An Algebraic Specification of First-Order Logic

Aad Mathijssen

10th June 2004

Motivation

Concrete datatypes for μCRL: algebraic specification of booleans, numbers, function types, sets, tables, etc.

Wish: when reading the equations from left to right, we obtain a complete, confluent and terminating rewrite system

Problem: computing the truth value of universal and existential quantifications is undecidable

Motivation (2)

First solution: leave out universal and existential quantifications

Not sufficient when we want to implement more complex data types.

Second solution: find at least a representation of quantifications

Question: can you do anything with these representations?

- rewriting: still research
- equations: completeness

Overview

- Formal introduction to algebraic specifications
- Specification of first-order logic:
  - Propositional logic
  - Binding
  - Quantifications
- Completeness

Signatures and terms

Algebraic specification: a description of a number of *abstract data types*

Signature Σ:
- set of sorts S
- set of functions F_n, of type $S^n \longrightarrow S$, for any $n \in \mathbb{N}$

Terms $T(\Sigma, X)_s$, for each set of variables X_s and all $s \in S$:
- every $x \in X_s$ is in $T(\Sigma, X)_s$
- every $c \in F_0$, of type $\longrightarrow s$, is in $T(\Sigma, X)_s$
- if all t_i are in $T(\Sigma, X)_{s_i}$ with $0 \leq i \leq n$, then $f(t_0, \ldots, t_n)$ is in $T(\Sigma, X)_s$, for all $f \in F_{n+1}$, of type $s_0 \times \cdots \times s_n \longrightarrow s$, and $n \in \mathbb{N}$

Σ-equations and validity

Σ-equations are of following form, for sort $s \in S$ and terms $t, u \in T(\Sigma, X)_s$:

$$t =_s u$$

Validity under a set of Σ-equations E:

$$\models_E t =_s u$$

Equivalent to: for all computation structures A of Σ and valuations v of X

$$[\![t]\!]_v^A = [\![u]\!]_v^A$$

Derivability

Derivability under a set of Σ-equations E:

$$\vdash_{E} t =_s u$$

Axioms:

(axiom)	$\vdash_{E} e$, for all $e \in E$

Inference rules:

(reflexivity)	$\vdash_{E} t =_s t$
(symmetry)	if $\vdash_{E} t =_s u$ then $\vdash_{E} u =_s t$
(transitivity)	if $\vdash_{E} t =_s u$ and $\vdash_{E} u =_s v$ then $\vdash_{E} t =_s v$
(congruence)	if $\vdash_{E} t_i =_{s_i} u_i$ then $\vdash_{E} f(t_0, \ldots, t_n) =_s f(u_0, \ldots, u_n)$
	for all $f \in F_{n+1}$, of type $s_0 \times \cdots \times s_n \longrightarrow s$, and $n \in \mathbb{N}$
(substitution)	if $\vdash_{E} t =_s u$ then $\vdash_{E} t[x := v] =_s u[x := v]$

Derivability (2)

Contextual congruence:
For all terms $t, u \in T(\Sigma, X)_s$ and contexts C of sort s':

$$\text{if } \vdash_{\mathrm{E}} t =_s u \text{ then } \vdash_{\mathrm{E}} C[t]_s =_{s'} C[u]_s$$

Calculational derivation:

$$\begin{array}{ll} & C[t]_s \\ =_{s'} & \{ \text{hint why} \vdash_{\mathrm{E}} t =_s u \} \\ & C[u]_s \\ =_{s'} & \{ \text{hint why} \vdash_{\mathrm{E}} u =_s v \} \\ & C[v]_s \end{array}$$

Justification of $\vdash_{\mathrm{E}} C[t]_s =_{s'} C[v]_s$.

Soundness and completeness

Soundness:
For all Σ-equations e and sets of Σ-equations E:

$$\text{if } \vdash_{\mathrm{E}} e \text{ then } \models_{\mathrm{E}} e$$

Completeness:
For all Σ-equations e and sets of Σ-equations E:

$$\text{if } \models_{\mathrm{E}} e \text{ then } \vdash_{\mathrm{E}} e$$

Propositional logic

Sorts: S contains $\mathbb{B}$

Functions: F_0 contains $\mathit{true}, \mathit{false} :\longrightarrow \mathbb{B}$

F_1 contains $\neg : \mathbb{B} \longrightarrow \mathbb{B}$

F_2 contains $\wedge, \vee, \Rightarrow, \Leftrightarrow : \mathbb{B} \times \mathbb{B} \longrightarrow \mathbb{B}$

Σ-equations: E contains, for certain variables $p, q, r \in X_{\mathbb{B}}$:

$$
\begin{array}{llll}
\neg\neg p =_{\mathbb{B}} p & \qquad & p \wedge \neg p =_{\mathbb{B}} \mathit{false} \\
p \wedge \mathit{false} =_{\mathbb{B}} \mathit{false} & & p \vee q =_{\mathbb{B}} \neg(\neg p \wedge \neg q) \\
p \wedge \mathit{true} =_{\mathbb{B}} p & & p \wedge (q \vee r) =_{\mathbb{B}} (p \wedge q) \vee (p \wedge r) \\
(p \wedge q) \wedge r =_{\mathbb{B}} p \wedge (q \wedge r) & & p \Rightarrow q =_{\mathbb{B}} \neg p \vee q \\
p \wedge q =_{\mathbb{B}} q \wedge p & & p \Leftrightarrow q =_{\mathbb{B}} (p \Rightarrow q) \wedge (q \Rightarrow p) \\
p \wedge p =_{\mathbb{B}} p & &
\end{array}
$$

The above equations can be derived for any term $p, q, r \in T(\Sigma, X)_{\mathbb{B}}$.
All desirable properties of true, false, $\vee$, $\Rightarrow$ and $\Leftrightarrow$ can be derived.

Conditional equations

For every $s \in S$:

- F_3 contains $ite : \mathbb{B} \times s \times s \rightarrow s$
- E contains, for certain variables $x, y \in X_{\mathbb{B}}$:

$$ite(true, x, y) =_s x$$
$$ite(false, x, y) =_s y$$

Equations of the form

$$t =_s ite(b, u, t)$$

are abbreviated to

$$t =_s u, \text{ if } b$$

This abbreviation will also be used in the context of the derivation symbol $\vdash_E$ and in derivations.

Binding

Lambda calculus with abstractions to booleans only.

S is partitioned into disjoint sets S_0 and $S_{0\to\mathbb{B}}$ of equal size, with:

- every $s \in S_0$ has a corresponding sort $s \to \mathbb{B} \in S_{0\to\mathbb{B}}$
- S_0 does not contain sorts with suffix $\to \mathbb{B}$

Basic elements, for all sorts $s \in S_0$:

- variables of sort s
- abstractions of terms in $T(\Sigma, X)_{\mathbb{B}}$ over terms in $T(\Sigma, X)_s$
- applications of terms in $T(\Sigma, X)_{s\to\mathbb{B}}$ to terms in $T(\Sigma, X)_s$

Basic elements

Positive numbers:

- S_0 contains $\mathbb{N}^+$
- functions $1 :\longrightarrow \mathbb{N}^+$, $+1 : \mathbb{N}^+ \longrightarrow \mathbb{N}^+$ and $eq : \mathbb{N}^+ \times \mathbb{N}^+ \longrightarrow \mathbb{B}$

We assume the following functions, for each $s \in S_0$:

variables: var_s of type $\mathbb{N}^+ \longrightarrow s$
abstraction: $\lambda_s\_.\_$ of type $\mathbb{N}^+ \times \mathbb{B} \longrightarrow (s \longrightarrow \mathbb{B})$
application: $\_.\_$ of type $(s \longrightarrow \mathbb{B}) \times s \longrightarrow \mathbb{B}$

$var_s(m)$ will be written as $\underline{m}_s$
$\lambda_s m.p$ binds all free variables $\underline{n}_s$ in p, where $\vdash_{\mathrm{E}} eq(m,n) =_{\mathbb{B}} true$

α-conversion and β-reduction need substitution.

Substitution

Substitution needs a way to:

- tell if variables occur free in terms
- calculate fresh variables

We assume the following functions, for each $s \in S_0$ and $s' \in S$:

substitution:	$\_[\_/\_]$ of type $s' \times s \times \mathbb{N}^+ \longrightarrow s'$
free occurrences:	occ_s of type $\mathbb{N}^+ \times s' \longrightarrow \mathbb{B}$
fresh variables:	$fresh$ of type $s \times \mathbb{B} \longrightarrow \mathbb{N}^+$
	$gfresh$ of type $\mathbb{N}^+ \times s \times \mathbb{B} \longrightarrow \mathbb{N}^+$

Substitution (2)

E contains, for all $s \in S_0$ and $s' \in S$, certain $t \in X_s$, $m, n \in X_{\mathbb{N}^+}$ and $p \in X_{\mathbb{B}}$, and all $c \in F_0$ of type $\rightarrow s'$, and $f \in F_{k+1}$ of type $t_0 \times \cdots \times t_k \rightarrow s'$, except for variables and lambda abstractions, for all $k \in \mathbb{N}$ and $s'' \in S_0$ different from s:

$$
\begin{array}{lll}
\underline{n}_s[t/m] & =_s & t, \quad \text{if } eq(m,n) \\
\underline{n}_s[t/m] & =_s & \underline{n}_s, \quad \text{if } \neg eq(m,n) \\
(\lambda_s n.p)[t/m] & =_{s\rightarrow\mathbb{B}} & \lambda_s n.p, \quad \text{if } eq(m,n) \\
(\lambda_s n.p)[t/m] & =_{s\rightarrow\mathbb{B}} & \lambda_s n.p, \quad \text{if } \neg eq(m,n) \wedge \neg occ_s(m,p) \\
(\lambda_s n.p)[t/m] & =_{s\rightarrow\mathbb{B}} & \lambda_s n.p[t/m], \\
 & & \text{if } (\neg eq(m,n) \wedge occ_s(m,p)) \wedge \neg occ_s(n,t) \\
(\lambda_s n.p)[t/m] & =_{s\rightarrow\mathbb{B}} & \lambda_s fresh(t,p).p[\underline{fresh(t,p)}_s/n][t/m], \\
 & & \text{if } (\neg eq(m,n) \wedge occ_s(m,p)) \wedge occ_s(n,t) \\
\underline{n}_{s''}[t/m] & =_{s''} & \underline{n}_{s''} \\
(\lambda_{s''} n.p)[t/m] & =_{s''\rightarrow\mathbb{B}} & \lambda_{s''} n.p[t/m] \\
c[t/m] & =_{s'} & c \\
f(t_0,\ldots,t_k)[t/m] & =_{s'} & f(t_0[t/m],\ldots,t_k[t/m])
\end{array}
$$

α-conversion and β-reduction

E contains, for certain $m, n \in X_{\mathbb{N}^+}$, $p \in X_{\mathbb{B}}$ and $t \in X_s$:

$$\begin{array}{lll} \lambda_s m.p & =_{s\to\mathbb{B}} & \lambda_s n.p[\underline{n}_s/m], \text{ if } \neg occ_s(n,p) \\ (\lambda_s m.p).t & =_{\mathbb{B}} & p[t/m] \end{array}$$

Example: For all $m, n \in T(\Sigma, X)_{\mathbb{N}^+}$, the term $\lambda_{\mathbb{B}} m.(\lambda_{\mathbb{B}} n.(\underline{n}_{\mathbb{B}} \wedge \underline{m}_{\mathbb{B}})).\underline{m}_{\mathbb{B}}$ expresses the identity function of sort $\mathbb{B}$.

Question: Is this representation of the lambda calculus confluent and terminating?

Quantifications

Functions $\forall$ and $\exists$, both of type $(s \to \mathbb{B}) \to \mathbb{B}$, for all $s \in S_0$.

Terms of the form $\forall(\lambda_s m.p)$ and $\exists(\lambda_s m.p)$ are abbreviated to $\forall_s m.p$ and $\exists_s m.p$.

E contains, for certain $m \in X_{\mathbb{N}^+}$, $p, q \in X_{\mathbb{B}}$ and $t \in X_s$:

$$
\begin{array}{lll}
\forall_s m.\mathit{false} & =_{\mathbb{B}} \mathit{false} & \\
\forall_s m.p & =_{\mathbb{B}} \forall_s m.p \wedge p[t/m] & \\
\forall_s m.(p \wedge q) & =_{\mathbb{B}} \forall_s m.p \wedge \forall_s m.q & \\
\forall_s m.(p \vee q) & =_{\mathbb{B}} p \vee \forall_s m.q, & \text{if } \neg occ_s(m,p) \\
\exists_s m.p & =_{\mathbb{B}} \neg \forall_s m.\neg p &
\end{array}
$$

The definition of the first equation is correct, because every sort s is sensible, i.e. it has at least one ground term.

Quantifications (2)

Lemma's, for all $s \in S_0$ and certain $m, n \in X_{\mathbb{N}^+}$, $p, q \in X_{\mathbb{B}}$ and $t \in X_s$:

$$
\begin{array}{lll}
\vdash_{\mathrm{E}} \forall_s m.p & =_{\mathbb{B}} \forall_s n.p[\underline{n}_s/m], & \text{if } \neg occ_s(n,p) \\
\vdash_{\mathrm{E}} \forall_s m.p & =_{\mathbb{B}} p, & \text{if } \neg occ_s(m,p) \\
\vdash_{\mathrm{E}} \exists_s m.true & =_{\mathbb{B}} true & \\
\vdash_{\mathrm{E}} \exists_s m.p & =_{\mathbb{B}} \exists_s m.p \vee p[t/m] & \\
\vdash_{\mathrm{E}} \exists_s m.(p \vee q) & =_{\mathbb{B}} \exists_s m.p \vee \exists_s m.q & \\
\vdash_{\mathrm{E}} \exists_s m.(p \wedge q) & =_{\mathbb{B}} p \wedge \exists_s m.q, & \text{if } \neg occ_s(m,p) \\
\vdash_{\mathrm{E}} \exists_s m.p & =_{\mathbb{B}} \exists_s n.p[\underline{n}_s/m], & \text{if } \neg occ_s(n,p) \\
\vdash_{\mathrm{E}} \exists_s m.p & =_{\mathbb{B}} p, & \text{if } \neg occ_s(m,p) \\
\vdash_{\mathrm{E}} p[t/m] & =_{\mathbb{B}} p[t/m] \wedge \exists_s m.p &
\end{array}
$$

Completeness

Theorem: For all $p \in T(\Sigma, X)_{\mathbb{B}}$:

$$\text{if } \models_{\mathrm{E}} p =_{\mathbb{B}} \mathit{true} \text{ then } \vdash_{\mathrm{E}} p =_{\mathbb{B}} \mathit{true}$$

Indirect proof: For all formulae ϕ and sequents Γ from the system of natural deduction *ND* and a certain translation function *tr*:

$$\begin{array}{ccc} \vdash_{\mathrm{E}} tr(\Gamma, \phi) & & \models_{\mathrm{E}} tr(\Gamma, \phi) \\ \text{`}\Uparrow\text{' (C1)} & & \text{`}\Downarrow\text{' (C2)} \\ \Gamma \vdash_{\mathrm{ND}} \phi & \text{`}\Leftarrow\text{' (C3)} & \Gamma \models_{\mathrm{ND}} \phi \end{array}$$

There must exist a corresponding $tr(\Gamma, \phi)$ for each p

Natural deduction: formulae

Formulae are built from the following elements:

- a set of domains Dom;
- a set of free variables FV and a set of bound variables BV;
- a set of predicates $Pred$ ranging over elements from $FV \cup BV$;
- the symbols $true$, $false$, $\neg$, $\wedge$, $\vee$, $\Rightarrow$ and $\Leftrightarrow$;
- quantifier symbols $\forall$ and $\exists$;
- punctuation symbols (and).

Inductive definition of $Form$:

- $P(a_0, \ldots, a_n) \in Form$, for any n-ary $P \in Pred$ and $a_i \in FV$;
- $true, false \in Form$;
- if $\phi \in Form$, then $\neg\phi \in Form$;
- if $\phi, \psi \in Form$, then $(\phi \wedge \psi), (\phi \vee \psi), (\phi \Rightarrow \psi), (\phi \Leftrightarrow \psi) \in Form$;
- if $\phi \in Form$, then $\forall x_d.\phi[a_d := x_d], \exists x_d.\phi[a_d := x_d] \in Form$, for any $a_d \in FV_d$, $x_d \in FV_d$ and $d \in Dom$.

Natural deduction: sequents and validity

The set of sequents is $\mathcal{P}(Form)$.

Definition of validity $\models_{\text{ND}}$, for all 'consistent' $\Gamma \in \mathcal{P}(Form)$, $\phi, \psi \in Form$, $P \in Pred$, $d \in Dom$ and $x_d \in BV_d$:

$\Gamma \models_{\text{ND}} P$	iff $I_\Gamma(P) = true$;
$\Gamma \models_{\text{ND}} true$	is valid;
$\Gamma \models_{\text{ND}} false$	is not valid;
$\Gamma \models_{\text{ND}} \neg\phi$	iff $\Gamma \models_{\text{ND}} \phi$ is not valid;
$\Gamma \models_{\text{ND}} \phi \wedge \psi$	iff both $\Gamma \models_{\text{ND}} \phi$ and $\Gamma \models_{\text{ND}} \psi$ are valid;
$\Gamma \models_{\text{ND}} \phi \vee \psi$	iff either $\Gamma \models_{\text{ND}} \phi$ or $\Gamma \models_{\text{ND}} \psi$ is valid, or both;
$\Gamma \models_{\text{ND}} \phi \Rightarrow \psi$	iff either $\Gamma \models_{\text{ND}} \phi$ is not valid or $\Gamma \models_{\text{ND}} \psi$ is valid, or both;
$\Gamma \models_{\text{ND}} \phi \Leftrightarrow \psi$	iff both $\Gamma \models_{\text{ND}} \phi$ and $\Gamma \models_{\text{ND}} \psi$ are either valid or not valid;
$\Gamma \models_{\text{ND}} \forall x_d.\phi$	iff $\Gamma \models_{\text{ND}} \phi[x_d := a_d]$ is valid for any $a_d \in FV_d$;
$\Gamma \models_{\text{ND}} \exists x_d.\phi$	iff $\Gamma \models_{\text{ND}} \phi[x_d := a_d]$ is valid for at least one $a_d \in FV_d$.

Natural deduction: derivability

Definition of $\vdash_{\text{ND}}$, for all $\Gamma, \Delta \in \mathcal{P}(\mathit{Form})$ and $\phi, \psi \in \mathit{Form}$:

Axiom: $\phi, \Gamma \vdash_{\text{ND}} \phi$

Thinning: $\dfrac{\Gamma \vdash_{\text{ND}} \phi}{\Gamma, \Delta \vdash_{\text{ND}} \phi}$

(false): $\dfrac{\Gamma \vdash_{\text{ND}} \mathit{false}}{\Gamma \vdash_{\text{ND}} \phi}$

$(\neg\ \mathrm{I})$: $\dfrac{\phi, \Gamma \vdash_{\text{ND}} \psi \qquad \phi, \Gamma \vdash_{\text{ND}} \neg\psi}{\Gamma \vdash_{\text{ND}} \neg\phi}$

$(\neg\ \mathrm{E})$: $\dfrac{\Gamma \vdash_{\text{ND}} \neg\neg\phi}{\Gamma \vdash_{\text{ND}} \phi}$

Natural deduction: derivability (2)

For all $\Gamma \in \mathcal{P}(Form)$ and $\phi, \phi_0, \phi_1, \psi, \theta \in Form$:

$$(\wedge\ \mathrm{I}):\ \frac{\Gamma \vdash_{\mathrm{ND}} \phi \qquad \Gamma \vdash_{\mathrm{ND}} \psi}{\Gamma \vdash_{\mathrm{ND}} \phi \wedge \psi}$$

$$(\wedge\ \mathrm{E}):\ \frac{\Gamma \vdash_{\mathrm{ND}} \phi_0 \wedge \phi_1}{\Gamma \vdash_{\mathrm{ND}} \phi_i}\ , i = 0, 1$$

$$(\vee\ \mathrm{I}):\ \frac{\Gamma \vdash_{\mathrm{ND}} \phi_i}{\Gamma \vdash_{\mathrm{ND}} \phi_0 \vee \phi_1}\ , i = 0, 1$$

$$(\vee\ \mathrm{E}):\ \frac{\Gamma \vdash_{\mathrm{ND}} \phi \vee \psi \qquad \phi, \Gamma \vdash_{\mathrm{ND}} \theta \qquad \psi, \Gamma \vdash_{\mathrm{ND}} \theta}{\Gamma \vdash_{\mathrm{ND}} \theta}$$

Natural deduction: derivability (3)

For all $\Gamma \in \mathcal{P}(Form)$ and $\phi, \phi_0, \phi_1, \psi \in Form$:

(⇒ I): $$\frac{\phi, \Gamma \vdash_{\mathrm{ND}} \psi}{\Gamma \vdash_{\mathrm{ND}} \phi \Rightarrow \psi}$$

(⇒ E): $$\frac{\Gamma \vdash_{\mathrm{ND}} \phi \Rightarrow \psi \qquad \Gamma \vdash_{\mathrm{ND}} \phi}{\Gamma \vdash_{\mathrm{ND}} \psi}$$

(⇔ I): $$\frac{\phi, \Gamma \vdash_{\mathrm{ND}} \psi \qquad \psi, \Gamma \vdash_{\mathrm{ND}} \phi}{\Gamma \vdash_{\mathrm{ND}} \phi \Leftrightarrow \psi}$$

(⇔ E): $$\frac{\Gamma \vdash_{\mathrm{ND}} \phi_0 \Leftrightarrow \phi_1}{\phi_i, \Gamma \vdash_{\mathrm{ND}} \phi_{1-i}}\ , i = 0, 1$$

Natural deduction: derivability (4)

For all $\Gamma \in \mathcal{P}(Form)$, $\phi, \psi \in Form$, $d \in Dom$, $x_d \in BV_d$ and $a_d \in FV_d$:

(∀ I): $\dfrac{\Gamma \vdash_{\text{ND}} \phi}{\Gamma \vdash_{\text{ND}} \forall x_d.\phi[a_d := x_d]}$, where a_d does not occur in Γ

(∀ E): $\dfrac{\Gamma \vdash_{\text{ND}} \forall x_d.\phi}{\Gamma \vdash_{\text{ND}} \phi[x_d := a_d]}$

(∃ I): $\dfrac{\Gamma \vdash_{\text{ND}} \phi[x_d := a_d]}{\Gamma \vdash_{\text{ND}} \exists x_d.\phi}$

(∃ E): $\dfrac{\Gamma \vdash_{\text{ND}} \exists x_d.\phi \qquad \phi[x_d := a_d], \Gamma \vdash_{\text{ND}} \psi}{\Gamma \vdash_{\text{ND}} \psi}$, where a_d does not occur in Γ, ϕ, ψ

Natural deduction: soundness and completeness

Soundness:
For all $\Gamma \in \mathcal{P}(Form)$ and $\phi \in Form$:

$$\text{if } \Gamma \vdash_{\text{ND}} \phi \text{ then } \Gamma \models_{\text{ND}} \phi$$

Completeness:
For all $\Gamma \in \mathcal{P}(Form)$ and $\phi \in Form$:

$$\text{if } \Gamma \models_{\text{ND}} \phi \text{ then } \Gamma \vdash_{\text{ND}} \phi$$

This is proof obligation (C3).

Translation from *ND* to *E*

Assumptions:

- every domain $d \in Dom$ has a corresponding sort $s^d \in S_0$;
- every variable $u_d \in FV_d \cup BV_d$ has a corresponding term $\underline{m^u}_{s^d}$, where $m^u \in T(\Sigma, X)_{\mathbb{N}^+}$, for any $d \in Dom$;
- every predicate $P \in Pred$ has a corresponding term $b^P \in T(\Sigma, X)_{\mathbb{B}}$.

Translation of formulae, for all $\phi, \psi \in Form$, $P \in Pred$, $d \in Dom$ and $x_d \in BV_d$:

$$
\begin{array}{llcll}
\overline{P} & =_{\mathbb{B}} b^P & \quad & \overline{\phi \vee \psi} & =_{\mathbb{B}} \overline{\phi} \vee \overline{\psi} \\
\overline{true} & =_{\mathbb{B}} true & & \overline{\phi \Rightarrow \psi} & =_{\mathbb{B}} \overline{\phi} \Rightarrow \overline{\psi} \\
\overline{false} & =_{\mathbb{B}} false & & \overline{\phi \Leftrightarrow \psi} & =_{\mathbb{B}} \overline{\phi} \Leftrightarrow \overline{\psi} \\
\overline{\neg \phi} & =_{\mathbb{B}} \neg \overline{\phi} & & \overline{\forall x_d . \phi} & =_{\mathbb{B}} \forall_{s^d} m^x . \overline{\phi} \\
\overline{\phi \wedge \psi} & =_{\mathbb{B}} \overline{\phi} \wedge \overline{\psi} & & \overline{\exists x_d . \phi} & =_{\mathbb{B}} \exists_{s^d} m^x . \overline{\phi}
\end{array}
$$

Translation from *ND* to *E* (2)

Property, for all $\phi \in Form$, $d \in Dom$ and $u_d, v_d \in FV_d \cup BV_d$:

$$\overline{\phi[u_d := v_d]} =_{\mathbb{B}} \overline{\phi}[\underline{m^v}_{s^d}/m^u]$$

Translation of sequents, for all $\Gamma \in \mathcal{P}(Form)$:

$$\overline{\emptyset} =_{\mathbb{B}} true \qquad \overline{\phi, \Gamma} =_{\mathbb{B}} \overline{\phi} \wedge \overline{\Gamma}$$

Definition of *tr*:

$$tr(\Gamma, \phi) \quad \text{iff} \quad \overline{\Gamma} =_{\mathbb{B}} \overline{\phi, \Gamma}$$

Translation of 'variable u_d may not occur in the sequent Γ':

$$\vdash_{\mathrm{E}} \neg occ_{s^d}(m^u, \overline{\Gamma}) =_{\mathbb{B}} true$$

Proof of (C1)

We need to prove, for all $\Gamma \in \mathcal{P}(Form)$ and $\phi \in Form$:

$$\text{if} \quad \Gamma \vdash_{\text{ND}} \phi \quad \text{then} \quad \vdash_{\text{E}} \overline{\Gamma} =_{\mathbb{B}} \overline{\phi} \wedge \overline{\Gamma}$$

Proof by induction on the structure of the derivability relation $\vdash_{\text{ND}}$.

Proof of (C2)

We need to prove, for all $\Gamma \in \mathcal{P}(Form)$ and $\phi \in Form$:

$$\text{if} \quad \models_{\text{E}} \overline{\Gamma} =_{\mathbb{B}} \overline{\phi} \wedge \overline{\Gamma} \quad \text{then} \quad \Gamma \models_{\text{ND}} \phi$$

Proof is still under construction.
Idea: proof by induction on the structure of ϕ.

Conclusions

Algebraic specification is a powerful formalism for specifying data types and properties of functions on these data types.

It is possible to specify a complete first-order logic.

Consequence of the proof of (C1): we can adopt every proved case of the proof as a lemma, such that we get more intuitive derivations of boolean Σ-equations.

The main problem of quantifications lies in the binding.

Use the results of the implementation of the lambda calculus and quantifications to implement other data types such as sets and tables.