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: $Mod(\Phi) = \{A \in Alg(\Sigma) \mid A \models \Phi\}$
- *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

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

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

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" ($\Phi \models \varphi \Longrightarrow \Phi \vdash \varphi$)

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

spec NaiveNat = **sort** *Nat*
ops $0 : Nat$;
$succ : Nat \to Nat$;
$\_+\_ : Nat \times Nat \to Nat$
axioms $\forall n{:}Nat \bullet n + 0 = n$;
$\forall n, m{:}Nat \bullet n + succ(m) = succ(n + m)$

Now:

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

(Nor: $\textsc{NaiveNat} \not\vdash \forall n, m{:}Nat \bullet n + m = m + n$)

How to fix this

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

Example

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

Now:

$$\textsc{Nat} \models \forall n, m{:}Nat \bullet n + m = m + n$$

Example′

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

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

Another example

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

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 $w \in S^*$, $s \in S$