On Refinements of Algebraic Specifications

Alexandre Madeira

Department of Mathematics, University of Aveiro
Informatics Department, University of Minho

CIC'09 Coinduction, Interaction and Composition
May 8, 2009, Braga, Portugal

Aims

In view of software reuse, it would be interesting:

- transpose the traditional assumption of the preservation of observability on the observational stepwise refinement process;
- Consider an alternative of the refinement concept based on general maps (and not just on signature morphisms)- refinements via interpretations.

Outline

Outline

The algebraic specification process

In general algebraic assumption:

- programs are algebras;
- computations are terms;

The algebraic specification process

In general algebraic assumption:

- programs are algebras;
- computations are terms;

To specify a software system:

- Define a (multi-sorted) an adequate signature
- Express the system requirements in a logical system;

The algebraic specification process

In general algebraic assumption:

- programs are algebras;
- computations are terms;

To specify a software system:

- Define a (multi-sorted) an adequate signature
- Express the system requirements in a logical system;

Algebraic Specification: $SP = (\Sigma, Mod(SP))$
Development task: Choose a Σ-algebra from $Mod(SP)$ to implement the system;

The software development - the stepwise refinement methodology

Definition (Refinement)

Let SP and SP′ be algebraic specifications. SP′ is a refinement of SP, ($SP \rightsquigarrow SP'$), if:

1. $Sig(SP) = Sig(SP')$;
2. $Mod(SP') \subseteq Mod(SP)$;

The software development - the stepwise refinement methodology

Definition (Refinement)

Let SP and SP' be algebraic specifications. SP' is a refinement of SP, ($SP \rightsquigarrow SP'$), if:

1. $Sig(SP) = Sig(SP')$;
2. $Mod(SP') \subseteq Mod(SP)$;

Stepwise Refinement Process:

$$SP_0 \rightsquigarrow SP_1 \rightsquigarrow SP_2 \rightsquigarrow \cdots \rightsquigarrow SP_{n-1} \rightsquigarrow SP_n,$$

Vertical composition: $SP \rightsquigarrow SP'$ and $SP' \rightsquigarrow SP''$ then $SP \rightsquigarrow SP''$:
$Sig(SP) = Sig(SP') = Sig(SP'')$ and
$Mod(SP'') \subseteq Mod(SP') \subseteq Mod(SP)$

Example of refinement

Spec Cell =

$[Sorts]$ $\quad elt; cell;$

$[OP]$ $\quad put : elt, cell \longrightarrow cell; get : cell \longrightarrow elt;$

$[AX]$ $\quad (\forall e : elt)(\forall c : cell) get(put(e, c)) = e;$

Spec $CELL^*$= enrich CELL by

$[AX]$ $\quad (\forall e, e' : elt)(\forall c : cell) put(e, put(e', c))) = put(e, c)$

$$CELL \rightsquigarrow CELL^*$$

The stepwise refinement methodology

Lemma (Satisfaction lemma-[GB92])

Let $\sigma : \Sigma \to \Sigma'$ *be a signature morphism,* ϕ *a* Σ*-equation and* A' *a* Σ'*-algebra. Then* $A' \models \sigma(\phi)$ *iff* $A' \restriction_\sigma \models \phi$.

The stepwise refinement methodology

Lemma (Satisfaction lemma-[GB92])

Let $\sigma : \Sigma \to \Sigma'$ be a signature morphism, ϕ a Σ-equation and A' a Σ'-algebra. Then $A' \models \sigma(\phi)$ iff $A' \restriction_\sigma \models \phi$.

Definition (σ-refinement)

$$SP \rightsquigarrow_\sigma SP'$$

if

$$Mod(SP') \restriction_\sigma \subseteq Mod(SP)$$

where $Mod(SP') \restriction_\sigma = \{A' \restriction_\sigma | A' \in Mod(SP')\}$.

The stepwise refinement methodology

Lemma (Satisfaction lemma-[GB92])

Let $\sigma : \Sigma \rightarrow \Sigma'$ be a signature morphism, ϕ a Σ-equation and A' a Σ'-algebra. Then $A' \models \sigma(\phi)$ iff $A' \restriction_{\sigma} \models \phi$.

Definition (σ-refinement)

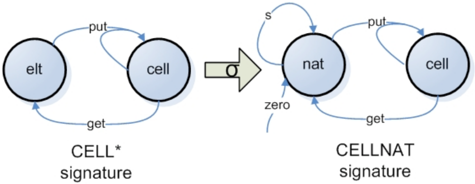
$$SP \leadsto_{\sigma} SP'$$
if
$$Mod(SP') \restriction_{\sigma} \subseteq Mod(SP)$$

where $Mod(SP') \restriction_{\sigma} = \{A' \restriction_{\sigma} | A' \in Mod(SP')\}$.

vertical composition: $SP \leadsto_{\sigma} SP' \leadsto_{\phi} SP''$
$Mod(SP'') \restriction_{\phi \circ \sigma} \subseteq Mod(SP') \restriction_{\sigma} \subseteq Mod(SP)$

Example of σ-refinement

$$
\begin{array}{llll}
\sigma : & Sig(CELL) & \rightarrow & Sig(CELLNAT) \\
 & elt & \rightarrow & nat \\
 & cell & \rightarrow & cell \\
 & get & \rightarrow & get \\
 & put & \rightarrow & put \\
 & & \rightarrow & s \\
 & & \rightarrow & zero
\end{array}
$$

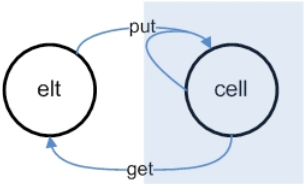


$$CELL \leadsto_{\sigma} CELLNAT$$

Encapsulated data and observability

In presence of encapsulated data:

- Observational equality (between elements);
- Observational equivalence (between models);



Observational equality

Definition (Observable context)

Let $\Sigma = (S, \Omega)$ *be a signature,* $Obs \subseteq S$ *a set of observable sorts,* X *a variable set for* Σ *and* $Z = (\{z_s\})_{s \in S}$ *a* S*-sorted set of singulares sets. A* s*-context over* Σ *is a term* $c \in T_\Sigma(X \cup \{z_s\})_{s'}$, $s' \in Obs$.

Definition (Observational equality)

For $a, b \in A_s$, $a \approx^A_{Obs} b$ *if* $\forall c \in \mathcal{C}^{Obs}_\Sigma(s)\ \forall \alpha, \beta : X \cup \{z_s\} \to A$ *such that* $\alpha(x) = \beta(x)$ *for all* $x \in X$ *and* $\alpha(z_s) = a$ *and* $\beta(z_s) = b \Rightarrow I_\alpha(c) = I_\beta(c)$.

Observational equality

Definition (Observable context)

Let $\Sigma = (S, \Omega)$ be a signature, $Obs \subseteq S$ a set of observable sorts, X a variable set for Σ and $Z = (\{z_s\})_{s \in S}$ a S-sorted set of singulares sets. A s-context over Σ is a term $c \in T_\Sigma(X \cup \{z_s\})_{s'}$, $s' \in Obs$.

Definition (Observational equality)

For $a, b \in A_s$, $a \approx^A_{Obs} b$ if $\forall c \in \mathcal{C}^{Obs}_\Sigma(s)$ $\forall \alpha, \beta : X \cup \{z_s\} \to A$ such that $\alpha(x) = \beta(x)$ for all $x \in X$ and $\alpha(z_s) = a$ and $\beta(z_s) = b \Rightarrow I_\alpha(c) = I_\beta(c)$.

- $\models_{\approx_{Obs}}$-observational satisfaction
- $A/\approx^A_{Obs}$-observational behaviour
- behavioural operator: $Mod($**behaviour** SP **wrt** $\approx_{Obs}) = \{A \in Alg(Sig(SP)) | A/\approx^A_{Obs} \in Mod(SP)\}$

Observational equality

Theorem (eg. [BH96])

$$A \models_{\approx_{Obs}} \phi \textit{ iff } A/\approx^{A}_{Obs} \models \phi$$

Fact

For $SP = \langle \Sigma, \Phi \rangle$:

$$Mod(\textbf{behaviour } SP \textbf{ wrt } \approx_{Obs}) = \{A \in Alg(\Sigma) | A/\approx^{A}_{Obs} \in SP\}$$

$$= \{A \in Alg(\Sigma) | A/\approx^{A}_{Obs} \models \Phi\} = \{A \in Alg(\Sigma) | A \models_{\approx_{Obs}} \Phi\}$$

Encapsulated data and observability

Spec $Cell^* =$

$[Sorts]$ $elt; cell;$

$[OP]$ $put : elt, cell \to cell; get : cell \to elt;$

$[AX]$ $(\forall e : elt)(\forall c : cell) get(put(e, c)) = e;$

$(\forall e, e' : elt)(\forall c : cell) put(e, put(e', c))) = put(e, c);$

$B_{elt} = \mathbb{N}$

$B_{cell} = \mathbb{N}^*$

$put^B(m, c) = mc$

$get^B(\epsilon) = 0$

$get^B(nc) = n$ for $n, m \in \mathbb{N}, c \in \mathbb{N}^*$

$$put^B(e, put^B(e', c)) = put^B(e, e'c) = ee'c \text{ and } put^B(e, c) = ec \text{ but,}$$

$$get^B(put^B(e, put^B(e', c))) = get(put^B(e, c))$$

$$get^B(put(e'', (put^B(e, put^B(e', c))))) = get^B(put(e'', (put^B(e, c)))$$

$$\vdots$$

Then, $put^B(e, put^B(e', c)) \approx^B_{Obs} put^B(e, c)$

"Traditional" observational stepwise refinement process

Definition (Observational σ-refinement)

Let SP and SP′ be algebraic specifications, Obs ⊆ S and $\sigma : Sig(SP) \to Sig(SP')$.

$$SP \rightsquigarrow_{\sigma}^{\approx Obs} SP'$$

if

$$\textbf{behaviour } SP \textbf{ wrt } \approx_{Obs \rightsquigarrow_\sigma} SP',$$

ie., when

$$Mod(SP') \restriction_\sigma \subseteq Mod(\textbf{behaviour } SP \textbf{ wrt } \approx_{Obs})$$

$$SP_1 \rightsquigarrow_{\sigma_1}^{\approx Obs} SP_2 \rightsquigarrow_{\sigma_2}^{\approx \sigma_1(Obs)} \ldots SP_{n-1} \rightsquigarrow_{\sigma_n}^{\approx \sigma_n \circ \cdots \circ \sigma_1(Obs)} SP_n \Rightarrow SP_1 \rightsquigarrow_{\sigma_n \circ \cdots \circ \sigma_1}^{\approx Obs} SP_n$$

Outline

Motivations

Changing of *Obs* set in observational stepwise refinement process:

- According to *O.O. Paradigm*, only the input/output data should be desencapsulated;
- For security reasons may be important encapsulate same data sorts;
- For software verification tasks;
- ...

The vertical composition of observational refinements

Fact ([Hen97])

Let

$$\begin{array}{ccccc} \Sigma & & \Sigma & & \Sigma \\ SP & \rightsquigarrow^{\approx_{Obs}} & SP' & \rightsquigarrow^{\approx_{Obs'}} & SP'' \\ A & & A & & \\ \approx^A_{Obs} & & \approx^A_{Obs'} & & \end{array}$$

Then, $\approx^A_{Obs} \geq \approx^A_{Obs'} \Rightarrow SP \rightsquigarrow^{\approx_{Obs}} SP''$

The vertical composition of observational refinements

Fact ([Hen97])

Let

$$\begin{array}{ccccc} \Sigma & & \Sigma & & \Sigma \\ SP & \rightsquigarrow^{\approx_{Obs}} & SP' & \rightsquigarrow^{\approx_{Obs'}} & SP'' \\ A & & A & & \\ \approx^{A}_{Obs} & & \approx^{A}_{Obs'} & & \end{array}$$

Then, $\approx^{A}_{Obs} \geq \approx^{A}_{Obs'} \Rightarrow SP \rightsquigarrow^{\approx_{Obs}} SP''$

Fact (Generalization)

Let

$$\begin{array}{ccccc} \Sigma & & \Sigma' & & \Sigma'' \\ SP & \rightsquigarrow^{\approx_{Obs}}_{\sigma} & SP' & \rightsquigarrow^{\approx_{Obs'}}_{\phi} & SP'' \\ A' \restriction_{\sigma} & & A' & & \\ \approx^{A' \restriction_{\sigma}}_{Obs} & & \approx^{A'}_{Obs'} & & \end{array}$$

Then, $\approx^{A' \restriction_{\sigma}}_{Obs} \geq (\approx^{A'}_{Obs'}) \restriction_{\sigma} \Rightarrow SP \rightsquigarrow^{\approx_{Obs}}_{\phi \circ \sigma} SP''$

The vertical composition of observational refinements

Fact ([Hen97])

Let

$$\begin{array}{ccccc} \Sigma & & \Sigma & & \Sigma \\ SP & \rightsquigarrow^{\approx_{Obs}} & SP' & \rightsquigarrow^{\approx_{Obs'}} & SP'' \\ A & & A & & \\ \approx^{A}_{Obs} & & \approx^{A}_{Obs'} & & \end{array}$$

Then, $\approx^{A}_{Obs} \geq \approx^{A}_{Obs'} \Rightarrow SP \rightsquigarrow^{\approx_{Obs}} SP''$

Fact (Generalization)

Let

$$\begin{array}{ccccc} \Sigma & & \Sigma' & & \Sigma'' \\ SP & \rightsquigarrow^{\approx_{Obs}}_{\sigma} & SP' & \rightsquigarrow^{\approx_{Obs'}}_{\phi} & SP'' \\ A' \upharpoonright_{\sigma} & & A' & & \\ \approx^{A' \upharpoonright_{\sigma}}_{Obs} & & \approx^{A'}_{Obs'} & & \end{array}$$

Then, $\approx^{A' \upharpoonright_{\sigma}}_{Obs} \geq (\approx^{A'}_{Obs'}) \upharpoonright_{\sigma} \Rightarrow SP \rightsquigarrow^{\approx_{Obs}}_{\phi \circ \sigma} SP''$

Intuitive idea: for $\sigma = id$, $Obs \subseteq Obs' \Rightarrow \mathcal{C}^{Obs}_{\Sigma} \subseteq \mathcal{C}^{Obs'}_{\Sigma} \Rightarrow \approx_{Obs} \geq \approx_{Obs'}$

The vertical composition of observational refinements

Definition (Observational morphism)

Let $\Sigma = (S, \Omega)$ *and* $\Sigma' = (S', \Omega')$ *signatures,* Obs *and* Obs' *sets of observable sorts for* Σ *and* Σ' *respectively, and* $\sigma : \Sigma \to \Sigma'$ *a signature morphism.* σ *is an* $Obs - Obs'$*-observational morphism if for any* $s \in Obs, \sigma(s) \in Obs'$*;*

The vertical composition of observational refinements

Definition (Observational morphism)

Let $\Sigma = (S, \Omega)$ *and* $\Sigma' = (S', \Omega')$ *signatures,* Obs *and* Obs' *sets of observable sorts for* Σ *and* Σ' *respectively, and* $\sigma : \Sigma \to \Sigma'$ *a signature morphism.* σ *is an* $Obs - Obs'$*-observational morphism if for any* $s \in Obs, \sigma(s) \in Obs'$*;*

Theorem

Let $\sigma : \Sigma \to \Sigma'$ *be an* $Obs - Obs'$*-observational morphism and* A *be a* Σ'*-algebra. Then* $(\approx_{Obs',A'}) \restriction_\sigma \leq \approx_{Obs,(A' \restriction_\sigma)}$ *.*

The vertical composition of observational refinements

Definition (Observational morphism)

Let $\Sigma = (S, \Omega)$ *and* $\Sigma' = (S', \Omega')$ *signatures,* Obs *and* Obs' *sets of observable sorts for* Σ *and* Σ' *respectively, and* $\sigma : \Sigma \to \Sigma'$ *a signature morphism.* σ *is an* $Obs - Obs'$*-observational morphism if for any* $s \in Obs, \sigma(s) \in Obs'$*;*

Theorem

Let $\sigma : \Sigma \to \Sigma'$ *be an* $Obs - Obs'$*-observational morphism and* A *be a* Σ'*-algebra. Then* $(\approx_{Obs', A'}) \restriction_\sigma \leq \approx_{Obs, (A' \restriction_\sigma)}$ *.*

Corollary

Let σ *be an* $Obs - Obs'$*-observable morphism and* ϕ *be an* $Obs' - Obs''$*-observable morphism. If* $SP \rightsquigarrow_{\sigma}^{\approx_{Obs}} SP'$ *and* $SP' \rightsquigarrow_{\phi}^{\approx_{Obs'}} SP''$*, then* $SP \rightsquigarrow_{\phi \circ \sigma}^{\approx_{Obs}} SP''$*.*

Concerning to the equational case:

Let ϕ be a Σ-equation and A be a Σ-algebra and $SP = \langle \Sigma, \Phi \rangle$ be an equational specification:

- $A \models_{\approx_{Obs}} \phi \Rightarrow A \models_{\approx_{Obs \setminus \{s\}}} \phi$
- $Mod(\textbf{behavioural } SP \textbf{ wrt } \approx_{Obs}) \subseteq Mod(\textbf{behavioural } SP \textbf{ wrt } \approx_{Obs \setminus \{s\}})$
- $SP \rightsquigarrow^{\approx_{Obs}} SP'$ implies $SP \rightsquigarrow^{\approx_{Obs \setminus \{s\}}} SP'$

Concerning to the equational case:

Let ϕ be a Σ-equation and A be a Σ-algebra and $SP = \langle \Sigma, \Phi \rangle$ be an equational specification:

- $A \models_{\approx_{Obs}} \phi \Rightarrow A \models_{\approx_{Obs \setminus \{s\}}} \phi$
- $Mod($**behavioural** SP **wrt** $\approx_{Obs}) \subseteq Mod($**behavioural** SP **wrt** $\approx_{Obs \setminus \{s\}})$
- $SP \rightsquigarrow^{\approx_{Obs}} SP'$ implies $SP \rightsquigarrow^{\approx_{Obs \setminus \{s\}}} SP'$

However, in the general case:

- $Mod($**behavioural** SP **wrt** $\approx_{Obs \setminus \{s\}}) \not\subseteq Mod($**behavioural** SP **wrt** $\approx_{Obs})$
- $SP \rightsquigarrow^{\approx_{Obs}} SP'$ does not implies $SP \rightsquigarrow^{\approx_{Obs \cup \{s\}}} SP'$

Desencapsulation in equational specifications

Lemma

Let $\Sigma = (S, \Omega)$ be a signature, Obs a set of observable sorts for Σ and Φ be a set of Σ-equations. Then, for any Σ-algebra A and for any $s \in Obs$,

$$A \models_{\approx_{Obs \cup \{s\}}} \Phi \text{ iff } \left(A \models_{\approx_{Obs}} \Phi \text{ and } A \models \Phi'\right)$$

where

$$\Phi' = \Phi_s \cup \{c(t) = c(t') \mid t \approx t' \in \Phi_h, h \in S \setminus (Obs \cup \{s\}), c \in \mathcal{C}_{\Sigma}^{\{s\}}(h)\}.$$

Notation

$$SP_s = \langle \Sigma, \Phi' \rangle$$

Desencapsulation in equational specifications

$$SP \rightsquigarrow^{\approx_{Obs}} SP'$$

$$Mod(SP') \subseteq Mod(\textbf{behaviour } SP \textbf{ wrt } \approx_{Obs})$$

$$Mod(SP') \cap Mod(SP_s) \subseteq Mod(\textbf{behaviour } SP \textbf{ wrt } \approx_{Obs}) \cap Mod(SP_s)$$

$$Mod(SP') \cap Mod(SP_s) \subseteq Mod(\textbf{behaviour } SP \textbf{ wrt } \approx_{Obs} \cap SP_s)$$

$$Mod(SP') \cap Mod(SP_s) \subseteq Mod(\textbf{behaviour } SP \textbf{ wrt } \approx_{Obs \cup \{s\}})$$

Desencapsulation in equational specifications

$$SP \rightsquigarrow^{\approx_{Obs}} SP'$$

$$Mod(SP') \subseteq Mod(\textbf{behaviour } SP \textbf{ wrt } \approx_{Obs})$$

$$Mod(SP') \cap Mod(SP_s) \subseteq Mod(\textbf{behaviour } SP \textbf{ wrt } \approx_{Obs}) \cap Mod(SP_s)$$

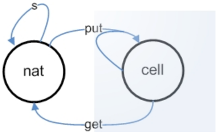
$$Mod(SP') \cap Mod(SP_s) \subseteq Mod(\textbf{behaviour } SP \textbf{ wrt } \approx_{Obs} \cap SP_s)$$

$$Mod(SP') \cap Mod(SP_s) \subseteq Mod(\textbf{behaviour } SP \textbf{ wrt } \approx_{Obs \cup \{s\}})$$

Theorem

Let Φ be a set of Σ-equations and $SP = \langle \Sigma, \Phi \rangle$ and SP' be two specifications such that $SP \rightsquigarrow^{\approx_{Obs}} SP'$. Then

$$SP \rightsquigarrow^{\approx_{Obs \cup \{s\}}} SP' \cap SP_s.$$



Spec $CELL^{*} =$

$(\forall e : elt)(\forall c : cell) get(put(e, c)) \approx e;$

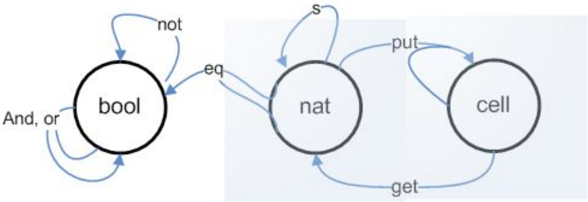
$(\forall e, e' : elt)(\forall c : cell) put(e, put(e', c))) \approx put(e, c);$

$$\Phi' = \Phi_s \cup \{c(t) \approx c(t') \mid t \approx t' \in \Phi_h, h \in S \setminus (Obs \cup \{s\}), c \in \mathcal{C}_{\Sigma}^{\{s\}}(h)\}.$$

Spec $CELL^{*}_{cell} =$

$(\forall e, e' : elt)(\forall c : cell) put(e, put(e', c))) \approx put(e, c);$

- For any SP such that $CELL^{*} \rightsquigarrow^{\approx_{elt}} SP$, we have that $CELL^{*} \rightsquigarrow^{\approx_{elt \cup cell}} SP + CELL^{*}_{cell}$



$$\Phi' = \Phi_s \cup \{c(t) \approx c(t') \mid t \approx t' \in \Phi_h, h \in S \setminus (Obs \cup \{s\}), c \in \mathcal{C}_\Sigma^{\{s\}}(h)\}.$$

Spec $CELLNATEQ_{nat}$ =

$(\forall x : nat).s(p(x)) \approx x$;

$(\forall e, e' : nat)(\forall c : cell).get(put(e, put(e', c))) \approx get(put(e, c))$;

$(\forall e, e', e'' : nat)(\forall c : cell).get(put(e'', put(e, put(e', c))))$
$\approx get(put(e'', put(e, c)))$;

⋮

Outline

The refinement by interpretation concept - motivations

The previous refinement formalizations are based on signatures morphisms:

- the formulas structure is preserved, ie.:
  $\sigma(f(a_1, \ldots, a_n)) = \sigma(f)(\sigma(a_1) \ldots \sigma(a_n))$;
- a formula is mapped into another one;
- the choice of other maps to translate specifications can be useful in the view of the software reuse.

Definitions and notation

- conditional equation: is a pair $\langle \Gamma, e \rangle$, for $\{e\} \cup \Gamma \subseteq_{fin} \mathrm{Eq}_{\Sigma}(X)$;
- translation from Σ to Σ': is a globally finite *multi-function* $\tau : \mathrm{Eq}_{\Sigma}(X) \rightarrow \mathrm{Eq}_{\Sigma'}(X')$

$\tau^* : Ceq_{\Sigma}(X) \rightarrow Ceq_{\Sigma'}(X')$

for any $\xi = \langle \Gamma, e \rangle \in Ceq_{\Sigma}(X)$,

$$\tau^*(\xi) = \{\langle \bigcup_{t \approx t' \in \Gamma} \tau(t \approx t'), e' \rangle : e' \in \tau(e)\}.$$

Interpretations

Definition (Interpretation)

Let $\tau : \mathrm{Eq}_{\Sigma}(X) \to \mathrm{Eq}_{\Sigma'}(X')$. Let SP be a specification over Σ. We say that τ interprets SP if there is a specification SP′ over Σ' such that, for any $\xi \in \mathrm{Ceq}_{\Sigma}(X)$,

$$SP \models \xi \text{ if and only if } SP' \models \tau(\xi).$$

Interpretations

Definition (Interpretation)

Let $\tau : \mathrm{Eq}_\Sigma(X) \to \mathrm{Eq}_{\Sigma'}(X')$. *Let SP be a specification over* Σ. *We say that* τ *interprets SP if there is a specification* SP' *over* Σ' *such that, for any* $\xi \in \mathrm{Ceq}_\Sigma(X)$,

$$SP \models \xi \text{ if and only if } SP' \models \tau(\xi).$$

Definition (τ-model)

Let $\tau : \mathrm{Eq}(X)_\Sigma \to \mathrm{Eq}(X)_{\Sigma'}$. *A* Σ'*-algebra* A' *is a* τ*-model of SP if for any* $\xi \in Fm(\Sigma)$, $SP \models \xi$ *implies* $A' \models \tau(\xi)$. *We define also* $SP^\tau = \langle \Sigma', Mod(SP^\tau) \rangle$ *where* $Mod(SP^\tau)$ *is the classe of* τ*-models of SP.*

Interpretations

Definition (Interpretation)

Let $\tau : \mathrm{Eq}_{\Sigma}(X) \to \mathrm{Eq}_{\Sigma'}(X')$. Let SP be a specification over Σ. We say that τ interprets SP if there is a specification SP′ over Σ' such that, for any $\xi \in \mathrm{Ceq}_{\Sigma}(X)$,

$$SP \models \xi \text{ if and only if } SP' \models \tau(\xi).$$

Definition (τ-model)

Let $\tau : \mathrm{Eq}(X)_{\Sigma} \to \mathrm{Eq}(X)_{\Sigma'}$. A Σ'-algebra A' is a τ-model of SP if for any $\xi \in Fm(\Sigma)$, $SP \models \xi$ implies $A' \models \tau(\xi)$. We define also $SP^{\tau} = \langle \Sigma', Mod(SP^{\tau}) \rangle$ where $Mod(SP^{\tau})$ is the classe of τ-models of SP.

Theorem

If τ interprets SP, then the specification SP^{τ} is the τ-interpretation of SP with the largest class of models.

Example of interpretation

Lemma ([BR03])

For $\mathcal{L}$ *the propositional language and* $\{\phi_1 \approx \psi_1, \ldots, \phi_n \approx \psi_n\}, \{\phi \approx \psi\} \subseteq \mathrm{Eq}_{\mathcal{L}}(X)$*:*

$$\mathit{BOOL} \models \langle\{\phi_1 \approx \psi_1, \ldots, \phi_n \approx \psi_n\}, \{\phi \approx \psi\}\rangle$$

iff

$$\mathit{HEYTING} \models \langle\{\neg\neg\phi_1 \approx \neg\neg\psi_1, \ldots, \neg\neg\phi_n \approx \neg\neg\psi_n\}, \{\neg\neg\phi \approx \neg\neg\psi\}\rangle$$

Therefore, the translation $\tau(t \approx t') = \{\neg\neg t \approx \neg\neg t'\}$ interprets BOOL on HEYTING.

Axiomatization of SP^{τ}

Proposition ([Cze01])

Let $\Sigma = (S, \Omega)$ *be a signature and* $\tau : \mathrm{Eq}(X)_{\Sigma} \to \mathrm{Eq}(X)_{\Sigma}$. *Then, TFAE:*

1. τ *commutes with substitutions, i.e., for any* $e \in \mathrm{Eq}_{\Sigma}(X)$, $\sigma(\tau(e)) = \tau(\sigma(e))$;
2. *There exists a* S*-sorted set of equations* $\Delta(x, y) \subseteq \mathrm{Eq}_{\Sigma}(X)$ *such that, for any* $t \approx t' \in \mathrm{Eq}_{\Sigma}(X)_s$, $\tau(t \approx t') = \Delta_s(t, t')$.

Axiomatization of SP^τ

Proposition ([Cze01])

Let $\Sigma = (S, \Omega)$ *be a signature and* $\tau : \mathrm{Eq}(X)_\Sigma \to \mathrm{Eq}(X)_\Sigma$. *Then, TFAE:*

1. τ *commutes with substitutions, i.e., for any* $e \in \mathrm{Eq}_\Sigma(X)$, $\sigma(\tau(e)) = \tau(\sigma(e))$;
2. *There exists a* S*-sorted set of equations* $\Delta(x, y) \subseteq \mathrm{Eq}_\Sigma(X)$ *such that, for any* $t \approx t' \in \mathrm{Eq}_\Sigma(X)_s$, $\tau(t \approx t') = \Delta_s(t, t')$.

Theorem

Let $\tau : \mathrm{Eq}(X)_\Sigma \to \mathrm{Eq}(X)_\Sigma$ *and* $SP = \langle \Sigma, \Phi \rangle$. *If* τ *interprets* SP *and commutes with arbitrary substitutions then* $SP^\tau = \langle \Sigma', \tau(\Phi) \rangle$. *Moreover, if* Φ *is finite then* SP^τ *is finitely axiomatized.*

Refinement via interpretations

Definition (Refinement via interpretation)

Let SP be a specification over Σ *and* $\tau : \mathrm{Eq}(X)_\Sigma \to \mathrm{Eq}(X')_{\Sigma'}$ *a translations which interprets SP.* $SP \rightharpoondown_\tau SP'$*,if for any* $\xi \in \mathrm{Ceq}_\Sigma(X)$,

$$SP \models \xi \text{ implies } SP' \models \tau(\xi).$$

Refinement via interpretations

Definition (Refinement via interpretation)

Let SP be a specification over Σ *and* $\tau : \mathrm{Eq}(X)_\Sigma \to \mathrm{Eq}(X')_{\Sigma'}$ *a translations which interprets SP.* $SP \rightharpoonup_\tau SP'$*,if for any* $\xi \in \mathrm{Ceq}_\Sigma(X)$,

$$SP \models \xi \text{ implies } SP' \models \tau(\xi).$$

Theorem

Let SP be a specifications over Σ *and* $\tau : \mathrm{Eq}(X)_\Sigma \to \mathrm{Eq}(X')_{\Sigma'}$ *a translations which interprets SP. Then, for every* SP' *specification over* Σ',

$$SP^\tau \rightsquigarrow SP' \text{ implies } SP \rightharpoonup_\tau SP'.$$

Refinement via interpretations

Definition (Refinement via interpretation)

Let SP be a specification over Σ *and* $\tau : \mathrm{Eq}(X)_\Sigma \to \mathrm{Eq}(X')_{\Sigma'}$ *a translations which interprets SP.* $SP \rightharpoondown_\tau SP'$*,if for any* $\xi \in \mathrm{Ceq}_\Sigma(X)$,

$$SP \models \xi \textit{ implies } SP' \models \tau(\xi).$$

Theorem

Let SP be a specifications over Σ *and* $\tau : \mathrm{Eq}(X)_\Sigma \to \mathrm{Eq}(X')_{\Sigma'}$ *a translations which interprets SP. Then, for every* SP' *specification over* Σ',

$$SP^\tau \rightsquigarrow SP' \textit{ implies } SP \rightharpoondown_\tau SP'.$$

Corollary

Let $\tau : \mathrm{Eq}(X)_\Sigma \to \mathrm{Eq}(X')_{\Sigma'}$ *which interprets* $SP = \langle \Sigma, \Phi \rangle$*. Then,*

$$SP' \models \tau(\Phi) \textit{ we have } SP \rightharpoondown_\tau SP'.$$

Example: Equality test

Spec Nat=

[*Sorts*] nat;

[*Op*] $s : nat \rightarrow nat$;

[*Ax*] $s(x) \approx s(y) \Rightarrow x \approx y$

Spec NatEq= enrich BOOL by

[*Sorts*] nat;

[*Op*] $s : nat \rightarrow nat; eq : nat, nat \rightarrow bool$;

[*Ax*] $eq(x, x) \approx true$

$eq(x, y) \approx true \Rightarrow eq(y, x) \approx true$;

$eq(x, y) \approx true \wedge eq(y, z) \approx true \Rightarrow eq(x, z) \approx true$;

$eq(x, y) \approx true \Rightarrow eq(s(x), s(y)) \approx true$;

$eq(s(x), s(y)) \approx true \Rightarrow eq(x, y) \approx true$;

Example: Equality test

Spec Nat=

[Sorts] $nat;$

[Op] $s : nat \to nat;$

[Ax] $s(x) \approx s(y) \Rightarrow x \approx y$

Spec NatEq= enrich BOOL by

[Sorts] $nat;$

[Op] $s : nat \to nat; eq : nat, nat \to bool;$

[Ax] $eq(x, x) \approx true$

$eq(x, y) \approx true \Rightarrow eq(y, x) \approx true;$

$eq(x, y) \approx true \wedge eq(y, z) \approx true \Rightarrow eq(x, z) \approx true;$

$eq(x, y) \approx true \Rightarrow eq(s(x), s(y)) \approx true;$

$eq(s(x), s(y)) \approx true \Rightarrow eq(x, y) \approx true;$

Considering:

$$\tau(x : nat \approx y : nat) = \{eq(x : nat, y : nat) \approx true\}$$

$NatEq \models eq(s(x), s(y)) \approx true \Rightarrow eq(x, y) \approx true$

$$Nat \rightharpoonup_\tau NatEq$$

Outline

It would be interesting:

- create a calculus for the refinements via interpretation;
- integrate the refinements via interpretations on the traditional stepwise refinement process;
- explore, in this perspective, other results from algebraic theory of deductive systems and from the theory of conservative translations.

M. Bidoit and R. Hennicker.
Behavioural theories and the proof of behavioural properties.
Theor. Comput. Sci., 165(1):3–55, 1996.

W. Blok and D. Pigozzi.
Abstract algebraic logic and the deduction theorem.
Preprint. To appear in the Bulletin of Symbolic Logic.

W. Blok and J. Rebagliato.
Algebraic semantics for deductive systems.
Studia Logica, 74(1-2):153–180, 2003.

J. Czelakowski.
Protoalgebraic Logics.
Trends in logic, Studia Logica Library, Kluwer Academic Publishers, 2001.

J. Goguen and R. Burstall.
Institutions: abstract model theory for specification and programming.
J. ACM, 39(1):95–146, 1992.

R. Hennicker.
Structural specifications with behavioural operators: semantics, proof methods and applications, 1997.
Habilitationsschrift, Institut fur Informatik, Ludwig-Maximilians-Universitat Munchen.

A. Madeira.
Observational refinement process.
Electron. Notes Theor. Comput. Sci., 214:103–129, 2008.

M. A. Martins.
Behavioral institutions and refinements in generalized hidden logics.
Journal of Universal Computer Science, 12(8):1020–1049, 2006.