

# An Algebraic Specification of First-Order Logic

Aad Mathijssen

10th June 2004

# Motivation

Concrete datatypes for  $\mu$ CRL: 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

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  $\Sigma$ :

- set of sorts S
- set of functions  $F_n$ , of type  $S^n \to S$ , 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  $\rightarrow s$ , is in  $T(\Sigma, X)_s$
- if all  $t_i$  are in  $T(\Sigma, X)_{s_i}$  with  $0 \le i \le n$ , then  $f(t_0, \ldots, t_n)$  is in  $T(\Sigma, X)_s$ , for all  $f \in F_{n+1}$ , of type  $s_0 \times \cdots \times s_n \to s$ , and  $n \in \mathbb{N}$

# **TU/e** technische universiteit eindhoven $\Sigma$ -equations and validity

 $\Sigma$ -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  $\Sigma$ -equations E:

 $\models_{\scriptscriptstyle \rm E} t =_s u$ 

Equivalent to: for all computation structures A of  $\Sigma$  and valuations v of X  $[\![t]]^A_v = [\![u]]^A_v$ 

# **TU/e** technische universiteit eindhoven Derivability

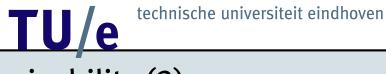
*Derivability* under a set of  $\Sigma$ -equations E:

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

Axioms:

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

### Inference rules: (reflexivity) $\vdash_{E} t =_{s} t$ (symmetry) $\text{if} \vdash_{E} t =_{s} u \text{ then} \vdash_{E} u =_{s} t$ (transitivity) $\text{if} \vdash_{E} t =_{s} u \text{ and} \vdash_{E} u =_{s} v \text{ then} \vdash_{E} t =_{s} v$ (congruence) $\text{if} \vdash_{E} t_{i} =_{s_{i}} u_{i} \text{ then} \vdash_{E} f(t_{0}, \dots, t_{n}) =_{s} f(u_{0}, \dots, u_{n})$ for all $f \in F_{n+1}$ , of type $s_{0} \times \dots \times s_{n} \to s$ , and $n \in \mathbb{N}$ (substitution) $\text{if} \vdash_{E} t =_{s} u \text{ then} \vdash_{E} t[x := v] =_{s} u[x := v]$



# Derivability (2)

#### Contextual congruence:

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

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

#### **Calculational derivation:**

$$C[t]_{s}$$

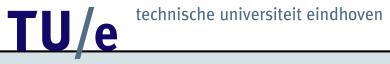
$$=_{s'} \{ \text{ hint why} \vdash_{E} t =_{s} u \}$$

$$C[u]_{s}$$

$$=_{s'} \{ \text{ hint why} \vdash_{E} u =_{s} v \}$$

$$C[v]_{s}$$

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



### Soundness and completeness

#### Soundness:

For all  $\Sigma$ -equations e and sets of  $\Sigma$ -equations E:

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

#### **Completeness:**

For all  $\Sigma$ -equations e and sets of  $\Sigma$ -equations E:

 $\text{if }\models_{\scriptscriptstyle \rm E} e \text{ then } \vdash_{\scriptscriptstyle \rm E} e$ 

# Propositional logic

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

## **Conditional equations**

For every  $s \in S$ :

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

$$\begin{array}{l} ite(true, x, y) =_s x \\ ite(false, x, y) =_s y \end{array}$$

Equations of the form

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

are abbreviated to

$$t =_s u$$
, if  $b$ 

This abbreviation will also be used in the context of the derivation symbol  $\vdash_{_{\rm E}}$  and in derivations.

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  $\rightarrow \mathbb{B}$

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  $1:\to \mathbb{N}^+$ ,  $+1:\mathbb{N}^+\to \mathbb{N}^+$  and  $eq:\mathbb{N}^+\times \mathbb{N}^+\to \mathbb{B}$

We assume the following functions, for each  $s \in S_0$ : variables:  $var_s$  of type  $\mathbb{N}^+ \to s$ abstraction:  $\lambda_{s-\cdot}$  of type  $\mathbb{N}^+ \times \mathbb{B} \to (s \to \mathbb{B})$ application:  $\_.\_$  of type  $(s \to \mathbb{B}) \times s \to \mathbb{B}$ 

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

 $\alpha\text{-conversion}$  and  $\beta\text{-reduction}$  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  $s' \times s \times \mathbb{N}^+ \to s'$ free occurrences:  $occ_s$  of type  $\mathbb{N}^+ \times s' \to \mathbb{B}$ fresh variables: fresh of type  $s \times \mathbb{B} \to \mathbb{N}^+$ qfresh of type  $\mathbb{N}^+ \times s \times \mathbb{B} \to \mathbb{N}^+$ 

# 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  $\to s'$ , and  $f \in F_{k+1}$  of type  $t_0 \times \cdots \times t_k \to s'$ , except for variables and lambda abstractions, for all  $k \in \mathbb{N}$  and  $s'' \in S_0$  different from s:

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

# **TU/e** technische universiteit eindhoven $\alpha$ -conversion and $\beta$ -reduction

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

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

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?

# **TU/e** technische universiteit eindhoven 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  $t\in X_s$ :

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

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

# **TU/e** technische universiteit eindhoven 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  $t \in X_s$ :

$$\begin{array}{lll} \vdash_{_{\mathrm{E}}} \forall_{s}m.p &=_{\mathbb{B}} \forall_{s}n.p[\underline{n}_{s}/m], & \text{if} \neg occ_{s}(n,p) \\ \vdash_{_{\mathrm{E}}} \forall_{s}m.p &=_{\mathbb{B}} p, & \text{if} \neg occ_{s}(m,p) \\ \vdash_{_{\mathrm{E}}} \exists_{s}m.true &=_{\mathbb{B}} true \\ \vdash_{_{\mathrm{E}}} \exists_{s}m.p &=_{\mathbb{B}} \exists_{s}m.p \lor p[t/m] \\ \vdash_{_{\mathrm{E}}} \exists_{s}m.(p \lor q) =_{\mathbb{B}} \exists_{s}m.p \lor \exists_{s}m.q \\ \vdash_{_{\mathrm{E}}} \exists_{s}m.(p \land q) =_{\mathbb{B}} p \land \exists_{s}m.q, & \text{if} \neg occ_{s}(m,p) \\ \vdash_{_{\mathrm{E}}} \exists_{s}m.p &=_{\mathbb{B}} \exists_{s}n.p[\underline{n}_{s}/m], & \text{if} \neg occ_{s}(n,p) \\ \vdash_{_{\mathrm{E}}} \exists_{s}m.p &=_{\mathbb{B}} p, & \text{if} \neg occ_{s}(m,p) \\ \vdash_{_{\mathrm{E}}} \exists_{s}m.p &=_{\mathbb{B}} p, & \text{if} \neg occ_{s}(m,p) \\ \vdash_{_{\mathrm{E}}} p[t/m] &=_{\mathbb{B}} p[t/m] \land \exists_{s}m.p \end{array}$$

## Completeness

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

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

**Indirect proof:** For all formulae  $\phi$  and sequents  $\Gamma$  from the system of natural deduction *ND* and a certain translation function *tr*:

| $\vdash_{\scriptscriptstyle \mathrm{E}} tr(\Gamma,\phi)$ |          | $\models_{\scriptscriptstyle \rm E} tr(\Gamma,\phi)$ |
|--|----------|--|
| '솪' (Cɪ)   |          | ' <b>∜' (C</b> 2)                                    |
| $\Gamma \vdash_{\mathrm{ND}} \phi$                       | '⇐' (C3) | $\Gamma\models_{\rm ND}\phi$                         |

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*,  $\neg$ ,  $\land$ ,  $\lor$ ,  $\Rightarrow$  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  $(\phi \land \psi), (\phi \lor \psi), (\phi \Rightarrow \psi), (\phi \Leftrightarrow \psi) \in Form$ ;
- 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

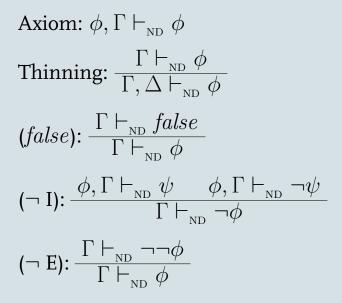
The set of sequents is  $\mathcal{P}(Form)$ .

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  $x_d \in BV_d$ :

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

## Natural deduction: derivability

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



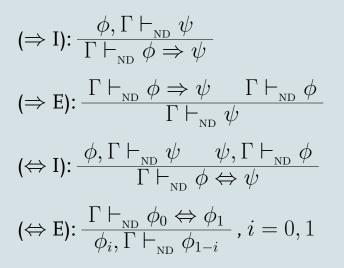
### Natural deduction: derivability (2)

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

$$(\land \mathbf{I}): \frac{\Gamma \vdash_{_{\mathrm{ND}}} \phi \quad \Gamma \vdash_{_{\mathrm{ND}}} \psi}{\Gamma \vdash_{_{\mathrm{ND}}} \phi \land \psi}$$
$$(\land \mathbf{E}): \frac{\Gamma \vdash_{_{\mathrm{ND}}} \phi_0 \land \phi_1}{\Gamma \vdash_{_{\mathrm{ND}}} \phi_i}, i = 0, 1$$
$$(\lor \mathbf{I}): \frac{\Gamma \vdash_{_{\mathrm{ND}}} \phi_i}{\Gamma \vdash_{_{\mathrm{ND}}} \phi_0 \lor \phi_1}, i = 0, 1$$
$$(\lor \mathbf{E}): \frac{\Gamma \vdash_{_{\mathrm{ND}}} \phi \lor \psi}{\Gamma \vdash_{_{\mathrm{ND}}} \phi \lor \psi} \quad \phi, \Gamma \vdash_{_{\mathrm{ND}}} \theta \quad \psi, \Gamma \vdash_{_{\mathrm{ND}}} \theta}{\Gamma \vdash_{_{\mathrm{ND}}} \theta}$$

## Natural deduction: derivability (3)

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



# **TU/e** technische universiteit eindhoven Natural deduction: derivability (4)

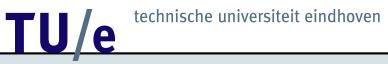
For all  $\Gamma \in \mathcal{P}(Form)$ ,  $\phi, \psi \in Form$ ,  $d \in Dom$ ,  $x_d \in BV_d$  and  $a_d \in FV_d$ :

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

$$(\forall E): \frac{\Gamma \vdash_{_{\mathrm{ND}}} \forall x_d.\phi}{\Gamma \vdash_{_{\mathrm{ND}}} \phi[x_d := a_d]}$$

$$(\exists I): \frac{\Gamma \vdash_{_{\mathrm{ND}}} \phi[x_d := a_d]}{\Gamma \vdash_{_{\mathrm{ND}}} \exists x_d.\phi}$$

$$(\exists E): \frac{\Gamma \vdash_{_{\mathrm{ND}}} \exists x_d.\phi}{\Gamma \vdash_{_{\mathrm{ND}}} \psi} \text{, where } a_d \text{ does not occur in } \Gamma, \phi, \psi$$



## Natural deduction: soundness and completeness

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

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

#### **Completeness:**

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

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

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  $x_d \in BV_d$ :

# **TU/e** technische universiteit eindhoven ranclation from ND to F(2)

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*:

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

Translation of 'variable  $u_d$  may not occur in the sequent  $\Gamma$ ':

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

# **TU/e** technische universiteit eindhoven Proof of (C1)

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

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

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

# **TU/e** technische universiteit eindhoven Proof of (C2)

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

$$\text{if} \hspace{0.2cm} \models_{\scriptscriptstyle \rm E} \overline{\Gamma} =_{\scriptscriptstyle \mathbb B} \overline{\phi} \wedge \overline{\Gamma} \hspace{0.2cm} \text{then} \hspace{0.2cm} \Gamma \models_{\scriptscriptstyle \rm ND} \phi$$

Proof is still under construction. Idea: proof by induction on the structure of  $\phi$ .

# 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  $\Sigma$ -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.