Universal algebra

Basics of universal algebra:

- signatures and algebras
- homomorphisms, subalgebras, congruences
- equations and varieties
- equational calculus
- equational specifications and initial algebras
- variations: partial algebras, first-order structures

Plus some hints on applications in

*foundations of software semantics, verification, specification, development.*
Tiny data type

Its **signature** $\Sigma$ (syntax):

<table>
<thead>
<tr>
<th>sorts</th>
<th>$\text{Int}, \text{Bool}$;</th>
</tr>
</thead>
<tbody>
<tr>
<td>opns</td>
<td>$0, 1: \text{Int}$;</td>
</tr>
<tr>
<td></td>
<td>$\text{plus}, \text{times}, \text{minus}: \text{Int} \times \text{Int} \to \text{Int}$;</td>
</tr>
<tr>
<td></td>
<td>$\text{false}, \text{true}: \text{Bool}$;</td>
</tr>
<tr>
<td></td>
<td>$\text{lteq}: \text{Int} \times \text{Int} \to \text{Bool}$;</td>
</tr>
<tr>
<td></td>
<td>$\text{not}: \text{Bool} \to \text{Bool}$;</td>
</tr>
<tr>
<td></td>
<td>$\text{and}: \text{Bool} \times \text{Bool} \to \text{Bool}$;</td>
</tr>
</tbody>
</table>

and $\Sigma$-**algebra** $\mathcal{A}$ (semantics):

<table>
<thead>
<tr>
<th>carriers</th>
<th>$\mathcal{A}<em>{\text{Int}} = \text{Int}, \mathcal{A}</em>{\text{Bool}} = \text{Bool}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>operations</td>
<td>$0_{\mathcal{A}} = 0, 1_{\mathcal{A}} = 1$</td>
</tr>
<tr>
<td></td>
<td>$\text{plus}<em>{\mathcal{A}}(n, m) = n + m, \text{times}</em>{\mathcal{A}}(n, m) = n \times m$</td>
</tr>
<tr>
<td></td>
<td>$\text{minus}_{\mathcal{A}}(n, m) = n - m$</td>
</tr>
<tr>
<td></td>
<td>$\text{false}<em>{\mathcal{A}} = \text{ff}, \text{true}</em>{\mathcal{A}} = \text{tt}$</td>
</tr>
<tr>
<td></td>
<td>$\text{lteq}_{\mathcal{A}}(n, m) = \text{tt} \text{ if } n \leq m \text{ else } \text{ff}$</td>
</tr>
<tr>
<td></td>
<td>$\text{not}_{\mathcal{A}}(b) = \text{tt} \text{ if } b = \text{ff} \text{ else } \text{ff}$</td>
</tr>
<tr>
<td></td>
<td>$\text{and}_{\mathcal{A}}(b, b') = \text{tt} \text{ if } b = b' = \text{tt} \text{ else } \text{ff}$</td>
</tr>
</tbody>
</table>
Signatures

**Algebraic signature:**

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

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

Alternatively:

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

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

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

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

- **$\Sigma$-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?
Subalgebras

- 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 $f : s_1 \times \ldots \times s_n \to s$ 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 $A \in \text{Alg}(\Sigma)$ 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 $A$. 
Homomorphisms

- for $A, B \in \text{Alg}(\Sigma)$, a $\Sigma$-homomorphism $h: A \rightarrow B$ is a function $h: |A| \rightarrow |B|$ that preserves the operations:
  
  - for $f: s_1 \times \ldots \times s_n \rightarrow s$ 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 $h: A \rightarrow B$ 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 $h: A \rightarrow B$ and $X \subseteq |A|$, $h(\langle A \rangle_X) = \langle B \rangle_{h(X)}$.

**Fact:** Identity function on the carrier of $A \in \text{Alg}(\Sigma)$ is a homomorphism $\text{id}_A: A \rightarrow A$. Composition of homomorphisms $h: A \rightarrow B$ and $g: B \rightarrow C$ is a homomorphism $h;g: A \rightarrow C$. 
Isomorphisms

- for $A, B \in \text{Alg}(\Sigma)$, 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 $i; i^{-1} = id_A$ and $i^{-1}; i = id_B$.

- $\Sigma$-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 $a \sim h(a)$ iff $h(a) = h(a')$, is a $\Sigma$-congruence on $A$. 
Quotients

- 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 $\Sigma$-homomorphism $[\_]_{\equiv} : A \to A/\equiv$.

**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 $h(A)$. 
Products

- 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":
  - 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, $\Sigma$-algebra $A$ and valuation $v : X \to |A|$, there is a unique $\Sigma$-homomorphism $v^\# : T_\Sigma(X) \to A$ that extends $v$. Moreover, for $t \in |T_\Sigma(X)|$, $v^\#(t) = t_A[v]$.

\[
\begin{array}{ccc}
X & \xrightarrow{id_X \to |T_\Sigma(X)|} & |T_\Sigma(X)| \\
\downarrow v & & \downarrow |v^\#| \\
\text{Set}^S & & T_\Sigma(X) \\
\downarrow \exists! v^\# & & \text{Alg}(\Sigma) \\
|A| & & A
\end{array}
\]
• *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

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

A \( \Sigma \)-equation \( \varphi \) is a semantic consequence of a set of \( \Sigma \)-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: \( \text{Th}(C) = \{ \varphi \mid C \models \varphi \} \)
- \( \Phi \models \varphi \iff \varphi \in \text{Th}(\text{Mod}(\Phi)) \)
- \( \text{Mod} \) and \( \text{Th} \) form a Galois connection
Equational calculus

\[
\begin{align*}
\forall X.t = t' & \quad \forall X.t' = t' \quad \forall X.t = t'' \\
\forall X.t = t & \quad \forall X.t' = t & \quad \forall X.t = t'' \\
\forall X.t_1 = t'_1 & \quad \ldots & \quad \forall X.t_n = t'_n \\
\forall X.f(t_1 \ldots t_n) = f(t'_1 \ldots t'_n) & \quad \forall X.t = t' \\
\forall Y.t[\theta] = t'[\theta]
\end{align*}
\]

for \( \theta : X \to \mathcal{T}_\Sigma(Y) \)

Mind the variables!

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

\[ \Phi \vdash_\Sigma \varphi \]

\[ \Sigma \text{-equation } \varphi \text{ is a proof-theoretic consequence of a set of } \Sigma \text{-equations } \Phi \]

if \( \varphi \) can be derived from \( \Phi \) by the rules.

How to justify this?

Semantics!
**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!
One motivation

Software systems (data types, 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 \(\sim\) signature

module \(\sim\) algebra

module specification \(\sim\) class of algebras
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**

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

Now:

\[
\text{NAIVENat} \not\models \forall n, m : \text{Nat}. n + m = m + n
\]

Perhaps worse:

There are models \( M \in \text{Mod(NAIVENat)} \) such that \( M \models 0 = \text{succ}(0) \), or even:

\[
M \models \forall n, m : \text{Nat}. n = m
\]
How to fix this

- **Constraints**: initiality: “no junk” & “no confusion”

  Also: *reachability* (“no junk”), and their more general versions (freeness, generation).

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

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

  – more about this elsewhere...

---

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.

BTW: Existence of initial (and free) models carries over to specifications with conditional equations, but not much further!
Example

\[
\text{spec } \text{Nat} = \text{initial} \{ \text{sort Nat} \\
\text{opns } 0: \text{Nat}; \\
\quad \text{succ}: \text{Nat} \rightarrow \text{Nat}; \\
\quad _+_: \text{Nat} \times \text{Nat} \rightarrow \text{Nat} \\
\text{axioms } \forall n: \text{Nat}. n + 0 = n; \\
\quad \forall n, m: \text{Nat}. n + \text{succ}(m) = \text{succ}(n + m) \}
\]

Now:

\[
\text{Nat} \models \forall n, m: \text{Nat}. n + m = m + n
\]
Try another example

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

\[
\text{opns 0: Nat; error: Nat;}
\]
\[
succ: Nat \rightarrow Nat;
\]
\[
-_+: Nat \times Nat \rightarrow Nat;
\]
\[
pred: Nat \rightarrow Nat
\]

\[
\text{axioms } \forall n:\text{Nat}. n + 0 = n;
\]
\[
\forall n, m:\text{Nat}. n + \text{succ}(m) = \text{succ}(n + m);
\]
\[
\forall n:\text{Nat}. \text{pred} (\text{succ}(n)) = n;
\]
\[
pred(0) = \text{error};
\]
\[
pred(\text{error}) = \text{error}; \text{succ}(\text{error}) = \text{error};
\]
\[
\forall n:\text{Nat}. \text{error} + n = \text{error}; \forall n:\text{Nat}. n + \text{error} = \text{error}
\]

Looks okay. But try to add multiplication:

\[
0 \ast n = 0; \text{succ}(m) \ast n = n + (m \ast n);
\]
\[
\text{error} \ast n = \text{error}; n \ast \text{error} = \text{error}
\]

\[\text{and now everything collapses!}\]
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** $h: A \rightarrow B$: map $h: |A| \rightarrow |B|$ 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 $h: |A| \rightarrowtail |B|$ 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:**

\[ \forall X.t \stackrel{\text{s}}{=} t' \]

as before

**Definedness formula:**

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

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

#### Satisfaction relation

Partial \( \Sigma \)-algebra \( A \) satisfies \( \forall X.t \stackrel{\text{s}}{=} t' \)

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

when for all \( v : X \to |A| \), \( t_A[v] \) 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 \( v : X \to |A| \), \( t_A[v] \) 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 \( \forall X.(\text{def } t \iff \text{def } t') \land (\text{def } t \Rightarrow t \overset{e}{=} t') \)
Example

```
spec NatPred = initial { sort Nat

  opns 0: Nat;

  succ: Nat \to Nat;

  _ + _: Nat \times Nat \to Nat;

  pred: Nat \to \text{? Nat}

  axioms \forall n: Nat. n + 0 = n;

  \forall n, m: Nat. n + \text{succ}(m) = \text{succ}(n + m);

  \forall n: Nat. \text{pred}(\text{succ}(n)) \Rightarrow n

}
```
First-order structures

- **First-order signature** $\Sigma = (S, \Omega, \Pi)$: algebraic signature $(S, \Omega)$ plus *predicate names*, classified by arities: $\Pi = \langle \Pi_w \rangle_{w \in S^*}$

- **First-order $\Sigma$-structure**: 

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

  consists of:

  - $(S, \Omega)$-algebra $(|A|, \langle f_A \rangle_{f \in \Omega})$
  
  - *predicates* (relations): $p_A \subseteq |A|_{s_1} \times \ldots \times |A|_{s_n}$, for $p: s_1 \times \ldots \times s_n$ (i.e., $p \in \Pi_{s_1\ldots s_n}$)

- **$\text{Str}(\Sigma)$** stands for the class of all first-order $\Sigma$-structures.
Fix a signature $\Sigma = (S, \Omega, \Pi)$ for a while.

**Few further notions**

- **substructure** $A_{sub} \subseteq A$: given by subset $|A_{sub}| \subseteq |A|$ closed under the operations and such that the inclusion preserves truth of predicates; the substructure is **closed** if the inclusion also preserves falsity of predicates;

- **homomorphism** $h: A \rightarrow B$: map $h: |A| \rightarrow |B|$ that preserves the results of operations and truth of predicates; it is **closed** if in addition it preserves falsity of predicates; (closed) homomorphisms are closed under composition;

- **congruence** $\equiv$ on $A$: equivalence $\equiv \subseteq |A| \times |A|$ closed under the operations; it is **closed** if in addition it preserves truth (and falsity) of predicates; (closed) congruences are kernels of (closed) homomorphisms;

- **quotient structures** $A/\equiv$: built in the natural way on the equivalence classes of $\equiv$ so that the natural map from $A$ to $A/\equiv$ is a homomorphism; it is closed if the congruence is closed.
Formulae

- **atomic** \( \Sigma \)-**formulae** over set \( X \) of variables:
  - \( t = t' \), where \( t, t' \in |T(s,\Omega)(X)|_s, s \in S \)
  - \( p(t_1, \ldots t_n) \), where \( p: s_1 \times \ldots \times s_n, t_1 \in |T(s,\Omega)(X)|_{s_1}, \ldots t_n \in |T(s,\Omega)(X)|_{s_n} \)
- \( \Sigma \)-**formulae** contain atomic formulae and are closed under logical connectives and quantification; \( \Sigma \)-**sentences** are \( \Sigma \)-formulae with no free variables
- *Satisfaction relation* defined as usual between \( \Sigma \)-structures \( A \) and \( \Sigma \)-sentences \( \varphi \)

\[
A \models \varphi
\]

As before, this yields the usual notions of the *class of models* for a set of sentences, the *semantic consequences* of a set of sentences, the *theory* of a class of models, etc.

Initial (and free) models exist for first-order specifications with universally quantified conditional atomic formulae, *but in general may fail to exist!*