Chapter 4: Semantics of ASL
1 Introduction

ASL is a specification language which was originally defined by a set of specification operators. Some of the operators of the original definition were operators to build structured specifications from smaller specifications or to make some modifications from a given specification, like for example the renaming of a specification. This language was not originally designed to be used directly but as a basis to define the semantics of higher-level specification languages. Two specification languages which used ASL to define their semantics were EML and PLUSS. A specific kind of operator which appeared in ASL was an operator which was used to \textit{behaviorally abstract} a given specification closing its model-theoretic semantics by an equivalence relation between algebras. Later on, different operators related to the one just described were developed. We will refer to these operators as \textit{behavioral operators}.

In this chapter we give a general semantic framework for \textit{behavioral operators}. In a general setting, these operators are parameterized by fixed but arbitrary equivalence relations. Three different kinds of \textit{behavioral operators} in ASL have been defined. We will refer to them as the \textit{abstract} operator, the \textit{behaviour} operator and the \textit{quotient} operator. All of them have a specification as an argument and they transform the model-theoretic semantics of the argument specification.

Intuitively, the \textit{abstract} operator extends the class of models of the argument specification with those models which are equivalent (by an equivalence relation between models) to some model belonging to the model-theoretical semantics of the argument specification.

The class of models of the \textit{behaviour} operator is defined by those models whose behaviour (denoted also as a model and defined via a congruence relation on values within a model) belongs to the class of models of the argument specification of the operator.

Finally, the class of models of the \textit{quotient} operator is defined by the closure under isomorphism of the quotient of the models associated to the semantics of the argument specification of the operator.

A formal semantics of these operators is given in [BHW95] by defining their signature and their model-theoretical semantics. They use a first-order logic with equality to define the sentences of specifications. Apart from giving the semantics of the operators, a theory which establishes different equivalences between the semantics of these operators is presented.

In [HS96], an alternative semantics for the \textit{behavioral operators} is given using just flat specifications as argument specifications. They use higher-order logic as specification logic and a similar theory as in [BHW95] to relate the semantics of the \textit{behavior} and \textit{abstract} operator is developed.

Since both semantics are quite independent of the specification logic of the specification language, it seems reasonable to make a generalisation of the semantics of these operators for an arbitrary but fixed institution. These institutions have to satisfy specific properties in order to include these operators in a version of ASL with structuring operators and they are a restricted ver-
sion of the institutions presented in [Tar85]. We refer to these institutions as
algebraic institutions (AINs) and the two main restrictions are that the cat-

gory of signatures is the category of first-order relational signatures and the
model functor of these institutions assigns to every first-order relational sig-
nature $\Sigma$ the category of $\Sigma$-algebras which we will denote as $\text{Alg}(\Sigma)$ instead
of an arbitrary category of models $\text{Mod}(\Sigma)$. This is necessary because these
institutions are used to define the semantics of different set of ASL operators
including behavioural operators which we will denote as $\text{BASLker}$ languages.
The signatures of the specification expressions of $\text{BASLker}$ languages are first-
order signatures since the semantics of some of the behavioural operators are
defined using a fixed but arbitrary partial $\Sigma$-congruence and therefore using
the internal structure of first-order signatures. These languages also include
the common operators of another set of operators of ASL which we will denote
as $\text{ASLker}$ languages. These common operators are base specifications (with
syntax $< \Sigma, \phi >$), a sum operator to define structured specifications and an
export operator. These restrictions are not needed to define the semantics of
the common operators of $\text{ASLker}$ languages. (See [BCH] for the semantics of
these operators in a fixed but arbitrary semiexact institution). In [BCH], the
semantics of the behavioural operators of $\text{BASLker}$ languages is given just for
an institution of infinitary first-order logic and for concrete observational equi-

In order to define a certain kind of proof systems for the deduction of sen-
tences from $\text{ASLker}$ languages, it is required additionally a normalisation func-
tion on specification expressions where the normal forms of specifications are
defined in terms of the export operator (with syntax $< \Sigma, \phi >, |\Sigma|$). This
normalisation function is also useful to relate the semantics of [BHW95] and [HS96].
We call any set of operators defined with a normalisation function and including
at least the common operators of $\text{ASLker}$ languages as $\text{ASLn}'$ language. To
generalise the semantics of [BHW95] and [HS96], we define behavioural algebraic
institutions ($\text{BAINs}$) which incorporates additional components to a fixed but
arbitrary algebraic institution (AINs) in order to define the semantics of the
behavioural operator of [HS96] and the normalisation function of the behavioural
operators with the semantics of [BHW95].

The structure of the chapter is as follows: first we introduce the abstract con-
cept of algebraic institution and we present two concrete algebraic institutions:
a first-order one and a higher-order one. Then, we give the semantics of the beha-
vioural operators and how to relate them in an arbitrary but fixed algebraic
institution following the ideas of [BHW95] and [BCH]. Next, we present beha-
vioural algebraic institutions, a normalisation function for the behavioural
operators presented previously and a relationship between the semantics of
[BHW95] and [HS96]. Finally, we present concrete equivalence relations and
a concrete behavioural algebraic institution using the concrete equivalences and
the higher-order algebraic institution presented in the first section.
2 Semiexact algebraic institutions

In this section, we will present the abstract semantic framework to define the semantics of different operators of ASL including the behavioural operators. We will assume predefined basic concepts of institutions, which can be found in [DGS91], [GB92], or in [Tar].

Definition 2.1 An algebraic institution \( \text{AINS} \) is an institution which consists of:

- The category of first-order relational signatures \( \text{AlgSig} \) whose objects are first-order relational signatures and morphisms are signature morphisms.
- a functor \( \text{Sen}_{\text{AINS}} : \text{AlgSig} \to \text{Set} \)
- the functor \( \text{Alg} : \text{AlgSig}^\text{op} \to \text{Cat} \) where:
  - for any \( \Sigma \in |\text{AlgSig}| \), \( \text{Alg}(\Sigma) \) is the category of \( \Sigma \)-algebras
  - for any morphism \( \sigma : \Sigma \to \Sigma' \) in \( \text{AlgSig} \), \( \text{Alg}(\sigma) \) is the reduct functor \( \downarrow_\sigma : \text{Alg}(\Sigma') \to \text{Alg}(\Sigma) \).
- for each \( \Sigma \in |\text{AlgSig}| \) a satisfaction relation

\[
|=_{\text{AINS},\Sigma} : |\text{Alg}(\Sigma)| \times \text{Sen}_{\text{AINS}}(\Sigma)
\]

such that

- the satisfaction condition holds for any signature morphism \( \sigma : \Sigma \to \Sigma' \) and for any formula \( \phi \in \text{Sen}_{\text{AINS}}(\Sigma) \). This condition is formally defined as:

\[
\forall A \in |\text{Alg}(\Sigma')|, A |_{=_{\text{AINS},\Sigma'}} \text{Sen}_{\text{AINS}}(\sigma)(\phi) \Leftrightarrow A_{\sigma} |_{=_{\text{AINS},\Sigma}} \phi
\]

- an abstract satisfaction condition holds for any formula \( \phi \) in \( \text{Sen}_{\text{AINS}}(\Sigma) \):

\[
\forall A, B \in |\text{Alg}(\Sigma)|, A \equiv B \Rightarrow (A |_{=_{\text{AINS},\Sigma}} \phi \Leftrightarrow B |_{=_{\text{AINS},\Sigma}} \phi)
\]

Notation and comments:

- The main differences between algebraic institutions and the original definition of institutions is that the category of signatures is not an arbitrary category but the category of first-order relational signatures and therefore the model functor is also restricted assigning to every \( \Sigma \)-algebra not an arbitrary category of models but the category of \( \Sigma \)-algebras. Another difference is that it is added explicitly the abstract satisfaction condition which almost all institutions satisfy.
• We will normally refer to first-order relational signatures just as relational signatures. For any relational signature \( \Sigma = (S, Op, Pr) \in \text{Algsig} \), the functions \text{Sorts}(\Sigma), \text{Ops}(\Sigma) \) and \text{Prs}(\Sigma) \) will return \( S, Op \) and \( Pr \) respectively.

For any relational signature \( \Sigma = (S, Op, Pr) \), if \( Pr = \emptyset \) we will denote it just by \( \Sigma = (S, Op) \) and it will be normally referred just as signature.

For any relational signature \( \Sigma = (S, Op, Pr) \in \text{Algsig} \) and for a fixed but arbitrary \( S \)-sorted infinite denumerable set of variables \( X \), \( T_\Sigma(X) \) will denote the \((S,Op)\)-term algebra freely generated by \( X \) and \( P_\Sigma(X) \) will denote the set of atoms of the form \( p(t_1, \ldots, t_n) \) where \( p : s_1 \times \ldots \times s_n \in \text{Prs}(\Sigma) \) and \( t_1 \in T_{\Sigma,s_1}(X), \ldots, t_n \in T_{\Sigma,s_n}(X) \).

For any algebra \( A \in \text{Algs}\text{ains}(\Sigma) \) and an \( S \)-sorted valuation \( \alpha : X \to A \), \( I_\alpha : T_\Sigma(X) \to A \) will denote the unique extension to a \( \Sigma \)-morphism of the valuation \( \alpha \), and for the case of \( p(t_1, \ldots, t_n) \in P_\Sigma(X) \), \( I_\alpha(p(t_1, \ldots, t_n)) \) will hold if and only if \( (I_\alpha t_1, \ldots, I_\alpha t_n) \in p_A \). We will refer to them as the interpretations of terms and atoms associated to \( \alpha \).

We will assume predefined the function \( \alpha \bigcup \{(x_1, v_1), \ldots, (x_n, v_n)\} \) which given a valuation \( \alpha : X \to A \) and a set of pairs of the form \( \{(x_1, v_1), \ldots, (x_n, v_n)\} \) such that \( x_i \in X_{s_i} \) and \( v_i \in A_{s_i} \) for any \( i \in [1..n] \), it will return the usual update of the valuation \( \alpha \) with the given set of pairs.

• If \( \Sigma \subseteq \Sigma' \), we will normally denote by \( \sigma : \Sigma \hookrightarrow \Sigma' \) the obvious embedding morphism which we will normally refer to as inclusion.

• Since it is well known that Algsig has pushouts, the pushout object of any pair of morphisms \( \sigma : \Sigma_0 \to \Sigma_1 \), \( \sigma' : \Sigma_0 \to \Sigma_2 \) in Algsig (where \( \Sigma_0, \Sigma_1, \Sigma_2 \in \text{Algsig} \)) will be denoted in general as \( PO(\sigma : \Sigma_0 \to \Sigma', \sigma' : \Sigma_0 \to \Sigma''') \) and if the pair of morphisms are both inclusions the pushout object will be normally denoted as \( \Sigma_1 +_{\Sigma_0} \Sigma_2 \) and the pushout morphisms as \( \text{incl} : \Sigma_1 \to \Sigma_1 +_{\Sigma_0} \Sigma_2, \text{inr} : \Sigma_2 \to \Sigma_1 +_{\Sigma_0} \Sigma_2 \). In this last case, we can assume in general that either incl or inr are inclusions but not both.

• We will also drop usually the subscript of the functor \( \text{Senains} \) and the subscripts of \( \models_{\text{ains}} \) if it can be inferred from the context.

We will also refer as

\[
\models_{\text{ains}} : \text{Algs}(\Sigma) \times \mathcal{P}(\text{Senains}(\Sigma))
\]

the obvious extension of the satisfaction relation to a set of sentences.

• For any signatures \( \Sigma, \Sigma' \in \text{Algsig} \) and for any signature morphism \( \sigma : \Sigma \to \Sigma' \), the morphism

\[
\text{Senains}(\sigma) : \text{Senains}(\Sigma) \to \text{Senains}(\Sigma')
\]

will be normally denoted just by \( \sigma : \\text{Senains}(\Sigma) \to \text{Senains}(\Sigma') \).
Definition 2.2 An institution $\text{INS} = (\text{Sign}_{\text{INS}}, \text{Sen}_{\text{INS}} : \text{Sign}_{\text{INS}} \to \text{Set}, \\
\text{Mod}_{\text{INS}} : \text{Sign}^2_{\text{INS}} \to \text{Cat}, \models_{\text{INS}} \Sigma \models_{\text{INS}} \Sigma', \text{Sign}_{\text{INS}})$ is semiequivalent if for any pushout in $\text{Sign}_{\text{INS}}$ (ind : $\Sigma_1 \to \Sigma', \text{inr} : \Sigma_2 \to \Sigma'$) of any pair of morphisms ($\sigma : \Sigma_0 \to \Sigma_1, \sigma' : \Sigma_0 \to \Sigma_2$) and for any models $M_1 \in \text{Mod}_{\text{INS}}(\Sigma_1), M_2 \in \text{Mod}_{\text{INS}}(\Sigma_2)$ such that $M_1|_{\sigma} = M_2|_{\sigma'}$, there exists an unique model $M \in \text{Mod}_{\text{INS}}(\Sigma')$ such that $M|_{\text{ind}} = M_1$ and $M|_{\text{inr}} = M_2$.

Notation: This definition is equivalent to the definition of institution with semi-composable signatures presented in [Tar].

Proposition 2.3 Any algebraic institution $\text{AINS}$ is semiequivalent.

Now we present two concrete algebraic institutions which we will use in the following chapters:

Proposition 2.4 The tuple

$$\text{FOLEQ} = (\text{AlgSig}, \text{Sen}_{\text{FOLEQ}}, \text{Alg}, \models_{\text{FOLEQ}} \Sigma \models_{\text{AlgSig}} \Sigma)$$

such that:

- $\text{Sen}_{\text{FOLEQ}} : \text{AlgSig} \to \text{Set}$ is a functor defined in the following way:
  - For each $\Sigma = (S, Op) \in |\text{AlgSig}|$, the set $\text{Sen}(\Sigma)$ is inductively defined by the following set of rules:
    - **true, false** $\in \text{Sen}(\Sigma)$.
    - If $t, r \in T_{\Sigma}(X_s)$ for $s \in \text{Sorts}(\Sigma)$ then $t = r \in \text{Sen}(\Sigma)$.
    - If $p : s_1 \times \ldots \times s_n \in \text{Prs}(\Sigma)$ and $t_1 \in T_{\Sigma}(X_{s_1}), \ldots, t_n \in T_{\Sigma}(X_{s_n})$ then $p(t_1, \ldots, t_n) \in \text{Sen}(\Sigma)$
    - If $\phi, \psi \in \text{Sen}(\Sigma)$ then $\neg \phi, \phi \land \psi, \phi \lor \psi, \phi \supset \psi \in \text{Sen}(\Sigma)$.
  - If $x \in X_s$ and $\phi \in \text{Sen}(\Sigma)$ then $\forall x : s.\phi, \exists x : s.\phi \in \text{Sen}(\Sigma)$
  - For each signature morphism $\sigma : \Sigma \to \Sigma'$ the morphism
    $\text{Sen}_{\text{FOLEQ}}(\sigma) : \text{Sen}_{\text{FOLEQ}}(\Sigma) \to \text{Sen}_{\text{FOLEQ}}(\Sigma')$
    is the usual renaming function between first-order sentences.

- for each $\Sigma \in |\text{AlgSig}|$ the satisfaction relation $\models_{\text{FOLEQ}, \Sigma}$ is defined as follows:
  $$\forall A \in |\text{Alg}(\Sigma)|, \forall \phi \in \text{Sen}_{\text{FOLEQ}}(\Sigma).A \models_{\text{FOLEQ}, \Sigma} \phi \iff$$
  $$\forall \rho \in X \to A.A \models_{\text{FOLEQ}, \Sigma, \rho} \phi$$

where for any $\Sigma \in |\text{AlgSig}|$ and for any $\rho : X \to A$, the relation $\models_{\text{FOLEQ}, \Sigma, \rho}$ is simultaneously defined by induction as follows:
- $A \models_{\rho} \text{true holds.}$

- If $I_{\rho}(t) = I_{\rho}(r)$ then $A \models_{\rho} t = r$ holds.

- If $A \models_{\rho} \phi$ does not hold then $A \models_{\rho} \neg \phi$ holds.

- If $A \models_{\rho} \phi$ holds and $A \models_{\rho} \psi$ holds then $A \models_{\rho} \phi \land \psi$ holds.

- If $A \models_{\rho} \phi$ holds or $A \models_{\rho} \psi$ holds then $A \models_{\rho} \phi \lor \psi$ holds.

- If $A \models_{\rho} \phi$ does not hold or $A \models_{\rho} \psi$ holds then $A \models_{\rho} \phi \lor \psi$ holds.

- If $\forall v \in A, A \models_{\rho \cup \{ \langle x, v \rangle \}} \phi$ holds then $A \models_{\rho} \forall x : s. \phi$ holds.

- If $A \models_{\rho \cup \{ \langle x, v \rangle \}} \phi$ holds for some $v \in A$, then $A \models_{\rho} \exists x : s. \phi$ holds.

is an algebraic institution.

**Proof sketch:**

We have to prove that $\text{Sen}_{\text{FOL=EQ}}$ is a functor (which is straightforward) and we have to show that the relation $\models_{\text{FOL=EQ}, \Sigma} \models_{\text{AlgSig}}$:

- satisfies the satisfaction conditions which follows by a generalization of the satisfaction lemma. See for example [GB92].

- satisfies the abstract satisfaction condition which follows trivially by the satisfaction condition.

Before presenting an algebraic institution for higher-order logic ($HOL$), we give some basic definitions which will be used in its definition.

**Definition 2.5** For each $\Sigma = (S, Op, Pr) \in [\text{AlgSig}]$, the set $\text{Types}_{HOL}(\Sigma)$ is inductively defined by the following set of rules:

- If $s \in S$ then $s \in \text{Types}_{HOL}(\Sigma)$.

- If $\tau_1 \in \text{Types}_{HOL}(\Sigma), \ldots, \tau_n \in \text{Types}_{HOL}(\Sigma)$ and $n \geq 0$ then $[\tau_1, \ldots, \tau_n] \in \text{Types}_{HOL}(\Sigma)$.

**Notation:** The type $[]$ will be normally denoted by $\text{Prop}$.

For any signature morphism $\sigma : \Sigma \to \Sigma'$, we will also denote by $\sigma$ the usual renaming function between types $\sigma : \text{Types}_{HOL}(\Sigma) \to \text{Types}_{HOL}(\Sigma')$.  

7
Definition 2.6 The semantic function $[\tau]_A$ is inductively defined for any type \( \tau \in \text{Type}_{\text{HOL}}(\Sigma) \) and for any \( \Sigma \)-algebra \( A \) as follows:

$$
[s]_A = A_s
$$

$$
[[\tau_1, \ldots, \tau_n]]_A = \mathcal{P}([\tau_1]_A \times \ldots \times [\tau_n]_A)
$$

Notation: The semantics of \text{Prop} is a set of two elements: the empty set and the set with the empty tuple. These two elements will be denoted as \text{ff} and \text{tt} respectively.

Definition 2.7 The set \text{Sen}_{\text{HOL}}(\Sigma, X_{\text{HOL}}, \tau) for a given \text{Type}_{\text{HOL}}(\Sigma)\text{-sorted infinite denumemable set of variables} \ X_{\text{HOL}} and for every \( \tau \in \text{Type}_{\text{HOL}}(\Sigma) \) is inductively defined by the following set of rules:

- If \( x \in X_{\text{HOL},\tau} \) then \( x \in \text{Sen}_{\text{HOL}}(\Sigma, X_{\text{HOL}}, \tau) \).

- If \( f : s_1 \times \ldots \times s_n \rightarrow s \in \text{Ops}(\Sigma) \), \( t_1 \in T_{\Sigma,s_1}(\text{<} X_{\text{HOL},s} \text{>} s \in S) \), \ldots,
  \( t_n \in T_{\Sigma,s_n}(\text{<} X_{\text{HOL},s} \text{>} s \in S) \)
  then \( f(t_1, \ldots, t_n) \in \text{Sen}_{\text{HOL}}(\Sigma, X_{\text{HOL}}, s) \).

- If \( p : s_1 \times \ldots \times s_n \in \text{Prs}(\Sigma) \), \( t_1 \in T_{\Sigma,s_1}(\text{<} X_{\text{HOL},s} \text{>} s \in S) \), \ldots,
  \( t_n \in T_{\Sigma,s_n}(\text{<} X_{\text{HOL},s} \text{>} s \in S) \)
  then \( p(t_1, \ldots, t_n) \in \text{Sen}_{\text{HOL}}(\Sigma, X_{\text{HOL}}, \text{Prop}) \).

- If \( \tau_1, \ldots, \tau_n \in \text{Type}_{\text{HOL}}(\Sigma) \), \( x_1 \in X_{\text{HOL},\tau_1}, \ldots, x_n \in X_{\text{HOL},\tau_n} \) and \( \phi \in \text{Sen}_{\text{HOL}}(\Sigma, X_{\text{HOL}}, \text{Prop}) \) then
  \( \lambda(x_1 : \tau_1, \ldots, x_n : \tau_n).\phi \in \text{Sen}_{\text{HOL}}(\Sigma, X_{\text{HOL}}, [\tau_1, \ldots, \tau_n]) \).

- If \( \tau_1, \ldots, \tau_n \in \text{Type}_{\text{HOL}}(\Sigma) \), \( t \in \text{Sen}_{\text{HOL}}(\Sigma, X_{\text{HOL}}, [\tau_1, \ldots, \tau_n]) \),
  \( t_1 \in \text{Sen}_{\text{HOL}}(\Sigma, X_{\text{HOL}}, \tau_1), \ldots, t_n \in \text{Sen}_{\text{HOL}}(\Sigma, X_{\text{HOL}}, \tau_n) \)
  then \( t(t_1, \ldots, t_n) \in \text{Sen}_{\text{HOL}}(\Sigma, X_{\text{HOL}}, \text{Prop}) \).
\[ \Phi = \left( \forall x : \tau . \phi \in \text{Sen}_{\text{HOL}}(\Sigma, X_{\text{HOL}}, \text{Prop}) \right). \]

- If \( \phi, \phi' \in \text{Sen}_{\text{HOL}}(\Sigma, X, \text{Prop}) \) then \( \phi \supset \phi' \in \text{Sen}_{\text{HOL}}(\Sigma, X, \text{Prop}) \).

**Notation:** We will denote by \( \text{Terms}_{\text{HOL}}(\Sigma, X_{\text{HOL}}) \) the set

\[ \bigcup_{\tau \in \text{Types}_{\text{HOL}}(\Sigma)} \text{Sen}_{\text{HOL}}(\Sigma, X_{\text{HOL}}, \tau) \]

For any signature morphism \( \sigma : \Sigma \to \Sigma' \), we will also denote by \( \sigma \) the usual renaming function between terms

\[ \sigma : \text{Terms}_{\text{HOL}}(\Sigma, X_{\text{HOL}}) \to \text{Terms}_{\text{HOL}}(\Sigma', X_{\text{HOL}}) \]

such that for any type \( \tau \in \text{Types}_{\text{HOL}}(\Sigma) \), for any higher-order sentence \( \phi \in \text{Sen}_{\text{HOL}}(\Sigma, X_{\text{HOL}}, \tau) \), \( \sigma(\phi) \in \text{Sen}_{\text{HOL}}(\Sigma', X_{\text{HOL}}, \sigma(\tau)) \).

The usual definition of \( \beta \)-equality between terms in \( \text{Terms}_{\text{HOL}}(\Sigma, X_{\text{HOL}}) \) identifying also \( \alpha \)-convertible terms will be denoted by \( =_{\beta} \) and the usual substitution operation avoiding name clashes will be denoted by \( t\{x / t\} \) for any \( t \in \text{Terms}_{\text{HOL}}(\Sigma, X_{\text{HOL}}) \), \( t' \in \text{Sen}_{\text{HOL}}(\Sigma, X_{\text{HOL}}, \tau) \) and \( x \in X_{\text{HOL}, \tau} \).

**Definition 2.8** The function \( [t]_{\rho, A} \) for any term \( t \in \text{Terms}_{\text{HOL}}(\Sigma, X_{\text{HOL}}) \), for any algebra \( A \in \text{Alg}(\Sigma) \), for any \( \text{Types}_{\text{HOL}}(\Sigma) \)-sorted valuation \( \rho \) which for every \( \tau \in \text{Types}_{\text{HOL}}(\Sigma) \), \( \rho_{\tau} \) has arity \( \rho_{\tau} : X_{\text{HOL}, \tau} \to [\tau]_{A} \) is inductively defined by the structure of \( t \) as follows:

\[ [x]_{\rho, A} = \rho_{\tau}(x) \]

\[ [f(t_{1}, \ldots, t_{n})]_{\rho, A} = f_{A}([t_{1}]_{\rho, A}, \ldots, [t_{n}]_{\rho, A}) \]

\[ [p(t_{1}, \ldots, t_{n})]_{\rho, A} = \text{if } ([t_{1}]_{\rho, A}, \ldots, [t_{n}]_{\rho, A}) \in p_{A} \text{ then } tt \text{ else } ff \]

\[ [\lambda(x_{1} : \tau_{1}, \ldots, x_{n} : \tau_{n}) . \phi]_{\rho, A} = \]

\[ \{ (v_{1}, \ldots, v_{n}) | v_{1} \in [\tau_{1}]_{\rho, A}, \ldots, v_{n} \in [\tau_{n}]_{\rho, A}, [\phi]_{\rho \cup \{ (x_{1}, v_{1}), \ldots, (x_{n}, v_{n}) \}} = tt \} \]

\[ [t \ (t_{1}, \ldots, t_{n})]_{\rho, A} = \]

\[ \text{if } ([t_{1}]_{\rho, A}, \ldots, [t_{n}]_{\rho, A}) \in [t]_{\rho, A} \text{ then } tt \text{ else } ff \]

\[ [\phi \supset \phi']_{\rho, A} = \text{if } [\phi]_{\rho, A} = tt \text{ then } [\phi]_{\rho, A} \text{ else } tt \]

\[ [\forall x : \tau . \phi]_{\rho, A} = \text{if } \forall v \in [\tau]_{A} . [\phi]_{\rho \cup \{ (x, v) \}, A} = tt \text{ then } tt \text{ else } ff \]
Proposition 2.9 The tuple

\[ \text{HOL} = (\text{AlgSig}, \text{Sen}_{\text{HOL}}, \text{Alg}, \models_{\text{HOL}, \Sigma}^{\Sigma \in \text{AlgSig}}) \]

such that:

- \text{Sen}_{\text{HOL}} : \text{AlgSig} \rightarrow \text{Set} is a functor defined in the following way:
  - For each \( \Sigma \in \text{AlgSig} \), the set \text{Sen}_{\text{HOL}}(\Sigma) is defined as
    \[ \text{Sen}_{\text{HOL}}(\Sigma) = \text{Sen}_{\text{HOL}}(\Sigma, \text{X}_{\text{HOL}}, \text{Prop}) \]
    for a given \( \text{Type}_{\text{HOL}}(\Sigma) \)-sorted infinite denumerable set of variables \( \text{X}_{\text{HOL}} \)
  - For each signature morphism \( \sigma : \Sigma \rightarrow \Sigma' \) the morphism
    \[ \text{Sen}_{\text{HOL}}(\sigma) : \text{Sen}_{\text{HOL}}(\Sigma) \rightarrow \text{Sen}_{\text{HOL}}(\Sigma') \]
    is the usual renaming function between sentences using \( \sigma : \Sigma \rightarrow \Sigma' \)
    and it will also be denoted just by \( \sigma \).

- for each \( \Sigma \in \text{AlgSig} \), for all \( A \in \text{Alg}(\Sigma) \), for all \( \phi \in \text{Sen}_{\text{HOL}}(\Sigma) \), the satisfaction relation \( A \models_{\Sigma} \phi \) holds if and only if for any \( \text{Type}_{\text{HOL}}(\Sigma) \)-sorted valuation \( \rho \) which for every \( \tau \in \text{Type}_{\text{HOL}}(\Sigma) \), \( \rho_\tau \) has arity \( \text{X}_{\text{HOL}, \tau} \rightarrow [\tau]_A, [\phi]_{\rho,A} = \text{tt} \)

is an algebraic institution.

Proof sketch:
We have to prove that \text{Sen}_{\text{HOL}} is a functor (which is straightforward) and we have to prove that the relation \( \models_{\text{HOL}, \Sigma}^{\Sigma \in \text{AlgSig}} \) satisfies:

- the satisfaction condition which follows by induction on \text{Terms}_{\text{HOL}} in a similar way as in the first-order case.

- the abstract satisfaction condition holds extending the isomorphism between algebras to higher-order types in the obvious way. See [HS96] for details of the proofs.

3 Abstract semantics of different operators of ASL

In this section, we will present the semantics of different operators of ASL including the behavioural operators presented in [BHW95] as we briefly explained in the introduction.
**Definition 3.1** An ASLker specification language with a fixed but arbitrary algebraic institution $AINS$ is a specification language defined with a set of operators including basic specifications, an export operator and an operator for structuring specifications which we will refer as the sum operator. The syntax of these operators is the following:

$$SP_0 ::= \langle \Sigma, \Phi \rangle$$

$$SP_1|\Sigma$$

$$SP_1 +_\Sigma SP_2$$

where the signature $\Sigma = (S, Op) \in |\textit{AlgSig}|$ and $\Phi \subseteq \textit{Sen}_{AINS}(\Sigma)$. Let ASL be an ASLker specification language and let $SP\textit{EX}(ASL)$ be the set of specification expressions of this language. The semantics of an ASLker language ASL is inductively defined by the functions $\text{Signature} : SP\textit{EX}(ASL) \rightarrow |\textit{AlgSig}|$, $\text{Symbols} : SP\textit{EX}(ASL) \rightarrow |\textit{AlgSig}|$ and $\text{Models} : SP\textit{EX}(ASL) \rightarrow \textit{Alg}(\text{Signature}(SP))$.

The function $\text{Signature}$ must return the signature with just the visible symbols of the given specification, whereas the function $\text{Symbols}$ must return the signature with the visible and hidden symbols of the given specification. These functions must satisfy the following conditions:

$$\text{Signature}(\langle \Sigma, \Phi \rangle) = \Sigma$$

$$\text{Symbols}(\langle \Sigma, \Phi \rangle) = \Sigma$$

$$\text{Models}(\langle \Sigma, \Phi \rangle) = \{ A \mid A \models_{AINS, \Sigma} \Phi \}$$

where the signature $\Sigma = (S, Op) \in |\textit{AlgSig}|$ and $\Phi \subseteq |\textit{Sen}_{AINS}(\Sigma)|$.

$$\text{Signature}(SP|\Sigma) = \Sigma$$

$$\text{Symbols}(SP|\Sigma) = \text{Symbols}(SP)$$

$$\text{Models}(SP|\Sigma) = \{ A|\Sigma \mid A \in \text{Models}(SP) \}$$

where $SP$ ranges over specification expressions, the signature $\Sigma = (S, Op) \in |\textit{AlgSig}|$ and $\Sigma \subseteq \text{Signature}(SP)$.
\[
\text{Signature}(SP_1 + \Sigma SP_2) = \text{Signature}(SP_1) + \Sigma \text{Signature}(SP_2)
\]
\[
\text{Symbols}(SP_1 + \Sigma SP_2) = \text{Symbols}(SP_1) + \Sigma \text{Symbols}(SP_2)
\]
\[
\text{Models}(SP_1 + \Sigma SP_2) = \\
\{ A \mid A \in \text{Alg}(\text{Signature}(SP_1) + \Sigma \text{Signature}(SP_2)), \\
A|_{int} \in \text{Models}(SP_1), A|_{nr} \in \text{Models}(SP_2) \}
\]

where the signature \( \Sigma = (S,Op) \in [\text{AlgSig}] \), \( SP_1, SP_2 \) ranges over specification expressions in \( \text{SPEX}(\text{ASLK}) \), \( \Sigma \subseteq \text{Signature}(SP_1) \), \( \Sigma \subseteq \text{Signature}(SP_2) \) and the pushouts
\[
\text{Signature}(SP_1) + \Sigma \text{Signature}(SP_2)
\]

and
\[
\text{Symbols}(SP_1) + \Sigma \text{Symbols}(SP_2)
\]

are the pushouts of the following diagram:

\[
\begin{array}{ccc}
\text{Sym}(SP_1) & \xrightarrow{is} & \text{Sym}(SP_1) + \Sigma \text{Sym}(SP_2) \\
\text{Sign}(SP_1) & \xrightarrow{i} & \Sigma \xrightarrow{i'} \text{Sign}(SP_2) & \xrightarrow{i's'} \text{Sym}(SP_2) \\
\end{array}
\]

Since \( \text{Signature}(SP_1) + \Sigma \text{Signature}(SP_2) \) is a pushout \( iss \) is the unique morphism with arity
\[
iss : \text{Signature}(SP_1) + \Sigma \text{Signature}(SP_2) \rightarrow \\
\text{Symbols}(SP_1) + \Sigma \text{Symbols}(SP_2)
\]

and the pushouts can be chosen in such a way that \( iss \) is an inclusion.

**Comment:** The functions \( \text{Signature} \) and \( \text{Symbols} \) are needed to define different proof systems for \( \text{ASLker} \) languages. See the next chapter for the definition of these proof systems.

**Definition 3.2** Let \( \text{ASL} \) be an \( \text{ASLker} \) specification language and \( \text{Symbols}_{nf} \) a function with arity \( \text{Symbols}_{nf} : \text{SPEX}(\text{ASLN}) \rightarrow [\text{AlgSig}] \). A function \( \eta f \) with arity \( \eta f : \text{SPEX}(\text{ASLN}) \rightarrow \text{SPEX}(\text{ASLN}) \) is a normalisation function if it satisfies the following conditions which we will refer as normalisation conditions:
• For all $SP \in SP\text{EX}(ASLN)$,

$$nf(SP) = < Symbols_{nf}(SP), \Phi > |Signature(SP)$$

for some $\Phi \subseteq |Sen_{AINS}(Symbols_{nf}(SP))|$.

• For all $SP \in SP\text{EX}(ASLN)$, $Symbols(SP) \subseteq Symbols_{nf}(SP)$.

• $A \in Models(nf(SP)) \Leftrightarrow A \in Models(SP)$

**Comment:** If the $ASLker$ language just contains the common operators of these languages, the function $Symbols_{nf}$ coincides with the functions $Symbols$, but this will not be the case for example for the common operators of $BASLnf$ specification languages presented in later sections. The function $Symbols_{nf}$ is also needed to define certain kind of proof systems associated to the $ASLnf$ specifications languages presented in next definition.

**Definition 3.3** An $ASLnf$ specification language is an $ASLker$ specification language whose semantic definition also requires the definition of a normalisation function $nf$ together with the function $Symbols_{nf}$. The functions $nf$ and $Symbols_{nf}$ must satisfy the following conditions:

$$nf(< \Sigma, \Phi >) = < \Sigma, \Phi > |\Sigma$$

$$Symbols_{nf}(< \Sigma, \Phi >) = \Sigma$$

where the signature $\Sigma = (S, Op) \in |AlgSig|$ and $\Phi \subseteq |Sen_{AINS}(\Sigma)|$

$$nf(SP_1 +_\Sigma SP_2) = < \Sigma'_1 +_\Sigma \Sigma'_2, int(\Phi_1) \cup int(\Phi_2) > |\Sigma_1 +_\Sigma \Sigma_2$$

$$Symbols_{nf}(SP_1 +_\Sigma SP_2) = \Sigma'_1 +_\Sigma \Sigma'_2$$

where $nf(SP_1) = < \Sigma'_1, \Phi_1 > |\Sigma_1$, $nf(SP_2) = < \Sigma'_2, \Phi_2 > |\Sigma_2$

and $int$ and $inr$ are fixed but arbitrary pushouts of $i_1 : \Sigma \hookrightarrow Signature(SP_1)$ and $i_2 : \Sigma \hookrightarrow Signature(SP_2)$

$$nf(SP|_\Sigma) = < \Sigma', \Phi > |\Sigma$$

$$Symbols_{nf}(SP) = \Sigma'$$

where $nf(SP) = < \Sigma', \Phi > |\Sigma'$.  

**Notation:** In the following, $ASL$ will range over an $ASLker$ or an $ASLnf$ specification language.
Proposition 3.4  The nf function of the previous definition satisfies the normalisation conditions.

Proof sketch:
The proof is by an easy induction on specification expressions.

Definition 3.5  Let ASL be an ASLker or an ASLnf specification language with an arbitrary but fixed algebraic institution AINS. For any specification expression \( SP \in SP\,EX(ASL) \) and for any sentence \( \phi \in \text{Scl}(\text{Signature}(SP)) \) the satisfaction relation \( \models_{\text{AINS,Signature}(SP)} \) is defined as follows:

\[
SP \models_{\text{AINS,Signature}(SP)} \phi \iff \forall A \in \text{Models}(SP). A \models_{\text{AINS,Signature}(SP)} \phi
\]

Definition 3.6  Assume that \( \Sigma \in |\text{AlgSig}| \). A partial \( \Sigma \)-congruence on a \( \Sigma \)-algebra \( A \) (normally denoted by \( \approx_A \)) is defined for every sort \( s \) in \( S \) as a symmetric and transitive relation (normally denoted as \( \approx_{s,A} \)) with domain \( s_A \times s_A \). This relation is compatible with every operation \( f_A : s_1 \times \cdots \times s_n \rightarrow s \) where \( f \in \text{Ops}(\Sigma) \). This means that for every \( v_1, w_1 \in s_1,A, \ldots, v_n, w_n \in s_n,A \), if \( v_1 \approx_{s_1,A} w_1, \ldots, v_n \approx_{s_n,A} w_n \) then \( f_A(v_1, \ldots, v_n) \approx_{s_A} f_A(w_1, \ldots, w_n) \), and it is also compatible with every predicate \( p_A : s_1 \times \cdots \times s_n \rightarrow A \) which means that for every \( v_1, w_1 \in s_1,A, \ldots, v_n, w_n \in s_n,A \), if \( v_n, w_n \in s_n,A \), if \( v_1 \approx_{s_1,A} w_1, \ldots, v_n \approx_{s_n,A} w_n \) then \( p_A(v_1, \ldots, v_n) \approx_{A} p_A(w_1, \ldots, w_n) \).

Notation: A family of partial \( \Sigma \)-congruences \( <\approx_A>_{A \in \text{Alg}(\Sigma)} \) will be denoted just by \( \approx \) if it can be inferred from the context.

Definition 3.7  Assume that \( \Sigma \in |\text{AlgSig}| \), let \( A \) be a \( \Sigma \)-algebra and let \( \approx_A \) be a partial \( \Sigma \)-congruence. The domain of \( \approx_A \) is defined for any sort \( s \in S \) as follows:

\[
\text{Dom}_s(\approx_A) = \{ v \mid v \approx_{s,A} v \}
\]

Definition 3.8  Let \( \Sigma = (S, \text{Op}) \) be a signature, let \( A \) be a \( \Sigma \)-algebra and let \( \approx_A \) be a partial \( \Sigma \)-congruence. For any \( s \in S \) and for any \( v \in \text{Dom}_s(\approx_A) \), the class \([v]_{\approx_A} \) is defined as follows:

\[
[v]_{\approx_A} = \{ v' \mid v' \approx_{A} v \}
\]

Definition 3.9  The quotient of an algebra \( A \in |\text{Alg}(\Sigma)| \) by a partial \( \Sigma \)-congruence \( \approx_A \) is defined as:

\[
s_A/\approx_A = \{ [v]_{\approx_A} \mid v \in s_A \text{ for every sort } s \text{ in } \Sigma \}
\]

\[
f_A/\approx_A([v_1]_{\approx_A}, \ldots, [v_n]_{\approx_A}) = [f_A(v_1, \ldots, v_n)]_{\approx_A}
\]

\[
p_A/\approx_A([v_1]_{\approx_A}, \ldots, [v_n]_{\approx_A}) \iff p_A/\approx_A(v_1, \ldots, v_n)
\]
**Definition 3.10** An equivalence relation between algebras of signature $\Sigma$ is a relation with domain included in $[\text{Alg}(\Sigma)] \times [\text{Alg}(\Sigma)]$ which is reflexive, symmetric and transitive, and it will be normally denoted by the symbol $\equiv$.

In the following, we define a list of semantic operators which are used for the definition the semantics of different operators for ASL.

**Definition 3.11** The operators on classes of models $\text{Iso}$, $\mathcal{J} \approx$, $\text{Abs}_\equiv$., $\text{Beh}_\equiv$ are formally defined as follows:

$$\text{Iso}(C) = \{ A \mid \exists B \in C. A \cong B \}$$

$$C/\approx = \{ A/\cong_A \mid A \in C \}$$

$$\text{Abs}_\equiv(C) = \{ A \mid \exists B \in C. A \equiv B \}$$

$$\text{Beh}_\equiv(C) = \{ A \mid A/\cong_A \in C \}$$

**Definition 3.12** A BASLker specification language is an ASLker specification language including the behaviour, abstract and quotient operators with syntax:

$$SP_0 ::= \text{behaviour } SP \text{ wrt } \cong \text{ (behaviour)}$$

$$\text{abstract } SP \text{ by } \equiv \text{ (abstract)}$$

$$SP/\approx \text{ (quotient)}$$

and the following semantics:

$$\text{Signature(behaviour } SP \text{ wrt } \cong ) = \text{Signature}(SP)$$

$$\text{Symbols(behaviour } SP \text{ wrt } \cong ) = \text{Symbols}(SP)$$

$$\text{Models(behaviour } SP \text{ wrt } \cong ) = \text{Beh}_\equiv(\text{Models}(SP))$$

$$\text{Signature(abstarct } SP \text{ by } \equiv ) = \text{Signature}(SP)$$

$$\text{Symbols(abstract } SP \text{ by } \equiv ) = \text{Symbols}(SP)$$

$$\text{Models(abstract } SP \text{ by } \equiv ) = \text{Abs}_\equiv(\text{Models}(SP))$$

$$\text{Signature}(SP/\approx ) = \text{Signature}(SP)$$

$$\text{Symbols}(SP/\approx ) = \text{Symbols}(SP)$$

$$\text{Models}(SP/\approx ) = \text{Iso}(\text{Models}(SP)/\cong )$$

15
where \( \cong \) and \( \equiv \) denote fixed but arbitrary family of partial Signature (SP)-congruences and equivalence relations respectively.

Finally, we relate the semantics of the behavioural operators of BASLker languages, giving first some properties on classes of \( \Sigma \)-algebras and specifications:

**Definition 3.13** A family of partial \( \Sigma \)-congruences is isomorphism compatible if for all \( \Sigma \)-algebras \( A \) and \( B \), if \( A \cong B \) then \( \cong A \equiv \cong B \).

**Definition 3.14** An equivalence relation between \( \Sigma \)-algebras is isomorphism protecting if for all \( \Sigma \)-algebras \( A \) and \( B \), \( A \equiv B \) implies \( A \cong B \).

**Definition 3.15** A family of partial \( \Sigma \)-congruences is weakly regular if for all \( A \in \text{Alg}(\Sigma) \), \( A/ \cong A \) is isomorphic to \( (A/ \equiv A)/\equiv A \).

**Definition 3.16** An equivalence relation between \( \Sigma \)-algebras (\( \equiv \)) is factorizable by a family of partial \( \Sigma \)-congruences if for all \( \Sigma \)-algebras \( A \) and \( B \), \( A \equiv B \) if and only if \( A/ \cong A \equiv B/ \cong B \).

**Proposition 3.17** If \( SP \) is closed under isomorphism, \( \cong \) is isomorphism compatible and \( \equiv \) is isomorphism protecting then behaviour \( SP \) wrt \( \cong \) and abstract \( SP \) by \( \equiv \) and \( SP/ \equiv \) are closed under isomorphism.

**Proof:**

Assume that \( SP \) is closed under isomorphism, \( \cong \) is isomorphism compatible and \( \equiv \) is isomorphism protecting.

What we have to prove for the above operators \( \text{Op}(SP) \) is that

\[
\forall A, B \in \text{Alg}(\text{Signature}(\text{Op}(SP))). \quad A \in \text{Models}(\text{Op}(SP)) \land A \equiv B \quad \Rightarrow \quad B \in \text{Models}(\text{Op}(SP))
\]

where the case of the quotient operator is obvious by definition.

1. \( \text{Op}(SP) = \text{behaviour} \ SP \) wrt \( \cong \)

Assume that \( A, B \in \text{Alg}(\text{Signature}(SP)) \).

By the definition of \( \text{Models}(\text{behaviour} \ SP \) wrt \( \cong ) \) and since \( A \in \text{Models}(\text{behaviour} \ SP \) wrt \( \cong ) \) we know that \( A/ \cong A \in \text{Models}(SP) \) and our goal is equivalent to \( B/ \equiv B \in \text{Models}(SP) \). Since \( \equiv \) is isomorphism compatible, we have that

\[
A \equiv B \Rightarrow A/ \cong A \equiv B/ \equiv B
\]

Since \( SP \) is closed under isomorphism we can derive that

\[
A/ \cong A \in \text{Models}(SP) \land A/ \equiv A \equiv B/ \equiv B \Rightarrow B/ \equiv B \in \text{Models}(SP)
\]

Using that \( A \equiv B \) and the two deduced propositions we trivially prove our goal.

16
2. $\text{Op}(SP) = \text{abstract } SP$ by $\equiv$
Assume that $A, B \in \text{Alg}(\text{Signature}(SP))$.
Since $\equiv$ is isomorphism protecting and $A \cong B$, we have that $A \equiv B$.
Since $A \equiv B$ and using $A \in \text{Models(abstract } SP \text{ by } \equiv)$, we have that $B \in \text{Models(abstract } SP \text{ by } \equiv)$

**Definition 3.18** Let ASL be an ASLker or ASLinf language. Two specifications $SP_1, SP_2 \in \text{SPEX(ASL)}$ are equal if the following holds:

$\text{Signature}(SP_1) = \text{Signature}(SP_2)$

$\forall A \in \text{Alg(Signature}(SP_1)). A \in \text{Models}(SP_1) \iff A \in \text{Models}(SP_2)$

**Theorem 3.19** For any specification expression $SP$ with signature $(S, \text{Op})$ which is closed under isomorphism, for any family of partial $\Sigma$-congruences ($\cong$) which is weakly regular and for any equivalence relation between $\Sigma$-algebras ($\equiv$) which is factorizable by $\cong$, the following equivalence between specifications holds:

$\text{abstract } SP \text{ by } \equiv = \text{behaviour } (SP/ \cong) \text{ wrt } \cong$

**Proof:**
Let $SP$ be a specification expression with signature $\Sigma$ which is closed under isomorphism, let $\cong$ be a weakly regular family of partial $\Sigma$-congruences and let $\equiv$ be an equivalence relation which is factorizable by $\cong$.
By the definition of equality of specifications, we have to prove that:

- $\text{Signature(abstract } SP \text{ by } \equiv) =$
  $\text{Signature(behaviour } SP/ \cong \text{ wrt } \cong)$ which holds since
  $\text{Signature(abstract } SP \text{ by } \equiv) =$
  $\text{Signature(behaviour } SP/ \cong \text{ wrt } \cong) = \text{Signature}(SP)$

- the following proposition holds:
  $\forall A \in \text{Alg(Signature}(SP)). A \in \text{Models(abstract } SP \text{ by } \equiv) \iff$
  $A \in \text{Models(behaviour } SP/ \cong \text{ wrt } \cong)$
Assume that $A \in \text{Alg(Signature}(SP))$ and assume that $A \in \text{Models(abstract } SP \text{ by } \equiv)$.
By the definition of $\text{Models(abstract } SP \text{ by } \equiv)$, we have that $\exists B \in \text{Models}(SP). A \equiv B$. Since $\equiv$ is factorizable by $\cong$ we have that last proposition is equivalent to

$\exists B \in \text{Models}(SP). A/\cong \cong B/\cong$

By the definition of $\text{Models(behaviour } SP \text{ wrt } \cong)$ and the definition of $\text{Models}(SP/ \cong)$ our goal is equivalent to

$A/ \cong \in \text{Iso(Models}(SP)/ \cong)$
which is true because by the definition of \( Iso(\text{Models}(SP)/\approx) \) our goal can be trivially proven equivalent to

\[
\exists B \in \text{Models}(SP). A/\approx \cong B/\approx
\]

4 Refinement of ASL specifications

As we mentioned in previous chapters, the main task of software design using the algebraic approach is the deduction of properties from algebraic specifications, the refinement of algebraic specifications and the verification of functional programs from algebraic specifications.

In this section, we give the semantics of different refinement relations between ASL specifications and some of their properties: standard refinement, behavioural refinement and abstract refinement.

**Definition 4.1 Standard refinement:** Let ASL be an ASLker specification language with an arbitrary but fixed algebraic institution AINS. Assume that \( SP, SPI \in \text{SPEX}(ASL) \). SPI is a refinement of SP (denoted by \( SP \rightsquigarrow SPI \)) if the following two conditions are satisfied:

- \( \text{Signature}(SPI) = \text{Signature}(SP) \)
- \( \text{Models}(SPI) \subseteq \text{Models}(SP) \)

**Notation:** In the following, for any refinement \( SP \rightsquigarrow SPI \) we will refer to SP as the abstract specification and SPI as the refined specification.

**Definition 4.2 Behavioural refinement:** Let ASL be an ASLker specification language with an arbitrary but fixed algebraic institution AINS. Assume that \( SP, SPI \in \text{SPEX}(ASL) \), \( A \in \text{Models}(SP) \) and assume that \( \approx \) is an isomorphism compatible family of partial \( \Sigma \)-congruences. SPI is a behavioural refinement of SP (denoted by \( SP \rightsquigarrow^\approx SPI \)) if (behaviour \( SP \) wrt \( \approx \)) \( \rightsquigarrow SPI \).

**Definition 4.3 Abstract refinement:** Let ASL be an ASLker specification language with an arbitrary but fixed algebraic institution AINS. Assume that \( SP, SPI \in \text{SPEX}(ASL) \), \( A \in \text{Models}(SP) \) and assume that \( \equiv \) is an isomorphism preserving equivalence relation between algebras in \( \text{Alg}(`\text{Signature}(SP)) \). SPI is an abstract refinement of SP (denoted by \( SP \rightsquigarrow^\equiv SPI \)) if (abstract \( SP \) by \( \equiv \)) \( \rightsquigarrow SPI \).

**Proposition 4.4 Monotonicity:** Let ASL be an ASLker specification language with an arbitrary but fixed algebraic institution AINS. If \( SP, SP', SPI, SPI' \)
$\in SP\text{EX}(\text{ASL})$, $\Sigma \in [\text{AlgSig}]_{\text{inl}}$ : $\Sigma \mapsto \text{Signature}(SP)$, $\text{inr} : \Sigma \mapsto \text{Signature}(SP')$

$SP \rightsquigarrow SPI$ and $SP' \rightsquigarrow SPI'$. Then the following holds:

$SP +\Sigma SP' \rightsquigarrow SPI +\Sigma SPI'$

$SP|\Sigma \rightsquigarrow SPI|\Sigma$

rename $SP$ by $\sigma \rightsquigarrow$ rename $SPI$ by $\sigma$

reach $SP$ with $F_\mathcal{R} \rightsquigarrow$ reach $SPI$ with $F_\mathcal{R}$

behaviour $SP$ wrt $\approx \rightsquigarrow$ behaviour $SPI$ wrt $\approx$

abstract $SP$ by $\equiv \rightsquigarrow$ abstract $SPI$ by $\equiv$

$SP/ \approx \rightsquigarrow SPI/ \approx$

**Proposition 4.5 Transitivity:** Let ASL be an ASLker specification language with an arbitrary but fixed algebraic institution $\text{AINS}$. If $SP, SPI, SPI' \in SP\text{EX}(\text{ASL})$ then if $SP \rightsquigarrow SPI$ and $SPI \rightsquigarrow SPI'$ then $SP \rightsquigarrow SPI'$

Although the behaviour and abstract implementation relation are defined in terms of the standard implementation relation, they don’t satisfy in general the main properties of the standard relation.

In [Hen97] it is studied under which conditions these properties hold. For example, for the case of transitivity of behaviour implementation of the form $SP \rightsquigarrow^\approx SPI$ and $SPI \rightsquigarrow^\approx SPI'$, $SP \rightsquigarrow^\approx SPI'$ holds if the partial congruence $\approx'$ is smaller than $\approx$ (which basically means that $\approx$ identifies more and distinguishes less) and $\approx$ is uniform. This last property requires that for any family of partial $\Sigma$-congruence $\approx'$ smaller than $\approx$ and for any algebra $A \in \text{Alg}(\text{Sig})$, $A/\approx' \cong (A/\approx_A)/(\approx_A/\approx')$.

If we always work with a fixed and concrete observational equality, then transitivity always hold. If we work with different observational equalities, it is required that the observable and input sorts of observational equalities of more concrete levels are included in the observable and input sorts of the equalities of more abstract levels.

## 5 BASLnf specification languages

In this section, we present the inductive definition of the normalisation function of the behavioural operators of BASLker specification languages using a fixed but arbitrary institution which extends algebraic institutions with several components such as a fixed but arbitrary family of partial congruences, a behavioural satisfaction relation and different functions which are used to define the normal forms of the behavioural operators. We will refer to these institutions as behavioural algebraic institutions. We also generalise the semantics of...
the behaviour operator with higher-order logic as specification logic of [HS96] using also behavioural algebraic institutions and we relate this semantics with the semantics of the behaviour operator of BASLker specification languages.

In [Hen97], the normalisation function of the behavioural operators with infinitary first-order logic as specification logic is presented. The equivalences of the behaviour and abstract operator are the observational and behavioural equality which are presented in the last section of this chapter. The normalisation functions of the behavioural and quotient operator are defined as the normalisation functions of structured specifications, the semantics of which are equivalent to the semantics of the behaviour and quotient operator.

In [HS96], the semantics of the behavioural operator is given in terms of a behavioural satisfaction relation which is denoted as $\models^\approx$ where $\approx$ is a fixed but arbitrary family of partial congruences and they use higher-order logic as specification logic.

They also define a relativization function which we will refer as $brcl$ relating standard and behavioural satisfaction in the following way:

$$\forall \Sigma \in \text{AlgSig}, \forall A \in \text{Alg}(\Sigma), \forall \phi \in \text{Sen}_{\text{HOL}}(\Sigma).$$

$$A \models^\approx \phi \iff A \models^{\text{brcl}}(\phi)$$

The semantics of the behavioural operator is defined using the behavioural satisfaction relation in the following way:

$$\text{Models(behaviour) } < \Sigma, \Phi > \text{ wrt } \approx = \{ A \in \text{Alg}(\Sigma) \mid A \models^\approx \Phi \}$$

This operator can only be applied to base specifications and because of the relation between behavioural and standard satisfaction, the semantics of the behavioural operator can also be defined as:

$$\text{Models(behaviour) } < \Sigma, \Phi > \text{ wrt } \approx = \{ A \in \text{Alg}(\Sigma) \mid A \models^{\text{brcl}}(\Phi) \}$$

Since they also prove that

$$\forall \Sigma \in \text{AlgSig}, \forall A \in \text{Alg}(\Sigma), \forall \phi \in \text{Sen}_{\text{HOL}}(\Sigma).$$

$$A/\approx \models \phi \iff A \models^\approx \phi$$

the relativization function $brcl$ also satisfies the following condition:

$$\forall \Sigma \in \text{AlgSig}, \forall A \in \text{Alg}(\Sigma), \forall \phi \in \text{Sen}_{\text{HOL}}(\Sigma).$$

$$A/\approx \models \phi \iff A \models^{\text{brcl}}(\phi)$$

In order to define the normalisation function of the behavioural operators of BASLker languages for an arbitrary algebraic institution and arbitrary equivalence relations, we have decided to define them in terms of the normal form of the argument specification of the behavioural operators. The definition of the
normal form will use functions on sentences based on the idea of relativization
function presented in [HS96] and above for the behaviour and quotient operator.
The normal form of the abstract operator will be defined as the normal form
of the behaviour of the quotient of the argument specification of the abstract
operator. It follows that this is the normal form of the abstract operator using
theorem 4.3.24 as in [Hen97].

The definition of the relativization functions for the normalisation functions
of $BASL_ker$ languages will need in general to extend the original signature of
the specification with extra symbols. For example, an alternative definition of
the normalisation function of the behaviour operator for the institution $HOL$
and for an observational equality is based on [Hen97], which, as we mentioned
above, proves the equivalence between the semantics of the behaviour operator
with the semantics of a structured specification. The symbols of the structured
specification extends the symbols of the behavioural operator with, for example,
a disjoint copy of the signature of the behaviour operator and symbols to denote
the observational equality. A possible definition of the relativization functions
uses the same symbols as the symbols of the structured specifications. The condi-
tions which must satisfy these relativization functions are more complicated
than the condition which satisfies the relativisation function of [HS96] presented
below.

Thus, we present in this section an institution which extends algebraic institu-
tions with a behavioural satisfaction relation and two different functions on
sentences for any inclusion $i : \Sigma \hookrightarrow \Sigma'$ in $AlgSig$. These institutions will
be referred as behavioural algebraic institution. One of this kind of functions will
be denoted by $brel[i, bi[\Sigma]]$ where $i$ is an inclusion with arity $i : \Sigma \hookrightarrow \Sigma'$ and
$bi[\Sigma]$ is an inclusion with arity $bi[\Sigma] : \Sigma' \hookrightarrow \Sigma''$ and these functions will be used
to define the normalisation functions of the behaviour operator. These functions
also can be used to define the normal form of the generalized semantics of the
behaviour operator presented in [HS96] extended to structured specifications. See subsection 4.6.4 for a concrete example of a definition of the inclusion
$bi[\Sigma]$ and the function $brel[i, bi[\Sigma]]$ in a behavioural algebraic institution with
higher-order logic as algebraic institution.

The inclusion $i$ of the function $brel[i, bi[\Sigma]]$ used in the normalisation function
of the behaviour operator $\text{behaviour } SP \ wrt \approx$ of $BASL_ker$ specifi-
cation languages will have arity $i : \text{Signature}(SP) \hookrightarrow \text{Symbols}_{n_f}(SP)$ whereas
the inclusion of the function $brel$ used in the behaviour operator of [HS96] will
be the identity $i : \text{Symbols}_{n_f}(SP) \hookrightarrow \text{Symbols}_{n_f}(SP)$.

The other kind of functions will be defined the normalisation function of
the semantics of the quotient operator and it will be denoted as $qrel[i, qi[\Sigma]]$
where $i$ is an inclusion with arity $i : \Sigma \hookrightarrow \Sigma'$ and $qi[\Sigma]$ is an inclusion with arity
$qi[\Sigma] : \Sigma' \hookrightarrow \Sigma''$.

The conditions which must satisfy these two kind of functions are presented
in the general definition of behavioural algebraic institutions ($BAINS$). See
next section for proofs that these concrete functions satisfy the general condi-
tions which are defined in $BAINS$. In this section, we present the definition
of $BAINS$, the alternative and generalised version of the semantics of the be-
haviour operator of [HS96] and how to relate it with \textit{BASLker} languages.

**Definition 5.1** A behavioural algebraic institution (\textit{BAINS}) consists of an algebraic institution and additionally the following 6 components:

- For any signature \( \Sigma \in |\text{AlgSig}| \), a fixed but arbitrary family of partial \( \Sigma \)-congruences \( \approx_A \subseteq \text{Alg}(|\Sigma|) \).
- For each signature \( \Sigma \subseteq |\text{AlgSig}| \) a behavioural satisfaction relation \( \approx_{\text{BALS,} \Sigma}^\phi \subseteq \text{Alg}(|\Sigma|) \times \text{Sen}_{\text{BAINS}}(|\Sigma|) \), which is related to the standard satisfaction by the behavioural satisfaction condition. This condition is defined as:

\[
\forall A \in \text{Alg}(|\Sigma|). \forall \phi \in \text{Sen}_{\text{BAINS}}(|\Sigma|). A/ \approx_{\text{BALS,} \Sigma} \phi \iff
A \models_{\text{BALS,} \Sigma} \phi
\]

- For each inclusion \( i : \Sigma \hookrightarrow \Sigma' \) in \( |\text{AlgSig}| \), an inclusion \( b_i[\Sigma] : \Sigma' \hookrightarrow \Sigma'' \) and a function

\[
\text{brel}[\Sigma, b_i[\Sigma]] : \mathcal{P}(\text{Sen}_{\text{BAINS}}(|\Sigma'|)) \rightarrow \mathcal{P}(\text{Sen}_{\text{BAINS}}(b_i[\Sigma]([\Sigma'])))
\]

which satisfies the following conditions which we will refer as the behavioural realizability conditions:

(1) \( \forall A' \in |\text{Alg}(|\Sigma'|)|. \forall \Phi \in \mathcal{P}(\text{Sen}_{\text{BAINS}}(|\Sigma'|)). \forall A \in |\text{Alg}(|\Sigma|)|.

\[
A'\rvert_{\Sigma'} = A/\approx \land A' \models \Phi \Rightarrow
\exists A'' \in |\text{Alg}(b_i[\Sigma]([\Sigma'])). A''\rvert_{\Sigma'} = A'' \land A'' \models \text{brel}[i, b_i[\Sigma]](\Phi)
\]

for a given \( \Sigma'\)-algebra \( A'' \) is defined as follows:

- \( A''\rvert_{\Sigma} = A \)
- \( A''_s = A'_s \) for any sort \( s \in \text{Sorts}(\Sigma') \cap \text{Sorts}(\Sigma) \)
- \( f_{A''} = f_{A'} \) for any sort \( f \in \text{Ops}(\Sigma') \cap \text{Ops}(\Sigma) \)
- \( P_{A''} = P_{A'} \) for any sort \( P \in \text{Pr}(\Sigma') \cap \text{Pr}(\Sigma) \)

and

(2) \( \forall A'' \in |\text{Alg}(b_i[\Sigma]([\Sigma'])). \forall \Phi \in \mathcal{P}(\text{Sen}_{\text{BAINS}}(|\Sigma'|)).

\[
A'' \models \text{brel}[i, b_i[\Sigma]](\Phi) \Rightarrow \exists A' \in |\text{Alg}(|\Sigma'|). A'\rvert_{\Sigma} = A''\rvert_{\Sigma}/\approx \land A' \models \Phi
\]

- For each inclusion \( i : \Sigma \hookrightarrow \Sigma' \) in \( |\text{AlgSig}| \), an inclusion \( q_i[\Sigma] : \Sigma' \hookrightarrow \Sigma'' \) and a function

\[
\text{qrel}[i, q_i[\Sigma]] : \mathcal{P}(\text{Sen}_{\text{BAINS}}(|\Sigma'|)) \rightarrow \mathcal{P}(\text{Sen}_{\text{BAINS}}(q_i[\Sigma]([\Sigma'])))
\]
which satisfies the following conditions which we will refer as the quotient relativization conditions:

(1) \( \forall A' \in |\text{Alg}(\Sigma')|, \forall \Phi \in \mathcal{P}(\text{Sen}_{\text{BAINS}}(\Sigma')), A' \models_{\Sigma} \Phi \Rightarrow \exists A'' \in \text{Alg}(q_i[\Sigma](\Sigma')), A''|_{\Sigma} \supseteq A'|_{\Sigma}/\approx \land A'' \models_{q_i[\Sigma](\Sigma')} \text{rel}[i, q_i[\Sigma]](\Phi) \)

(2) \( \forall A'' \in \text{Alg}(q_i[\Sigma](\Sigma')), \forall \Phi \in \mathcal{P}(\text{Sen}_{\text{BAINS}}(\Sigma')), A'' \models_{q_i[\Sigma](\Sigma')} \text{rel}[i, q_i[\Sigma]](\Phi) \Rightarrow \exists A' \in |\text{Alg}(\Sigma')|, A'|_{\Sigma} \supseteq A''|_{\Sigma}/\approx \land A' \models_{\Sigma, BAINS} \Phi \)

Remark: If the inclusion \( i : \Sigma \rightarrow \Sigma \) then the first behavioural relativization condition can be rewritten to:

\( \forall A \in |\text{Alg}(\Sigma)|, \forall \Phi \in \mathcal{P}(\text{Sen}_{\text{BAINS}}(\Sigma)), A/\approx \models \Phi \Rightarrow \exists A' \in |\text{Alg}(b_i[\Sigma](\Sigma'))|, A'|_{\Sigma} = A \land A' \models b\text{rel}[i, b_i[\Sigma]](\Phi) \)

Definition 5.2 A BASLnf specification language is an ASLnf specification language over a behavioural algebraic institution BAINS with additionally the behaviour, abstract and quotient operator with the same additional semantic conditions as in BASLker and additionally the following conditions:

Let \( \text{nf}(SP) \) be \( < \Sigma', \Phi > |\Sigma \). Then:

\[ \text{nf(behaviour \ SP \ wrt} \ \approx) = \]

\[ < \text{Symbols}_{nf}(\text{behaviour \ SP \ wrt} \ \approx), \text{brel}[i, b_i[\Sigma]](\Phi) > |\Sigma \]

\[ \text{Symbols}_{nf}(\text{behaviour \ SP \ wrt} \ \approx) = b_i[\Sigma](\Sigma') \]

\[ \text{nf(abstract \ SP \ by} \ \equiv) = \text{nf(behaviour \ SP/} \ \approx \ \text{wrt} \ \approx) \]

\[ \text{Symbols}_{nf}(\text{abstract \ SP \ by} \ \equiv) = \]

\[ \text{Symbols}_{nf}(\text{behaviour \ SP/} \ \approx \ \text{wrt} \ \approx) \]

\[ \text{nf}(SP/ \ \approx) = < \text{Symbols}_{nf}(SP/ \ \approx), \text{qrel}[i, q_i[\Sigma]](\Phi) > |\Sigma \]

\[ \text{Symbols}_{nf}(\text{nf}(SP/ \ \approx)) = q_i[\Sigma](\Sigma') \]

where \( \text{brel}[i, b_i[\Sigma]] : \text{Sen}_{\text{BAINS}}(\Sigma') \rightarrow \text{Sen}_{\text{BAINS}}(b_i[\Sigma](\Sigma')) \) is a function which satisfies the behavioural relativization conditions where \( i \) has arity \( i : \Sigma \hookrightarrow \Sigma' \)
and \( qrel[i,qi[\Sigma]] : \text{Sen}_{BAINS}(\Sigma') \to \text{Sen}_{BAINS}(qi[\Sigma][\Sigma']) \) is a function which satisfies the quotient relativization conditions.

**Theorem 5.3** \( BASInf \) is an \( ASInf \) specification language.

**Proof:**
The proof is by induction on specification expressions. Let \( bchop \) denote any of the three behavioural operators. It is trivial to show that

\[
\text{Signature}(bchop(SP)) = \text{Signature}(nf(bchop(SP)))
\]

and that there exists an inclusion between \( \Sigma' \) and \( \text{Symbols}_{nf}(SP) \) for all the behavioural operators. Besides, we have to show that

\[
A \in bchop(SP) \iff A \in nf(bchop(SP))
\]

for every behavioural operator.

We assume that \( nf(SP) = < \Sigma', \Phi > |_\Sigma \) where \( \Sigma' = \text{Symbols}_{nf}(SP) \) and \( \Sigma = \text{Signature}(SP) \) for every argument specification \( SP \) of any behavioural operator \( bchop(SP) \).

- \( \text{behaviour}_{nf} \ SP \ \text{wrt} \ \approx \)

By the definition of the behaviour operator we know that:

\[
A \in \text{Models}(\text{behaviour} \ SP \ \text{wrt} \ \approx) \iff A/\approx \in \text{Models}(SP)
\]

By the induction hypotheses, \( A/\approx \in \text{Models}(SP) \) can be rewritten to:

\[
\exists A' \in |\text{Alg}(\Sigma')|.A'|_\Sigma = A/\approx \land A' \models \Phi
\]

Let \( A' \) be the \( \Sigma' \)-algebra such that

\[
A'|_\Sigma = A/\approx \land A' \models \Phi
\]

By the first behavioural relativization condition of the function \( brel[i,bi[\Sigma]] \), we can deduce that

\[
\exists A'' \in \text{Alg}(bi[\Sigma][\Sigma']).A''|_\Sigma = A'' \land A'' \models brel[i,bi[\Sigma]][\Phi]
\]

where \( A'' \) is defined as:

\[
A''|_\Sigma = A \\
A'' = A' \text{ for any sort } s \in \text{Sorts}(\Sigma') - \text{Sorts}(\Sigma) \\
f_{A''} = f_{A'} \text{ for any sort } f \in \text{Ops}(\Sigma') - \text{Ops}(\Sigma) \\
\Pi_{A''} = \Pi_A \text{ for any sort } P \in \text{Pr}(\Sigma') - \text{Pr}(\Sigma)
\]

And therefore we have that:

\[
A \in \text{Models}(< bi[\Sigma][\Sigma'],brel[i,bi[\Sigma']][\Phi]> |_\Sigma)
\]

24
and therefore $A \in Models(nf(\text{behaviour SP wrt } \approx))$.

By the definition of $Models(nf(\text{behaviour SP wrt } \approx))$ we know that

$$\exists A' \in Alg(\text{bi}[\Sigma](\Sigma')). A'|\Sigma = A \land A' \models_{\Sigma} \text{brel}[i, qi[\Sigma]](\Phi)$$

Let $A'$ be a $\Sigma'$-algebra such that $A'|\Sigma = A$ and $A' \models_{\Sigma} \text{brel}[i, qi[\Sigma]](\Phi)$. By the second condition of the behavioural relativization function we can deduce that

$$\exists A'' \in |Alg(\Sigma')|. A''|\Sigma = A/\approx \land A'' \models \Phi$$

and by the induction hypotheses we have that

$$A \in Models(\text{behaviour SP wrt } \approx)$$

$\cdot$ \ $SP/nf \approx$:

$\Rightarrow$)

By the definition of $Models(SP/ \approx)$, we know that

$$A \in Models(SP/ \approx) \Leftrightarrow \exists A' \in |Alg(\Sigma)|. A \equiv A'/\approx \land A' \in Models(SP)$$

By the induction hypotheses, we can transform the right hand side part of the previous proposition to:

$$\exists A'' \in |Alg(\Sigma')|. A''|\Sigma = A' \land A \equiv A'/\approx \land A'' \models \Phi$$

Let $A''$ be a $\Sigma'$-algebra such that

$$A''|\Sigma = A' \land A \equiv A'/\approx \land A'' \models \Phi$$

By the condition of quotient relativization function we can deduce that

$$\exists A'''' \in Alg(qi[\Sigma](\Sigma')). A''''|\Sigma \equiv A'/\approx \land$$

$$A'''' \models_{\Sigma', BAINS} \text{qrel}[i, qi[\Sigma]](\Phi)$$

and therefore $A \in Models(nf(SP/ \approx))$

$\Leftarrow$):

By the definition of $Models(nf(SP/ \approx))$ we know that

$$\exists A' \in Alg(qi[\Sigma](\Sigma')). A'|\Sigma = A \land A' \models_{\Sigma} \text{qrel}[i, qi[\Sigma]](\Phi)$$

Let $A'$ be a $\Sigma'$-algebra such that $A'|\Sigma = A$ and $A' \models_{\Sigma} \text{qrel}[i, qi[\Sigma]](\Phi)$. By the second condition of the quotient relativization function we can deduce that

$$\exists A'' \in |Alg(\Sigma')|. A'|\Sigma \equiv A'/\approx \land A'' \models_{\Sigma', BAINS} \Phi$$

and by the induction hypotheses we have that $A \in Models(SP/ \approx)$. 

25
abstract \( SP \) by \( \equiv \): We know by theorem 4.3.24 that
\[\text{abstract } SP \text{ by } \equiv = \text{behaviour } SP/ \approx \text{ wrt } \equiv.\]
Since we know by induction hypotheses that
\[A \in Models(nf(SP)) \iff A \in Models(SP),\]
by the induction condition of the quotient operator we know that \( SP/ \approx = nf(SP/ \approx) \) and by the induction condition of the behavioural operator we know that
\[\text{behaviour } SP/ \approx \text{ wrt } \approx = nf(\text{abstract } SP \text{ by } \equiv).\]
Thus abstract \( SP \) by \( \equiv = nf(\text{abstract } SP \text{ by } \equiv).\)

Finally, we relate the semantics of the behaviour operator of \( BASLnf \) languages with the generalisation of the semantics of this operator given in [HS96].

In the section of further work of [HS96], general lines are given to define the semantics of this operator for structured specifications. The \( Models \) function is defined using an auxiliary function as follows:
\[Models(\text{behaviour } SP \text{ wrt } \approx) = Mod_A(SP)\]
and the auxiliary function \( Mod_A \), following the underlying ideas of [HS96], can be defined for the common operators of \( ASLnf \) languages and for an arbitrary but fixed behavioural institution \( BAINS \) as follows:
\[Mod_A(<\Sigma, \Phi>) = \{ A \in Alg(\Sigma) | A \models \Phi \}\]
\[Mod_A(SP_1 +_\Sigma SP_2) = \{ A | A \in Alg(\text{Signature}(SP_1) +_\Sigma \text{Signature}(SP_2)), A|_{nt} \in Mod_A(SP_1), A|_{nr} \in Mod_A(SP_2) \}\]
\[Mod_A(SP|_\Sigma) = \{ A|_\Sigma | A \in Mod_A(SP) \}\]

Note that the \( Mod_A \) is not defined for the behavioural operator.

An alternative possible way to give semantics to the behaviour operator in \( ASLnf \) languages which is equivalent to the previous extension under the same
syntactic restrictions is as follows:

\[
\text{Signature}(\text{behaviour}_{nf} \ SP \ \text{wrt} \ \approx) = \text{Signature}(SP)
\]

\[
\text{Symbols}(\text{behaviour}_{nf} \ SP \ \text{wrt} \ \approx) = \text{Symbols}(SP)
\]

\[
\text{Models}(\text{behaviour}_{nf} \ SP \ \text{wrt} \ \approx) = \{ A \in \text{Alg}(\Sigma') \mid A \models \Phi \}
\]

\[
nf(\text{behaviour}_{nf} \ SP \ \text{wrt} \ \approx) = < \text{Symbols}_{nf}(\text{behaviour}_{nf} \ SP \ \text{wrt} \ \approx),
\]

\[
\text{brcl}[i, bi[\Sigma']](\Phi) > |\Sigma|
\]

\[
\text{Symbols}_{nf}(\text{behaviour}_{nf} \ SP \ \text{wrt} \ \approx) = bi[\Sigma'](\Sigma')
\]

where \( nf(SP) = < \Sigma', \Phi > |\Sigma, \)

\[
\text{brcl}[i, bi[\Sigma']]: \mathcal{P}(\text{Sen}_{BAISS}(\Sigma')) \rightarrow \mathcal{P}(\text{Sen}_{BAISS}(bi[\Sigma'](\Sigma')))
\]

is a function which satisfies the behavioural relativization condition, and \( i \) has arity \( i: \Sigma' \hookrightarrow \Sigma'. \)

One way to see that this alternative semantics is equivalent to the one presented in [HS96] is to define the ASLn language ASLN with just the common operators of ASLn languages and prove the following proposition for any \( SP \in SP\chi(ASLN): \)

\[
\forall A \in \text{Models}(SP), A \in \text{Mod}_{nf}(SP) \iff
\]

\[
A \in \text{Models}(\text{behaviour}_{nf} \ SP \ \text{wrt} \ \approx)
\]

which follows trivially by the behavioural relativization conditions. Finally, we can relate our alternative semantics with the semantics of the behaviour operator of BASLker languages with the following theorem:

**Theorem 5.4** Let BAINS be a behavioural algebraic institution. Let BSPL\(_1\) be a BASLn specification language over BAINS and let BSPL\(_2\) be the ASLn language with additionally the behaviour operator with the alternative semantics presented below and the same institution BAINS. Let \( SP_1 \) be a specification expression of BSPL\(_1\) and let \( SP_2 \) be a specification expression of BSPL\(_2\) such that

\[
A \in \text{Models}(SP_1) \implies A \in \text{Models}(SP_2).
\]

\[
nf(SP_1) = nf(SP_2) = < \Sigma', \Phi > |\Sigma
\]

27
Under these assumptions the following holds:

\[ A \in \text{Models}(\text{behaviour}_{SP_2} \text{ wrt } \approx) \Rightarrow A \in \text{Models}(\text{behaviour}_{SP_1} \text{ wrt } \approx) \]

**Proof:**

Assume that \( A \in \text{Models}(\text{behaviour}_{SP_2} \text{ wrt } \approx) \). By the definition of the semantics of this behavioural operator, the previous proposition is equivalent to

\[ \exists A' \in \text{Alg}(bi[\Sigma']([\Sigma'])), A'|\Sigma = A \land A' \models_{brel[i, bi[\Sigma']]}(\Phi) \]

By the behavioural relativization condition we can deduce that \( A'/\approx \models \Phi \) and by the behavioural satisfaction condition we can deduce that \( A' \models_{\approx} \Phi \). Since \( A'/\Sigma = A \) we have that \( A \in \text{Models}(\text{behaviour}_{SP_1} \text{ wrt } \approx) \).

Note that the left implication of the propositions which relate the class of models of the behavioural operators doesn’t seem to hold since from our point of view it is necessary that an equivalent or more general condition to the following one

\[ \forall A'' \in |\text{Alg}(\Sigma'')|, \forall \Phi \in \text{Scn}_{BAINS}(\Sigma''), \exists A \in |\text{Alg}(\Sigma)|, A''|\Sigma = A/\approx \land A'' \models_{\Sigma, BAINS} \Phi \Rightarrow \]

\[ \exists A' \in |\text{Alg}(\Sigma')|, A''|\Sigma = A'/\approx \land A'' \models_{\Sigma', BAINS} \Phi \]

must be satisfied for any pair of inclusions \( i : \Sigma \hookrightarrow \Sigma', i : \Sigma' \hookrightarrow \Sigma'' \) of a fixed but arbitrary behavioural algebraic institutions and we have not succeeded to find it for example for the institution of this kind with a higher-order logic presented in the next section.

6 **B HOL: A behavioural algebraic institution**

In this section, we present a behavioural algebraic institution which extends the algebraic institution \( HOL \) presented also in this chapter. The family of partial congruences of this concrete behavioural algebraic institution will be the observational equality which we described in the chapter of type theory and it will be presented in this section in a more general way.

First, we present the definition of the behavioural satisfaction relation for an arbitrary but fixed family of partial congruences as in [HS96]. Then, we define the observational equality for first-order relational signatures which will be referred to as relational observational equality and for first-order signatures which will be referred to as observational equality, but in both cases they will be denoted by the same symbol. The formulation of the observational equality is the same as in [BHW95] or [Hen97].

Finally, we define the relativisation functions for the institution \( HOL \) and we present the behavioural algebraic institution \( B HOL \).
6.1 The behavioural satisfaction relation

In order to define the behavioural satisfaction relation for the algebraic institution \( \text{HOL} \) it is necessary to extend the given fixed but arbitrary family of partial \( \Sigma \)-congruences to higher-order types. We present this extension just for the general case because the instantiation to the relational observational equality is obvious.

**Definition 6.1** Let \( \approx \) be a family of partial \( \Sigma \)-congruences. The relation \( \approx_{A,[\tau_1,\ldots,\tau_n]} \): 
\[
[\tau_1, \ldots, \tau_n]_A \times [\tau_1, \ldots, \tau_n]_A \text{ for any } \tau_i \in \text{Types}_{\text{HOL}}(\Sigma), \ldots, \tau_n \in \text{Types}_{\text{HOL}}(\Sigma) \text{ and for any } A \in |\text{Alg}(\Sigma)| \text{ is defined as follows:}
\]
\[
p \approx_{A,[\tau_1,\ldots,\tau_n]} p' \iff \forall v_1, v'_1 \in [\tau_1]_A \ldots \forall v_n, v'_n \in [\tau_n]_A.
\]
\[
equiv{A,\tau_i} v_1 \wedge \ldots \wedge \equiv{A,\tau_i} v'_n \Rightarrow ((v_1, \ldots, v_n) \in p \iff (v'_1, \ldots, v'_n) \in p')
\]

**Definition 6.2** For any relational signature \( \Sigma \in |\text{AlgSig}| \), for any sort \( s \in \text{Sorts}(\Sigma) \), for any \( \Sigma \)-algebra \( A \), a value \( v \in A_s \) respects a partial \( \Sigma \)-congruence if \( v \approx_A v \). A predicate \( p \in [\tau_1, \ldots, \tau_n]_A \) respects \( \approx_{A,[\tau_1,\ldots,\tau_n]} \) for any \( \tau_i \in \text{Types}_{\text{HOL}}(\Sigma), \ldots, \tau_n \in \text{Types}_{\text{HOL}}(\Sigma) \) if the following condition holds:
\[
\forall v_1, v'_1 \in [\tau_1]_A \ldots \forall v_n, v'_n \in [\tau_n]_A.
\]
\[
equiv{A,\tau_i} v_1 \wedge \ldots \wedge \equiv{A,\tau_i} v'_n \Rightarrow ((v_1, \ldots, v_n) \in p \iff (v'_1, \ldots, v'_n) \in p')
\]

**Proposition 6.3** \( \approx_{A,\tau} \) is a partial equivalence relation for any \( \tau \in \text{Types}_{\text{HOL}}(\Sigma) \).

**Definition 6.4** The semantic function \( [\tau]^\approx_A \) is inductively defined for any type \( \tau \in \text{Types}_{\text{HOL}}(\Sigma) \) and for any \( \Sigma \)-algebra \( A \) as follows:
\[
[\tau]^\approx_A = \{ v \in A_s \mid v \text{ respects } \approx \}
\]
\[
[[\tau_1, \ldots, \tau_n]]^\approx_A = \{ p \in \mathcal{P}([\tau_1]^\approx_A \times \ldots \times [\tau_n]^\approx_A) \mid p \text{ respects } \approx \}
\]

**Notation:** The semantics of \( \text{Prop} \) is a set of two elements: the empty set and the set with the empty tuple. These two elements will be denoted as \( \textbf{ff} \) and \( \textbf{tt} \) respectively.

**Definition 6.5** The function \( [t]^\approx_A \) for any term \( t \in \text{Terms}_{\text{HOL}}(\Sigma,X_{\text{HOL}}) \), for any algebra \( A \in \text{Alg}(\Sigma) \), for any \( \text{Types}_{\text{HOL}}(\Sigma) \)-sorted valuation \( p \) which
for every \( \tau \in \text{Types}_{\text{HOL}}(\Sigma) \), \( \rho_\tau \) has arity \( \rho_\tau : X_{\text{HOL}, \tau} \rightarrow [\tau]_A^S \) is inductively defined by the structure of \( t \in \text{Terms}(\Sigma, X) \) as follows:

\[
[x_t]_{\rho_\tau, A} = \rho_\tau(x)
\]

\[
[f(t_1, \ldots, t_n)]_{\rho_\tau, A} = f_A([t_1]_{\rho_\tau, A}, \ldots, [t_n]_{\rho_\tau, A})
\]

\[
[p(t_1, \ldots, t_n)]_{\rho_\tau, A} = \text{if } ([t_1]_{\rho_\tau, A}, \ldots, [t_n]_{\rho_\tau, A}) \in p_A \text{ then } \text{tt} \text{ else } \text{ff}
\]

\[
[\lambda(x_1 : \tau_1, \ldots, x_n : \tau_n).\phi]_{\rho_\tau, A} = \\
\{ (v_1, \ldots, v_n) | v_1 \in [\tau_1]_{\rho_\tau, A}, \ldots, v_n \in [\tau_n]_{\rho_\tau, A}, [\phi]_{\rho_\lambda \{ (x_1, v_1), \ldots, (x_n, v_n) \}} = \text{tt} \}
\]

\[
[t \ (t_1, \ldots, t_n)]_{\rho_\tau, A} = \\
\text{if } ([t_1]_{\rho_\tau, A}, \ldots, [t_n]_{\rho_\tau, A}) \in [t]_{\rho_\tau, A} \text{ then } \text{tt} \text{ else } \text{ff}
\]

\[
[\phi \supset \phi']_{\rho_\tau, A} = \text{if } [\phi]_{\rho_\tau, A} = \text{tt} \text{ then } [\phi']_{\rho_\tau, A} \text{ else } \text{tt}
\]

\[
[\forall x : \tau.\phi]_{\rho_\tau, A} = \text{if } \forall v \in [\tau]_{\rho_\tau, A}, [\phi]_{\rho_\forall \{ (x, v) \}, A} = \text{tt} \text{ then } \text{tt} \text{ else } \text{ff}
\]

**Definition 6.6** For each \( \Sigma \in [\text{AlgSig}] \), for all \( A \in [\text{Alg}(\Sigma)] \), for all \( \phi \in \text{Sen}_{\text{HOL}}(\Sigma) \), the satisfaction relation \( A \models \Sigma \phi \) holds if and only if for any \( \text{Types}_{\text{HOL}}(\Sigma) \)-sorted valuation \( \rho \) which for every \( \tau \in \text{Types}_{\text{HOL}}(\Sigma) \), \( \rho_\tau \) has arity \( \rho_\tau : X_{\text{HOL}, \tau} \rightarrow [\tau]_A^S \) is

6.2 The observational equality

In this subsection, we present the observational equality together with a behavioural equality between algebras which is factorizable by the observational equality.

**Definition 6.7** Given a signature \( \Sigma = (S, \text{Op}) \in [\text{AlgSig}] \) and a set of sorts \( \text{In} \subseteq \text{Sorts}(\Sigma) \), \( \Sigma \) is sensible wrt \( \text{In} \) if for all \( s \in \text{Sorts}(\Sigma) - \text{In} \), there exists a term \( t \) of sort \( s \) built with function symbols in \( \Sigma \) and variables of sort \( s \in \text{In} \).

**Definition 6.8** Let \( \Sigma = (S, \text{Op}) \) be a signature in \([\text{AlgSig}]\), let \( \text{In} \) and \( \text{Obs} \) be two set of sorts s.t. \( \text{In}, \text{Obs} \subseteq \text{Sorts}(\Sigma) \) and let \( X_{\text{In}} \) be an In-set of sort of variables. The \( \text{Sorts}(\Sigma) \)-sorted set of contexts \( C_{\Sigma, \text{Obs}}(X_{\text{In}}) \) is defined for each sort \( s \) as the set of terms \( T_{\Sigma}(X_{\text{In}} \cup z_s) \) of result sort in \( \text{Obs} \) such that \( z_s \) is a free variable which satisfies the condition \( \{ z_s \} \cap X_s = \emptyset \). This set is also denoted as \( C_{\Sigma, \text{Obs}}(X_{\text{In}}, z_s) \) for every sort \( s \in \text{S} \).

**Definition 6.9** Let \( \Sigma = (S, \text{Op}) \) be a signature in \([\text{AlgSig}]\), let \( \text{Obs} \) and \( \text{In} \) be two set of sorts s.t. \( \text{Obs}, \text{In} \subseteq \text{Sorts}(\Sigma) \) and \( \text{In} \) is sensible wrt \( \Sigma \). Let \( A \) be
a $\Sigma$-algebra. The observational equality $\approx_{A,\text{obs}}^{\text{obs}^+}$ is formally defined for each sort $s$ and for each $v, w \in A[X_{tn}]$, as follows:

$$v \approx_{A,\text{obs}}^{\text{obs}^+} w \iff$$

$$\forall \alpha \in C_{\Sigma,\text{obs}}(X_{tn}, z) \forall \alpha \in X_{tn} \rightarrow A[X_{tn}], \quad I_{a \cup \{(z, v)\}}(\epsilon) = I_{a \cup \{(z, w)\}}(\epsilon) \quad \text{if } s \in S - \text{Obs}$$

$$v = w \quad \text{if } s \in \text{Obs}$$

**Proposition 6.10** Let $\Sigma = (S, Op)$ be a signature in $\text{AlgSig}$, let $\text{Obs}$ and $\text{In}$ be two sets of sorts s.t. $\text{Obs, In} \subseteq \text{Sorts}(\Sigma)$. The observational equality $\approx_{A,\text{obs}}^{\text{obs}^+}$ is a family of partial $\Sigma$-congruences.

**Proposition 6.11** Let $\Sigma = (S, Op)$ be a signature in $\text{AlgSig}$, let $\text{Obs}$ and $\text{In}$ be two sets of sorts s.t. $\text{Obs, In} \subseteq \text{Sorts}(\Sigma)$ and let $A$ be a $\Sigma$-algebra. The observational equality $\approx_{A,\text{obs}}^{\text{obs}^+}$ is weakly regular.

**Definition 6.12** Let $\Sigma = (S, Op)$ be a relational signature in $\text{AlgSig}$, let $\text{In}$ and $\text{Obs}$ be two sets of sorts s.t. $\text{In, Obs} \subseteq \text{Sorts}(\Sigma)$ and let $X_{tn}$ be an $\text{In}$-sorted set of variables. The behavioural equality between $\Sigma$-algebras $\equiv_{\text{obs}, tn}$ is formally defined as:

$$A \equiv_{\text{obs}, tn} B \iff \forall t, r \in T_{\Sigma}(X_{tn}), \forall \alpha \in X_{tn} \rightarrow A, \forall \beta \in X_{tn} \rightarrow B, \quad I_{a}(t) = I_{a}(r) \iff I_{b}(t) = I_{b}(r)$$

**Proposition 6.13** Let $\Sigma = (S, Op)$ be a signature in $\text{AlgSig}$, and let $\text{In}$ and $\text{Obs}$ be two sets of sorts s.t. $\text{In, Obs} \subseteq \text{Sorts}(\Sigma)$, $\equiv_{\text{obs}, tn}$ is an equivalence relation.

**Proposition 6.14** Let $\Sigma = (S, Op)$ be a signature in $\text{AlgSig}$, let $\text{In}$ and $\text{Obs}$ be two sets of sorts s.t. $\text{In, Obs} \subseteq \text{Sorts}(\Sigma)$ and let $A$ be a $\Sigma$-algebra. $\equiv_{\text{obs}, tn}$ is factorizable by $\approx_{A,\text{obs}}^{\text{obs}^+}$.

### 6.3 The relational observational equality

In this subsection, we extend the observational equality to relational signatures giving also a relational behavioural equality which is factorizable by the relational observational equality.

**Definition 6.15** Let $\Sigma$ be a relational signature in $\text{AlgSig}$, let $\text{In}$ and $\text{Obs}$ be two sets of sorts s.t. $\text{In, Obs} \subseteq \text{Sorts}(\Sigma)$ and let $X_{tn}$ be an $\text{In}$-sorted set of variables. The $\text{Sorts}(\Sigma)$-sorted set of contexts $PC_{\Sigma,\text{obs}}(X_{tn})$ is defined for each
sort $s$ as the set of atoms $P_s(X_{1n} \cup z_s)$ such that $z_s$ is a free variable which satisfies the condition $\{z_s\} \cap X_s = \emptyset$. This set is also denoted as $PC_s,\text{Obs}(X_{1n}, z_s)$ for every sort $s \in S$.

**Definition 6.16** Let $\Sigma$ be a relational signature in $\text{AlgSig}$, let $\text{Obs}$ and $\text{In}$ be two set of sorts s.t. $\text{Obs}, \text{In} \subseteq \text{Sorts}(\Sigma)$ and $\text{In}$ is sensible wrt $\Sigma$. Let $A$ be a $\Sigma$-algebra. The relational observational equality ($\approx_{\text{Obs}, \text{In}}^A$) is formally defined for each sort $s$ and for each $v, w \in A[X_{1n}]$ as follows:

$$v \approx_{s, A} w \iff$$

$$\forall c \in C_{\Sigma, \text{Obs}}(X_{1n}, z_s). \forall \alpha \in X_{1n} \rightarrow A[X_{1n}].$$

$$I_{a \cup \{z, v\}}(c) = I_{a \cup \{z, w\}}(c) \land$$

$$\forall pc \in PC_{\Sigma, \text{Obs}}(X_{1n}, z_s). \forall \alpha \in X_{1n} \rightarrow A[X_{1n}].$$

$$I_{a \cup \{z, v\}}(pc) \iff I_{a \cup \{z, w\}}(pc)$$

if $s \in S - \text{Obs}$

$$v = w$$

if $s \in \text{Obs}$

**Notation:** To denote an observational equality $\approx_{\text{Obs}, \text{In}}^A$ we will normally drop the subscript denoting an algebra $A$ if it can be inferred from the context.

The following propositions can be proven in a similar way as in [BHW95]

**Proposition 6.17** Let $\Sigma$ be a relational signature in $\text{AlgSig}$, let $\text{Obs}$ and $\text{In}$ be two set of sorts s.t. $\text{Obs}, \text{In} \subseteq \text{Sorts}(\Sigma)$. The relational observational equality ($\approx_{\text{Obs}, \text{In}}^A$) is a family of partial $\Sigma$-congruences.

**Proposition 6.18** Let $\Sigma$ be a relational signature in $\text{AlgSig}$, let $\text{Obs}$ and $\text{In}$ be two set of sorts s.t. $\text{Obs}, \text{In} \subseteq \text{Sorts}(\Sigma)$ and let $A$ be a $\Sigma$-algebra. The relational observational equality ($\approx_{\text{Obs}, \text{In}}^A$) is weakly regular.

**Definition 6.19** Let $\Sigma$ be a signature, let $\text{In}$ and $\text{Obs}$ be two sets of sorts s.t. $\text{In}, \text{Obs} \subseteq \text{Sorts}(\Sigma)$ and let $X_{1n}$ be an $\text{In}$-sorted set of variables. The relational behavioural equality between $\Sigma$-algebras $\equiv_{\text{Obs}, \text{In}}$ is formally defined as:

$$A \equiv_{\text{Obs}, \text{In}} B \iff \forall t, r \in T_{\Sigma}(X_{1n}). \forall \alpha \in X_{1n} \rightarrow A. \forall \beta \in X_{1n} \rightarrow B.$$

$$I_\alpha(t) = I_\beta(t) \iff I_\beta(r) \land$$

$$\forall p, p' \in P_{\Sigma}(X_{1n}). \forall \alpha \in X_{1n} \rightarrow A. \forall \beta \in X_{1n} \rightarrow B.$$

$$I_\alpha(p) \iff I_\alpha(p') \iff I_\beta(p) \iff I_\beta(p')$$

32
Proposition 6.20 Let $\Sigma$ be a signature and let $I_n$ and $\text{Obs}$ be two sets of sorts s.t. $I_n, \text{Obs} \subseteq \text{Sorts}(\Sigma)$. $\equiv_{\text{Obs},I_n}$ is an equivalence relation.

Proposition 6.21 Let $\Sigma$ be a signature, let $I_n$ and $\text{Obs}$ be two sets of sorts s.t. $I_n, \text{Obs} \subseteq \text{Sorts}(\Sigma)$ and let $A$ be a $\Sigma$-algebra. $\equiv_{\text{Obs},I_n}$ is factorizable by $\approx_A^{\text{Obs},I_n}$.

6.4 The relativization functions

One possible way to define the functions which have to satisfy the behavioural relativization conditions is to define for any inclusion $i : \Sigma \hookrightarrow \Sigma'$, the inclusion $\text{bihol}[\Sigma] : \Sigma \hookrightarrow \text{bihol}[\Sigma](\Sigma')$ with a disjoint copy of $\Sigma'$ which we will denote as $\text{Copy}'(\Sigma')$ and to define the function

$$brehol[i, \text{bihol}[\Sigma]] : P(\text{Sen}_{\text{HOL}}(\Sigma')) \rightarrow P(\text{Sen}_{\text{HOL}}(\text{bihol}[\Sigma](\Sigma')))$$

in such a way that if a $\text{bihol}[\Sigma](\Sigma')$-algebra $A''$ satisfies the set of sentences $brehol[i, \text{bihol}[\Sigma]](\Phi)$, then $A''$ must also satisfy the following condition:

$$A''|_{\text{Copy}'(\Sigma)} \models A''|_{\Sigma'} \equiv_{\text{Obs},I_n} A''|_{\text{Copy}'(\Sigma)} \models \text{Copy}'(\Phi)$$

This can be achieved by defining the set of sentences $brehol[i, \text{bihol}[\Sigma]](\Phi)$ as the union of $\text{Copy}'(\Phi)$ with an axiomatization in higher-order logic of the observational equality $\approx_{\text{Obs},I_n}$ and an axiomatization of a pseudo epimorphism between $A''|_{\Sigma}$ and $A''|_{\text{Copy}'(\Sigma)}$. This axiomatization requires extra symbols to denote the observational equality and the pseudo epimorphism. The axiomatization of the observational equality is based on [HS96] and the axiomatization of the pseudo epimorphism is based on [Hen97].

For example, for the signature of a base specification $\text{Container}$ with sorts $\text{Container}, \text{Elem}, \text{Nat}$ and $\text{Bool}$, operations $\emptyset : \text{Container}, \text{insert} : \text{Elem} \times \text{Container} \rightarrow \text{Container}, \text{union} : \text{Container} \times \text{Container} \rightarrow \text{Container}, \ldots$ which will be denoted by $\text{Contsign}$ and set of sentences $\text{Contax}$ including

$$\forall s : \text{Container}, \text{union} \emptyset s = s$$

$$\forall s, s' : \text{Container}, \text{insert} (\text{insert} e s) s' (\text{insert} e (\text{union} s s'))$$

$\text{bihol}[\text{Contsign}]$ would be defined as:

$$\text{bihol}[\text{Contsign}](\text{Contsign}) = \text{Contsign} \cup \text{Copy}(\text{Contsign}) \cup$$

$$\{ s \times s | s \in \text{Sorts}(\text{Contsign}) \} \cup \{ \pi_s : s \rightarrow \text{Copy}(s) | s \in \text{Sorts}(\text{Contsign}) \}$$

(when $\text{Copy}(\text{Contsign})$ is a disjoint copy of the signature $\text{Contsign}$, the symbols $\sim_s$ are used to denote the observational equality and the symbols $\pi_s$ to denote a pseudoepimorphism), and the set of sentences $brehol[i, \text{bihol}[\text{Contsign}]](\text{Contax})$ will include the axioms $\text{Contax}$ appropriately renamed for the signature $\text{Copy}(\text{Contsign})$, axioms to define the indistinguishability relation $\langle \text{Indist}_\text{rel}[\text{Contsign}][\sim], \text{Obs}, I_n \rangle$.
and axioms to establish a pseudo-epimorphism \( \text{pEpi}[\text{biho}l(\text{Contsign})] \)
for given sorts \( \text{Obs}, \text{In} \) which determine the indistinguishability relation. These 
set of axioms are detailed in this section.

To define the functions which have to satisfy the quotient relativization conditions we proceed in a similar way defining for any inclusion \( i : \Sigma \hookrightarrow \Sigma' \), the 
inclusion \( \text{qihol}[\Sigma] : \Sigma' \hookrightarrow \text{biho}l(\Sigma)(\Sigma') \) with also a disjoint copy of \( \Sigma' \) which we 
will also denote as \( \text{Copy}'(\Sigma') \), and defining the function

\[
\text{qrelhol}[i, \text{qihol}[\Sigma]] : \mathcal{P}(\text{Sen}_B\text{HOL}(\Sigma')) \rightarrow \mathcal{P}(\text{Sen}_B\text{HOL}(\text{qihol}[\Sigma](\Sigma')))
\]

in such a way that if a \( \text{qihol}[\Sigma](\Sigma') \)-algebra \( A' \) satisfies a set of sentences 
\( \text{qrelhol}[i, \text{qihol}[\Sigma]](\Phi) \), then \( A' \) must also satisfy the following condition:

\[
A'_{\Sigma} \cong A'_{\text{Copy}'(\Sigma')} \cong \text{Copy}'(\Phi) \wedge A'_{\text{Copy}'(\Sigma')} \models \text{Copy}'(\Phi)
\]

This can be achieved in a symmetrical way as in the definition of the set of sentences 
\( \text{brelhol}[i, \text{biho}l(\Sigma)](\Phi) \) by defining the set of sentences \( \text{qrelhol}[i, \text{qihol}[\Sigma]](\Phi) \) 
as the union of \( \text{Copy}(\Phi) \) with an axiomatization in higher-order logic of the 
observational equality \( \cong \text{Copy}_{\text{Obs}}, \text{Copy}_{\text{In}} \) (instead of \( \cong \text{Obs}, \text{In} \)) and an axiomatization of a pseudo epimorphism between \( A'_{\Sigma} \) and \( A'_{\text{Copy}'(\Sigma')} \) (instead of a pseudo epimorphism between \( A'_{\Sigma} \) and \( A'_{\text{Copy}'(\Sigma')} \)).

For the same signature \( \text{Contsign} \) and the set of axioms \( \text{Contax} \) presented above, \( \text{qihol}[\text{Contsign}] \) would be defined as

\[
\text{qihol}[\text{Contsign}](\text{Contsign}) = \text{Contsign} \cup \text{Copy}(\text{Contsign}) \cup \{\sim_{\text{Copy}}(s) : \text{Copy}(s) \times \text{Copy}(s) \mid s \in \text{Sorts}(\text{Contsign})\} \cup \{\exists_{\text{Copy}}(s) : \text{Copy}(s) \rightarrow s \mid s \in \text{Sorts}(\text{Contsign})\}
\]

and the set of sentences \( \text{qrelhol}[i, \text{qihol}[\text{Contsign}]](\text{Contax}) \) will also include the axioms \( \text{Copy}(\text{Contax}) \), axioms to define the indistinguishability relation 
\( \text{Indist}\text{-rel}[\text{Copy}(\text{Contsign}[\sim]), \text{Obs}, \text{In}] \) and axioms to establish a pseudoepimorphism \( \text{pEpi}[\text{qihol}[\text{Contsign}], \text{Obs}, \text{In}] \).

See the rest of this subsection for a complete formal definition of the functions

\[
\text{brelhol}[i, \text{biho}l(\Sigma)] : \mathcal{P}(\text{Sen}_B\text{HOL}(\Sigma')) \rightarrow \mathcal{P}(\text{Sen}_B\text{HOL}(\text{biho}l(\Sigma)(\Sigma')))
\]

and the functions

\[
\text{qrelhol}[i, \text{qihol}[\Sigma]] : \mathcal{P}(\text{Sen}_B\text{HOL}(\Sigma')) \rightarrow \mathcal{P}(\text{Sen}_B\text{HOL}(\text{qihol}[\Sigma](\Sigma')))
\]

for any inclusion \( i : \Sigma \hookrightarrow \Sigma' \) and the next subsection for proofs that these functions 
satisfy the behavioural and quotient relativization conditions respectively.

**Definition 6.22** The relational signature \( \Sigma[\sim] \) is defined for any signature \( \Sigma \in \text{AlgSig} \) and for any Sorts(\Sigma)-set of new symbols \( \sim \) as:

\[
\Sigma[\sim] = \Sigma \cup \{\sim_s : s \times s \mid s \in S\}
\]
**Definition 6.23** The relational signature $\Sigma[\sim, \pi_{\text{Copy}}]$ for any signature $\Sigma \in \text{AlgSig}$, for any bijective signature morphism $\text{Copy} : \Sigma \rightarrow \text{Copy}(\Sigma)$ such that $\Sigma \cap \text{Copy}(\Sigma) = \emptyset$ and for any $\text{Sorts}(\Sigma)$-set of new symbols $\sim$ and $\pi$ is defined as:

$$\Sigma[\sim, \pi_{\text{Copy}}] = \Sigma[\sim] \cup \text{Copy}(\Sigma) \cup \{ \pi_s : s \rightarrow \text{Copy}(s) \mid s \in S \}$$

**Remark:** The relational signature $\text{Copy}(\Sigma)[\sim, \pi_{\text{Copy}}]_1$ stands for the following signature:

$$\text{Copy}(\Sigma)[\sim, \pi_{\text{Copy}}]_1 = \text{Copy}(\Sigma)[\sim] \cup \Sigma \cup \{ \pi_s : \text{Copy}(s) \rightarrow s \mid s \in S \}$$

**Definition 6.24** For any inclusion $i : \Sigma \hookrightarrow \Sigma'$ in AlgSig, the inclusion $\text{bihol}[\Sigma] : \Sigma' \hookrightarrow \text{bihol}[\Sigma](\Sigma')$ is defined as follows:

$$\text{bihol}[\Sigma](\Sigma') = \text{Copy}'(\Sigma') \cup \Sigma'[\sim, \pi_{\text{Copy}}]$$

where $\text{Copy}'$ is a pushout morphism of the bijective signature morphism $\text{Copy}$ and the inclusion $i : \Sigma \hookrightarrow \Sigma'$ such that

$$\text{Copy}'(\Sigma') \cap \Sigma'[\sim, \pi_{\text{Copy}}] = \text{Copy}'(\Sigma')$$

**Definition 6.25** For any inclusion $i : \Sigma \hookrightarrow \Sigma'$ in AlgSig, the inclusion $\text{qihol}[\Sigma] : \Sigma' \hookrightarrow \text{qihol}[\Sigma](\Sigma')$ is defined as follows:

$$\text{qihol}[\Sigma](\Sigma') = \text{Copy}'(\Sigma') \cup \text{Copy}'(\Sigma')[\sim, \pi_{\text{Copy}}]_1$$

where $\text{Copy}'$ is a pushout morphism of the bijective signature morphism $\text{Copy}$ and the inclusion $i : \Sigma \hookrightarrow \Sigma'$ such that

$$\text{Copy}'(\Sigma') \cap \text{Copy}'(\Sigma')[\sim, \pi_{\text{Copy}}]_1 = \text{Copy}'(\Sigma')$$

**Definition 6.26** The $\text{Sorts}(\Sigma)$-set of sentences $D[\Sigma, \text{In}]$ for any $\text{In} \subseteq \text{Sorts}(\Sigma)$ is defined as follows:

$$D[\Sigma, \text{In}] = \{ D_s[\Sigma, \text{In}] \mid s \in \text{Sorts}(\Sigma) \}$$

where

$$D_s[\Sigma, \text{In}] = \lambda x : s.((\forall P : [s].(\forall x_1 : s_1 \ldots \exists x_n : s_n \ldots \forall x_n : s_n \ldots \exists x_n : s_n \ldots \forall x_n : s_n \ldots P(x_1) \lor P(f(x_1, \ldots, x_n))) \lor P x)) \quad \text{if } s \in S - \text{In}$$

$$D_s[\Sigma, \text{In}] = \lambda x : s.\text{true} \quad \text{if } s \in \text{In}$$
**Definition 6.27** The set of sentences $\text{Indist}_{\text{rel}}[\Sigma[\sim], \text{Obs}, \text{In}]$ is defined as follows:

\[
\text{Indist}_{\text{rel}}[\Sigma[\sim], \text{Obs}, \text{In}] = \\
\{ \text{Indist}_{\text{rel}}[\Sigma[\sim], \text{Obs}, \text{In}, s] \mid s \in \text{Sorts}(\Sigma) \}
\]

where

\[
\text{Indist}_{\text{rel}}[\Sigma[\sim], \text{Obs}, \text{In}, s] = \lambda x : s. \lambda y : s. \exists s' \in \text{Sorts}(\Sigma) R_{s'} : [s', s'].
\]

\[
R_s(x, y) \land \text{OBSEQ}([S, Op], R, \text{Obs}, \text{In}) \land \\
\text{CONG}[\Sigma, R, \text{Obs}, \text{In}] \land D_s[\Sigma, \text{In}](x) \land D_s[\Sigma, \text{In}](y)
\]

\[
\text{OBSEQ}[\Sigma, R, \text{Obs}, \text{In}] = \bigwedge_{\text{obs} \in \text{Obs}} \forall v : \text{obs}. \forall w : \text{obs}. \\
D_{\text{obs}}[\Sigma, \text{In}](v) \land D_{\text{obs}}[\Sigma, \text{In}](w) \Rightarrow (R_{\text{obs}}(v, w) \Leftrightarrow v = w)
\]

\[
\text{CONG}[\Sigma, R, \text{Obs}, \text{In}] = \\
\bigwedge_{f : s \times \ldots \times s \rightarrow s \in \text{Op}(\Sigma)} \forall x_1 : s_1. \forall y_1 : s_1. \ldots \forall x_n : s_n. \forall y_n : s_n. \\
(R_{s_1}(x_1, y_1) \land \ldots \land R_{s_n}(x_n, y_n) \Rightarrow \\
R_{s}(f(x_1, \ldots, x_n), f(y_1, \ldots, y_n))) \land \\
\bigwedge_{p : s \times \ldots \times s \in \text{Pr}^s(\Sigma)} \forall x_1 : s_1. \forall y_1 : s_1. \ldots \forall x_n : s_n. \forall y_n : s_n. \\
(R_{s_1}(x_1, y_1) \land \ldots \land R_{s_n}(x_n, y_n) \Rightarrow \\
(p(x_1, \ldots, x_n) \Leftrightarrow p(y_1, \ldots, y_n))
\]

**Definition 6.28** The set of sentences $p\text{Epi}_{BHO}L[\Sigma[\sim], \pi_{C_{\text{opy}}}, \text{Obs}, \text{In}]$ is defined as:

\[
p\text{Epi}_{BHO}L[\Sigma[\sim], \pi_{C_{\text{opy}}}, \text{Obs}, \text{In}] = \text{Hom}[\Sigma[\sim], \pi_{C_{\text{opy}}}, \text{Obs}, \text{In}] \cup \\
E\text{pihom}[\Sigma[\sim], \pi_{C_{\text{opy}}}, \text{Obs}, \text{In}] \cup \sim \text{comp}[\Sigma[\sim], \pi_{C_{\text{opy}}}, \text{Obs}, \text{In}]
\]

36
where:

\[
Hom[\Sigma\sim, \pi_{C_{\text{sys}}}], \text{Obs}, \text{In}] = \bigcup_{f:s_1 \times \ldots \times s_n \rightarrow s \in \text{Ops}(\Sigma)} \{ \forall t_1 : s_1, \ldots, \forall t_n : s_n . \}
\]

\[
\bigwedge_{i \in [1..n]} D_{s_i}[\Sigma, \text{In}](t_i) \Rightarrow \pi_{C_{\text{sys}}}(f(t_1, \ldots, t_n)) = \text{Copy}(f)(\pi_{C_{\text{sys}}}(t_1), \ldots, \pi_{C_{\text{sys}}}(t_n)) \}
\]

\[
\bigcup_{p:s_1 \times \ldots \times s_n \rightarrow s \in \text{Prs}(\Sigma)} \{ \forall t_1 : s_1, \ldots, \forall t_n : s_n . \}
\]

\[
\bigwedge_{i \in [1..n]} D_{s_i}[\Sigma, \text{In}](t_i) \Rightarrow p(t_1, \ldots, t_n) \Leftrightarrow \text{Copy}(p)(\pi_{C_{\text{sys}}}(t_1), \ldots, \pi_{C_{\text{sys}}}(t_n))
\]

\[
\text{Epiphom}[\Sigma\sim, \pi_{C_{\text{sys}}}], \text{Obs}, \text{In}] = \bigcup_{s \in S} \{ \forall y : \text{Copy}(s), \exists x : s.D_{s}[\Sigma, \text{In}](x) \land \pi_{C_{\text{sys}}}(x) = y \}
\]

\[
\sim - \text{comp}[\Sigma\sim, \pi_{C_{\text{sys}}}], \text{Obs}, \text{In}] = \bigcup_{s \in S} \{ \forall x : s.\forall y : s.D_{s}[\Sigma, \text{In}](x) \land D_{s}[\Sigma, \text{In}](y) \Rightarrow x \sim y \Leftrightarrow \pi_{C_{\text{sys}}}(x) = \pi_{C_{\text{sys}}}(y) \}
\]

**Definition 6.29** For any inclusion \(i: \Sigma \hookrightarrow \Sigma' \) in AlgSig, the function

\[
\text{brethol}[i, \text{bihol}[\Sigma]] : \mathcal{P}(\text{Scl}_{\text{BHOL}}(\Sigma')) \rightarrow \mathcal{P}(\text{Scl}_{\text{BHOL}}(\text{bihol}[\Sigma](\Sigma')))
\]

is defined as follows:

\[
\text{brethol}[i, \text{bihol}[\Sigma]](\Phi) = \text{Copy}(\Phi) \cup \text{pEp}_{\text{BHOL}}[\sim, \pi_{C_{\text{sys}}}], \text{Obs}, \text{In}] \cup \{ \forall x : s.\forall y : s. \sim x , y \Leftrightarrow \text{Indist}_{\text{rel}}[\sim], \text{Obs}, \text{In}, s](x,y) \mid s \in \text{Sorts}(\Sigma) \} \cup \{ \forall x : s.\forall y : s. \sim x , y \Leftrightarrow x = y \mid s \in \text{Sorts}(\Sigma') - \text{Sorts}(\Sigma) \} \cup \{ \forall x : s. \pi_s(x) = x \mid s \in \text{Sorts}(\Sigma') - \text{Sorts}(\Sigma) \}
\]

37
Definition 6.30  For any inclusion $i : \Sigma \hookrightarrow \Sigma'$ in AlgSig, the function
grelh[i,qihol[\Sigma]] : \mathcal{P}(\text{Sen}_{BHOL}(\Sigma')) \to \mathcal{P}(\text{Sen}_{BHOL}(qihol(\Sigma)(\Sigma')))
is defined as follows:

grelh[i,qihol[\Sigma]](\Phi) =

\begin{align*}
&\text{Copy}(\Phi) \cup p\text{Epi}_{BHOL}[\text{Copy}(\Sigma)[\sim], \pi_{\text{copy}(s)}, \text{Copy}(\text{Obs}), \text{Copy}(\text{In})] \cup \\
&\{\forall x : \text{Copy}(s), \forall y : \text{Copy}(s), x \sim_{\text{copy}(s)} y \Rightarrow \\
&\text{Indist}_r\text{el}[\text{Copy}(\Sigma)[\sim], \text{Copy}(\text{Obs}), \text{Copy}(\text{In})](x, y) \mid s \in \text{Sorts}(\Sigma)\} \cup \\
&\{\forall x : \text{Copy}(s), \forall y : \text{Copy}(s), x \sim_{\text{copy}(s)} y \Rightarrow \\
&x = y \mid s \in \text{Sorts}(\Sigma') - \text{Sorts}(\Sigma)\} \cup \\
&\{\forall x : \text{Copy}(s), \pi_{\text{copy}(s)}(x) = x \mid s \in \text{Sorts}(\Sigma') - \text{Sorts}(\Sigma)\}
\end{align*}

Lemma 6.31  If $R_{A^{\text{Obs,In}}}$ is a partial $\Sigma$-congruence which satisfies the following condition:

\[ \forall \text{obs} \in \text{Obs}, \forall v, w \in A_{\text{obs}}[X_{\text{In}}], (v R_{A, \text{obs}}^{\text{Obs,In}} w \iff v = w) \]

then

\[ \forall s \in S, \forall v, w \in A[X_{\text{In}}], v R_{A,s}^{\text{Obs,In}} w \Rightarrow v \approx_{A,s}^{\text{Obs,In}} w \]

Proof:

It follows by context induction. The proof of the general case uses that $R$ is a partial $\Sigma$-congruence which coincides with the set theoretical equality for observable sorts.

The following lemma can also be found in [HS96]:

Lemma 6.32  The sentence $\text{Indist}_r\text{el}[\Sigma[\sim], \text{Obs}, \text{In}, s]$ for any sort $s \in \text{Sorts}(\Sigma)$ and for any free variables $x, y \in X_s$ satisfies the following condition which we will refer as the indistinguishability condition:

\[ \forall s \in S, \forall A \in \mathcal{A}(\Sigma), \forall \rho \in \{x, y\} \to A, \]

\[ [\text{Indist}_r\text{el}[\Sigma[\sim], \text{Obs}, \text{In}](x, y)]_{\rho,A, s} \Leftrightarrow \rho(x) \approx_{A,s}^{\text{Obs,In}} \rho(y) \]

6.5  The institution BHOL

Theorem 6.33  The algebraic institution BHOL extended with the following components:
• For each $\Sigma \in [\text{AlgSig}]$ and for a fixed but arbitrary sets $\text{Obs}, \text{In} \subseteq \text{Sorts}(\Sigma)$, the relational observational equality.

• For each $\Sigma \in [\text{AlgSig}]$ the behavioural satisfaction relation defined in section 4.6.1.

• For each inclusion $i : \Sigma \hookrightarrow \Sigma'$ in $\text{AlgSig}$, the inclusion $\text{bihol}[\Sigma] : \Sigma' \hookrightarrow \text{bihol}[\Sigma](\Sigma')$ and the function $\text{brehol}[i, \text{bihol}[\Sigma]]$

• For each inclusion $i : \Sigma \hookrightarrow \Sigma'$ in $\text{AlgSig}$, the inclusion $\text{qihol}[\Sigma] : \Sigma' \hookrightarrow \text{qihol}[\Sigma](\Sigma')$ and the function $\text{qrehol}[i, \text{qihol}[\Sigma]]$

is a behavioural algebraic institution.

**Proof:**

We have to prove that

• for each $\Sigma \in [\text{AlgSig}]$ and for a fixed but arbitrary sets $\text{Obs}, \text{In} \subseteq \text{Sorts}(\Sigma)$, the relational observational equality $\approx_{\text{Obs}, \text{In}}$ is a family of partial $\Sigma$-congruences which follows by proposition 4.6.17.

• The behavioural satisfaction relation satisfies the behavioural satisfaction condition. See theorem 3.35 of [HS96].

• For each inclusion $i : \Sigma \hookrightarrow \Sigma'$ in $\text{AlgSig}$, the function $\text{brehol}[i, \text{bihol}[\Sigma]]$ satisfies the behavioural relativization conditions.

- (1): Assume that $A' \in [\text{Alg}(\Sigma')]$ and $\Phi \in \mathcal{P}(\text{Sen}_{BHOL}(\Sigma'))$, and let $A$ be a $\Sigma$-algebra such that

$$A'|_{\Sigma} = A/\approx \land A' \models \Phi$$

To prove that

$$\exists A'' \in \text{Alg}(\text{bi}[\Sigma](\Sigma')). A''|_{\Sigma'} = A'' \land A'' \models \text{brehol}[i, \text{bi}[\Sigma]](\Phi)$$

where the $\Sigma'$-algebra $A''$ is defined as follows:

$$A''|_{\Sigma} = A$$

$$A''|_{\Sigma'} = A' \quad \text{for any sort } s \in \text{Sorts}(\Sigma') - \text{Sorts}(\Sigma)$$

$$f_{A''} = f_{A'} \quad \text{for any sort } f \in \text{Ops}(\Sigma') - \text{Ops}(\Sigma)$$

$$P_{A''} = P_{A'} \quad \text{for any sort } P \in \text{Pr}(\Sigma') - \text{Pr}(\Sigma)$$

we will define a $\text{bi}[\Sigma](\Sigma')$-algebra $A''$ such that

$$A''|_{\Sigma'} = A'' \land A'' \models \text{brehol}[i, \text{bi}[\Sigma]](\Phi)$$

The $\text{bi}[\Sigma](\Sigma')$-algebra $A''$ is defined as follows:

* $A''|_{\Sigma} = A$.  

39
* $A''_s = A'_s$ for any sort $s \in \text{Sorts}(\Sigma') - \text{Sorts}(\Sigma)$

* $f_{A''} = f_{A'}$ for any sort $f \in \text{Ops}(\Sigma') - \text{Ops}(\Sigma)$

* $P_{A''} = P_{A'}$ for any sort $P \in Pr(\Sigma') - Pr(\Sigma)$

* $A''|_{C_{\Sigma,y}(\Sigma)} = A'_s \approx_{A'}^{\text{Obs},\text{In}}|_{C_{\Sigma,y-1}}$.

* $A''_{\text{ops},(s)} = A'_s$ for any sort $s \in \text{Sorts}(\Sigma') - \text{Sorts}(\Sigma)$

* $\text{Copy}^I(f)_{A''} = f_{A'}$ for any sort $f \in \text{Ops}(\Sigma') - \text{Ops}(\Sigma)$

* $\text{Copy}^I(P)_{A''} = P_{A'}$ for any sort $P \in Pr(\Sigma') - Pr(\Sigma)$

* $A''_{\text{t} _{\Sigma}} = \epsilon_{A,s}$ for every sort $s \in S$.
  where
  \[ \epsilon_{A,s}(x) = [v]|_{\text{Obs},\text{In}} \text{ for all } s \in \text{Sorts}(\Sigma). \]

* $A''_{\text{a} _{\Sigma}} = \text{id}_{A,s}$ for every sort $s \in \text{Sorts}(\Sigma') - \text{Sorts}(\Sigma)$.
  where
  \[ \text{id}_{A,s}(x) = v \]

* $A''_{\text{r} _{\Sigma}} = \approx_{A,s}^{\text{Obs},\text{In}}$ for every sort $s \in \text{Sorts}(\Sigma)$.

* $A''_{\text{ops},(s)} = \approx_{A,s}$ for every sort $s \in \text{Sorts}(\Sigma') - \text{Sorts}(\Sigma)$.

It is obvious that $A''|_{\Sigma'} = A''$ and to prove that

\[ A'' \models \text{brhol}[i,\text{bholl}[\Sigma]][\Phi] \]

we have to prove that

* $A'' \models \text{brhol}[\Sigma][\text{bholl}(\Sigma'),\text{bholl}(\Phi)]$ which holds by the satisfaction lemma since $\Phi \in \mathcal{P}(\text{Seh BHOL}(\Sigma'))$, $A''|_{C_{\Sigma,y}'}(\Sigma') = A'|_{C_{\Sigma,y}'}$ and $A' \models \text{bholl}(\Phi)$.

* $A'' \models \text{Indist}_\text{rel}[\Sigma][\sim],\text{Obs},\text{In}$ holds since for every sort $s \in \text{Sorts}(\Sigma)$ $A''_{\text{a} _{\Sigma}} = \approx_{A,s}^{\text{Obs},\text{In}}$ and $\text{Indist}_\text{rel}[\Sigma][\sim],\text{Obs},\text{In}$ satisfies the indistinguishability condition.

* $A'' \models \text{bihol}[\Sigma]\{x : \text{Copy}(s),\forall y : \text{Copy}(s), x \sim_{C_{\Sigma,y}'} y \iff x = y \mid s \in \text{Sorts}(\Sigma') - \text{Sorts}(\Sigma)\}$

since for every sort $s \in \text{Sorts}(\Sigma') - \text{Sorts}(\Sigma)$ $A''_{\text{a} _{\Sigma}}$, is the set theoretical equality.

40
* $A'' \models pEpi[\Sigma, \pi_{C_{\text{obj}}}]$ holds since for every sort $s \in \text{Sort}(\Sigma)$, $A''_s$ can be seen as an epimorphism between $A$ and $A^{\text{Obs},\text{In}}_s$ and therefore $A''$ satisfies $\text{Hom}(\Sigma, \pi_{C_{\text{obj}}}, \text{Obs}, \text{In})$, $\text{Epihom}(\Sigma[\pi_{C_{\text{obj}}}], \text{Obs}, \text{In})$ and also $\sim^{\text{comp}}[\Sigma, \pi_{C_{\text{obj}}}, \text{Obs}, \text{In}]$.

* $A'' = \{ \forall x : \pi, \pi(x) = x \mid s \in \text{Sort}(\Sigma') - \text{Sort}(\Sigma) \}$ since for every sort $s \in \text{Sort}(\Sigma') - \text{Sort}(\Sigma)$ $A''_s$ is the identity function.

(2): Assume that $A'' \models \text{Alg(bihol}[\Sigma](\Sigma'))$, $\Phi \in \mathcal{P}(\text{SenBHOL}(\Sigma))$ and $A'' \models \text{rel}[i, b][\Sigma](\Phi)$.

We have to show that

$$\exists A' \in \text{Alg}(\Sigma'), A'|_{\Sigma} = A''|_{\Sigma} \approx \land A' \models \Phi$$

Since $A''|_{C_{\text{obj}}(\Sigma)} = \text{Copy'}(\Phi)$ and because of the indistinguishability condition together with the definition of $pEpi[\Sigma, \pi_{C_{\text{obj}}}]$ we can deduce that

$$A''|_{C_{\text{obj}}(\Sigma)} \cong A''|_{\Sigma}^{\text{Obs},\text{In}} \vert_{C_{\text{obj}}-1}$$

Finally, by the abstract satisfaction condition and the satisfaction condition of the institution we can prove our goal.

* For each inclusion $i : \Sigma \hookrightarrow \Sigma'$ in $\text{AlgSi}$, the function $\text{qrelhol}[i, \text{qihol}[\Sigma]]$ satisfies the quotient relativization conditions.

- (1)$\forall A' \in \text{Alg}(\Sigma'), \forall \Phi \in \mathcal{P}(\text{SenBHOL}(\Sigma'))$. $A' \models \Phi \Rightarrow$

$$\exists A'' \in \text{Alg}(\text{qihol}[\Sigma](\Sigma')), A''|_{\Sigma} \cong A'|_{\Sigma} \approx \land A'' \models \text{qrelhol}[\Sigma](\Sigma')(\Phi)$$

Assume that $A' \in \text{Alg}(\Sigma'), \Phi \in \mathcal{P}(\text{SenBHOL}(\Sigma'))$ and $A' \models \Sigma', \text{BHOL} \Phi$. To prove that

$$\exists A'' \in \text{Alg}(\text{qihol}[\Sigma](\Sigma')), A''|_{\Sigma} \cong A'|_{\Sigma} \approx \land A'' \models \Sigma', \text{BHOL} \text{qrelhol}[i, \text{qihol}[\Sigma]](\Phi)$$

we will define an algebra $A''$ such that

$$A''|_{\Sigma} \cong A'|_{\Sigma} \approx \land A'' \models \text{qrelhol}[i, \text{qihol}[\Sigma]](\Phi)$$

The algebra $A''$ is defined as follows:

* $A''|_{\Sigma} = A'|_{\Sigma} \approx A^{\text{Obs},\text{In}}_s$.

* $A'' = A'$ for any sort $s \in \text{Sort}(\Sigma') - \text{Sort}(\Sigma)$

* $f_{A''} = f_{A'}$ for any sort $f \in \text{Ops}(\Sigma') - \text{Ops}(\Sigma)$
* $P_{A''} = P_{A'}$ for any sort $P \in Pr(\Sigma') - Pr(\Sigma)$

* $A''|_{C_{op}'(\Sigma')} = A'|_{C_{op'} - 1}$.

* $A''_{c_{op'},(t)} = \epsilon_{A'|_{c_{op'}}, C_{op'}(t)}$ for every sort $s \in Sorts(\Sigma)$.

where

$\epsilon_{A', C_{op'}(t)}(v) = [v]_{C_{op'}(Obs), C_{op'}(In)}$

* $A''_{c_{op'},(t)} = id_{A'|_{c_{op'}}, C_{op'}(t)}$ for every sort $s \in Sorts(\Sigma') - Sorts(\Sigma)$.

where

$id_{A', s}(v) = v$

* $A''_{c_{op'},(t)} = {}^sA|_{c_{op'} - 1, c_{op'}(t)}$ for every sort $s \in Sorts(\Sigma)$.

It is obvious that $A''|_{\Sigma} \equiv A'|_{\Sigma/\approx}$ since $A''|_{\Sigma} = A'|_{\Sigma/\approx}$.

To prove that

$A'' \models_{\Sigma', BHO} \forall x : Copy'(s), \forall y : Copy'(s)$

we have to prove that

* $A'' \models_{\Sigma', [\Sigma', [\Sigma', BHO]} Copy'(\Phi)$ which holds by the satisfaction lemma since $\Phi \in P(Val_BHO(\Sigma'))$, $A''|_{c_{op'}(\Sigma')} = A'|_{c_{op'} - 1}$ and $A' \models_{\Sigma', BHO} \Phi$.

* $A'' \models \Indist_{rel}[Copy'(\Sigma)|\sim|, Copy'(Obs), Copy'(In)]$ holds since for every sort $s \in Sorts(\Sigma')$ $A''_{c_{op'},(t)} = {}^sA|_{c_{op'} - 1, c_{op'}(t)}$ and $\Indist_{rel}[Copy'(\Sigma)|\sim|, Copy'(Obs), Copy'(In)]$ satisfies the indistinguishability condition.

* $A'' \models_{\forall x : Copy'(s), \forall y : Copy'(s)} \forall x : Copy'(s), \forall y : Copy'(s)$

$x \sim_{c_{op'}(t)} y \iff x = y \mid s \in Sorts(\Sigma') - Sorts(\Sigma)$ since for every sort $s \in Sorts(\Sigma') - Sorts(\Sigma)$ $A''_{c_{op'}(t)}$ is the set theoretical equality.

* $A'' \models pEpi[Copy'(\Sigma)|\sim|, \pi_{C_{op'} - 1}]$ holds since for every sort $s \in Sorts(\Sigma)$, $A''_{c_{op'}(t)}$ can be seen as an epimorphism between $A'|\Sigma$ and $A'|\Sigma$ and therefore $A''$ satisfies $Hom[\Sigma|\sim|, \pi_{C_{op'}}, Obs, In], Epihom[\Sigma|\pi_{C_{op'}}, Obs, In]$ and also $\sim$ $comp[\Sigma|\sim|, \pi_{C_{op'}}, Obs, In]$.

* $A'' \models \forall x : Copy'(s), \pi_{C_{op'}(t)}(x) = x \mid s \in Sorts(\Sigma') - Sorts(\Sigma)$ since for every sort $s \in Sorts(\Sigma') - Sorts(\Sigma)$ $A''_{c_{op'}(t)}$ is the identity function.

- (2) $\forall A'' \in Alg(qihol)[\Sigma')(\Sigma'), \forall \Phi \in P(Val_BHO(\Sigma'))$. 

42
\[ A'' \models_{\Sigma, \text{B HOL}} qrelhol[i, qihol[\Sigma]](\Phi) \Rightarrow \]

\[ \exists A' \in [\text{Alg}(\Sigma')]. A''|_{\Sigma} \cong A'|_{\Sigma} / \equiv \land A' \models_{\Sigma, \text{B HOL}} \Phi \]

Assume that \( A'' \in \text{Alg}(qihol[\Sigma](\Sigma')) \), \( \Phi \in \mathcal{P}(\text{Sen}_{\text{B HOL}}(\Sigma')) \), and

\[ A'' \models_{\Sigma, \text{B HOL}} qrelhol[i, qihol[\Sigma]](\Phi) \]

By the definition of \( qrelhol[\Sigma](\Sigma')(\Phi) \) we know that

\[ A''|_{\text{Copy}(\Sigma)}|_{\text{Copy}} \models_{\Sigma, \text{B HOL}} \Phi \]

By the definitions of \( \text{Indist}_{\text{rel}}[\text{Copy}(\Sigma)[\sim, \text{Copy}(\text{Obs}), \text{Copy}(\text{In})] \]

and \( p\text{Epi}(\text{Copy}(\Sigma)[\sim, \pi_{\text{Copy}}^{-1}], \text{Copy}(\text{Obs}), \text{Copy}(\text{In})] \)

we have that \( A''|_{\text{Copy}(\Sigma)}|_{\text{Copy}} / \approx_{\text{Obs}, \text{In}} \cong A''|_{\Sigma}. \)
References


