

An Institution for UML 2.0 Interactions

María Victoria Cengarle¹ and Alexander Knapp²

¹ Technische Universität München
cengarle@in.tum.de

² Universität Augsburg
knapp@informatik.uni-augsburg.de

Abstract. This work presents the theory of UML 2.0 interactions, that is proven to define an institution.

1 Introduction

The present work is devoted to the language of UML 2.0 interactions. We develop an abstract syntax, a formal semantics based on previous works [1,2], a first-order extension of the language, and the corresponding extension of the formal semantics. We discuss some alternatives for the semantics, in particular in the light of institutions [3].

Preliminaries

A class hierarchy \mathbf{C} is a pair (C, \leq_C) with C a set of class names and $\leq_C \subseteq C \times C$ an inheritance relation. An object domain \mathbf{O} for \mathbf{C} is a family $(O_c)_{c \in C}$ of sets of object identifiers verifying $O_{c_1} \subseteq O_{c_2}$ if $c_1 \leq_C c_2$. A class hierarchy $\mathbf{C}' = (C', \leq_{C'})$ is called a subhierarchy of a class hierarchy $\mathbf{C} = (C, \leq_C)$ if $C \subseteq C'$ and $\leq_{C'} = \leq_C|_{C'}$, i.e. $\{(c'_1, c'_2) \in \leq_C \mid c'_1, c'_2 \in C'\}$. In this case, we write $\mathbf{C}' \subseteq \mathbf{C}$.

2 Atomic interactions

2.1 Syntax

An interaction signature, or IL-signature for short, is a pair $\Sigma = (\mathbf{C}, M)$, where \mathbf{C} is a class hierarchy and M a set of messages. If $c \in C$, by abuse of notation we may write $c \in \Sigma$. Likewise, if $m \in M$, we may write $m \in \Sigma$. An IL-signature $\Sigma' = (\mathbf{C}', M')$ is a subsignature of $\Sigma = (\mathbf{C}, M)$, denoted by $\Sigma' \subseteq \Sigma$, if $\mathbf{C}' \subseteq \mathbf{C}$ and $M' \subseteq M$.

Given a signature $\Sigma = (\mathbf{C}, M)$, with $\mathbf{C} = (C, \leq_C)$, and variables $X = (X^c)_{c \in C}$, we define the set T of atomic formulas over Σ and X by

$$\begin{aligned} Q &::= \text{skip} \mid \text{snd}(x^c, y^d, m) \mid \text{rcv}(x^c, y^d, m) \\ T &::= Q \\ &\quad \mid \text{strict}(T, T) \mid \text{seq}(T, T) \mid \text{par}(T, T) \mid \text{alt}(T, T) \\ &\quad \mid \text{loop}(T) \mid \text{neg}(T) \mid \text{assert}(\Sigma', T) \end{aligned}$$

where $c, d \in C$, $x^c \in X^c$, $y^d \in X^d$, $m \in M$, and $\Sigma' \subseteq \Sigma$.

Given IL-signatures $\Sigma_1 = (\mathbf{C}_1, M_1)$ and $\Sigma_2 = (\mathbf{C}_2, M_2)$, an IL-signature morphism $\sigma : \Sigma_1 \rightarrow \Sigma_2$ monotonically maps class names to class names, and maps messages to messages. Given a set of variables $X_2 = (X_2^{c_2})_{c_2 \in \mathbf{C}_2}$, we define a set $X_2|_\sigma = X_1 = (X_1^{c_1})_{c_1 \in \mathbf{C}_1}$ by $X_1^{c_1} = X_2^{\sigma(c_1)}$. Signature morphisms extend to atomic formulas over Σ_1 and $X_2|_\sigma$ as follows:

$$\begin{aligned}
\sigma(\text{skip}) &= \text{skip} \\
\sigma(\text{snd}(x^{c_1}, y^{d_1}, m_1)) &= \text{snd}(x^{\sigma(c_1)}, y^{\sigma(d_1)}, \sigma(m_1)) \\
\sigma(\text{rcv}(x^{c_1}, y^{d_1}, m_1)) &= \text{rcv}(x^{\sigma(c_1)}, y^{\sigma(d_1)}, \sigma(m_1)) \\
\sigma(\text{strict}(T_1^1, T_1^2)) &= \text{strict}(\sigma(T_1^1), \sigma(T_1^2)) \\
\sigma(\text{seq}(T_1^1, T_1^2)) &= \text{seq}(\sigma(T_1^1), \sigma(T_1^2)) \\
\sigma(\text{par}(T_1^1, T_1^2)) &= \text{par}(\sigma(T_1^1), \sigma(T_1^2)) \\
\sigma(\text{alt}(T_1^1, T_1^2)) &= \text{alt}(\sigma(T_1^1), \sigma(T_1^2)) \\
\sigma(\text{loop}(T_1^1)) &= \text{loop}(\sigma(T_1^1)) \\
\sigma(\text{neg}(T_1^1)) &= \text{neg}(\sigma(T_1^1)) \\
\sigma(\text{assert}(\Sigma_1', T_1^1)) &= \text{assert}(\sigma(\Sigma_1'), \sigma(T_1^1))
\end{aligned}$$

2.2 Semantics

Given an IL-signature $\Sigma = (\mathbf{C}, M)$ with $\mathbf{C} = (C, \leq_C)$, a *structure* \mathcal{I} for Σ is a triple $\mathcal{I} = (\mathbf{O}, \mathcal{M}, \mu)$ where $\mathbf{O} = (O^c)_{c \in C}$ is an object domain for \mathbf{C} , \mathcal{M} a domain of actual messages, and $\mu : M \rightarrow \mathcal{M}$ is an interpretation function for messages.³ By abuse of notation, if $o \in O^c$ for a $c \in C$, we write $o \in \mathcal{I}$; likewise, for a message instance $m \in \mathcal{M}$ we may write $m \in \mathcal{I}$.

Given moreover variables $X = (X^c)_{c \in C}$, a *valuation* $\beta = (\beta^c)_{c \in C}$ for X in \mathcal{I} assigns values to variables, i.e., $\beta^c : X^c \rightarrow O^c$ for every $c \in C$. When clear from the context we may write simply β instead of β^c .

A subsignature $\Sigma' = (\mathbf{C}', M') \subseteq \Sigma$ with $\mathbf{C}' = (C', \leq_{C'})$ induces a set of *reachable traces* $\mathcal{T}(\Sigma', \mathcal{I})$ defined by

$$\mathcal{T}(\Sigma', \mathcal{I}) = \{ e_1 \cdot e_2 \cdots e_n \mid \forall i \in \{1, \dots, n\}. \exists o, o' \in \bigcup_{c \in C'} O^c. \exists m \in \mu(M'). \\ e_i = \text{snd}(o, o', m) \vee e_i = \text{rcv}(o, o', m) \}$$

(Notice that Σ' might cover Σ entirely.) The set $\mathbb{T}(\mathcal{I})$ of all traces of the structure \mathcal{I} is defined by

$$\mathbb{T}(\mathcal{I}) = \{ e_1 \cdot e_2 \cdots e_n \mid \forall i \in \{1, \dots, n\}. \exists o, o', m \in \mathcal{I}. \\ e_i = \text{snd}(o, o', m) \vee e_i = \text{rcv}(o, o', m) \}$$

Obviously, $\mathcal{T}(\Sigma, \mathcal{I}) \subseteq \mathbb{T}(\mathcal{I})$. The equality does not hold in general: it fails when not every message instance of the structure can be denoted by the signature.

The sets $\mathcal{P}_{\mathcal{I}}(T, \beta)$ and $\mathcal{N}_{\mathcal{I}}(T, \beta)$ of *positive* resp. *negative* traces of an atomic formula T over Σ and X in the structure \mathcal{I} under valuation β are inductively defined as given in Tab. 1. It is easy to check that $\mathcal{P}_{\mathcal{I}}(T, \beta) \subseteq \mathcal{T}(\Sigma, \mathcal{I})$ and $\mathcal{N}_{\mathcal{I}}(T, \beta) \subseteq \mathcal{T}(\Sigma, \mathcal{I})$.

³ Messages are treated almost as were they variables. Alternatively, they might have a structure, that need be reflected by the interpretation function μ . These matters, for the purposes of the present work, are irrelevant.

T	$\mathcal{P}_{\mathcal{I}}(T, \beta)$		$\mathcal{N}_{\mathcal{I}}(T, \beta)$	
skip	$\{\varepsilon\}$		\emptyset	
$\text{snd}(x^c, y^d, m)$	$\{\text{snd}(\beta(x^c), \beta(y^d), \mu(m))\}$		\emptyset	
$\text{rcv}(x^c, y^d, m)$	$\{\text{rcv}(\beta(x^c), \beta(y^d), \mu(m))\}$		\emptyset	
$\text{strict}(T_1, T_2)$	$\mathcal{P}_{\mathcal{I}}(T_1, \beta); \mathcal{P}_{\mathcal{I}}(T_2, \beta)$		$(\mathcal{P}_{\mathcal{I}}(T_1, \beta); \mathcal{N}_{\mathcal{I}}(T_2, \beta)) \cup$ $(\mathcal{N}_{\mathcal{I}}(T_1, \beta); \mathcal{P}_{\mathcal{I}}(T_2, \beta)) \cup$ $(\mathcal{N}_{\mathcal{I}}(T_1, \beta); \mathcal{N}_{\mathcal{I}}(T_2, \beta))$	^a
$\text{seq}(T_1, T_2)$	$\mathcal{P}_{\mathcal{I}}(T_1, \beta);_{\infty} \mathcal{P}_{\mathcal{I}}(T_2, \beta)$	^b	$(\mathcal{P}_{\mathcal{I}}(T_1, \beta);_{\infty} \mathcal{N}_{\mathcal{I}}(T_2, \beta)) \cup$ $(\mathcal{N}_{\mathcal{I}}(T_1, \beta);_{\infty} \mathcal{P}_{\mathcal{I}}(T_2, \beta)) \cup$ $(\mathcal{N}_{\mathcal{I}}(T_1, \beta);_{\infty} \mathcal{N}_{\mathcal{I}}(T_2, \beta))$	^{a b}
$\text{par}(T_1, T_2)$	$\mathcal{P}_{\mathcal{I}}(T_1, \beta) \parallel \mathcal{P}_{\mathcal{I}}(T_2, \beta)$	^c	$(\mathcal{P}_{\mathcal{I}}(T_1, \beta) \parallel \mathcal{N}_{\mathcal{I}}(T_2, \beta)) \cup$ $(\mathcal{N}_{\mathcal{I}}(T_1, \beta) \parallel \mathcal{P}_{\mathcal{I}}(T_2, \beta)) \cup$ $(\mathcal{N}_{\mathcal{I}}(T_1, \beta) \parallel \mathcal{N}_{\mathcal{I}}(T_2, \beta))$	^c
$\text{alt}(T_1, T_2)$	$\mathcal{P}_{\mathcal{I}}(T_1, \beta) \cup \mathcal{P}_{\mathcal{I}}(T_2, \beta)$		$\mathcal{N}_{\mathcal{I}}(T_1, \beta) \cup \mathcal{N}_{\mathcal{I}}(T_2, \beta)$	^a
$\text{loop}(T')$	$\bigcup_{n \in \mathbb{N}} \mathcal{P}_{\mathcal{I}}(T', \beta)^{;n}_{\infty}$	^d	$\bigcup_{n \in \mathbb{N}} (\mathcal{P}_{\mathcal{I}}(T', \beta) \cup \mathcal{N}_{\mathcal{I}}(T', \beta))^{;n}_{\infty}$ $\setminus \mathcal{P}_{\mathcal{I}}(T', \beta)^{;n}_{\infty}$	^e
$\text{neg}(T')$	$\{\varepsilon\}$		$\mathcal{P}_{\mathcal{I}}(T', \beta) \cup \mathcal{N}_{\mathcal{I}}(T', \beta)$	^a
$\text{assert}(\Sigma', T')$	$\mathcal{P}_{\mathcal{I}}(T', \beta)$		$\mathcal{T}(\Sigma', \mathcal{I}) \setminus \mathcal{P}_{\mathcal{I}}(T', \beta)$	^f

^a The set of negative traces associated with strict sequential composition, weak sequential composition, disjunction, and negation of interaction(s), respectively, differs from our original definition in [1] and is aligned with the one by [2].

^b Weak sequential composition of two traces t_1 and t_2 , denoted by $t_1;_{\infty} t_2$ (see [1]), defines not necessarily a single trace but possibly a set of traces. For sets of traces \mathcal{T}_1 and \mathcal{T}_2 , their weak sequential composition $\mathcal{T}_1;_{\infty} \mathcal{T}_2$ is extensionally defined by $\bigcup_{t_1 \in \mathcal{T}_1, t_2 \in \mathcal{T}_2} t_1;_{\infty} t_2$.

^c Parallel composition of two traces t_1 and t_2 , denoted by $t_1 \parallel t_2$ (see [1]), defines not necessarily a single trace but possibly a set of traces. For sets of traces \mathcal{T}_1 and \mathcal{T}_2 , their parallel composition $\mathcal{T}_1 \parallel \mathcal{T}_2$ is extensionally defined by $\bigcup_{t_1 \in \mathcal{T}_1, t_2 \in \mathcal{T}_2} t_1 \parallel t_2$.

^d Given a set of traces \mathcal{T} , its n -th weak sequential composition $\mathcal{T};_{\infty}^n$ is inductively defined by $\mathcal{T};_{\infty}^0 \stackrel{\text{def}}{=} \{\varepsilon\}$ and $\mathcal{T};_{\infty}^{n+1} \stackrel{\text{def}}{=} \mathcal{T};_{\infty} \mathcal{T};_{\infty}^n$. The set of positive traces of $\text{loop}(T')$ in \mathcal{I} according to β is equivalent to $\bigcup_{n \in \mathbb{N}} \mathcal{P}_{\mathcal{I}}(\text{seq}^n(T'), \beta)$ where by $\text{seq}^n(T)$ we abbreviate the term defined by $\text{seq}^0(T) \stackrel{\text{def}}{=} \text{skip}$ and $\text{seq}^{n+1}(T) \stackrel{\text{def}}{=} \text{seq}(T, \text{seq}^n(T))$.

^e Similarly, the set of negative traces, in \mathcal{I} according to β , of $\text{loop}(T')$ is equivalent to $\bigcup_{n \in \mathbb{N}} \mathcal{N}_{\mathcal{I}}(\text{seq}^n(T'), \beta)$.

^f The assertion needs to be made with respect to a set of lifelines or class names as well as with respect to a set of messages, lifelines and messages gathered in a subsignature, in order to ensure the invariance of the semantics under change of notation; see Sect. 2.3.

Table 1. Semantics of propositional formulas

Given IL-signatures $\Sigma_1 = (\mathbf{C}_1, M_1)$ and $\Sigma_2 = (\mathbf{C}_2, M_2)$ with $\mathbf{C}_i = (C_i, \leq_{C_i})$ ($i = 1, 2$), given an IL-signature morphism $\sigma : \Sigma_1 \rightarrow \Sigma_2$, and given a Σ_2 -structure $\mathcal{I}_2 = (\mathbf{O}_2, \mathcal{M}_2, \mu_2)$, the *reduct* of \mathcal{I}_2 along σ is the Σ_1 -structure $\mathcal{I}_2|_\sigma = (\mathbf{O}_2|_\sigma, \mathcal{M}_2, \mu_2|_\sigma)$ whose components are defined as follows:

$$\begin{aligned} \mathbf{O}_2|_\sigma &= ((O_2|_\sigma)^c)_{c \in C_1} \text{ with } (O_2|_\sigma)^c \stackrel{\text{def}}{=} O_2^{\sigma(c)} \text{ for each } c \in C_1 \\ \mu_2|_\sigma(m) &\stackrel{\text{def}}{=} \mu_2(\sigma(m)) \text{ for each } m \in M_1 \end{aligned}$$

Moreover, given variables $X_2 = (X_2^c)_{c \in C_2}$ and given a valuation β_2 for X_2 in \mathcal{I}_2 , the *reduct* of β_2 along σ is a valuation for $X_2|_\sigma$ (see Sect. 2.1) in $\mathcal{I}_2|_\sigma$ defined by $(\beta_2|_\sigma)^c(x^c) \stackrel{\text{def}}{=} \beta_2^{\sigma(c)}(x^{\sigma(c)})$.

2.3 Semantic invariance under change of notation

Let $\Sigma_1 = (\mathbf{C}_1, M_1)$ and $\Sigma_2 = (\mathbf{C}_2, M_2)$ be IL-signatures, let $\sigma : \Sigma_1 \rightarrow \Sigma_2$ be an IL-signature morphism. Let $X_2 = (X_2^{c_2})_{c_2 \in C_2}$ be a set of variables and $X_1 = X_2|_\sigma$. Let $\mathcal{I}_2 = (\mathbf{O}_2, \mathcal{M}_2, \mu_2)$ be a Σ_2 -structure and $\mathcal{I}_1 = \mathcal{I}_2|_\sigma$ with $\mu_1 = \mu_2|_\sigma$. Let β_2 be a valuation for X_2 in \mathcal{I}_2 and $\beta_1 = \beta_2|_\sigma$.

Semantic invariance under change of notation is formulated as $\mathcal{P}_{\mathcal{I}_2}(\sigma(T_1), \beta_2) = \mathcal{P}_{\mathcal{I}_1}(T_1, \beta_1)$ and $\mathcal{N}_{\mathcal{I}_2}(\sigma(T_1), \beta_2) = \mathcal{N}_{\mathcal{I}_1}(T_1, \beta_1)$ for any atomic formula T_1 over Σ_1 and X_1 . This is shown by induction on the structure of T_1 .

The base cases are as follows:

$$\begin{aligned} \mathcal{P}_{\mathcal{I}_2}(\sigma(\text{skip}), \beta_2) &= \{\varepsilon\} = \mathcal{P}_{\mathcal{I}_1}(\text{skip}, \beta_1) \\ \mathcal{N}_{\mathcal{I}_2}(\sigma(\text{skip}), \beta_2) &= \emptyset = \mathcal{N}_{\mathcal{I}_1}(\text{skip}, \beta_1) \\ \mathcal{P}_{\mathcal{I}_2}(\sigma(\text{snd}(x^{c_1}, y^{d_1}, m_1)), \beta_2) &= \mathcal{P}_{\mathcal{I}_2}(\text{snd}(x^{\sigma(c_1)}, y^{\sigma(d_1)}, \sigma(m_1)), \beta_2) \\ &= \{\text{snd}(\beta_2(x^{\sigma(c_1)}), \beta_2(y^{\sigma(d_1)}), \mu_2(\sigma(m_1)))\} \\ &= \{\text{snd}(\beta_1(x^{c_1}), \beta_1(y^{d_1}), \mu_1(m_1))\} \\ &= \mathcal{P}_{\mathcal{I}_1}(\text{snd}(x^{c_1}, y^{d_1}, m_1), \beta_1) \\ \mathcal{N}_{\mathcal{I}_2}(\sigma(\text{snd}(x^{c_1}, y^{d_1}, m_1)), \beta_2) &= \emptyset = \mathcal{N}_{\mathcal{I}_1}(\text{snd}(x^{c_1}, y^{d_1}, m_1), \beta_1) \\ &\text{(Analogously for } T_1 \text{ the formula } \text{rcv}(x^{c_1}, y^{d_1}, m_1)\text{.)} \end{aligned}$$

For any other case, the thesis obviously holds by the induction hypothesis and by definition of reduct. The only interesting case is given by the set of negative traces of an assertion. If $\Sigma'_1 = (\mathbf{C}'_1, M'_1) \subseteq \Sigma_1$ with $\mathbf{C}'_1 = (C'_1, \leq_{C'_1|_{C'_1}})$, then

$$\begin{aligned} \mathcal{N}_{\mathcal{I}_2}(\sigma(\text{assert}(\Sigma'_1, T_1)), \beta_2) &= \mathcal{N}_{\mathcal{I}_2}(\text{assert}(\sigma(\Sigma'_1), \sigma(T_1)), \beta_2) \\ &= \mathcal{T}(\sigma(\Sigma'_1), \mathcal{I}_2) \setminus \mathcal{P}_{\mathcal{I}_2}(\sigma(T_1), \beta_2) \\ &\stackrel{\spadesuit}{=} \mathcal{T}(\Sigma'_1, \mathcal{I}_1) \setminus \mathcal{P}_{\mathcal{I}_2}(\sigma(T_1), \beta_2) \\ &\stackrel{\text{IH}}{=} \mathcal{T}(\Sigma'_1, \mathcal{I}_1) \setminus \mathcal{P}_{\mathcal{I}_1}(T_1, \beta_1) \\ &= \mathcal{N}_{\mathcal{I}_1}(\text{assert}(\Sigma'_1, T_1), \beta_1) \end{aligned}$$

The equality marked with a ♠ deserves a little digression:

$$\begin{aligned}
& \mathcal{T}(\sigma(\Sigma'_1), \mathcal{I}_2) \\
&= \{ e_1 \cdot e_2 \cdots e_n \mid \forall i \in \{1, \dots, n\}. \exists o, o' \in \cup_{c \in \sigma(\Sigma'_1)} O_2^c. \exists m \in \sigma(\Sigma'_1). \\
&\quad e_i = \text{snd}(o, o', \mu_2(m)) \vee e_i = \text{rcv}(o, o', \mu_2(m)) \} \\
&= \{ e_1 \cdot e_2 \cdots e_n \mid \forall i \in \{1, \dots, n\}. \exists o, o' \in \cup_{c \in \Sigma'_1} O_2^{\sigma(c)}. \exists m \in \Sigma'_1. \\
&\quad e_i = \text{snd}(o, o', \mu_2(\sigma(m))) \vee e_i = \text{rcv}(o, o', \mu_2(\sigma(m))) \} \\
&= \{ e_1 \cdot e_2 \cdots e_n \mid \forall i \in \{1, \dots, n\}. \exists o, o' \in \cup_{c \in \Sigma'_1} (O_2|_\sigma)^c. \exists m \in \Sigma'_1. \\
&\quad e_i = \text{snd}(o, o', \mu_2|_\sigma(m)) \vee e_i = \text{rcv}(o, o', \mu_2|_\sigma(m)) \} \\
&= \mathcal{T}(\Sigma'_1, \mathcal{I}_2|_\sigma) \\
&= \mathcal{T}(\Sigma'_1, \mathcal{I}_1)
\end{aligned}$$

i.e., that equality holds by definition.

This equality also holds if unreachable messages are used, i.e., if any $m \in \mathcal{M}_2$ is used instead of only those message instances in the image of μ_2 . Moreover, unreachability of messages is *not* what makes the semantic invariance fail when considering the complement wrt. the set of all possible traces. If the definition of negative traces for assert were the complement of the positive ones with respect to the universe of *all* possible traces in the structure, then the semantic invariance would not hold. Indeed, in the alternative reasoning:

$$\begin{aligned}
\mathcal{N}_{\mathcal{I}_2}(\sigma(\text{assert}(\Sigma'_1, T_1)), \beta_2) &= \mathcal{N}_{\mathcal{I}_2}(\text{assert}(\sigma(\Sigma'_1), \sigma(T_1)), \beta_2) \\
&= \mathbb{T}(\mathcal{I}_2) \setminus \mathcal{P}_{\mathcal{I}_2}(\sigma(T_1), \beta_2) \\
&\stackrel{\clubsuit}{=} \mathbb{T}(\mathcal{I}_1) \setminus \mathcal{P}_{\mathcal{I}_2}(\sigma(T_1), \beta_2) \\
&\stackrel{\text{IH}}{=} \mathbb{T}(\mathcal{I}_1) \setminus \mathcal{P}_{\mathcal{I}_1}(T_1, \beta_2|_\sigma) \\
&= \mathcal{N}_{\mathcal{I}_1}(\text{assert}(\Sigma'_1, T_1), \beta_1)
\end{aligned}$$

exactly the equality marked with a ♣ does not hold in general, since

$$\begin{aligned}
\mathbb{T}(\mathcal{I}_1) &= \{ e_1 \cdot e_2 \cdots e_n \mid \forall i \in \{1, \dots, n\}. \exists o, o', m \in \mathcal{I}_1. \\
&\quad e_i = \text{snd}(o, o', m) \vee e_i = \text{rcv}(o, o', m) \} \\
&\subseteq \{ e_1 \cdot e_2 \cdots e_n \mid \forall i \in \{1, \dots, n\}. \exists o, o', m \in \mathcal{I}_2. \\
&\quad e_i = \text{snd}(o, o', m) \vee e_i = \text{rcv}(o, o', m) \} \\
&= \mathbb{T}(\mathcal{I}_2)
\end{aligned}$$

since $o \in \mathcal{I}_1 \not\stackrel{\rightarrow}{\Leftarrow} o \in \mathcal{I}_2$, i.e., unreachable types make the semantic invariance fail.

3 First-order interactions

In this section we introduce connectors for the definition of composed interactions, and show that the obtained language IL of UML Interactions is an institution.

3.1 Syntax

An IL-signature declares classes (or, by abuse of notation, types) that may be in an inheritance relation, and messages. Usually in a object-oriented setting, messages are

method calls or signals, and are (inductively) defined given declarations of (typed) methods and signals within classes.

Formally, and as in the atomic case, an IL-signature $\Sigma = (\mathbf{C}, M)$ declares a class hierarchy \mathbf{C} and a set M of *messages*. The collection of all IL-signatures is denoted by Sign_{IL} .

As discussed in Sect. 2.1, the messages could be typed; this matter, for the purposes of the present work, is irrelevant. The same with attributes that are possibly declared within a class, as well as inheritance and packaging mechanisms. By abuse of notation, we write $c \in \Sigma$ if c is a class name of the IL-signature Σ . We likewise write $m \in \Sigma$ if m is a message of the IL-signature Σ .

Let $\Sigma = (\mathbf{C}, M)$ be an IL-signature with $\mathbf{C} = (C, \leq_C)$, let $X = (X^c)_{c \in C}$ be a set of typed variables. The language of *propositional* (Σ, X) -formulas has the form

$$T ::= Q \mid Op(T, \dots, T)$$

where Q ranges over members of a family of atomic formulas (e.g., single events or pomsets over Σ and X), and Op over n -ary IL-operators (like strict, seq, par, etc., as in Sect. 2.1, for instance). The language of *first-order* (Σ, X) -formulas has the form

$$\psi ::= T \mid x_1^c = x_2^c \mid \neg\psi \mid \psi \wedge \psi \mid (\forall x^c)\psi$$

where $x^c, x_1^c, x_2^c \in X^c$ for $c \in \Sigma$.

Notice that propositional (Σ, X) -formulas may contain occurrences of variables in $(X^c)_{c \in C}$ with C the types provided by Σ .

The notion of closed formula is defined as usual; closed (Σ, X) -formulas are called Σ -sentences. Abbreviations like $\psi_1 \vee \psi_2$ and $\psi_1 \Rightarrow \psi_2$ are likewise defined as usual.

The same as in Sect. 2.1, given IL-signatures $\Sigma_1 = (\mathbf{C}_1, M_1)$ and $\Sigma_2 = (\mathbf{C}_2, M_2)$, an IL-signature morphism $\sigma : \Sigma_1 \rightarrow \Sigma_2$ is a pair of maps $\langle \sigma_C, \sigma_M \rangle$ between classes and messages, with σ_C monotonic. If the messages are typed, then the map σ_M must be type preserving, i.e., compatible with σ_C . IL-signature morphisms extend to formulas in a natural way.

IL-signatures and IL-signature morphisms define a finitely cocomplete category.

3.2 “Natural” semantics

Given an IL-signature Σ , the notion of Σ -structure is defined as above in Sect. 2.2 and likewise denoted by \mathcal{I} . Also the notion of valuation, both for classes and for messages, is as defined above and denoted by β . Finally, the set of traces associated with an IL-signature, a structure and a valuation, denoted by $\mathcal{T}(\Sigma, \mathcal{I})$ is as well unchanged.

We call an *implementation* any set of traces $I \subseteq \mathcal{T}(\Sigma, \mathcal{I})$. The *satisfaction relation*, a 5-place relation between signatures, structures, valuations, implementations and formulas, is defined for atomic interactions as follows:

$$\mathcal{I}, I, \beta \models_{\Sigma} T \text{ if } \mathcal{P}_{\mathcal{I}}(T, \beta) \cap I \neq \emptyset \text{ and } \mathcal{N}_{\mathcal{I}}(T, \beta) \cap I = \emptyset$$

For first-order interactions, the canonical extension is used:

$$\begin{aligned}
\mathcal{I}, I, \beta &\models_{\Sigma} x_1^c = x_2^c \text{ if } \beta(x_1^c) = \beta(x_2^c) \\
\mathcal{I}, I, \beta &\models_{\Sigma} \neg\psi \text{ if } \mathcal{I}, I, \beta \not\models_{\Sigma} \psi \\
\mathcal{I}, I, \beta &\models_{\Sigma} \psi_1 \wedge \psi_2 \text{ if } \mathcal{I}, I, \beta \models_{\Sigma} \psi_1 \text{ and } \mathcal{I}, I, \beta \models_{\Sigma} \psi_2 \\
\mathcal{I}, I, \beta &\models_{\Sigma} (\forall x^c)\psi \text{ if } \mathcal{I}, I, \beta[x^c \mapsto o] \models_{\Sigma} \psi \text{ for all } o \in O^c
\end{aligned}$$

where $\mathcal{I} = (\mathbf{O}, \mathcal{M}, \mu)$ and $\mathbf{O} = (O^c)_{c \in C}$ if $\Sigma = (\mathbf{C}, M)$ with $\mathbf{C} = (C, \leq_C)$.

For trace sets, also a notion of reduct is defined. Let Σ_1 and Σ_2 be IL-signatures, let $\sigma : \Sigma_1 \rightarrow \Sigma_2$ be an IL-signature morphism, let $X_2 = (X_2^c)_{c \in \Sigma_2}$ be a set of typed variables. Let $\mathcal{I}_2 = (\mathbf{O}_2, \mathcal{M}_2)$ be a Σ_2 -structure with $\mathbf{O}_2 = (O_2^c)_{c \in \Sigma_2}$, let β_2 be a valuation for X_2 in \mathcal{I}_2 . The reduct of an implementation $I \subseteq \mathcal{T}(\Sigma_2, \mathcal{I}_2)$ along σ is denoted by $I|_{\sigma}$ and defined by $I|_{\sigma} = I \cap \mathcal{T}(\Sigma_1, \mathcal{I}_2|_{\sigma}) \subseteq \mathcal{T}(\Sigma_1, \mathcal{I}_2|_{\sigma})$.

The *satisfaction condition* for first-order interactions, namely

$$\mathcal{I}_2|_{\sigma}, I_2|_{\sigma}, \beta_2|_{\sigma} \models_{\Sigma_1} \psi_1 \text{ iff } \mathcal{I}_2, I_2, \beta_2 \models_{\Sigma_2} \sigma(\psi_1)$$

holds for atomic interactions:

$$\begin{aligned}
&\mathcal{I}_2, I_2, \beta_2 \models_{\Sigma_2} \sigma(T_1) \\
&\text{iff } \mathcal{P}_{\mathcal{I}_2}(\sigma(T_1), \beta_2) \cap I_2 \neq \emptyset \text{ and } \mathcal{N}_{\mathcal{I}_2}(\sigma(T_1), \beta_2) \cap I_2 = \emptyset \text{ by definition} \\
&\text{iff } \mathcal{P}_{\mathcal{I}_2|_{\sigma}}(T_1, \beta_2|_{\sigma}) \cap I_2 \neq \emptyset \text{ and } \mathcal{N}_{\mathcal{I}_2|_{\sigma}}(T_1, \beta_2|_{\sigma}) \cap I_2 = \emptyset \text{ (see Sect. 2.3)} \\
&\text{iff } (\mathcal{P}_{\mathcal{I}_2|_{\sigma}}(T_1, \beta_2|_{\sigma}) \cap \mathcal{T}(\Sigma_1, \mathcal{I}_2|_{\sigma})) \cap I_2 \neq \emptyset \text{ and} \\
&\quad (\mathcal{N}_{\mathcal{I}_2|_{\sigma}}(T_1, \beta_2|_{\sigma}) \cap \mathcal{T}(\Sigma_1, \mathcal{I}_2|_{\sigma})) \cap I_2 = \emptyset \\
&\quad \text{since } \mathcal{P}_{\mathcal{I}_2|_{\sigma}}(T_1, \beta_2|_{\sigma}) \subseteq \mathcal{T}(\Sigma_1, \mathcal{I}_2|_{\sigma}) \text{ and} \\
&\quad \quad \mathcal{N}_{\mathcal{I}_2|_{\sigma}}(T_1, \beta_2|_{\sigma}) \subseteq \mathcal{T}(\Sigma_1, \mathcal{I}_2|_{\sigma}) \\
&\text{iff } \mathcal{P}_{\mathcal{I}_2|_{\sigma}}(T_1, \beta_2|_{\sigma}) \cap (\mathcal{T}(\Sigma_1, \mathcal{I}_2|_{\sigma}) \cap I_2) \neq \emptyset \text{ and} \\
&\quad \mathcal{N}_{\mathcal{I}_2|_{\sigma}}(T_1, \beta_2|_{\sigma}) \cap (\mathcal{T}(\Sigma_1, \mathcal{I}_2|_{\sigma}) \cap I_2) = \emptyset \\
&\quad \text{by associativity of intersection} \\
&\text{iff } \mathcal{P}_{\mathcal{I}_2|_{\sigma}}(T_1, \beta_2|_{\sigma}) \cap I_2|_{\sigma} \neq \emptyset \text{ and } \mathcal{N}_{\mathcal{I}_2|_{\sigma}}(T_1, \beta_2|_{\sigma}) \cap I_2|_{\sigma} = \emptyset \\
&\quad \text{by definition of reduct of implementations} \\
&\text{iff } \mathcal{I}_2|_{\sigma}, I_2|_{\sigma}, \beta_2|_{\sigma} \models_{\Sigma_1} T_1
\end{aligned}$$

For negation and conjunction, the satisfaction condition obviously holds by induction hypothesis. For universally quantified interactions,

$$\begin{aligned}
&\mathcal{I}_2, I_2, \beta_2 \models_{\Sigma_2} \sigma((\forall x^c)\psi_1) \\
&\text{iff } \mathcal{I}_2, I_2, \beta_2 \models_{\Sigma_2} (\forall x^{\sigma(c)})\sigma(\psi_1) \\
&\text{iff } \mathcal{I}_2, I_2, \beta_2[x^{\sigma(c)} \mapsto o] \models_{\Sigma_2} \sigma(\psi_1) \text{ for all } o \in O_2^{\sigma(c)} \\
&\text{iff } \mathcal{I}_2|_{\sigma}, I_2|_{\sigma}, \beta_2|_{\sigma}[x^{\sigma(c)} \mapsto o]|_{\sigma} \models_{\Sigma_1} \psi_1 \text{ for all } o \in O_2^{\sigma(c)} = (O_2|_{\sigma})^c \text{ by IH} \\
&\text{iff } \mathcal{I}_2|_{\sigma}, I_2|_{\sigma}, (\beta_2|_{\sigma})[x^c \mapsto o] \models_{\Sigma_1} \psi_1 \text{ for all } o \in (O_2|_{\sigma})^c \text{ by lemma 1 below} \\
&\text{iff } \mathcal{I}_2|_{\sigma}, I_2|_{\sigma}, \beta_2|_{\sigma} \models_{\Sigma_1} (\forall x^c)\psi_1
\end{aligned}$$

Lemma 1. $(\beta_2|_{\sigma})[x^c \mapsto o] = \beta_2[x^{\sigma(c)} \mapsto o]|_{\sigma}$

Proof.

$$\begin{aligned}
(\beta_2|_\sigma)[x^c \mapsto o](y^d) &= \begin{cases} \beta_2|_\sigma(y^d) & \text{if } y^d \neq x^c \\ o & \text{if } y^d = x^c \end{cases} \\
&= \begin{cases} \beta_2(y^{\sigma(d)}) & \text{if } y^d \neq x^c \\ o & \text{if } y^d = x^c \end{cases} \\
&= \begin{cases} \beta_2(y^{\sigma(d)}) & \text{if } y^{\sigma(d)} \neq x^{\sigma(c)} \\ o & \text{if } y^{\sigma(d)} = x^{\sigma(c)} \end{cases} \\
&= \beta_2[x^{\sigma(c)} \mapsto o](y^{\sigma(d)}) \\
&= \beta_2[x^{\sigma(c)} \mapsto o]|_\sigma(y^d)
\end{aligned}$$

3.3 IL institution

For an IL-signature Σ we define the notion of Σ -homomorphism between Σ -structures. Let $\mathcal{I}_1 = (\mathbf{O}_1, \mathcal{M}_1, \mu_1)$ and $\mathcal{I}_2 = (\mathbf{O}_2, \mathcal{M}_2, \mu_2)$ be Σ -structures. A Σ -homomorphism $h = ((h_c)_{c \in \Sigma}, h_m)$ from \mathcal{I}_1 to \mathcal{I}_2 consists of a family $h = (h_c)_{c \in \Sigma}$ of mappings $h_c : O_1^c \rightarrow O_2^c$ for each $c \in \Sigma$ and a mapping $h_m : \mathcal{M}_1 \rightarrow \mathcal{M}_2$ such that $h_m(\mu_1(m)) = \mu_2(m)$ for each $m \in \Sigma$.

It is easy to see that Σ -homomorphisms can be composed, and that the composition of Σ -homomorphisms is associative. Obviously there also exist identity Σ -homomorphisms. Thus, Σ -structures define a category.

Given that the satisfaction condition holds, IL-signatures, their sentences and their structures, and the satisfaction relation, define an institution [3]. The advantages of institutions are largely discussed in the literature and thus omitted here.

3.4 Alternative semantics

An alternative definition of the semantics of a first-order interaction could be given that takes, instead of a single implementation, two sets of traces into account. These sets can be regarded as the set of *proven positive* traces of an implementation and the set of *proven negative* traces of the same implementation. The satisfaction relation, thus, contains 6-tuples and is defined for the atomic case as follows:

$$\mathcal{I}, (P, N), \beta \models_\Sigma T \text{ if } \mathcal{P}_T(T, \beta) \subseteq P \text{ and } \mathcal{N}_T(T, \beta) \subseteq N$$

The satisfaction condition is termed almost the same as in the previous case:

$$\mathcal{I}_2|_\sigma, (P_2|_\sigma, N_2|_\sigma), \beta_2|_\sigma \models_{\Sigma_1} \psi_1 \text{ iff } \mathcal{I}_2, (P_2, N_2), \beta_2 \models_{\Sigma_2} \sigma(\psi_1)$$

where the definition of reduct of sets of traces, be these positive or negatives, coincides with the definition of reduct of an implementation as given above.

For atomic interactions, the satisfaction condition holds:

$$\begin{aligned}
& \mathcal{I}_2, (P_2, N_2), \beta_2 \models_{\Sigma_2} \sigma(T_1) \\
& \text{iff } \mathcal{P}_{\mathcal{I}_2}(\sigma(T_1), \beta_2) \subseteq P_2 \text{ and } \mathcal{N}_{\mathcal{I}_2}(\sigma(T_1), \beta_2) \subseteq N_2 \text{ by definition} \\
& \text{iff } \mathcal{P}_{\mathcal{I}_2|\sigma}(T_1, \beta_2|\sigma) \subseteq P_2 \text{ and } \mathcal{N}_{\mathcal{I}_2|\sigma}(T_1, \beta_2|\sigma) \subseteq N_2 \text{ (see Sect. 2.3)} \\
& \text{iff } (\mathcal{P}_{\mathcal{I}_2|\sigma}(T_1, \beta_2|\sigma) \cap \mathcal{T}(\Sigma_1, \mathcal{I}_2|\sigma)) \subseteq (P_2 \cap \mathcal{T}(\Sigma_1, \mathcal{I}_2|\sigma)) \text{ and} \\
& \quad (\mathcal{N}_{\mathcal{I}_2|\sigma}(T_1, \beta_2|\sigma) \cap \mathcal{T}(\Sigma_1, \mathcal{I}_2|\sigma)) \subseteq (N_2 \cap \mathcal{T}(\Sigma_1, \mathcal{I}_2|\sigma)) \\
& \quad \text{since for any sets } S_1, S_2 \text{ and } S_3, \text{ if } S_1 \subseteq S_2, \text{ then } S_1 \cap S_3 \subseteq S_2 \cap S_3 \\
& \quad \text{on the one hand, and on the other} \\
& \quad \mathcal{P}_{\mathcal{I}_2|\sigma}(T_1, \beta_2|\sigma) \subseteq \mathcal{T}(\Sigma_2, \mathcal{I}_2|\sigma), \\
& \quad \mathcal{N}_{\mathcal{I}_2|\sigma}(T_2, \beta_2|\sigma) \subseteq \mathcal{T}(\Sigma_2, \mathcal{I}_2|\sigma) \text{ and by transitivity of } \subseteq \quad (\diamond) \\
& \text{iff } \mathcal{P}_{\mathcal{I}_2|\sigma}(T_1, \beta_2|\sigma) \subseteq P_2|\sigma \text{ and } \mathcal{N}_{\mathcal{I}_2|\sigma}(T_1, \beta_2|\sigma) \subseteq N_2|\sigma \\
& \quad \text{by definition of reduct of trace sets} \\
& \text{iff } \mathcal{I}_2|\sigma, (P_2|\sigma, N_2|\sigma), \beta_2|\sigma \models_{\Sigma_1} T_1
\end{aligned}$$

Finally, a definition

$$\mathcal{I}, (P, N), \beta \models_{\Sigma} T \text{ if } \mathcal{P}_{\mathcal{I}}(T, \beta) \subseteq P \text{ and } \mathcal{N}_{\mathcal{I}}(T, \beta) \supseteq N$$

does not fulfil the satisfaction condition. In the reasoning of above, the step marked with a \diamond does not hold if the subset relation for the negative traces is inverted, since in general $N_2 \not\subseteq N_2 \cap \mathcal{T}(\Sigma_1, \mathcal{I}_2|\sigma)$. Consider the following example: Σ_1 contains only one class name `person` and only one message `hello`, Σ_2 contains class names `child` and `parent` and the same messages as Σ_1 , $\sigma : \Sigma_1 \rightarrow \Sigma_2$ maps `person` to `child` and is the identity on messages; \mathcal{I}_2 has one object `john` of type `child`, one object `mary` of type `parent`, a message `hello`, and μ_2 is the identity. Let t_1 be the trace `snd(john, mary, hello)`, and let t_2 be the trace `snd(john, john, hello)`. If $N_2 = \{t_1, t_2\}$, then $N_2|\sigma = N_2 \cap \mathcal{T}(\Sigma_1, \mathcal{I}_2|\sigma) = \{t_2\}$, since `person` is mapped by σ to `child`, and `mary` is a `parent`. Thus, $\{t_1, t_2\} = N_2 \not\subseteq N_2|\sigma = \{t_2\}$ given that $t_1 \neq t_2$. The term T_1 for this example is `neg(snd(x^{person} , x^{person} , hello))`, under a valuation $\beta_2|\sigma(x^{\text{person}}) = \text{john}$.

4 Discussion

In [2], a unary operator `refuse` is proposed, with

$$\begin{aligned}
& \mathcal{P}_{\mathcal{I}}(\text{refuse}(T), \beta) = \emptyset \text{ and} \\
& \mathcal{N}_{\mathcal{I}}(\text{refuse}(T), \beta) = \mathcal{P}_{\mathcal{I}}(T, \beta) \cup \mathcal{N}_{\mathcal{I}}(T, \beta).
\end{aligned}$$

The operator for negation can be obtained from `refuse` by defining

$$\text{neg}(T) = \text{alt}(\text{refuse}(T), \text{skip}),$$

i.e., `refuse` is more primitive.

The `ignore` operator can be obtained by defining `ignore($m; T$) = par(loop(m), T)`, where m , which stands for the sending or reception of a message m by any life-line, should then be added contiguous to the event base cases and whose semantics is given by $\mathcal{P}_{\mathcal{I}}(m, \beta) = \bigcup_{c_1, c_2 \in C} \{\text{snd}(s, r, \beta(m)), \text{rcv}(s, r, \beta(m)) : s \in O^{c_1}, r \in O^{c_2}\}$ and $\mathcal{N}_{\mathcal{I}}(m, \beta) = \emptyset$. This abbreviation for `ignore`, together with the altered semantics

for alt (in the negative fragment), conceptually corresponds to the semantics as defined in [1].

We have disregarded the loop operator with finite bounds and only considered the unbounded version $\text{loop}(0, *, T)$. We chose to do so because the other cases are only syntactic sugar. Indeed, $\text{loop}(n, n, T)$ is equivalent to $\text{seq}^n(T)$, where

$$\begin{aligned} \text{seq}^0(T) &\stackrel{\text{def}}{=} \text{skip} \quad \text{and} \\ \text{seq}^{n+1}(T) &\stackrel{\text{def}}{=} \text{seq}(T, \text{seq}^n(T)). \end{aligned}$$

Moreover, and taking advantage of the associativity of alt, $\text{loop}(m, n, T)$ is equivalent to $\text{alt}(\text{seq}^m(T), \dots, \text{seq}^n(T))$.

The semantics for assert needed to be adjusted in order to obtain invariance under change of notation, a desired because useful property of institutions. As discussed in Sect. 2.3, the additional operand, a (sub-)signature, allows to circumscribe the universe of traces with respect to which the complement is taken. Only reachable types are considered; a condition of this kind might also be necessary if the messages had structure, and because of this reason we take a subsignature and not just a subset of class names as additional operand. In this way, the semantics of assert remains stable if the signature is extended.

Another interesting possibility is to consider assert a quantifier:

$$\begin{aligned} \text{assert}(x^c, T) \quad \text{with} \\ \mathcal{P}_{\mathcal{I}}(\text{assert}(x^c, T), \beta) &= \mathcal{P}_{\mathcal{I}}(T, \beta) \quad \text{and} \\ \mathcal{N}_{\mathcal{I}}(\text{assert}(x^c, T), \beta) &= \{ e_1 \cdot e_2 \cdots e_n \mid \forall i \in \{1, \dots, n\}. \exists o, o', m \in \mathcal{I}. \\ &\quad (e_i = \text{snd}(o, o', m) \vee e_i = \text{rcv}(o, o', m)) \wedge \\ &\quad (o = \beta(x^c) \vee o' = \beta(x^c)) \} \\ &\quad \setminus \mathcal{P}_{\mathcal{I}}(T, \beta) \end{aligned}$$

The semantics of assert as well as the first-order fragment of the language was sensefully chosen. Particular applications may need further connectives, and possibly another notion of semantics, be this for assert or negation or whatever operand. The satisfaction condition should be respected particularly when interactions are to be combined in a bigger landscape, for instance when different software designers work on separate parts of a system, in order for the local properties to globally hold in the larger context.

References

1. Cengarle, M.V., Knapp, A.: UML 2.0 Interactions: Semantics and Refinement. In Jürjens, J., Fernandez, E.B., France, R., Rumpe, B., eds.: 3rd Int'l Workshop on Critical Systems Development with UML (CSDUML'04, Proceedings), Technical Report TUM-I0415, Institut für Informatik, Technische Universität München (2004) 85–99
2. Haugen, Ø., Husa, K.E., Runde, R.K., Stølen, K.: STAIRS towards formal design with sequence diagrams. Journal of Software and System Modeling (SoSyM) 4(4) (005) 355–367
3. Goguen, J.A., Burstall, R.M.: Introducing Institutions. In Clarke, E.M., Kozen, D., eds.: Logic of Programs. Volume 164 of Lecture Notes in Computer Science., Springer (1983) 221–256