Multiple Parameterised Specifications with Sharing

Ionuț Țuțu

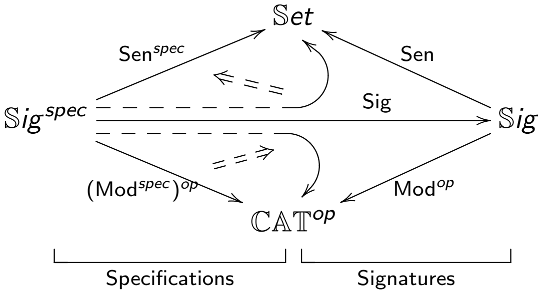
Department of Computer Science, University of Leicester

"Simion Stoilow" Institute of Mathematics of the Romanian Academy

Sinaia, 2012

$$\mathsf{Sig} \colon \mathsf{Specifications} \to \mathsf{Signatures}$$

- parameterised specifications and their instantiation depend heavily on the properties of both signatures and Sig



The signatures

Example (Many-sorted algebra (MSA))

signatures: (S, F)

- S is a set of sorts,
- F is a family $\{F_{w \to s} \mid w \in S^*, s \in S\}$ of operation symbols

morphisms: $\varphi \colon (S, F) \to (S', F')$

- $\varphi^{st} \colon S \to S'$ is a function
- $\varphi^{op} = \{\varphi^{op}_{w \to s} \colon F_{w \to s} \to F'_{\varphi^{st}(w) \to \varphi^{st}(s)}\}$ is a family of functions

Example (Order-sorted algebra (OSA))

Example (Partial algebra (PA))

The signatures

Example (Many-sorted algebra (**MSA**))

Example (Order-sorted algebra (**OSA**))

signatures: $(S, \leq, F)$

- (S, F) is a **MSA** signature such that $s_1 \leq s_2$ whenever $w_1 \leq w_2$ and $F_{w_1 \to s_1} \cap F_{w_2 \to s_2} \neq \emptyset$
- $(S \leq)$ is a partially ordered set

morphisms: $\varphi\colon (S, \leq, F) \to (S', \leq', F')$

- $\varphi\colon (S, F) \to (S', F')$ is a **MSA** signature morphism
- $\varphi^{st}\colon (S, \leq) \to (S', \leq')$ is a monotone function

Example (Partial algebra (**PA**))

The signatures

Example (Many-sorted algebra (MSA))

Example (Order-sorted algebra (OSA))

Example (Partial algebra (PA))

signatures: (S, F, TF)

- (S, F) is a **MSA** signature
- $TF = \{TF_{w \to s} \subseteq F_{w \to s}\}$ is a family of total operation symbols

morphisms: $\varphi\colon (S, F, TF) \to (S', F', TF')$

- $\varphi\colon (S, F) \to (S', F')$ is a **MSA** signature morphism
- $\varphi^{op}_{w \to s}(TF_{w \to s}) \subseteq TF'_{\varphi^{st}(w) \to \varphi^{st}(s)}$

Definition (Parameterised specification)

A *parameterised specification*, denoted $SP(P)$, consists in a specification morphism $\iota\colon P \to SP$ such that $\mathrm{Sig}(\iota)$ is the inclusion $\mathrm{Sig}(P) \subseteq \mathrm{Sig}(SP)$.

> an inclusion of signatures $\mathrm{Sig}(P) \subseteq \mathrm{Sig}(SP)$ such that for each model M of SP, the reduct $M\restriction_{\mathrm{Sig}(P)}$ is a model of P

Example ($\mathrm{ELT} \xrightarrow{\iota} \mathrm{LIST}$)

```
mod* ELT {
  [ Elt ]
}
```

```
mod! LIST(E :: ELT) {
  [ List ]

  op nil : -> List
  op __ : Elt List -> List
}
```

Definition (Inclusion)

An *inclusion of (algebraic) signatures* is a signature morphism with all components defined as set-theoretic inclusions.

Example (Inclusions of MSA signatures)

$(S, F) \subseteq (S', F')$:

- $S \subseteq S'$
- $F_{w \to s} \subseteq F'_{w \to s}$

Example (Inclusions of OSA signatures)

Example (Inclusions of PA signatures)

Definition (Inclusion)

An *inclusion of (algebraic) signatures* is a signature morphism with all components defined as set-theoretic inclusions.

Example (Inclusions of MSA signatures)

Example (Inclusions of OSA signatures)

$(S, \leq, F) \subseteq (S', \leq', F')$:

- $(S, F) \subseteq (S', F')$
- $(S, \leq) \subseteq (S', \leq')$

Example (Inclusions of PA signatures)

Definition (Inclusion)

An *inclusion of (algebraic) signatures* is a signature morphism with all components defined as set-theoretic inclusions.

Example (Inclusions of MSA signatures)

Example (Inclusions of OSA signatures)

Example (Inclusions of PA signatures)

$(S, F, TF) \subseteq (S', F', TF')$:

- $(S, F) \subseteq (S', F')$
- $TF_{w \to s} \subseteq TF'_{w \to s}$

Single parameterised specifications

Instantiation of parameters

Definition (Instantiation of parameters)

- consider a parameterised specification $SP(P)$ and
- a specification morphism $v\colon P \to P'$ that *preserves* P'

The instantiation of the parameterised specification $SP(P)$ *by* v is

$$SP(P \Leftarrow v) = SP \star v' \cup P' \star i$$

given by the *pushout* of signatures depicted below.

$$\begin{array}{ccc}
\mathrm{Sig}(P) \cup (\mathrm{Sig}(SP) \cap \mathrm{Sig}(P')) & \xrightarrow{\subseteq} & \mathrm{Sig}(SP) \\
{\scriptstyle v \vee \mathrm{id}} \big\downarrow & \mathrm{PO} & \big\downarrow {\scriptstyle v'} \\
\mathrm{Sig}(P') & \xrightarrow[i]{} & \Sigma'
\end{array}$$

Definition

Unions are least upper bounds in the category of inclusions.
Dually, *intersections* are greatest lower bounds.

Example (Unions and intersections of MSA signatures)

$(S_1, F_1) \cup (S_2, F_2) = (S, F)$

- $S = S_1 \cup S_2$
- $F_{w \to s} = \bigcup_{\substack{i \in \{1,2\} \\ w \in S_i^*,\, s \in S_i}} (F_i)_{w \to s}$

$(S_1, F_1) \cap (S_2, F_2) = (S, F)$

- $S = S_1 \cap S_2$
- $F_{w \to s} = (F_1)_{w \to s} \cap (F_2)_{w \to s}$

Example (Unions and intersections of OSA signatures)

Example (Unions and intersections of PA signatures)

Definition

Unions are least upper bounds in the category of inclusions.
Dually, *intersections* are greatest lower bounds.

Example (Unions and intersections of **MSA** signatures)

Example (Unions and intersections of **OSA** signatures)

- unions may not exist because of antisymmetry

Example (Unions and intersections of **PA** signatures)

Definition

Unions are least upper bounds in the category of inclusions.
Dually, *intersections* are greatest lower bounds.

Example (Unions and intersections of MSA signatures)

Example (Unions and intersections of preorder-based OSA signatures)

$(S_1, \leq_1, F_1) \cup (S_2, \leq_2, F_2) = (S, \leq, F)$

- $(S, F) = (S_1, F_1) \cup (S_2, F_2)$
- $\leq = (\leq_1 \cup \leq_2)^{m*}$

$(S_1, \leq_1, F_1) \cap (S_2, \leq_2, F_2) = (S, \leq, F)$

- $(S, F) = (S_1, F_1) \cap (S_2, F_2)$
- $\leq = \leq_1 \cap \leq_2$

Example (Unions and intersections of PA signatures)

Definition

Unions are least upper bounds in the category of inclusions.
Dually, *intersections* are greatest lower bounds.

Example (Unions and intersections of **MSA** signatures)

Example (Unions and intersections of preorder-based **OSA** signatures)

Example (Unions and intersections of **PA** signatures)

$(S_1, F_1, TF_1) \cup (S_2, F_2, TF_2) = (S, F, TF)$

- $(S, F) = (S_1, F_1) \cup (S_2, F_2)$
- $TF = TF_1 \cup TF_2$

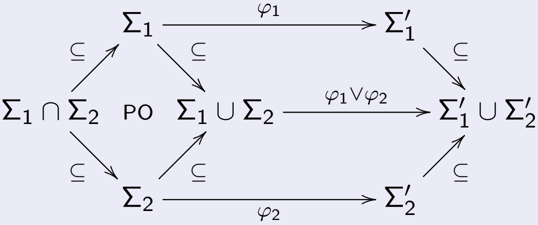
$(S_1, F_1, TF_1) \cap (S_2, F_2, TF_2) = (S, F, TF)$

- $(S, F) = (S_1, F_1) \cup (S_2, F_2)$
- $TF = TF_1 \cap TF_2$

Definition (Compatible morphisms)

Two morphisms $\varphi_1 \colon \Sigma_1 \to \Sigma'_1$ and $\varphi_2 \colon \Sigma_2 \to \Sigma'_2$ are *compatible* when

$$(\Sigma_1 \cap \Sigma_2 \subseteq \Sigma_1); \varphi_1; (\Sigma'_1 \subseteq \Sigma'_1 \cup \Sigma'_2) = (\Sigma_1 \cap \Sigma_2 \subseteq \Sigma_2); \varphi_2; (\Sigma'_2 \subseteq \Sigma'_1 \cup \Sigma'_2).$$



A morphism φ *preserves* a signature Σ if φ and 1_Σ are compatible. It *strongly preserves* a signature Σ when, in addition to preserving Σ, it satisfies $\mathrm{cod}(\varphi) \cap \Sigma \subseteq \mathrm{dom}(\varphi) \cap \Sigma$.

Single parameterised specifications
Instantiation of parameters

Definition (Instantiation of parameters)

- consider a parameterised specification $SP(P)$ and
- a specification morphism $v\colon P \to P'$ that *preserves* P'

The instantiation of the parameterised specification $SP(P)$ *by* v is

$$SP(P \Leftarrow v) = SP \star v' \cup P' \star i$$

given by the *pushout* of signatures depicted below.

$$\begin{array}{ccc}
\mathrm{Sig}(P) \cup (\mathrm{Sig}(SP) \cap \mathrm{Sig}(P')) & \xrightarrow{\subseteq} & \mathrm{Sig}(SP) \\
{\scriptstyle v \vee \mathrm{id}} \downarrow & \mathrm{PO} & \downarrow {\scriptstyle v'} \\
\mathrm{Sig}(P') & \xrightarrow[i]{} & \Sigma'
\end{array}$$

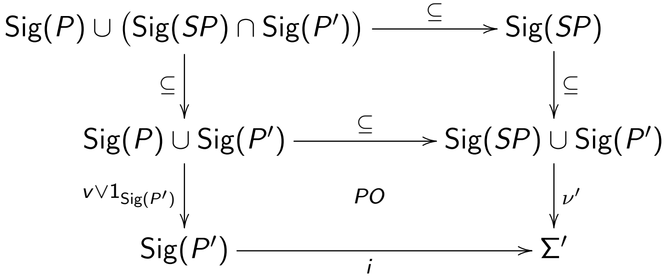
Example (Lists of natural numbers)

$$
\begin{array}{ccc}
\left(\{\texttt{Elt}\}, \emptyset\right) & \xrightarrow{\subseteq} & \left(\begin{array}{l}\{\texttt{Elt}, \texttt{List}\},\\ \{\texttt{nil} \colon [] \to \texttt{List}, \_\_ \colon \texttt{Elt List} \to \texttt{List}\}\end{array}\right) \\
\Big\downarrow {\scriptstyle \texttt{Elt} \mapsto \texttt{Nat}} & & \Big\downarrow {\scriptstyle \texttt{Elt} \mapsto \texttt{Nat}} \\
\left(\begin{array}{l}\{\texttt{Nat}\},\\ \{\texttt{0} \colon [] \to \texttt{Nat}, \texttt{s}\_ \colon \texttt{Nat} \to \texttt{Nat}\}\end{array}\right) & \xrightarrow[\subseteq]{} & \left(\begin{array}{l}\{\texttt{Nat}, \texttt{List}\},\\ \{\texttt{0} \colon [] \to \texttt{Nat}, \texttt{s}\_ \colon \texttt{Nat} \to \texttt{Nat},\\ \;\texttt{nil} \colon [] \to \texttt{List}, \_\_ \colon \texttt{Nat List} \to \texttt{List}\}\end{array}\right)
\end{array}
$$

Single parameterised specifications

Equivalent definition for the instantiation of parameters

- consider the single instantiation $SP(P \Leftarrow v)$



Proposition

The outer square is a pushout square if and only if the lower square is a pushout square.

Single parameterised specifications

Instantiation of parameters via free extensions

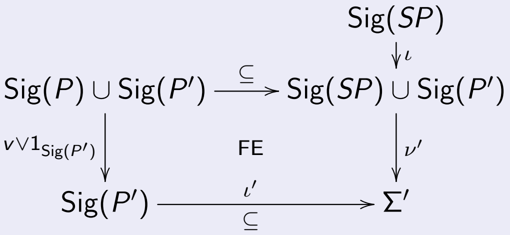
Definition (Instantiation of parameters)

- consider a parameterised specification $SP(P)$ and
- a specification morphism $v\colon P \to P'$ that preserves P'

The instantiation of the parameterised specification $SP(P)$ *by* v is

$$SP(P \Leftarrow v) = SP \star (\iota; \nu') \cup P' \star \iota'$$

given by the *free extension* depicted below.



Single parameterised specifications

Free extensions along inclusions

Definition (Free extension)

Let $\varphi_1\colon \Sigma_1 \to \Sigma_1'$ be a signature morphism and $\Sigma_1 \subseteq \Sigma_2$.
A *free extension* of φ_1 along $\Sigma_1 \subseteq \Sigma_2$ is a signature morphism $\varphi_2\colon \Sigma_2 \to \Sigma_2'$ such that the square below is a pushout square and every signature preserved by φ_1 is also preserved by φ_2.

$$
\begin{array}{ccc}
\Sigma_1 & \xrightarrow{\subseteq} & \Sigma_2 \\
{\scriptstyle \varphi_1}\downarrow & PO & \downarrow{\scriptstyle \varphi_2} \\
\Sigma_1' & \xrightarrow[\subseteq]{} & \Sigma_2'
\end{array}
$$

Example (Free extensions of functions)

A function $f\colon A \to A'$ admits free extensions along $A \subseteq B$ if and only if A' and $B \setminus A$ are disjoint. The free extension $g\colon B \to B'$ is defined by $B' = (B \setminus A) \cup A'$ and

$$g(a) = \begin{cases} f(a) & a \in A, \\ a & a \notin A. \end{cases}$$

Proposition (Free extensions of MSA signature morphisms)

Every **MSA** *signature morphism* $\varphi_1\colon (S_1, F_1) \to (S_1', F_1')$ *such that* $(S_1', F_1') \subseteq (S_1, F_1)$ *has free extensions* φ_2 *along any inclusion of signatures* $(S_1, F_1) \subseteq (S_2, F_2)$.

Moreover, for any fixed signature (S_0, F_0)*, we can choose the free extension* $\varphi_2\colon (S_2, F_2) \to (S_2', F_2')$ *such that*

$$(S_0, F_0) \cap (S_2', F_2') \subseteq (S_0, F_0) \cap (S_2, F_2).$$

Single parameterised specifications

Predicates on natural numbers

Example (Predicates on natural numbers)

$$
\begin{array}{ccc}
\left(\begin{array}{l}\{\mathtt{Elt}, \mathtt{Nat}\},\\ \{\mathtt{0}\colon [] \to \mathtt{Nat}, \mathtt{s\_}\colon \mathtt{Nat} \to \mathtt{Nat}\}\end{array}\right) & \xrightarrow{\subseteq} & \left(\begin{array}{l}\{\mathtt{Elt}, \mathtt{Bool}, \mathtt{Nat}\},\\ \{\mathtt{true}\colon [] \to \mathtt{Bool}, \mathtt{false}\colon [] \to \mathtt{Bool}\\ \ \mathtt{P}\colon \mathtt{Elt} \to \mathtt{Bool},\\ \ \mathtt{0}\colon [] \to \mathtt{Nat}, \mathtt{s\_}\colon \mathtt{Nat} \to \mathtt{Nat}\}\end{array}\right) \\
\Big\downarrow {\scriptstyle \mathtt{Elt} \mapsto \mathtt{Nat}} & & \Big\downarrow {\scriptstyle \mathtt{Elt} \mapsto \mathtt{Nat}} \\
\left(\begin{array}{l}\{\mathtt{Nat}\},\\ \{\mathtt{0}\colon [] \to \mathtt{Nat}, \mathtt{s\_}\colon \mathtt{Nat} \to \mathtt{Nat}\}\end{array}\right) & \xrightarrow[\subseteq]{} & \left(\begin{array}{l}\{\mathtt{Nat}, \mathtt{Bool}\},\\ \{\mathtt{0}\colon [] \to \mathtt{Nat}, \mathtt{s\_}\colon \mathtt{Nat} \to \mathtt{Nat},\\ \ \mathtt{true}\colon [] \to \mathtt{Bool}, \mathtt{false}\colon [] \to \mathtt{Bool}\\ \ (\mathtt{P}, \mathtt{Elt}, \mathtt{Bool}, \mathtt{F}_0)\colon \mathtt{Nat} \to \mathtt{Bool}\}\end{array}\right)
\end{array}
$$

Proposition (Free extensions of **OSA** signature morphisms)

- *are obtained by lifting free extensions from the category of* **MSA** *signatures*

$$\begin{array}{ccc}
(S_1, \leq_1, F_1) & \xrightarrow{\subseteq} & (S_2, \leq_2, F_2) \\
\Big\downarrow \varphi_1 & & \Big\downarrow \varphi_2 \\
(S_1', \leq_1', F_1') & \xrightarrow[\subseteq]{} & (S_2', (\varphi_2^{st}(\leq_2) \cup \leq_1')^{m*}, F_2')
\end{array}$$

Proposition (Free extensions of **PA** signature morphisms)

Proposition (Free extensions of OSA signature morphisms)

- *are obtained by lifting free extensions from the category of* **MSA** *signatures*

$$\begin{array}{ccc} (S_1,\ \leq_1, F_1) & \xrightarrow{\subseteq} & (S_2,\ \leq_2, F_2) \\ \downarrow{\scriptstyle \varphi_1} & & \downarrow{\scriptstyle \varphi_2} \\ (S_1',\ \leq_1', F_1') & \xrightarrow[\subseteq]{} & (S_2', (\varphi_2^{st}(\leq_2) \cup \leq_1')^{m*}, F_2') \end{array}$$

Proposition (Free extensions of PA signature morphisms)

Proposition (Free extensions of **OSA** signature morphisms)

- *note that we can no longer always choose* φ_2 *such that*

$$(S_0, \leq_0, F_0) \cap (S'_2, \leq'_2, F'_2) \subseteq (S_0, \leq_0, F_0) \cap (S_2, \leq_2, F_2)$$

for a fixed signature $(S_0, \leq_0, F_0)$

$$(\{\mathtt{s}, \mathtt{s}'\}, \{\mathtt{s} \leq_0 \mathtt{s}'\}, \emptyset)$$

$$\begin{array}{ccc}
\big(\{\mathtt{t}, \mathtt{s}'\}, \emptyset, \emptyset\big) & \xrightarrow{\subseteq} & \big(\{\mathtt{s}, \mathtt{t}, \mathtt{s}'\}, \{\mathtt{s} \leq_2 \mathtt{t}\}, \emptyset\big) \\
{\scriptstyle \mathtt{t} \mapsto \mathtt{s}'} \downarrow & & \downarrow {\scriptstyle \mathtt{t} \mapsto \mathtt{s}'} \\
\big(\{\mathtt{s}'\}, \emptyset, \emptyset\big) & \xrightarrow[\subseteq]{} & \big(\{\mathtt{s}, \mathtt{s}'\}, \{\mathtt{s} \leq'_2 \mathtt{s}'\}, \emptyset\big)
\end{array}$$

Proposition (Free extensions of **PA** signature morphisms)

Proposition (Free extensions of **OSA** signature morphisms)

Proposition (Free extensions of **PA** signature morphisms)

- *are obtained by lifting free extensions from the category of* **MSA** *signatures*

$$\begin{array}{ccc}
(S_1, F_1, TF_1) & \xrightarrow{\subseteq} & (S_2, F_2, TF_2) \\
\Big\downarrow \varphi_1 & & \Big\downarrow \varphi_2 \\
(S_1', F_1', TF_1') & \xrightarrow[\subseteq]{} & (S_2', F_2', \varphi_2^{op}(TF_2) \cup TF_1')
\end{array}$$

- *moreover, for any fixed signature* (S_0, F_0, TF_0)*, we can choose a free extension* φ_2 *such that*

$$(S_0, F_0, TF_0) \cap (S_2', F_2', TF_2') \subseteq (S_0, F_0, TF_0) \cap (S_2, F_2, TF_2)$$

Proposition (Free extensions of **OSA** signature morphisms)

Proposition (Free extensions of **PA** signature morphisms)

- *are obtained by lifting free extensions from the category of* **MSA** *signatures*

$$\begin{array}{ccc}
(S_1, F_1, TF_1) & \xrightarrow{\subseteq} & (S_2, F_2, TF_2) \\
\Big\downarrow{\scriptstyle \varphi_1} & & \Big\downarrow{\scriptstyle \varphi_2} \\
(S'_1, F'_1, TF'_1) & \xrightarrow[\subseteq]{} & (S'_2, F'_2, \varphi_2^{op}(TF_2) \cup TF'_1)
\end{array}$$

- *moreover, for any fixed signature* (S_0, F_0, TF_0)*, we can choose a free extension* φ_2 *such that*

$$(S_0, F_0, TF_0) \cap (S'_2, F'_2, TF'_2) \subseteq (S_0, F_0, TF_0) \cap (S_2, F_2, TF_2)$$

Proposition (Free extensions of OSA signature morphisms)

Proposition (Free extensions of PA signature morphisms)

- *are obtained by lifting free extensions from the category of* **MSA** *signatures*

$$\begin{array}{ccc} (S_1, F_1, TF_1) & \xrightarrow{\subseteq} & (S_2, F_2, TF_2) \\ \downarrow \varphi_1 & & \downarrow \varphi_2 \\ (S'_1, F'_1, TF'_1) & \xrightarrow[\subseteq]{} & (S'_2, F'_2, \varphi_2^{op}(TF_2) \cup TF'_1) \end{array}$$

- *moreover, for any fixed signature* (S_0, F_0, TF_0)*, we can choose a free extension* φ_2 *such that*

$$(S_0, F_0, TF_0) \cap (S'_2, F'_2, TF'_2) \subseteq (S_0, F_0, TF_0) \cap (S_2, F_2, TF_2)$$

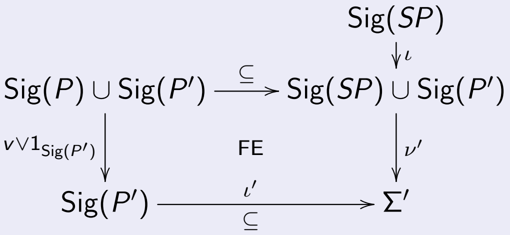
Definition (Instantiation of parameters)

- consider a parameterised specification $SP(P)$ and
- a specification morphism $v\colon P \to P'$ that preserves P'

The instantiation of the parameterised specification $SP(P)$ by v is

$$SP(P \Leftarrow v) = SP \star (\iota; \nu') \cup P' \star \iota'$$

given by the *free extension* depicted below.



Example (Lists of natural numbers via free extensions)

$$\begin{pmatrix} \{\mathtt{Elt}, \mathtt{Nat}\}, \\ \{0 \colon [] \to \mathtt{Nat}, \mathtt{s\_} \colon \mathtt{Nat} \to \mathtt{Nat}\} \end{pmatrix} \xrightarrow{\subseteq} \begin{pmatrix} \{\mathtt{Elt}, \mathtt{List}, \mathtt{Nat}\}, \\ \{\mathtt{nil} \colon [] \to \mathtt{List}, \mathtt{\_\_} \colon \mathtt{Elt}\ \mathtt{List} \to \mathtt{List} \\ \ 0 \colon [] \to \mathtt{Nat}, \mathtt{s\_} \colon \mathtt{Nat} \to \mathtt{Nat}\} \end{pmatrix}$$

$$\downarrow \mathtt{Elt} \mapsto \mathtt{Nat} \qquad\qquad \downarrow \mathtt{Elt} \mapsto \mathtt{Nat}$$

$$\begin{pmatrix} \{\mathtt{Nat}\}, \\ \{0 \colon [] \to \mathtt{Nat}, \mathtt{s\_} \colon \mathtt{Nat} \to \mathtt{Nat}\} \end{pmatrix} \xrightarrow[\subseteq]{} \begin{pmatrix} \{\mathtt{Nat}, \mathtt{List}\}, \\ \{0 \colon [] \to \mathtt{Nat}, \mathtt{s\_} \colon \mathtt{Nat} \to \mathtt{Nat}, \\ \ \mathtt{nil} \colon [] \to \mathtt{List}, \mathtt{\_\_} \colon \mathtt{Nat}\ \mathtt{List} \to \mathtt{List}\} \end{pmatrix}$$

Multiple parameterised specifications

The definition

Definition

A *multiple parameterised specification* is a specification with several parameters, denoted $SP(P_1, \ldots, P_n)$.

Example (ELT $\xrightarrow{\iota_1}$ PAIR $\xleftarrow{\iota_2}$ ELT)

```
mod* ELT {
  [ Elt ]
}
```

```
mod! PAIR(E1 :: ELT, E2 :: ELT) {
  [ Pair ]
  op <_,_> : Elt.E1 Elt.E2 -> Pair
}
```

- for any multiple parameterised specification $SP(P_1, \ldots, P_n)$ we have that $SP(P_1 \cup \cdots \cup P_n)$ is a single parameterised specification

Proposition

Let $\{v_i \colon P_i \to P'_i \mid 1 \leq i \leq n\}$ *be a set of pairwise compatible morphisms. Then there exists* $\bigvee v_i \colon \bigcup P_i \to \bigcup P'_i$ *such that*

$$\left(P_j \subseteq \bigcup P_i\right); \bigvee v_i = v_j; \left(P'_j \subseteq \bigcup P'_i\right), \quad \textit{for any } 1 \leq j \leq n.$$

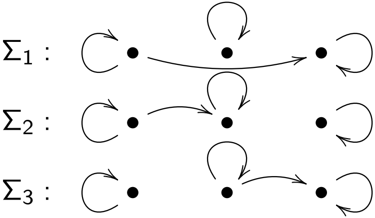
Moreover, if the inclusion system is distributive, for any set of signatures $\{Q_j \mid 1 \leq j \leq m\}$, *if* v_i *preserves* Q_j, *for all* $1 \leq i \leq n$ *and* $1 \leq j \leq m$, *then the join* $\bigvee v_i$ *preserves the union* $\bigcup Q_j$.

Multiple parameterised specifications
On distributivity

Proposition

The **MSA** *inclusion system and the* **PA** *inclusion system are both distributive.*

- note that the **OSA** inclusion system is not distributive

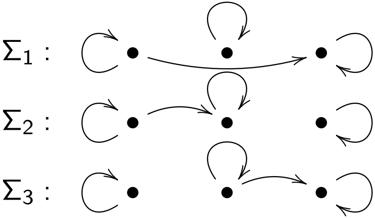


$$\Sigma_1 \cap (\Sigma_2 \cup \Sigma_3) \supsetneq (\Sigma_1 \cap \Sigma_2) \cup (\Sigma_1 \cap \Sigma_3)$$

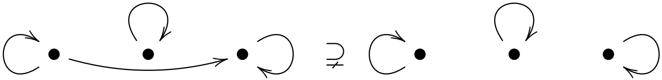
Proposition

The **MSA** *inclusion system and the* **PA** *inclusion system are both distributive.*

- note that the **OSA** inclusion system is not distributive



$$\Sigma_1 \cap (\Sigma_2 \cup \Sigma_3) \supsetneq (\Sigma_1 \cap \Sigma_2) \cup (\Sigma_1 \cap \Sigma_3)$$



Definition (Simultaneous instantiation of parameters)

Let us consider a multiple parameterised specification $\Sigma(P_1, \ldots, P_n)$ and a set of pairwise compatible morphisms $\{v_i \colon P_i \to P'_i \mid 1 \leq i \leq n\}$ such that any morphism v_i preserves any specification P'_j, for $1 \leq i, j \leq n$.

The *simultaneous instantiation* of $\Sigma(P_1, \ldots, P_n)$ by $\{v_1, \ldots, v_n\}$, denoted

$$\Sigma(\{P_i \Leftarrow v_i \mid 1 \leq i \leq n\}),$$

is defined as the single parameter instantiation

$$\Sigma(\bigsqcup P_i \Leftarrow \bigvee v_i).$$

Example (Pairs of natural numbers and Boolean values)

$$
\begin{array}{ccc}
\left(\begin{array}{l}
\{\mathtt{Elt}_1, \mathtt{Elt}_2, \mathtt{Nat}, \mathtt{Bool}\}, \\
\{0\colon [] \to \mathtt{Nat}, \mathtt{s}\_\colon \mathtt{Nat} \to \mathtt{Nat}, \\
\ \mathtt{true}\colon [] \to \mathtt{Bool}, \mathtt{false}\colon [] \to \mathtt{Bool}\}
\end{array}\right)
& \xrightarrow{\subseteq} &
\left(\begin{array}{l}
\{\mathtt{Elt}_1, \mathtt{Elt}_2, \mathtt{Pair}, \mathtt{Nat}, \mathtt{Bool}\}, \\
\{\langle\_,\_\rangle\colon \mathtt{Elt}_1\ \mathtt{Elt}_2 \to \mathtt{Pair}, \\
\ 0\colon [] \to \mathtt{Nat}, \mathtt{s}\_\colon \mathtt{Nat} \to \mathtt{Nat}, \\
\ \mathtt{true}\colon [] \to \mathtt{Bool}, \mathtt{false}\colon [] \to \mathtt{Bool}\}
\end{array}\right) \\
\Big\downarrow {\scriptstyle \mathtt{Elt}_1 \mapsto \mathtt{Nat},\ \mathtt{Elt}_2 \mapsto \mathtt{Bool}} & & \Big\downarrow {\scriptstyle \mathtt{Elt}_1 \mapsto \mathtt{Nat},\ \mathtt{Elt}_2 \mapsto \mathtt{Bool}} \\
\left(\begin{array}{l}
\{\mathtt{Nat}, \mathtt{Bool}\}, \\
\{0\colon [] \to \mathtt{Nat}, \mathtt{s}\_\colon \mathtt{Nat} \to \mathtt{Nat}, \\
\ \mathtt{true}\colon [] \to \mathtt{Bool}, \mathtt{false}\colon [] \to \mathtt{Bool}\}
\end{array}\right)
& \xrightarrow[\subseteq]{} &
\left(\begin{array}{l}
\{\mathtt{Nat}, \mathtt{Bool}, \mathtt{List}\}, \\
\{0\colon [] \to \mathtt{Nat}, \mathtt{s}\_\colon \mathtt{Nat} \to \mathtt{Nat}, \\
\ \mathtt{true}\colon [] \to \mathtt{Bool}, \mathtt{false}\colon [] \to \mathtt{Bool}, \\
\ \langle\_,\_\rangle\colon \mathtt{Nat}\ \mathtt{Bool} \to \mathtt{Pair}\}
\end{array}\right)
\end{array}
$$

Proposition

Let $\Sigma(P_1, \ldots, P_n)$ be a multiple parameterised specification and $v_i\colon P_i \to P'_i$ a morphism that preserves the instance P'_i and all parameter specifications P_j, for $1 \leq j \neq i \leq n$.

If the instantiation of parameters is based on free extensions then $\Sigma(P_i \Leftarrow v_i)$ is a parameterised specification, with the parameters $\{P_j \mid 1 \leq j \neq i \leq n\}$.

Definition (Sequential instantiation of parameters)

Let us consider a multiple parameterised specification $\Sigma(P_1, \ldots, P_n)$ and a set of pairwise compatible morphisms $\{v_i \colon P_i \to P'_i \mid 1 \leq i \leq n\}$ such that for every $1 \leq i \leq n$, v_i preservers P'_i and all P_j, where $1 \leq j \neq i \leq n$.

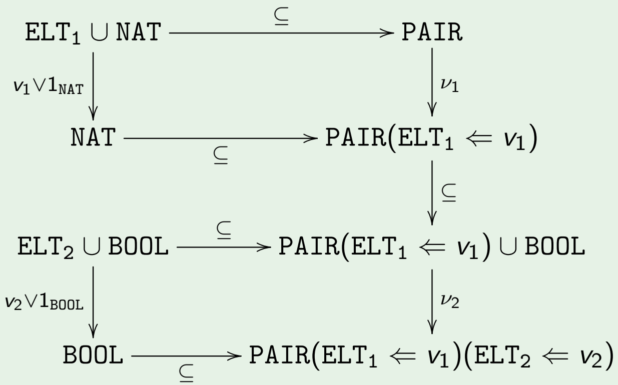
The *sequential instantiation* of $\Sigma(P_1, \ldots, P_n)$ by $\{v_1, \ldots, v_n\}$, denoted

$$\Sigma(P_i \Leftarrow v_i)_{1 \leq i \leq n},$$

is defined as the iterated (single) parameter instantiation

$$\Sigma(P_0 \Leftarrow v_0) \cdots (P_{n-1} \Leftarrow v_{n-1}).$$

Example (Pairs of natural numbers and Boolean values)



Theorem

Let $\Sigma(P_1, \ldots, P_n)$ *be a multiple parameterised specification and* $\{v_i \colon P_i \to P'_i \mid 1 \leq i \leq n\}$ *a set of morphisms such that for every* $1 \leq i \leq n$, v_i *preserves* P_j *and* P'_k, *for all* $1 \leq j \neq i, k \leq n$.

If the instantiation of parameters is based on free extensions then the simultaneous and the sequential instantiation procedures produce isomorphic results, provided that for any morphism $v \colon P \to P'$ *for which we consider free extensions, and any signature* Q *preserved by* v, *we can choose a free extension of* v *that strongly preserves* Q.

$$\Sigma\left(\bigsqcup P_i \Leftarrow \bigvee v_i\right) \quad \cong \quad \Sigma(P_i \Leftarrow v_i)_{1 \leq i \leq n}$$

- when the additional constraint doesn't hold, intricate sharing between the instances of the parameters and the body of the parameterised specification may lead to non-isomorphic results of the two instantiation procedures

Example (Pairs with an observation)

$$\big(\{\mathtt{Elt}_1\},\emptyset\big) \xrightarrow{\subseteq} \left(\begin{array}{l}\{\mathtt{Elt}_1,\mathtt{Elt}_2,\mathtt{Pair}\} \\ \{\langle\_,\_\rangle\colon \mathtt{Elt}_1\ \mathtt{Elt}_2 \to \mathtt{Pair}, \\ \ \mathtt{obs}\colon \mathtt{Pair} \to \mathtt{Elt}_1\}\end{array}\right) \xleftarrow{\subseteq} \big(\{\mathtt{Elt}_2\},\emptyset\big)$$

$$\big(\{\mathtt{Elt}_1\},\emptyset\big) \xrightarrow{\mathtt{Elt}_1 \mapsto \mathtt{Nat}} \left(\begin{array}{l}\{\mathtt{Nat}\}, \\ \{0\colon [] \to \mathtt{Nat}, \mathtt{s}\_\colon \mathtt{Nat} \to \mathtt{Nat}\}\end{array}\right)$$

$$\big(\{\mathtt{Elt}_2\},\emptyset\big) \xrightarrow{\mathtt{Elt}_2 \mapsto \mathtt{Nat}} \left(\begin{array}{l}\{\mathtt{Nat},\mathtt{Pair}\}, \\ \{0\colon [] \to \mathtt{Nat}, \mathtt{s}\_\colon \mathtt{Nat} \to \mathtt{Nat}, \\ \ \mathtt{obs}\colon \mathtt{Pair} \to \mathtt{Nat}\}\end{array}\right)$$

- when the additional constraint doesn't hold, intricate sharing between the instances of the parameters and the body of the parameterised specification may lead to non-isomorphic results of the two instantiation procedures

Example (Pairs with an observation)

- by simultaneous instantiation we may obtain

$$\mathtt{PAIR}^{\mathsf{obs}}(\mathtt{ELT}_1 \cup \mathtt{ELT}_2 \Leftarrow v_1 \vee v_2) = \left[\begin{array}{l} \{\mathtt{Nat}, \mathtt{Pair}\} \\ \{0\colon [] \to \mathtt{Nat}, \mathtt{s\_}\colon \mathtt{Nat} \to \mathtt{Nat}, \\ \quad \langle \_, \_ \rangle\colon \mathtt{Nat}\ \mathtt{Nat} \to \mathtt{Pair}, \\ \quad \mathtt{obs}\colon \mathtt{Pair} \to \mathtt{Nat}, \\ \quad \mathtt{obs}'\colon \mathtt{Pair} \to \mathtt{Nat}\} \end{array}\right]$$

- when the additional constraint doesn't hold, intricate sharing between the instances of the parameters and the body of the parameterised specification may lead to non-isomorphic results of the two instantiation procedures

Example (Pairs with an observation)

- by sequential instantiation we may obtain

$$\mathtt{PAIR}^{\mathsf{obs}}(\mathtt{ELT}_1 \Leftarrow v_1) = \left(\begin{array}{l} \{\mathtt{Nat}, \mathtt{Elt}_2, \mathtt{Pair}\} \\ \{\mathtt{0}\colon [] \to \mathtt{Nat}, \mathtt{s}\_\colon \mathtt{Nat} \to \mathtt{Nat}, \\ \ \langle \_, \_ \rangle \colon \mathtt{Nat}\ \mathtt{Elt}_2 \to \mathtt{Pair}, \\ \ \mathtt{obs}\colon \mathtt{Pair} \to \mathtt{Nat}\} \end{array} \right)$$

$$\mathtt{PAIR}^{\mathsf{obs}}(\mathtt{ELT}_1 \Leftarrow v_1)(\mathtt{ELT}_2 \Leftarrow v_2) = \left(\begin{array}{l} \{\mathtt{Nat}, \mathtt{Pair}\} \\ \{\mathtt{0}\colon [] \to \mathtt{Nat}, \mathtt{s}\_\colon \mathtt{Nat} \to \mathtt{Nat}, \\ \ \langle \_, \_ \rangle \colon \mathtt{Nat}\ \mathtt{Nat} \to \mathtt{Pair}, \\ \ \mathtt{obs}\colon \mathtt{Pair} \to \mathtt{Nat}\} \end{array} \right)$$

Thank you!