An Institutional Approach to Modularisation in Event-B

Proof of Institution and Amalgamation Properties

Marie Farrell*, Rosemary Monahan, and James F. Power

Dept. of Computer Science, Maynooth University, Co. Kildare, Ireland

Abstract. By defining an institution $\mathcal{EVT}$ for Event-B we have increased the modularity and potential for interoperability of the Event-B language. This document contains the proofs that $\mathcal{EVT}$ is a valid institution and that it supports the relevant modularisation constructs.

Since an institution is composed of signatures, sentences, models and a satisfaction relation there are a number of proofs that need to be undertaken in order to prove its validity. Section 1 provides the proof that signatures and signature morphisms over $\mathcal{EVT}$ define a category **Sign**. Section 2 provides the proof that **Sen** is indeed a functor mapping each signature to a corresponding set of sentences. Section 3 provides a proof that **Mod** is actually a functor. This proof shows that models over $\mathcal{EVT}$ signatures form a category and that the model reduct is a functor mapping models over one signature to models over another. In section 4 we define the satisfaction relation for $\mathcal{EVT}$ and then we prove that it meets the requirements imposed by the satisfaction condition for institutions in general.

$\mathcal{EVT}$ uses specification building operators to provide modularisation constructs for Event-B but, in order to use them, it is necessary to prove that all pushouts exist in the category of signatures, **Sign**, and that every such pushout diagram admits amalgamation. These proofs are provided in Section 5.

1 Sign, the category of $\mathcal{EVT}$ signatures

In this section we prove that signatures over $\mathcal{EVT}$ and their respective morphisms form a category. In particular, we prove we can compose signature morphisms, that this composition is associative and that there exists an identity signature morphism.

Lemma 1. *Signatures and signature morphisms define a category **Sign**. The objects are signatures and the arrows are signature morphisms*

* This project is funded by the Irish Research Council, email: `mfarrell@cs.nuim.ie`

Proof. Let $\Sigma = \langle S, \Omega, \Pi, E, V \rangle$ be a signature where $\langle S, \Omega, \Pi \rangle$ is a signature over $\mathcal{FOPEQ}$, the institution for first-order logic with equality [3]. E is a set of event names and V is a set of sort indexed variable names.

We define the **signature morphism** $\sigma : \Sigma \to \Sigma'$ to be a five-tuple containing σ_S, σ_Ω, σ_Π, σ_E and σ_V. Here σ_S, σ_Ω, σ_Π are the mappings taken from the corresponding signature morphism in $\mathcal{FOPEQ}$ with composition, associativity thereof and identities as in $\mathcal{FOPEQ}$. In particular,

$\sigma_S : S \to S'$ is a function mapping sort names to sort names.
$\sigma_\Omega : \Omega \to \Omega'$ is a family of functions mapping operation names in Ω, respecting the arities and result sorts.
$\sigma_\Pi : \Pi \to \Pi'$ is a family of functions mapping the predicate names in Π, respecting the arities and sorts.
$\sigma_E : E \to E'$ is a function mapping event names. Every set of event names E must contain a distinguished event name `Initialisation`, and σ_E maps the initialisation event in Σ to the initialisation event in Σ'.
$\sigma_V : V \to V'$ is a sort-preserving function on variable names (working in a similar manner to the sort-preserving mapping for constant symbols under σ_Ω).

Composition of signature morphisms:

- Signature morphisms can be composed. Since event names are not concerned with sort/arity it is easy to see that the composition holds for σ_E.

  Taking $V = \{v_1 : s_1, v_2 : s_2, v_3 : s_1, ...\}$ then the composition for σ_V is given by:

  $$\sigma_2(\sigma_1(v_i : s_i)) = \sigma_2(\sigma_{1_V}(v_i) : \sigma_{1_S}(s_i)) = \sigma_{2_V}(\sigma_{1_V}(v_i)) : \sigma_{2_S}(\sigma_{1_S}(s_i))$$

  Let $\sigma_1 : \Sigma_1 \to \Sigma_2$ and $\sigma_2 : \Sigma_2 \to \Sigma_3$. We can check that $\sigma_2 \circ \sigma_1$ is actually a morphism.

  - For all $e \in E$ we have that $\sigma_1(e) \in \Sigma_2$ and $\sigma_2(e) \in \Sigma_3$. Therefore $\sigma_2(\sigma_1(e)) \in \Sigma_3$ so
    $$e \in \Sigma_1 \Rightarrow \sigma_2 \circ \sigma_1(e) \in \Sigma_3$$
  - For all $(v_i : s_i) \in V$ we have that $\sigma_1(v_i : s_i) \in \Sigma_2$ and $\sigma_2(v_i : s_i) \in \Sigma_3$. Therefore $\sigma_2(\sigma_1(v_i : s_i)) \in \Sigma_3$ so
    $$(v_i : s_i) \in \Sigma_1 \Rightarrow \sigma_2 \circ \sigma_1(v_i : s_i) \in \Sigma_3$$

- Composition of signature morphisms is associative, i.e.

  $$(\sigma_3 \circ \sigma_2) \circ \sigma_1 = \sigma_3 \circ (\sigma_2 \circ \sigma_1)$$

  For $e \in E$: $\sigma_2 \circ \sigma_1(e) = \sigma_2(\sigma_1(e))$ and so $\sigma_3 \circ (\sigma_2 \circ \sigma_1)(e) = \sigma_3(\sigma_2(\sigma_1(e)))$ by the definition of composition. This is equal to $\sigma_3 \circ \sigma_2 \circ (\sigma_1(e))$ which is the same as $(\sigma_3 \circ \sigma_2) \circ \sigma_1(e)$. The proof for V is similar to the proof for Ω in $\mathcal{FOPEQ}$.

Identity morphism for signatures: For any signature Σ, there exists an identity signature morphism $id_\Sigma : \Sigma \rightarrow \Sigma$.

id_E and id_V are such that $id_E(e) = e$ and $id_V(v : s) = v : id_S(s) = v : s$. This morphism satisfies the signature morphism condition since $e \in E \Rightarrow id_E(e) \in E$ and $v : s \in V \Rightarrow id_V(v : s) \in V$.

□

2 The functor Sen, giving $\mathcal{EVT}$ sentences

In this section we prove that **Sen** is a functor that provides for each signature a set of sentences and for each signature morphism a function mapping the corresponding sentences. In particular, we prove that **Sen** preserves composition and identity of signature morphisms.

Lemma 2. *There is a functor* ***Sen*** : ***Sign*** $\rightarrow$ ***Set*** *giving for each signature Σ a set of sentences (objects in the category* ***Set****) and for each signature morphism $\sigma : \Sigma_1 \rightarrow \Sigma_2$ (arrows in the category* ***Sign****) a function $Sen(\sigma) : Sen(\Sigma_1) \rightarrow Sen(\Sigma_2)$ (arrows in the category* ***Set****) translating sentences.*

Proof. For any $\Sigma = \langle S, \Omega, \Pi \rangle$ in $\mathcal{FOPEQ}$, a Σ-sentence is a set of closed first-order formulae built out of atomic formulae using $(\wedge, \vee, \neg, \Rightarrow, \iff, \exists, \forall)$. Formulae are term equalities where the terms are $\langle S, \Omega \rangle$-terms (algebraic terms) over the predicates, variables and `true` and `false`.

Taking $\Sigma = \langle S, \Omega, \Pi, E, V \rangle$, we define two types of **sentences** over $\mathcal{EVT}$:

- The first represents an *invariant definition* in an Event-B specification. It consists of pair $\langle inv, \phi \rangle$ where inv is a fixed keyword to identify invariant sentences and ϕ is a $\mathcal{FOPEQ}$ sentence whose only free variables are drawn from V.
- The second represents an *event definition* in an Event-B specification. It consists of a pair $\langle e, \phi \rangle$ where $e \in E$ and ϕ is a $\mathcal{FOPEQ}$ sentence whose only free variables are drawn from V and V'. Pairing ϕ with an event name e provides a definition for that event.

Note: V' is V with all of the variable names primed to denote their value after the operation. We assume the existence of a suitable renaming function ι that primes variable names.

Sentence morphisms: **Sen** is a functor therefore it is necessary to map the signature morphisms to corresponding functions over sentences. The functor maps morphisms to sentence morphisms respecting sort, arity and initialisation events. The domain and codomain of $Sen(\sigma)$ are their respective images under σ.

Composition of sentence morphisms: We can show that $Sen(\sigma_2 \circ \sigma_1) = Sen(\sigma_2) \circ Sen(\sigma_1)$.

$Sen(\sigma_2) \circ Sen(\sigma_1)$ is the application of the signature morphism σ_1 a sentence $\langle e, \phi \rangle$ or $\langle inv, \phi \rangle$ composed with the application of σ_2 and since signature morphisms can be composed this is the same as $Sen(\sigma_2 \circ \sigma_1)$.

Identity morphism for sentences: Let id_{Σ_1} be an identity signature morphism as defined in Lemma 1. Since signature morphisms already preserve identity and $Sen(id_{\Sigma_1})$ is the application of the identity signature morphisms to every element of the sentence, then the identities are preserved.

□

3 The functor Mod, giving $\mathcal{EVT}$ models

In this section we prove that every signature corresponds to a category of models with model morphisms as arrows (Lemma 3). Then, for each signature morphism we define the model reduct as a functor from models over one signature to models over another (Lemma 4). Finally, we prove that **Mod** is actually a functor (Lemma 5).

Lemma 3. *For any signature Σ there is a category of models* ***Mod***(Σ) *where the objects in the category are models and the arrows are model morphisms.*

Proof. We begin the proof by defining a model of $\mathcal{FOPEQ}$. Given a signature $\Sigma = \langle S, \Omega, \Pi \rangle$, a model over $\mathcal{FOPEQ}$ consists of a carrier set $|A|_s$ for each sort name $s \in S$, a function $f_A : |A|_{s_1} \times ... \times |A|_{s_n} \rightarrow |A|_s$ for each operation name $f \in \Omega_{s_1 \ldots s_n, s}$ and a relation $p_A \subseteq |A|_{s_1} \times ... \times |A|_{s_n}$ for each predicate name $p \in \Pi_{s_1 \ldots s_n}$.

Given $\Sigma = \langle S, \Omega, \Pi, E, V \rangle$, **Mod**$(\Sigma)$ provides a category of models where a **model** over Σ is composed of a $\mathcal{FOPEQ}$-model paired with a relation R. For each event/variable name pair $(e, v) \in E \times V$, the relation R contains a $(e, v$-indexed) relation over the carrier set of the corresponding variable's sort.

Intuitively, a model over Σ maps each pair (e, v), consisting of an event and variable name, to a relation over the sort carrier set of v, where each maplet in this relation maps a *before*-value to an *after*-value for v when used in event e.

If $V = (v_1, \ldots, v_n)$ are the variables from Σ, then for any event e, we defined $R_e = (R_{e,v_1} \times \ldots \times R_{e,v_n})$, the set of n-tuples, each denoting a possible before/after value for the variables in e.

The kth element of R_e has the form $((b_{1,k}, a_{1,k}), \ldots, (b_{n,k}, a_{n,k}))$ and from this we form the functions $b_{e,k} = \{v_1 \mapsto b_{1,k}, \ldots, v_n \mapsto b_{n,k}\}$ and $a_{e,k} = \{v_1 \mapsto a_{1,k}, \ldots, v_n \mapsto a_{n,k}\}$ For example, given event name *inc* and integer variable x and boolean variable y then an example of a model might be:

$$\Big\langle A, R = \big\{\{..., \langle 1, 2\rangle, \langle 2, 3\rangle, ...\}_{\mathrm{inc,x}}, \{\langle \mathtt{false}, \mathtt{false}\rangle, \langle \mathtt{true}, \mathtt{false}\rangle\}_{\mathrm{inc,y}}\big\}\Big\rangle$$

where A is a standard first-order model over $\langle S, \Omega, \Pi \rangle$ providing semantics for the sorts, operations and predicates in the usual way for first-order logic. In this example, this model would correspond to an event that incremented the integer variable by 1 and set the boolean variable to false.

Model morphisms: In $\mathcal{FOPEQ}$ a model morphism $h : A_1 \to A_2$ is a family of functions $h = \langle h_s :: |A_1|_s \to |A_2|_s \rangle_{s \in S}$ which respects the sorts and arities of the operations and predicates. $\mathcal{EVT}$ models have the form $\langle A, R \rangle$ so $\mathcal{EVT}$-morphisms are given by the $\mathcal{FOPEQ}$-morphisms for A applied to the relation for each (e, v) pair in R.

Thus there is a model morphism $\mu : \langle A_1, R_1 \rangle \to \langle A_2, R_2 \rangle$ if there is a $\mathcal{FOPEQ}$-model morphism $h : A_1 \to A_2$ and we extend this to the value-pairs in the relation R. That is, for any element $\langle d_b, d_a \rangle_{e,v}$ in R_1 we have the mapping $\langle h(d_b), h(d_a) \rangle_{e,v}$ in R_2. The composition of model morphisms, their associativity and identity derives from that of $\mathcal{FOPEQ}$.

Composition of model morphisms: Let $M_i = \langle A_i, R_i \rangle$ be a model and $h_i : M_i \to M_{i+1}$ be a model morphism where $i \in \{1, 2, 3...\}$

Composition of model morphisms is associative:

$$\begin{aligned}
(h_3 \circ h_2) \circ h_1 &= h_3 \circ (h_2 \circ h_1) \\
(h_3 \circ h_2)(h_1(M_1)) &= h_3 \circ (h_2(h_1(M_1))) \\
(h_3 \circ h_2)(M_2) &= h_3 \circ (h_2(M_2)) \\
h_3(h_2(M_2)) &= h3(h_2(M_2)) \\
h_3(M_3) &= h_3(M_3) \\
M_4 &= M_4
\end{aligned}$$

Identity morphism for models: For any model M_i there exists an identity model morphism $h_{id} : M_i \to M_i$. If $M_i = \langle A_i, R_i \rangle$ then $h_{id}(M_i) = \langle A_i, R_i \rangle$

□

Lemma 4. *For each signature morphism $\sigma : \Sigma_1 \to \Sigma_2$ the model reduct is a functor $\boldsymbol{Mod}(\sigma)$ from Σ_2-models to Σ_1-models.*

Proof. Let $M_2 = \langle A_2, R_2 \rangle$ be a Σ_2-model respectively. Then the reduct $M_2|_\sigma$ collapses the model to only contain signature items supported by Σ_1 and consists of the pair $M_2|_\sigma = \langle A_2|_\sigma, R_2|_\sigma \rangle$ such that

- $A_2|_\sigma$ is the reduct as defined in $\mathcal{FOPEQ}$
- $R_2|_\sigma$ is defined as follows: For any event-variable name pair (e_1, v_1) in Σ_1 there is a corresponding event-variable name pair $(\sigma(e_1), \sigma(v_1))$ in Σ_2. Then for every value pair $\langle d_b, d_a \rangle_{\sigma(e_1), \sigma(v_1)} \in R_2$ we have $\langle d_b, d_a \rangle_{e_1, v_1} \in R_2|_\sigma$.

Preservation of composition for model reducts: Given model morphisms
$h_1 : M_1 \to M_2$, $h_2 : M_2 \to M_3$: we must show $(h_2 \circ h_1)|_\sigma = h_2|_\sigma \circ h_1|_\sigma$.

For any $r \in R$, $(h_2 \circ h_1)|_\sigma = (h_2 \circ h_1)_\sigma(r)$ which by definition of composition of morphisms is $(h_2)_\sigma \circ ((h_1)_\sigma(r))$ which equals $((h_2)_\sigma \circ (h_1)_\sigma)(r)$ which is $h_2|_\sigma \circ h_1|_\sigma$

Preservation of identities for model reducts: The reduct of the identity is the identity.

Let id_{M_2} be an identity Σ_2-morphism then $id_{M_2}|_\sigma$ is an identity Σ_1-morphism h_1 defined by $h_1(r) = id_{M_2}|_\sigma(r) = r$ for any $r \in R$.

For the components belonging to A these proofs follow the corresponding proofs in $\mathcal{FOPEQ}$. □

Lemma 5. *There is a functor **Mod** giving a category **Mod**(Σ) of models for each signature Σ, and for each signature morphism $\sigma : \Sigma_1 \rightarrow \Sigma_2$ a functor **Mod**(σ) from Σ_2-models to Σ_1-models.*

Proof. Proving that **Mod** is a functor:

For each $\sigma : \Sigma_1 \rightarrow \Sigma_2$ in **Sign** there is an arrow in $\mathbf{Sign}^{op}$ going in the opposite direction. By Lemma 4, the image of this arrow in $\mathbf{Sign}^{op}$ is $\mathbf{Mod}(\sigma) : \mathbf{Mod}(\Sigma_2) \rightarrow \mathbf{Mod}(\Sigma_1)$ in **Cat**. By Lemma 3, the image of a signature **Sign** is an object $\mathbf{Mod}(\Sigma)$ in **Cat**. Therefore, domain and codomain of the image of an arrow are the images of the domain and codomain respectively.

Preservation of composition: $\mathbf{Mod}(\sigma_2 \circ \sigma_1) = \mathbf{Mod}(\sigma_2) \circ \mathbf{Mod}(\sigma_1)$
Let $\sigma_1 : \Sigma_1 \rightarrow \Sigma_2$ and $\sigma_2 : \Sigma_2 \rightarrow \Sigma_3$ be signature morphisms and let $M_i = \langle A_i, R_i \rangle$ be a model over Σ_i and let h_i be a Σ_i-model morphism. $i \in \{1, 2, 3\}$.

- $M_3|_{\sigma_2 \circ \sigma_1} = (M_3|_{\sigma_2})|_{\sigma_1}$
  By definition of reduct $M_3|_{\sigma_2} = \langle A_3, R_3 \rangle|_{\sigma_2} = \langle A_2, R_2 \rangle = M_2$.
  Then $(M_3|_{\sigma_2})|_{\sigma_1} = M_2|_{\sigma_1} = \langle A_2, R_2 \rangle|_{\sigma_1} = \langle A_1, R_1 \rangle = M_1$.
  By composition of signature morphisms $\sigma_2 \circ \sigma_1 : \Sigma_1 \rightarrow \Sigma_3$. So $M_3|_{\sigma_2 \circ \sigma_1} = \langle A_3, R_3 \rangle|_{\sigma_2 \circ \sigma_1} = \langle A_1, R_1 \rangle = M_1$
  Therefore $M_3|_{\sigma_2 \circ \sigma_1} = (M_3|_{\sigma_2})|_{\sigma_1}$

- $h_3|_{\sigma_2 \circ \sigma_1} = (h_3|_{\sigma_2})|_{\sigma_1}$
  Proof similar to above.

Preservation of identities: Let id_{Σ_1} be an identity signature morphism as defined in Lemma 1. Since signature morphisms already preserve identity and $Mod(id_{\Sigma_1})$ is the application of the identity signature morphisms to every part of the model, then the identities are preserved.

□

4 The Satisfaction relation for $\mathcal{EVT}$

In this section we prove that the satisfaction condition that we have defined for $\mathcal{EVT}$ preserves the satisfaction condition imposed on institutions. *"Truth is invariant under change of notation"* [1].

Satisfaction over $\mathcal{EVT}$: The satisfaction relation for sentences over $\mathcal{EVT}$ is broken into two parts: satisfaction of invariant sentences and satisfaction of event sentences.

Invariant sentences: Given a Σ-model $\langle A, R\rangle$ and some invariant sentence $\langle inv, \phi\rangle$ over $\mathcal{EVT}$, we define $\langle A, R\rangle \models_{\mathcal{EVT}} \langle inv, \phi\rangle$ if and only if for each event name e in the signature Σ we have

$$A \ \models_{\mathcal{FOPEQ}} \ \phi[\overline{x}/b_{e,k}] \wedge \phi[\overline{x}/a_{e,k}]$$

where $\phi[\overline{x}/b_{e,k}]$ (resp. $\phi[\overline{x}/a_{e,k}]$) denotes the evaluation of ϕ using the variable-to-value mapping for the before-state given by $b_{e,k}$ (resp. the after-state given by $a_{e,k}$).

Event sentences: Given a Σ-model $\langle A, R\rangle$ and some event sentence $\langle e, \phi\rangle$ over $\mathcal{EVT}$ we define $\langle A, R\rangle \models_{\mathcal{EVT}} \langle e, \phi\rangle$ if and only if

$$A \ \models_{\mathcal{FOPEQ}} \ \phi[\overline{x}/b_{e,k}][\iota^{-1}(\overline{x}')/a_{e,k}]$$

where ι^{-1} un-primes the variable names, and thus ϕ is evaluated using the variable-to-value mapping given by $b_{e,k}$ and then $a_{e,k}$ as above.

An embedding from $\mathcal{EVT}$ models to $\mathcal{FOPEQ}$ models: To simplify the proof of the satisfaction condition, we show how to fold an $\mathcal{EVT}$ model into an $\mathcal{FOPEQ}$ model.

The intuition here is that the set of before- and after- state values for the free variables in an $\mathcal{EVT}$-signature are represented by a distinguished relation symbol in the corresponding $\mathcal{FOPEQ}$ signature, with models and sentences similarly updated to reflect this. For any event e, we expect to be able to write this predicate in the form $r_e(\overline{x}, \overline{x}')$ where $\overline{x}$ and $\overline{x}'$ correspond to tuples of the free variables from the corresponding $\mathcal{EVT}$-signature (unprimed and primed). Of course, such variables must be internalised as bound variables in $\mathcal{FOPEQ}$.

Given the $\mathcal{EVT}$ Σ-model $\langle A, R\rangle$, we construct:

- The $\mathcal{FOPEQ}$ signature Σ^R, which is just the $\mathcal{FOPEQ}$ component of Σ augmented with a new predicate symbol r_e for each event name e. This relation has arity and sorts corresponding to the number and sort of the variables in the V component of Σ (doubled), which we assume to be in a fixed order.
- The $\mathcal{FOPEQ}$ model A^R, which is just the $\mathcal{FOPEQ}$ component A of the $\mathcal{EVT}$ model, with an added interpretation that maps each predicate symbol r_e to the relation $\{(b_{1,k}, \ldots, b_{n,k}) \mapsto (a_{1,k}, \ldots, a_{n,k}) \mid ((b_{1,k}, a_{1,k}), \ldots, (b_{n,k}, a_{n,k})) \in R_e, 1 \leq k \leq |R_e|\}$, where $V = (v_1, \ldots, v_n)$ are the variables from Σ as before.

For example, in section 3 we described an example for an event named *inc* and variable names x and y with the model containing:

$$\Big\langle A, R = \big\{\{\ldots, \langle 1, 2\rangle, \langle 2, 3\rangle, \ldots\}_{\text{inc,x}}, \{\langle \texttt{false}, \texttt{false}\rangle, \langle \texttt{true}, \texttt{false}\rangle\}_{\text{inc,y}}\big\}\Big\rangle$$

In this case, the corresponding $\mathcal{FOPEQ}$ model A^R would map the predicate symbol r_{inc} to the relation that contains the mappings for tuples of type $(int \times bool)^2$ including

$$\left\{\begin{array}{c} \ldots \\ (1,\mathtt{false}){\mapsto}(2,\mathtt{false}), (1,\mathtt{true}){\mapsto}(2,\mathtt{false}), \\ (2,\mathtt{false}){\mapsto}(3,\mathtt{false}), (2,\mathtt{true}){\mapsto}(3,\mathtt{false}), \\ \ldots \end{array}\right\}$$

Using this, we can now rephrase the satisfaction condition as:

Invariant sentences: $\langle A, R\rangle \models_{\mathcal{EVT}} \langle inv, \phi\rangle$ if and only if for each event name in the signature Σ

$$A^R \quad \models_{\mathcal{FOPEQ}} \quad (\forall \overline{x}, \overline{x}' \cdot r_e(\overline{x}, \overline{x}') \Rightarrow (\phi(\overline{x}) \wedge \phi(\overline{x}')))$$

Event sentences: For any event name e, we have $\langle A, R\rangle \models_{\mathcal{EVT}} \langle e, \phi\rangle$ if and only if

$$A^R \quad \models_{\mathcal{FOPEQ}} \quad (\forall \overline{x}, \overline{x}' \cdot r_e(\overline{x}, \overline{x}') \Rightarrow \phi(\overline{x}, \overline{x}'))$$

Theorem 1. *Given signatures Σ_1 and Σ_2, a signature morphism $\sigma : \Sigma_1 \to \Sigma_2$, a Σ_2-model M_2 and a Σ_1-sentence ψ, the following satisfaction condition holds:*

$$Mod(\sigma)(M_2) \models_{\Sigma_1} \psi \quad \iff \quad M_2 \models_{\Sigma_2} Sen(\sigma)(\psi)$$

Proof. Since there are two kinds of sentences over $\mathcal{EVT}$, invariant sentences and event sentences, we must prove that the satisfaction condition holds in both cases:

Invariant sentences: $Mod(\sigma)(M_2) \models_{\Sigma_1} \psi \iff M_2 \models_{\Sigma_2} Sen(\sigma)(\psi)$

Proof. If

$$\psi = \langle inv, \phi\rangle$$

then $Mod(\sigma)(M_2) \models_{\Sigma_1} \langle inv, \phi\rangle$ means that for all $e \in E$,

$$A_2^R|_\sigma \models_{\mathcal{FOPEQ}} (\forall \overline{x}, \overline{x}' \cdot r_e(\overline{x}, \overline{x}') \Rightarrow (\phi(\overline{x}) \wedge \phi(\overline{x}'))$$

and $M_2 \models_{\Sigma_2} Sen(\sigma)(\langle inv, \phi\rangle)$ means that for all $e \in E$,

$$A_2^R \models_{\mathcal{FOPEQ}} Sen(\sigma)(\forall \overline{x}, \overline{x}' \cdot r_e(\overline{x}, \overline{x}') \Rightarrow (\phi(\overline{x}) \wedge \phi(\overline{x}')))$$

Since bound variables are unchanged under $\mathcal{FOPEQ}$ signature morphisms,

$$Sen(\sigma)(\forall \overline{x}, \overline{x}' \cdot r_e(\overline{x}, \overline{x}') \Rightarrow (\phi(\overline{x}) \wedge \phi(\overline{x}'))) = (\forall \overline{x}, \overline{x}' \cdot r_{\sigma(e)}(\overline{x}, \overline{x}') \Rightarrow (\sigma(\phi)(\overline{x}) \wedge \sigma(\phi)(\overline{x}')))$$

Thus, the proof goal now becomes

$$A_2^R|_\sigma \models_{\mathcal{FOPEQ}} (\forall \overline{x}, \overline{x}' \cdot r_e(\overline{x}, \overline{x}') \Rightarrow \phi(\overline{x}, \overline{x}'))$$
$$\iff A_2^R \models_{\mathcal{FOPEQ}} Sen(\sigma)(\forall \overline{x}, \overline{x}' \cdot r_e(\overline{x}, \overline{x}') \Rightarrow \phi(\overline{x}, \overline{x}'))$$

The only difference between the left and right sides of $\iff$ are the names used for sorts, operations, predicates, events and variables which does not affect satisfaction. Moreover, we have phrased satisfaction over $\mathcal{EVT}$ as satisfaction over $\mathcal{FOPEQ}$ which holds. □

Event sentences: $M_2 \models_{\Sigma_2} Sen(\sigma)(\psi) \iff Mod(\sigma)(M_2) \models_{\Sigma_1} \psi$

Proof. In a similar way to that of invariant sentences, we can rephrase the satisfaction condition for event sentences of the form $\psi = \langle e, \phi \rangle$ as

$$\begin{aligned} & A_2^R \models_{\mathcal{FOPEQ}} Sen(\sigma)(\forall \overline{x}, \overline{x}' \cdot r_e(\overline{x}, \overline{x}') \Rightarrow \phi(\overline{x}, \overline{x}')) \\ \iff & A_2^R|_\sigma \models_{\mathcal{FOPEQ}} (\forall \overline{x}, \overline{x}' \cdot r_e(\overline{x}, \overline{x}') \Rightarrow \phi(\overline{x}, \overline{x}')) \end{aligned}$$

where

$$Sen(\sigma)(\forall \overline{x}, \overline{x}' \cdot r_e(\overline{x}, \overline{x}') \Rightarrow \phi(\overline{x}, \overline{x}')) = (\forall \overline{x}, \overline{x}' \cdot r_{\sigma(e)}(\overline{x}, \overline{x}') \Rightarrow \sigma(\phi)(\overline{x}, \overline{x}'))$$

As above, the only difference between the left and right sides are the names used for sorts, operations, predicates, events and variables. The validity of this sentence then derives from the validitiy of the satisfaction relation in $\mathcal{FOPEQ}$. □

5 Modularisation: amalgamation in $\mathcal{EVT}$

Pushouts and amalgamation are required for an institution to have good modularity properties with respect to the specification building operators [2]. In fact, amalgamation properties are a required for good parametrisation behaviour [3]. In this section we prove that $\mathcal{EVT}$ has pushouts and the amalgamation property.

An institution has the amalgamation property if all pushouts in **Sign** exist and every pushout diagram in **Sign** admits amalgamation [3].

We split the proof of amalgamation into two lemmas as follows:

Lemma 6. *Pushouts exist in **Sign***

Proof. Given two signature morphisms $\sigma_1 : \Sigma \to \Sigma_1$ and $\sigma_2 : \Sigma \to \Sigma_2$ a pushout is a triple $(\Sigma', \sigma'_1, \sigma'_2)$ with $\sigma'_1 \circ \sigma_1 = \sigma'_2 \circ \sigma_2$ that satisfies the universal property: for every other such triple $(\Sigma'', \sigma''_1, \sigma''_2)$ with $\sigma''_1 \circ \sigma_1 = \sigma''_2 \circ \sigma_2$ there exists a unique morphism $u : \Sigma' \to \Sigma''$ such that the diagram commutes:

$$\begin{tikzcd}
 & \Sigma \arrow[dl, "\sigma_1"'] \arrow[dr, "\sigma_2"] & \\
\Sigma_1 \arrow[dr, "\sigma'_1"] \arrow[ddr, "\sigma''_1"'] & & \Sigma_2 \arrow[dl, "\sigma'_2"'] \arrow[ddl, "\sigma''_2"] \\
 & \Sigma' \arrow[d, dashed, "u"] & \\
 & \Sigma'' &
\end{tikzcd}$$

Given some signature $\Sigma = \langle S, \Omega, \Pi, E, V \rangle$, and signature morphisms $\sigma_1 : \Sigma \rightarrow \Sigma_1, \sigma_2 : \Sigma \rightarrow \Sigma_2$ we will construct the pushout $(\Sigma', \sigma'_1, \sigma'_2)$. Since $\mathcal{FOPEQ}$ admits amalgamation and is semi-exact, all pushouts exist in $\mathbf{Sign}_{\mathcal{FOPEQ}}$ and the **Mod** functor maps them to pullbacks in **Cat** [4]. Hence, our pushout construction follows $\mathcal{FOPEQ}$ for the elements that $\mathcal{FOPEQ}$ has in common with $\mathcal{EVT}$. In $\mathbf{Sig}_{\mathcal{EVT}}$ the only additional elements are E and V, and these are described below:

- *Set of event names E:*
  Since the event names have no other dependencies we can construct $E' = (E_1 + E_2)/\sim$ where $\sim$ is the equivalence relation over E' such that, for any $e \in E$ the following diagram commutes:

$$\begin{tikzcd}
 & e \in E \arrow[dl, "\sigma_1"'] \arrow[dr, "\sigma_2"] & \\
\sigma_1(e) \in E_1 \arrow[dr, "\sigma'_1"'] & & \sigma_2(e) \in E_2 \arrow[dl, "\sigma'_2"] \\
 & e' \in E' &
\end{tikzcd}$$

  Here e' is the (representative of the) equivalence class in E'. Since signature morphisms map initialisation event names to initialisation event names the pushout does likewise.

- *Set of variable names V:* In a similar manner we can construct $V' = (V_1 + V_2)/\sim$, where here $\sim$ is the equivalence relation over V' such that, for any $v \in V$ we have:

$$\begin{tikzcd}
 & v : s \in V \arrow[dl, "\sigma_1"'] \arrow[dr, "\sigma_2"] & \\
\sigma_1(v) : \sigma_1(s) \in V_1 \arrow[dr, "\sigma'_1"'] & & \sigma_2(v) : \sigma_2(s) \in V_2 \arrow[dl, "\sigma'_2"] \\
 & v' : s' \in V' &
\end{tikzcd}$$

  Here v' is the (represenataive of the) equivalence class in the set V', and s' is the corresponding sort obtained form the pushout for sorts in S as defined in $\mathcal{FOPEQ}$.

Universality property for the pushout: Given any other triple $(\Sigma'', \sigma''_1, \sigma''_2)$ we can construct the unique morphism $u : \Sigma' \rightarrow \Sigma''$ such that the following diagrams commute for event names and variable names respectively:

$$\begin{tikzcd}
 & E \arrow[dl, "\sigma_1"'] \arrow[dr, "\sigma_2"] & \\
E_1 \arrow[dr, "\sigma_1'"] \arrow[ddr, "\sigma_1''"'] & & E_2 \arrow[dl, "\sigma_2'"'] \arrow[ddl, "\sigma_2''"] \\
 & E' \arrow[d, dashed, "u"] & \\
 & E'' &
\end{tikzcd}
\qquad
\begin{tikzcd}
 & V \arrow[dl, "\sigma_1"'] \arrow[dr, "\sigma_2"] & \\
V_1 \arrow[dr, "\sigma_1'"] \arrow[ddr, "\sigma_1''"'] & & V_2 \arrow[dl, "\sigma_2'"'] \arrow[ddl, "\sigma_2''"] \\
 & V' \arrow[d, dashed, "u"] & \\
 & V'' &
\end{tikzcd}$$

The S, Ω and Π components of u follow from $\mathcal{FOPEQ}$, and the unique mapping for elements of E' and V' follow the corresponding construction for sets and sorts.

□

Lemma 7. *Every pushout diagram in **Sign** admits amalgamation*

Consider the following diagram in **Sign**

$$\begin{tikzcd}
 & \Sigma' & \\
\Sigma_1 \arrow[ur, "\sigma_1'"] & & \Sigma_2 \arrow[ul, "\sigma_2'"'] \\
 & \Sigma \arrow[ul, "\sigma_1"] \arrow[ur, "\sigma_2"'] &
\end{tikzcd}$$

This diagram admits amalgamation if:

(a) for $M_1 \in |\mathbf{Mod}(\Sigma_1)|$ and $M_2 \in |\mathbf{Mod}(\Sigma_2)|$ such that $M_1|_{\sigma_1} = M_2|_{\sigma_2}$, there exists a unique model (the amalgamation of M_1 and M_2) $M' \in |\mathbf{Mod}(\Sigma')|$ such that $M'|_{\sigma_1'} = M_1$ and $M'|_{\sigma_2'} = M_2$.

(b) for any two model morphisms $f_1 : M_{11} \to M_{12}$ in $\mathbf{Mod}(\Sigma_1)$ and $f_2 : M_{21} \to M_{22}$ in $\mathbf{Mod}(\Sigma_2)$ such that $f_1|_{\sigma_1} = f_2|_{\sigma_2}$, there exists a unique model morphism (the amalgamation of f_1 and f_2) $f' : M_1' \to M_2'$ in $\mathbf{Mod}(\Sigma')$ such that $f'|_{\sigma_1'} = f_1$ and $f'|_{\sigma_2'} = f_2$.

We handle both of these conditions separately by splitting this lemma into two sublemmas:

Lemma 7(a) *For $M_1 \in |\boldsymbol{Mod}(\Sigma_1)|$ and $M_2 \in |\boldsymbol{Mod}(\Sigma_2)|$ such that $M_1|_{\sigma_1} = M_2|_{\sigma_2}$, there exists a unique model (the amalgamation of M_1 and M_2) $M' \in |\boldsymbol{Mod}(\Sigma')|$ such that $M'|_{\sigma_1'} = M_1$ and $M'|_{\sigma_2'} = M_2$.*

Proof. Consider the commutative diagram below

$$\begin{tikzcd}
 & M' = \langle A', R' \rangle \arrow[dl, "Mod(\sigma_1')"'] \arrow[dr, "Mod(\sigma_2')"] & \\
M_1 = \langle A_1, R_1 \rangle \arrow[dr, "Mod(\sigma_1)"'] & & M_2 = \langle A_2, R_2 \rangle \arrow[dl, "Mod(\sigma_2)"] \\
 & M = \langle A, R \rangle &
\end{tikzcd}$$

with signature morphisms $\sigma_1, \sigma_2, \sigma_1'$ and σ_2'.

We define $M = M_1|_{\sigma_1} = M_2|_{\sigma\text{„}}$ where $A = A_1|_{\sigma_1} = A_2|_{\sigma_2}$ is as expected in $\mathcal{FOPEQ}$ and $R = R_1|_{\sigma_1} = R_2|_{\sigma_2}$.

The task at hand is that of constructing M' which is composed of A' and R'. A' is the unique $\mathcal{FOPEQ}$-model (amalgamation of A_1 and A_2) over $\mathcal{FOPEQ}$. We construct the unique relation R' which is the amalgamation of R_1 and R_2 by pushing the relevant mappings from $\mathcal{FOPEQ}$ through to the indexed value-pairs of the model elements in $\mathcal{EVT}$.

Specifically, starting from any value-pair $\langle d_{b_0}, d_{a_0} \rangle_{e,v} \in R$ we construct the corresponding value-pair in R' so that the following diagram commutes:

$$\begin{tikzcd}
 & \langle h'(d_{b_0}), h'(d_{a_0}) \rangle_{\sigma'(e), \sigma'(v)} \in R' \arrow[dl, "Mod(\sigma_1')"'] \arrow[dr, "Mod(\sigma_2')"] & \\
= \langle h_1(d_{b_0}), h_1(d_{a_0}) \rangle_{\sigma_1(e), \sigma_1(v)} \in R_1 \arrow[dr, "Mod(\sigma_1)"'] & & \langle h_2(d_{b_0}), h_2(d_{a_0}) \rangle_{\sigma_2(e), \sigma_2(v)} \in R_2 \arrow[dl, "Mod(\sigma_2)"] \\
 & \langle d_{b_0}, d_{a_0} \rangle_{e,v} \in R &
\end{tikzcd}$$

Here $h' = (h_1 + h_2)$, the corresponding function over the carrier-sets in M' obtained from $\mathcal{FOPEQ}$, and $\sigma' = (\sigma_1' \circ \sigma_1) + (\sigma_2' \circ \sigma_2)$ the mapping for variable and event names obtained from the corresponding construction in **Sign**.

□

Lemma 7(b) *For any two model morphisms $f_1 : M_{11} \to M_{12}$ in **Mod**(Σ_1) and $f_2 : M_{21} \to M_{22}$ in **Mod**(Σ_2) such that $f_1|_{\sigma_1} = f_2|_{\sigma_2}$, there exists a unique model morphism (the amalgamation of f_1 and f_2) called $f' : M_1' \to M_2'$ in **Mod**(Σ'), such that $f'|_{\sigma_1'} = f_1$ and $f'|_{\sigma_2'} = f_2$.*

Proof. Here, we are given the morphisms f_1 and f_2 and their common reduct f_0, and must construct f' so that the following diagram commutes:

$$\begin{tikzcd}
& f' : M'_1 \to M'_2 \arrow[dl, "Mod(\sigma'_1)"'] \arrow[dr, "Mod(\sigma'_2)"] & \\
f_1 : M_{11} \to M_{12} \arrow[dr, "Mod(\sigma_1)"'] & & f_2 : M_{21} \to M_{22} \arrow[dl, "Mod(\sigma_2)"] \\
& f_0 : M_{01} \to M_{02} &
\end{tikzcd}$$

Since each $\mathcal{EVT}$-model has a $\mathcal{FOPEQ}$ model as its first component, each of the $\mathcal{EVT}$-model morphisms f_0, f_1, f_2 and f' must have an underlying model morphism in $\mathcal{FOPEQ}$, which we denote f_0^-, f_1^-, f_2^- and $f^{'-}$ respectively. To build the amalgamation for $\mathcal{EVT}$-models we must show how to extend these to cover the relations that are the second component of these models.

Suppose we start with any f_0-maplet of the form

$$\langle d_{b_0}, d_{a_0} \rangle_{e_0, v_0} \mapsto \langle f_0^-(d_{b_0}), f_0^-(d_{a_0}) \rangle_{e_0, v_0} \ \in \ f_0$$

where f_0^- is the underlying map on data types from the $\mathcal{FOPEQ}$ model morphism.

Then the original two functions in f_1 and f_2 must have maplets of the form

$$\langle h_1(d_{b_0}), h_1(d_{a_0}) \rangle_{\sigma_1(e_0), \sigma_1(v_0)} \mapsto \langle f_1^-(h_1(d_{b_0})), f_1^-(h_1(d_{a_0})) \rangle_{\sigma_1(e_0), \sigma_1(v_0)} \ \in \ f_1$$

and

$$\langle h_2(d_{b_0}), h_2(d_{a_0}) \rangle_{\sigma_2(e_0), \sigma_2(v_0)} \mapsto \langle f_2^-(h_2(d_{b_0})), f_2^-(h_2(d_{a_0})) \rangle_{\sigma_2(e_0), \sigma_2(v_0)} \ \in \ f_2$$

where f_1^- and f_2^- are again the data type maps from the underlying $\mathcal{FOPEQ}$ model morphism, and h_1 and h_2 are obtained from $Mod(\sigma_1)$ and $Mod(\sigma_2)$.

We then can construct the elements of the unique model morphism f', which is the amalgamation of f_1 and f_2, as f'-maplets of the form:

$$\begin{aligned} &\langle h'(d_{b_0}), h'(d_{a_0}) \rangle_{\sigma'(e_0), \sigma'(v_0)} \\ &\qquad\qquad \mapsto \langle f^{'-}(h'(d_{b_0})), f^{'-}(h'(d_{a_0})) \rangle_{\sigma'(e_0), \sigma'(v_0)} \ \in \ f' \end{aligned}$$

As before, $h' = (h_1 + h_2)$, the corresponding function over the carrier-sets in M' obtained from $\mathcal{FOPEQ}$, and $\sigma' = (\sigma'_1 \circ \sigma_1) + (\sigma'_2 \circ \sigma_2)$ the mapping for variable and event names obtained from the corresponding construction in **Sign**. Here $f^{'-} = f_1^- + f_2^-$ is the amalgamation from the corresponding diagram for model morphisms in $\mathcal{FOPEQ}$, which ensures that the data values are mapped to corresponding values in the model M'_2.

□

References

1. J. A. Goguen and R. M. Burstall. Institutions: abstract model theory for specification and programming. *Journal of the ACM*, 39(1):95–146, 1992.
2. T. Mossakowski and M. Roggenbach. Structured CSP - a process algebra as an institution. In *Recent Trends in Algebraic Development Techniques*, volume 4409 of *LNCS*, pages 92–110. 2007.
3. D. Sanella and A. Tarlecki. *Foundations of Algebraic Specification and Formal Software Development*. Springer, 2012.
4. D. Sanella and A. Tarlecki. Property oriented semantics of structured specifications. *Mathematical Structures in Computer Science*, 2013.