$ 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 $A \in \text{Alg}(\Sigma)$ and $X \subseteq |A|$, the subalgebra of $A$ generated by $X$, $\langle A \rangle_X$, is the least subalgebra of $A$ that contains $X$.

$A \in \text{Alg}(\Sigma)$ is reachable if $\langle A \rangle_{\emptyset}$ coincides with $A$.

**Fact:** For any $A \in \text{Alg}(\Sigma)$ 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 $A$. 

---

Andrzej Tarlecki: Category Theory, 2017
Homomorphisms

- for $A, B \in \text{Alg}(\Sigma)$, a $\Sigma$-homomorphism $h: A \to B$ is a function $h: |A| \to |B|$ that preserves the operations:
  - for $f: s_1 \times \ldots \times s_n \to s$ 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 $A_{\text{sub}}$ of $A$ and $B_{\text{sub}}$ of $B$, the image of $A_{\text{sub}}$ under $h$, $h(A_{\text{sub}})$, is a subalgebra of $B$, and the coimage of $B_{\text{sub}}$ under $h$, $h^{-1}(B_{\text{sub}})$, 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:** If two homomorphisms $h_1, h_2: A \to B$ coincide on $X \subseteq |A|$, then they coincide on $\langle A \rangle_X$.

**Fact:** Identity function on the carrier of $A \in \text{Alg}(\Sigma)$ is a homomorphism $\text{id}_A: A \to A$. 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 $\Sigma$-isomorphism is any $\Sigma$-homomorphism $i : A \to B$ that has an inverse, i.e., a $\Sigma$-homomorphism $i^{-1} : B \to A$ such that $i \circ i^{-1} = \text{id}_A$ and $i^{-1} \circ i = \text{id}_B$.

- $\Sigma$-algebras are isomorphic if there exists an isomorphism between them.

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

**Fact:** Identities are isomorphisms, and any composition of isomorphisms is an isomorphism.
• for $A \in \text{Alg}(\Sigma)$, a $\Sigma$-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 $\Sigma$-algebra $A$, there exists the least congruence on $A$ that contains $R$.

**Fact:** For any $\Sigma$-homomorphism $h : A \to B$, the kernel of $h$, $K(h) \subseteq |A| \times |A|$, where $a \equiv_{K(h)} a'$ iff $h(a) = h(a')$, is a $\Sigma$-congruence on $A$. 
for $A \in \text{Alg}(\Sigma)$ and $\Sigma$-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 $f : s_1 \times \ldots \times s_n \to s$ 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 $\Sigma$-homomorphisms $[\_]_{\equiv} : A \to A/\equiv$.

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

**Fact:** For any $\Sigma$-homomorphism $h : A \to B$, $A/K(h)$ is isomorphic with $h(A)$. 
• for $A_i \in \text{Alg}(\Sigma)$, $i \in \mathcal{I}$, 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 $f : s_1 \times \ldots \times s_n \to s$ 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 $\Sigma$-algebras, projections $\pi_i(a) = a(i)$, where $i \in \mathcal{I}$ and $a \in \prod_{i \in \mathcal{I}} |A_i|$, are $\Sigma$-homomorphisms $\pi_i : \prod_{i \in \mathcal{I}} A_i \to A_i$.

Define the product of the empty family of $\Sigma$-algebras. When the projection $\pi_i$ is an isomorphism?
Consider an $S$-sorted set $X$ of variables.

- **terms** $t \in |T_\Sigma(X)|$ are built using variables $X$, constants and operations from $\Omega$ in the usual way: $|T_\Sigma(X)|$ is the least set such that
  - $X \subseteq |T_\Sigma(X)|$
  - for $f : s_1 \times \ldots \times s_n \to s$ and $t_1 \in |T_\Sigma(X)|_{s_1}, \ldots, t_n \in |T_\Sigma(X)|_{s_n}$, $f(t_1, \ldots, t_n) \in |T_\Sigma(X)|_s$

- for any $\Sigma$-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 $f : s_1 \times \ldots \times s_n \to s$ 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":
  
  $f_{T_{\Sigma}(X)}(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$.

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

**Fact:** For any $S$-sorted sets $X$, $Y$ and $Z$ (of variables) and substitutions

$\theta_1 : X \to |T_\Sigma(Y)|$ and $\theta_2 : Y \to |T_\Sigma(Z)|$

\[
\theta_1^\# ; \theta_2^\# = (\theta_1 ; \theta_2^\#)^\#
\]
Equations

- **Equation:**

\[ \forall X. t = t' \]

where:
- \( X \) is a set of variables, and
- \( t, t' \in |T_\Sigma(X)|_s \) are terms of a common sort.

- **Satisfaction relation:** \( \Sigma \)-algebra \( A \) satisfies \( \forall X. t = t' \)

\[ A \models \forall X. t = t' \]

when for all \( v: X \to |A|, t_A[v] = t'_A[v] \).
**Semantic entailment**

\[ \Phi \models_\Sigma \varphi \]

**Σ-equation** \( \varphi \) **is a semantic consequence of a set of Σ-equations** \( \Phi \)

if \( \varphi \) holds in every \( \Sigma \)-algebra that satisfies \( \Phi \).

**BTW:**

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

\[ \langle \Sigma, \Phi \rangle \]

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

BUT:

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

Equational specifications typically admit a lot of undesirable “modules”
Example

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

Now:

\[
\text{NaiveNat} \not\models \forall n, m: \text{Nat} \bullet n + m = m + n
\]
How to fix this

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

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

Proof (idea):

- $I$ is the quotient of the algebra of ground $\Sigma$-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 $\text{Mod}(\Phi)$.

**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

\[
\text{spec } \text{Nat} = \text{free } \{ \text{sort } \text{Nat} \\
\quad \text{ops } 0 : \text{Nat}; \\
\quad \quad \text{succ} : \text{Nat} \to \text{Nat}; \\
\quad \quad _+ _ : \text{Nat} \times \text{Nat} \to \text{Nat} \\
\quad \text{axioms } \forall n: \text{Nat} \bullet n + 0 = n; \\
\quad \quad \forall n, m: \text{Nat} \bullet n + \text{succ}(m) = \text{succ}(n + m) \\
\} 
\]

Now:

\[
\text{Nat} \models \forall n, m: \text{Nat} \bullet n + m = m + n
\]
Example′

\[
\text{spec } \mathbb{N} \text{′} = \text{free type } \mathbb{N} ::= 0 \mid \text{succ}(\mathbb{N})
\]

\[
\text{op } + : \mathbb{N} \times \mathbb{N} \to \mathbb{N}
\]

\[
\text{axioms } \forall n: \mathbb{N} \bullet n + 0 = n;
\]

\[
\forall n, m: \mathbb{N} \bullet n + \text{succ}(m) = \text{succ}(n + m)
\]

\[
\mathbb{N} \equiv \mathbb{N} \text{′}
\]
Another example

```latex
spec String =
  generated { sort String
    ops nil: String;
    a, ..., z: String;
    _ ^ _: String \times String \to String }
axioms \forall s: String \bullet s ^ nil = s;
\forall s: String \bullet nil ^ s = s;
\forall s, t, v: String \bullet s ^ (t ^ v) = (s ^ t) ^ v
  }
```
Equational calculus

\[
\begin{align*}
\forall X.t & = t' & \forall X.t & = t' & \forall X.t' & = t'' \\
\forall X.t & = t & \forall X.t' & = t & \forall X.t & = t'' \\
\forall X.t_1 = t'_1 & \ldots & \forall X.t_n = t'_n & \forall X.t & = t' \\
\forall X.f(t_1 \ldots t_n) & = f(t'_1 \ldots t'_n) & \forall Y.t[\theta] & = t'[\theta] \\
\text{for } \theta : X \to |T_\Sigma(Y)|
\end{align*}
\]

Mind the variables!

\[
a = b \text{ does not follow from } a = f(x) \text{ and } f(x) = b, \text{ unless} \ldots
\]
Proof-theoretic entailment

\[ \Phi \vdash \Sigma \varphi \]

\(\Sigma\)-equation \(\varphi\) is a proof-theoretic consequence of a set of \(\Sigma\)-equations \(\Phi\) if \(\varphi\) can be derived from \(\Phi\) by the rules.

How to justify this?

Semantics!
Soundness & completeness

Fact: The equational calculus is sound and complete:

\[ \Phi \models \varphi \iff \Phi \vdash \varphi \]

- soundness: “all that can be proved, is true” (\( \Phi \models \varphi \iff \Phi \vdash \varphi \))
- completeness: “all that is true, can be proved” (\( \Phi \models \varphi \implies \Phi \vdash \varphi \))

Proof (idea):
- soundness: easy!
- completeness: not so easy!
Moving between signatures

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

$\sigma : \Sigma \rightarrow \Sigma'$

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

**Translating syntax**

- *translation of variables*: $X \mapsto X'$, where $X'_{s'} = \bigsqcup_{\sigma(s) = s'} X_s$
- *translation of terms*: $\sigma : |T_\Sigma(X)|_s \to |T_{\Sigma'}(X')|_{\sigma(s)}$, 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**

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

*Note the contravariance!*
Satisfaction condition

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

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

**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 \( \Sigma \)-equations \( \Phi \) and \( \Sigma \)-equation \( \varphi \):

\[
\Phi \models_\Sigma \varphi \implies \sigma(\Phi) \models_{\Sigma'} \sigma(\varphi)
\]

Moreover, if \( \sigma : \text{Alg}(\Sigma') \to \text{Alg}(\Sigma) \) is surjective then:

\[
\Phi \models_\Sigma \varphi \iff \sigma(\Phi) \models_{\Sigma'} \sigma(\varphi)
\]

In general, the equivalence does not hold!
Specification morphisms

**Specification morphism:**

\[ \sigma : \langle \Sigma, \Phi \rangle \to \langle \Sigma', \Phi' \rangle \]

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

\[ M' \in \text{Mod}(\Phi') \implies M'|_\sigma \in \text{Mod}(\Phi) \]

**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:

\[ \sigma : \langle \Sigma, \Phi \rangle \rightarrow \langle \Sigma', \Phi' \rangle \]

is **conservative** if for all \( \Sigma \)-equations \( \varphi \):

\[ \Phi' \models_{\Sigma'} \sigma(\varphi) \Rightarrow \Phi \models_{\Sigma} \varphi \]

**BTW:** for all specification morphisms

\[ \Phi \models_{\Sigma} \varphi \Rightarrow \Phi' \models_{\Sigma'} \sigma(\varphi) \]

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

(i.e., \( \models_{\sigma} : \text{Mod}(\Phi') \rightarrow \text{Mod}(\Phi) \) is surjective).

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

**In general, the equivalence does not hold!**
More general signature morphisms

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

\[
\delta : \Sigma \to \Sigma'
\]

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

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

- 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** $\Sigma$: as before

- **Partial $\Sigma$-algebra**:

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

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

  **BTW:** Constants may be undefined as well.

- $\text{PAlg}(\Sigma)$ stands for the class of all partial $\Sigma$-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** $h: A \to B$: map $h: |A| \to |B|$ 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 $h: |A| \rightharpoonup |B|$ 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.
(Strong) equation:  
\[ \forall X. t \overset{s}{=} t' \]  

as before

Definedness formula:  
\[ \forall X. \text{def} \ t \]

where \( X \) is a set of variables, \( t \in |T_\Sigma(X)| \) is a term

Satisfaction relation

partial \( \Sigma \)-algebra \( A \) satisfies \( \forall X. t \overset{s}{=} t' \)

\[ A \models \forall X. t \overset{s}{=} t' \]

when for all \( v: X \rightarrow |A| \), \( t_A[v] \) is defined iff \( t'_A[v] \) is defined, and then \( t_A[v] = t'_A[v] \)

partial \( \Sigma \)-algebra \( A \) satisfies \( \forall X. \text{def} \ t \)

\[ A \models \forall X. \text{def} \ t \]

when for all \( v: X \rightarrow |A| \), \( t_A[v] \) is defined
An alternative

- **(Existence) equation:**

\[ \forall X. t \overset{e}{=} t' \]

where:
- \( X \) is a set of variables, and
- \( t, t' \in |T_\Sigma(X)|_s \) are terms of a common sort.

- **Satisfaction relation:** \( \Sigma \)-algebra \( A \) satisfies \( \forall X. t \overset{e}{=} t' \)

\[ A \models \forall X. t \overset{e}{=} t' \]

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

**BTW:**

- \( \forall X. t \overset{e}{=} t' \) iff \( \forall X. (t \overset{s}{=} t' \land \text{def } t) \)

- \( \forall X. t \overset{s}{=} t' \) iff \( \forall X. (\text{def } t \iff \text{def } t') \land (\text{def } t \implies t \overset{e}{=} t') \)
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: $\delta : \Sigma \rightarrow \Sigma'$ maps sort names to sort names, and operation names $f : s_1 \times \ldots s_n \rightarrow s$ to sequences $\langle \varphi_i, t_i \rangle_{i \geq 0}$, where $\varphi_i$ is a $\Sigma'$-formula and $t_i$ is a $\Sigma'$-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

\[\text{spec } \text{NatPred} = \text{free } \{ \text{sort } \text{Nat} \]

\[\text{ops } 0 : \text{Nat};\]

\[\text{succ} : \text{Nat} \rightarrow \text{Nat};\]

\[\_ + \_ : \text{Nat} \times \text{Nat} \rightarrow \text{Nat}\]

\[\text{pred} : \text{Nat} \rightarrow ? \text{Nat}\]

\[\text{axioms} \; \forall n : \text{Nat} \Rightarrow n + 0 = n;\]

\[\forall n, m : \text{Nat} \Rightarrow n + \text{succ}(m) = \text{succ}(n + m)\]

\[\forall n : \text{Nat} \Rightarrow \text{pred}(\text{succ}(n)) \Rightarrow n;\]

\}
Example'

\[
\text{spec } \text{NatPred}' = \text{free type } Nat ::= 0 | \text{succ}(\text{pred} : ? Nat)
\]

\[
\text{op } \_ + \_ : Nat \times Nat \to Nat
\]

\[
\text{axioms } \forall n: Nat \bullet n + 0 = n;
\]

\[
\forall n, m: Nat \bullet n + \text{succ}(m) = \text{succ}(n + m)
\]

\[
\text{NatPred } \equiv \text{NatPred}'
\]