An Example of Heterogeneous Structured Specification: a Travel Agency [⋆]

Sophie Coudert, Pascale Le Gall and Gilles Bernot

L.a.M.I., Université d'Évry, Cours Monseigneur Roméro, 91025 Evry Cedex, France
{coudert,bernot,legall}@lami.univ-evry.fr

Résumé. Notre spécification formelle simplifiée d'une agence de voyage combine 7 modules et utilise 3 formalismes de spécification différents (équationnel finiment engendré, observationnel et non déterministe), une primitive d' oubli d'opérations et 5 ponts hétérogènes permettant de connecter les composants. Sur cette spécification, nous montrons comment mener une preuve hétérogène. Notre système de preuve utilise aussi bien les relations d'inférence associées aux formalismes que des relations d'inférence associées aux primitives de structurations. Ceci permet l' héritage de propriétés à travers la structure. Parmi ces primitives, nous en présentons une nouvelle [BCLG96]: un pont hétérogène pour passer d'un formalisme à un autre. Cet exemple illustre comment composer formalismes de spécification et primitives de structuration d'une manière générale.

Mots-clés: spécifications formelles, theorie de preuve, spécification hétérogène, preuve hétérogène, étude de cas, systèmes largement distribués.

Abstract. Our simplified formal specification of a travel agency combines 7 modules, and uses 3 different specification languages (with respectively finitely generated equational, behavioural and non deterministic semantics), 1 operation hiding primitive and 5 heterogeneous bridges to glue these heterogeneous components together. We demonstrate on this specification how a *heterogeneous proof* can be handled. Our proof system involves both the inference relations associated to the formalisms and property inheritance relations associated to the structuring primitives, among which a new one [BCLG96]: an heterogeneous bridge which allows to move from one formalism to another. This example illustrates how to combine formal specification primitives and formalisms in a general way.

Keywords: formal specification, proof theory, heterogeneous specification, heterogeneous proof, case study, widely distributed system, feature integration.

[⋆] This work was partly supported by the ESPRIT-IV Working Group 22704 ASPIRE, the ESPRIT-IV Working Group 23531 FIREworks and the French "PRC-GDR de programmation."

Introduction

Widely distributed systems are now mostly built by successive additions of “ready to use” already specified and implemented components or features. This new practice induces a growing interest for the introduction of *heterogeneity in formal specifications*, where modules have to cohabit with less hierarchical components (e.g., features [KC95]) and components are specified according to different specification languages.

The example explained in this article is a simplified specification of a travel agency. It remains representative enough to demonstrate all the mechanisms needed to perform a formal proof in such a heterogeneous setting. So, this example shows that heterogeneity in formal specifications and proofs is within the reach of the current researches in formal methods.

Section 1 briefly outlines the underlying heterogeneous framework and the definition of a heterogeneous proof system. This underlying theoretical approach consists in unifying the different notions of specification module, specification primitive and “heterogeneous bridge” by the definition of *specification generator* with associated proof mechanisms. Section 2 contains the informal specification in 2.1, some remainders in 2.2 about the three needed formal specification logics (equational, behavioural and non deterministic), the definition of four heterogeneous bridges in 2.4, and the heterogeneous formal specification itself in 2.3. Section 3 shows how to prove the optimal agency choice of a given operator for a given travel between two cities.

1 Theoretical Foundations

1.1 Heterogeneous Structured Specifications

Heterogeneous structured specifications are build on *homogeneous specification frameworks* and specification building primitives called *constructors*.

– Homogeneous specification frameworks are general logics [Mes89] except that, as in [BCLG96], the satisfaction condition and the $\vdash$-translation are required on signature isomorphisms only. These conditions are translated in the heterogeneous frame later on, in order to define modular constructors.
– Constructors generalize the common aspects of usual specification building primitives such as *use, forget, restrict, basic specifications* and so on. From the syntactic side, all of them take as input some specifications (with an associated signature) and return as output a global structured specification (whose signature is uniquely determined by the input signatures). From the semantic side, they characterize the models of the output specification from the ones of the input specifications, and from the logic side they define the proof mechanism to inherit properties through the specification structure.

Indeed, constructors encompass as well the heterogeneity of structured specifications. In our framework, the input signatures of a constructor can belong to different general logics. We consider a category *Sig* which is the union of all the signature categories of all the considered homogeneous specification frameworks.

Moreover, even if a tuple of specifications has exported signatures which cope with the input signatures of a constructor, the output specification is not necessarily acceptable. It means that *domains* of acceptable specifications have to be considered. We will call *feature* a “constructor” where no consideration of domain is taken into account. A feature is not necessarily suitable when added to some previously existing heterogeneous structured specifications, thus features have no associated proof mechanism to inherit properties.

Definition 1. *A* feature *is a tuple* (f, Sem_f)*, where* f *is the* feature denotation *provided with a profile* $\Sigma_1 \cdots \Sigma_n \to \Sigma$ *in* Sig^+ *and* Sem_f *is the* feature semantics*: a set of functions from* $mod(\Sigma_1) \times \cdots \times mod(\Sigma_n)$ *to* $mod(\Sigma)$

A set F of features being given, a *heterogeneous specification* is simply a well typed F-term and its semantics is obtained by the corresponding composition of the possible semantic functions of the occurring features. Then a *heterogeneous* structured *specification* on a set F of *constructors* will be a heterogeneous specification which satisfies the domain restrictions at each constructor occurrence.

Definition 2. *A* constructor *is a tuple* $(f : \Sigma_1 \cdots \Sigma_n \to \Sigma, Sem_f, Dom_f, \Vdash_f)$, *where* (f, Sem_f) *is a feature, the* constructor domain Dom_f *is a subset of* $\coprod_i mod(\Sigma_i)$, *and* $\Vdash_f$ *is a binary relation included in* $\mathcal{P}(\coprod_i sen(\Sigma_i)) \times sen(\Sigma)$, *sound with respect to* Dom_f *(according to the heterogeneous specification semantics described above).*

All classical homogeneous specification building primitives [Wir93,HST94] give rise to a set of constructors. For example, a specification module ΔP which uses two sub-specifications P_1 and P_2 is modelized in general by a constructor $f = use_{\Delta P}$ whose profile is $\Sigma_1\Sigma_2 \to \Sigma$ (where $\Sigma = \Delta\Sigma \cup \Sigma_1 \cup \Sigma_2$ whatever the union means). The deduction mechanism $\Vdash_f$ contains the direct translations of Σ_1- and Σ_2-sentences into Σ-sentences, as well as the axiom introductions from ΔP. The semantic side can be to the one of [Bid87] for instance. For such constructors f based on the *use* primitive, the satisfaction condition and the $\vdash$-translation are consequently retrieved.

A new kind of constructor is used in our example in order to allow heterogeneity (see section 2.4). It is based on morphisms of specification formalisms.

Definition 3.
Let $a = (Sig_a, sen_a, mod_a, \vdash_a, \models_a)$ *and* $b = (Sig_b, sen_b, mod_b, \vdash_b, \models_b)$ *be specification formalisms. A morphism* $\mu : a \to b$ *consists of*

– *A functor* $Tr_\mu : Sig_a \to Sig_b$, *called* signature transposition functor
– *A* Sig_a*-indexed family* Ext_μ *such that* $Ext_{\mu,\Sigma} : mod_a(\Sigma) \to mod_b(Tr_\mu(\Sigma))$ *are partial functions called* model extraction functions
– *A family* $\Vdash_\mu$ *indexed by* Sig_a, *such that* $\Vdash_{\mu,\Sigma}$ *is a binary relation included in* $\mathcal{P}(sen_a(\Sigma)) \times sen_b(Tr_\mu(\Sigma))$, *called* heterogeneous inference bridge

such that the following properties are satisfied:

– Ext_μ *is a (partial) natural transformation from* mod_a *to* $mod_b \circ Tr_\mu$
– $\Vdash_\mu$ *is monotonic*
– $\Vdash$-translation*: for any isomorphism* $\sigma : \Sigma \to \Sigma'$ *in* Sig_a, *any* $\Gamma \subseteq sen_a(\Sigma)$, *and any* $\varphi \in sen_b(Tr_\mu(\Sigma))$, $\Gamma \Vdash_{\mu,\Sigma} \varphi$ *if and only if* $\sigma(\Gamma) \Vdash_{\mu,\Sigma'} Tr_\mu(\sigma)(\varphi)$
– heterogeneous soundness*: for any* $\Sigma \in Sig_a$, *for any* $\Gamma \subseteq sen_a(\Sigma)$, *for any* $\varphi \in sen_b(Tr_\mu(\Sigma))$, *if* $\Gamma \Vdash_{\mu,\Sigma} \varphi$ *then for all* $M \in mod_a(\Sigma)$, *we have* $[M \models_a \Gamma \implies Ext_{\mu,\Sigma}(M) \models_b \varphi]$

The main peculiarity of this definition with respect to other notions of specification formalism morphisms [AC92,GB84,SS92,Mes89] is the covariance and partiality of Ext_μ, which is coherent with the needed covariance and domains of the constructor semantics.

Similarly to the other primitives, morphisms of specification formalisms allow to define a family of heterogeneous bridge constructors from a formalism to another one in a natural way (the profile of $f = het_{\mu,\Sigma}$ is $\Sigma \to Tr_\mu(\Sigma)$).

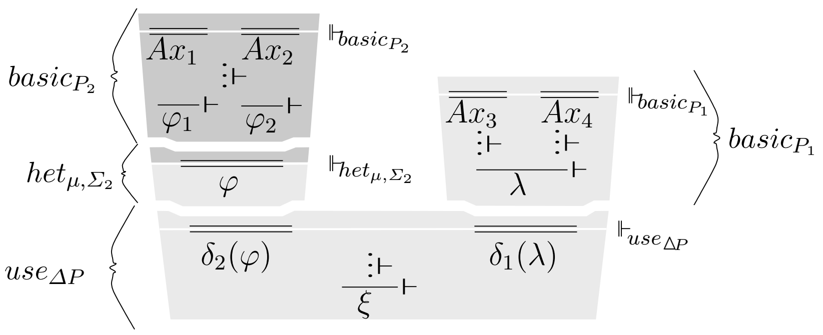
1.2 Proof system and term structure

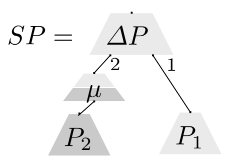
Our proof system ($\Vvdash$) for structured specifications recursively combines two kinds of inference steps:

– The homogeneous inferences allow to deduce properties about the model of a specification at a given level of the structure from properties about the same level, by using the corresponding homogeneous inference system $\vdash$.

– The structure inferences allow to deduce, from properties about imported specifications, properties about the exported one, by using the constructor inference bridges $\Vdash_f$. The structure inferences allow to extract properties through the structure of the specification.

With such steps, we can deduce a sentence ξ from the axioms of a heterogeneous specification SP, as illustrated on the following figure. We write $SP \Vvdash \xi$. Of course our proof system is sound[1] and ξ will be satisfied by all the models of SP.





$basic_{P_2} \Vvdash \varphi_1$ and $basic_{P_2} \Vvdash \varphi_2$
$basic_{P_1} \Vvdash \lambda$
$use_{\Delta P}(het_{\mu,\Sigma_2}(basic_{P_2}), basic_{P_1}) \Vvdash \delta_1(\lambda)$
$het_{\mu,\Sigma_2}(basic_{P_2}) \Vvdash \varphi$
$use_{\Delta P}(het_{\mu,\Sigma_2}(basic_{P_2}), basic_{P_1}) \Vvdash \delta_2(\varphi)$
$use_{\Delta P}(het_{\mu,\Sigma_2}(basic_{P_2}), basic_{P_1}) \Vvdash \xi$

This shape illustrates how the structure of the proof follows the term structure of the specification and introduces our conventional proof notations. The different intensities of grey represent different underlying homogeneous frameworks. The white interstices separate the input/output sides in the structure. Properties get over it with the $\Vdash_f$-relations (doubles lines). On this shape we inherit $\delta_1(\lambda)$ from P_1 and $\delta_2(\varphi)$ from P_2 to prove ξ in the ΔP module, using its homogeneous inference $\vdash$ (simple lines). The properties inherited from P_2, which are expressed in another formalism, have to pass through an inference bridge, corresponding to the heterogeneous constructor $het\mu, \Sigma_2$. We obtain not only a structured ([HST94,Wir93,HW]but also a heterogeneous inference system.

2 The Example

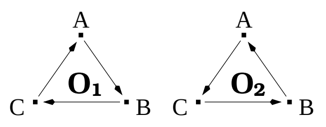
2.1 Informal specification

We specify a travel agency which selects for its customers the operator which potentially offers the less expensive path from a city to another city. We have classical modules for booleans and positive integers (**Bool** and **Int**). In a module **Map**, we specify cities and direct links between them. In a module **Netwk**, we specify networks as sets of links. Then, in a module **Path**, we define paths in a network as lists of links verifying certain properties. In a module **Oper** we specify two operators, each of them owning its own network. Finally, the agency is specified in the module **Agency**. Each module of this problem has its own “natural” specification language and semantics. We use three formalisms:

- Most of the modules (**Bool**, **Int**, **Map**, **Path**, **Agency**) are specified according to the classical equational logic [GTW78,EM85] with finitely generated semantics (denoted FG).
- The networks (**Netwk**) are sets. They are specified in a natural way according to a formalism with observational semantics [BHW95](denoted OBS), which is especially adequate for this.
- **Oper** is specified according to a formalism with non-deterministic semantics (denoted ND) [WM95]. Indeed, as in general routing problems, there can be several paths for a given travel, so that an operator proposes them in a non deterministic way.

[1] It comes from the soundness of both $\vdash$ and $\Vdash$

On this specification, we show how to perform a sound formal proof in such a heterogeneous setting. The property we demonstrate is that for a travel from the city **A** to the city **B**, the Agency will select the first operator ($\mathbf{O}_1$). The underlying intuition is trivial: an operator can only propose paths belonging to its own network, and $\mathbf{O}_1$ can propose a direct link from **A** to **B** which is impossible for $\mathbf{O}_2$, as we can see on the figure. The cost of a path is its length, to simplify things. So, the operator which offers the less expensive path is $\mathbf{O}_1$.



2.2 Homogeneous Specification Frameworks

In this section, we briefly recall the homogeneous specification frameworks we use in the specification.

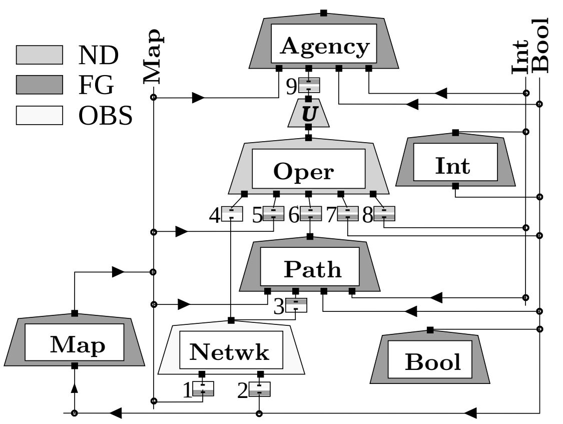
FG: We adopt a "locally finitely generated" framework where signatures $\Sigma = (G, S, \Omega, F)$ satisfy that (S, F) is a usual signature of equational logic as in [GTW78], $G \subseteq S$ is the set of locally finitely generated sorts, and $\Omega = \{\Omega_s\}_{s \in G}$ is the generator family, which is included in F. Sentences are defined as usual in equational logic (on (S, F)). Lastly, a Σ-algebra A is required to satisfy that $eval : \mathbf{T}_\Omega(\coprod_{s \in S \setminus G} A_s) \to A$ is surjective. Then, obviously, the corresponding homogeneous inference system is the equational calculus enriched with structural induction on Ω_s for $s \in G$.

OBS: The syntax is the one of classical equational logic [GTW78] except that each signature $\Sigma = (S, Obs, F)$ defines a set of observable sorts $Obs \subseteq S$. The satisfaction of equalities is defined "up to observation," i.e., $t \approx t'$ if and only if for any context c of sort in Obs, $c(t)$ is actually equal to $c(t')$. The corresponding homogeneous inference system is the equational calculus enriched with context induction as defined by Rolf Hennicker, e.g., in [Hen91].

ND: A sentence is a clause made of the three kinds of atoms described below. $t \doteq t'$ means that t and t' are deterministic and have the same result. $t \neq t'$ means that t and t' cannot have the same results. $t \prec t'$ means that all possible results of t are also possible results of t' (i.e., t is more deterministic than t'). The semantics is defined in such a way that a term t evaluates to a set of possible results. An algebra is said deterministic if any evaluation results in a singleton. Notice that for such algebras, the atoms $t \doteq t'$ and $t \prec t'$ are equivalent, and equivalent to the negation of $t \neq t'$. Lastly, the inference system is given in Annex A.

2.3 Structure of the formal specification

The structure of the formal specification has three FG-modules (**Int**, **Bool**, **Map**, **Agency**), one ND-module (**Oper**) and one OBS-module (**Netwk**), connected as shown on the figure. **Bool** is a *basic* module, the other ones are *use* modules. In order to connect heterogeneous components, we need heterogeneous bridges. For example, the bridge numbered 6, on the figure allows the **Oper** module to have a view of the **Path** module, which is FG while the operator is ND. Under the bridge numbered 9, between the **Agency** and the **Oper** modules, we have another example of structuring primitive: the *forget* constructor. It allows us to forget the nondeterministic operations of the **Oper** module, in order to retrieve deterministic algebras before the heterogeneous bridge. With our constructor representation, the specification and its sub-specifications can be denoted by terms. Let $\overline{\triangle P}$ denotes the specification with $\triangle P$ as top module, let het_i denote the bridge numbered i, and let $\triangle P$ denote the constructor $use_{\triangle P}$ (**Bool** denotes $basic_{\mathbf{Bool}}$) for short.



We have:

$$\overline{\mathbf{Bool}} = \mathbf{Bool},\ \overline{\mathbf{Int}} = \mathbf{Int}(\overline{\mathbf{Bool}}),\ \overline{\mathbf{Map}} = \mathbf{Map}(\overline{\mathbf{Bool}}),$$
$$\overline{\mathbf{Netwk}} = \mathbf{Netwk}(het_1(\overline{\mathbf{Map}}), het_2(\overline{\mathbf{Bool}}))$$
$$\overline{\mathbf{Path}} = \mathbf{Path}(\overline{\mathbf{Map}}, het_3(\overline{\mathbf{Netwk}}), \overline{\mathbf{Bool}}, \overline{\mathbf{Int}})$$
$$\overline{\mathbf{Oper}} = \mathbf{Oper}(het_4(\overline{\mathbf{Netwk}}), het_5(\overline{\mathbf{Map}}), het_6(\overline{\mathbf{Path}}), het_7(\overline{\mathbf{Bool}}), het_8(\overline{\mathbf{Int}}))$$
$$\overline{\mathbf{Agency}} = \mathbf{Agency}(\overline{\mathbf{Map}}, het_9(U(\overline{\mathbf{Oper}})), \overline{\mathbf{Bool}}, \overline{\mathbf{Int}})$$

2.4 Heterogeneous Bridges

The heterogeneous bridges from a homogeneous specification framework to another one are almost obvious for FG, OBS and ND (according to definition 3). Lest us very briefly outline the ones we use.

From OBS to FG (used for het_3): Tr_μ simply removes S_{Obs} from the signatures; $Ext_{\mu,\Sigma}$ is the quotient by the observational congruence $\equiv_{Obs}$ as defined in [BHW95]; $\Vdash_\mu$ leaves formulas unchanged (more formally, it is the reverse membership).

From FG to OBS (used for het_1 and het_2): Tr_μ makes any sort observable ($S_{Obs} = S$); $Ext_{\mu,\Sigma}$ is the identity; $\Vdash_\mu$ leaves formulas unchanged.

From FG to ND (used for het_5 to het_8): Tr_μ is the identity; $Ext_{\mu,\Sigma}$ turns a FG-algebra into a ND-algebra by translating the value v of an evaluation into the singleton $\{v\}$; $\Vdash_\mu$ translates any positive conditional formula into the corresponding clause, and ignores the other formulas.

From ND to FG (used for het_9): Tr_μ is the identity; $Ext_{\mu,\Sigma}$ is the reverse translation (from $\{v\}$ to v) on deterministic algebras and is undefined elsewhere; $\Vdash_\mu$ translates a clause into a positive conditional formula when possible, and ignores the other clauses.
Comment: before applying het_9, we apply a forgetful functor to the imported ND-specification, thus all remaining algebras are deterministic and Ext_9 is total. Indeed, this ensures the modular consistency of the heterogeneous structured specification. A detailed study about the consistency of heterogeneous structured specification can be found in [Cou].

From OBS to ND (used for het_4): simply the composition from OBS to FG and then to ND.

2.5 Main Modules

The modules **Int** and **Bool** are classical ones. The **Map** module simply specifies cities ($\mathbf{A}, \cdots$), links ($\mathbf{l}(\_,\_)$) between cities and the corresponding equalities; we only give its signature. We focus on the other specification modules. For **Netwk**, we specify sets of links with membership as observation ($\diamond$ being the infix insertion).

<u>**Map**, FG</u>

Inputs: 1:**Bool**
Sorts: *city*, *link*
Operations:
$^*\mathbf{l}() : city \times city \longrightarrow link;$
$^*\mathbf{A} :\longrightarrow city;\ ^*\mathbf{B} :\longrightarrow city;$
$^*\mathbf{C} :\longrightarrow city;\ ^*\mathbf{D} :\longrightarrow city;$
$\mathbf{eq}_c() : city \times city \longrightarrow bool;$
$\mathbf{eq}_l() : link \times link \longrightarrow bool;$

<u>**Netwk**, OBS</u>

Inputs: 1:**Map**, 2:**Bool**
Sorts: *net* **Obs:** *bool*
Operations:
$\emptyset :\longrightarrow net;\ \_ \diamond \_ : link \times net \longrightarrow net;$
$\_ \in \_ : link \times net \longrightarrow bool;$
Axioms:
1: $l \in \emptyset \approx \mathbf{F}$
2: $\mathbf{eq}_l(l,l') \approx \mathbf{T} \Rightarrow l \in l' \diamond s \approx \mathbf{T}$
3: $\mathbf{eq}_l(l,l') \approx \mathbf{F} \Rightarrow l \in l' \diamond s \approx l \in s$

Path, FG

Inputs:
1:**Map**, 2:**Netwk**, 3:**Path**, 4:**Bool**, 5:**Map**,

Sorts: *list*

Operations:

$*[\_] : \mathit{link} \longrightarrow \mathit{list};$

$*\_ :: \_ : \mathit{link} \times \mathit{list} \longrightarrow \mathit{list};$

$\mathbf{p}() : \mathit{list}, \mathit{city} \times \mathit{city} \times \mathit{net} \longrightarrow \mathit{bool};$

$\mathbf{\$}() : \mathit{list} \longrightarrow \mathit{nat};$

Axioms:

1: $\mathbf{p}([\mathbf{l}(c,c')], c_1, c'_1, n) \Rightarrow c = c_1$

2: $\mathbf{p}([\mathbf{l}(c,c')], c_1, c'_1, n) \Rightarrow c' = c'_1$

3: $\mathbf{p}([\mathbf{l}(c,c')], c_1, c'_1, n) = \mathbf{T} \Rightarrow \mathbf{l}(c,c') \in n$

4: $c = c_1 \wedge c' = c'_1 \wedge \mathbf{l}(c,c') \in n$
$\qquad \Rightarrow \mathbf{p}([\mathbf{l}(c,c')], c_1, c'_1, n)$

5: $\mathbf{p}(p, c', c'', n) \wedge \mathbf{l}(c,c') \in n$
$\qquad \Leftrightarrow \mathbf{p}(\mathbf{l}(c,c') :: p, c, c'', n)$

6: $\mathbf{\$}([l]) = \mathbf{s}(\mathbf{0})$ 7: $\mathbf{\$}(l :: p) = \mathbf{s}(\mathbf{\$}(p))$

Oper, ND

Inputs:
1:**Netwk**, 2:**Map**, 3:**Path**, 4:**Bool**, 5:**Int**

Sorts: *oper*

Operations:

$\mathbf{O}_1 :\longrightarrow \mathit{oper};$ $\mathbf{O}_2 :\longrightarrow \mathit{oper};$

$\mathbf{nw}() : \mathit{oper} \longrightarrow \mathit{net};$

$\mathbf{pt}() : \mathit{oper} \times \mathit{city} \times \mathit{city} \longrightarrow \mathit{list};$

$\widehat{\_,\_,\_} : \mathit{oper} \times \mathit{city} \times \mathit{city} \longrightarrow \mathit{nat};$

Axioms:

1: $\mathbf{O}_1 \doteq \mathbf{O}_1$ 2: $\mathbf{O}_2 \doteq \mathbf{O}_2$

3: $\mathbf{nw}(\mathbf{O}_1) \doteq \mathbf{l}(\mathbf{A},\mathbf{B}) \diamond \mathbf{l}(\mathbf{B},\mathbf{C}) \diamond [\mathbf{l}(\mathbf{C},\mathbf{A})] \diamond \emptyset$

4: $\mathbf{nw}(\mathbf{O}_2) \doteq \mathbf{l}(\mathbf{C},\mathbf{B}) \diamond \mathbf{l}(\mathbf{B},\mathbf{A}) \diamond [\mathbf{l}(\mathbf{A},\mathbf{C})] \diamond \emptyset$

5: $\mathbf{p}(\mathbf{pt}(o,c,c'), c, c', \mathbf{nw}(o))$

6: $[\mathbf{l}(\mathbf{A},\mathbf{B})] \prec \mathbf{pt}(\mathbf{O}_1, \mathbf{A}, \mathbf{B})$

7: $\mathbf{\$}(\mathbf{pt}(o,c,c')) \geq \widehat{o,c,c'}$

8: $\widehat{o,c,c'} \prec \mathbf{\$}(\mathbf{pt}(o,c,c'))$

Agency, FG

Inputs: 1:**Map**, 2:**Oper**, 3:**Bool**, 4:**Int**

Operations: $\mathbf{sl}() : \mathit{city} \times \mathit{city} \longrightarrow \mathit{oper};$

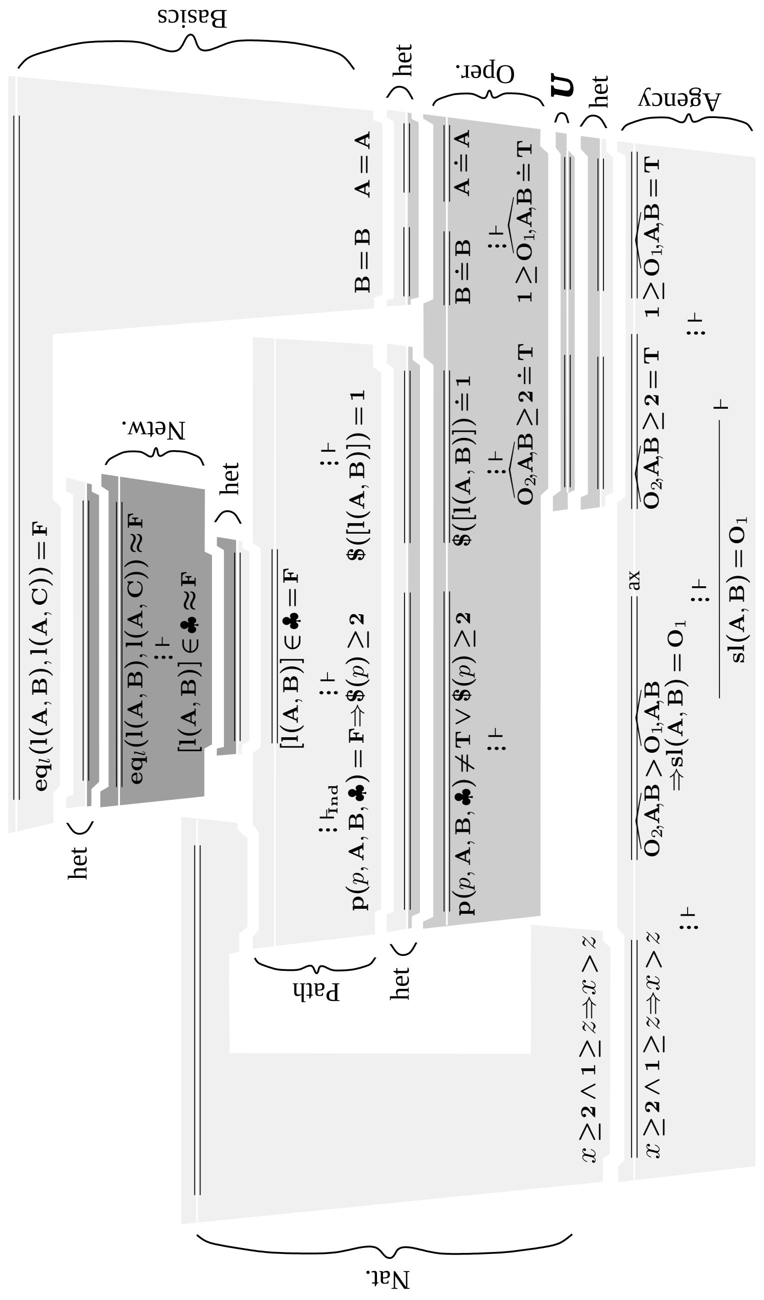
Axioms: 1:
$\widehat{\mathbf{O}_2,c,c'} > \widehat{\mathbf{O}_1,c,c'} \Leftrightarrow \mathbf{sl}(c,c') = \mathbf{O}_1$

Conventionally, a star before an operation name in a FG-signature means "belongs to Ω". Moreover, in order to increase legibility, we sometimes use a predicate notation for boolean terms. So we write $p(\boldsymbol{x})$ for $p(\boldsymbol{x}) = \mathbf{T}$ (resp. $p(\boldsymbol{x}) \approx \mathbf{T}$ in OBS or $p(\boldsymbol{x}) \doteq \mathbf{T}$ in ND), and so on.

With these conventions, we specify paths from a city to another one, in a network, as non empty lists of links such that the links belong to the network and are consecutive: $\mathbf{p}(p,c,c',n)$ means that p is a path from c to c' in the network n. Their cost ($\mathbf{\$}()$) is their length. This is done with the FG homogeneous framework. There are two operators ($\mathbf{O}_1, \mathbf{O}_2$), each of them having its own network ($\mathbf{nw}()$). They propose paths between cities with a non-deterministic operation $\mathbf{pt}()$ because there can be several possibilities. A function ($\widehat{\_,\_,\_}$) gives the best price an operator can propose for a given travel. For such a travel, the agency will select the operator with the best optimum. In our proof example, this operator is $\mathbf{O}_1$, who proposes a direct link (axiom 6). Notice that this optimum operation is deterministic; it will be preserved by the *forget* module $\boldsymbol{U}$, and then inherited by **Agency**.

3 A heterogeneous proof

The general aspect of the proof is drawn in the following figure.



This figure only mentions the main lemmas to get the result $\mathbf{sl(A,B)=O_1}$. It should hopefully help the reader to follow the formal proof given below.

Let $x\geq\mathbf{2}\wedge\mathbf{1}\geq z\Rightarrow x>z$ be denoted **@1** and $\mathbf{l(A,B)}\in\clubsuit\approx\mathbf{F}$ be denoted **@2**, where $\clubsuit$ denotes the network of $\mathbf{O}_2$: $\mathbf{l(C,B)}\diamond\mathbf{l(B,A)}\diamond[\mathbf{l(A,C)}]\diamond\emptyset$. We prove $\mathbf{sl(A,B)=O_1}$ in **Agency**, using the inherited properties $\mathbf{1}\geq\widehat{\mathbf{O_1},\mathbf{A}},\mathbf{B}\doteq\mathbf{T}$ denoted **@5** and using $\widehat{\mathbf{O_2},\mathbf{A}},\mathbf{B}\geq\mathbf{2}\doteq\mathbf{T}$ denoted **@11** which are proved later on in $\overline{\textbf{Oper}}$. (**@1** and **@2** are proved in [Cou])

The annotation to the right of each inference line follows the following conventions: *ax* means axiom introduction (resp. *as* when substituted); *u* means imported via a *use* constructor (resp. *h* for a heterogeneous bridge); the subscript of *ax* or *as* indicates the source module (its first character) and the superscript indicates the axiom number; the subscript of *u* indicates the top level module of the *use* constructor and the superscript indicates the number of the considered import slot; the subscript of *h* indicates the number of the bridge on the specification shape. Moreover, obvious sequences of elementary inference rules are contracted into one step. For example: the axioms are often introduced in an already instancied form; we exploit directly any useful consequence of modus ponens. We also abreviate $\mathbf{s(s(0))}$ as $\mathbf{2}$ (resp. $\mathbf{s(0)}$ as $\mathbf{1}$).

$$\dfrac{\dfrac{\dfrac{@1}{x \geq \mathbf{2} \wedge \mathbf{1} \geq z \Rightarrow x > z}\;u_A^4}{\widehat{\mathbf{O_2,A,B}} \geq \mathbf{2} \wedge \mathbf{1} \geq \widehat{\mathbf{O_1,A,B}} \Rightarrow \widehat{\mathbf{O_2,A,B}} > \widehat{\mathbf{O_1,A,B}}}\;\scriptstyle{SUB} \qquad \dfrac{@5}{\mathbf{1} \geq \widehat{\mathbf{O_1,A,B}} = \mathbf{T}}\;h_9u_A^2}{\widehat{\mathbf{O_2,A,B}} \geq \mathbf{2} \Rightarrow \widehat{\mathbf{O_2,A,B}} > \widehat{\mathbf{O_1,A,B}} \quad @12}\;\scriptstyle{MPG}$$

$$\dfrac{\dfrac{}{\widehat{\mathbf{O_2,A,B}} > \widehat{\mathbf{O_1,A,B}} \Rightarrow \mathbf{sl(A,B)} = \mathbf{O_1}}\;as_A^1 \qquad \dfrac{@12 \quad \dfrac{@11}{\widehat{\mathbf{O_2,A,B}} \geq \mathbf{2} = \mathbf{T}}\;h_9u_A^2}{\widehat{\mathbf{O_2,A,B}} > \widehat{\mathbf{O_1,A,B}}}\;\scriptstyle{MP}}{\mathbf{sl(A,B)} = \mathbf{O_1}}\;\scriptstyle{MP}$$

Proof of $\mathbf{1} \geq \widehat{\mathbf{O_1,A,B}} \doteq \mathbf{T}$ @5 in the **Oper**module.

$$\dfrac{\dfrac{}{\mathbf{\$(pt}(o,c,c')) \geq \widehat{o,c,c'}}\;ax_O^7 \quad \dfrac{\dfrac{}{\mathbf{A} = \mathbf{A}}\;\scriptstyle{REF}}{\mathbf{A} \doteq \mathbf{A}}\;h_5u_O^2 \quad \dfrac{\dfrac{}{\mathbf{B} = \mathbf{B}}\;\scriptstyle{REF}}{\mathbf{B} \doteq \mathbf{B}}\;h_5u_O^2 \quad \dfrac{}{\mathbf{O_1} \doteq \mathbf{O_1}}\;ax_O^1}{\mathbf{\$(pt(O_1,A,B))} \geq \widehat{\mathbf{O_1,A,B}} \quad @4}\;sub$$

$$\dfrac{\dfrac{@4 \quad \dfrac{}{\mathbf{[l(A,B)]} \prec \mathbf{pt(O_1,A,B)}}\;ax_O^6}{\mathbf{\$([l(A,B)])} \geq \widehat{\mathbf{O_1,A,B}}}\;rg3 \qquad \dfrac{\dfrac{}{\mathbf{\$([l(A,B)])} = \mathbf{1}}\;as_P^6}{\mathbf{\$([l(A,B)])} \doteq \mathbf{1}}\;h_6u_O^3}{\mathbf{1} \geq \widehat{\mathbf{O_1,A,B}} \quad @5}\;rg2$$

Proof of @11 in **Oper**, using @8 $\mathbf{p}(p,\mathbf{A},\mathbf{B},\clubsuit) \Rightarrow \mathbf{\$}(p) \geq \mathbf{2}$ (proof in Annex B).

$$\dfrac{\dfrac{}{\mathbf{p(pt}(o,c,c'),c,c',\mathbf{nw}(o))}\;ax_O^5 \quad \dfrac{}{\mathbf{A} \doteq \mathbf{A}}\;{\scriptstyle REF}\,h_5u_O^2 \quad \dfrac{}{\mathbf{B} \doteq \mathbf{B}}\;{\scriptstyle REF}\,h_5u_O^2 \quad \dfrac{}{\mathbf{O_2} \doteq \mathbf{O_2}}\;ax_O^2}{\mathbf{p(pt(O_2,A,B),A,B,nw(O_2))} \quad @6}\;sub$$

$$\dfrac{@6 \quad \dfrac{}{\mathbf{nw(O_2)} \doteq \clubsuit}\;ax_O^4}{\mathbf{p(pt(O_2,A,B),A,B},\clubsuit) \quad @7}\;rg2$$

$$\dfrac{@7 \quad \dfrac{\dfrac{@8}{\mathbf{p}(p,\mathbf{A},\mathbf{B},\clubsuit) \neq \mathbf{T} \vee \mathbf{\$}(p) \geq \mathbf{2}}\;h_6u_0^3 \quad \dfrac{}{\mathbf{pt(O_2,A,B)} \doteq \mathbf{pt(O_2,A,B)}}\;h_6u_O^3}{\mathbf{p(pt(O_2,A,B),A,B},\clubsuit) \neq \mathbf{T} \vee \mathbf{\$(pt(O_2,A,B))} \geq \mathbf{2}}\;sub}{\mathbf{\$(pt(O_2,A,B))} \geq \mathbf{2} \quad @9}\;cut$$

$$\dfrac{\dfrac{}{\widehat{o,c,c'} \prec \mathbf{\$(pt}(o,c,c'))}\;ax_0^8 \quad \dfrac{}{\mathbf{A} \doteq \mathbf{A}}\;{\scriptstyle REF}\,h_5u_O^2 \quad \dfrac{}{\mathbf{B} \doteq \mathbf{B}}\;{\scriptstyle REF}\,h_5u_O^2 \quad \dfrac{}{\mathbf{O_2} \doteq \mathbf{O_2}}\;ax_O^2}{\widehat{\mathbf{O_2,A,B}} \prec \mathbf{\$(pt(O_2,A,B))} \quad @10}\;sub$$

$$\dfrac{@10 \quad @9}{\widehat{\mathbf{O_2,A,B}} \geq \mathbf{2} \quad @11}\;rg3$$

4 Conclusion

The example developed here has been inspired by a more general routing problem in telecommunication networks. This agency example is of course considerably simplified with respect to the original problem, however it remains representative enough to illustrate our approach.

Our approach is resolutely both formal and pragmatic. We do not so much care about completeness, totality or exhaustiveness in our definitions. We preferably care about a good compromise between a sufficient power to prove what we need and a reasonable simplicity in order to preserve legibility as much as possible.

- *Completeness*: we do not require for a homogeneous inference system to be complete (e.g., $\vdash_{FG}$ is not complete) and we never ask for a heterogeneous bridge to be complete in any sense.
- *Totality*: we do not require for the morphisms of specification framework to completely map the power of one logic into another one (e.g., Ext_μ is a partial functor) and each constructor has a given domain, which means that it can be inadequate for some imported specifications.

– *Exhaustiveness*: the reader may have noticed that our heterogeneous bridges are rather simple and that it would have been possible to extract more formulas from a formalism to another one (e.g., the restriction to positive conditional formulas in the bridge from FG to ND is arbitrary, but it has the advantage to be easily understandable).

Our aim has been to show on an example that "cross-country" formal proofs are possible in a heterogeneous setting, while remaining perfectly sound.

Annex A

FG inference system: (for a signature $\Sigma = (G, S, \Omega, F)$)

$$\frac{}{\alpha \Rightarrow \alpha}\ {\scriptstyle TAU} \qquad \frac{}{t = t}\ {\scriptstyle REF} \qquad \frac{\Gamma \Rightarrow \alpha}{\Gamma \wedge \beta \Rightarrow \alpha}\ {\scriptstyle MON} \qquad \frac{\Gamma \Rightarrow t = t'}{\Gamma \Rightarrow t' = t}\ {\scriptstyle SYM}$$

$$\frac{\Gamma \Rightarrow t = t' \quad \Gamma \Rightarrow t' = t''}{\Gamma \Rightarrow t = t''}\ {\scriptstyle TRA} \qquad \forall \rho : T_\Sigma(V) \to T_\Sigma(V),\ \frac{\Gamma \Rightarrow \alpha}{\rho(\Gamma) \Rightarrow \rho(\alpha)}\ {\scriptstyle SUB}$$

$$\forall f \in \Sigma,\ \frac{\Gamma \Rightarrow t_i = t'_i\ (i = 1..n)}{\Gamma \Rightarrow f(t_1, \cdots, t_n) = f(t'_1, \cdots, t'_n)}\ {\scriptstyle REM} \qquad \frac{\Gamma \wedge \alpha \Rightarrow \beta \quad \Gamma \Rightarrow \alpha}{\Gamma \Rightarrow \beta}\ {\scriptstyle MP}$$

and the structural induction: for each set of sentences Φ and property ϕ,

$\forall s \in G,\ \dfrac{\Phi}{\phi(x_s)}\ {\scriptstyle IND_s}$ iff $\forall f \in \Omega_s,\ \dfrac{\Phi \cup \{\phi(x_1), \cdots, \phi(x_n)\}}{\phi(f(x_1, \cdots, x_n))}\ {\scriptstyle \Sigma \cup \{x_1, \cdots, x_n\}}$,

where $x_1, \cdots, x_n$ are the s-inputs of f

ND inference system:

Rules:

R1:	a- $\dfrac{}{x \neq y, x \doteq y}\ rg1$	b- $\dfrac{}{x \neq t, x \prec t}\ rg1$	$x, y \in \mathcal{V}$
R2:	$\dfrac{C^x_t \quad D, s \doteq t}{C^x_s, D}\ rg2$		
R3:	$\dfrac{C^x_t \quad D, s \prec t}{C^x_s, D}\ rg3$		x not in a right-hand side of $\prec$ in C.
R4:	$\dfrac{C, s \preceq t \quad D, s \neq t}{C, D}\ cut$		($\preceq$ being either $\doteq$ or $\prec$)
R5:	$\dfrac{C}{C, e}\ wea$		
R6:	$\dfrac{C, x \neq t}{\vdash C^x_t}\ eli$		$x \in \mathcal{V} - \mathcal{V}[t]$, at most one x in C

Derived rules:

PER:	a- $\dfrac{}{x \doteq x}\ per$	b- $\dfrac{}{t \prec t}\ per$	
REN:	$\dfrac{C}{C^x_y}\ ner$		
SUB:	$\dfrac{C \quad D, t \doteq t}{C^x_t, D}\ sub$		
INTR:	$\dfrac{C^y_t}{C^y_x, x \neq t}\ inc$		y not in a right-hand side of $\prec$

Annex B

Proof of $\mathbf{@8} : \mathbf{p}(p, \mathbf{A}, \mathbf{B}, \clubsuit) \Rightarrow \mathbf{\$}(p) \geq \mathbf{2}$

Let $\Phi = \{\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5, \varphi_6, \varphi_7, \varphi_8, \varphi_9, \varphi_{10}\}$, with

$$\overline{\overline{\mathbf{p}([\mathbf{l}(c, c')], \mathbf{A}, \mathbf{B}, \clubsuit) \Rightarrow c = \mathbf{A} \quad \varphi_1}}\ as^1_P \qquad \overline{\overline{\mathbf{p}([\mathbf{l}(c, c')], \mathbf{A}, \mathbf{B}, \clubsuit) \Rightarrow c' = \mathbf{B} \quad \varphi_2}}\ as^2_P$$

$$\overline{\overline{\mathbf{p}([\mathbf{l}(c, c')], \mathbf{A}, \mathbf{B}, \clubsuit) \Rightarrow [\mathbf{l}(c, c')] \in \clubsuit \quad \varphi_3}}\ as^3_P \qquad \frac{\mathbf{@2}}{\overline{\mathbf{l}(\mathbf{A}, \mathbf{B}) \in \clubsuit = \mathbf{F} \quad \varphi_4}}\ u^4_P$$

$$\overline{\overline{\mathbf{\$}(l :: p) = \mathbf{s}(\mathbf{\$}(p)) \quad \varphi_6}}\ ax^7_P \qquad \overline{\overline{\mathbf{\$}([l]) = \mathbf{s}(\mathbf{0}) \quad \varphi_7}}\ ax^6_P \qquad \overline{\overline{\mathbf{s}(\mathbf{s}(\mathbf{0})) \geq \mathbf{2} \quad \varphi_8}}\ as^2_I u^4_P$$

$$\overline{\overline{\mathbf{1} \geq \mathbf{2} = \mathbf{F} \quad \varphi_5}}\ as^4_I u^4_P \qquad \overline{\overline{x \geq y \wedge y \geq z \Rightarrow x \geq z \quad \varphi_9}}\ ax^5_I u^4_P \qquad \overline{\overline{\mathbf{s}(x) \geq x \quad \varphi_{10}}}\ ax^3_I u^4_P$$

$$\frac{\Phi}{\mathbf{p}(p,\mathbf{A},\mathbf{B},\clubsuit)\Rightarrow \$(p)\geq \mathbf{2}}\;{}^{\scriptscriptstyle IND}\!\mathit{list} \quad \text{iff} \quad \frac{\Phi}{\mathbf{p}([l],\mathbf{A},\mathbf{B},\clubsuit)\Rightarrow \$([l])\geq \mathbf{2}\quad @\mathbf{8}_a}\;\scriptstyle\varSigma \quad \text{and}$$

$$\frac{\Phi\cup\{\mathbf{p}(p,\mathbf{A},\mathbf{B},\clubsuit)\Rightarrow \$(p)\geq \mathbf{2}\}}{\mathbf{p}(l\diamond p,\mathbf{A},\mathbf{B},\clubsuit)\Rightarrow \$(l\diamond p)\geq \mathbf{2}\quad @\mathbf{8}_b}\;\scriptstyle\varSigma\cup\{p\}$$

Proof of $@\mathbf{8}_a$: $\mathbf{p}([l],\mathbf{A},\mathbf{B},\clubsuit)\Rightarrow \$([l])\geq \mathbf{2}$ (l is $\mathbf{l}(c,c')$ by induction) .

$$\frac{\dfrac{}{\mathbf{p}([\mathbf{l}(c,c')],\mathbf{A},\mathbf{B},\clubsuit)\Rightarrow c=\mathbf{A}}\;\varphi_1}{\mathbf{p}([\mathbf{l}(c,c')],\mathbf{A},\mathbf{B},\clubsuit)\Rightarrow \mathbf{l}(c,c')\in\clubsuit=\mathbf{l}(\mathbf{A},c')\in\clubsuit\quad @\mathbf{81}}\;\scriptstyle\mathit{REM}$$

$$\frac{\dfrac{@\mathbf{81}\quad \dfrac{\dfrac{}{\mathbf{p}([\mathbf{l}(c,c')],\mathbf{A},\mathbf{B},\clubsuit)\Rightarrow c'=\mathbf{B}}\;\varphi_2}{\mathbf{p}([\mathbf{l}(c,c')],\mathbf{A},\mathbf{B},\clubsuit)\Rightarrow \mathbf{l}(\mathbf{A},c')\in\clubsuit=\mathbf{l}(\mathbf{A},\mathbf{B})\in\clubsuit}\;\scriptstyle\mathit{REM}}{\mathbf{p}([\mathbf{l}(c,c')],\mathbf{A},\mathbf{B},\clubsuit)\Rightarrow \mathbf{l}(c,c')\in\clubsuit=\mathbf{l}(\mathbf{A},\mathbf{B})\in\clubsuit}\;\scriptstyle\mathit{TRA}}{\mathbf{p}([\mathbf{l}(c,c')],\mathbf{A},\mathbf{B},\clubsuit)\Rightarrow \mathbf{l}(\mathbf{A},\mathbf{B})\in\clubsuit=\mathbf{l}(c,c')\in\clubsuit\quad @\mathbf{82}}\;\scriptstyle\mathit{SYM}$$

$$\frac{@\mathbf{82}\quad \dfrac{}{\mathbf{p}([\mathbf{l}(c,c')],\mathbf{A},\mathbf{B},\clubsuit)\Rightarrow \mathbf{l}(c,c')\in\clubsuit}\;\varphi_3}{\mathbf{p}([\mathbf{l}(c,c')],\mathbf{A},\mathbf{B},\clubsuit)\Rightarrow \mathbf{l}(\mathbf{A},\mathbf{B})\in\clubsuit\quad @\mathbf{83}}\;\scriptstyle\mathit{TRA}$$

$$\frac{\dfrac{\dfrac{\dfrac{}{\mathbf{l}(\mathbf{A},\mathbf{B})\in\clubsuit=\mathbf{F}}\;\varphi_4}{\mathbf{F}=\mathbf{l}(\mathbf{A},\mathbf{B})\in\clubsuit}\;\scriptstyle\mathit{SYM}\quad \dfrac{}{\mathbf{l}(\mathbf{A},\mathbf{B})\in\clubsuit=\mathbf{T}\Rightarrow\mathbf{l}(\mathbf{A},\mathbf{B})\in\clubsuit=\mathbf{T}}\;\scriptstyle\mathit{TAU}}{\mathbf{l}(\mathbf{A},\mathbf{B})\in\clubsuit=\mathbf{T}\Rightarrow\mathbf{F}=\mathbf{T}}\;\scriptstyle\mathit{TRA}\quad @\mathbf{83}}{\mathbf{p}([\mathbf{l}(c,c')],\mathbf{A},\mathbf{B},\clubsuit)\Rightarrow \mathbf{F}=\mathbf{T}\quad @\mathbf{84}}\;\scriptstyle\mathit{MP}$$

$$\frac{\dfrac{\dfrac{\dfrac{}{\$([l])=\mathbf{1}}\;\varphi_7}{\$([\mathbf{l}(c,c')])=\mathbf{1}}\;\scriptstyle\mathit{SUB}}{\$([\mathbf{l}(c,c')])\geq\mathbf{2}=\mathbf{1}\geq\mathbf{2}}\;\scriptstyle\mathit{REM}\quad \dfrac{\dfrac{}{\mathbf{1}\geq\mathbf{2}=\mathbf{F}}\;\varphi_5\quad @\mathbf{84}}{\mathbf{p}([\mathbf{l}(c,c')],\mathbf{A},\mathbf{B},\clubsuit)\Rightarrow \mathbf{1}\geq\mathbf{2}=\mathbf{T}}\;\scriptstyle\mathit{TRA}}{\mathbf{p}([\mathbf{l}(c,c')],\mathbf{A},\mathbf{B},\clubsuit)\Rightarrow \$([\mathbf{l}(c,c')])\geq\mathbf{2}=\mathbf{T}\quad @\mathbf{8}_a}\;\scriptstyle\mathit{TRA}$$

Proof of $@\mathbf{8}_b$: $\mathbf{p}(l::p,\mathbf{A},\mathbf{B},\clubsuit)\Rightarrow \$(l::p)\geq\mathbf{2}$, from $@\mathbf{86}$: $\mathbf{s}(\$(p))\geq\mathbf{2}=\mathbf{T}$:

$$\frac{\dfrac{\dfrac{\dfrac{}{\$(l::p)=\mathbf{s}(\$(p))}\;\varphi_6}{\$(l::p)\geq\mathbf{2}=\mathbf{s}(\$(p))\geq\mathbf{2}}\;\scriptstyle\mathit{REM}\quad @\mathbf{86}}{\$(l::p)\geq\mathbf{2}=\mathbf{T}}\;\scriptstyle\mathit{TRA}}{\mathbf{p}(l::p,\mathbf{A},\mathbf{B},\clubsuit)=\mathbf{T}\Rightarrow \$(l::p)\geq\mathbf{2}=\mathbf{T}\quad @\mathbf{8}_b}\;\scriptstyle\mathit{MON}$$

$$\frac{\Phi}{\mathbf{s}(\$(p))\geq\mathbf{2}\quad @\mathbf{86}}\;{}^{\scriptscriptstyle IND}\!\mathit{list}\quad\text{iff}\quad \frac{\Phi}{\mathbf{s}(\$([l]))\geq\mathbf{2}\quad @\mathbf{86}_a}\;\scriptstyle\varSigma\quad \&\quad \frac{\Phi\cup\{\mathbf{s}(\$(p))\geq\mathbf{2}\}}{\mathbf{s}(\$(l::p))\geq\mathbf{2}\quad @\mathbf{86}_b}\;\scriptstyle\varSigma\cup\{p\}$$

$$\frac{\dfrac{\dfrac{}{\$([l])=\mathbf{s}(\mathbf{0})}\;\varphi_7}{\mathbf{s}(\$([l]))\geq\mathbf{2}=\mathbf{s}(\mathbf{s}(\mathbf{0}))\geq\mathbf{2}}\;\scriptstyle\mathit{REM}\quad \dfrac{}{\mathbf{s}(\mathbf{s}(\mathbf{0}))\geq\mathbf{2}}\;\varphi_8}{\mathbf{s}(\$([l]))\geq\mathbf{2}\quad @\mathbf{86}_a}\;\scriptstyle\mathit{TRA}$$

$@\mathbf{86}_b$: $\mathbf{s}(\$(l::p))\geq\mathbf{2}$ with HR: $\mathbf{s}(\$(p))\geq\mathbf{2}$

$$\frac{\dfrac{\dfrac{}{x\geq y\wedge y\geq z\Rightarrow x\geq z}\;\varphi_9}{\mathbf{s}(\mathbf{s}(\$(p)))\geq\mathbf{s}(\$(p))\wedge\mathbf{s}(\$(p))\geq\mathbf{2}\Rightarrow\mathbf{s}(\mathbf{s}(\$(p)))\geq\mathbf{2}}\;\scriptstyle\mathit{SUB}\quad \dfrac{\dfrac{}{\mathbf{s}(x)\geq x}\;\varphi_{10}}{\mathbf{s}(\mathbf{s}(\$(p)))\geq\mathbf{s}(\$(p))}\;\scriptstyle\mathit{SUB}}{\mathbf{s}(\$(p))\geq\mathbf{2}\Rightarrow\mathbf{s}(\mathbf{s}(\$(p)))\geq\mathbf{2}\quad @\mathbf{85}}\;\scriptstyle\mathit{MPG}$$

$$\frac{\dfrac{\dfrac{}{\$(l::p)=\mathbf{s}(\$(p))}\;\varphi_6}{\mathbf{s}(\$(l::p))\geq\mathbf{2}=\mathbf{s}(\mathbf{s}(\$(p)))\geq\mathbf{2}}\;\scriptstyle\mathit{REM}\quad \dfrac{@\mathbf{85}\quad \dfrac{}{\mathbf{s}(\$(p))\geq\mathbf{2}}\;\scriptstyle\mathit{HR}}{\mathbf{s}(\mathbf{s}(\$(p)))\geq\mathbf{2}}\;\scriptstyle\mathit{MPG}}{\mathbf{s}(\$(l::p))\geq\mathbf{2}\quad @\mathbf{86}_b}\;\scriptstyle\mathit{TRA}$$

References

[AC92] E. Astesiano and M. Cerioli. Relationships between logical frameworks. In LNCS, editor, *Recent Trends in Data Type Specification*, volume 655, pages 101–126, Dourdan, 1992.

[BCLG96] G. Bernot, S. Coudert, and P. Le Gall. Towards heterogeneous formal specifications. In *AMAST'96, Munich*, volume 1101, pages 458–472. Springer, LNCS, 1996.

[BHW95] M. Bidoit, R. Hennicker, and M. Wirsing. Behavioural and abstractor specifications. *Science of Computer Programming*, 25:149–186, 1995.

[Bid87] M. Bidoit. The stratified loose approach : a generalization of initial and loose semantics. In Springer-Verlag LNCS 332, editor, *Recent Trends in Data Type Specification, Gullane, Scotland*, pages 1–22, July 1987.

[Cou] S. Coudert. Cadre de spécifications hétérogènes. Université d'Evry, forthcoming Thesis.

[EM85] H. Ehrig and B. Mahr. *Fundamentals of Algebraic Specification 1. Equations and initial semantics*, volume 6. Springer-Verlag,EATCS Monographs on Theoretical Computer Science, 1985.

[GB84] J.A. Goguen and R.M. Burstall. Introducing institutions. In Springer-Verlag LNCS 164, editor, *Proc. of the Workshop on Logics of Programming*, pages 221–256, 1984.

[GTW78] J.A. Goguen, J.W. Thatcher, and E.G. Wagner. An initial algebra approach to the specification, correctness, and implementation of abstract data types. In R.T. Yeh Printice-Hall, editor, *Current Trends in Programming Methodology*, volume IV, pages 80–149, 1978. Also IBM Report RC 6487, October 1976.

[Hen91] R. Hennicker. Context induction: a proof principle for behavioural abstractions and algebraic implementations. *Formal Aspects of Computing*, 3(4):326–345, 1991.

[HST94] R. Harper, D. Sannella, and A. Tarlecki. Structured theory presentations and logic representations. *Annals of Pure and Applied Logic*, 67:113–160, 1994.

[HW] R. Hennicker and M. Wirsing. Proof systems for structured algebraic specifications: An overview. *To appear in: Proc. FCT '97.*

[KC95] editors: K.E. Cheng, T. Ohta. Proc. of the third conference on feature interactions in telecommunication systems iii. Amsterdam, 1995. IOS Press., 1995.

[Mes89] J. Meseguer. General logics. In North-Holland, editor, *Proc. Logic. Colloquium '87*, Amsterdam, 1989.

[SS92] A. Salibra and G. Scollo. A soft stairway to institutions. In LNCS, editor, *Recent Trends in Data Type Specification*, volume 655, pages 320–329, Dourdan, 1992.

[Wir93] M. Wirsing. Structured specifications: syntax, semantics and proof calculus. In Brauer W. Bauer F. and Schwichtenberg H., editors, *Logic and Algebra of Specification*, pages 411–442. Springer, 1993.

[WM95] M. Walicki and Meldal. A complete calculus for the multialgebraic and functionnal sémantics of nondeterminism. ACM Transactions on Programming Langages and Systems, 17: 2, p. 366-393, 1995-03, 1995.