Heterogeneous bridges revisited

Till Mossakowski

BISH, Dept. of Computer Science, University of Bremen

Abstract. Heterogeneous specifications consist of parts written in different logics. Heterogeneous bridges have been introduced [4,8] as a means to obtain truly heterogeneous proofs for heterogeneous specifications — however, the notions have been very ad-hoc. We here take the recently developed foundations for heterogeneous specifications and present a calculus for heterogeneous proofs that allows to use heterogeneous bridges as well. As an example, we demonstrate how a proof goal in a toy heterogeneous specification can be decomposed into two subgoals, one in first-order logic and the other one in propositional modal logic. We also show that the heterogeneous bridges of [4] can be simulated in our approach.

1 Introduction

Heterogeneous multi-logic specifications are needed whenever complex problems have different aspects that are best specified in different logics. A combination of all the used logics would become too complex in many cases.

Presently, there are two lines of research about heterogeneous specification. One line emphasizes heterogeneous proofs: heterogeneous specifications and heterogeneous bridges (i.e. bridges between proof systems for different logics) have been introduced [4,8]. However, these bridges are introduced in an ad-hoc manner.

The other line of research has achieved a much cleaner foundation of heterogeneous specification, via model-theoretic heterogeneous structuring constructs [21] and Grothendieck institutions [9]. However, proof theoretic investigations here have mainly been limited to global encodings of all the involved logics into some “universal” logic ([21,15]).

The aim of the present work is to combine the solid semantic foundation with heterogeneous bridges, such that it becomes possible to exploit the power of specialized proof tools. In [13], we have already started to go in this direction, and have introduced a calculus for heterogeneous proofs. However, it turns out that this calculus —while allowing heterogeneous proofs in principle— still has the tendency to shift proof goals toward some “rich” or “universal” logic. We therefore extend this calculus by new rules that support heterogeneous bridges similar in spirit to those of [4,8].

As an application, we give a sample heterogeneous proof involving modal logic and first-order logic. The main proof goal is split into two subgoals, one of which is entirely within propositional modal logic, while the other one is within
first-order logic. Both subgoals a linked by a “bridging lemma”. The subgoals can then be discharged using tools for the individual logics.

2 The formal basis for heterogeneous specification

When studying heterogeneous specification, we want to focus on the structuring and abstract from the details of the underlying logical systems. Therefore, we recall the abstract notion of institution [11], covering model theory. Proof theory is captured by the notion of entailment system [12].

Definition 2.1. An institution $I = (\text{Sign}, \text{Sen}, \text{Mod}, \models)$ consists of

- a category $\text{Sign}$ of signatures,
- a functor $\text{Sen} : \text{Sign} \to \text{Set}$ giving the set of sentences $\text{Sen}(\Sigma)$ over each signature $\Sigma$, and for each signature morphism $\sigma : \Sigma \to \Sigma'$, the sentence translation map $\text{Sen}(\sigma) : \text{Sen}(\Sigma) \to \text{Sen}(\Sigma')$, where often $\text{Sen}(\sigma)(\varphi)$ is written as $\sigma(\varphi)$,
- a functor $\text{Mod} : \text{Sign}^{op} \to \text{CAT}$ giving the category of models over a given signature, and for each signature morphism $\sigma : \Sigma \to \Sigma'$, the reduct functor $\text{Mod}(\sigma) : \text{Mod}(\Sigma') \to \text{Mod}(\Sigma)$, where often $\text{Mod}(\sigma)(M')$ is written as $M'|_{\sigma}$,
- a satisfaction relation $\models_{\Sigma} \subseteq \text{Mod}(\Sigma) \times \text{Sen}(\Sigma)$ for each $\Sigma \in \text{Sign}$, such that for each $\sigma : \Sigma \to \Sigma'$ in $\text{Sign},$

$$M' \models_{\Sigma'} \sigma(\varphi) \iff M'|_{\sigma} \models_{\Sigma} \varphi$$

holds for each $M' \in \text{Mod}(\Sigma')$ and each $\varphi \in \text{Sen}(\Sigma)$ (satisfaction condition).

Within an arbitrary but fixed institution, we can easily define the usual notion of logical consequence or semantical entailment: Given a set of $\Sigma$-sentences $\Gamma$ and a $\Sigma$-sentence $\varphi$, we say

$$\Gamma \models_{\Sigma} \varphi \ (\varphi \text{ follows from } \Gamma)$$

iff for all $\Sigma$-models $M$, we have

$$M \models_{\Sigma} \Gamma \text{ implies } M \models_{\Sigma} \varphi.$$ 

Here, $M \models_{\Sigma} \Gamma$ means that $M \models_{\Sigma} \psi$ for each $\psi \in \Gamma$.

We will also freely use other standard logical terminology when working within an arbitrary but fixed institution.

A logic is an institution equipped with an entailment system consisting of an entailment relation $\vdash_{\Sigma} \subseteq \text{Sen}(\Sigma) \times \text{Sen}(\Sigma)$, for each $\Sigma \in \text{Sign}$, such that the following conditions are satisfied:

1. reflexivity: for any $\varphi \in \text{Sen}(\Sigma)$, $\vdash_{\Sigma} \varphi$,
2. monotonicity: if $\Gamma \vdash_{\Sigma} \varphi$ and $\Gamma' \supseteq \Gamma$ then $\Gamma' \vdash_{\Sigma} \varphi$,
3. transitivity: if $\Gamma \vdash_{\Sigma} \varphi_i$, for $i \in I$, and $\Gamma \cup \{\varphi_i \mid i \in I\} \vdash_{\Sigma} \psi$, then $\Gamma \vdash_{\Sigma} \psi$, 

2
4. \( \vdash \)-translation: if \( \Gamma \vdash \Sigma \varphi \), then for any \( \alpha: \Sigma \rightarrow \Sigma' \) in \( \text{Sign} \), \( \sigma[\Gamma] \vdash_{\Sigma'} \sigma(\varphi) \),
5. soundness: for any \( \Sigma \in \{\text{Sign}\} \), \( \Gamma \subseteq \text{Sen}(\text{Sign}) \) and \( \varphi \in \text{Sen}(\Sigma) \),
\[
\Gamma \vdash \Sigma \varphi \Rightarrow \Gamma \vdash_{\Sigma} \varphi.
\]
A logic will be called \emph{complete} if, in addition, the converse of the soundness implication holds.

We now come to relations between institutions.

\textbf{Definition 2.2.} Given institutions \( I \) and \( J \), an \emph{institution comorphism} [10, 20, 12] \( \mu = (\Phi, \alpha, \beta): I \rightarrow J \) consists of
- a functor \( \Phi: \text{Sign} \rightarrow \text{Sign} \)',
- a natural transformation \( \alpha: \text{Sen}' \rightarrow \text{Sen} \circ \Phi \),
- a natural transformation \( \beta: \text{Mod}' \circ \Phi \rightarrow \text{Mod} \)

such that the following satisfaction condition is satisfied for all \( \Sigma \in \text{Sign}' \), \( M' \in \text{Mod}'(\Phi(\Sigma)) \) and \( \varphi \in \text{Sen}'(\Sigma) \):
\[
M' \vdash_{\Phi(\Sigma)}^{J} \alpha_{\Sigma}(\varphi) \Leftrightarrow \beta_{\Sigma}(M') \vdash_{\Sigma}^{J} \varphi.
\]
Together with obvious identities and composition, this gives a category \emph{coIns} of institutions and comorphisms.

An institution comorphism is called \emph{model-expansive}, if the model translations \( \beta_{\Sigma} \) are surjective on objects. It is called \emph{conservative}, if for each \( I \)-signature \( \Sigma \), each set of \( \Sigma \)-sentences \( \Gamma \) and each \( \Sigma \)-sentence \( \varphi \), we have
\[
\Gamma \vdash_{J}^{I} \varphi \text{ if } \alpha_{\Sigma}(\Gamma) \vdash_{\Phi(\Sigma)}^{J} \alpha_{\Sigma}(\varphi).
\]
The well-known “borrowing theorem” [7] states each model-expansive comorphism is conservative. The importance of conservativeness lies in the fact that if \( J \) comes with a sound and complete logic, \( I \) can be turned into one by simply putting
\[
\Gamma \vdash_{J}^{I} \varphi \text{ if } \alpha_{\Sigma}(\Gamma) \vdash_{\Phi(\Sigma)}^{J} \alpha_{\Sigma}(\varphi).
\]
This is called “borrowing” of logical structure.

The following definition makes precise the central data that have to be provided as a basis for heterogeneous specification.

\textbf{Definition 2.3.} An \emph{indexed coconstitution} is a functor \( \mathcal{I}: \text{Ind} \rightarrow \text{coIns} \) into the category of institutions and institution comorphisms. Typically some of the institutions come as logics, which will play a rôle only later.

Any graph of institutions and comorphisms can be extended to an indexed coconstitution by taking \( \text{Ind} \) to be the free category over the graph, basically consisting of paths.

An indexed coconstitution can be flattened, using the so-called \emph{Grothendieck construction} [9]. The basic idea here is that all signatures of all institutions are put side by side, and a signature morphism in this large realm of signatures consists of an intra-institution signature morphism plus an inter-institution translation (along some institution comorphism from the indexed coinstitution). The other components are then defined in a straightforward way.
Definition 2.4. Given an indexed coinstitution $I: \text{Ind} \to \text{coIns}$, define the Grothendieck institution $I^\# = (\text{Sign}^\#, \text{Sen}^\#, \text{Mod}^\#, \#)$ as follows:

- signatures in $I^\#$ are pairs $(\Sigma, i)$, where $i \in [\text{Ind}]$ and $\Sigma$ a signature in the institution $I(i)$,
- signature morphisms $(\sigma, d): (\Sigma_1, i) \to (\Sigma_2, j)$ consist of a morphism $d : i \to j \in \text{Ind}$ and a signature morphism $\Phi^I(d) : \Sigma_1 \to \Sigma_2$ (here, $I(d) : I(i) \to I(j)$ is the institution comorphism corresponding to the arrow $d : i \to j$ in the institution graph, and $\Phi^I(d)$ is its signature translation component),
- the $(\Sigma, i)$-sentences are the $\Sigma$-sentences in $I(i)$, and sentence translation along $(\sigma, d)$ is the composition of sentence translation along $I(d)$ with sentence translation along $\sigma$,
- the $(\Sigma, i)$-models are the $\Sigma$-models in $I(i)$, and model reduction along $(\sigma, d)$ is the composition of model translation along $I(d)$ with model reduction along $\sigma$,
- satisfaction w.r.t. $(\Sigma, i)$ is satisfaction w.r.t. $\Sigma$ in $I(i)$.

3 Modal logic and first-order logic

Consider the following graph of institutions resp. logics and comorphisms:

![Graph of Institutions and Logics]

This graph consists of the following institutions and logics:

**Prop** is propositional logic. Signatures are sets of propositional constants, and signature morphisms are functions between these. A model is a valuation of the propositional constants with truth values. Sentences are built from the propositional constants using the logical connectives $\lor$, $\land$ and $\neg$. Model reduction and sentence translation is straightforward. Satisfaction is defined inductively as usual.

This institution can also be equipped with some propositional entailment system; we thus obtain a complete logic.
FOL is many-sorted first-order logic. Signatures consist of a set of sorts, a set of function symbols and a set of predicate symbols (each symbol coming with a string of argument sorts and, for function symbols, a result sort). Signature morphisms map the three components in a compatible way. Models are first order structures, and sentences are the usual first-order sentences built from equations, predicate applications and logical connectives and quantifiers \( \forall \), \( \exists \). Satisfaction is defined inductively in the usual way. A detailed description of this institution can be found in [11].

Again, this institution can be equipped with a first-order entailment system; leading to a complete logic.

PropModal is propositional modal logic. Signatures are those from Prop. A model is a Kripke structure, i.e. a set of worlds with a binary reachability relation, and, for each propositional constant, a subset of the set of worlds indicating the worlds in which this constant holds. Sentences are built from the propositional constants using the logical connectives as well as the modal prefix operators \( \Box \) and \( \Diamond \). Satisfaction is defined via Kripke semantics: a sentence holds in a model if it holds in all worlds of that model, and the holding of a sentence in a particular world is defined inductively, such that \( \Box \) means “for all reachable worlds” and \( \Diamond \) means “for some reachable world”.

Again, this institution can be equipped with a complete entailment system (see, for example, [11]).

IndexedPropModal (indexed propositional modal logic) is a weak fragment of first-order modal logic. Signatures consist of a set of sorts and a set of predicate symbols. Signature morphisms are similar to those in FOL. A model is a first-order Kripke structure with constant domains. This means that each sort is interpreted as a carrier set that is fixed in all worlds. Each predicate symbol is interpreted as a family of relations on the appropriate carriers, indexed by the set of worlds. Sentences are built from applications of the predicate symbols to variables using the logical connectives as well as the modal operators \( \Box \) and \( \Diamond \). Within one sentence, we require that for each argument position, the variable appearing in that position is the same for all predicate applications. A sentence holds in a model if it holds in all worlds and for all valuations of the variables.

We do not equip this institution with an entailment system; rather, we have chosen the fragment of first-order modal logic in such a way that for the purpose of performing proofs, we still can reduce it to propositional modal logic. Indeed, IndexedPropModal just formalizes the common practice to work with propositions in PropModal that are indexed by some (possibly infinite) set.

The institution comorphisms shown in the above graph are all trivial inclusions, except the comorphisms from IndexedPropModal into FOL and PropModal. We now describe the latter ones.

The comorphism from IndexedPropModal into FOL maps a IndexedPropModal-signature to a FOL-signature by adding a sort \( W \) (for the set of worlds) to the sort set, and adding \( W \) to the argument sort string of each predicate symbol.

The institution comorphisms shown in the above graph are all trivial inclusions, except the comorphisms from IndexedPropModal into FOL and PropModal. We now describe the latter ones.

The comorphism from IndexedPropModal into FOL maps a IndexedPropModal-signature to a FOL-signature by adding a sort \( W \) (for the set of worlds) to the sort set, and adding \( W \) to the argument sort string of each predicate symbol.
Also, a binary relation \( R : \mathcal{W} \times \mathcal{W} \) is added (this will be the accessibility relation). An IndexedPropModal-sentence is translated to FOL as follows:

\[
\begin{align*}
\alpha(p(x_1, \ldots, x_n)) & \equiv p(x_1, \ldots, x_n, w) \\
\alpha(\varphi \land \psi) & \equiv \alpha(\varphi) \land \alpha(\psi) \\
\alpha(\varphi \lor \psi) & \equiv \alpha(\varphi) \lor \alpha(\psi) \\
\alpha(\neg \varphi) & \equiv \neg \alpha(\varphi) \\
\alpha(\Box \varphi) & \equiv \forall w' : W. R(w, w') \Rightarrow \alpha(\varphi)[w/w'] \\
\alpha(\Diamond \varphi) & \equiv \exists w' : W. R(w, w') \land \alpha(\varphi)[w/w']
\end{align*}
\]

Finally, a FOL-model is mapped to an IndexedPropModal-model by taking the carrier set for \( W \) as the set of worlds, the interpretation of \( R \) as the accessibility relation, and defining a predicate \( p \) to hold on arguments \( a_1, \ldots, a_n \) in world \( w \) if \( p \) holds on arguments \( a_1, \ldots, a_n, w \) in the FOL-model. The satisfaction condition for this comorphism is shown by a straightforward induction over the sentences. Moreover, it is easy to see that the model translation can be reversed and hence that this comorphism is model-expansive.

The comorphism from IndexedPropModal to PropModal maps an IndexedPropModal-signature to a PropModal-signature by just forgetting the sorts and the arguments of the predicate symbols, ending up with a set of propositional constants. Similarly, sentences are mapped by forgetting the argument variables for the predicate symbols. Finally, a PropModal-model can be extended to an IndexedPropModal-model by interpreting all carrier sets a singletons; the predicates then actually degenerate to propositional constants, and these are obtained from the PropModal-model. Again, the satisfaction condition is straightforward.

Note that the latter comorphism is not model-expansive: only models with singleton carriers are in the image of the model translation, which is clearly not surjective. However, we still have:

**Proposition 3.1.** The comorphism from IndexedPropModal to PropModal is conservative.

Proof. In order to show this, we need only to show that

\[
\alpha^\Sigma(\Gamma) \models_{\text{PropModal}} \alpha^\Sigma(\varphi) \text{ implies } \Gamma \models_{\text{IndexedPropModal}} \varphi,
\]

since the converse implication already follows from the satisfaction condition. Therefore, assume that \( \alpha^\Sigma(\Gamma) \models_{\text{PropModal}} \alpha^\Sigma(\varphi) \), and further assume that a \( \Sigma \)-model \( m \) satisfies \( \Gamma \). Recall that for each sentence in IndexedPropModal, we require that for each argument position, the variable appearing in that position is the same for all predicate applications. By renaming the variables in \( \Gamma \) and \( \varphi \), we can assume w.l.o.g. that this holds uniformly for all sentences in \( \Gamma \cup \{ \varphi \} \).

Now \( m \models_{\Sigma} \Gamma \) means that for each variable valuation \( \nu \) assigning values to all variables in \( \Gamma \cup \{ \varphi \} \), we have \( m, \nu \models_{\text{IndexedPropModal}} \varphi \). Fix such a valuation \( \nu \). From \( \nu \), we can obtain a propositional \( \Phi(\Sigma) \)-model \( \hat{\nu} \) by just interpreting the propositional constant \( p \) to be the predicate \( p \) in \( m \) applied to the values
determined by $\nu$ (note that each argument position of a predicate corresponds to a fixed variable). We then easily obtain $\nu \models_{\phi[\Sigma]}^{\text{PropMod}} \alpha_\Sigma(\Gamma)$, and, by assumption, $\nu \models_{\phi[\Sigma]}^{\text{PropMod}} \alpha_\Sigma(\varphi)$, and finally $\Gamma \models_{\Sigma}^{\text{IndexedPropMod}} \varphi$. □

Consider the sample heterogeneous specification in Fig. 1. It starts with a theory in IndexedPropMod, specifying something about persons that may be married, dead and immortal. (Intuitively, the modalities are interpreted as temporal ones here.) This theory is then translated to FOL along the comorphism making worlds explicit defined above. Then, the resulting theory is enriched by some first-order formula involving existential quantification. Call the thus obtained theory $T$. Finally, using the then $%\text{implies}$ annotation, a proof obligation $\varphi$ is expressed. The proof obligation is again written in IndexedPropMod, but it is implicitly coerced to FOL using the same comorphism as above in order to be compatible with $T$. The proof obligation expresses that $T$ implies $\varphi$.

spec Marriage =
{ logic IndexedPropMod
sort Person
props isMarried, immortal, dead: Person;
isMarriedTo: Person × Person
var $x, y$: Person
$\bullet$ isMarriedTo$(x, y)$ \implies $\neg$dead$(x)$
$\bullet$ immortal$(x)$ \iff $\neg \Diamond$dead$(x)$ \} with logic $\rightarrow$ FOL
then var $x$: Person; $w$: World
$\bullet$ isMarried$(x, w)$ \implies $\exists y$: Person $\bullet$ isMarriedTo$(x, y, w)$
then $%\text{implies}$
logic IndexedPropMod \} implicitly
var $x$: Person
$\bullet$ $(\Diamond$isMarried$(x)) \Rightarrow$ immortal$(x)$ \} coerced
end

Fig. 1. A sample heterogeneous specification

4 Heterogeneous development graphs

We now come to heterogeneous specification over a graph of institutions and logics such as the one introduced in the previous section. We here use development graphs, as introduced in [3, 16]. They are used to encode the structured specifications in various phases of software development. Roughly speaking, each node of a development graph represents a theory. The definition links of the graph define how theories can make use of other theories. Arising proof obligations are attached as so-called theorem links to the graph.
Development graphs are defined over an arbitrary but fixed institution or logic. In [15, 13], we have studied heterogeneous development graphs as well (these are just development graphs over a Grothendieck logic).

For sake of simplicity, we here introduce development graphs without hiding.

**Definition 4.1.** A heterogeneous development graph (over some given indexed constitutions $\mathcal{I}$) is an acyclic directed graph $\mathcal{S} = (\mathcal{N}, \mathcal{L})$.

$\mathcal{N}$ is a set of nodes. Each node $N \in \mathcal{N}$ is a tuple $(\Sigma^N, \Gamma^N)$ such that $\Sigma^N \in \text{Sign}^\#$ is a (Grothendieck) signature and $\Gamma^N \subseteq \text{Sen}^\#(\Sigma^N)$ is the set of local axioms of $N$.

$\mathcal{L}$ is a set of directed links, so-called *global definition links*, between elements of $\mathcal{N}$. Each global definition link from a node $M$ to a node $N$ is annotated with a signature morphism $\sigma : \Sigma^M \to \Sigma^N \in \text{Sign}^\#$. The link is then denoted as $M \xrightarrow{\sigma} N$.

**Definition 4.2.** Given a node $N \in \mathcal{N}$, its associated class $\text{Mod}_S(N)$ of models (or $N$-models for short) consists of those $\Sigma^N$-models $n$ for which

1. $n$ satisfies the local axioms $\Gamma^N$, and
2. for each $K \xrightarrow{\sigma} N \in \mathcal{S}$, $n|_{\sigma}$ is a K-model.

The notion of *global reachability* is defined inductively: A node $M$ is globally reachable from a node $N$ via a signature morphism $\sigma$, $N \xrightarrow{\sigma} M$ for short, iff either $N = M$ and $\sigma = \text{id}$, or $N \xrightarrow{\sigma'} K \in \mathcal{S}$, and $K \xrightarrow{\sigma''} M$, with $\sigma = \sigma'' \circ \sigma'$.

Complementary to definition links, which define the theories of related nodes, we introduce the notion of a *theorem link* with the help of which we are able to *postulate* relations between different theories. Global and local theorem links (denoted by $N \xrightarrow{\sigma} M$ and $N \xrightarrow{\sigma} M$, resp., where $\sigma : \Sigma^N \to \Sigma^M$) are the central data structure to represent proof obligations arising in formal developments.

**Definition 4.3.** Let $\mathcal{S}$ be a development graph. $\mathcal{S}$ *implies* a global theorem link $N \xrightarrow{\sigma} M$ (denoted $\mathcal{S} \models N \xrightarrow{\sigma} M$), iff for all $m \in \text{Mod}_S(M)$, $\sigma|_m \in \text{Mod}_S(N)$. $\mathcal{S}$ implies a local theorem link $N \xrightarrow{\sigma} M$, if for all $m \in \text{Mod}_S(M)$, $\sigma|_m \models \Gamma^N$.

As an example, the heterogeneous development graph for the specification MARRIAGE is shown in Fig. 2.

The *theory* of a node $M$ in a development graph $\mathcal{S}$, written $\text{Th}_S(M)$, is defined to be

$$ \bigcup_{K \xrightarrow{\sigma} M \in \mathcal{S}} \sigma(\Gamma^K) $$

A global definition link $M \xrightarrow{\sigma} N$ in a development graph is *model expansive* if every $M$-model can be expanded along $\sigma$ to an $N$-model. We will
allow annotating a global definition link as $M \xrightarrow{\sigma} N$, which expresses that it is model expansive. Similarly, $M \xrightarrow{\sigma|\Delta} N$ shall expresses a conservative extension, which means that $\mathsf{Th}_\Delta(N) \models \sigma(\varphi)$ implies $\mathsf{Th}_\Delta(M) \models \varphi$ for all $\Sigma^M$-sentences $\varphi$. These annotations can be seen as another kind of proof obligation. Note that we here, unlike in [16,13], distinguish between model expansion and conservativeness (and what we now call model expansion has been called conservativeness there). The reason is that for the first time we have encountered a morphism (namely that from from IndexedPropModal to PropModal) that is conservative, but not model expansive.

5 Proof rules for heterogeneous development graphs

Suppose that we now want to prove the abovementioned proof obligation. There are several ways to do this. The simplest way is just to use heterogeneous borrowing, as described in [15]. This means that everything is translated to FOL (this is possible, because in our institution graph, each institution is embedded in FOL). Then, one can use the entailment system (resp. a corresponding theorem prover) for FOL.

However, this is unsatisfactory, because in practice it is more efficient to use the entailment system (resp. a corresponding theorem prover) for PropModal for those parts of the proof that can be carried out within PropModal. Therefore, in [13], we have introduced proof rules for truly heterogeneous proofs, and proved a soundness and (relative) completeness theorem for these rules.

Unfortunately, upon inspection of the proof of the completeness theorem, it turns out that for our example, again everything is mapped to FOL (after all, this is possible, and the completeness theorem has only to guarantee that there
is a proof, and need not deliver the most efficient one). In order to obtain a
better fine-tuning of heterogeneous proofs, we here extend the rule system of
[13] with two additional rules: local borrowing and composition. These rules are
central for obtaining heterogeneous bridges.

\[\begin{align*}
\text{Glob-Decomposition:} & \quad K \xrightarrow{\sigma \circ \sigma'} M \quad \text{for each } K \xrightarrow{\sigma'} N \\
& \quad N \xrightarrow{\sigma'} M
\end{align*}\]

\[\begin{align*}
\text{Subsumption:} & \quad N \xrightarrow{\sigma'} M \\
& \quad N \xrightarrow{\sigma'} M
\end{align*}\]

\[\begin{align*}
\text{Global Borrowing:} & \quad M \xrightarrow{\sigma} N \\
& \quad M' \xrightarrow{\sigma'} N' \\
& \quad M \xrightarrow{\sigma} N \\
& \quad M' \xrightarrow{\sigma'} N' \\
& \quad \sigma' \circ \theta = \theta' \circ \sigma \\
& \quad N' \text{ is isolated}
\end{align*}\]

\[\begin{align*}
\text{Local Borrowing:} & \quad M \xrightarrow{\sigma} N \\
& \quad M' \xrightarrow{\sigma'} N' \\
& \quad M \xrightarrow{\sigma} N \\
& \quad M' \xrightarrow{\sigma'} N' \\
& \quad \theta : \Sigma^M \rightarrow \Sigma^{M'} \\
& \quad \sigma' \circ \theta = \theta' \circ \sigma \\
& \quad \theta(G^M) \subseteq G^{M'} \\
& \quad N' \text{ is isolated}
\end{align*}\]

\[\begin{align*}
\text{Composition:} & \quad K \xrightarrow{\sigma} L \xrightarrow{\theta} M \\
& \quad K \xrightarrow{\theta \circ \sigma} M
\end{align*}\]

\[\begin{align*}
\text{Basic Inference:} & \quad \text{Th}_S(M) \vdash_{\Sigma^M} \sigma(\varphi) \text{ for each } \varphi \in \Gamma^N \\
& \quad N \xrightarrow{\sigma} M
\end{align*}\]

A node is said to be isolated, if its set of local axioms is empty, and all the
ingoing global definition links are those shown in the proof rule. Isolated nodes
are typically not present in the original development graph, but added during proofs.

We further postulate that we can infer model-expansiveness and conserva-
tiveness of global definition links if the corresponding signature morphism just
consists of a comorphism component (i.e. the intra-logic component is the iden-
tity), and the comorphism enjoys model-expansiveness or conservativeness, resp.

Due to the omission of hiding, the following result is much simpler than that
in [13]:

\footnote{Note that we have omitted rules from [13] dealing with hiding and conservativeness.}
Theorem 5.1. The proof rules are sound. Under the assumption that a subset \( \mathcal{L} \) of the institutions of the indexed constitution \( \mathcal{I} \) come as complete logics, and that each institution in \( \mathcal{I} \) is mapped to and institution in \( \mathcal{L} \) by a conservative comorphism in \( \mathcal{I} \), the proof rules are also complete. \( \Box \)

6 A sample heterogeneous proof

We now use the proof calculus for heterogeneous development graphs to prove the proof obligation of the example. Recall the corresponding development graph:

\[
\begin{array}{ccc}
& \text{FOL}_1 & \\
\text{IPM}_1 & \xrightarrow{(id, id)} & \xleftarrow{(id, id)} & \text{FOL}_2 \\
& \text{IPM}_2 & \\
\end{array}
\]

The global theorem link in the heterogeneous development graph is discharged by successive backwards applications of the proof rules, thereby reducing the theorem link to simpler ones, until all of them can be removed.

The entire proof is shown in Fig. 3. The first step just decomposes the global theorem link into several local ones. One of the latter can trivially be discharged in step 2, a second one in step 3. (The application of basic inference in the third step is trivial because the node FOL\(_2\) does not contain local axioms.) At this point, we now could apply basic inference also for the remaining local theorem link. However, this would be basic inference entirely in FOL.

Instead, we try to decompose the proof goal into subgoals in FOL and in modal logic. This is done in step 4, using the rule Composition. This step is the key step of the whole proof, because here the heterogeneous bridge, similar in spirit to the bridges of [4, 8], is constructed. We therefore introduce a new node, IPM\(_3\). It has an ongoing global definition link from IPM\(_1\) and a local axiom \( \text{isMarried}(x) \Rightarrow \neg \text{dead}(x) \). This local axiom can be seen as a lemma bridging between IndexedPropModal and FOL: it is formulated in IndexedPropModal, proved in FOL, and used in an IndexedPropModal proof. The fourth step thus leaves us with two new theorem links.

The first of these is a global one, and hence we decompose it in the step 5. One of the resulting local theorem links can be discharged trivially in step 6. The other one is solved by basic inference in FOL in step 7. This is done as follows: we need to show that the local axiom of IPM\(_3\), namely \( \text{isMarried}(x) \Rightarrow \neg \text{dead}(x) \), follows from FOL\(_1\). Now the translation along \( \rho \) is \( \text{isMarried}(x, w) \Rightarrow \neg \text{dead}(x, w) \), and this follows from \( \text{isMarried}(x, w) \Rightarrow \exists y \bullet \text{isMarriedTo}(x, y, w) \) (a local axiom of FOL\(_1\)) and \( \text{isMarriedTo}(x, y, w) \Rightarrow \neg \text{dead}(x, w) \) (the translation of \( \text{isMarriedTo}(x, y) \Rightarrow \neg \text{dead}(x) \)) coming from IPM\(_1\) by inference in the FOL entailment system.
Fig. 3. A sample heterogeneous proof
It remains to solve the remaining local theorem link, which lives entirely in IndexedPropModal. We now can exploit the way how IndexedPropModal was built; namely in such a way that there is a conservative comorphism \( \mu \) into PropModal. We add two new nodes to the development graph, PM2 and PM3, whose signatures are the translations of those of IPM2 and IPM3, resp. PM3 gets a global definition link coming from IPM3 (via \( \mu \)), while PM2 only gets the translation of the local axiom in IPM2 along \( \mu \) as local axiom (the result of the translation is \( \Boxis\text{-}Married \Rightarrow \text{immortal} \), a purely modal propositional formula).

We now can apply Local Borrowing in order to translate the local theorem link from IndexedPropModal into PropModal in step 8. Finally, in step 9, this link is discharged by basic inference in PropModal: from \( \Boxis\text{-}Married \), we get \( \Box\neg\text{dead} \) by the bridge lemma (i.e. the local axiom of IPM3). By propositional modal reasoning, we get \( \neg\Diamond\text{dead} \). By the translation of one of the local IPM1 axioms (\( \text{immortal} \Leftrightarrow \neg\Diamond\text{dead} \)), we get \( \text{immortal} \). This completes the heterogeneous proof.

7 Related work

We now compare our approach to the original approach to heterogeneous bridges in [4]. One minor difference is that in [4], signatures and models are translated covariantly (as by institution morphisms), while we translate them contravariantly (using institution comorphisms). However, by the results of [2], we can often switch between these two views, so the difference is rather inessential here. We therefore recast the definition of [4] for the comorphism case, make the indexing more obvious, and use heterogeneous development graphs instead of the heterogeneous specifications of [4]. We also turn some of the technical conditions into (in our eyes) more natural ones, but the general idea remains that of [4].

Semi-comorphisms are just comorphisms without sentence translation components. Let \( \text{semi-cols} \) be the category of institutions and semi-comorphisms. Indexed constitutions and the Grothendieck construction then easily generalize to semi-comorphisms.

Definition 7.1. A heterogeneous framework \( (I, \vdash) \) consists of

- an indexed semi-coinstitution \( I: \text{Ind} \rightarrow \text{semi-cols} \),
- a family \( (\vdash_{d, \Sigma})_{d, \Sigma \in \text{Sign}^I} \) with \( \vdash_{d, \Sigma} \subseteq \mathcal{P}(\text{Sen}^I(\Sigma)) \times \text{Sen}^I(\Phi^d(\Sigma)) \), called heterogeneous inference bridge,

such that

- \( \vdash_{d, \Sigma} \) is monotonic,
- for any \( d: i \rightarrow j \in \text{Ind} \), any signature morphism \( \sigma: \Sigma \rightarrow \Sigma' \) in \( \text{Sign}^I \), any \( \Gamma \subseteq \text{Sen}^I(\Sigma) \), and any \( \varphi \in \text{Sen}^I(\Phi^d(\Sigma)) \),

\[
\Gamma \vdash_{d, \Sigma} \varphi \text{ implies } (\sigma(\Gamma)) \vdash_{d, \Sigma'} \Phi^d(\sigma)(\varphi) \quad (\vdash\text{-translation}),
\]

13
for any $d: i \rightarrow j \in Ind$, any $\Gamma \subseteq \text{Sen}^f(\Sigma)$, any $\varphi \in \text{Sen}^d(\Phi^d(\Sigma))$, and any $m \in \text{Mod}^d(\Phi(\Sigma))$ with $\Gamma \vdash_{d, \Sigma} \varphi$,

$$\text{if } \beta^d_\Sigma(m) \models_\Sigma \Gamma, \text{ then } m \models_{\Phi(\Sigma)} \varphi \quad \text{(heterogeneous soundness).}$$

**Definition 7.2.** Given a heterogeneous framework $(I, \vdash)$ and a heterogeneous development graph $\mathcal{S}$ over $I$, the corresponding heterogeneous inference system is the least binary relation $\vdash_{\mathcal{S}}$ between nodes $N$ in $\mathcal{S}$ and $\Sigma^N$-sentences such that

- $\vdash_{\mathcal{S}}$ is compatible with $\vdash$: if $\Sigma^N = (\Sigma, i)$ and $\Gamma^N \vdash_\Sigma \varphi$, then $N \vdash \varphi$;
- $\vdash_{\mathcal{S}}$ is compatible with $\vdash$: if $M \vdash_{\mathcal{S}} \Gamma$ and $\Gamma \vdash_{d, \Sigma} \varphi$, then $\vdash_{\mathcal{S}} \sigma(\varphi)$;
- $\vdash_{\mathcal{S}}$ is transitive: if $M \vdash_{\mathcal{S}} Th_{\mathcal{S}}(N)$ and $N \vdash_{\mathcal{S}} \varphi$, then $M \vdash_{\mathcal{S}} \varphi$.

Our main criticism of this approach is the ad-hoc nature of the bridge relation $\vdash$. We find it more natural to generate it via the entailment relation $\vdash$ of a logic in connection with the sentence translation of a comorphism:

**Proposition 7.3.** Any indexed coconstitution $I$ equipped with a set $\mathcal{L}$ of logics satisfying the assumptions for the completeness result in Theorem 5.1 leads to a heterogeneous framework by just putting

$$\Gamma \vdash_{d, \rightarrow j, \Sigma} \varphi \quad \text{iff} \quad \alpha^d(\alpha^d(\Gamma)) \vdash_{\Phi^d(\Phi^d(\Sigma))}^k \alpha^d(\varphi),$$

where $d: j \rightarrow k$ is such that $I(d')$ is a conservative comorphism into some logic in $\mathcal{L}$.

**Theorem 7.4.** When using the construction of Prop. 7.3, the heterogeneous inference system $\vdash_{\mathcal{S}}$ induced by the constructed heterogeneous framework can be simulated using our proof rules for heterogeneous development graphs.

We cannot expect to reverse the above construction and get an indexed coconstitution out of a heterogeneous framework, because there is no general way to obtain the sentence translation. So heterogeneous frameworks are more general than our approach. However, we believe that this extra generality is not of much use (any practical bridge will be obtained as in Prop. 7.3), and moreover comes with the cost of introducing the bridge relation $\vdash$ in an ad-hoc manner. Note that it is possible (and important) also to include semi-(co)morphisms in our framework [14], but this is mainly for relating specification and programming languages, and not for heterogeneous bridges.

8 Conclusion

We have demonstrated that heterogeneous bridges (and hence truly heterogeneous proofs), introduced in [4, 8] in an ad-hoc manner, can be obtained in the
semantic framework of Grothendieck institutions. Instead of using a global encoding into some "universal" logic, truly heterogeneous proofs have the advantage to better exploit specialized tool support. Our running example introduces an institution isomorphism from IndexedPropModal to PropModal, allowing indexing for propositional modal logic while still reusing propositional modal proof calculi. This isomorphism also seems to be the first example of a conservative but not model-expansive isomorphism.

Compared with logic combination [19, 6, 5], heterogeneous specification has only weaker forms of feature interaction. Our running example could have been formulated in a nicer way with full first-order modal logic, which can be obtained from combining first-order logic with modal logic, in such a way that even the proof calculi can be combined automatically [6, 5]. However, with this approach, it is still difficult to really obtain combined calculi that are complete. Moreover, the combined calculi have to be implemented. Here, heterogeneous specification and heterogeneous proofs are more flexible: they support better re-use of existing (sometimes highly specialized) proof tools for individual logic. Moreover, there is some kind of feature interaction as well, as demonstrated in our example: propositional modal logic and first-order logic interact through the institution isomorphism \( \rho \) making the worlds explicit. Last but not least, heterogeneous specification is more widely applicable than logic combination: one needs just one metaformalism for all logics, while meta- frameworks for logic combination usually have to be fine-tuned for each new type of logic they are applied to.

Concerning tool development, the development graphs that we have taken as a structuring instrument have been used for industrial-scale applications with hundreds of specifications. Tools such as MAYA [18] provide a management of proofs. We are planning to make MAYA heterogeneous. An abstract interface for logics is provided in form of a Haskell type class, while Haskell's existential types are used to obtain heterogeneity [17].

Acknowledgments Thanks to Andrzej Tarlecki, Serge Autexier and Dieter Hutter for useful cooperation and discussions, and to Lutz Schröder for valuable comments. This work has been supported by the project MULTIPLE of the Deutsche Forschungsgemeinschaft under Grant KR 1191/3-1.

References