Synchronization of logics§

Amílcar Sernadas, Cristina Sernadas and Carlos Caleiro

Departamento de Matemática, IST
Av. Rovisco Pais, 1096 Lisboa, Portugal
email:{acs,css,ccal}@math.ist.utl.pt

November, 1996

Abstract

Motivated by applications in software engineering, we propose two forms of combination of logics: synchronization on formulae and synchronization on models. We start by reviewing satisfaction systems, consequence systems, one-step derivation systems and theory spaces, as well as their functorial relationships. We define the synchronization on formulae of two consequence systems and provide a categorial characterization of the construction. For illustration we consider the synchronization of linear temporal logic and equational logic. We define the synchronization on models of two satisfaction systems and provide a categorial characterization of the construction. We illustrate the technique in two cases: linear temporal logic versus equational logic; and linear temporal logic versus branching temporal logic. Finally, we lift the synchronization on formulae to the category of logics over consequence systems.

Key words: combination of logics, synchronization on formulae, synchronization on models, temporal logic, equational logic.

1 Introduction

The software development process is usually carried out in a heterogeneous environment of methods, languages, formalisms and platforms [1]. No universal formalism is in sight. Instead, it seems necessary to provide combinations of formalisms supporting the different viewpoints. In particular, since specifications in a given formalism can be seen to lead to theories in the appropriate logic (or institution) [6], it seems necessary to provide a variety of mechanisms for combining different logics.

The software engineering community has recently started to address the problem of relating, structuring and combining specification logics (for some theoretical results see [7, 21, 1]). But the bulk of contributions to the area of combination of logics has been made elsewhere with different motivations (see for instance [24, 15, 9, 10, 12, 4]).

§To appear in the Journal *Studia Logica*, special issue on "Combining Logics", numbers 4 and 5, 1997, edited by Dov Gabbay and Fiora Pirri.

Motivated by the software engineering experience, we present here two new techniques for combining logics: synchronization on formulae and synchronization on models. Both techniques are reminiscent of the technique proposed by Winskel for synchronizing processes on actions [25], leading to a constraint on the resulting behavior.

When synchronizing on formulae, the language of the combined logic is a subset of the cartesian product of the given languages: a pair $\langle \varphi_1, \varphi_2 \rangle$ indicates that φ_1 in one given logic is to be interpreted in the other logic as φ_2, or vice-versa. When synchronizing on models, the language of the combined logic is the disjoint union of the given languages, but the class of models of the resulting logic is a subset of the cartesian product of the given classes of models: the interaction between the two logics is imposed by constraining the class of pairs of models.

Synchronization on formulae can be studied at the level of consequence systems (language plus consequence) since models are irrelevant. Synchronization on models must be studied at the level of satisfaction systems (language plus models plus satisfaction). For this reason we start the paper by reviewing in Section 2 satisfaction systems, consequence systems, one-step derivation systems, and theory spaces. We also present functorial relationships (adjunctions) between their categories, with special attention to the novel adjunction between one-step derivation systems and consequence systems. One-step derivation systems are necessary for the categorial characterization of synchronization on formulae. Theory spaces are included for completeness.

These categories are alternative universes over which we can define logics [11]: satisfaction systems (Hell), consequence/derivation systems (Earth) and theory spaces (Heaven). For instance, logics as institutions [13] can be defined as signature-indexed categories of satisfaction systems, and logics as π-institutions [8, 19] can be defined as signature-indexed categories of consequence systems.

Herein, we avoid the signature-indexing technique typical of the institution approach. Instead, we present first in Sections 3 and 4 the synchronization techniques at the level of the instances of the logics, and only in Section 5 we address the problem of synchronization at the level of the logics. Clearly, following tradition we identify a logic with the collection of its instances.

Section 3 presents the mechanism for synchronization of consequence systems on formulae. Section 4 presents the mechanism for synchronization of satisfaction systems on models. In both sections we study the example of combining linear temporal logic systems and unsorted equational logic systems. In Section 4, we also analyze in some detail the combination of linear and branching temporal logic systems. Both sections follow a common structure: motivation, definition, worked examples, and categorial characterization.

A modicum of category theory is helpful for reading the whole paper. But a significant part of it can be grasped avoiding categorial concepts: basically forget the morphisms and look at categories as classes and at functors as maps, avoiding completely the categorial characterization subsections. As a reference to category theory we recommend the classical textbook by Mac Lane [16]. On foundational issues concerning for instance categories of classes, we adopt Mac Lane's approach. For other approaches see [5].

2 Hell, Earth and Heaven

We review here the basic universes (satisfaction systems, consequence systems, one-step derivation systems, and theory spaces) that we shall need later on, and examine their functorial relationships. Namely, we establish an adjunction between the category of consequence systems and the category of one-step derivation systems via the traditional fixed point construction.

In the sequel, Set_* denotes the category of pointed sets. The objects of this category are sets with a selected element and the morphisms are total maps that preserve the selected objects. Similarly, Cls_* denotes the category of pointed classes.

We first review with slight adaptation the general notions of satisfaction system and satisfaction system morphism, following the ideas in [2].

Definition 2.1 A *satisfaction system* is a triple $S = \langle L, M, \Vdash \rangle$ where:

- $L \in |Set_*|$ is a pointed set with selected element t;
- $M \in |Cls_*|$ is a pointed class with selected element z;
- $\Vdash \subseteq M \times L$ is a relation such that:
  - $\{m \in M : m \Vdash t\} = M$;
  - $\{\varphi \in L : z \Vdash \varphi\} = L$.

Clearly, L is the *language*, M is the class of *models*, $\Vdash$ is the *satisfaction* relation, t is the formula "true" and z is an "improper" model that satisfies all formulae.

Definition 2.2 A *satisfaction system morphism* $h : \langle L, M, \Vdash \rangle \to \langle L', M', \Vdash' \rangle$ is a pair $h = \langle h\!\uparrow, h\!\downarrow \rangle$ such that:

- $h\!\uparrow : L \to L'$ is a morphism in Set_* and
- $h\!\downarrow : M' \to M$ is a morphism in Cls_* such that:

$$m' \Vdash' h\!\uparrow\!(\varphi) \text{ iff } h\!\downarrow\!(m') \Vdash \varphi.$$

Recall that in Set_* morphisms preserve selected elements. Thus, $h\!\uparrow\!(t) = t'$ above. Likewise for morphisms in Cls_* i.e., $h\!\downarrow\!(z') = z$.

Note that the formula translation map $h\!\uparrow$ and the reduct map $h\!\downarrow$ are contravariant.

Prop/Definition 2.3 Satisfaction systems and their morphisms constitute the category *Sat*.

We now review the notions of consequence system and consequence system morphism, following [23].

Definition 2.4 A *consequence system* is a pair $C = \langle L, \vdash \rangle$ where $L \in |Set_*|$ is a pointed set with selected element t and $\vdash: \wp L \to \wp L$ is a map such that:

- $\{t\}^\vdash \subseteq \emptyset^\vdash$;
- $\Gamma \subseteq \Gamma^\vdash$ (reflexivity);
- $(\Gamma^\vdash)^\vdash \subseteq \Gamma^\vdash$ (idempotence);
- if $\Gamma \subseteq \Psi$ then $\Gamma^\vdash \subseteq \Psi^\vdash$ (monotonicity).

Clearly, L is the *language*, t is the formula "true" and $\vdash$ is the *consequence operator*. The properties of the consequence operator are the usual ones. We just point out that idempotence corresponds to cut when defining consequence as a relation.

Definition 2.5 A *consequence system morphism* $h : \langle L, \vdash \rangle \to \langle L', \vdash' \rangle$ is a morphism $h : L \to L'$ in Set_* such that:

$$h(\Gamma^\vdash) \subseteq h(\Gamma)^{\vdash'}.$$

Prop/Definition 2.6 Consequence systems and their morphisms constitute the category *Csy*.

In the sequel, we shall also need the notions of one-step derivation system and one-step derivation system morphism. The idea is to be able to obtain a consequence system from each one-step derivation system by the usual fixed point construction which will be made functorial.

Definition 2.7 A (one-step) *derivation system* is a pair $D = \langle L, \Delta \rangle$ where $L \in |Set_*|$ is a pointed set with selected element t and $\Delta : \wp L \to \wp L$ is a map such that:

- $\Delta(\{t\}) \subseteq \Delta(\emptyset)$;
- $\Gamma \subseteq \Delta(\Gamma)$ (reflexivity);
- if $\Gamma \subseteq \Psi$ then $\Delta(\Gamma) \subseteq \Delta(\Psi)$ (monotonicity).

Derivation systems are similar to consequence systems. The derivation operator does not have the idempotence property, as expected in a one-step operator.

Definition 2.8 A *derivation system morphism* $h : \langle L, \Delta \rangle \to \langle L', \Delta' \rangle$ is a morphism $h : L \to L'$ in Set_* such that:

$$h(\Delta(\Gamma)) \subseteq \Delta'(h(\Gamma)).$$

Prop/Definition 2.9 Derivation systems and their morphisms constitute the category *Dsy*.

Finally, for the sake of the completeness of the picture, we also introduce the notions of theory space and theory space morphism.

Definition 2.10 A *theory space* is a complete lattice $P = \langle \Omega, \leq \rangle$.

Definition 2.11 A *theory space morphism* $h : \langle \Omega, \leq \rangle \to \langle \Omega', \leq' \rangle$ is a map $h : \Omega \to \Omega'$ such that $h(\bigvee A) = \bigvee h(A)$.

Note that although both joins and meets exist in a theory space only joins are required to be preserved by the morphisms.

Prop/Definition 2.12 Theory spaces and their morphisms constitute the category *Tsp*.

We establish now functorial relationships between these categories, starting with a coreflection between satisfaction systems and consequence systems.

Prop/Definition 2.13 The maps

- $F(\langle L, \vdash \rangle) = \langle L, \{\Gamma^{\vdash} : \Gamma \in \wp L\}, \in \rangle$
  with L as the selected element in $\{\Gamma^{\vdash} : \Gamma \in \wp L\}$;
- $F(h : \langle L, \vdash \rangle \to \langle L', \vdash' \rangle) = \langle h, h^{-1} \rangle$

constitute the functor $F : Csy \to Sat$.

Prop/Definition 2.14 The maps

- $G(\langle L, M, \Vdash \rangle) = \langle L, \models \rangle$ and
- $G(h : \langle L, M, \Vdash \rangle \to \langle L', M', \Vdash' \rangle) = h{\uparrow}$

constitute the functor $G : Sat \to Csy$.

Clearly, $\models$ above denotes the semantic entailment operator induced by the satisfaction relation at hand: $\Gamma^{\models} = \{\varphi : \Gamma \models \varphi\}$ where $\Gamma \models \varphi$ iff for each model m if $m \Vdash \gamma$ for every $\gamma \in \Gamma$ then $m \Vdash \varphi$.

Theorem 2.15 The adjunction $F \dashv G$ is a coreflection.

Proof:
The counit is $\epsilon = \{\epsilon_S\}_{S \in |Sat|}$ where $\epsilon_S = \langle \epsilon_S{\uparrow}, \epsilon_S{\downarrow} \rangle$ is defined as follows for each $S = \langle L, M, \Vdash \rangle$:

- $\epsilon_S{\uparrow} = id_L$;
- $\epsilon_S{\downarrow} = \lambda\, m.\, \{\varphi \in L : m \Vdash \varphi\}$.

$$
\begin{tikzcd}
Csy \arrow[r, shift left, "F"] & Sat \arrow[l, shift left, "G"]
\end{tikzcd}
$$

$$
\begin{tikzcd}
\langle L, \models \rangle & & F(\langle L, \models \rangle) \arrow[r, "\epsilon_S"] & \langle L, M, \Vdash \rangle \\
\langle L', \vdash' \rangle \arrow[u, dotted, "k"'] & & F(\langle L', \vdash' \rangle) \arrow[u, "F(k)"] \arrow[ur, "g"'] &
\end{tikzcd}
$$

Figure 1: Universal property of the counit morphism ϵ_S for $S = \langle L, M, \Vdash \rangle$, with $C' = \langle L', \vdash' \rangle$.

1. $\epsilon_S : F(\langle L, \models \rangle) \to \langle L, M, \Vdash \rangle$ is a morphism in Sat:

$\epsilon_S\!\downarrow\!(m)$ is closed under $\models$, both $\epsilon_S\!\uparrow$ and $\epsilon_S\!\downarrow$ preserve selected elements and the satisfaction condition is immediate.

2. Universal property (see Figure 1):

Given any morphism $g : F(C') \to S$ in Sat the unique $k : C' \to G(S)$ in Csy such that $\epsilon_S \circ F(k) = g$ is $k = g\!\uparrow$.

3. The unit $\eta_{\langle L, \vdash \rangle}$ is $id_{\langle L, \vdash \rangle}$:

The semantic entailment in $G(F(\langle L, \vdash \rangle))$ is defined by $\Psi \models \varphi$ iff $\Psi \subseteq \Gamma^{\vdash} \Rightarrow \varphi \in \Gamma^{\vdash}$ for every $\Gamma^{\vdash}$ such that $\Gamma \in \wp L$. QED

As expected there is also a very close relationship between the categories *Dsy* and *Csy*: a reflection. But first we need to recall the following fixed point result originally due to Tarski (for the proof see [22] and for further details see [17]).

Theorem 2.16 Let $\langle U, \leq \rangle$ be a complete lattice and $f : U \to U$ a monotonic map such that $u \leq f(u)$. Then, for each $u \in U$ there is an ordinal ν such that $f(f^{\nu}(u)) = f^{\nu}(u)$ and $u \leq f^{\nu}(u)$.

In the sequel, we denote $f^{\nu}(u)$ by $\mathrm{lfp}(f, u)$ where ν is the least ordinal such that $f(f^{\nu}(u)) = f^{\nu}(u)$. One might expect that the fixed point construction is functorial. Indeed:

Prop/Definition 2.17 The maps

- $X(\langle L, \Delta \rangle) = \langle L, \vdash \rangle$ where $\Gamma^{\vdash} = \mathrm{lfp}(\Delta, \Gamma)$;
- $X(h : \langle L, \Delta \rangle \to \langle L', \Delta' \rangle) = h$

constitute the functor $X : Dsy \to Csy$.

Proof:

Clearly, we are working in the complete lattice $\langle \wp L, \subseteq \rangle$.

1. $X(\langle L, \Delta \rangle)$ is a consequence system:

Clearly, $\{t\}^{\vdash} \subseteq \emptyset^{\vdash}$ due to the fact that $\Delta(\{t\}) \subseteq \Delta(\emptyset)$ and using also the monotonicity of Δ. Reflexivity is immediate from the reflexivity and monotonicity of Δ. Clearly, $\text{lfp}(\Delta, \text{lfp}(\Delta, \Gamma)) = \text{lfp}(\Delta, \Gamma)$, and thus idempotence holds. Also monotonicity easily follows from the monotonicity of Δ.

2. $X(h)$ is a morphism in *Csy*:
We start by proving by induction that $h(\Delta^{\nu}(\Gamma)) \subseteq \Delta'^{\nu}(h(\Gamma))$ for any ordinal ν. Easily, $h(\Gamma) = h(\Delta^0(\Gamma)) \subseteq \Delta'^0(h(\Gamma)) = h(\Gamma)$, which sets the base case.
We first prove the induction step for any successor ordinal $\nu = \epsilon + 1$:

$$\begin{aligned} h(\Delta^{\epsilon+1}(\Gamma)) &= h(\Delta(\Delta^{\epsilon}(\Gamma))) \\ &\subseteq \Delta'(h(\Delta^{\epsilon}(\Gamma))) && (h \text{ is a morphism in } Dsy) \\ &\subseteq \Delta'(\Delta'^{\epsilon}(h(\Gamma))) && (\text{induction hypothesis and monotonicity of } \Delta'). \end{aligned}$$

We now prove $h(\Delta^{\nu}(\Gamma)) \subseteq \Delta'^{\nu}(h(\Gamma))$ for ν any limit ordinal, i.e. $\Delta^{\nu}(\Gamma) = \bigvee\{\Delta^{\epsilon}(\Gamma) : \epsilon < \nu\}$:

$$\begin{aligned} h(\bigvee\{\Delta^{\epsilon}(\Gamma) : \epsilon < \nu\}) &= \bigvee h(\{\Delta^{\epsilon}(\Gamma) : \epsilon < \nu\}) \\ &\subseteq \bigvee\{\Delta'^{\epsilon}(h(\Gamma)) : \epsilon < \nu\} && (\text{induction hypothesis}) \\ &= \Delta'^{\nu}(h(\Gamma)). \end{aligned}$$

Therefore, it is immediate that $h(\Gamma^{\vdash}) \subseteq h(\Gamma)^{\vdash'}$. QED

Prop/Definition 2.18 The maps

- $Y(\langle L, \vdash\rangle) = \langle L, \vdash\rangle$;
- $Y(h : \langle L, \vdash\rangle \to \langle L', \vdash'\rangle) = h$

constitute the functor $Y : Csy \to Dsy$.

Theorem 2.19 The adjunction $X \dashv Y$ is a reflection.

Proof:
The unit is $\eta = \{\eta_D\}_{D \in |Dsy|}$ where $\eta_D = id_L$ for each $D = \langle L, \Delta\rangle$.
1. $\eta_D : \langle L, \Delta\rangle \to Y(X(\langle L, \Delta\rangle))$ is a morphism in *Dsy*:
Obviously, $\Gamma \subseteq \Gamma^{\vdash} = \text{lfp}(\Delta, \Gamma)$. Thus, $\Delta(\Gamma) \subseteq \Gamma^{\vdash}$ by monotonicity of Δ and the definition of fixed point.

2. Universal property:

Given any morphism $h : D \to Y(C')$ in *Dsy* the unique morphism $g : X(D) \to C'$ in *Csy* such that $Y(g) \circ \eta_D = h$ is $g = h$.

2.1. $h : X(D) \to C'$ is a morphism in *Csy*:

It is enough to prove, by induction, that $h(\Delta^{\nu}(\Gamma)) \subseteq h(\Gamma)^{\vdash'}$ for any ordinal ν. For the base case, easily, $h(\Delta^0(\Gamma)) = h(\Gamma) \subseteq h(\Gamma)^{\vdash'}$ since $\vdash'$ must be reflexive. We first prove the induction step for any successor ordinal $\nu = \epsilon + 1$:

$$\begin{aligned} h(\Delta^{\epsilon+1}(\Gamma)) &= h(\Delta(\Delta^{\epsilon}(\Gamma))) \\ &\subseteq h(\Delta^{\epsilon}(\Gamma))^{\vdash'} && (h \text{ is a morphism in } Dsy) \\ &\subseteq (h(\Gamma)^{\vdash'})^{\vdash'} && (\text{induction hypothesis and monotonicity of } \vdash') \\ &\subseteq h(\Gamma)^{\vdash'} && (\text{idempotence of } \vdash'). \end{aligned}$$

We now prove $h(\Delta^{\nu}(\Gamma)) \subseteq h(\Gamma)^{\vdash'}$ for ν any limit ordinal:
$h(\Delta^{\nu}(\Gamma)) = h(\bigvee\{\Delta^{\epsilon}(\Gamma) : \epsilon < \nu\}) = \bigvee\{h(\Delta^{\epsilon}(\Gamma)) : \epsilon < \nu\} \subseteq h(\Gamma)^{\vdash'}$ (induction hypothesis and definition of join).

2.2. Commutativity and uniqueness:
Clearly, $Y(h) \circ \eta_D = h$ and uniqueness follows from the fact that $\eta_D = id_L$.

3. The counit $\epsilon_{\langle L,\vdash\rangle}$ is $id_{\langle L,\vdash\rangle}$:
The consequence operation in $X(Y(\langle L,\vdash\rangle))$ is $\lambda\,\Gamma.\text{lfp}(\vdash,\Gamma)$. But, since $\vdash$ must be idempotent, $\Gamma^{\vdash} = \text{lfp}(\vdash,\Gamma)$. QED

Finally, we establish a reflection between the categories of consequence systems and theory spaces.

Prop/Definition 2.20 The maps

- $H(\langle L,\vdash\rangle) = \langle\{\Gamma^{\vdash} : \Gamma \in \wp L\}, \subseteq\rangle$;
- $H(h : \langle L,\vdash\rangle \to \langle L',\vdash'\rangle) = \lambda\,\Gamma.\, h(\Gamma)^{\vdash'}$

constitute the functor $H : Csy \to Tsp$.

Clearly, the join of theories is the closure of their union. And the meet is simply their intersection (no need to close since the intersection of theories is itself a theory).

Prop/Definition 2.21 The maps

- $K(\langle\Omega,\leq\rangle) = \langle\Omega, \lambda\,A.\,\{a \in \Omega : a \leq \bigvee A\}\rangle$,
  with selected element $\bigvee\emptyset$;
- $K(h : \langle\Omega,\leq\rangle \to \langle\Omega',\leq'\rangle) = h$

constitute the functor $K : Tsp \to Csy$.

Theorem 2.22 The adjunction $H \dashv K$ is a reflection.

Proof:
The unit is $\eta = \{\eta_C\}_{C\in|Csy|}$ where $\eta_C = \lambda\,\varphi.\,\{\varphi\}^{\vdash}$ for each $C = \langle L,\vdash\rangle$.
1. $\eta_C : \langle L,\vdash\rangle \to K(H(\langle L,\vdash\rangle))$ is a morphism in Csy:
$\{\varphi\}^{\vdash} \in \{\Gamma^{\vdash} : \Gamma \in \wp L\}$, the selected element is preserved $\{t\}^{\vdash} = \emptyset^{\vdash} = \bigvee\emptyset$ and since $\Gamma^{\vdash} \subseteq (\bigcup\{\{\varphi\}^{\vdash} : \varphi \in \Gamma\})^{\vdash} = \bigvee\{\{\varphi\}^{\vdash} : \varphi \in \Gamma\} = \bigvee\eta_C(\Gamma)$ also $\eta_C(\varphi) = \{\varphi\}^{\vdash} \subseteq \Gamma^{\vdash} \subseteq \bigvee\eta_C(\Gamma)$ for each $\varphi \in \Gamma^{\vdash}$.

2. Universal property:
Given any morphism $f : C \to K(P')$ in Csy the unique $g : H(C) \to P'$ such that $K(g) \circ \eta_C = f$ is $g = \lambda\,\Gamma.\bigvee f(\Gamma)$.

3. The counit $\epsilon_{\langle\Omega,\leq\rangle}$ is an isomorphism:
$\epsilon_{\langle\Omega,\leq\rangle} = \lambda\,A.\bigvee A$ with inverse $\epsilon^{-1}_{\langle\Omega,\leq\rangle} = \lambda\,a.\{b \in \Omega : b \leq a\}$. QED

$$\begin{tikzcd} Sat \arrow[r, shift right, "G"'] & Csy \arrow[l, shift right, "F"'] \arrow[r, shift left, "H"] \arrow[d, shift left, "Y"] & Tsp \arrow[l, shift left, "K"] \\ & Dsy \arrow[u, shift left, "X"] & \end{tikzcd}$$

Figure 2: The whole picture: *Sat*, *Csy*, *Dsy*, *Tsp* and the functors between them.

3 Synchronizing consequence systems

3.1 Motivation

We are given two consequence systems C_1, C_2, say of linear temporal logic (LT) and unsorted equational logic (EQ) respectively. We would like to combine the two systems in order to be able to work in the same system with both kinds of reasoning. Furthermore, by imposing that certain formulae of C_1 have counterparts in C_2 and vice-versa, we would like that results of say equational reasoning are propagated to the other side and can be used to make further derivations using temporal reasoning.

The main ideas underlying the proposed synchronization of consequence systems on formulae are: (1) the language of the combined system is included in the cartesian product of the given languages; (2) not all pairs are used since we may want to impose that for instance the equation $g(c) = h(c)$ of EQ goes together with the formula $a_{g(c)} \Leftrightarrow a_{h(c)}$ of LT; (3) when a formula has no counterpart in the other logic we impose that it goes together with the t of the other logic; (4) the envisaged consequence operator of the resulting system should allow the mixed reasoning taking advantage of the two given consequence operators and of the "bridges" between the two logics established by the pairs. For instance, from $\Gamma = \{\langle t, g(x) = h(x)\rangle, \langle a_{g(c)}, x = x\rangle\}$ we can on EQ grounds derive $g(c) = h(c)$ (by substitution), that is we reach the pair $\langle a_{g(c)} \Leftrightarrow a_{h(c)}, g(c) = h(c)\rangle$. Then, on LT grounds (actually purely propositional reasoning) we can derive $a_{h(c)}$, that is we reach the pair $\langle a_{h(c)}, x = x\rangle$.

This example is presented in detail below, after the formal definition of synchronization of consequence systems on formulae.

3.2 Synchronization on formulae

Before defining formally the envisaged synchronization construction we need to recall that products exist in Set_*. The product of L_1 and L_2 is the cartesian product $L_1 \times L_2$ with selected element $\langle t_1, t_2\rangle$ and with the usual projections p_1 and p_2.

Prop/Definition 3.1 Let $C_1 = \langle L_1, \vdash_1\rangle$ and $C_2 = \langle L_2, \vdash_2\rangle$ be consequence systems. Let $\alpha_1 : L_1 \to L_1 \times L_2$ and $\alpha_2 : L_2 \to L_1 \times L_2$ be morphisms in Set_*

such that $p_1(\alpha_1(\varphi_1)) = \varphi_1$ and $p_2(\alpha_2(\varphi_2)) = \varphi_2$. Then, the synchronization on formulae of C_1 and C_2 by α_1 and α_2 is the consequence system

$$C_1 \overset{\alpha_1\alpha_2}{\bullet} C_2 = \langle L, \vdash \rangle$$

where:

- $L = \alpha_1(L_1) \cup \alpha_2(L_2)$;
- $\Gamma^{\vdash} = \mathrm{lfp}(\Delta, \Gamma)$ where $\Delta(\Gamma) = \Gamma \cup \alpha_1(p_1(\Gamma)^{\vdash_1}) \cup \alpha_2(p_2(\Gamma)^{\vdash_2})$.

Proof:
Clearly, Δ is reflexive and monotonic. Moreover, since $\{t_1\}^{\vdash_1} \subseteq \emptyset^{\vdash_1}$ and $\{t_2\}^{\vdash_2} \subseteq \emptyset^{\vdash_2}$ we have that $\alpha_1(\{t_1\}^{\vdash_1}) \cup \alpha_2(\{t_2\}^{\vdash_2}) \subseteq \alpha_1(\emptyset^{\vdash_1}) \cup \alpha_2(\emptyset^{\vdash_2})$. Thus, clearly $\langle t_1, t_2 \rangle = \alpha_1(t_1) = \alpha_2(t_2) \in \Delta(\emptyset)$, and $\Delta(\{\langle t_1, t_2 \rangle\}) \subseteq \Delta(\emptyset)$. Furthermore, $X(\langle L, \Delta \rangle) = C_1 \overset{\alpha_1\alpha_2}{\bullet} C_2$. QED

In this construction α_1 and α_2 are called the formula-synchronization maps. When combining two given consequence systems the choice of these maps is crucial. The following guidelines are useful for choosing the value of $\alpha_1(\varphi_1) = \langle \varphi_1, \varphi_2 \rangle$:

1. φ_2 will play the role of the counterpart in C_2 of φ_1;
2. if no formula of C_2 represents φ_1, then φ_2 should be t_2.

And the same applies mutatis mutandis to α_2. Clearly, by imposing $\alpha_1(\varphi_1) = \langle \varphi_1, t_2 \rangle$ for every φ_1 and $\alpha_2(\varphi_2) = \langle t_1, \varphi_2 \rangle$ for every φ_2 we obtain the extreme case in which no interaction is prescribed between the two systems.

Proposition 3.2 The formula-synchronization maps are morphisms in *Csy*:

- $\alpha_1 : C_1 \to C_1 \overset{\alpha_1\alpha_2}{\bullet} C_2$;
- $\alpha_2 : C_2 \to C_1 \overset{\alpha_1\alpha_2}{\bullet} C_2$.

Proof:
Clearly, for each $i = 1, 2$, we have:
$\alpha_i(\Gamma_i^{\vdash_i}) = \alpha_i(p_i(\alpha_i(\Gamma_i))^{\vdash_i}) \subseteq \Delta(\alpha_i(\Gamma_i)) \subseteq \alpha_i(\Gamma_i)^{\vdash}$. QED

Therefore, the consequences in C_i for each $i = 1, 2$ are preserved in the resulting system $C_1 \overset{\alpha_1\alpha_2}{\bullet} C_2$. But note that the extension is not necessarily conservative: we may infer more in the combined system within the language of one of the given consequence systems. Of course, in the case of no interaction the resulting extension is conservative.

3.3 Worked example: LT versus EQ

3.3.1 Propositional linear temporal logic

Recall that a propositional linear temporal signature Π is just a set (of propositional symbols). The set of Π-formulae is inductively defined as follows:

$$L_\Pi = t \mid \Pi \mid (\neg\, L_\Pi) \mid (L_\Pi \Rightarrow L_\Pi) \mid (\mathsf{X}\, L_\Pi) \mid (\mathsf{G}\, L_\Pi).$$

For the sake of simplicity we do not use the "until" operator so common in linear temporal logic.

In the sequel, we feel free to use the well known abbreviations of the derived propositional connectives e.g., $\Leftrightarrow$. Moreover, we also use the abbreviation $(\mathsf{F}\,\varphi) \stackrel{abv}{\equiv} (\neg(\mathsf{G}(\neg\,\varphi)))$.

For each propositional linear temporal logic signature Π, we establish the consequence system:

$$cLT_\Pi = \langle L_\Pi, \vdash_\Pi \rangle$$

as follows:

- L_Π is the the set of Π-formulae with selected element t;
- $\Gamma^{\vdash_\Pi}$ is the set of Π-formulae that can be derived from Γ.

We do not describe in detail a derivation calculus for this temporal logic: see instead [14] for instance. However, it is useful to recall the following examples of derivations:

- $\emptyset \vdash_\Pi (\varphi \Rightarrow \varphi)$;
- $\emptyset \vdash_\Pi ((\mathsf{G}\,\varphi) \Rightarrow (\mathsf{X}\,\varphi))$;
- $\{\varphi\} \vdash_\Pi (\mathsf{G}\,\varphi)$.

3.3.2 Unsorted equational logic

Recall that an unsorted equational signature Σ is a family $\{\Sigma_n\}_{n\in I\!N}$ of pairwise disjoint sets: each Σ_n is the set of operation symbols of arity n. Recall also that a Σ-equation is a triple $\langle X, u, u'\rangle$ where X is a set (of variables) disjoint from each Σ_n, and u, u' are Σ-terms over X, that is elements of $T_\Sigma(X)$. We may write $u = u'$ taking X to be the set of variables occurring in u or u'.

For each unsorted equational signature Σ, we establish the consequence system

$$cEQ_\Sigma = \langle L_\Sigma, \vdash_\Sigma \rangle$$

as follows:

- L_Σ is the the set of Σ-equations with selected element $\langle \{x\}, x, x\rangle$;
- $\Gamma^{\vdash_\Sigma}$ is the set of equations that can be derived from Γ.

Again, we do not describe in detail a derivation calculus for this equational logic: see instead [18] for instance. However, it is useful to recall the following examples of derivations:

- $\emptyset \vdash_\Sigma x = x$;
- $\{u = u'\} \vdash_\Sigma u' = u$;
- $\{u = u'\} \vdash_\Sigma u^x_v = u'^x_v$.

Clearly, u^x_v is the term obtained from u by replacing simultaneously and uniformly each occurrence of the variable x by the term v.

3.3.3 Synchronization on formulae

Assume that we are given the consequence systems cLT_{Π} and cEQ_{Σ} with the proviso that $\{a_u : u \in T_{\Sigma}(\emptyset)\} \subseteq \Pi$. The practical motivation is to combine temporal reasoning according to cLT_{Π} with equational reasoning according to cEQ_{Σ} in order to specify the behavior of an object with the time-dependent attribute a: the formula a_u is intended to assert that the value of a is u. The equational reasoning is used for deriving assertions on values. The temporal reasoning is used for deriving assertions about the behavior.

We choose the formula-synchronizing maps to be as follows:

- $\alpha_1 : L_{\Pi} \to L_{\Pi} \times L_{\Sigma}$ such that $\alpha_1(\varphi) = \langle \varphi, x = x \rangle$;
- $\alpha_2 : L_{\Sigma} \to L_{\Pi} \times L_{\Sigma}$ such that

$$\alpha_2(\langle X, u, u' \rangle) = \begin{cases} \langle (a_u \Leftrightarrow a_{u'}), \langle X, u, u' \rangle \rangle & \text{pt } X = \emptyset \\ \langle t, \langle X, u, u' \rangle \rangle & \text{otherwise} \end{cases} .$$

Clearly, we do not want to pass any assertion on the temporal side to the equational side, hence the definition of α_1 putting together every temporal formula with $x = x$. On the other hand, we do want to pass assertions about values to the temporal side: therefore, ground equations are put together with the corresponding equivalences.

Let us see an example of "mixed" deduction in $cLT_{\Pi} \overset{\alpha_1 \ \alpha_2}{\bullet} cEQ_{\Sigma}$. It is easy to verify that in the resulting consequence system we have:

$$\langle a_{h(c)}, x = x \rangle \in \{\langle a_{g(c)}, x = x \rangle, \langle t, g(x) = h(x) \rangle\}^{\vdash}.$$

Indeed, letting $\Gamma = \{\langle a_{g(c)}, x = x \rangle, \langle t, g(x) = h(x) \rangle\}$, we have:

- $g(x) = h(x) \in p_2(\Gamma)$;
- $g(c) = h(c) \in p_2(\Gamma)^{\vdash_{\Sigma}}$;
- $\langle (a_{g(c)} \Leftrightarrow a_{h(c)}), g(c) = h(c) \rangle \in \alpha_2(p_2(\Gamma)^{\vdash_{\Sigma}})$;
- $\langle (a_{g(c)} \Leftrightarrow a_{h(c)}), g(c) = h(c) \rangle \in \Delta(\Gamma)$;
- similarly, $\langle a_{g(c)}, x = x \rangle \in \Delta(\Gamma)$;
- $(a_{g(c)} \Leftrightarrow a_{h(c)}), a_{g(c)} \in p_1(\Delta(\Gamma))$;
- $a_{h(c)} \in p_1(\Delta(\Gamma))^{\vdash_{\Pi}}$;
- $\langle a_{h(c)}, x = x \rangle \in \alpha_1(p_1(\Delta(\Gamma))^{\vdash_{\Pi}})$;
- $\langle a_{h(c)}, x = x \rangle \in \Delta^2(\Gamma)$;
- $\langle a_{h(c)}, x = x \rangle \in \Gamma^{\vdash}$.

3.4 Categorial characterization

We already pointed out that the formula-synchronizing maps α_1, α_2 are morphisms in *Csy*. We can further characterize the nature of $C_1 \overset{\alpha_1 \alpha_2}{\bullet} C_2$ within the category *Csy*. We first consider the case when the formula-synchronizing maps introduce no interaction:

Proposition 3.3 Let $i_1 : L_1 \to L_1 \times L_2$ and $i_2 : L_2 \to L_1 \times L_2$ be the morphisms in Set_* such that:

- $i_1(\varphi_1) = \langle \varphi_1, t_2 \rangle$;
- $i_2(\varphi_2) = \langle t_1, \varphi_2 \rangle$.

Then, $C_1 \overset{i_1 i_2}{\bullet} C_2$ is the coproduct $C_1 + C_2$ of C_1 and C_2 with the injections i_1 and i_2.

Proof:
1. i_1 and i_2 are morphisms in *Csy*:
Clearly, i_j is a formula-synchronization map for $j = 1, 2$ i.e., $p_j(i_j(\varphi_j)) = \varphi_j$. Therefore, both are morphisms according to 3.2.
2. Universal property:
Let $h_1 : C_1 \to C'$ and $h_2 : C_2 \to C'$ be any consequence system morphisms. Consider the map $h : C_1 \overset{i_1 i_2}{\bullet} C_2 \to C'$ such that $h(i_1(\varphi_1)) = h_1(\varphi_1)$ and $h(i_2(\varphi_2)) = h_2(\varphi_2)$. Of course h is a consequence system morphism and it is the unique such that $h \circ i_1 = h_1$ and $h \circ i_2 = h_2$. QED

The diagram in Figure 3 depicts the universal property of this coproduct with respect to any formula-synchronization of C_1 and C_2. The morphism in Set_*

$$(\alpha_1 + \alpha_2) : L_1 + L_2 \to \alpha_1(L_1) \cup \alpha_2(L_2)$$

is such that $(\alpha_1 + \alpha_2)(i_j(\varphi_j)) = \alpha_j(\varphi_j)$ for $j = 1, 2$.

$$\begin{tikzcd}
C_1 \arrow[dr, "i_1"] \arrow[ddr, bend right, "\alpha_1"'] & & C_2 \arrow[dl, "i_2"'] \arrow[ddl, bend left, "\alpha_2"] \\
& C_1 + C_2 \arrow[d, dotted, "\alpha_1+\alpha_2"] & \\
& C_1 \overset{\alpha_1 \alpha_2}{\bullet} C_2 &
\end{tikzcd}$$

Figure 3: $\alpha_1 + \alpha_2$ is the unique morphism such that $(\alpha_1 + \alpha_2) \circ i_1 = \alpha_1$ and $(\alpha_1 + \alpha_2) \circ i_2 = \alpha_2$.

Proposition 3.4 For every $\alpha_1 : L_1 \to L_1 \times L_2$ and $\alpha_2 : L_2 \to L_1 \times L_2$ in Set_* and every consequence system C' such that there are morphisms $h_1 : C_1 \to C'$ and $h_2 : C_2 \to C'$, there is at most one morphism k from $C_1 \overset{\alpha_1 \alpha_2}{\bullet} C_2$ to C' in *Csy* such that:

- $h_1 = k \circ \alpha_1$;
- $h_2 = k \circ \alpha_2$.

Proof:
Straightforward consequence of the universal property of the coproduct and the fact that $(\alpha_1 + \alpha_2)$ is an epimorphism. QED

It remains to see if the consequence system $C_1 \overset{\alpha_1 \alpha_2}{\bullet} C_2$ can be obtained from C_1 and C_2 by a categorial construction in the general case (for any formula-synchronizing maps). To this end, we have to use the relationship between *Csy* and *Dsy*, since the categorial construction we want is carried out in the latter category.

Prop/Definition 3.5 The maps

- $N(\langle L, \Delta \rangle) = L$;
- $N(h : \langle L, \Delta \rangle \to \langle L', \Delta' \rangle) = h$

constitute the functor $N : Dsy \to Set_*$.

Proposition 3.6 For each $\langle L, \Delta \rangle$ in *Dsy* and each map $h : L \to L'$ in Set_* such that $\Delta(h^{-1}(\{t'\})) \subseteq \Delta(\emptyset)$, the morphism $h : \langle L, \Delta \rangle \to \langle L', \Delta' \rangle$ is cocartesian by N for h on $\langle L, \Delta \rangle$ where $\Delta'(\Gamma') = \Gamma' \cup h(\Delta(h^{-1}(\Gamma')))$.

Proof:
1. $\langle L', \Delta' \rangle$ is a derivation system:
Reflexivity is immediate from the definition of Δ' and monotonicity is straightforward from the fact that $h^{-1}(\Gamma') \subseteq h^{-1}(\Psi')$ whenever $\Gamma' \subseteq \Psi'$. Moreover, $\Delta'(\{t'\}) = \{t'\} \cup h(\Delta(h^{-1}(\{t'\}))) \subseteq \{t'\} \cup h(\Delta(\emptyset)) = \Delta'(\emptyset)$.
2. h is a morphism in *Csy*:
Easily, $h(\Delta(\Gamma)) \subseteq h(\Delta(h^{-1}(h(\Gamma)))) \subseteq \Delta'(h(\Gamma))$.
3. Universal property (see Figure 4):
Assume that $g : \langle L, \Delta \rangle \to \langle L'', \Delta'' \rangle$ is any morphism in *Csy* and that $f : L' \to L''$ is any morphism in Set_* such that $f \circ h = N(g)$. Then, f is the unique morphism in *Csy* such that $f \circ h = g$. We just prove that f is a morphism in *Csy*. Let $\varphi' \in \Delta'(\Gamma')$. We have two cases. Firstly assume that $\varphi' \in h(\Delta(h^{-1}(\Gamma')))$. Then, $f(\varphi') \in f(h(\Delta(h^{-1}(\Gamma'))))$ and so $f(\varphi') \in N(g)(\Delta(h^{-1}(\Gamma')))$. Therefore, $f(\varphi') \in \Delta''(N(g)(h^{-1}(\Gamma')))$, $f(\varphi') \in \Delta''(f(h(h^{-1}(\Gamma'))))$ and so $f(\varphi') \in \Delta''(f(\Gamma'))$. Secondly assume that $\varphi' \in \Gamma'$. Then, $f(\varphi') \in f(\Gamma')$ and so $f(\varphi') \in \Delta''(f(\Gamma'))$. QED

We denote the codomain of the cocartesian morphism above by $h(\langle L, \Delta \rangle)$.

Proposition 3.7 $C_1 \overset{\alpha_1 \alpha_2}{\bullet} C_2 = X((\alpha_1 + \alpha_2)(Y(C_1 + C_2)))$.

$$\begin{tikzcd} Dsy \arrow[r, "N"] & Set_* \end{tikzcd}$$

$$\begin{tikzcd} \langle L,\Delta\rangle \arrow[r, "h"] \arrow[dr, "g"'] & h(\langle L,\Delta\rangle) \arrow[d, dashed, "f"] \\ & \langle L'',\Delta''\rangle \end{tikzcd} \qquad \begin{tikzcd} L \arrow[r, "h"] \arrow[dr, "N(g)"'] & L' \arrow[d, "f"] \\ & L'' \end{tikzcd}$$

Figure 4: Universal property of the cocartesian morphism $h : \langle L,\Delta\rangle \to h(\langle L,\Delta\rangle)$ by N for $h : L \to L'$ on $\langle L,\Delta\rangle$, in case $\Delta(h^{-1}(\{t'\})) \subseteq \Delta(\emptyset)$.

Proof:
Note that $(\alpha_1 + \alpha_2)^{-1}(\Gamma) = \{i_1(\varphi_1) : \alpha_1(\varphi_1) \in \Gamma\} \cup \{i_2(\varphi_2) : \alpha_2(\varphi_2) \in \Gamma\}$. Moreover, $(\alpha_1 + \alpha_2)^{-1}(\Gamma) = i_1(p_1(\Gamma)) \cup i_2(p_2(\Gamma))$. We have:

$Y(C_1 + C_2) = C_1 + C_2$;

$(\alpha_1 + \alpha_2)(Y(C_1 + C_2)) = \langle \alpha(L_1) \cup \alpha_2(L_2), \Delta\rangle$

where $\Delta(\Gamma) = \Gamma \cup (\alpha_1 + \alpha_2)((\alpha_1 + \alpha_2)^{-1}(\Gamma)^{\vdash})$ since 3.6 applies i.e., $(\alpha_1 + \alpha_2)(i_j(\varphi_j)) = \langle t_1, t_2\rangle$ iff $p_j(\alpha_j(\varphi_j)) = t_j$ iff $\varphi_j = t_j$ for $j = 1, 2$;

$X((\alpha_1 + \alpha_2)(Y(C_1 + C_2))) = C_1 \overset{\alpha_1\,\alpha_2}{\bullet} C_2$. QED

4 Synchronizing satisfaction systems

4.1 Motivation

We are given two satisfaction systems S_1, S_2, say of linear temporal logic (LT) and equational logic (EQ) respectively. We would like to combine the two systems in order to be able to work in a single satisfaction system with formulae from both. The models should be pairs putting together a model from each of the given satisfaction systems.

For synchronization purposes we just forbid some of the pairs. For instance, we may want to impose that we keep only the pairs $\langle m_1, m_2\rangle$ such that if $m_2 \Vdash_\Sigma g(c) = h(c)$ then $m_1 \Vdash_\Pi (a_{g(c)} \Leftrightarrow a_{h(c)})$, in order to achieve at the model level an interaction similar to the one described in the previous section at the formula level.

4.2 Synchronization on models

Before defining formally the envisaged synchronization construction we need to recall that coproducts exist in Set_*. The coproduct $L_1 + L_2$ is the disjoint union of $L_1 \setminus \{t_1\}$ and $L_2 \setminus \{t_2\}$ plus the selected element t and with the expected injections i_1 and i_2. Clearly, $i_j(t_j) = t$ for $j = 1, 2$.

We also need to use the products in Cls_*. The product of M_1 and M_2 is the class $M_1 \times M_2$ with selected element $\langle z_1, z_2\rangle$ and with the usual projections P_1

and P_2.

Definition 4.1 Let $S_1 = \langle L_1, M_1, \Vdash_1\rangle$ and $S_2 = \langle L_2, M_2, \Vdash_2\rangle$ be satisfaction systems. Let $R \subseteq M_1 \times M_2$. Then, the synchronization on models of S_1 and S_2 by R is the satisfaction system

$$S_1 \overset{R}{\bullet} S_2 = \langle L, R, \Vdash\rangle$$

where:

- $L = L_1 + L_2$;
- $\langle m_1, m_2\rangle \Vdash i_j(\varphi_j)$ iff $m_j \Vdash_j \varphi_j$.

In this construction R is called the model-synchronization relation. When combining two given satisfaction systems a good choice of this relation is obviously crucial. Clearly, if no interaction is desired then R should be chosen to be $M_1 \times M_2$.

Proposition 4.2 The following pairs are morphisms in *Sat*:

- $\langle i_1, P_1|_R\rangle : S_1 \to S_1 \overset{R}{\bullet} S_2$;
- $\langle i_2, P_2|_R\rangle : S_2 \to S_1 \overset{R}{\bullet} S_2$.

Proof:
The result is immediate from the definition of $\Vdash$ since, for $j = 1, 2$:
$P_j|_R(\langle m1, m2\rangle) = m_j$ provided that $\langle m1, m2\rangle \in R$. QED

Corollary 4.3 The following maps are morphisms in *Csy*:

- $i_1 : G(S_1) \to G(S_1 \overset{R}{\bullet} S_2)$;
- $i_2 : G(S_2) \to G(S_1 \overset{R}{\bullet} S_2)$.

Therefore, the consequences in $G(S_i)$ for each $i = 1, 2$ are preserved in the resulting consequence system $G(S_1 \overset{R}{\bullet} S_2)$. Again, note that the extension is not necessarily conservative: we may infer more in the combined system within the language of one of the given consequence systems. Of course, in the case of no interaction the resulting extension is conservative.

4.3 Worked example: LT versus EQ

4.3.1 Models of propositional linear temporal logic

For each signature Π, a Π-model is either the improper model z or a map $m : I\!N \to \wp\Pi$ indicating for each instant n the propositional symbols that hold at n. A proper model m satisfies at n a formula φ, written $m, n \Vdash_\Pi \varphi$, iff:

- true provided that φ is t;

- $\varphi \in m(n)$ provided that $\varphi \in \Pi$;
- $m, n \not\Vdash_{\Pi} \psi$ provided that φ is $(\neg\, \psi)$;
- $m, n \Vdash_{\Pi} \psi'$ or $m, n \not\Vdash_{\Pi} \psi$ provided that φ is $(\psi \Rightarrow \psi')$;
- $m, n+1 \Vdash_{\Pi} \psi$ provided that φ is $(\mathsf{X}\, \psi)$;
- $m, n+k \Vdash_{\Pi} \psi$ for every $k \in I\!N$ provided that φ is $(\mathsf{G}\, \psi)$.

A proper model m satisfies a formula φ, written $m \Vdash_{\Pi} \varphi$, iff $m, n \Vdash_{\Pi} \varphi$ for every $n \in I\!N$. For each signature Π, we establish the satisfaction system

$$sLT_{\Pi} = \langle L_{\Pi}, M_{\Pi}, \Vdash_{\Pi} \rangle$$

where M_{Π} is the class of all Π-models.

4.3.2 Models of unsorted equational logic

For each signature Σ, a Σ-model is a Σ-algebra

$$A = \langle U, \{\mu_n : \Sigma_n \to [U^n \to U]\}_{n \in I\!N} \rangle.$$

The selected model is taken to be an algebra whose domain D is a singleton. Given a variable assignment $\rho : X \to U$, the value of a term $u \in T_{\Sigma}(X)$ given by ρ in A, written $[\![u]\!]_{A,\rho}$, is inductively defined as usual.

An algebra A satisfies an equation $\langle X, u, u' \rangle$ iff $[\![u]\!]_{A,\rho} = [\![u']\!]_{A,\rho}$ for every variable assignment $\rho : X \to U$. For each signature Σ, we establish the satisfaction system

$$sEQ_{\Sigma} = \langle L_{\Sigma}, M_{\Sigma}, \Vdash_{\Sigma} \rangle$$

where M_{Σ} is the class of all Σ-algebras.

4.3.3 Synchronization on models

Assume that we are given the satisfaction systems sLT_{Π} and sEQ_{Σ} with the proviso that $\{a_u : u \in T_{\Sigma}(\emptyset)\} \subseteq \Pi$. The practical motivation is to combine sLT_{Π} with sEQ_{Σ} in order to specify the behavior of an object with the time-dependent attribute a: as before, the formula a_u is intended to assert that the value of a is u.

We choose the model-synchronizing relation to be as follows:

- $\langle m, A \rangle \in R$ iff $m \Vdash_{\Pi} (a_u \Leftrightarrow a_{u'})$ whenever $A \Vdash_{\Sigma} \langle \emptyset, u, u' \rangle$.

That is, we only accept pairs $\langle m, A \rangle$ such that if the equational model A satisfies a ground equation then the temporal model m satisfies the corresponding equivalence on the attribute values.

It is worthwhile to look at an example to see how the resulting semantic entailment reflects the intended interaction. It is easy to verify that:

$$a_{h(c)} \in \{a_{g(c)}, g(x) = h(x)\}^{\vDash}.$$

Indeed, if $\langle m, A \rangle \Vdash \{a_{g(c)}, g(x) = h(x)\}$ then we have:

- $A \Vdash_\Sigma g(x) = h(x)$ by definition of $\Vdash$;
- $A \Vdash_\Sigma g(c) = h(c)$ by soundness of substitution in EQ;
- $m \Vdash_\Pi (a_{g(c)} \Leftrightarrow a_{h(c)})$ by definition of R;
- $m \Vdash_\Pi a_{g(c)}$ by definition of $\Vdash$;
- $m \Vdash_\Pi a_{h(c)}$ by soundness of propositional reasoning in LT;
- $\langle m, A\rangle \Vdash a_{h(c)}$ by definition of $\Vdash$.

4.4 Worked example: LT versus BT

4.4.1 Propositional branching temporal logic

We recall that a propositional branching temporal signature Π is just a set (of propositional symbols). The set of Π-formulae is inductively defined as follows:

$$L_\Pi = t \mid \Pi \mid (\neg\, L_\Pi) \mid (L_\Pi \Rightarrow L_\Pi) \mid (\forall\, \mathsf{X}\, L_\Pi) \mid (\forall\, \mathsf{G}\, L_\Pi) \mid (\exists\, \mathsf{G}\, L_\Pi).$$

For the sake of simplicity we do not use the quantified "until" operators, thus restricting to the branching logic proposed in [3].

Once again, some abbreviations are useful. Namely:

- $(\forall\, \mathsf{F}\, \varphi) \stackrel{abv}{\equiv} (\neg(\exists\, \mathsf{G}(\neg\, \varphi)))$;
- $(\exists\, \mathsf{F}\, \varphi) \stackrel{abv}{\equiv} (\neg(\forall\, \mathsf{G}(\neg\, \varphi)))$.

For each signature Π, a Π-model is a triple $M = \langle W, S, V\rangle$ where:

- W is a set (of worlds);
- $S \subseteq W^2$ is a binary (successor) relation s.t. for each $w \in W$ exists $w' \in W$ with $\langle w, w'\rangle \in S$;
- $V : W \to \wp\Pi$ is a map (valuation of propositional symbols on worlds).

We also recall that a w-branch over M is a sequence $\beta : I\!N \to W$ such that $\beta(0) = w$ and $\langle \beta(i), \beta(i+1)\rangle \in S$ for each $i \in I\!N$.

A model M satisfies at $w \in W$ a formula φ, written $M, w \Vdash_\Pi \varphi$, iff:

- true provided that φ is t;
- $\varphi \in V(w)$ provided that $\varphi \in \Pi$;
- $M, w \nVdash_\Pi \psi$ provided that φ is $(\neg\, \psi)$;
- $M, w \Vdash_\Pi \psi'$ or $M, w \nVdash_\Pi \psi$ provided that φ is $(\psi \Rightarrow \psi')$;
- $M, \beta(1) \Vdash_\Pi \psi$ for every w-branch β provided that φ is $(\forall\, \mathsf{X}\, \psi)$;
- $M, \beta(i) \Vdash_\Pi \psi$ for every $i \in I\!N$ and every w-branch β provided that φ is $(\forall\, \mathsf{G}\, \psi)$;

- $M, \beta(i) \Vdash_\Pi \psi$ for every $i \in I\!N$ and some w-branch β provided that φ is $(\exists\, \mathsf{G}\, \psi)$.

A model M satisfies a formula φ, written $M \Vdash_\Pi \varphi$, iff $M, w \Vdash_\Pi \varphi$ for every $w \in W$. For each signature Π, we establish the satisfaction system

$$sBT_\Pi = \langle L_\Pi, M_\Pi, \Vdash_\Pi \rangle$$

where M_Π is the class of all Π-models. Clearly, the selected improper model corresponds to the case where the set of worlds is empty i.e., $\langle \emptyset, \emptyset, V_\emptyset \rangle$ with $V_\emptyset$ the unique map from $\emptyset$ to $\wp\Pi$.

4.4.2 Synchronization on models

Assume that we are given the satisfaction systems sLT_{Π_1} and sBT_{Π_2} with the proviso that $\Pi_1 \subseteq \Pi_2$. The practical motivation is to combine sLT_{Π_1} with sBT_{Π_2} in such a way that linear properties of one branch over a branching model can be related to its branching properties.

We therefore choose the model-synchronizing relation to be as follows:

- $\langle m, M \rangle \in R$ iff either m is the improper linear model or there exists a branch β over M such that $m = \lambda\, i.V(\beta(i)) \cap \Pi_1$.

That is, we only accept pairs $\langle m, M \rangle$ such that if the linear model m is proper then it must correspond to a branch over M.

Let us illustrate by means of an example the kind of interaction achieved by such a synchronization. It is easy to verify that given $p \in \Pi_1$:

$$(\mathsf{G}(\mathsf{F}\, p)) \in \{(\forall\, \mathsf{F}\, p)\}^{\vDash}.$$

In fact, if $\langle m, M \rangle \Vdash \{(\forall\, \mathsf{F}\, p)\}$ then we have:

- $M \Vdash_{\Pi_2} (\forall\, \mathsf{F}\, p)$ by definition of $\Vdash$;
- $M, \beta(i) \Vdash_{\Pi_2} (\forall\, \mathsf{F}\, p)$ for each $i \in I\!N$ by definition of $\Vdash_{\Pi_2}$;
- $M, \beta(j) \Vdash_{\Pi_2} p$ for some $j \geq i$ and each $i \in I\!N$ by definition of $\Vdash_{\Pi_2}$ at $\beta(i)$;
- $p \in V(\beta(j))$ for some $j \geq i$ and each $i \in I\!N$ by definition of $\Vdash_{\Pi_2}$ at $\beta(j)$;
- $p \in m(j)$ for some $j \geq i$ and each $i \in I\!N$ by definition of R;
- $m, j \Vdash_{\Pi_1} p$ for some $j \geq i$ and each $i \in I\!N$ by definition of $\Vdash_{\Pi_1}$ at j;
- $m, i \Vdash_{\Pi_1} (\mathsf{F}\, p)$ for each $i \in I\!N$ by definition of $\Vdash_{\Pi_1}$ at i;
- $m \Vdash_{\Pi_1} (\mathsf{G}(\mathsf{F}\, p))$ by definition of $\Vdash_{\Pi_1}$;
- $\langle m, M \rangle \Vdash (\mathsf{G}(\mathsf{F}\, p))$ by definition of $\Vdash$

provided that m is proper with corresponding branch β. If not the result is trivial.

Of course only branching to linear extrapolation of properties was possible with this kind of interaction. Let us see how we can go in the opposite direction too. Assume the existence of a propositional symbol $\delta \in \Pi_2$ such that $\{w \in W : \delta \in V(w)\} = \{w_0\}$.

Consider then the model-synchronizing relation:

- $\langle m, M\rangle \in R^\delta$ iff either both m and M are improper or there exists a w_0-branch β over M such that $m = \lambda\, i.V(\beta(i)) \cap \Pi_1$.

That is, we only accept pairs $\langle m, M\rangle$ where both are improper or m corresponds to a branch over M, its initial state being uniquely determined by the propositional symbol δ.

It is now possible to check that given $p, q \in \Pi_1$:

$$(\delta \Rightarrow (\exists\, \mathsf{F}\, p)) \in \{(\mathsf{G}\, q), (\forall\, \mathsf{F}(q \Rightarrow p))\}^{\models}.$$

In fact, if $\langle m, M\rangle \Vdash \{(\mathsf{G}\, q), (\forall\, \mathsf{F}(q \Rightarrow p))\}$ then we have:

- $M \Vdash_{\Pi_2} (\forall\, \mathsf{F}(q \Rightarrow p))$ by definition of $\Vdash$;
- $M, \beta(0) \Vdash_{\Pi_2} (\forall\, \mathsf{F}(q \Rightarrow p))$ by definition of $\Vdash_{\Pi_2}$;
- $M, \beta(j) \Vdash_{\Pi_2} (q \Rightarrow p)$ for some $j \in I\!N$ by definition of $\Vdash_{\Pi_2}$ at $\beta(0)$;
- similarly, $m \Vdash_{\Pi_1} (\mathsf{G}\, q)$ by definition of $\Vdash$;
- $m, i \Vdash_{\Pi_1} q$ for each $i \in I\!N$ by definition of $\Vdash_{\Pi_1}$;
- $q \in m(i)$ for each $i \in I\!N$ by definition of $\Vdash_{\Pi_1}$ at i;
- $q \in V(\beta(i))$ for each $i \in I\!N$ by definition of R;
- $M, \beta(i) \Vdash_{\Pi_2} q$ for each $i \in I\!N$ by definition of $\Vdash_{\Pi_2}$ at $\beta(i)$;
- $M, \beta(j) \Vdash_{\Pi_2} p$ for some $j \in I\!N$ by definition of $\Vdash_{\Pi_2}$ at $\beta(j)$;
- $M, \beta(0) \Vdash_{\Pi_2} (\exists\, \mathsf{F}\, p)$ by definition of $\Vdash_{\Pi_2}$ at $\beta(0)$;
- $M \Vdash_{\Pi_2} (\delta \Rightarrow (\exists\, \mathsf{F}\, p))$ by definition of $\Vdash_{\Pi_2}$ and the hypothesis on δ;
- $\langle m, M\rangle \Vdash (\delta \Rightarrow (\exists\, \mathsf{F}\, p))$ by definition of $\Vdash$

provided that m is proper with corresponding w_0-branch β of M. Otherwise both are improper and the result is immediate.

Other model-synchronization relations could have been considered. For instance, restricting to acyclic or even tree-like branching models and considering the existence of a positional symbol distinguishing exactly the worlds visited by the branch considered would enable the expression of inevitability properties over the selected branch of the branching model, thus rendering corresponding linear fairness properties. A similar technique is used, for instance, in [20]. We refrain to present the details here since a full study of the possible synchronizations for this or any other particular case is out of the scope of this paper.

4.5 Categorial characterization

We already pointed out that the pairs $\langle i_1, P_1|_R\rangle$ and $\langle i_2, P_2|_R\rangle$ are morphisms in *Sat*. We can further characterize the nature of $S_1 \overset{R}{\bullet} S_2$ within the category *Sat*. We first consider the case when the model-synchronizing relation introduces no interaction, that is when $R = M_1 \times M_2$.

Proposition 4.4 The system $S_1 \overset{M_1\times M_2}{\bullet} S_2$ is the coproduct $S_1 + S_2$ of S_1 and S_2 with the injections $\langle i_1, P_1\rangle$ and $\langle i_2, P_2\rangle$.
Proof:
1. $\langle i_1, P_1\rangle$ and $\langle i_2, P_2\rangle$ are morphisms in *Sat*:
Clearly, $\langle i_j, P_j\rangle$ is such that $P_j = P_j|_{M_1\times M_2}$ for $j = 1, 2$. Therefore, both are morphisms according to 4.2.
2. Universal property:
Let $h_1 : S_1 \to S'$ and $h_2 : S_2 \to S'$ be any satisfaction system morphisms. Consider then $h : S_1 \overset{M_1\times M_2}{\bullet} S_2 \to S'$ such that $(\uparrow h)(i_1(\varphi_1)) = (\uparrow h_1)(\varphi_1)$, $(\uparrow h)(i_2(\varphi_2)) = (\uparrow h_2)(\varphi_2)$ and $(\downarrow h)(m') = \langle(\downarrow h_1)(m'), (\downarrow h_2)(m')\rangle$. Of course h is a satisfaction system morphism and it is the unique such that $h \circ \langle i_1, P_1\rangle = h_1$ and $h \circ \langle i_2, P_2\rangle = h_2$. QED

The diagram in Figure 5 depicts the universal property of this coproduct with respect to any model-synchronization of S_1 and S_2. The morphism in $Set_* \times Cls_*^{op}$

$$(\langle i_1, P_1|_R\rangle + \langle i_2, P_2|_R\rangle) : \langle L_1 + L_2, M_1 \times M_2\rangle \to \langle L_1 + L_2, R\rangle$$

is the pair $\langle id_{L_1+L_2}, inc_R\rangle$ where inc_R is the inclusion map.

Proposition 4.5 For every $R \subseteq M_1 \times M_2$ and every satisfaction system S' such that there are morphisms $h_1 : S_1 \to S'$ and $h_2 : S_2 \to S'$, there is at most one morphism k from $S_1 \overset{R}{\bullet} S_2$ in *Sat* to S' such that:

- $h_1 = k \circ \langle i_1, P_1|_R\rangle$;
- $h_2 = k \circ \langle i_2, P_2|_R\rangle$.

Proof:
Straightforward consequence of the universal property of the coproduct and the fact that $(\langle i_1, P_1|_R\rangle + \langle i_2, P_2|_R\rangle)$ is an epimorphism: epi on formulae and mono on models. QED

It remains to see if the satisfaction system $S_1 \overset{R}{\bullet} S_2$ can be obtained from S_1 and S_2 by a categorial construction in the general case (for any model-synchronizing relation) in a way similar to the one described in the previous section for the case of formula-synchronization.

$$\begin{tikzcd}
S_1 \arrow[dr, "\langle i_1, P_1\rangle"] \arrow[ddr, bend right, "\langle i_1, P_1|_R\rangle"'] & & S_2 \arrow[dl, "\langle i_2, P_2\rangle"'] \arrow[ddl, bend left, "\langle i_2, P_2|_R\rangle"] \\
& S_1 + S_2 \arrow[d, dotted, "\langle id_{L_1+L_2}, inc_R\rangle"] & \\
& S_1 \overset{R}{\bullet} S_2 &
\end{tikzcd}$$

Figure 5: $\langle id_{L_1+L_2}, inc_R\rangle$ is the unique morphism such that $\langle id_{L_1+L_2}, inc_R\rangle \circ \langle i_1, P_1\rangle = \langle i_1, P_1|_R\rangle$ and $\langle id_{L_1+L_2}, inc_R\rangle \circ \langle i_2, P_2\rangle = \langle i_2, P_2|_R\rangle$.

Prop/Definition 4.6 The maps

- $N(\langle L, M, \Vdash\rangle) = \langle L, M\rangle$;
- $N(h : \langle L, M, \Vdash\rangle \to \langle L', M', \Vdash'\rangle) = h$

constitute the functor $N : Sat \to Set_* \times Cls_*^{op}$.

Proposition 4.7 For each $\langle L, M, \Vdash\rangle$ in Sat and each $h : \langle L, M\rangle \to \langle L', M'\rangle$ in $Set_* \times Cls_*^{op}$ such that $h\!\uparrow$ is surjective and $h\!\uparrow\!(\varphi_1) = h\!\uparrow\!(\varphi_2)$ implies $h\!\downarrow\!(m') \Vdash \varphi_1$ iff $h\!\downarrow\!(m') \Vdash \varphi_2$ for each $m' \in M'$, the morphism $h : \langle L, M, \Vdash\rangle \to \langle L', M', \Vdash'\rangle$ is cocartesian by N for h on $\langle L, M, \Vdash\rangle$ where $m' \Vdash' h\!\uparrow\!(\varphi)$ iff $h\!\downarrow\!(m') \Vdash \varphi$.

Proof:
1. $\langle L', M', \Vdash'\rangle$ is a satisfaction system:
Clearly, $\Vdash'$ is well defined. The other conditions are straightforward from the fact that h is a morphism in $Set_* \times Cls_*^{op}$.
2. h is a morphism in Sat:
Trivial, by definition of $\Vdash'$.
3. Universal property:
Assume that $g : \langle L, M, \Vdash\rangle \to \langle L'', M'', \Vdash''\rangle$ is any morphism in Sat and that $f : \langle L', M'\rangle \to \langle L'', M''\rangle$ is any morphism in $Set_* \times Cls_*^{op}$ such that $f \circ h = N(g)$. Then, f is the unique morphism in Sat such that $f \circ h = g$. We just prove that f is a morphism in Sat.
For each $\varphi' = h\!\uparrow\!(\varphi)$ and m'' we have: $m'' \Vdash'' f\!\uparrow\!(\varphi')$ iff $m'' \Vdash'' f\!\uparrow\!(h\!\uparrow\!(\varphi))$ iff $m'' \Vdash'' g\!\uparrow\!(\varphi)$ iff $g\!\downarrow\!(m'') \Vdash \varphi$ iff $h\!\downarrow\!(f\!\downarrow\!(m'')) \Vdash \varphi$ iff $f\!\downarrow\!(m'') \Vdash' h\!\uparrow\!(\varphi)$ iff $f\!\downarrow\!(m'') \Vdash' \varphi'$. QED

We denote the codomain of the cocartesian morphism above by $h(\langle L, M, \Vdash\rangle)$.

Proposition 4.8 $S_1 \overset{R}{\bullet} S_2 = \langle id_{L_1+L_2}, inc_R\rangle(S_1 + S_2)$.

Proof:
Obviously 4.7 applies since $id_{L_1+L_2}$ is a bijection. And clearly, the satisfaction

relation $\Vdash'$ of $\langle id_{L_1+L_2}, inc_R\rangle(S_1 + S_2)$ is such that, for each $m \in R$ and $\varphi \in L_1 + L_2$, $m \Vdash' \varphi$ iff $m \Vdash \varphi$ where $\Vdash$ is the satisfaction relation of $S_1 + S_2$. Therefore, for $j = 1, 2$, $m \Vdash' i_j(\varphi_j)$ iff $m_j \Vdash_j \varphi_j$ and thus the desired equality $S_1 \overset{R}{\bullet} S_2 = \langle id_{L_1+L_2}, inc_R\rangle(S_1 + S_2)$ holds. QED

5 Synchronizing logics

For the sake of simplicity, herein we only consider logics on Earth and their synchronization on formulae. But we do not expect any trouble in adapting the ideas to logics in Hell and their synchronization on models.

5.1 Category of logics

We define a logic to be the indexed class of its instances (consequence systems), where indexes take values over some base class.

Definition 5.1 A *logic* is a map $\mathcal{L} : B \to |Csy|$ where B is a class (the base of $\mathcal{L}$).

Definition 5.2 A *logic morphism* is a pair $T = \langle k, \alpha\rangle : \mathcal{L} \to \mathcal{L}'$ where:

- $k : B \to B'$ is a map;
- $\alpha : \{\alpha_b : \mathcal{L}'(k(b)) \to \mathcal{L}(b)\}_{b \in B}$ is a family of morphisms in *Csy*.

We might be tempted to make these definitions functorial. Namely, we might define $\mathcal{L}$ as a functor from a category B to *Csy*, establishing a notion close to the concept of entailment system [8, 19]. Or we might attempt to make k a functor in the definition of the morphism. However, the latter is not feasible if we want to keep the possibility of non conservative combinations and still have a logic morphism from the combined logic to each of the given logics. And the former possibility seems to restrict too much the morphisms we allow between instances (in this case consequence systems) of a logic.

Prop/Definition 5.3 Logics and their morphisms constitute the category *Log*.

5.2 Definition of synchronization of logics on Earth

Definition 5.4 Let:

- $\mathcal{L}_1, \mathcal{L}_2$ be logics;
- $\alpha_1 = \{\alpha_{1b_1b_2}\}_{b_1 \in B_1, b_2 \in B_2}$
  where $\alpha_{1b_1b_2} : L_1 \to L_1 \times L_2$ such that $p_1(\alpha_{1b_1b_2}(\varphi_1)) = \varphi_1$;
- $\alpha_2 = \{\alpha_{2b_1b_2}\}_{b_1 \in B_1, b_2 \in B_2}$
  where $\alpha_{2b_1b_2} : L_2 \to L_1 \times L_2$ such that $p_2(\alpha_{2b_1b_2}(\varphi_2)) = \varphi_2$.

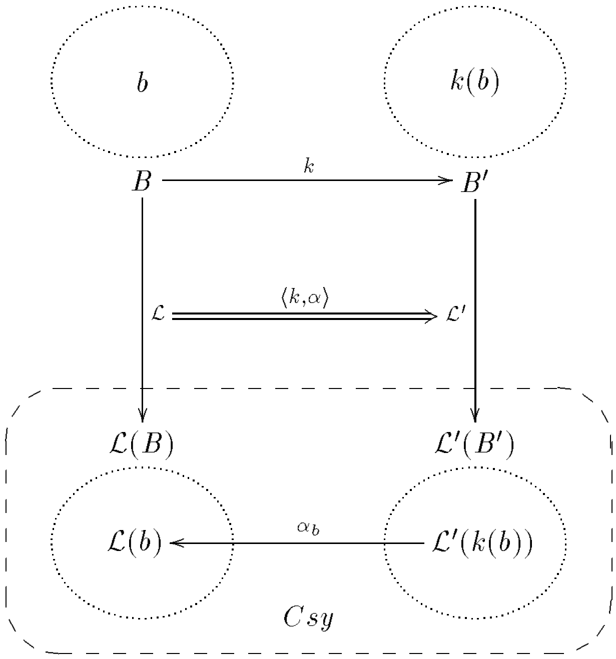

Figure 6: $\langle k, \alpha \rangle : \mathcal{L} \to \mathcal{L}'$ is a logic morphism provided that $k : B \to B'$ is a map and $\alpha_b : \mathcal{L}'(k(b)) \to \mathcal{L}(b)$ is a morphism in Csy for each $b \in B$.

Then, the synchronization on formulae of $\mathcal{L}_1$ and $\mathcal{L}_2$ by the families α_1 and α_2 is the logic

$$\mathcal{L}_1 \overset{\alpha_1 \alpha_2}{\bullet} \mathcal{L}_2 : B_1 \times B_2 \to |Csy|$$

such that:

- $\mathcal{L}_1 \overset{\alpha_1 \alpha_2}{\bullet} \mathcal{L}_2(\langle b_1, b_2 \rangle) = \mathcal{L}_1(b_1) \overset{\alpha_{1b_1b_2} \alpha_{2b_1b_2}}{\bullet} \mathcal{L}_2(b_2)$.

Proposition 5.5 The following pairs are morphisms in *Log*:

- $\langle p_1, \alpha_1 \rangle : \mathcal{L}_1 \overset{\alpha_1 \alpha_2}{\bullet} \mathcal{L}_2 \to \mathcal{L}_1$;
- $\langle p_2, \alpha_2 \rangle : \mathcal{L}_1 \overset{\alpha_1 \alpha_2}{\bullet} \mathcal{L}_2 \to \mathcal{L}_2$.

5.3 Example: linear temporal logic versus equational logic

The logic $\mathcal{L}_{LT}$ maps each signature Π to the consequence system cLT_Π: the base B_{LT} is the class of all propositional linear temporal logic signatures.

The logic $\mathcal{L}_{EQ}$ maps each signature Σ to the consequence system cEQ_Σ: the base B_{EQ} is the class of all unsorted equational logic signatures.

We choose the following formula-synchronization families:

- $\alpha_{1\Pi\Sigma} : L_\Pi \to L_\Pi \times L_\Sigma$ such that $\alpha_{1\Pi\Sigma}(\varphi) = \langle \varphi, x = x \rangle$;
- $\alpha_{2\Pi\Sigma} : L_\Sigma \to L_\Pi \times L_\Sigma$ such that

$$\alpha_{2\Pi\Sigma}(\langle X, u, u' \rangle) = \begin{cases} \langle (a_u \Leftrightarrow a_{u'}), \langle X, u, u' \rangle \rangle & \text{pt} \;\; \{a_u : u \in T_\Sigma(\emptyset)\} \subseteq \Pi \\ & \qquad \text{and } X = \emptyset \\ \langle t, \langle X, u, u' \rangle \rangle & \text{otherwise} \end{cases} .$$

6 Concluding remarks

Motivated by software engineering applications, we developed two alternative techniques for combining logic systems: synchronization on formulae (of consequence systems) and synchronization on models (of satisfaction systems). We provided categorial characterizations of both constructions using cocartesian liftings. In the former case, it was necessary to carry out the categorial construction in the category of one-step derivation systems. We illustrated the two techniques with simple but sufficiently meaningful examples. Finally, we outlined the technique for synchronizing logics on top of the technique for synchronizing consequence systems.

The adopted categorial approach was found to be useful namely in order to assess the universality of the constructions. For instance, we might ask if the proposed construction for synchronizing on formulae is the "right one". The fact that we were able to establish a universal property indicates that it is at least "canonical" in some well defined context.

The proposed techniques are very general in the sense that nothing is assumed about the systems or logics to be combined besides that they establish (monotonic) consequence operators.

We are now working on further developments of these two synchronization techniques, namely establishing a relationship between them and obtaining results about the preservation of relevant properties like soundness and completeness.

We are also working on a categorial characterization of fibring. Contrarily to synchronization, fibring allows the mixing of connectives from both logics in a formula [12]. Nevertheless, the categorial approach is also workable in the case of fibring, again providing useful insights. At this stage, the relationship between fibring and synchronization is still not clear.

The combination of theory spaces and finding a suitable notion of "effective" combination of logics in the sense of computability theory are two other research issues of interest.

It seems evident that a variety of combination techniques will be necessary for dealing with the multiformalism problems that appear in software engineering. Further experimental work is needed on the practical application of all combination techniques already identified in the literature before methodological principles can be found. Meanwhile, we expect that other useful forms of combination will arise besides synchronization and fibring.

Acknowledgments

We are grateful to our colleagues in the ACL initiative for many discussions on the role of categorial techniques in computing. We are also grateful to our colleagues in the FLIRTS group, namely Maura Cerioli, José Fiadeiro, Till Mossakowski, Wiesiek Pawlowski, Gianna Reggio, Antonino Salibra, Giuseppe Scollo, Andrzej Tarlecki, Uwe Wolter and Elena Zucca with whom we discussed the idea of combining logics within the institutional setting. And we would like to express our gratitude to Mark Ryan and Pierre-Yves Schobbens who gave us feedback in the early stages of this research. This work was partially supported by PRAXIS XXI Project 2/2.1/MAT/262/94 SitCalc and ESPRIT IV Working Group 22704 ASPIRE.

References

[1] M. Arrais and J. Fiadeiro. Unifying theories in different institutions. In M. Haveraaen, O. Owe, and O.-J. Dahl, editors, *Recent Trends in Data Type Specification*, pages 81–101. Springer-Verlag, LNCS 1130, 1996.

[2] J. Barwise. Axioms for abstract model theory. *Annals of Mathematical Logic*, 7:221–265, 1974.

[3] M. Ben-Ari, Z. Manna, and A. Pnueli. The temporal logic of branching time. In *8th ACM Symposium on Principles of Programming Languages*, pages 164–176. Williamsburg, VA, 1981. Also, *Acta Informatica*, 20(3):207-226, 1983.

[4] P. Blackburn and M. de Rijke. Why combine logics. *Studia Logica*, 1996. In print.

[5] A. Blass. The interaction between category theory and set theory. In J. W. Gray, editor, *Mathematical Applications of Category Theory*, pages 5–29. AMS Series in Contemporary Mathematics, 1984.

[6] R. Burstall and J. Goguen. Putting theories together to make specifications. In R. Reddy, editor, *Proc. 5th Intl. Joint Conf. on Artificial Intelligence*, pages 1045–1058, Cambridge, Massachusetts, 1977.

[7] M. Cerioli and J. Meseguer. May I borrow your logic. In A. Borzyszkowki and S. Sokolowski, editors, *Mathematical Foundations of Computer Science 1993*, pages 342–351. LNCS, Springer-Verlag, 1993.

[8] J. Fiadeiro and A. Sernadas. Structuring theories on consequence. In D. Sannella and A. Tarlecki, editors, *Recent Trends in Data Type Specification*, pages 44–72. Springer-Verlag, 1988.

[9] M. Finger and D. Gabbay. Adding a temporal dimension to a logic system. *Journal of Logic, Language and Information*, 1:203–233, 1992.

[10] M. Finger and D. Gabbay. Combining temporal logic systems. *Notre Dame Journal of Formal Logic*, 37:204–232, 1996.

[11] D. Gabbay, editor. *What is a logical system?* Oxford Science Publications, 1994.

[12] D. Gabbay. An overview of fibred semantics and the combination of logics. In F. Baader and K. Schulz, editors, *Frontiers of Combining Systems*, pages 1–55. Kluwer Academic Publishers, 1996.

[13] J. Goguen and R. Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the ACM*, 39(1):95–146, 1992.

[14] R. Goldblatt. *Logics of Time and Computation.* CSLI, 1992. Second edition.

[15] M. Kracht and F. Wolter. Properties of independently axiomatizable bimodal logics. *Journal of Symbolic Logic*, 56(4):1469–1485, 1991.

[16] S. Mac Lane. *Categories for the Working Mathematician.* Springer-Verlag, 1971.

[17] J.-L. Lassez, V. L. Nguyen, and E. A. Sonenberg. Fixed point theorems and semantics: A folk tale. *Information Processing Letters*, 14(3):112–116, 1982.

[18] M. Manzano. *Extensions of First Order Logic.* Cambridge University Press, 1996.

[19] J. Meseguer. General logics. In H.-D. Ebbinghaus et al, editor, *Proceedings of the Logic Colloquium, 1987*, pages 275–329. North-Holland, 1989.

[20] W. Penczek. A temporal logic for event structures. *Fundamenta Informaticae*, 11(3):297–326, 1988.

[21] A. Tarlecki. Moving between logical systems. In M. Haveraaen, O. Owe, and O.-J. Dahl, editors, *Recent Trends in Data Type Specification*, pages 478–502. Springer-Verlag, LNCS 1130, 1996.

[22] A. Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5:285–309, 1955.

[23] A. Tarski. On the concept of logical consequence. In J. H. Woodger, editor, *Logic, Semantics, Metamathematics*, pages 409–420. Oxford Clarendon Press, 1956.

[24] R. Thomason. Combinations of tense and modality. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic II*, pages 135–165. Kluwer Academic Publishers, 1984.

[25] G. Winskel and M. Nielsen. Models of concurrency. In D. Gabbay S. Abramsky and T. Maibaum, editors, *Handbook of Logic in Computer Science 4*, pages 1–148. Oxford Science Publications, 1995.