

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* Σ (syntax):

sorts $Int, Bool$;
opns $0, 1 : Int$;
 $plus, times, minus : Int \times Int \rightarrow Int$;
 $false, true : Bool$;
 $lteq : Int \times Int \rightarrow Bool$;
 $not : Bool \rightarrow Bool$;
 $and : Bool \times Bool \rightarrow Bool$;

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

carriers $\mathcal{A}_{Int} = Int, \mathcal{A}_{Bool} = Bool$

operations

$0_{\mathcal{A}} = 0, 1_{\mathcal{A}} = 1$

$plus_{\mathcal{A}}(n, m) = n + m, times_{\mathcal{A}}(n, m) = n * m$

$minus_{\mathcal{A}}(n, m) = n - m$

$false_{\mathcal{A}} = ff, true_{\mathcal{A}} = tt$

$lteq_{\mathcal{A}}(n, m) = tt \text{ if } n \leq m \text{ else } ff$

$not_{\mathcal{A}}(b) = tt \text{ if } b = ff \text{ else } ff$

$and_{\mathcal{A}}(b, b') = tt \text{ if } b = b' = tt \text{ else } ff$

$\mathcal{A}'_{Int} = \{min, \dots, -1, 0, 1, \dots, max\} \dots$
redefines (the semantics of) TINY

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* Ω , and *arity and result sort functions*

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

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

Compare the two notions

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

Algebras

- Σ -*algebra*:

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

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

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

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 such a Σ -algebra $A_{sub} \in \mathbf{Alg}(\Sigma)$ such that $|A_{sub}| \subseteq |A|$ and
 - for $f: s_1 \times \dots \times s_n \rightarrow s$ and $a_1 \in |A_{sub}|_{s_1}, \dots, a_n \in |A_{sub}|_{s_n}$,
$$f_{A_{sub}}(a_1, \dots, a_n) = f_A(a_1, \dots, a_n)$$
(any subalgebra is given by a subset $|A_{sub}| \subseteq |A|$ closed under the operations)
- for $A \in \mathbf{Alg}(\Sigma)$ and $X \subseteq |A|$, the *subalgebra of A generated by X* , $\langle A \rangle_X$, is the least subalgebra of A that contains X .
- $A \in \mathbf{Alg}(\Sigma)$ 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 $h: A \rightarrow B$ is a function $h: |A| \rightarrow |B|$ that preserves the operations:
 - for $f: s_1 \times \dots \times s_n \rightarrow s$ and $a_1 \in |A|_{s_1}, \dots, a_n \in |A|_{s_n}$,
$$h_s(f_A(a_1, \dots, a_n)) = f_B(h_{s_1}(a_1), \dots, h_{s_n}(a_n))$$

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

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

Isomorphisms

- for $A, B \in \mathbf{Alg}(\Sigma)$, a Σ -*isomorphism* is any Σ -homomorphism $i: A \rightarrow B$ that has an *inverse*, i.e., a Σ -homomorphism $i^{-1}: B \rightarrow A$ 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 $f: s_1 \times \dots \times s_n \rightarrow s$ and $a_1, a'_1 \in |A|_{s_1}, \dots, a_n, a'_n \in |A|_{s_n}$,
if $a_1 \equiv_{s_1} a'_1, \dots, a_n \equiv_{s_n} a'_n$ then $f_A(a_1, \dots, a_n) \equiv_s f_A(a'_1, \dots, a'_n)$.

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 contains R .

Fact: For any Σ -homomorphism $h: A \rightarrow B$, 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 $f: s_1 \times \dots \times s_n \rightarrow s$ and $a_1 \in |A|_{s_1}, \dots, a_n \in |A|_{s_n}$,
$$f_{A/\equiv}([a_1]_{\equiv}, \dots, [a_n]_{\equiv}) = [f_A(a_1, \dots, a_n)]_{\equiv}$$

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

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

Fact: For any Σ -homomorphism $h: A \rightarrow B$, $A/K(h)$ 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 $f: s_1 \times \dots \times s_n \rightarrow s$ and $a_1 \in |\prod_{i \in \mathcal{I}} A_i|_{s_1}, \dots, a_n \in |\prod_{i \in \mathcal{I}} A_i|_{s_n}$, for $i \in \mathcal{I}$, $f_{\prod_{i \in \mathcal{I}} A_i}(a_1, \dots, a_n)(i) = f_{A_i}(a_1(i), \dots, a_n(i))$

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 $\pi_i: \prod_{i \in \mathcal{I}} A_i \rightarrow A_i$.

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 $f: s_1 \times \dots \times s_n \rightarrow s$ and $t_1 \in |T_\Sigma(X)|_{s_1}, \dots, t_n \in |T_\Sigma(X)|_{s_n}$,
 $f(t_1, \dots, t_n) \in |T_\Sigma(X)|_s$
- for any Σ -algebra A and valuation $v: X \rightarrow |A|$, *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 $x \in X_s$, $s \in S$
 - $(f(t_1, \dots, t_n))_A[v] = f_A((t_1)_A[v], \dots, (t_n)_A[v])$, for $f: s_1 \times \dots \times s_n \rightarrow s$ and $t_1 \in |T_\Sigma(X)|_{s_1}, \dots, t_n \in |T_\Sigma(X)|_{s_n}$

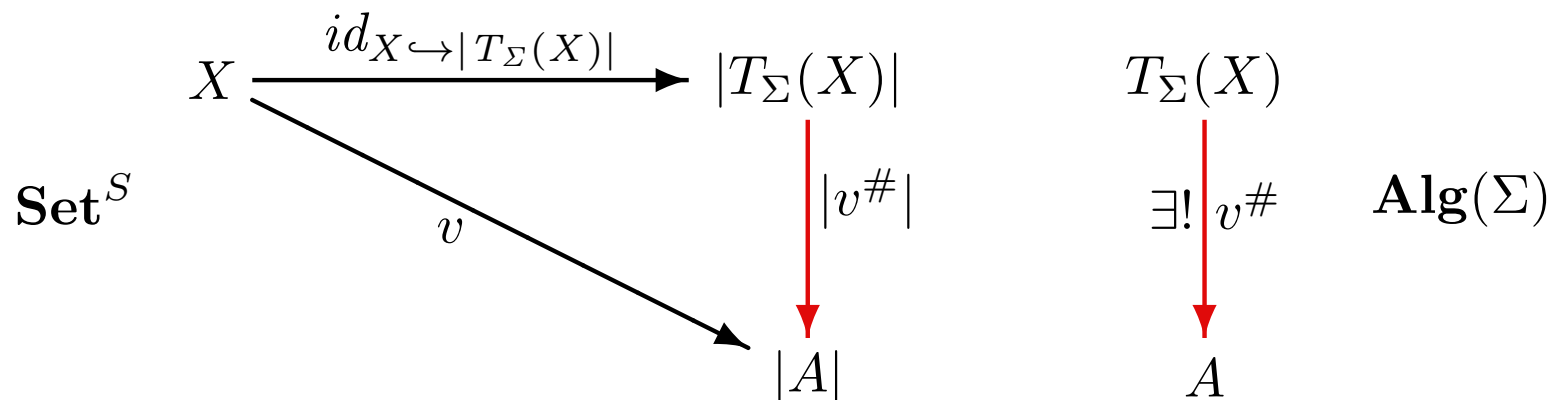
Above and in the following: assuming unambiguous “parsing” of terms!

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 $f: s_1 \times \dots \times s_n \rightarrow s$ and $t_1 \in |T_\Sigma(X)|_{s_1}, \dots, t_n \in |T_\Sigma(X)|_{s_n}$,
 $f_{T_\Sigma(X)}(t_1, \dots, t_n) = f(t_1, \dots, t_n)$.

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



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

$$A \models \forall X.t = t'$$

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

Semantic entailment

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

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 calculus

$$\begin{array}{c}
 \frac{}{\forall X.t = t} \qquad \frac{\forall X.t = t'}{\forall X.t' = t} \qquad \frac{\forall X.t = t' \quad \forall X.t' = t''}{\forall X.t = t''} \\
 \\
 \frac{\forall X.t_1 = t'_1 \quad \dots \quad \forall X.t_n = t'_n}{\forall X.f(t_1 \dots t_n) = f(t'_1 \dots t'_n)} \qquad \frac{\forall X.t = t'}{\forall Y.t[\theta] = t'[\theta]} \text{ for } \theta: X \rightarrow |T_\Sigma(Y)|
 \end{array}$$

Mind the variables!

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

Proof-theoretic entailment

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

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:*

$$\Phi \models \varphi \iff \Phi \vdash \varphi$$

- **soundness:** “all that can be proved, is true” ($\Phi \vdash \varphi \implies \Phi \models \varphi$)
- **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 \rightsquigarrow signature
module \rightsquigarrow algebra
module specification \rightsquigarrow class of algebras

Equational specifications

$$\langle \Sigma, \Phi \rangle$$

- 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.*

"if" is delicate

Equational specifications typically admit a lot of undesirable "modules"

Example

spec NAIVENAT = **sort** *Nat*

opns $0: \textit{Nat}$;

$\textit{succ}: \textit{Nat} \rightarrow \textit{Nat}$;

$- + -: \textit{Nat} \times \textit{Nat} \rightarrow \textit{Nat}$

axioms $\forall n: \textit{Nat}. n + 0 = n$;

$\forall n, m: \textit{Nat}. n + \textit{succ}(m) = \textit{succ}(n + m)$

Now:

$\text{NAIVENAT} \not\models \forall n, m: \textit{Nat}. n + m = m + n$

Perhaps worse:

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

$M \models \forall n, m: \textit{Nat}. n = m$

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. . .

Institutions!

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 $I \in \text{Mod}(\Phi)$ such that for every Σ -algebra $M \in \text{Mod}(\Phi)$ 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 $\text{Mod}(\Phi)$.

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

```
spec NAT = initial { sort Nat  
                      opns 0: Nat;  
                          succ: Nat → Nat;  
                          _ + _: Nat × Nat → Nat  
                      axioms  $\forall n:Nat. n + 0 = n$ ;  
                           $\forall n, m:Nat. n + succ(m) = succ(n + m)$   
                      }
```

Now:

$$\text{NAT} \models \forall n, m:Nat. n + m = m + n$$

Try another example

```
spec NATPRED = sort Nat  
    opns 0: Nat; error: Nat;  
        succ: Nat  $\rightarrow$  Nat;  
         $- + -$ : Nat  $\times$  Nat  $\rightarrow$  Nat;  
        pred: Nat  $\rightarrow$  Nat  
    axioms  $\forall n:Nat. n + 0 = n$ ;  
         $\forall n, m:Nat. n + succ(m) = succ(n + m)$ ;  
         $\forall n:Nat. pred(succ(n)) = n$ ;  
        pred(0) = error;  
        pred(error) = error; succ(error) = error;  
         $\forall n:Nat. error + n = error$ ;  $\forall n:Nat. n + error = error$ 
```

Looks okay. But try to add multiplication:

```
0 * n = 0; succ(m) * n = n + (m * n);  
error * n = error; n * error = error
```

and now everything collapses!

Partial algebras

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

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

as before, but operations $f_A: |A|_{s_1} \times \dots \times |A|_{s_n} \rightharpoonup |A|_s$, for $f: s_1 \times \dots \times s_n \rightarrow s$, 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* $h: A \rightarrow B$: 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 $h: |A| \rightharpoonup |B|$ 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:

$$\forall X. t \stackrel{s}{=} t'$$

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 Σ -algebra A *satisfies* $\forall X. t \stackrel{s}{=} t'$

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

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 Σ -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:

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

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

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

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

BTW:

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

Example

```
spec NATPRED = initial { sort Nat  
    opns 0: Nat;  
        succ: Nat → Nat;  
        _ + _: Nat × Nat → Nat;  
        pred: Nat →? Nat  
    axioms  $\forall n:Nat. n + 0 = n$ ;  
             $\forall n, m:Nat. n + succ(m) = succ(n + m)$ ;  
             $\forall n:Nat. pred(succ(n)) \stackrel{s}{=} n$   
}
```

First-order structures

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

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

consists of:

- (S, Ω) -algebra $(|A|, \langle f_A \rangle_{f \in \Omega})$
 - *predicates* (relations): $p_A \subseteq |A|_{s_1} \times \dots \times |A|_{s_n}$,
for $p: s_1 \times \dots \times s_n$ (i.e., $p \in \Pi_{s_1 \dots s_n}$)
- **Str**(Σ) stands for the class of all first-order Σ -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 Σ -formulae* over set X of variables:
 - $t = t'$, where $t, t' \in |T_{(S,\Omega)}(X)|_s$, $s \in S$
 - $p(t_1, \dots, t_n)$, where $p: s_1 \times \dots \times s_n$, $t_1 \in |T_{(S,\Omega)}(X)|_{s_1}$, \dots , $t_n \in |T_{(S,\Omega)}(X)|_{s_n}$
- Σ -*formulae* contain atomic formulae and are closed under logical connectives and quantification; Σ -*sentences* are Σ -formulae with no free variables
- *Satisfaction relation* defined as usual between Σ -structures A and Σ -sentences φ

$$A \models \varphi$$

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!*