Chapter 5: Proof systems for ASL
1 Introduction

In this chapter, we present different consequence relations which are sound and in some cases complete with respect to the consequence relation between sentences and specifications of different ASLker and ASLnf languages and a proof system for the refinement relation between specifications presented in the chapter of the semantics of ASL. The consequence relations will be defined in terms of proof systems in order to give adequate representations of the consequence relations in the type theory UTT following the techniques presented in chapter 3. Most of the consequence relations will be presented for a given ASLker specification language in some cases with an arbitrary but fixed algebraic institution and in others with the concrete algebraic institutions FOLEG and HOL presented in the previous chapter. For the latter, we will also present consequence relations for the institutions of FOLEG and HOL by means of proof systems.

The judgements which we will use for the definition of the proof systems for the algebraic institutions FOLEG and HOL are:

- \( \phi \). This judgement means that the formula \( \phi \) holds.
- \( \phi : \tau \). This judgement will be used for the definition of \( \Pi_{HOL} \) and it means that the higher-order formula \( \phi \) has type \( \tau \).
- \( \phi =_{\beta} \phi' \). This judgement will be used for the definition of \( \Pi_{HOL} \) and it means that the higher-order formulas \( \phi \) and \( \phi' \) are equivalent by \( \beta \)-equality.

The proof system for FOLEG (\( \Pi_{FOLEG}(\Gamma, \Sigma) \)) for any set of sentences \( \Gamma \) with signature \( \Sigma \) will be formulated as a natural deduction system with the only sequent \( \Gamma \vdash x_{FOLEG} \phi \) where in this case \( x_{FOLEG} \) is a finite set of pairs of variable and its associated sort and the proof system for HOL (\( \Pi_{HOL}(\Gamma, \Sigma) \)) for any set of assumptions \( \Gamma \) with signature \( \Sigma \) will be formulated with the following sequents:

- \( \Gamma \vdash x_{HOL} \phi \) where in this case \( x_{HOL} \) is a finite set of pairs of variable and its associated higher-order type
- \( x_{HOL} \vdash x : \tau \) where \( x \) is a higher-order formula and \( \tau \) its associated type in \( Type_{HOL}(\Sigma) \).
- \( \phi =_{\beta} \phi' \) where \( \phi \) and \( \phi' \) are higher-order formulas.

In the following definitions, \( INSFH \) will range over FOLEG and HOL:

**Definition 1.1** The consequence relation \( \vdash_{\Sigma_{INSFH}(\Gamma, \Sigma)} : P(|Sen_{INSFH}(\Sigma)|) \times |Sen_{INSFH}(\Sigma)| \) for a given \( \Sigma \in |AlgSig| \) is defined as follows:

\[ \Gamma \vdash_{\Sigma} \phi \iff \text{Any closed sequent of the form } \Gamma \vdash x \phi \] has a derivation in \( \Pi_{INSFH}(\Gamma, \Sigma) \).
Definition 1.2 A consequence relation $\vdash_{\Sigma INSFH}(\Gamma, \Sigma)$: \( \mathcal{P}([\text{Sen INSFH}(\Sigma)]) \times [\text{Sen INSFH}(\Sigma)] \) is sound if it satisfies the following condition:

$$\forall \phi \in \text{Sen INSFH}(\Sigma), \Gamma \vdash_{\Sigma INSFH}(\Gamma, \Sigma) \phi \Rightarrow \left( \bigwedge \phi \in \Gamma \bigwedge A \vdash_{\Sigma INSFH} \psi \right) \Rightarrow (A \vdash_{\Sigma INSFH} \phi)$$

Definition 1.3 A consequence relation $\vdash_{\Sigma INSFH}(\Gamma, \Sigma)$: \( \mathcal{P}([\text{Sen INSFH}(\Sigma)]) \times [\text{Sen INSFH}(\Sigma)] \) is sound and complete if it satisfies the following condition:

$$\forall \phi \in \text{Sen INSFH}(\Sigma), \Gamma \vdash_{\Sigma INSFH}(\Gamma, \Sigma) \phi \Leftrightarrow \left( \bigwedge \phi \in \Gamma \bigwedge A \vdash_{\Sigma INSFH} \psi \right) \Rightarrow (A \vdash_{\Sigma INSFH} \phi)$$

The proof systems which are used in the definition of consequence relations can be divided into non-compositional and compositional proof systems. The non-compositional proof systems are inductively defined by specification expressions for ASLker specifications languages assigning for every case specific rules and axioms, and for ASLnf specification languages, these proof systems are defined for the normal forms of the specifications. The compositional proof systems are defined by a single set of rules in such a way that the derivations of the proof systems preserve the structure of the specification.

Our usual denotation of these proof systems will be $\Pi_{AINS}^{ASL}$ where $AINS$ is a concrete or an arbitrary institution and ASL is an ASLker or an ASLnf specification language. If the definition of the proof system is not the same for all the specification expressions of the language, we will denote by $\Pi_{AINS}^{ASL}(SP)$ the proof system associated to the specification expression $SP$ of ASL.

Finally, we present a proof system for the refinement of specifications of a concrete BASLker specification language which we will refer to as RBASL with a fixed but arbitrary algebraic institution.

2 Non-compositional proof systems

2.1 Inductively defined by specification expressions

In this subsection, we present this kind of proof systems for a BASLker specification language which we will denote as RBASL with the algebraic institutions FOLEQ and HOL. RBASL includes a renaming and a reachability operator apart from the common operators of BASLker languages. We present first the language RBASL and then the proof systems and consequence relations associated to these institutions.

Definition 2.1 A reachability constraint for a given signature $\Sigma = (S, Op)$ is a pair of a set of sorts and a set of functions $(S_R, F_R)$ such that $S_R \subseteq S$ and for any $f : s_1 \times \ldots \times s_n \rightarrow s \in F_R, s \in S_R$. 

3
Definition 2.2 For any signature $\Sigma = (S, Op) \in \text{AlgSig}$, an algebra $A \in \text{Alg}(\Sigma)$ satisfies a reachability constraint $(S_\mathcal{R}, \mathcal{F}_\mathcal{R})$ of $\Sigma (A \models (S_\mathcal{R}, \mathcal{F}_\mathcal{R}))$ if the following condition holds:

\[
A \models (S_\mathcal{R}, \mathcal{F}_\mathcal{R}) \iff \forall s \in S_\mathcal{R}, \forall v \in A_s. \exists t \in T(s, s_r)(X_{s-s_r}) \exists a : X_{s-s_r} \rightarrow A.I_0(t) = v
\]

Definition 2.3 RBASL is a BASLker language with additionally the following operators:

$\text{Signature (rename } SP \text{ by } \sigma) = \Sigma$

$\text{Symbols (rename } SP \text{ by } \sigma) = \sigma'(\text{Symbols}(SP))$

$\text{Models (rename } SP \text{ by } \sigma) = \{ A \in \text{Alg}(\Sigma)|A|_\sigma \in \text{Models}(SP) \}$

$\text{Signature (reach } SP \text{ with } (S_\mathcal{R}, \mathcal{F}_\mathcal{R})) = \text{Signature}(SP)$

$\text{Symbols (reach } SP \text{ with } (S_\mathcal{R}, \mathcal{F}_\mathcal{R})) = \text{Symbols}(SP)$

$\text{Models (reach } SP \text{ with } (S_\mathcal{R}, \mathcal{F}_\mathcal{R})) = \{ A \in \text{Models}(SP) | A \models (S_\mathcal{R}, \mathcal{F}_\mathcal{R}) \}$

where $\sigma : \text{Signature}(SP) \rightarrow \Sigma$ is a bijective signature morphism and $\sigma'$ is the pushout morphism of the following diagram:

\[
\begin{array}{ccc}
\text{Sym}(SP) & \xrightarrow{\sigma'} & \text{PO}(i, \sigma) \\
i & \downarrow & \downarrow \\
\text{Sign}(SP) & \xrightarrow{\sigma} & \Sigma
\end{array}
\]

where $i$ is the inclusion $i : \text{Signature}(SP) \hookrightarrow \text{Symbols}(SP)$.

Similarly to the consequence relations associated to the proof systems for FOQUE and HOL, we can define consequence relations of sentences of FOQUE and HOL from specification expressions using non-compositional proof systems. The definition of these consequence relation will use a function $\text{Symbols}$, which given a proof system $\Pi^{\text{ASL}}_{\text{INSFH}}$ (where again $\text{INSFH}$ ranges over FOQUE and HOL) and a specification expression $SP \in \text{SPEX}(ASL)$ will return the symbols of the proof system $\Pi^{\text{ASL}}_{\text{INSFH}}(SP)$ since we won’t assume that the symbols of the specification expression $SP$ is equal to the symbols of the proof system $\Pi^{\text{ASL}}_{\text{INSFH}}(SP)$. We will just assume that there exists an inclusion morphism between $\text{Signature}(SP)$ and $\text{Symbols}(\Pi^{\text{ASL}}_{\text{INSFH}}, SP)$. Also, the definition of the consequence relation will use the function $\Gamma_{\text{env}}$ which given a proof system and an specification expression $SP \in \text{SPEX}(ASL)$ will return the set of assumptions which we can use for the derivation of sentences from the given
specification expression. Finally, it will also be used the function $Rules$ which
given a proof system and an specification expression, it will return the proof
rules of the proof system.

For any bijective signature morphism $\sigma : Symbols(\Pi^{ASL}_{INSFH}(SP)) \rightarrow \Sigma'$, we will also denote by $\sigma$ the renaming function which given a proof system $\Pi^{ASL}_{INSFH}(SP)$ will return the resulting proof system of applying to every sentence of every rule of the given proof system the morphism between sentences $\sigma : Sen_{INSFH}(Symbols(\Pi^{ASL}_{INSFH}(SP))) \rightarrow Sen_{INSFH}(\Sigma').$

The definition of the consequence relation associated to these non-compositional
proof systems and some of its properties are as follows:

**Definition 2.4** The consequence relation $\vdash_{\Pi^{ASL}_{INSFH}}$ which relates specification expressions $SP \in SP\mathcal{E}(ASL)$ with sentences $\phi \in Sen(Symbols(\Pi^{ASL}_{INSFH},SP))$ is defined as follows:

$$SP \vdash_{\Pi^{ASL}_{INSFH}} \phi \iff \text{Any closed sequent of the form}$$

$$\Gamma \vdash_{\Pi^{ASL}_{INSFH},SP} \Rightarrow Symbols(\Pi^{ASL}_{INSFH},SP), \phi \text{ has a derivation in } Rules(\Pi^{ASL}_{INSFH},SP).$$

**Definition 2.5** For any signature $\Sigma \in Alg\mathcal{Sig}$, a $\Sigma$-algebra $A \in Alg(\Sigma)$ satisfies a closed sequent of the form $\Gamma \Rightarrow X \phi$ if the following condition holds:

$$\left( \bigwedge_{\psi \in \Gamma} A \vdash_{E,INSFH} \psi \right) \Rightarrow A \vdash_{E,INSFH} \phi$$

**Definition 2.6** For any signature $\Sigma \in Alg\mathcal{Sig}$, a $\Sigma$-algebra $A \in Alg(\Sigma)$ satisfies a closed sequent of the form $X \triangleright \phi : \tau$ if $[\phi]_{\rho,A} \in [\tau]_A$ for any $Type_{HOL}(\Sigma)$-sorted valuation $\rho$ such that for any $x : \tau \in X$, $x \in Dom,(\rho_x)$.

**Definition 2.7** For any signature $\Sigma \in Alg\mathcal{Sig}$, a $\Sigma$-algebra $A \in Alg(\Sigma)$ satisfies a closed sequent of the form $\phi =_{X,\beta} \phi'$ if $[\phi]_{\rho,A} = [\phi']_{\rho,A}$ for any $Type_{HOL}(\Sigma)$-sorted valuation $\rho$ such that for any $x : \tau \in X$, $x \in Dom,(\rho_x)$.

**Definition 2.8** For any signature $\Sigma \in Alg\mathcal{Sig}$, a $\Sigma$-algebra $A \in Alg(\Sigma)$ satisfies $\Pi^{ASL}_{INSFH}$ if for all the rules of the natural deduction system the following condition holds:

- If $A$ satisfies all the sequents of the premises then $A$ satisfies the sequent of the conclusion.

**Definition 2.9** A consequence relation $\vdash_{\Pi^{ASL}_{INSFH}(SP)}$ is sound if for all specification expressions $SP \in SP\mathcal{E}(ASL)$ and for all $A \in Models(SP)$ there exists a $Symbols(\Pi^{ASL}_{INSFH}(SP))$-algebra $A'$ which satisfies the following conditions (which we will refer as soundness conditions):
• \( A' |_{\text{Signature}(SP)} = A \).

• \( \forall \psi \in \Gamma \text{env}(\Pi_{\text{INSFH}}^{\text{ASL}}, SP), A' \models \psi \).

• \( A' \) satisfies Rules(\( \Pi_{\text{INSFH}}^{\text{ASL}}, SP \)).

**Proposition 2.10** If \( \phi \in \text{Sen}_{\text{INS}}(\text{Signature}(SP)) \) and \( \vdash_{\Pi_{\text{INSFH}}^{\text{ASL}}, SP} \theta \) is sound then \( SP \vdash_{\Pi_{\text{INSFH}}^{\text{ASL}}, SP} \phi \supset SP \models \phi \).

**Proof:**

Assume that \( SP \vdash_{\Pi_{\text{INSFH}}^{\text{ASL}}, SP} \phi \).

We have to show that for all \( A \in \text{Models}(SP) \), \( A \models \phi \).

Let \( \Lambda' \) be the Symbols(\( \Pi_{\text{INSFH}}^{\text{ASL}}, SP \))-algebra which satisfies the soundness conditions for a given Signature(\( SP \))-algebra \( \Lambda \).

Since \( SP \vdash_{\Pi_{\text{INSFH}}^{\text{ASL}}, SP} \phi \), there exists a derivation of the sequent

\[ \Gamma \text{env}(\Pi_{\text{INSFH}}^{\text{ASL}}, SP) \Rightarrow \text{Symbols}(\Pi_{\text{INSFH}}^{\text{ASL}}, SP), X \phi \]

and since \( A' \) satisfies \( \Pi_{\text{INSFH}}^{\text{ASL}, SP} \) we can deduce that

\[ ( \bigwedge_{\psi \in \Gamma \text{env}(\Pi_{\text{INSFH}}^{\text{ASL}}, SP)} \models \text{Symbols}(\Pi_{\text{INSFH}}^{\text{ASL}}, SP) \psi ) \Rightarrow A' \models \text{Symbols}(\Pi_{\text{INSFH}}^{\text{ASL}}, SP) \phi \]

By the second soundness condition, we have that \( A' \models \text{Symbols}(\Pi_{\text{INSFH}}^{\text{ASL}}, SP) \phi \) and since \( A' |_{\text{Signature}(SP)} = A \) and \( \phi \in \text{Sen}_{\text{INSFH}}(\text{Signature}(SP)) \), we have that \( A \models \text{Signature}(SP) \).

**Definition 2.11** A consequence relation \( \vdash_{\Pi_{\text{INSFH}}^{\text{ASL}, SP}} \) is sound and complete if for all specification expressions \( SP \in \text{SPEC}(\text{ASL}) \), for all sentences \( \phi \in \text{Sen}_{\text{INSFH}}(\text{Symbols}(\Pi_{\text{INSFH}}^{\text{ASL}}, SP)) \) for all \( A \in \text{Models}(SP) \), if \( A \models \phi \) there exists a Symbols(\( \Pi_{\text{INSFH}}^{\text{ASL}, SP} \))-algebra \( A' \) such that \( A' \) satisfies the soundness conditions and \( A' \models \phi \) if and only if \( SP \vdash_{\Pi_{\text{INSFH}}^{\text{ASL}, SP}} \phi \).

Different kind of non-compositional proof systems inductively defined by specification expressions have been developed for the behavioural operators. In [HS96], these operators can just be applied to basic specifications, whereas in [HWW97] they can be applied to any kind of specification expressions. In [HS96] it is developed specific proof systems for the behaviour and the abstract operator for standard and behavioural theories and they use higher-order logic as specification logic, whereas in [HWW97] they present proof systems for the behaviour, abstract and quotient operator but just for standard theories and they use finitary and infinitary first-order logic. In this subsection, we will present in an arbitrary but fixed algebraic institutions, proof systems for the common operators of ASLker specification languages, and after that we will present two proof systems inductively defined by specification expressions for the BADSLker
specification language `RBASL`: one with the algebraic institution `FOLEQ` and 
the other with the algebraic institution `HOL`. Again, the design of the proof 
systems is based on [Hen97] and the axiomatisation of the indistinguishability 
relation for the proof system of higher-order logic is based on [HS96].

Basically, the proof systems for the behavioural operators are defined as the 
proof systems of structured specifications whose semantics are equivalent to the 
semantics of the behavioural operators. The result which states the equivalence 
is presented in this chapter in theorem 5.2.39 for the proof system of higher-
order logic. For first-order logic, the proof system is designed in a similar way 
but instead of having an axiomatisation of the indistinguishability relation in 
first-order logic, we have proof rules to derive proofs about indistinguishability.

### 2.1.1 Proof system for the common operators

**Definition 2.12** Let ASL be any ASLker specification language with an arbi-
trary but fixed algebraic institution AINS. The proof system $\Pi_{AINS}(ASL)$ 
is inductively defined for the two common operators of ASLker languages as 
follows:

\[
\begin{align*}
\text{Rules}(\Pi_{AINS}^{ASL}, <\Sigma, \Phi>) &= \Pi_{AINS}(\Phi, \Sigma) \\
\text{Symbols}(\Pi_{AINS}^{ASL}, <\Sigma, \Phi>) &= \Sigma \\
\Gamma_{env}(\Pi_{AINS}^{ASL}, <\Sigma, \Phi>) &= \Phi
\end{align*}
\]

\[
\begin{align*}
\text{Rules}(\Pi_{AINS}^{ASL}, SP_1 +\Sigma SP_2) &= \text{inr}''\left(\text{Rules}(\Pi_{AINS}^{ASL}, SP_1) \cup \text{inr}''(\text{Rules}(\Pi_{AINS}^{ASL}, SP_2))\right) \\
\text{Symbols}(\Pi_{AINS}^{ASL}, SP_1 +\Sigma SP_2) &= \text{inr}''(\text{Symbols}(\Pi_{AINS}^{ASL}, SP_1) \cup \text{inr}''(\text{Symbols}(\Pi_{AINS}^{ASL}, SP_2)) \\
\Gamma_{env}(\Pi_{AINS}^{ASL}, SP_1 +\Sigma SP_2) &= \text{inr}''(\Gamma_{env}(\Pi_{AINS}^{ASL}, SP_1) \cup \text{inr}''(\Gamma_{env}(\Pi_{AINS}^{ASL}, SP_2))
\end{align*}
\]

\[
\begin{align*}
\text{Rules}(\Pi_{AINS}^{ASL}, SP|\Sigma) &= \text{Rules}(\Pi_{AINS}^{ASL}, SP) \\
\text{Symbols}(\Pi_{AINS}^{ASL}, SP|\Sigma) &= \text{Symbols}(\Pi_{AINS}^{ASL}, SP) \\
\Gamma_{env}(\Pi_{AINS}^{ASL}, SP|\Sigma) &= \Gamma_{env}(\Pi_{AINS}^{ASL}, SP)
\end{align*}
\]

where the overloaded symbols `inr''` and `inr''` are the pushout morphisms of the 
following diagram:
\[
\text{Sym}(\Pi_{ALS}^{SL}, SP_1) \xRightarrow{\text{int}^\prime} \text{Sym}(\Pi_{ALS}^{\text{INS}}, SP_1) + \Sigma \text{Sym}(\Pi_{ALS}^{\text{INS}}, SP_2)
\]

\[
\Sigma \xrightarrow{i} \text{Sign}(SP_2) \xrightarrow{\text{int}^\prime} \text{Sym}(\Pi_{ALS}^{\text{INS}}, SP_2)
\]

and \(\text{int}^\prime\) and \(\text{int}^\prime\) also denote the usual renaming functions between environments and proof systems which use the pushouts morphisms with the same name.

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

\[
\text{iss} : \text{Signature}(SP_1) + \Sigma \text{Signature}(SP_2) \rightarrow \text{Symbols}(\Pi_{ALS}^{\text{INS}}, SP_1) + \Sigma \text{Symbols}(\Pi_{ALS}^{\text{INS}}, SP_2)
\]

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

**Theorem 2.13** Let \(\text{ASLK}\) be the \(\text{ASLker}\) specification language with just the two common operators of this class of languages. If \(\models_{\Pi_{ALS}^{\text{INS}}} (\theta)\) is sound then \(\models_{\Pi_{ALS}^{\text{INS}}} (\phi)\) is sound and if \(\models_{\Pi_{ALS}^{\text{INS}}} (\theta)\) is sound and complete then \(\models_{\Pi_{ALS}^{\text{INS}}} (\phi)\) is sound and complete.

### 2.1.2 The proof system \(\Pi_{\text{FOL EQ}}^{R_{ALS}}\)

**Definition 2.14** The natural deduction system \(\Pi_{\text{FOL EQ}}^{R_{ALS}}(\Gamma)\) is defined by the following set of rules for any \(\Gamma \in P(\text{Signature}(\Sigma))\):

\[
\Gamma \vdash \text{true} \quad (T) \\
\Gamma \vdash \phi_1 \land \phi_2 \quad (\land I) \\
\Gamma \vdash \phi_1 \land \phi_2 \quad (\land E I) \\
\Gamma \vdash \phi_1 \lor \phi_2 \quad (\lor I) \\
\Gamma \vdash \phi_1 \lor \phi_2 \quad (\lor E I) \\
\Gamma \vdash \phi_1 \lor \phi_2 \lor \psi \\
\Gamma \vdash \phi_1 \lor \phi_2 \lor \psi \\
\neg \Gamma \vdash \phi \quad (\neg I) \\
\Gamma \vdash \phi \quad (\neg E)
\]
\[
\begin{align*}
\Gamma \cup \phi &\Rightarrow X \phi' & \Gamma \Rightarrow X \phi \supset \phi' & \Gamma \Rightarrow X \phi' & \Gamma \Rightarrow X \phi \\
\text{(c i)} & & \Gamma \Rightarrow X \phi' & \text{(c e)} \\
\Gamma \Rightarrow X \phi[t/x] & t \in T_{\Sigma_s}(X) & \exists x : s. \phi & \text{(\exists I)} \\
\Gamma \Rightarrow X \exists x : s. \phi & \Gamma \cup \{ \phi \} \Rightarrow X u_{\{x:s\}} \psi & \Gamma \Rightarrow X \psi & \text{(\exists E)} \\
\Gamma \Rightarrow X \forall x : s. \phi & \Gamma \Rightarrow X \phi[t/x] & t \in T_{\Sigma_s}(X) & \text{(\forall I)} \\
\Gamma \Rightarrow X \forall x : s. \phi & \Gamma \Rightarrow X \phi[t/x] & t \in T_{\Sigma_s}(X) & \text{(\forall E)} \\
\end{align*}
\]

In the next definitions, we present different relational signatures which extend a given signature with the following symbols:

- Symbols to denote the indistinguishability relation for every sort of the signature and symbols to denote a definedness predicate also for every sort of the signature.
- The same extension as in the previous one plus symbols to denote a pseudoequimorphism between the original signature and a disjoint copy.
- The same extension as in the previous one plus symbols to denote contexts and context application.

**Definition 2.15** The relational signature \(\Sigma[-, D]\) is defined for any signature \(\Sigma = (S, Op)\) and for any \(S\)-families of new symbols \(\sim, D\) as:

\[\Sigma[-, D] = \Sigma \cup \{ \sim : s \times s | s \in S \} \cup \{ D : s | s \in S \}\]

**Definition 2.16** The relational signature \(\Sigma[-, D, \pi_{\text{Copy}}]\) for any signature \(\Sigma = (S, Op)\), for any bijective signature morphism \(\text{Copy} : \Sigma \rightarrow \text{Copy}(\Sigma)\) such that \(\Sigma \cap \text{Copy}(\Sigma) = \emptyset\) and for any \(S\)-family of new symbols \(\sim, D\) and \(\pi\) is defined as:

\[\Sigma[-, D, \pi_{\text{Copy}}] = \Sigma[-, D] \cup \text{Copy}(\Sigma) \cup \{ \pi_s : s \rightarrow \text{Copy}(s) | s \in S \}\]
**Remark:** The relational signature $\text{Copy}(\Sigma)[\sim, D, \pi_{\text{Copy}_y}]$ stands for the following signature:

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

**Definition 2.17** The relational signature $\Sigma[\sim, D, \pi_{\text{Copy}_y}, z, c, c_{\text{app}}]$ for any signature $\Sigma = (S, Op)$, for any bijective signature morphism $\text{Copy} : \Sigma \rightarrow \text{Copy}(\Sigma)$ such that $\Sigma \cap \text{Copy}(\Sigma) = \emptyset$, for any $S$-family of new symbols $\sim, \pi$ and $z$ and for any new symbols $c, c_{\text{app}}$ is defined as follows:

$$\text{Sorts}(\Sigma[\sim, D, \pi_{\text{Copy}_y}, z, c, c_{\text{app}}]) =$$

$$\text{Sorts}(\Sigma)[\sim, D, \pi_{\text{Copy}_y}] \cup \{ c[r \rightarrow s] \mid r, s \in S \}$$

$$\text{Ops}(\Sigma[\sim, D, \pi_{\text{Copy}_y}, z, c, c_{\text{app}}]) = \text{Ops}(\Sigma[\sim, D, \pi_{\text{Copy}_y}]) \cup$$

$$\{ c[s] : c[s \rightarrow s] \mid s \in S \} \cup$$

$$\{ c[r, f] : c[r \rightarrow s_1] \times \ldots \times c[r \rightarrow s_n] \rightarrow c[r \rightarrow s] \mid$$

$$r \in S, f : s_1 \times \ldots \times s_n \rightarrow s \in Op \} \cup$$

$$\{ c_{\text{app}}[r, s] : c[r \rightarrow s] \times r \rightarrow s \mid r \in S, s \in S \}$$

$$\text{Prs}(\Sigma[\sim, D, \pi_{\text{Copy}_y}, z, c, c_{\text{app}}]) = Pr(\Sigma[\sim, D])$$

In the following, some proof rules to prove sentences of the form $\forall x : S.D_s(x) \supset \phi$ are presented:

**Definition 2.18** The set of rules $D_{sr}[\Sigma[\sim, D], In]$ is defined as follows:

$$D_{sr}[\Sigma[\sim, D], In] = \{ D_{sr}[\Sigma[\sim, D], In, s] \mid s \in In \}$$

and the rule $D_{sr}[\Sigma[\sim, D], In, s]$ is defined for every sort $s \in S - In$ as follows:

$$\Gamma \Rightarrow X \forall x_1 : s_1, \ldots, \forall x_n : s_n,$$

$$\bigwedge_{s, \sim} \phi \{ x_1 / x \} \supset \phi \{ f(x_1, \ldots, x_n) / x \} \mid$$

$$f : s_1 \times \ldots \times s_n \rightarrow s \in \text{Ops}(\Sigma)$$

$$\Gamma \Rightarrow X \forall x : s.D_s(x) \supset \phi$$

Next, proof rules to define the proof system associated to the reachability operator are presented:

**Definition 2.19** The set of rules $\text{Reach}_{sr}[\Sigma, (S_R, F_R)]$ is defined as follows:

$$\text{Reach}_{sr}[\Sigma, (S_R, F_R)] = \{ \text{Reach}_{sr}[\Sigma, (S_R, F_R), s] \mid s \in S_R \}$$
and the rule $\text{Reach}_{\omega}[\Sigma, (S_R, F_R), s]$ is defined for every sort $s \in S_R$ as follows:

$$
\begin{align*}
\{ & \Gamma \Rightarrow_X \forall x_1 : s_1, \ldots, \forall x_n : s_n, \\
& \wedge_{s_i = s} \phi \{ x_i / x \} \supset \phi \{ f(x_1, \ldots, x_n) / x \} \mid \\
& f : s_1 \times \ldots \times s_n \rightarrow s \in F_R \} \\
& \Gamma \Rightarrow_X \forall x : s. \phi
\end{align*}
$$

In the following, proof rules to derive proofs about indistinguishability are presented. The proof rules appear in the proof system of the behavioural operators and there are two different kind of proof rules:

- proof rules which define the indistinguishability relation.
- proof rules to perform context induction to reason about formulas of the form $\forall ext : c[r \rightarrow s]. \phi$

**Definition 2.20** The set of rules $\text{Indist}_{\omega}[\Sigma[\sim]]$ is defined as follows:

- For each sort $s \in S - \text{Obs}$, the following rule:

$$
\begin{align*}
\Gamma \Rightarrow_X & \bigwedge_{obs \in \text{Obs}} \forall ext : c[r \rightarrow obs]. \\
& c_{app}[s, obs](ext, t) = c_{app}[s, obs](ext, r) \\
\Gamma \Rightarrow_X & t \sim_s r \\
& c_{app}[s, obs](ext, t) = c_{app}[s, obs](ext, r)
\end{align*}
$$

- The following context induction rule for each $r \in S$ and $s \in S - \text{In}$ such that $r \neq s$:

$$
\begin{align*}
\{ & \Gamma \Rightarrow_X \forall ext_1 : c[r \rightarrow s_1], \ldots, \forall ext_n : c[r \rightarrow s_n], \\
& \wedge_{s_i = s} \phi \{ ext_i / ext \} \supset \phi \{ c[r, f](ext_1, \ldots, ext_n) / ext \} \mid \\
& f : s_1 \times \ldots \times s_n \rightarrow s \in \text{Op} \} \\
& \Gamma \Rightarrow_X \forall ext : c[r \rightarrow s]. \phi
\end{align*}
$$

- The following context induction rule for each $r \in S$ and $s \in S - \text{In}$ such that $r = s$:

$$
\begin{align*}
\{ & \Gamma \Rightarrow_X \forall ext_1 : c[r \rightarrow s_1], \ldots, \forall ext_n : c[r \rightarrow s_n], \phi \{ c[z] / ext \} \wedge \\
& \wedge_{s_i = s} \phi \{ ext_i / ext \} \supset \phi \{ c[r, f](ext_1, \ldots, ext_n) / ext \} \mid \\
& f : s_1 \times \ldots \times s_n \rightarrow s \in \text{Op} \} \\
& \Gamma \Rightarrow_X \forall ext : c[r \rightarrow s]. \phi
\end{align*}
$$

- For each sort $s \in \text{Obs}$, the following rule:

$$
\begin{align*}
\Gamma \Rightarrow_X & t = r \\
\Gamma & \not\Rightarrow_X t \sim_s r \\
& D_s(t) \& D_s(r)
\end{align*}
$$
In the following, you will find the axiomatisation of the pseudomorphism used in the proof system for the behavioural operator. This axiomatisation is formulated over the signature $\Sigma[\sim, D, \pi_{\text{Copy}}]$ and it contains the following axioms:

- axioms to determine that $\pi_{\text{Copy}}$ is an homomorphism.
- axioms to determine that the homomorphism $\pi_{\text{Copy}}$ is surjective.
- axioms to establish the compatibility between the indistinguishability relation for every sort of the signature and the set theoretical equality associated to the disjoint copy of the given sort.

**Definition 2.21** The set of sentences $pEpifOOLEQ[\Sigma[\sim, D, \pi_{\text{Copy}}]]$ is defined as:

$$pEpifOOLEQ[\Sigma[\sim, D, \pi_{\text{Copy}}], Obs, In] = \text{Hom}[\Sigma[\sim, D, \pi_{\text{Copy}}], Obs, In] \cup \text{Epihom}[\Sigma[\sim, D, \pi_{\text{Copy}}], Obs, In] \cup \sim - \text{comp}[\Sigma[\sim, D, \pi_{\text{Copy}}], Obs, In]$$

where

$$\text{Hom}[\Sigma[\sim, \pi_{\text{Copy}}], Obs, In] = \bigcup_{f: s_1 \times \ldots \times s_n \rightarrow s \in O} \{ \forall t_1 : s_1, \ldots, t_n : s_n \}$$

$$\bigwedge_{i \in [1..n], s_i \in S \sim In} D_{s_i}(t_i) \Rightarrow$$

$$\pi_{\text{Copy}}(s)(f(t_1, \ldots, t_n)) = \text{Copy}(f)(\pi_{\text{Copy}}(s_1)(t_1), \ldots, \pi_{\text{Copy}}(s_n)(t_n))$$

$$\text{Epihom}[\Sigma[\sim, \pi_{\text{Copy}}], Obs, In] =$$

$$\bigcup_{s \in S \sim In} \{ \forall y : \text{Copy}(s), \exists x : s. D_s[\Sigma, In](x) \land \pi_{\text{Copy}}(s)(x) = y \}$$

$$\bigcup_{s \in In} \{ \forall y : \text{Copy}(s), \exists x : s. \pi_{\text{Copy}}(s)(x) = y \}$$

$$\sim - \text{comp}[\Sigma[\sim, \pi_{\text{Copy}}], Obs, In] =$$

$$\bigcup_{s \in S \sim In} \{ \forall x : s. \forall y : s. D_s[\Sigma, In](x) \land D_s[\Sigma, In](y) \Rightarrow$$

$$x \sim_s y \Leftrightarrow \pi_{\text{Copy}}(s)(x) = \pi_{\text{Copy}}(s)(y) \}$$

$$\bigcup_{s \in In} \{ \forall x : s. \forall y : s. x \sim_s y \Leftrightarrow \pi_{\text{Copy}}(s)(x) = \pi_{\text{Copy}}(s)(y) \}$$

12
Finally, we present an axiomatisation of the function application between contexts and values.

**Definition 2.22** The set of sentences $Ax_{\mathcal{A}_{app}}[\Sigma[\sim, D, \pi_{C_{app}}, z, c, c_{app}]]$ is defined as:

$$Ax_{\mathcal{A}_{app}}[\Sigma[\sim, D, \pi_{C_{app}}, z, c, c_{app}]] =$$

$$\{Ax_{\mathcal{A}_{app}}[\Sigma[\sim, D, \pi_{C_{app}}, z, c, c_{app}], r, s] \mid r \in S, s \in S\}$$

where

$$Ax_{\mathcal{A}_{app}}[\Sigma[\sim, D, \pi_{C_{app}}, z, c, c_{app}], r, s] =$$

$$\forall v : r.c_{app}[r, s](c[x], v) = v \wedge c_{app}[r, s](x, v) = x \wedge$$

$$\bigwedge_{f : s_1 \times \cdot \cdot \cdot \times s_n \rightarrow s \in O} \forall e_{ext_1 : e[r \rightarrow s_1], \cdot \cdot \cdot , e_{ext_n : e[r \rightarrow s_n]}} \forall v : r.
\ c_{app}[r, s](c[r, f](e_{ext_1}, \cdot \cdot \cdot , e_{ext_n}), v) =$$

$$\ \ \ \ \ \ \ \ \ \ \ \ f(c_{app}[r, s], (c, e_{ext_1}, \cdot \cdot \cdot , c_{app}[r, s], (e_{ext_n}, v))) , if \ r = s$$

$$\forall v : r.c_{app}[r, s](x, v) = x \wedge$$

$$\bigwedge_{f : s_1 \times \cdot \cdot \cdot \times s_n \rightarrow s \in O} \forall e_{ext_1 : e[r \rightarrow s_1], \cdot \cdot \cdot , e_{ext_n : e[r \rightarrow s_n]}} \forall v : r.
\ c_{app}[r, s](c[r, f](e_{ext_1}, \cdot \cdot \cdot , e_{ext_n}), v) =$$

$$\ \ \ \ \ \ \ \ \ \ \ \ f(c_{app}[r, s], (c, e_{ext_1}, \cdot \cdot \cdot , c_{app}[r, s], (e_{ext_n}, v))) , if \ r \neq s$$

**Definition 2.23** The proof system $\Pi_{\mathcal{BASL}}^{\mathcal{FOL}EQ}$ is inductively defined for the specific operators of the language as follows:

Rules($\Pi_{\mathcal{BASL}}^{\mathcal{FOL}EQ}$, rename $SP$ by $\sigma$) = $\sigma''(Rules(\Pi_{\mathcal{BASL}}^{\mathcal{FOL}EQ}, SP))$

Symbols($\Pi_{\mathcal{BASL}}^{\mathcal{FOL}EQ}$, rename $SP$ by $\sigma$) =

$$\sigma''(Symbols(\Pi_{\mathcal{BASL}}^{\mathcal{FOL}EQ}, SP))$$

$\Gamma_{env}(\Pi_{\mathcal{BASL}}^{\mathcal{FOL}EQ}$, rename $SP$ by $\sigma$) =

$$\sigma''(\Gamma_{env}(\Pi_{\mathcal{BASL}}^{\mathcal{FOL}EQ}, SP))$$
$Rules(\Pi^\text{RBASL}_{\text{FOLEQ}}, \text{reach } SP \text{ with } (\mathcal{S}_R, \mathcal{F}_R)) = Rules(\Pi^\text{RBASL}_{\text{FOLEQ}}, SP) \cup$

$\text{Reach}_{sr}[\text{Signature}(SP), (\mathcal{S}_R, \mathcal{F}_R)]$

$Symbols(\Pi^\text{RBASL}_{\text{FOLEQ}}, \text{reach } SP \text{ with } (\mathcal{S}_R, \mathcal{F}_R)) =$

$Symbols(\Pi^\text{RBASL}_{\text{FOLEQ}}, SP)$

$\Gamma_{env}(\Pi^\text{RBASL}_{\text{FOLEQ}}, \text{reach } SP \text{ with } (\mathcal{S}_R, \mathcal{F}_R)) = \Gamma_{env}(\Pi^\text{RBASL}_{\text{FOLEQ}}, SP)$

$Rules(\Pi^\text{RBASL}_{\text{FOLEQ}}, \text{behaviour } SP \text{ wrt } \approx) =$

$Copy''(Rules(\Pi^\text{RBASL}_{\text{FOLEQ}}, SP)) \cup$

$Indist_{sr}[\Sigma[\cdot], \text{Obs}, \text{In}] \cup D_{sr}[\Sigma[\cdot, D], \text{In}]$

$Symbols(\Pi^\text{RBASL}_{\text{FOLEQ}}, \text{behaviour } SP \text{ wrt } \approx) =$

$Copy''(Symbols(\Pi^\text{RBASL}_{\text{FOLEQ}}, SP)) \cup \Sigma[\cdot, \pi_{\text{copy}}, z, c, \textit{appd}]$

$\Gamma_{env}(\Pi^\text{RBASL}_{\text{FOLEQ}}, \text{behaviour } SP \text{ wrt } \approx) =$

$\Gamma_{env}(\text{rename } SP \text{ by } \text{Copy}) \cup$

$pEp_{\Pi^\text{FOLEQ}}[\Sigma[\cdot, \pi_{\text{copy}}, z, c, \textit{appd}]] \cup$

$Ax_{\textit{appd}}[\Sigma[\cdot, \pi_{\text{copy}}, z, c, \textit{appd}], >)$

$Rules(\Pi^\text{RBASL}_{\text{FOLEQ}}, \text{abstract } SP \text{ by } \equiv) =$

$Rules(\Pi^\text{RBASL}_{\text{FOLEQ}}, \text{behaviour } SP/\approx \text{ wrt } \approx)$

$Symbols(\Pi^\text{RBASL}_{\text{FOLEQ}}, \text{abstract } SP \text{ by } \equiv) =$

$Symbols(\Pi^\text{RBASL}_{\text{FOLEQ}}, \text{behaviour } SP/\approx \text{ wrt } \approx)$

$\Gamma_{env}(\Pi^\text{RBASL}_{\text{FOLEQ}}, \text{abstract } SP \text{ by } \equiv) =$

$\Gamma_{env}(\Pi^\text{RBASL}_{\text{FOLEQ}}, \text{behaviour } SP/\approx \text{ wrt } \approx)$
\[\text{Rules}(\Pi^{\text{RBA}}_{\text{FOL}} \circ S.P \rightarrow \approx) =\]
\[\text{Rules}(\Pi^{\text{RBA}}_{\text{FOL}} \circ \text{rename } S.P \text{ by } \text{Copy}) \cup\]
\[\text{Indist}_{\leq} \text{r} \pi \{\text{Copy}(\Sigma)[\sim, \pi_{\text{Copy} \equiv \approx}, \text{copyz}, \text{copyg}, \text{copyc}, \text{app}]\},\]
\[\text{Copy}(\text{Obs}), \text{Copy}(\text{In})\} \cup D_{\leq} \pi \{\text{Copy}(\Sigma)[\sim, D], \text{Copy}(\text{In})\}\]
\[\text{Symbols}(\Pi^{\text{RBA}}_{\text{FOL}} \circ S.P \rightarrow \approx) =\]
\[\text{Copy}''(\text{Symbols}(S.P, \Pi^{\text{RBA}}_{\text{FOL}})) \cup\]
\[\text{Copy}(\Sigma)[\sim, \pi_{\text{Copy} \equiv \approx}, \text{copyz}, \text{copyg}, \text{copyc}, \text{app}]\]
\[\Gamma_{\text{env}}(\Pi^{\text{RBA}}_{\text{FOL}} \circ S.P \rightarrow \approx) =\]
\[\Gamma_{\text{env}}(\text{rename } S.P \text{ by } \text{Copy}) \cup\]
\[pE_{\text{pushout}} \{\text{Copy}(\Sigma)[\sim, \pi_{\text{Copy} \equiv \approx}, \text{copyz}, \text{copyg}, \text{copyc}, \text{app}]\},\]
\[\text{Copy}(\text{Obs}), \text{Copy}(\text{In})\} \cup\]
\[\text{Ax}_{\text{app}} \{\text{Copy}(\Sigma)[\sim, \pi_{\text{Copy} \equiv \approx}, \text{copyz}, \text{copyg}, \text{copyc}, \text{app}]\},\]
\[\text{Copy}(\text{Obs}), \text{Copy}(\text{In})\} >\]

where \(\Sigma\) is the signature of the argument specification \(S.P\) for every case and the overloaded function symbol \(\sigma''\) is used here as the pushout of the following diagram:

\[
\begin{array}{ccc}
\text{Sym}(\Pi^{\text{RBA}}_{\text{FOL}} \circ S.P) & \xrightarrow{\sigma''} & \text{PO}(i, \sigma'') \\
\Sigma & \downarrow & \Sigma' \\
i & \sigma & \\
\end{array}
\]

where \(i : \Sigma \hookrightarrow \text{Sym}(\Pi^{\text{RBA}}_{\text{FOL}} \circ S.P)\) and it is also used as the renaming function of environments and of the proof system \(\Pi^{\text{RBA}}_{\text{FOL}}(S.P)\) which use the pushout just described in the obvious and standard way, and the overloaded symbol \(\text{Copy}''\) is the pullout morphism of the following diagram:

\[
\begin{array}{ccc}
\text{Sym}(\Pi^{\text{RBA}}_{\text{FOL}} \circ S.P) & \xrightarrow{\text{Copy}''} & \text{PO}(i, \text{Copy}) \\
\Sigma & \downarrow \text{Copy} & \text{Copy}(\Sigma) \\
\end{array}
\]

15
where \( i : \Sigma \hookrightarrow \text{Symbols}(\Pi^\text{RBASL}_{\text{FOLEQ}}, SP) \) and

\[
\text{Copy}''(\text{Symbols}(\Pi^\text{RBASL}_{\text{FOLEQ}}, SP)) \cap \\
\Sigma[\sim, \pi_{\text{copy}}, \varepsilon, c, \text{copy} \_{\Pi^\text{RBASL}_{\text{FOLEQ}}}] = \text{Copy}(\Sigma)
\]

and also

\[
\text{Copy}''(\text{Symbols}(\Pi^\text{RBASL}_{\text{FOLEQ}}, SP)) \cap \text{Copy}(\Sigma) \\
[\sim, \pi_{\text{copy}}^{-1}, \text{copy}_z, \text{copy}_c, \text{copy}_{c\_\text{FPPD}}] = \text{Copy}(\Sigma)
\]

The symbol \text{Copy}'' is also used as the renaming functions of environments and of set of rules using the pushout morphism just described with the same name in the obvious and standard way.

Note that although the proof system for the behaviour and quotient operators is defined in terms of the proof system of a structured specification expression, it is not needed to apply to the symbols of these proof systems the pushouts to avoid name clashes in structured specifications since the conditions which satisfy the pushout \text{Copy}'' guarantee no name clashes.

**Theorem 2.24** The consequence relation \( \vdash_{\Pi^\text{RBASL}_{\text{FOLEQ}}} \) is sound.

**Proof:**

The proof is by induction on specification expressions. For the two common operators the proof is trivial since \( \vdash_{\Pi^\text{FOLEQ}(\emptyset)} \) is sound. For the rest of the cases, we have to prove the following propositions:

- For all specification expressions \( SP \in \text{SPEX}(\text{RBASL}) \) such that \( \text{Signature}(SP) = \Sigma = (S, O, p) \), if \( \vdash_{\Pi^\text{RBASL}_{\text{FOLEQ}}(SP)} \) is sound then \( \vdash_{\Pi^\text{RBASL}_{\text{FOLEQ}}(SP|\Sigma)} \) is sound which follows trivially.

- For all specification expressions \( SP \in \text{SPEX}(\text{RBASL}) \) such that \( \text{Signature}(SP) = \Sigma = (S, O, p) \), if \( \vdash_{\Pi^\text{RBASL}_{\text{FOLEQ}}(SP)} \) is sound then \( \vdash_{\Pi^\text{RBASL}_{\text{FOLEQ}}(\text{rename SP by } \sigma)} \) is sound where \( \sigma : \text{Signature}(SP) \rightarrow \Sigma' \).

Assume that \( SP \in \text{SPEX}(\text{RBASL}) \) and assume that \( \vdash_{\Pi^\text{RBASL}_{\text{FOLEQ}}(SP)} \) is sound.

We have to prove that for any \( A \in \text{Models}(\text{rename SP by } \sigma) \) there exists a \( \text{Symbols}(\Pi^\text{RBASL}_{\text{FOLEQ}}(\text{rename SP by } \sigma)) \)-algebra (which we will refer as \( A' \)) such that:

- \( A'|_\Sigma = A \).

- \( \forall \psi \in \Gamma_{\text{env}}(\Pi^\text{ASL}_{\text{AINS}}, \text{rename SP by } \sigma), A' \models \psi \).

- \( A' \) satisfies \( \Pi^\text{ASL}_{\text{AINS}}(\text{rename SP by } \sigma) \).
Since $A|_{\sigma} \in Models(SP)$, by soundness of $\vdash_{\Pi^{RBASL}_{FOLEQ}(SP)}$, we know that there exists a $Symbols(\Pi^{RBASL}_{FOLEQ}(SP))$-algebra which we will refer as $A''$ such that:

- $A''|_{\Sigma} = A|_{\sigma}$.
- $\forall \psi \in \Gamma_{env}(\Pi^{ASL}_{AINS}(SP)). A' \models \psi$.
- $A''$ satisfies $\Pi^{ASL}_{AINS}(SP)$.

It is straightforward to prove that the $Symbols(\Pi^{RBASL}_{FOLEQ} (rename \ SP \ by \ \sigma))$-algebra $A' = A''|_{\{i: \sigma\}}$ where $\sigma'$ is the pushout morphism of $i: \Sigma \rightarrow Symbols(\Pi^{ASL}_{AINS}(SP))$ and $\sigma: \Sigma \rightarrow \Sigma'$ of the definition the proof system for this operator, satisfies the soundness conditions of the satisfaction condition of $AINS$ and because $A''$ satisfies the soundness conditions for the proof system $\Pi^{ASL}_{AINS}(SP)$.

> For all specification expressions $SP \in SPEX(RBASL)$ such that $Signature(SP) = \Sigma = (S, O_P)$, if $\vdash_{\Pi^{RBASL}_{FOLEQ}(SP)}$ is sound then $\vdash_{\Pi^{ASL}_{FOLEQ}(reach \ SP \ with \ (S_z, F_R))}$ is sound.

Assume that $SP \in SPEX(RBASL)$ and assume that $\vdash_{\Pi^{PBASL}_{FOLEQ}(SP)}$ is sound.

To prove that $\vdash_{\Pi^{PBASL}_{FOLEQ}(reach \ SP \ with \ (S_z, F_R))}$ is sound, we have to define for any $A \in Models(reach \ SP \ with \ (S_z, F_R))$, a $Symbols(\Pi^{RBASL}_{FOLEQ}(reach \ SP \ with \ (S_z, F_R)))$-algebra $B$ which satisfies the soundness conditions.

Assume that $A \in Models(reach \ SP \ with \ (S_z, F_R))$. By the semantics of the reachability operator, we know also that $A \in Models(SP)$.

Since $\Pi^{RBASL}_{FOLEQ}(SP)$ is sound, we know that there exists a $Symbols(\Pi^{RBASL}_{FOLEQ}(SP))$-algebra $A'$ such that $A'$ satisfies the soundness conditions and therefore $A'|_{Signature(SP)} = A$.

Assume that $B$ is a $Symbols(\Pi^{RBASL}_{FOLEQ}(SP))$-algebra which satisfies the soundness conditions and $B'|_{Signature(SP)} = A$. To prove that $B$ satisfies the soundness conditions we have to show that:

- $B|_{Signature(SP)} = A$ which obviously holds

- $\forall \psi \in \Gamma_{env}(\Pi^{ASL}_{AINS},reach \ SP \ with \ (S_z, F_R)). B \models \psi$ which holds since $\Gamma_{env}(\Pi^{ASL}_{AINS},reach \ SP \ with \ (S_z, F_R)) = \Gamma_{env}(\Pi^{ASL}_{AINS},SP)$.

- $B$ satisfies $\Pi^{RBASL}_{FOLEQ}(reach \ SP \ with \ (S_z, F_R))$ because $B$ satisfies $\Pi^{RBASL}_{FOLEQ}(SP)$ and for each sort $s \in S_R$ B satisfies $Reach_{s}[\Sigma, (S_z, F_R), s]$
and it follows trivially by term induction on the carrier set $B$, since 
$$B|_{Signature(SP)} = A$$ satisfies the reachability constraint.

- For all specification expressions $SP \in SPEX(RBASL)$ such that $Signature(SP) = \Sigma = (S, Op)$, if $\vdash_{\Pi_{RBASL}}(SP)$ is sound then $\vdash_{\Pi_{RBASL}}(\text{behaviour } SP \text{ wrt } \approx)$ is sound.

Assume that $SP \in SPEX(RBASL)$ and assume that $\vdash_{\Pi_{RBASL}}(SP)$ is sound.

To prove that $\vdash_{\Pi_{RBASL}}(\text{behaviour } SP \text{ wrt } \approx)$ is sound, we have to define for any $A \in Models(behaviour SP wrt \approx)$, a $Symbols(\Pi_{RBASL}(\text{behaviour } SP \text{ wrt } \approx))$-algebra $B$ such that:

- $B|_{Signature(SP)} = A$

- $\forall \psi \in \Gamma \models (\Pi_{ASL}, SP). B \models \psi$.

- $B$ satisfies $\Pi_{RBASL}(\text{behaviour } SP \text{ wrt } \approx)$

Since $\Pi_{RBASL}(SP)$ is sound, there exists a $Symbols(\Pi_{RBASL}(SP))$-algebra $A'$ such that $A'|_{Signature(SP)} = A/\approx^\text{obs}$ and $A'$ satisfies $\Pi_{RBASL}(SP)$.

The $Symbols(\Pi_{RBASL}(\text{behaviour } SP \text{ wrt } \approx))$-algebra $B$ is defined as follows:

$$B|_{Signature(SP)} = A$$ (and therefore it satisfies the first condition).

$$B|_{Copy^u(Symbols(\Pi_{RBASL}(SP)))} = A'|_{Copy^u - 1}$$

$$B|_{\tau_s} = \epsilon_{A,s} \text{ for every sort } s \in S.$$  

where $$\epsilon_{A,s}(v) = [v]_{\approx^\text{obs}, I_n} \text{ for all } v \in A_s.$$
The natural text is as follows:

\[ B|_{s|} = \approx_{A,s}^{I_n,Ob} \text{ for every sort } s \in S. \]

\[ B|_{r|s} = \{ \lambda frv : A[I_n]_r, I_{\alpha\cup(z_r,frv)} \} (erv) \mid \]

\[ \alpha \in (X_{I_n} \rightarrow A), \text{ and } z_r \in T_{\Sigma_r}(X_{I_n} \cup \{ z_r \}) \]

\[ \text{for every sort } s, r \in S \]

for a \( S \)-sorted infinite enumerable set of variables such that

\[ c[z_r]_B = z_s \text{ for every sort } s \in S \]

\[ c[r,f]_B(ex_1, \ldots, ex_n) = f(ex_1, \ldots, ex_n) \text{ for every sort } r \in S, \]

\[ f : s_1 \times \ldots \times s_n \rightarrow s \in Op, \]

\[ cx_1 \in T_{\Sigma_1}(X_{I_n} \cup \{ z_r \}), \ldots, cx_n \in T_{\Sigma_n}(X_{I_n} \cup \{ z_r \}) \]

\[ c_{app}[r,s]_B(\lambda frv : A[I_n]_r, I_{\alpha\cup(z_r,frv)} \} (erv),rv) = \]

\[ I_{\alpha\cup(z_r,rv)} \} (erv) \text{ for all } \alpha : X_{I_n} \rightarrow A[I_n], \]

\[ \text{for all } erv \in T_{\Sigma_r}(X_{I_n} \cup \{ z_r \}) \text{ and for all } rv \in A[I_n]. \]

To show that

\[ \forall \psi \in \Gamma_{env}(\Pi_{A|NS}^{AF}), \text{ behaviour } SP \text{ wrt } \approx_{A|NS}, B \models \psi \]

we have to prove the following propositions:

- \[ \forall \psi \in \Gamma_{env}(\Pi_{A|NS}^{AF}), \text{ rename } SP \text{ by } Copy, B \models \psi \]

which follows by the induction hypotheses and the proof of soundness for the rename operator.

- \[ B \models pEpi_{ROLEQ}[\Sigma[\sim, \pi_{Copy}, z, c, c_{app}]] \wedge \]

\[ B \models A_{x\text{-app}}[\Sigma[\sim, \pi_{Copy}, z, c, c_{app}]] \]

\[ B \models pEpi_{ROLEQ}[\Sigma[\sim, \pi_{Copy}, z, c, c_{app}]] \text{ because for every sort } s \]

\[ B_s, \text{ can be seen as an epimorphism between } A \text{ and } A/\approx_{A,s}^{Ob, I_n} \]

and
therefore $B$ satisfies $\text{Hom}[[\Sigma[\pi_{\text{Copy}}]]$, $\text{Epihom}[[\Sigma[\pi_{\text{Copy}}]]$ and also $\sim -\text{comp}[[\Sigma[\sim, \pi_{\text{Copy}}]]$. The proof of $B \models A x_c \text{app}[[\Sigma[\sim, \pi_{\text{Copy}}, z, e, c_c\text{app}]]$ is straightforward.

And to show that $B$ satisfies $\Pi_{FOL\text{EQ}}^{RB\text{ASL}}(\text{behaviour}\ SP\ \text{wrt}\ \approx)$ we have to show the following propositions:

- $B|_{\text{Copy}}^\ast(\text{Symbols}(\Pi_{FOL\text{EQ}}^{RB\text{ASL}}(SP)))$ satisfies $\Pi_{FOL\text{EQ}}^{RB\text{ASL}}(\text{rename}\ SP\ \text{by}\ \text{Copy})$.

Since $A \in \text{Models}(\text{behaviour}\ SP\ \text{wrt}\ \approx)$, we have that $A/\approx_{A}^{\text{Obs}, In} \in \text{Models}(SP)$ and by induction hypotheses we have that $A'$ satisfies $\Pi_{FOL\text{EQ}}^{RB\text{ASL}}(SP)$ where $A'|_{\text{Signature}(SP)} = A/\approx_{A}^{\text{Obs}, In}$. Therefore,

- $B|_{\text{Copy}}^\ast(\text{Symbols}(\Pi_{FOL\text{EQ}}^{RB\text{ASL}}(SP)))$ satisfies $\Pi_{FOL\text{EQ}}^{RB\text{ASL}}(\text{rename}\ SP\ \text{by}\ \text{Copy})$.

- $B$ satisfies $\text{Indist}_{\text{rel}}(x)[\Sigma[\sim, \pi_{\text{Copy}}, z, e, c_c\text{app}, \text{Obs}, \text{In}].$

$B$ satisfies

$$
\Gamma \vdash x \ \bigwedge_{\text{obs} \in \text{Obs}_s} \forall \text{ext} : c[r \rightarrow \text{obs}],
$$

from each sort $s \in S - \text{Obs}$ and

$$\Gamma \vdash x \ t = r \ \bigwedge_{\text{obs} \in \text{Obs}_s} D_s[\Sigma, \text{In}_s](t) \& D_s[\Sigma, \text{In}_s](r)
$$

for each sort $s \in S$ because $B|_s = \approx_{A, s}\text{In}_s^{\text{Obs}}$.

$B$ satisfies

$$\{\Gamma \vdash x \forall \text{ext}_1 : c[r \rightarrow s_1], \ldots, \forall \text{ext}_n : c[r \rightarrow s_n], \phi \{\text{ext}_1/\text{ext}, \ldots, \text{ext}_n/\text{ext}\} \in \phi \{f[s_1, \ldots, s_n / s \rightarrow s \in \text{Op}]\}
$$

by induction on the context $\text{ext} : c[r \rightarrow s]$

- For all specification expressions $SP \in \text{SPEX}(RB\text{ASL})$ such that $\text{Signature}(SP) = \Sigma = (S, Op)$, if $\vdash_{\Pi_{FOL\text{EQ}}^{RB\text{ASL}}(SP)}$ is sound then $\vdash_{\Pi_{FOL\text{EQ}}^{RB\text{ASL}}(SP/\approx)}$ is sound.

Assume that $SP \in \text{SPEX}(RB\text{ASL})$ and assume that $\vdash_{\Pi_{FOL\text{EQ}}^{RB\text{ASL}}(SP)}$ is sound. To prove that $\vdash_{\Pi_{FOL\text{EQ}}^{RB\text{ASL}}(SP/\approx)}$ is sound, we have to define for any $A \in \text{Models}(SP/\approx)$, a $\text{Symbols}(\Pi_{FOL\text{EQ}}^{RB\text{ASL}}(SP/\approx))$-algebra $B$ such that:

- $B|_{\text{Signature}(SP)} = A$

- $\forall \psi \in \text{Env}(\Pi_{ASL}^{\text{ALNS}}, SP/\approx), B \models \psi$

- $B$ satisfies $\Pi_{FOL\text{EQ}}^{RB\text{ASL}}(SP/\approx)$
By the definition of $\text{Models}(SP/\approx)$, we know that there exists a $\text{Signature}(SP)$-algebra $A'$ such that $A' / \approx_{\text{In}_{\text{Ob}}^{\text{Ob}} / A'} \cong A$ and $A' \in \text{Models}(SP)$. In the following, we will denote by $h : A'/\approx_{\text{Ob}_A,\text{In}} \to A$ the isomorphism between $A'/\approx_{\text{Ob}_A,\text{In}}$ and $A$. By induction hypotheses, we know that there exists a $\text{Symbols}(\Pi_{\text{BASL}}(SP))$-algebra $A''$ such that $A''|_{\text{Signature}(SP)} = A'$ and $A''$ satisfies $\Pi_{\text{BASL}}(SP)$. The $\text{Symbols}(\Pi_{\text{FOLEQ}}(SP/\approx))$-algebra $B$ is defined as follows:

$B|_{\Sigma} = A$(and therefore it satisfies the first condition).

$B_{|_{\text{Copy}}^n(\text{Symbols}(\Pi_{\text{BASL}}(SP)))} = A''|_{\text{Copy}^{n-1}}$.

$B_{\text{Copy}^{s}} = \epsilon_{A_{\text{Copy}^{-1}}, \text{Copy}(s)}$ for every sort $s \in S$.

where

$\epsilon_{A, \text{Copy}(s)}(v) = h([v]_{\text{Copy}(\text{Ob}), \text{Copy}(\text{In})})$ for any $A$ in $\text{Alg}(\text{Copy}(\Sigma))$,

for any $s \in S$ and for any $v \in A_{\text{Copy}(s)}$.

$B_{\text{Copy}^{s}} = \approx_{\text{Copy}^{-1}, \text{Copy}(s)}$.

$B|_{\text{Copy}(s), \text{Copy}(t)} = \{\lambda\alpha : A'|_{\text{Copy}^{-1}}[\text{Copy}(\text{In})], A_{\text{Copy}(s)}, A_{\text{Copy}(t)}, \{c_{\text{Copy}(s)}, c_{\text{Copy}(t)}\} |$ $\alpha \in (X_{\text{Copy}(\text{In})} \to A')$, $c_{\text{Copy}(s)} \in T_{\text{Copy}(\Sigma), \text{Copy}(s)}(X_{\text{Copy}(\text{In})} \cup \{z_{\text{Copy}(s)}\})$ for every sort $s, r \in S$ and for a $S$-sorted infinite enumerable set of variables such that for every sort $s \in S X_s \cap z_s = \emptyset$.

$e[z_{\text{Copy}(s)}]B = z_{\text{Copy}(s)}$ for every sort $s \in S$.

$e[\text{Copy}(r), \text{Copy}(f)]B(c_1, \ldots, c_n) = \text{Copy}(f)(c_1, \ldots, c_n)$

21
for every sort \( r \in S \), \( f : s_1 \times \ldots \times s_n \to s \in Op \),

\[ \exists x_1 \in T_{C_{\text{copy}}(\Sigma), C_{\text{copy}}(s_1)}(X_{C_{\text{copy}}(In)} \cup \{ \pi_{C_{\text{copy}}(r)} \}), \ldots, \]

\[ \exists x_n \in T_{C_{\text{copy}}(\Sigma), C_{\text{copy}}(s_n)}(X_{C_{\text{copy}}(In)} \cup \{ \pi_{C_{\text{copy}}(r)} \}) \]

\[ c_{\text{app}}(\text{Copy}(r), \text{Copy}(s)) \]

\[ (\lambda f v : A'[\text{Copy}(In)]C_{\text{copy}}(r), I_{\alpha \{ (z_{C_{\text{copy}}(r)}, f v) \}}(c z v), v v) = \]

\[ I_{\alpha \{ (z_{C_{\text{copy}}(r)}, f v) \}}(c z v) \text{ for all } \alpha : X_{C_{\text{copy}}(In)} \to A[\text{Copy}(In)], \]

for all \( c z v \in T_{C_{\text{copy}}(\Sigma), C_{\text{copy}}(s)}(X_{C_{\text{copy}}(In)} \cup \{ \pi_{C_{\text{copy}}(r)} \}) \)

and for all \( v v \in A[\text{Copy}(In)]_{\text{Copy}(s)} \).

To show that

\[ \forall \psi \in \Gamma \text{ in } \Pi_{\text{ALNS}}^{ASL}, SP/ \approx, B \models \psi \]

we have to show the following propositions:

- \( \forall \psi \in \Gamma \text{ in } \Pi_{\text{ALNS}}^{ASL}, \text{rename } SP \text{ by } \text{Copy}), B \models \psi \) which follows by the induction hypotheses for the proof of soundness for the rename operator.

- \( B \models pEpi_{\text{FOL}}[\text{Copy}(\Sigma)[\sim, \pi_{C_{\text{copy}}-1}, z, c, c_{\text{app}}]] \wedge \)

\[ B \models A x c_{\text{app}}(\text{Copy}(\Sigma)[\sim, \pi_{C_{\text{copy}}-1}, z, c, c_{\text{app}}]) \]

\( B \models pEpi_{\text{FOL}}[\text{Copy}(\Sigma)[\sim, \pi_{C_{\text{copy}}-1}, z, c, c_{\text{app}}]] \) because for every sort \( s \) \( B_{s_{\text{rename}(r)}} \) can be seen as an epimorphism between \( A' \) and \( A \) and therefore \( B \) satisfies \( \text{Hom}(\text{Copy}(\Sigma)[\sim, \pi_{C_{\text{copy}}}], Epi_{\text{hom}}(\text{Copy}(\Sigma)[\sim, \pi_{C_{\text{copy}}}]) \) and also \( \sim_{\text{copy}} \text{Map}(\text{Copy}(\Sigma)[\sim, \pi_{C_{\text{copy}}}]). \) The proof of \( B \models A x c_{\text{app}}(\Sigma[\sim, \pi_{C_{\text{copy}}}, z, c, c_{\text{app}}]) \) is straightforward.

And to show that \( B \) satisfies \( \Pi_{\text{FOL}}^{ASL}(SP/ \approx) \) we have to show the following propositions:

- \( B|_{\text{copy}''(\text{Symbol}(\Pi_{\text{FOL}}^{ASL}(SP)))} \) satisfies \( \Pi_{\text{FOL}}^{ASL}(\text{rename } SP \text{ by } \text{Copy}). \)

Since \( A' \in \text{Models}(SP) \), by induction hypotheses we have that \( A'' \) satisfies \( \Pi_{\text{FOL}}^{ASL}(SP) \) where \( A''|_{\text{Signature}(SP)} = A' \). Therefore, \( B|_{\text{copy}''(\text{Symbol}(\Pi_{\text{FOL}}^{ASL}(SP)))} \) satisfies \( \Pi_{\text{FOL}}^{ASL}(\text{rename } SP \text{ by } \text{Copy}). \)

- \( B \) satisfies \( \text{Indist}_{\text{rel}}[\text{Copy}(\Sigma)[\sim, \pi_{C_{\text{copy}}-1}, z, c, c_{\text{app}}], \text{Copy}(\text{Obs}), \text{Copy}(\text{In})] \) in the same way as in the case of the behavioural operator.

22
• For all specification expressions $SP \in SPEX(RBASL)$ such that $\text{Signature}(SP) = \Sigma = (S, Op)$, if $\vdash^{\Pi^{RBASL}_{\text{POLEQ}}}(SP)$ is sound then $\vdash^{\Pi^{RBASL}_{\text{POLEQ}}}($abstract $SP$ by $\equiv$) is sound.

It is trivial since by the two previous proofs we know that if $\vdash^{\Pi^{RBASL}_{\text{POLEQ}}}(SP)$ is sound then $\vdash^{\Pi^{RBASL}_{\text{POLEQ}}}($behaviour $SP/\equiv$ wrt $\equiv$) is sound.

**Theorem 2.25** There is no sound and complete consequence relation of the form $\vdash^{\Pi^{RBASL}_{\text{POLEQ}}}$ for RBASL.

**Proof:**

See [MS85]. The basic idea of the proof is to define a specification $SP$ such that the set $\{ \phi \mid SP \models \phi \}$ is not recursively enumerable since they work with consequence relations $\vdash$ which are recursively enumerable. They show that the set of sentences which satisfy the single sorted algebra $NAT$ with carrier set the natural numbers and with the usual operations $\text{zero}, \text{succ}, \text{subtract}, \text{add}, \text{multiply}$ (where $\text{subtract}_{NAT}(m, n) = 0$ if $n > m$) is not recursively enumerable. They use equational logic with data constraints to define an specification whose class of models is the class of algebras isomorphic to $NAT$. Since all the consequence relations of our proof systems are recursively enumerable because we work with natural deduction systems with a finite set of finitary rules, the proof is valid for our framework if we define the specification in RBASL. We will refer to this specification as $NSP$ and it is defined as follows:

$$NSP = \text{reach Nat with} (\text{nat}, (\text{zero} : \text{nat}, \text{succ} : \text{nat} \rightarrow \text{nat}))$$

where

$$\text{Nat} = < (\text{nat}, \text{zero} : \text{nat}, \text{succ} : \text{nat} \rightarrow \text{nat}, \text{add} : \text{nat} \times \text{nat} \rightarrow \text{nat}, \text{multiply} : \text{nat} \times \text{nat} \rightarrow \text{nat}),$$

$$\langle \forall m, n : \text{nat}. \text{add}(n, \text{zero}) = n \land \text{add}(n, \text{succ}(m)) = \text{succ} \text{add}(n, m) \rangle \land$$

$$\langle \forall m, n : \text{nat}. \text{subtract}(\text{zero}, n) = \text{zero} \land$$

$$\text{subtract}(\text{succ}(n), \text{succ}(m)) = \text{subtract}(n, m) \rangle \land$$

$$\langle \forall m, n : \text{nat}. \text{multiply}(n, \text{zero}) = \text{zero} \land$$

$$\text{multiply}(n, \text{succ}(m)) = \text{add}(n, \text{multiply}(n, m)) \rangle \land$$

$$\langle \forall m, n : \text{nat}. -(n = n + \text{succ}(m)) \rangle >$$

It is trivial to show that

$$\text{Models}(NSP) = \{ A \mid A \cong \text{NAT} \}$$

23
since one can prove that there exists an isomorphism between any model of
\(N\mathcal{S}P\) and the initial model of \(\mathcal{N}\mathcal{S}\mathcal{P}\) which is isomorphic to \(N\mathcal{A}\mathcal{T}\).

2.1.3 The proof system \(\Pi_{\mathcal{H}OL}^{\mathcal{RBA}\mathcal{SL}}\)

Before presenting the proof system \(\Pi_{\mathcal{H}OL}^{\mathcal{RBA}\mathcal{SL}}\) associated to the specification
language \(\mathcal{RBA}\mathcal{SL}\), we present the proof system of the higher order logic \(\mathcal{HOL}\)
\(\Pi_{\mathcal{H}OL}(\Gamma)\). This proof system is split in three different kind of rules: derivation
rules, typing rules and proof rules which determine the \(\beta\)-equality. All of them
are standard rules presented in a natural deduction style.

**Definition 2.26** The natural deduction system \(\Pi_{\mathcal{H}OL}(\Gamma)\) is defined by the fol-
lowing set of rules for any \(\Gamma \in \mathcal{P}([\mathcal{S}c_{\mathcal{A}N\mathcal{S}}(\Sigma)]):\)

\[
\begin{align*}
\Gamma \cup \phi \Rightarrow x \phi' \\
\Rightarrow x \phi \supset \phi' \quad (\Box i)
\end{align*}
\]

\[
\begin{align*}
\Gamma \Rightarrow x \phi \supset \phi' \quad \Gamma \Rightarrow x \phi \\
\Rightarrow \exists x \phi \quad (\Box e)
\end{align*}
\]

\[
\begin{align*}
\Gamma \Rightarrow \exists x \phi \quad \\
\Rightarrow \Gamma \forall x : \tau \phi \quad (\forall i)
\end{align*}
\]

\[
\begin{align*}
\Gamma \Rightarrow \forall x : \tau \phi \quad X \triangleright t : \tau \\
\Rightarrow \Gamma \Rightarrow x \phi[t/x] \quad (\forall e)
\end{align*}
\]

\[
\begin{align*}
\Gamma \Rightarrow x \phi \quad \phi \equiv_{x, \beta} \phi' \\
\Rightarrow \Gamma \Rightarrow x \psi \quad (CONV)
\end{align*}
\]

where the set of typing rules of this proof system is:

\[
\begin{align*}
X \triangleright x_r : \tau \\
x_r \in X_r & \quad (ASS)
\end{align*}
\]

\[
\begin{align*}
X \triangleright t_1 : s_1 & \quad \ldots \quad X \triangleright t_n : s_n \\
X \triangleright f(t_1, \ldots, t_n) : s & \quad s : s_1 \times \ldots \times s_n \rightarrow s \in \Sigma & \quad (APPL)
\end{align*}
\]

\[
\begin{align*}
X \triangleright \lambda(x_1 : \tau_1, \ldots, x_n : \tau_n) \phi & \quad \phi : \mathcal{P}rop \\
X \triangleright \lambda(x_1 : \tau_1, \ldots, x_n : \tau_n) \phi : [\tau_1, \ldots, \tau_n] & \quad (\lambda A B S)
\end{align*}
\]

\[
\begin{align*}
X \triangleright t_1 : \tau & \quad \ldots \quad X \triangleright t_n : \tau_n \\
X \triangleright t : [\tau_1, \ldots, \tau_n] & \quad X \triangleright t(t_1, \ldots, t_n) : \mathcal{P}rop \quad (\lambda A P P L)
\end{align*}
\]

\[
\begin{align*}
X \cup x : \tau \phi & \quad \phi : \mathcal{P}rop \\
X \triangleright \forall x : \tau \phi : \mathcal{P}rop & \quad (\forall)
\end{align*}
\]

\[
\begin{align*}
X \triangleright \phi : \mathcal{P}rop \quad X \triangleright \phi' : \mathcal{P}rop \\
X \triangleright \phi \supset \phi' : \mathcal{P}rop & \quad (\Box)
\end{align*}
\]
and the \( \beta \)-equality rules are:

\[
\begin{align*}
\frac{x_\tau = \beta, X \ x_\tau}{x \in X_\tau} \quad & (\text{Var}_\tau) \\
\frac{X \triangleright t_1 : s_1 \ldots X \triangleright t_n : s_n}{t_1 = \beta, X \ t_1' \ldots t_n = \beta, X \ t_n'} \quad & (t_1, \ldots, t_n) = \beta, X \ f(t_1', \ldots, t_n') \ f : s_1 \times \ldots \times s_n \to s_n \in \Sigma \quad & (\text{Term}_\tau) \\
\frac{X \triangleright t_1 : X \triangleright t_1 : \ldots \triangleright t_n : t_n}{t_1 = \beta, X \ t_1' \ldots t_n = \beta, X \ t_n'} \quad & (\text{App}_\tau) \\
\frac{X \triangleright \lambda(x_1 : \tau_1, \ldots, x_n : \tau_n) \phi : [\tau_1, \ldots, \tau_n]}{\lambda(x_1 : \tau_1, \ldots, x_n : \tau_n) \phi \{t_1/x_1\} \ldots \{t_n/x_n\} = \beta, X \ \phi'} \quad & (\text{Lambda}_\tau) \\
\frac{X \cup \{x : \tau\} \triangleright \phi : \text{Prop}}{\forall x : \tau. \phi = \beta, X \ \phi'} \quad & (\text{Forall}_\tau) \\
\frac{\phi' : \text{Prop}}{\phi = \beta, X \ \phi'} \quad & (\text{Sym})
\end{align*}
\]

\textbf{Definition 2.27} The encoding of the logical operators \textsc{false}, \textsc{true}, =, \vee, \wedge, \exists \ is as follows:

\[
\begin{align*}
\text{false} &= _{\text{def}} \forall P : \text{Prop}. P \\
\text{true} &= _{\text{def}} \forall P : \text{Prop}. P \supset P \\
t =_\tau r &= _{\text{def}} \forall P : [\tau]. P \supset P \ r \\
\phi \land \phi' &= _{\text{def}} \forall P : \text{Prop}. (\phi \land \phi' \supset P) \supset P \\
\phi \lor \phi' &= _{\text{def}} \forall P : \text{Prop}. (\phi \lor \phi' \supset P) \supset P \\
\exists x : \tau. \phi &= _{\text{def}} \forall P : \text{Prop}. ((\exists x : \tau. \phi) \supset P) \supset P
\end{align*}
\]
Proposition 2.28 The following set of rules is admissible in the proof system \( \Pi_{HOL}(\Gamma) \) for any environment \( \Gamma \in \mathcal{P}([\text{Sen}_{\text{AINS}}(\Sigma)]) \):

\[
\begin{align*}
\Gamma &\vdash X \text{ true} \\
\Gamma &\vdash X \text{ false} \\
\Gamma \& \Gamma_1 \& \Gamma_2 &\vdash X \phi_1 \land \phi_2 \\
\Gamma &\vdash X \phi_1 \\
\Gamma &\vdash X \phi_2
\end{align*}
\]

\( (T) \)

\( (F) \)

\( (\land E) \)

\( (\land I) \)

\( (\lor I) \)

\( (\lor E) \)

\( (\exists I) \)

\( (\exists E) \)

\[\Gamma \vdash X \phi \]

\[\Gamma \vdash X \psi \]

\[\Gamma \vdash X \neg \phi \]

\[\Gamma \vdash X \neg \psi \]

\( (\neg I) \)

\( (\neg E) \)

In the next definitions, we present different relational signatures which extend a given signature with the following symbols:

- Symbols to denote the indistinguishability relation for every sort of the signature.

- The same extension as in the previous one plus symbols to denote a pseudopeormorphism between the original signature and a disjoint copy.

Definition 2.29 The relational signature \( \Sigma[\sim] \) is defined for any signature \( \Sigma = (S, Op) \) and for any \( S \)-family of new symbols \( \sim \) as:

\[\Sigma[\sim] = \Sigma \cup \{\sim_s : s \times s | s \in S\}\]
**Definition 2.30** The relational signature \( \Sigma[\sim, \pi_{C_{\text{op}}}] \) for any signature \( \Sigma = (S, Op) \), for any bijective signature morphism \( \text{Copy} : \Sigma \to \text{Copy}(\Sigma) \) such that \( \Sigma \cap \text{Copy}(\Sigma) = \emptyset \) and for any \( S \)-family of new symbols \( \sim \) and \( \pi \) is defined as:

\[
\Sigma[\sim, \pi_{C_{\text{op}}}] = \Sigma[\sim] \cup \text{Copy}(\Sigma) \cup \{ \pi_s : s \to \text{Copy}(s) | s \in S \}
\]

**Remark:** The relational signature \( \text{Copy}(\Sigma)[\sim, \pi_{C_{\text{op}}-1}] \) stands for the following signature:

\[
\text{Copy}(\Sigma)[\sim, \pi_{C_{\text{op}}-1}] = \text{Copy}(\Sigma)[\sim] \cup \Sigma \cup \{ \pi_s : \text{Copy}(s) \to s | s \in S \}
\]

Here we present some axioms used to define the proof system for the reachability operator:

**Definition 2.31** The set of sentences \( \text{Reach}_{ax}[\Sigma, (S_R, F_R)] \) for any reachability constraint \( (S_R, F_R) \) of \( \Sigma \) is defined as follows:

\[
\text{Reach}_{ax}[\Sigma, (S_R, F_R)] = \{ \forall p : [s].(\forall f : s_1 \times \ldots \times s_n \to s \in F_R \}
\]

\[
(\forall i \in [1..n], i \neq s) \cup P(x_i) \cup P(f(x_1, \ldots, x_n)) \cup \forall x : s.P(x) | s \in F_R \}
\]

In the following, there are some definitions which axiomatise the indistinguishability relation in higher-order logic. This axiomatisation contains the following axioms:

- Axioms to determine which a given value is reachable.
- Axioms to determine which a given relation coincide with the set-theoretical equality.
- Axioms to determine that a family of relations is a congruence.

**Definition 2.32** The Sorts(\( \Sigma \))-family of sentences \( D[\Sigma, \text{In}] \) for any \( \text{In} \subseteq \text{Sorts}(\Sigma) \) is defined as follows:

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

where

\[
D_s[\Sigma, \text{In}] = \lambda x : s.(\forall p : [s].(\forall f : s_1 \times \ldots \times s_n \to s \in Op_\Sigma(\Sigma) \}
\]

\[
(\forall i \in [1..n], i \neq s) \cup P(x_i) \cup P(f(x_1, \ldots, x_n))) \cup P(x) \text{, if } s \in S - \text{In}
\]

\[
D_s[\Sigma, \text{In}] = \lambda x : s.\text{true} \text{, if } s \in \text{In}
\]
**Definition 2.33** The set of sentences Indist\(_\mathcal{rel}([S, Op][\sim], Obs, In]\) is defined as follows:

\[
\text{Indist}_{\mathcal{rel}}([S, Op][\sim], Obs, In] = \{ \text{Indist}_{\mathcal{rel}}([S, Op][\sim], Obs, In, s] \mid s \in S \}
\]

where

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

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

\[
\text{OBSEQ}([S, Op], R, Obs, In] = \bigwedge_{obs \in Obs} \forall v : obs. \forall w : obs.
\]

\[
D_{obs}[\Sigma, In](v) \land D_{obs}[\Sigma, In](w) \Rightarrow (R_{obs}(v, w) \Leftrightarrow v = w)
\]

\[
\text{CONG}([S, Op], R, Obs, In] = \bigwedge_{\Sigma \in \text{Obs}} \forall x_1 : s_1, \ldots \forall x_n : s_n. \forall y_1 : s_1, \ldots \forall y_n : s_n.
\]

\[
(R_{s_1}(x_1, y_1) \land \ldots \land R_{s_n}(x_n, y_n)) \Rightarrow
\]

\[
R_s(f(x_1, \ldots, x_n), f(y_1, \ldots, y_n)).
\]

Here, we present an axiomatisation of a pseudoepimorphism over the signature \(\Sigma[\sim, \pi_{C_{opg}}]\) in higher-order logic. This axiomatisation is equivalent to the one presented in first-order logic.

**Definition 2.34** The set of sentences pEpip\(_{HOL}([\Sigma[\sim, \pi_{C_{opg}}], Obs, In]\) is defined as:

\[
pEpip_{HOL}([\Sigma[\sim, \pi_{C_{opg}}], Obs, In] = \text{Hom}([\Sigma[\sim, \pi_{C_{opg}}], Obs, In] \cup \text{Epip}([\Sigma[\sim, \pi_{C_{opg}}], Obs, In] \cup \sim \text{comp}([\Sigma[\sim, \pi_{C_{opg}}], Obs, In]
\]

where

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

\[
\bigwedge_{i \in [1..n]} D_{s_i}[\Sigma, In](t_i) \Rightarrow
\]

\[
\pi_{C_{opg}}(f(t_1, \ldots, t_n)) = \text{Copy}(f)(\pi_{C_{opg}}(s_1(t_1), \ldots, \pi_{C_{opg}}(s_n(t_n))).
\]

28
\[
\text{Epithom}[\Sigma [\sim, \pi_{\text{Copy}}], \text{Obs}, \text{In}] = \\
\bigcup_{s \in S} \{ y : \exists x : s.D_s[\Sigma, \text{In}](x) \land \pi_{\text{Copy}, s}(x) = y \}
\]

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

In the following definitions, we define specification expressions using an extended version of RBASL with relational signatures and we prove some equivalences between the semantics of some of these specification expressions and the semantics of the behavioural operator. The semantics of RBASL with relational signatures is extended in the obvious way basically just replacing signatures by relational signatures. The specification expressions defined below are used to establish an equivalence between the semantics of behavioural operators and structured specifications which are the basis to define the proof system \( \Pi_{RBASL} \).

**Definition 2.35** The specification expression \( \text{BSP}[\Sigma [\sim, \pi_{\text{Copy}}], \text{Obs}, \text{In}] \) for any signature \( \Sigma = (S, \text{Op}) \), for any bijective signature morphism \( \text{Copy} : \Sigma \to \text{Copy}(\Sigma) \) such that \( \Sigma \cap \text{Copy}(\Sigma) = \emptyset \) and for any \( S \)-family of new symbols \( \sim \) and \( \pi \) is defined as:

\[
\text{BSP}[\Sigma [\sim, \pi_{\text{Copy}}], \text{Obs}, \text{In}] = \langle \Sigma [\sim, \pi_{\text{Copy}}], \\
\{ \forall x : s. \forall y : s.x \sim s.y \Leftrightarrow \text{Indist}_{\text{rel}}[\Sigma [\sim], \text{Obs}, \text{In}, s](x, y) \mid s \in S \} \cup \\
\text{pEpi}[\Sigma [\sim, \pi_{\text{Copy}}], \text{Obs}, \text{In}] \rangle
\]

**Definition 2.36** For any specification expression \( \text{SP} \) with signature \( \Sigma = (S, \text{Op}) \), for any signature morphism \( \text{Copy} : \Sigma \to \text{Copy}(\Sigma) \) and \( \text{Copy}' : \Sigma \to \text{Copy}'(\Sigma) \) such that \( \Sigma \cap \text{Copy}(\Sigma) = \emptyset \) and \( \Sigma \cap \text{Copy}'(\Sigma) = \emptyset \), for any \( S \)-family of new symbols \( \sim \) and \( \pi \), the specification expression \( \text{GBSP}[\text{SP}, \text{Copy}, \Sigma [\sim, \pi_{\text{Copy}}], \text{Obs}, \text{In}] \) is defined as:

\[
\text{GBSP}[\text{SP}, \text{Copy}, \Sigma [\sim, \pi_{\text{Copy}}], \text{Obs}, \text{In}] = \\
\text{rename } \text{SP} \text{ by } \text{Copy} + \text{BSP}[\Sigma [\sim, \pi_{\text{Copy}}], \text{Obs}, \text{In}]
\]
Lemma 2.37 Let $R_{A, \phi, s}^{obs, 1n}$ be a $S$-family of partial congruences which satisfies the following condition:

$$\forall obs \in \text{Obs}, \forall v, w \in A_{\phi, s}[X_{\phi, s}], (v R_{A, \phi, s}^{obs, 1n} w \iff v = w)$$

then

$$\forall s \in S, \forall v, w \in A[X_{\phi, s}], v R_{A, s}^{obs, 1n} w \implies v \approx_{A, s}^{obs, 1n} w$$

Proof sketch:
It follows by context induction. The proof of the general case uses that $R$ is a $S$-family of partial congruences which coincides with the set theoretical equality for observable sorts.

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

Lemma 2.38 The sentence Indist $\langle \Sigma, \text{Op} \rangle [\sim], \text{Obs}, \text{In}, s]$ for any sort $s \in S$ and for any free variables $x, y \in X$, satisfies the following condition which we will refer as the indistinguishability condition:

$$\forall s \in S, \forall A \in \text{Alg}(\Sigma), \forall \rho \in \{x, y\} \to A.$$

$$[\text{Indist}_{\langle \Sigma, \text{Op} \rangle}[\sim], \text{Obs}, \text{In}](x, y)]\rho, A \iff \rho(x) \approx_{A, s}^{obs, 1n} \rho(y)$$

Theorem 2.39 For any specification expression $SP$ with signature $\Sigma = (S, \text{Op})$, for any signature morphism $\text{Copy} : \Sigma \to \text{Copy}(\Sigma)$ such that $\Sigma \cap \text{Copy}(\Sigma) = \emptyset$, for any $S$-family of symbols $\sim$ and $\pi$ the two following equivalences between specification expressions hold:

**behaviour $SP$ wrt $\approx = \text{GBSP}[SP, \text{Copy}, \Sigma[\sim, \pi_{\text{Copy}}], \text{Obs}, \text{In}]_{\Sigma}$$

$SP/ \approx = \text{GBSP}[SP, \text{Copy}, \text{Copy}(\Sigma)[\sim, \pi_{\text{Copy}}], \text{Copy}(\text{Obs}), \text{Copy}(\text{In})]_{\Sigma}$

Proof:

- **behaviour $SP$ wrt $\approx = \text{GBSP}[SP, \text{Copy}, \Sigma[\sim, \pi_{\text{Copy}}], \text{Obs}, \text{In}]_{\Sigma}$.**

Assume that $\text{BEHSP} = \text{BSP}[\Sigma[\sim, \pi_{\text{Copy}}], \text{Obs}, \text{In}]_{\Sigma}$. We differentiate two cases:

- $A \in \text{Models}(\text{behaviour } SP \text{ wrt } \approx) \implies A \in \text{Models}(\text{GBSP}[SP, \text{Copy}, \Sigma[\sim, \pi_{\text{Copy}}], \text{Obs}, \text{In}]_{\Sigma})$.

Assume that $A \in \text{Models}(\text{behaviour } SP \text{ wrt } \approx)$. The proposition which we have to prove is equivalent to

$$\exists B \in \text{Models}(\text{GBSP}[SP, \text{Copy}, \Sigma[\sim, \pi_{\text{Copy}}], \text{Obs}, \text{In}]_{\Sigma}).B|_{\Sigma} = A$$

To prove this proposition we will build an algebra $B \in \text{Models}(\text{GBSP}[SP, \text{Copy}, \Sigma[\sim, \pi_{\text{Copy}}], \text{Obs}, \text{In}]_{\Sigma})$ such that $B|_{\Sigma} = A$. The algebra $B$ is defined as follows:
* $B|\Sigma = A$.

* $B|_{C_{\text{copy}}(\Sigma)} = A/\cong_A^{\text{Obs,In}}|_{C_{\text{copy}}(\Sigma)}$.

* $B_s = \epsilon_A s$ for every sort $s \in S$, where

  \[ \epsilon_A s(v) = [v] \cong_A^{\text{Obs,In}} s \forall v \in A_s. \]

* $B_s = \cong_A^{\text{Obs,In}} s$ for every sort $s \in S$

To show that $B \in \text{Models}(\text{GBSP}[\text{SP, Copy, } \Sigma[\sim, \pi_{C_{\text{copy}}}], \text{Obs, In}])$ we have to show that $B|_{C_{\text{copy}}(\Sigma)} \cong_{C_{\text{copy}}} \in \text{Models}(\text{SP})$ which obviously holds since $B|_{C_{\text{copy}}(\Sigma)} \cong_{C_{\text{copy}}} = A/\cong_A^{\text{Obs,In}}$ and $A/\cong_A^{\text{Obs,In}} \in \text{Models}(\text{SP})$.

We have to show also that

\[ B \in \text{Models}(\langle \Sigma[\sim, \pi_{C_{\text{copy}}}], \text{Indist}_{\text{re}}[\Sigma[\sim, \text{Obs, In}], \text{In}] \cup pEpi[\Sigma[\sim, \pi_{C_{\text{copy}}}], \text{Obs, In}] >) \]

This proposition holds because $B \models \text{Indist}_{\text{re}}([\Sigma[\sim, \text{Obs, In}])$ and $B \models pEpi[\Sigma[\sim, \pi_{C_{\text{copy}}}], \text{Obs, In}]$.

$B \models \text{Indist}_{\text{re}}([\Sigma[\sim, \text{Obs, In}])$ holds since for every sort $s B_s = \cong_B^{\text{Obs,In}} s$ and $\text{Indist}_{\text{re}}([\Sigma[\sim, \text{Obs, In}])$ satisfies the indistinguishability condition, and $B \models pEpi[\Sigma[\sim, \pi_{C_{\text{copy}}}]$ holds since for every sort $s$, $B_s$ can be seen as an epimorphism between $A$ and $A/\cong_A^{\text{Obs,In}}$ and therefore $B$ satisfies $\text{Hom}[\Sigma[\pi_{C_{\text{copy}}}], \text{Obs, In}], \text{Epithom}[\Sigma[\pi_{C_{\text{copy}}}], \text{Obs, In}]$ and also $\sim \sim \text{comp}[\Sigma[\sim, \pi_{C_{\text{copy}}}], \text{Obs, In}].$

\[ \text{Assume that } A \in \text{Models}(\text{GBSP}[\text{SP, Copy, } \Sigma[\sim, \pi_{C_{\text{copy}}}], \text{Obs, In}]|_{\Sigma}). \]

This proposition can be rewritten to

\[ \exists B \in \text{Models}(\text{GBSP}[\text{SP, Copy, } \Sigma[\sim, \pi_{C_{\text{copy}}}], \text{Obs, In}]).B|\Sigma = A \]

Let $B$ be a Signature (GBSP$[\text{SP, Copy, } \Sigma[\sim, \pi_{C_{\text{copy}}}], \text{Obs, In}]$)-algebra such that $B \in \text{Models}(\text{GBSP}[\text{SP, Copy, } \Sigma[\sim, \pi_{C_{\text{copy}}}], \text{Obs, In}])$ and $B|\Sigma = A$. By the definition of Models(GBSP$[\text{SP, Copy, } \Sigma[\sim, \pi_{C_{\text{copy}}}], \text{Obs, In}]$) we have that $B \in \text{Models}(\text{BEHSP})$ and because of the indistinguishability condition together with the definition of $pEpi[\Sigma[\sim, \pi_{C_{\text{copy}}}]$ we have that

\[ \exists C \in \text{Alg}(\text{Copy}(\Sigma)).C|_{C_{\text{copy}}} \cong A/\cong_A^{\text{Obs,In}} \text{ and } B|_{C_{\text{copy}}(\Sigma)} = C \]

Let $C$ be a Copy($\Sigma$)-algebra such that $C|_{C_{\text{copy}}} \cong A/\cong_A^{\text{Obs,In}}$ and $B|_{C_{\text{copy}}(\Sigma)} = C$. By the definition of Models(GBSP$[\text{SP, Copy, } \Sigma[\sim, \pi_{C_{\text{copy}}}], \text{Obs, In}]$) and since $B|_{C_{\text{copy}}(\Sigma)} = C$ we have that $C \in$
Models (rename $SP$ by $Copy$) and since $C_{\text{copy}} \cong A/\equiv_{\text{obs}, In}^O$ and $SP$ is closed under isomorphism we have that $A/\equiv_{\text{obs}, In}^O \in Models(SP)$ and therefore

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

- $SP/ \approx = GBS[SP, Copy, Copy(\Sigma)[\sim, \pi_{\text{copy} - 1}], Copy(\text{Obs}), Copy(\text{In})]|\Sigma.$ Assume that $QGBSP = GBS[SP, Copy, Copy(\Sigma)[\sim, \pi_{\text{copy} - 1}], Copy(\text{Obs}), Copy(\text{In})]|\Sigma.$ We differentiate two cases:

  - $A \in Models(SP/ \approx) \Rightarrow A \in Models(QGBSP)$

Assume that $A \in Models(SP/ \approx).$ The proposition which we have to prove is equivalent to

$$\exists B \in Models(QGBSP). B|\Sigma = A$$

To prove this proposition we will build an algebra $B \in Models(QGBSP)$ such that $B|\Sigma = A.$ By the proposition $A \in Models(SP/ \approx)$ we know that

$$\exists A' \in \text{Alg}(\text{Signature}(SP)). A'/\approx_{A'}^{\text{obs}, \text{In}} \cong A \& A' \in Models(SP)$$

Let $A'$ be a $\text{Signature}(SP)$-algebra such that $A'/\approx_{A'}^{\text{obs}, \text{In}} \cong A$ and $A' \in Models(SP)$ and let $h: A'/\approx_{A'}^{\text{obs}, \text{In}} \to A$ be the isomorphism which relates each other. The algebra $B$ is defined as follows:

* $B|\Sigma = A.$
* $B|C_{\text{copy}}(\Sigma) = A'|C_{\text{copy} - 1}.$
* $B_{\pi_{\text{copy} - 1}(s)} = \epsilon_{A'|C_{\text{copy} - 1}, C_{\text{copy}}(s)}$ for every sort $s \in S.$ where
  $$\epsilon_{A', C_{\text{copy}}(s)}(v) = h([v]|_{C_{\text{copy}}(\text{Obs}), C_{\text{copy}}(\text{In})})$$
* $B_{\pi_{\text{copy} - 1}(s)} = \approx_{A'|C_{\text{copy} - 1}, C_{\text{copy}}(s)}$.

To show that $B \in Models(QGBSP)$ we have to show that $B|C_{\text{copy}}(\Sigma)|\Sigma \in Models(SP)$ which obviously holds since $B|C_{\text{copy}}(\Sigma)|C_{\text{copy}} = A'$ and we have to show also that

$$B \in Models(BSP[Copy(\Sigma)[\sim, \pi_{\text{copy} - 1}], Copy(\text{Obs}), Copy(\text{In})])$$

which holds since $B \models Indist rel[Copy(\Sigma)[\sim], Obs, In] \& B \models pEpi[\Sigma[\sim, \pi_{\text{copy} - 1}]].$
$B \models \text{Indist}_\text{rel}[\text{Copy}(\Sigma)[~],\text{Obs},\text{In}]$ holds since for every sort $s$

$B_{\text{Copy}(s)} = \approx_{\text{Copy}(\Sigma)}^{\text{Copy}(\text{Obs})}$ and $\text{Indist}_\text{rel}[\text{Copy}(\Sigma)[~],\text{Copy}(\text{Obs}),\text{Copy}(\text{In})]$

satisfies the indistinguishability condition. $B \models pEpi[\text{Copy}(\Sigma)[~],\pi_{\text{Copy}(\text{Obs})},\text{Copy}(\text{In})]$ holds since for every sort $s$, $B_{\text{Copy}(s)}$ can see an epimorphism between $A'$ and $A$ and therefore $B$ satisfies $\text{Hom}[[\Sigma][~],\pi_{\text{Copy}(\text{Obs})},\text{Obs},\text{In}]$ and also $\sim_{\text{comp}}[\Sigma][~],\pi_{\text{Copy}(\text{Obs})},\text{Obs},\text{In}]$.

$A \in \text{Models}(QGBSP) \Rightarrow A \in \text{Models}(SP/\equiv)$. Assume that $A \in \text{Models}(QGBSP)$. By this proposition we know that there exists a Signature($QGBSP$)-algebra $B$ such that $B \in \text{Models}(QGBSP)$ and $B[\Sigma] = A$. By the definition of Models($QGBSP$) we have that $B[\text{Copy}(\Sigma)]|_{\text{Copy}(\Sigma)} \in \text{Models}(SP)$ and together with the definitions of $\text{Indist}_\text{rel}[\text{Copy}(\Sigma)[~],\text{Copy}(\text{Obs}),\text{Copy}(\text{In})]$ and $pEpi[\text{Copy}(\Sigma)[~],\pi_{\text{Copy}(\text{Obs})},\text{Copy}(\text{In})]$, we have that $A \equiv B|_{\text{Copy}(\Sigma)|_{\text{Copy}(\Sigma)}}^{\text{copy}(\text{Inc})_{\equiv}}$. Therefore, by the definition of $\text{Models}(SP/\equiv_{\text{copy}(\text{Inc})_{\equiv}})$ we have that $A \in \text{Models}(SP/\equiv_{\text{copy}(\text{Inc})_{\equiv}})$.

**Definition 2.40** The proof system $\Pi_{HOL}^{\text{BSL}}$ is inductively defined for the specific operators of the language as follows:

\[
\text{Rules}(\Pi_{HOL}^{\text{BSL}}, \text{rename } SP \text{ by } \sigma) = \sigma^\sigma(\text{Rules}(\Pi_{HOL}^{\text{BSL}}, SP))
\]

\[
\text{Symbols}(\Pi_{HOL}^{\text{BSL}}, \text{rename } SP \text{ by } \sigma) = \sigma^\sigma(\text{Symbols}(\Pi_{HOL}^{\text{BSL}}, SP))
\]

\[
\Gamma_{\text{env}}(\Pi_{HOL}^{\text{BSL}}, \text{rename } SP \text{ by } \sigma) = \sigma^\sigma(\Gamma_{\text{env}}(\Pi_{HOL}^{\text{BSL}}, SP))
\]

\[
\text{Rules}(\Pi_{HOL}^{\text{BSL}}, \text{reach } SP \text{ with } (S_R, F_R)) = \text{Rules}(\Pi_{HOL}^{\text{BSL}}, SP)
\]

\[
\text{Symbols}(\Pi_{HOL}^{\text{BSL}}, \text{reach } SP \text{ with } (S_R, F_R)) = \text{Symbols}(\Pi_{HOL}^{\text{BSL}}, SP)
\]

\[
\Gamma_{\text{env}}(\Pi_{HOL}^{\text{BSL}}, \text{reach } SP \text{ with } (S_R, F_R)) = \Gamma_{\text{env}}(\Pi_{HOL}^{\text{BSL}}, SP) \cup \text{Reach}_{\text{BSL}}[\text{Signature}(SP), (S_R, F_R)]
\]

33
\[ \text{Rules}(\Pi_{\text{HOL}}^{\text{RBSL}}, \text{behaviour } SP \text{ wrt } \approx) = \]
\[ \text{Copy}''(\text{Rules}(\Pi_{\text{HOL}}^{\text{RBSL}}, SP)) \]
\[ \text{Symbols}(\Pi_{\text{HOL}}^{\text{RBSL}}, \text{behaviour } SP \text{ wrt } \approx) = \]
\[ \text{Copy}''(\text{Symbols}(\Pi_{\text{HOL}}^{\text{RBSL}}, SP) \cup \Sigma[\sim, \pi_{\text{Copy}}]) \]
\[ \Gamma_{\text{env}}(\Pi_{\text{HOL}}^{\text{RBSL}}, \text{behaviour } SP \text{ wrt } \approx) = \]
\[ \Gamma_{\text{env}}(\text{rename } SP \text{ by } \text{Copy}) \cup \]
\[ \Gamma_{\text{env}}(BSP[\Sigma[\sim, \pi_{\text{Copy}}], \text{Obs}, \text{In}]) \]

\[ \text{Rules}(\Pi_{\text{HOL}}^{\text{RBSL}}, \text{abstract } SP \text{ by } \equiv) = \]
\[ \text{Rules}(\Pi_{\text{HOL}}^{\text{RBSL}}, \text{behaviour } SP/ \approx \text{ wrt } \approx) \]
\[ \text{Symbols}(\Pi_{\text{HOL}}^{\text{RBSL}}, \text{abstract } SP \text{ by } \equiv) = \]
\[ \text{Symbols}(\Pi_{\text{HOL}}^{\text{RBSL}}, \text{behaviour } SP/ \approx \text{ wrt } \approx) \]
\[ \Gamma_{\text{env}}(\Pi_{\text{HOL}}^{\text{RBSL}}, \text{abstract } SP \text{ by } \equiv) = \]
\[ \Gamma_{\text{env}}(\Pi_{\text{HOL}}^{\text{RBSL}}, \text{behaviour } SP/ \approx \text{ wrt } \approx) \]

\[ \text{Rules}(\Pi_{\text{HOL}}^{\text{RBSL}}, SP/ \approx) = \text{Copy}''(\text{Rules}(\Pi_{\text{HOL}}^{\text{RBSL}}, SP)) \]
\[ \text{Symbols}(\Pi_{\text{HOL}}^{\text{RBSL}}, SP/ \approx) = \]
\[ \text{Copy}''(\text{Symbols}(\Pi_{\text{HOL}}^{\text{RBSL}}, SP)) \cup \text{Copy}(\Sigma)[\sim, \pi_{\text{Copy}} - 1] \]
\[ \Gamma_{\text{env}}(\Pi_{\text{HOL}}^{\text{RBSL}}, SP/ \approx) = \Gamma_{\text{env}}(\Pi_{\text{HOL}}^{\text{RBSL}}, \text{rename } SP \text{ by } \text{Copy}) \cup \]
\[ \Gamma_{\text{env}}(BSP[\text{Copy}(\Sigma)[\sim, \pi_{\text{Copy}} - 1], \text{Copy}(\text{Obs}), \text{Copy}(\text{In})]) \]

where $\Sigma$ is the signature of the argument specification $SP$ for every case and the overloaded function symbol $\sigma''$ is used here as the pushout of the following
\[
\text{diagram:}
\]

\[
\begin{array}{c}
\text{Sym}(\Pi^{RBASL}_{HOL}, SP) \xrightarrow{\sigma^\text{eq}} \text{PO}(i, \sigma^p) \\
\downarrow i \\
\Sigma \\
\downarrow \sigma \\
\Sigma' \\
\end{array}
\]

where \( i : \Sigma \hookrightarrow \text{Sym}(\Pi^{RBASL}_{HOL}, SP) \) and it is also used as the renaming function of environments and of the proof system \( \Pi^{RBASL}_{HOL}(SP) \) which use the pushout just described, and the overloaded symbol \( \text{Copy}'' \) is the pushout morphism of the following diagram:

\[
\begin{array}{c}
\text{Sym}(\Pi^{RBASL}_{HOL}, SP) \xrightarrow{\text{Copy}''} \text{PO}(i, \text{Copy}) \\
\downarrow i \\
\Sigma \\
\downarrow \text{Copy}'' \\
\Sigma' \\
\end{array}
\]

where \( i : \Sigma \hookrightarrow \text{Symbols}(\Pi^{RBASL}_{HOL}, SP) \) and

\[
\text{Copy}''(\text{Symbols}(\Pi^{RBASL}_{HOL}, SP)) \cap \Sigma[\sim, \pi_{\text{Copy}''}] = \text{Copy}''(\Sigma)
\]

The symbol \( \text{Copy}'' \) is also used as the renaming functions of environments and of set of rules using the pushout morphism just described with the same name.

**Theorem 2.41** The consequence relation \( \vdash_{\Pi^{RBASL}_{HOL}} \) is sound.

**Proof:**
The proof is by induction of specification expression in a similar way as the proof of soundness of \( \vdash_{\Pi^{RBASL}_{FOLEQ}} \), where the proofs for the cases of the behaviour and quotient operator, theorem 2.42 is used.

**Theorem 2.42** There is no sound and complete consequence relation of the form \( \vdash_{\Pi^{RBASL}_{HOL}} \) for RBASL.

**Proof:**
It follows in the same way as the incompleteness of \( \vdash_{\Pi^{RBASL}_{FOLEQ}} \).

### 2.2 The normal form approach

In this subsection, we present the consequence relation \( \vdash_{n_f,BASLN,\Sigma} \) which relates specification expressions \( SP \in \text{SP\!EX}(BASLN) \) of an arbitrary but fixed BASLN format specification language BASLN with a fixed but arbitrary behavioural algebraic institution BAINS with signature \( \Sigma \in [|\text{AlgSig}] \) and sentences \( \phi \in [\text{SenBAINS}(\Sigma)] \). The consequence relation is defined using a consequence relation for the behavioural algebraic institution BAINS which we will refer to as \( \vdash_{BAINS,\Sigma} \).

The consequence relation \( \vdash_{n_f,BAINS,\Sigma} \) is sound if \( \vdash_{BAINS,\Sigma} \) is sound and \( \vdash_{n_f,\Sigma} \) is sound and complete if \( \vdash_{BAINS,\Sigma} \) is sound and complete.
Definition 2.43 Assume that $\vdash_{BAINS, \Sigma}: P([Sen_{BAINS}(\Sigma)] \times \{Sen_{BAINS}(\Sigma)]$ where $BAINS$ is a behavioural algebraic institution and $\Sigma \in [\text{AlgSig}]$. The consequence relation $\vdash_{nf, BAILN, \Sigma}: S P E X(BAILN) \times \{Sen_{BAINS}(\Sigma)]$ where $BAILN$ is an arbitrary but fixed $BAILN_f$ specification language is defined as follows:

$$SP \vdash_{nf, BAILN, \Sigma} \phi \iff \Psi \vdash_{BAINS, \Sigma'} \phi$$

where $nf(SP) = \langle \Sigma', \Psi \rangle > \Sigma$.

Theorem 2.44 $\vdash_{nf, BAILN, \Sigma}$ is sound if $\vdash_{BAINS, \Sigma}$ is sound and $\vdash_{nf, BAILN, \Sigma}$ is sound and complete if $\vdash_{BAINS, \Sigma}$ is sound and complete.

3 Compositional proof systems

In this section we present the compositional proof system $\Pi^{SASL}$ for an $ASLker$ language which we will denote as $SASL$ with a fixed but arbitrary algebraic instution $AINS$. $SASL$ includes just a renaming operator apart from the common operators of $ASLker$ languages and see [BCH] or [Hen97] for an extension of the compositional proof system to observability operators.

Definition 3.1 $SASL$ is a $ASLker$ language with additionally the following operators:

- Signature(rename $SP$ by $\sigma) = \Sigma$
- Symbols(rename $SP$ by $\sigma) = \sigma'(\text{Symbols}(SP))$
- Models(rename $SP$ by $\sigma) = \{ A \in \text{Alg}(\Sigma)|A[\sigma] \in \text{Models}(SP) \}$

where $\sigma : \text{Signature}(SP) \rightarrow \Sigma$ is a bijective signature morphism and $\sigma'$ is the pushout morphism of the following diagram:

$$\begin{array}{c}
\text{Sym}(SP) \xrightarrow{\sigma'} \text{PO}(i, \sigma) \\
i \downarrow \quad \downarrow \sigma \\
\text{Sign}(SP) \xrightarrow{i} \Sigma
\end{array}$$

where $i$ is the inclusion $i : \text{Signature}(SP) \hookrightarrow \text{Symbols}(SP)$.

The proof system has just the sequent $SP \vdash_{\Pi^{SASL}} \phi$ where $SP \in SP EX(SASL)$ and $\phi \in Sen_{AINS}(\text{Signature}(SP))$ and uses the consequence relation $\vdash_{\Pi_{AINS}} : P([Sen_{AINS}(\Sigma)]) \times \{Sen_{AINS}(\text{Signature}(SP))\}$ It is defined by the following set of rules:

\begin{align*}
\text{(basic)} & \quad \Sigma, \{\phi_1, \ldots, \phi_n\} \vdash_{\Pi_{AINS}} \phi_i \mid \phi_i \in \langle \phi_1, \ldots, \phi_n \rangle \
\end{align*}

36
\[
\begin{array}{c}
SP \vdash_{\Pi_{A^\text{INS}}^{\text{ASL}}} \phi_1 \quad \ldots \quad SP \vdash_{\Pi_{A^\text{INS}}^{\text{ASL}}} \phi_n \quad \{\phi_1, \ldots, \phi_n\} \vdash_{\Pi_{A^\text{INS}}} \phi \\
\hline
\{ (pi) \}
SP \vdash_{\Pi_{A^\text{INS}}^{\text{ASL}}} \phi \\
\hline
\text{(suml)} \quad SP \vdash_{\Pi_{A^\text{INS}}^{\text{ASL}}} \phi \\
\quad SP + \Sigma SP' \vdash_{\Pi_{A^\text{INS}}^{\text{ASL}}} \phi, \\
\hline
\text{(sumr)} \quad SP \vdash_{\Pi_{A^\text{INS}}^{\text{ASL}}} \phi \\
\quad SP + \Sigma SP' \vdash_{\Pi_{A^\text{INS}}^{\text{ASL}}} \phi, \\
\hline
\text{(exp)} \quad SP \vdash_{\Pi_{A^\text{INS}}^{\text{ASL}}} \phi \\
\quad SP|\Sigma \vdash_{\Pi_{A^\text{INS}}^{\text{ASL}}} \phi \in \text{Sen}_{A^\text{INS}}(\Sigma), \\
\hline
\text{(ren)} \quad \text{rename } SP \text{ by } \sigma \vdash_{\Pi_{A^\text{INS}}^{\text{ASL}}} \phi \\
\end{array}
\]

One can prove soundness of the proof system with respect to the satisfaction relation between specifications and sentences and to prove completeness it is needed to assume some properties of the arbitrary but fixed algebraic institution and its associated consequence relation \( \vdash_{\Sigma, A^\text{INS}} \). See [Cen94] for a proof for the case of a first-order logic and [BCH] for a general proof using institutions.

4 Proof system for refinement

The proof system for the refinement of specifications of the \( BASL_{ker} \) specification language \( RBASL \) presented in previous section with a fixed but arbitrary algebraic institution \( A^\text{INS} \) \( \left( \Pi_{RBASL}^{\text{INS}} \right) \) has just the sequent \( SP \Rightarrow SPI \) for any specification expressions \( SP, SPI \in SP\text{EX}(RBASL) \) and uses the following definition:

**Definition 4.1** Let \( ASL \) be an \( ASL_{ker} \) specification language with an arbitrary but fixed algebraic institution \( A^\text{INS} \). Assume that \( SP, SP' \in SP\text{EX}(ASL) \). \( SP \) is a persistent extension of \( SP' \) (denoted by \( \text{PEXTOF}(SP, SP') \)) if the following condition holds:

- There exists an inclusion with arity \( \text{Signature}(SP') \hookrightarrow \text{Signature}(SP) \)
- \( \text{Models}(SP') = \text{Models}(SP) \upharpoonright \text{Signature}(SP') \).
This proof system is inductively defined by the abstract specification expression as follows:

\[
(\text{basic}) \quad \frac{\langle \Sigma, \Phi \rangle \ggg \text{SPI}}{\text{Signature}(\text{SPI}) = \Sigma \land (\text{SPI} \models \Phi)}
\]

\[
SP' \ggg \text{rename} \quad \frac{\text{SPI}|_{\text{inr}(\text{Signature}(\text{SPI}))}}{\text{by} \quad \text{inrsig}^{-1}}
\]

\[
SP \ggg \text{rename} \quad \frac{\text{SPI}|_{\text{inl}(\text{Signature}(\text{SPI}))}}{\text{by} \quad \text{inlsig}^{-1}}
\]

\[
(\text{sum}) \quad \frac{SP + \Sigma \ggg \text{SPI}'}{SP' \ggg \text{SPI}}
\]

\[
(\text{export}) \quad \frac{SP \ggg \text{SPI}'}{SP|_{\Sigma} \ggg \text{SPI}} \quad \text{Signature}(\text{SPI}) = \Sigma \land \text{PEXTOF}(\text{SPI}', \text{SPI})
\]

\[
(\text{reach}) \quad \frac{\text{reach} \; SP \; \text{with} \; (S_R, \mathcal{F}_R) \ggg \text{SPI}}{\text{Mod}(\text{SPI}) = (S_R, \mathcal{F}_R)}
\]

\[
(\text{rename}) \quad \frac{SP \ggg \text{rename} \; \text{SPI} \; \text{by} \; \sigma^{-1}}{\text{rename} \; SP \; \text{by} \; \sigma \ggg \text{SPI}}
\]

\[
(\text{behaviour}) \quad \frac{\text{behaviour} \; SP \; \text{wrt} \; \approx \ggg \text{SPI}}{SP \ggg \text{SPI}/ \approx}
\]

\[
(\text{abstract}) \quad \frac{\text{behaviour} \; SP \; \text{wrt} \; \approx \ggg \text{SPI} \quad \text{Bche}(SP)}{\text{abstract} \; SP \; \text{by} \; \equiv \ggg \text{SPI}}
\]

\[
(\text{quotient}) \quad \frac{SP \ggg \text{SPI}' \; \text{by} \; \equiv \ggg \text{SPI}}{SP/ \approx \ggg \text{SPI}} \quad \text{Cond}(SP, \text{SPI}, \text{SPI}')
\]

where

\[
\text{Cond}(SP, \text{SPI}, \text{SPI}') = (\text{Signature}(\text{SPI}) = \text{Signature}(SP) \land
\]

\[
\text{Mod}(\text{SPI}/ \approx) = \text{Mod}(\text{SPI})
\]

\[
\text{Bche}(SP) = \text{Models}(SP) \subset \text{Models}(\text{behaviour} \; SP \; \text{wrt} \; \approx)
\]

and

\[
\text{inl} : \text{Signature}(SP) \rightarrow \text{Signature}(SP) + \Sigma \text{Signature}(SP')
\]

and

\[
\text{inr} : \text{Signature}(SP') \rightarrow \text{Signature}(SP) + \Sigma \text{Signature}(SP')
\]
are the pushouts morphisms of $i : \Sigma \hookrightarrow \text{Signature}(SP)$ and $i' : \Sigma \hookrightarrow \text{Signature}(SP')$, 
$\text{inl}(\text{Signature}(SP))$, $\text{inr}(\text{Signature}(SP'))$ are the obvious subsignatures of $\text{Signature}(SP) + \Sigma$
$\text{Signature}(SP')$ and $\text{inlsign} : \text{Signature}(SP) \rightarrow \text{inl}(\text{Signature}(SP))$ and $\text{inrsign} :$
$\text{Signature}(SP') \rightarrow \text{inr}(\text{Signature}(SP'))$ are the obvious signature morphisms defined with the pushouts morphisms $\text{inl}$ and $\text{inr}$. The proof of the following theorem can be found in [BCH] and in [Hen97].

Theorem 4.2 For any specification expressions $SP, SPI \in \text{SPEX(RBASL)}$, 
$SP \rightsquigarrow SPI$ if and only if there exists a derivation of the sequent $SP \gg SPI$
$SPI \in \Delta_{\text{RBAISL}}(SP \gg SPI)$
References


