Satisfiability Calculus: The Semantic Counterpart of a Proof Calculus in General Logics

Carlos Gustavo López Pombo, Pablo F. Castro, Nazareno M. Aguirre, Thomas E. Maibaum

Satisfiability Calculus: The Semantic Counterpart of a Proof Calculus in General Logics

Carlos G. López Pombo[1,3], Pablo F. Castro[2,3], Nazareno M. Aguirre[2,3], and Thomas S.E. Maibaum[4]

[1] Departmento de Computación, FCEyN, Universidad de Buenos Aires, Argentina. clpombo@dc.uba.ar
[2] Departmento de Computación, FCEFQyN, Universidad Nacional de Río Cuarto, Argentina. {pcastro,naguirre}@dc.exa.unrc.edu.ar
[3] Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina
[4] Department of Computing & Software, McMaster University, Canada. tom@maibaum.org

Abstract. Since its introduction by Goguen and Burstall in 1984, the theory of institutions has been one of the most widely accepted formalizations of abstract model theory. This work was extended by a number of researchers, José Meseguer among them, who presented *General Logics*, an abstract framework that complements the model theoretical view of institutions by defining the categorical structures that provide a proof theory for any given logic. In this paper we intend to complete this picture by providing the notion of *Satisfiability Calculus*, which might be thought of as the semantical counterpart of the notion of proof calculus, that provides the formal foundations for those proof systems that use model construction techniques to prove or disprove a given formula, thus "implementing" the satisfiability relation of an institution.

1 Introduction

The theory of institutions, presented by Goguen and Burstall in [1], provides a formal and generic definition of what a logical system is, from a model theoretical point of view. This work evolved in many directions: in [2], Meseguer complemented the theory of institutions by providing a categorical characterization for the notions of entailment system (also called π-institutions by other authors in [3]) and the corresponding notion of proof calculi; in [4, 5] Goguen and Burstall, and Tarlecki, respectively, extensively investigated the ways in which institutions can be related; in [6], Sannella and Tarlecki studied how specifications in an arbitrary logical system can be structured; in [7], Tarlecki presented an abstract theory of software specification and development; in [8, 9] and [10, 11], Mossakowski and Tarlecki, and Diaconescu, respectively, proposed the use of institutions as a foundation for heterogeneous environments for software specification. Institutions have also been used as a very general version of abstract model theory [12], offering a suitable formal framework for addressing heterogeneity in specifications [13, 14], including applications to UML [15] and other languages related to computer science and software engineering.

Extensions of institutions to capture proof theoretical concepts have been extensively studied, most notably by Meseguer [2]. Essentially, Meseguer proposes the extension of entailment systems with a categorical concept expressive enough to capture the notion of *proof* in an abstract way. In Meseguer's words:

> *A reasonable objection to the above definition of logic*[5] *is that it abstracts away the structure of proofs, since we know only that a set Γ of sentences entails another sentence φ, but no information is given about the internal structure of such a $\Gamma \vdash \varphi$ entailment. This observation, while entirely correct, may be a virtue rather than a defect, because the entailment relation is precisely what remains* <u>*invariant*</u> *under many equivalent proof calculi that can be used for a logic.*

Before Meseguer's work, there was an imbalance in the definition of a logic in the context of institution theory, since the deductive aspects of a logic were not taken into account. Meseguer concentrates on the proof theoretical aspects of a logic, providing not only the definition of entailment system, but also complementing it with the notion of proof calculus, obtaining what he calls a *logical system*. As introduced by Meseguer, the notion of proof calculus provides, intuitively, an implementation of the entailment relation of a logic. Indeed, Meseguer corrected the inherent imbalance in favour of models in institutions, enhancing syntactic aspects in the definition of logical systems.

However, the same lack of an operational view observed in the definition of entailment systems still appears with respect to the notion of satisfiability, i.e., the satisfaction relation of an institution. In the same way that an entailment system may be "implemented" in terms of different proof calculi, a satisfaction relation may be "implemented" in terms of different satisfiability procedures. Making these satisfiability procedures explicit in the characterization of logical systems is highly relevant, since many successful software analysis tools are based on particular characteristics of these satisfiability procedures. For instance, many automated analysis tools rely on model construction, either for proving properties, as with model-checkers, or for finding counterexamples, as with tableaux techniques or SAT-solving based tools. These techniques constitute an important stream of research in logic, in particular in relation to (semi-)automated software validation and verification.

These kinds of logical systems can be traced back to the works of Beth [16, 17], Herbrand [18] and Gentzen [19]. Beth's ideas were used by Smullyan to formulate the tableaux method for first-order predicate logic [20]. Herbrand's and Gentzen's works inspired the formulation of resolution systems presented by Robinson [21]. Methods like those based on resolution and tableaux are strongly

[5] **Authors' note**: Meseguer refers to a logic as a structure that is composed of an entailment system together with an institution, see Def. 6.

related to the semantics of a logic; one can often use them to guide the construction of models. This is not possible in *pure* deductive methods, such as natural deduction or Hilbert systems, as formalized by Meseguer. In this paper, our goal is to provide an abstract characterization of this class of semantics based tools for logical systems. This is accomplished by introducing a categorical characterization of the notion of satisfiability calculus which embraces logical tools such as tableaux, resolution, Gentzen style sequents, etc. As we mentioned above, this can be thought of as a formalization of a semantic counterpart of Meseguer's proof calculus. We also explore the concept of mappings between satisfiability calculi and the relation between proof calculi and satisfiability calculi.

The paper is organized as follows. In Section 2 we present the definitions and results we will use throughout this paper. In Section 3 we present a categorical formalization of satisfiability calculus, and prove relevant results underpinning the definitions. We also present examples to illustrate the main ideas. Finally in Section 4 we draw some conclusions and describe further lines of research.

2 Preliminaries

From now on, we assume the reader has a nodding acquaintance with basic concepts from category theory [22, 23]. Below we present the basic definitions and results we use throughout the rest of the paper. In the following, we follow the notation introduced in [2].

An *Institution* is an abstract formalization of the model theory of a logic by making use of the relationships existing between signatures, sentences and models. These aspects are reflected by introducing the category of signatures, and by defining functors going from this category to the categories Set and Cat, to capture sets of sentences and categories of models, respectively, for a given signature. The original definition of institutions is the following:

Definition 1. ([1]) *An* institution *is a structure of the form* $\langle \mathsf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \{\models^{\Sigma}\}_{\Sigma \in |\mathsf{Sign}|}\rangle$ *satisfying the following conditions:*

- Sign *is a category of signatures,*
- $\mathbf{Sen} : \mathsf{Sign} \to \mathsf{Set}$ *is a functor. Let* $\Sigma \in |\mathsf{Sign}|$*, then* $\mathbf{Sen}(\Sigma)$ *returns the set of* Σ*-sentences,*
- $\mathbf{Mod} : \mathsf{Sign}^{\mathsf{op}} \to \mathsf{Cat}$ *is a functor. Let* $\Sigma \in |\mathsf{Sign}|$*, then* $\mathbf{Mod}(\Sigma)$ *returns the category of* Σ*-models,*
- $\{\models^{\Sigma}\}_{\Sigma \in |\mathsf{Sign}|}$*, where* $\models^{\Sigma} \subseteq |\mathbf{Mod}(\Sigma)| \times \mathbf{Sen}(\Sigma)$*, is a family of binary relations,*

and for any signature morphism $\sigma : \Sigma \to \Sigma'$*,* Σ*-sentence* $\phi \in \mathbf{Sen}(\Sigma)$ *and* Σ'*-model* $\mathcal{M}' \in |\mathbf{Mod}(\Sigma)|$*, the following* $\models$*-invariance condition holds:*

$$\mathcal{M}' \models^{\Sigma'} \mathbf{Sen}(\sigma)(\phi) \quad \textit{iff} \quad \mathbf{Mod}(\sigma^{\mathsf{op}})(\mathcal{M}') \models^{\Sigma} \phi \ .$$

Roughly speaking, the last condition above says that *the notion of truth is invariant with respect to notation change.* Given $\Sigma \in |\mathsf{Sign}|$ and $\Gamma \subseteq \mathbf{Sen}(\Sigma)$,

$\mathbf{Mod}(\Sigma,\Gamma)$ denotes the full subcategory of $\mathbf{Mod}(\Sigma)$ determined by those models $\mathcal{M} \in |\mathbf{Mod}(\Sigma)|$ such that $\mathcal{M} \models^{\Sigma} \gamma$, for all $\gamma \in \Gamma$. The relation $\models^{\Sigma}$ between sets of formulae and formulae is defined in the following way: given $\Sigma \in |\mathsf{Sign}|$, $\Gamma \subseteq \mathbf{Sen}(\Sigma)$ and $\alpha \in \mathbf{Sen}(\Sigma)$, $\Gamma \models^{\Sigma} \alpha$ if and only if $\mathcal{M} \models^{\Sigma} \alpha$, for all $\mathcal{M} \in |\mathbf{Mod}(\Sigma,\Gamma)|$.

An *entailment system* is defined in a similar way, by identifying a family of *syntactic* consequence relations, instead of a family of semantic consequence relations. Each of the elements in this family is associated with a signature. These relations are required to satisfy reflexivity, monotonicity and transitivity. In addition, a notion of translation between signatures is considered.

Definition 2. ([2]) *An* entailment system *is a structure of the form* $\langle \mathsf{Sign}, \mathbf{Sen}, \{\vdash^{\Sigma}\}_{\Sigma \in |\mathsf{Sign}|}\rangle$ *satisfying the following conditions:*

- Sign *is a category of signatures,*
- $\mathbf{Sen} : \mathsf{Sign} \to \mathsf{Set}$ *is a functor. Let* $\Sigma \in |\mathsf{Sign}|$*; then* $\mathbf{Sen}(\Sigma)$ *returns the set of* Σ*-sentences, and*
- $\{\vdash^{\Sigma}\}_{\Sigma \in |\mathsf{Sign}|}$, *where* $\vdash^{\Sigma} \subseteq 2^{\mathbf{Sen}(\Sigma)} \times \mathbf{Sen}(\Sigma)$, *is a family of binary relations such that for any* $\Sigma, \Sigma' \in |\mathsf{Sign}|$, $\{\phi\} \cup \{\phi_i\}_{i \in \mathcal{I}} \subseteq \mathbf{Sen}(\Sigma)$, $\Gamma, \Gamma' \subseteq \mathbf{Sen}(\Sigma)$, *the following conditions are satisfied:*
  1. *reflexivity:* $\{\phi\} \vdash^{\Sigma} \phi$,
  2. *monotonicity: if* $\Gamma \vdash^{\Sigma} \phi$ *and* $\Gamma \subseteq \Gamma'$, *then* $\Gamma' \vdash^{\Sigma} \phi$,
  3. *transitivity: if* $\Gamma \vdash^{\Sigma} \phi_i$ *for all* $i \in \mathcal{I}$ *and* $\{\phi_i\}_{i \in \mathcal{I}} \vdash^{\Sigma} \phi$, *then* $\Gamma \vdash^{\Sigma} \phi$, *and*
  4. $\vdash$*-translation: if* $\Gamma \vdash^{\Sigma} \phi$, *then for any morphism* $\sigma : \Sigma \to \Sigma'$ *in* Sign, $\mathbf{Sen}(\sigma)(\Gamma) \vdash^{\Sigma'} \mathbf{Sen}(\sigma)(\phi)$.

Definition 3. ([2]) *Let* $\langle \mathsf{Sign}, \mathbf{Sen}, \{\vdash^{\Sigma}\}_{\Sigma \in |\mathsf{Sign}|}\rangle$ *be an entailment system. Its category* Th *of theories is a pair* $\langle \mathcal{O}, \mathcal{A}\rangle$ *such that:*

- $\mathcal{O} = \{\, \langle \Sigma, \Gamma\rangle \mid \Sigma \in |\mathsf{Sign}| \text{ and } \Gamma \subseteq \mathbf{Sen}(\Sigma) \,\}$, *and*
- $\mathcal{A} = \left\{ \sigma : \langle \Sigma,\Gamma\rangle \to \langle \Sigma',\Gamma'\rangle \;\middle|\; \begin{array}{l} \langle \Sigma,\Gamma\rangle, \langle \Sigma',\Gamma'\rangle \in \mathcal{O}, \\ \sigma : \Sigma \to \Sigma' \text{ is a morphism in } \mathsf{Sign} \text{ and} \\ \text{for all } \gamma \in \Gamma, \Gamma' \vdash^{\Sigma'} \mathbf{Sen}(\sigma)(\gamma) \end{array} \right\}$.

In addition, if a morphism $\sigma : \langle \Sigma,\Gamma\rangle \to \langle \Sigma',\Gamma'\rangle$ satisfies $\mathbf{Sen}(\sigma)(\Gamma) \subseteq \Gamma'$, it is called *axiom preserving*. By retaining those morphisms of Th that are axiom preserving, we obtain the subcategory Th_0. If we now consider the definition of $\mathbf{Mod}$ extended to signatures and sets of sentences, we get a functor $\mathbf{Mod} : \mathsf{Th}^{\mathsf{op}} \to \mathsf{Cat}$ defined as follows: let $T = \langle \Sigma,\Gamma\rangle \in |\mathsf{Th}|$, then $\mathbf{Mod}(T) = \mathbf{Mod}(\Sigma,\Gamma)$.

Definition 4. ([2]) *Let* $\langle \mathsf{Sign}, \mathbf{Sen}, \{\vdash^{\Sigma}\}_{\Sigma \in |\mathsf{Sign}|}\rangle$ *be an entailment system and* $\langle \Sigma,\Gamma\rangle \in |\mathsf{Th}_0|$. *We define* $^{\bullet} : 2^{\mathbf{Sen}(\Sigma)} \to 2^{\mathbf{Sen}(\Sigma)}$ *as follows:* $\Gamma^{\bullet} = \{\, \gamma \mid \Gamma \vdash^{\Sigma} \gamma \,\}$. *This function is extended to elements of* Th_0*, by defining it as follows:* $\langle \Sigma,\Gamma\rangle^{\bullet} = \langle \Sigma,\Gamma^{\bullet}\rangle$. $\Gamma^{\bullet}$ *is called the theory generated by* Γ.

Definition 5. ([2]) *Let* $\langle \mathsf{Sign}, \mathbf{Sen}, \{\vdash^{\Sigma}\}_{\Sigma \in |\mathsf{Sign}|}\rangle$ *and* $\langle \mathsf{Sign}', \mathbf{Sen}', \{\vdash'^{\Sigma}\}_{\Sigma \in |\mathsf{Sign}'|}\rangle$ *be entailment systems,* $\Phi : \mathsf{Th}_0 \to \mathsf{Th}'_0$ *be a functor and* $\alpha : \mathbf{Sen} \to \mathbf{Sen}' \circ \Phi$ *a natural transformation.* Φ *is said to be* α-sensible *if and only if the following conditions are satisfied:*

1. *there is a functor* $\Phi^{\diamond} : \mathsf{Sign} \to \mathsf{Sign}'$ *such that* $\mathbf{sign}' \circ \Phi = \Phi^{\diamond} \circ \mathbf{sign}$*, where* $\mathbf{sign}$ *and* $\mathbf{sign}'$ *are the forgetful functors from the corresponding categories of theories to the corresponding categories of signatures, that when applied to a given theory project its signature, and*
2. *if* $\langle \Sigma, \Gamma \rangle \in |\mathsf{Th}_0|$ *and* $\langle \Sigma', \Gamma' \rangle \in |\mathsf{Th}'_0|$ *such that* $\Phi(\langle \Sigma, \Gamma \rangle) = \langle \Sigma', \Gamma' \rangle$*, then* $(\Gamma')^{\bullet} = (\emptyset' \cup \alpha_{\Sigma}(\Gamma))^{\bullet}$*, where* $\emptyset' = \alpha_{\Sigma}(\emptyset)$[6]*.*

Φ is said to be α-simple *if and only if* $\Gamma' = \emptyset' \cup \alpha_{\Sigma}(\Gamma)$ *is satisfied in Condition 2, instead of* $(\Gamma')^{\bullet} = (\emptyset' \cup \alpha_{\Sigma}(\Gamma))^{\bullet}$*.*

It is straightforward to see, based on the monotonicity of $^{\bullet}$, that α-simplicity implies α-sensibility. An α-sensible functor has the property that the associated natural transformation α depends only on signatures. Now, from Definitions 1 and 2, it is possible to give a definition of *logic* by relating both its model-theoretic and proof-theoretic characterizations; a coherence between the semantic and syntactic relations is required, reflecting the soundness and completeness of standard deductive relations of logical systems.

Definition 6. ([2]) *A* logic *is a structure of the form* $\langle \mathsf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \{\vdash^{\Sigma}\}_{\Sigma \in |\mathsf{Sign}|}, \{\models^{\Sigma}\}_{\Sigma \in |\mathsf{Sign}|} \rangle$ *satisfying the following conditions:*

- $\langle \mathsf{Sign}, \mathbf{Sen}, \{\vdash^{\Sigma}\}_{\Sigma \in |\mathsf{Sign}|} \rangle$ *is an entailment system,*
- $\langle \mathsf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \{\models^{\Sigma}\}_{\Sigma \in |\mathsf{Sign}|} \rangle$ *is an institution, and*
- *the following* soundness *condition is satisfied: for any* $\Sigma \in |\mathsf{Sign}|$*,* $\phi \in \mathbf{Sen}(\Sigma)$*,* $\Gamma \subseteq \mathbf{Sen}(\Sigma)$*:* $\Gamma \vdash^{\Sigma} \phi \quad$ *implies* $\quad \Gamma \models^{\Sigma} \phi$.

A logic is complete *if, in addition, the following condition is also satisfied: for any* $\Sigma \in |\mathsf{Sign}|$*,* $\phi \in \mathbf{Sen}(\Sigma)$*,* $\Gamma \subseteq \mathbf{Sen}(\Sigma)$*:* $\Gamma \models^{\Sigma} \phi \quad$ *implies* $\quad \Gamma \vdash^{\Sigma} \phi$.

Definition 2 associates deductive relations to signatures. As already discussed, it is important to analyze how these relations are obtained. The next definition introduces the notion of *proof calculus.* It formalizes the possibility of associating a proof-theoretic structure to the deductive relations introduced by the definitions of entailment systems. In [2, Ex. 11, pp. 15], Meseguer presents natural deduction as a proof calculus for first-order predicate logic by resorting to *multicategories* (see [2, Definition 10]).

Definition 7. ([2]) *A* proof calculus *is a structure of the form* $\langle \mathsf{Sign}, \mathbf{Sen}, \{\vdash^{\Sigma}\}_{\Sigma \in |\mathsf{Sign}|}, \mathbf{P}, \mathbf{Pr}, \pi \rangle$ *satisfying the following conditions:*

- $\langle \mathsf{Sign}, \mathbf{Sen}, \{\vdash^{\Sigma}\}_{\Sigma \in |\mathsf{Sign}|} \rangle$ *is an entailment system,*
- $\mathbf{P} : \mathsf{Th}_0 \to \mathsf{Struct}_{PC}$ *is a functor. Let* $T \in |\mathsf{Th}_0|$*, then* $\mathbf{P}(T) \in |\mathsf{Struct}_{PC}|$ *is the proof-theoretical structure of* T*,*
- $\mathbf{Pr} : \mathsf{Struct}_{PC} \to \mathsf{Set}$ *is a functor. Let* $T \in |\mathsf{Th}_0|$*, then* $\mathbf{Pr}(\mathbf{P}(T))$ *is the set of proofs of* T*; the composite functor* $\mathbf{Pr} \circ \mathbf{P} : \mathsf{Th}_0 \to \mathsf{Set}$ *will be denoted by* $\mathbf{proofs}$*, and*

[6] $\emptyset'$ is not necessarily the empty set of axioms. This fact will be clarified later on.

- π : **proofs** $\dot{\to}$ **Sen** *is a natural transformation such that for each* $T = \langle \Sigma, \Gamma \rangle \in |\mathsf{Th}_0|$ *the image of* $\pi_T : \mathbf{proofs}(T) \to \mathbf{Sen}(T)$ *is the set* $\Gamma^\bullet$*. The map* π_T *is called the* projection from proofs to theorems *for the theory* T*.*

Finally, a *logical system* is defined as a logic plus a proof calculus for its proof theory.

Definition 8. ([2]) *A* logical system *is a structure of the form*

$$\langle \mathsf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \{\vdash^\Sigma\}_{\Sigma \in |\mathsf{Sign}|}, \{\models^\Sigma\}_{\Sigma \in |\mathsf{Sign}|}, \mathbf{P}, \mathbf{Pr}, \pi \rangle$$

satisfying the following conditions:

- $\langle \mathsf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \{\vdash^\Sigma\}_{\Sigma \in |\mathsf{Sign}|}, \{\models^\Sigma\}_{\Sigma \in |\mathsf{Sign}|} \rangle$ *is a logic, and*
- $\langle \mathsf{Sign}, \mathbf{Sen}, \{\vdash^\Sigma\}_{\Sigma \in |\mathsf{Sign}|}, \mathbf{P}, \mathbf{Pr}, \pi \rangle$ *is a proof calculus.*

3 Satisfiability Calculus

In Section 2, we presented the definitions of institutions and entailment systems. Additionally, we presented Meseguer's categorical formulation of proof that provides operational structure for the abstract notion of entailment. In this section, we provide a categorical definition of a satisfiability calculus, providing a corresponding operational formulation of satisfiability. A satisfiability calculus is the formal characterization of a method for constructing models of a given theory, thus providing the semantic counterpart of a proof calculus. Roughly speaking, the semantic relation of satisfaction between a model and a formula can also be *implemented* by means of some kind of structure that depends on the model theory of the logic. The definition of a satisfiability calculus is as follows:

Definition 9. [Satisfiability Calculus] *A* satisfiability calculus *is a structure of the form* $\langle \mathsf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \{\models^\Sigma\}_{\Sigma \in |\mathsf{Sign}|}, \mathbf{M}, \mathbf{Mods}, \mu \rangle$ *satisfying the following conditions:*

- $\langle \mathsf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \{\models^\Sigma\}_{\Sigma \in |\mathsf{Sign}|} \rangle$ *is an institution,*
- $\mathbf{M} : \mathsf{Th}_0 \to \mathsf{Struct}_{SC}$ *is a functor. Let* $T \in |\mathsf{Th}_0|$*, then* $\mathbf{M}(T) \in |\mathsf{Struct}_{SC}|$ *is the model structure of* T*,*
- $\mathbf{Mods} : \mathsf{Struct}_{SC} \to \mathsf{Cat}$ *is a functor. Let* $T \in |\mathsf{Th}_0|$*, then* $\mathbf{Mods}(\mathbf{M}(T))$ *is the category of canonical models of* T*; the composite functor* $\mathbf{Mods} \circ \mathbf{M} : \mathsf{Th}_0 \to \mathsf{Cat}$ *will be denoted by* **models***, and*
- $\mu : \mathbf{models}^{\mathsf{op}} \dot{\to} \mathbf{Mod}$ *is a natural transformation such that, for each* $T = \langle \Sigma, \Gamma \rangle \in |\mathsf{Th}_0|$*, the image of* $\mu_T : \mathbf{models}^{\mathsf{op}}(T) \to \mathbf{Mod}(T)$ *is the category of models* $\mathbf{Mod}(T)$*. The map* μ_T *is called the* projection of the category of models *of the theory* T*.*

The intuition behind the previous definition is that, for any theory T, the functor $\mathbf{M}$ assigns a model structure for T in the category Struct_{SC}[7]. For instance,

[7] Notice that the target of functor $\mathbf{M}$, when applied to a theory T, is not necessarily a model, but a structure which, under certain conditions, can be considered a representation of the category of models of T.

in propositional tableaux, a good choice for $Struct_{SC}$ is the collection of legal tableaux, where the functor M maps a theory to the collection of tableaux obtained for that theory. The functor **Mods** projects those particular structures that represent sets of conditions that can produce canonical models of a theory $T = \langle \Sigma, \Gamma \rangle$ (i.e., the structures that represent canonical models of Γ). For example, in the case of propositional tableaux, this functor selects the open branches of tableaux, that represent satisfiable sets of formulae, and returns the collections of formulae obtained by closuring these sets. Finally, for any theory T, the functor μ_T relates each of these sets of conditions to the corresponding canonical model. Again, in propositional tableaux, this functor is obtained by relating a closured set of formulae with the models that can be defined from these sets of formulae in the usual ways [20].

Example 1. ***[Tableaux Method for First-Order Predicate Logic]*** Let us start by presenting the tableaux method for first-order logic. Let us denote by $\mathbb{I}_{FOL} = \langle \mathsf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \{\models^{\Sigma}\}_{\Sigma \in |\mathsf{Sign}|} \rangle$ the institution of first-order predicate logic. Let $\Sigma \in |\mathsf{Sign}|$ and $S \subseteq \mathbf{Sen}(\Sigma)$; then a *tableau* for S is a tree such that:

1. the nodes are labeled with sets of formulae (over Σ) and the root node is labeled with S,
2. if u and v are two connected nodes in the tree (u being an ancestor of v), then the label of v is obtained from the label of u by applying one of the following rules:

$$\frac{X \cup \{A \wedge B\}}{X \cup \{A \wedge B, A, B\}}\ [\wedge] \qquad \frac{X \cup \{A \vee B\}}{X \cup \{A \vee B, A\} \qquad X \cup \{A \vee B, B\}}\ [\vee]$$

$$\frac{X \cup \{\neg\neg A\}}{X \cup \{\neg\neg A, A\}}\ [\neg_1] \qquad \frac{X \cup \{A\}}{X \cup \{A, \neg\neg A\}}\ [\neg_2] \qquad \frac{X \cup \{A, \neg A\}}{\mathbf{Sen}(\Sigma)}\ [\mathit{false}]$$

$$\frac{X \cup \{\neg(A \wedge B)\}}{X \cup \{\neg(A \wedge B), \neg A \vee \neg B\}}\ [DM_1] \qquad \frac{X \cup \{\neg(A \vee B)\}}{X \cup \{\neg(A \vee B), \neg A \wedge \neg B\}}\ [DM_2]$$

$$\frac{X \cup \{(\forall x)P(x)\}}{X \cup \{(\forall x)P(x), P(t)\}}\ [\forall] \qquad \frac{X \cup \{(\exists x)P(x)\}}{X \cup \{(\exists x)P(x), P(c)\}}\ [\exists]$$

where, in the last rules, c is a new constant and t is a ground term. A sequence of nodes $s_0 \xrightarrow{\tau_0^{\alpha_0}} s_1 \xrightarrow{\tau_1^{\alpha_1}} s_2 \xrightarrow{\tau_2^{\alpha_2}} \dots$ is a *branch* if: *a*) s_0 is the root node of the tree, and *b*) for all $i \leq \omega$, $s_i \rightarrow s_{i+1}$ occurs in the tree, $\tau_i^{\alpha_i}$ is an instance of one of the rules presented above, and α_i are the formulae of s_i to which the rule was applied. A branch $s_0 \xrightarrow{\tau_0^{\alpha_0}} s_1 \xrightarrow{\tau_1^{\alpha_1}} s_2 \xrightarrow{\tau_2^{\alpha_2}} \dots$ in a tableau is *saturated* if there exists $i \leq \omega$ such that $s_i = s_{i+1}$. A branch $s_0 \xrightarrow{\tau_0^{\alpha_0}} s_1 \xrightarrow{\tau_1^{\alpha_1}} s_2 \xrightarrow{\tau_2^{\alpha_2}} \dots$ in a tableau is *closed* if there exists $i \leq \omega$ and $\alpha \in \mathbf{Sen}(\Sigma)$ such that $\{\alpha, \neg\alpha\} \subseteq s_i$.

Let $s_0 \xrightarrow{\tau_0^{\alpha_0}} s_1 \xrightarrow{\tau_1^{\alpha_1}} s_2 \xrightarrow{\tau_2^{\alpha_2}} \dots$ be a branch in a tableau. Examining the rules presented above, it is straightforward to see that every s_i with $i < \omega$ is a set of formulae. In each step, we have either the application of a rule decomposing one formula of the set into its constituent parts with respect to its major connective, while preserving satisfiability, or the application of the rule $[false]$ denoting the fact that the corresponding set of formulae is unsatisfiable. Thus, the limit set of the branch is a set of formulae containing sub-formulae (and "*instances*" in the case of quantifiers) of the original set of formulae for which the tableau was built. As a result of this, every open branch expresses, by means of a set of formulae, the class of models satisfying them.

In order to define the tableau method as a satisfiability calculus, we provide formal definitions for $\mathbf{M}$, $\mathbf{Mods}$ and μ. The proofs of the lemmas and properties shown below are straightforward using the introduced definitions. The interested reader can find these proofs in [24]. First, we introduce the category $Str^{\Sigma,\Gamma}$ of tableaux for sets of formulae over signature Σ and assuming the set of axioms Γ. In $Str^{\Sigma,\Gamma}$, objects are sets of formulae over signature Σ, and morphisms represent tableaux for the set occurring in their target and having subsets of the set of formulae occurring at the end of open branches, as their source.

Definition 10. *Let $\Sigma \in |\mathsf{Sign}|$ and $\Gamma \subseteq \mathbf{Sen}(\Sigma)$, then we define $Str^{\Sigma,\Gamma} = \langle \mathcal{O}, \mathcal{A} \rangle$ such that $\mathcal{O} = 2^{\mathbf{Sen}(\Sigma)}$ and $\mathcal{A} = \{\alpha : \{A_i\}_{i \in \mathcal{I}} \to \{B_j\}_{j \in \mathcal{J}} \mid \alpha = \{\alpha_j\}_{j \in \mathcal{J}}\}$, where for all $j \in \mathcal{J}$, α_j is a branch in a tableau for $\Gamma \cup \{B_j\}$ with leaves $\Delta \subseteq \{A_i\}_{i \in \mathcal{I}}$. It should be noted that $\Delta \models_{\Sigma} \Gamma \cup \{B_j\}$.*

Lemma 1. *Let $\Sigma \in |\mathsf{Sign}|$ and $\Gamma \subseteq \mathbf{Sen}(\Sigma)$; then $\langle Str^{\Sigma,\Gamma}, \cup, \emptyset \rangle$, where $\cup : Str^{\Sigma,\Gamma} \times Str^{\Sigma,\Gamma} \to Str^{\Sigma,\Gamma}$ is the typical bi-functor on sets and functions, and $\emptyset$ is the neutral element for $\cup$, is a strict monoidal category.*

Using this definition we can introduce the category of legal tableaux, denoted by $Struct_{SC}$.

Definition 11. *Struct_{SC} is defined as $\langle \mathcal{O}, \mathcal{A} \rangle$ where $\mathcal{O} = \{Str^{\Sigma,\Gamma} \mid \Sigma \in |\mathsf{Sign}| \wedge \Gamma \subseteq \mathbf{Sen}(\Sigma)\}$, and $\mathcal{A} = \{\widehat{\sigma} : Str^{\Sigma,\Gamma} \to Str^{\Sigma',\Gamma'} \mid \sigma : \langle \Sigma, \Gamma \rangle \to \langle \Sigma', \Gamma' \rangle \in ||\mathsf{Th}_0||\}$, the homomorphic extension of the morphisms in $||\mathsf{Th}_0||$.*

Lemma 2. *Struct_{SC} is a category.*

The functor $\mathbf{M}$ must be understood as the relation between a theory in $|\mathsf{Th}_0|$ and its category of structures representing legal tableaux. So, for every theory T, $\mathbf{M}$ associates the strict monoidal category [22] $\langle Str^{\Sigma,\Gamma}, \cup, \emptyset \rangle$, and for every theory morphism $\sigma : \langle \Sigma, \Gamma \rangle \to \langle \Sigma', \Gamma' \rangle$, $\mathbf{M}$ associates a morphism $\widehat{\sigma} : Str^{\Sigma,\Gamma} \to Str^{\Sigma',\Gamma'}$ which is the homomorphic extension of σ to the structure of the tableaux.

Definition 12. *$\mathbf{M} : \mathsf{Th}_0 \to \mathsf{Struct}_{SC}$ is defined as $\mathbf{M}(\langle \Sigma, \Gamma \rangle) = \langle Str^{\Sigma,\Gamma}, \cup, \emptyset \rangle$ and $\mathbf{M}(\sigma : \langle \Sigma, \Gamma \rangle \to \langle \Sigma', \Gamma' \rangle) = \widehat{\sigma} : \langle Str^{\Sigma,\Gamma}, \cup, \emptyset \rangle \to \langle Str^{\Sigma',\Gamma'}, \cup, \emptyset \rangle$, the homomorphic extension of σ to the structures in $\langle Str^{\Sigma,\Gamma}, \cup, \emptyset \rangle$.*

Lemma 3. *$\mathbf{M}$ is a functor.*

In order to define $Mods$, we need the following auxiliary definition, which resembles the usual construction of maximal consistent sets of formulae.

Definition 13. *Let $\Sigma \in |\mathsf{Sign}|$, $\Delta \subseteq \mathbf{Sen}(\Sigma)$, and consider $\{F_i\}_{i<\omega}$ an enumeration of $\mathbf{Sen}(\Sigma)$ such that for every formula α, its sub-formulae are enumerated before α. Then $Cn(\Delta)$ is defined as follows:*

- $Cn(\Delta) = \bigcup_{i<\omega} Cn^i(\Delta)$
- $Cn^0(\Delta) = \Delta$, $Cn^{i+1}(\Delta) = \begin{cases} Cn^i(\Delta) \cup \{F_i\} & \text{, if } Cn^i(\Delta) \cup \{F_i\} \text{ is consistent.} \\ Cn^i(\Delta) \cup \{\neg F_i\} & \text{, otherwise.} \end{cases}$

Given $\langle \Sigma, \Gamma \rangle \in |\mathsf{Th}_0|$, the functor **Mods** provide the means for obtaining the category containing the closure of those structures in $Str^{\Sigma,\Gamma}$ that represent the closure of the branches in saturated tableaux.

Definition 14. $\mathbf{Mods} : \mathsf{Struct}_{SC} \to \mathsf{Cat}$ *is defined as:*

$$\begin{aligned}\mathbf{Mods}(\langle Str^{\Sigma,\Gamma}, \cup, \emptyset \rangle) = \{\langle \Sigma, Cn(\widetilde{\Delta}) \rangle \mid (\exists \alpha : \Delta \to \emptyset \in ||Str^{\Sigma,\Gamma}||) \\ (\widetilde{\Delta} \to \emptyset \in \alpha \wedge (\forall \alpha' : \Delta' \to \Delta \in ||Str^{\Sigma,\Gamma}||)(\Delta' = \Delta))\}\end{aligned}$$

and for all $\sigma : \Sigma \to \Sigma' \in |\mathsf{Sign}|$ (and $\widehat{\sigma} : \langle Str^{\Sigma,\Gamma}, \cup, \emptyset \rangle \to \langle Str^{\Sigma',\Gamma'}, \cup, \emptyset \rangle \in ||\mathsf{Struct}_{SC}||$), the following holds:

$$\mathbf{Mods}(\widehat{\sigma})(\langle \Sigma, Cn(\widetilde{\Delta}) \rangle) = \langle \Sigma', Cn(\mathbf{Sen}(\sigma)(Cn(\widetilde{\Delta}))) \rangle.$$

Lemma 4. **Mods** *is a functor.*

Finally, the natural transformation μ relates the structures representing saturated tableaux with the model satisfying the set of formulae denoted by the source of the morphism.

Definition 15. *Let $\langle \Sigma, \Gamma \rangle \in |\mathsf{Th}_0|$, then we define $\mu_\Sigma : \mathbf{models}^{\mathsf{op}}(\langle \Sigma, \Gamma \rangle) \to \mathbf{Mod}_{FOL}(\langle \Sigma, \Gamma \rangle)$ as $\mu_\Sigma(\langle \Sigma, \Delta \rangle) = \mathbf{Mod}(\langle \Sigma, \Delta \rangle)$.*

Fact 1 *Let $\Sigma \in |\mathsf{Sign}_{FOL}|$ and $\Gamma \subseteq \mathbf{Sen}_{FOL}(\Sigma)$. Then $\mu_{\langle \Sigma, \Gamma \rangle}$ is a functor.*

Lemma 5. *μ is a natural transformation.*

Now, from Lemmas 3, 4, and 5, and considering the hypothesis that $\mathbb{I}_{FOL}$ is an institution, the following corollary follows.

Corollary 1. *$\langle \mathsf{Sign}_{FOL}, \mathbf{Sen}_{FOL}, \mathbf{Mod}_{FOL}, \{\models^{\Sigma}_{FOL}\}_{\Sigma \in |\mathsf{Sign}_{FOL}|}, \mathbf{M}, \mathbf{Mods}, \mu \rangle$ is a satisfiability calculus.*

Another important kind of system used by automatic theorem provers are the so-called resolution methods. Below, we show how any resolution system conforms to the definition of satisfiability calculus.

*Example 2. **[Resolution Method for First-Order Predicate Logic]*** Let us describe resolution for first-order logic as introduced in [25]. We use the following notation: [] denotes the empty list; $[A]$ denotes the unitary list containing the formula A; $\ell_0, \ell_1, \ldots$ are variables ranging over lists; and $\ell_i + \ell_j$ denotes the concatenation of lists ℓ_i and ℓ_j. Resolution builds a list of lists representing a disjunction of conjunctions. The rules for resolution are the following:

$$\frac{\ell_0 + [\neg\neg A] + \ell_1}{\ell_0 + [A] + \ell_1}\,[\neg\neg] \quad \frac{\begin{array}{c}\ell_0 + [\neg A] + \ell_1 \\ \ell'_0 + [A] + \ell'_1\end{array}}{\ell_0 + \ell_1 + \ell'_0 + \ell'_1}\,[\neg] \quad \frac{\ell_0 + [A \wedge A'] + \ell_1}{\ell_0 + [A, A'] + \ell_1}\,[\wedge]$$

$$\frac{\ell_0 + [\neg(A \vee A')] + \ell_1}{\ell_0 + [\neg A, \neg A'] + \ell_1}\,[\neg\wedge] \quad \frac{\ell_0 + [A \vee A'] + \ell_1}{\begin{array}{c}\ell_0 + [A] + \ell_1 \\ \ell_0 + [A'] + \ell_1\end{array}}\,[\vee] \quad \frac{\ell_0 + [\neg(A \wedge A')] + \ell_1}{\begin{array}{c}\ell_0 + [\neg A] + \ell_1 \\ \ell_0 + [\neg A'] + \ell_1\end{array}}\,[\neg\wedge]$$

$$\text{for any closed term } t\ \frac{\ell_0 + [\forall x : A(x)] + \ell_1}{\ell_0 + [A[x/t]] + \ell_1}\,[\forall]$$

$$\text{for a new constant } c\ \frac{\ell_0 + [\exists x : A(x)] + \ell_1}{\ell_0 + [A[x/c]] + \ell_1}\,[\exists]$$

where $A(x)$ denotes a formula with free variable x, and $A[x/t]$ denotes the formula resulting from replacing variable x by term t everywhere in A. For the sake of simplicity, we assume that lists of formulae do not have repeated elements. A resolution is a sequence of lists of formulae. If a resolution contains an empty list (i.e., $[]$), we say that the resolution is closed; otherwise it is an *open* resolution. For every signature $\Sigma \in |\mathsf{Sign}|$ and each $\Gamma \subset \mathbf{Sen}(\Sigma)$, we denote by $Str^{\Sigma,\Gamma}$ the category whose objects are lists of formulae, and where every morphism $\sigma : [A_0, \ldots, A_n] \to [A'_0, \ldots, A'_m]$ represents a sequence of application of resolution rules for $[A'_0, \ldots, A'_m]$. Then, Struct_{SC} is a category whose objects are $Str^{\Sigma,\Gamma}$, for each signature $\Sigma \in |\mathsf{Sign}|$ and set of formulae $\Gamma \in \mathbf{Sen}(\Sigma)$, and whose morphisms are of the form $\widehat{\sigma} : Str^{\Sigma,\Gamma} \to Str^{\Sigma',\Gamma'}$, obtained by homomorphically extending $\sigma : \langle \Sigma, \Gamma \rangle \to \langle \Sigma', \Gamma' \rangle$ in $||\mathsf{Th}_0||$.

As for the case of Example 1, the functor $\mathbf{M} : \mathbf{Th}_0 \to \mathsf{Struct}_{SC}$ is defined as $\mathbf{M}(\langle \Sigma, \Gamma \rangle) = \langle Str^{\Sigma,\Gamma}, \cup, \emptyset \rangle$, and $\mathbf{Mods} : \mathsf{Struct}_{SC} \to \mathsf{Set}$ is defined as in the previous example.

A typical use for the methods involved in the above described examples is the search for counterexamples of a given logical property. For instance, to search for counterexamples of an intended property in the context of the tableaux method, one starts by applying rules to the negation of the property, and once a saturated tableau is obtained, if all the branches are closed, then there is no model of the axioms and the negation of the property, indicating that the latter is a theorem. On the other hand, if there exists an open branch, the limit set of that branch characterizes a class of counterexamples for the formula. Notice the contrast with Hilbert systems, where one starts from the axioms, and then applies deduction rules until the desired formula is obtained.

3.1 Mapping Satisfiability Calculi

In [4] the original notion of morphism between Institutions was introduced. Meseguer defines the notion of plain map in [2], and in [5] Tarlecki extensively discussed the ways in which different institutions can be related, and how they should be interpreted. More recently, in [26] all these notions of morphism were investigated in detail. In this work we will concentrate only on institution representations (or comorphisms in the terminology introduced by Goguen and Rosu), since this is the notion that we have employed to formalize several concepts arising from software engineering, such as *data refinement* and *dynamic reconfiguration* [27, 28]. The study of other important kinds of functorial relations between satisfiability calculi are left as future work. The following definition is taken from [5], and formalizes the notion of institution representation.

Definition 16. ([5]) *Let* $\mathbb{I} = \langle \mathsf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \{\models_\Sigma\}_{\Sigma \in |\mathsf{Sign}|}\rangle$ *and* $\mathbb{I}' = \langle \mathsf{Sign}', \mathbf{Sen}', \mathbf{Mod}', \{\models'_\Sigma\}_{\Sigma \in |\mathsf{Sign}'|}\rangle$ *be institutions. Then,* $\langle \gamma^{Sign}, \gamma^{Sen}, \gamma^{Mod}\rangle : I \to I'$ *is an* institution representation *if and only if:*

- $\gamma^{Sign} : \mathsf{Sign} \to \mathsf{Sign}'$ *is a functor,*
- $\gamma^{Sen} : \mathbf{Sen} \dot{\to} \gamma^{Sign} \circ \mathbf{Sen}'$*, is a natural transformation,*
- $\gamma^{Mod} : (\gamma^{Sign})^{\mathsf{op}} \circ \mathbf{Mod}' \dot{\to} \mathbf{Mod}$*, is a natural transformation,*

such that for any $\Sigma \in |\mathsf{Sign}|$*, the function* $\gamma^{Sen}_\Sigma : \mathbf{Sen}(\Sigma) \to \mathbf{Sen}'(\gamma^{Sign}(\Sigma))$ *and the functor* $\gamma^{Mod}_\Sigma : \mathbf{Mod}'(\gamma^{Sign}(\Sigma)) \to \mathbf{Mod}(\Sigma)$ *preserve the following satisfaction condition: for any* $\alpha \in \mathbf{Sen}(\Sigma)$ *and* $\mathcal{M}' \in |\mathbf{Mod}(\gamma^{Sign}(\Sigma))|$,

$$\mathcal{M}' \models^{\gamma^{Sign}(\Sigma)} \gamma^{Sen}_\Sigma(\alpha) \quad \textit{iff} \quad \gamma^{Mod}_\Sigma(\mathcal{M}') \models^\Sigma \alpha \ .$$

An institution representation $\gamma : I \to I'$ expresses how the “poorer” set of sentences (respectively, category of models) associated with I is encoded in the “richer” one associated with I'. This is done by:

- constructing, for a given I-signature Σ, an I'-signature into which Σ can be interpreted,
- translating, for a given I-signature Σ, the set of Σ-sentences into the corresponding I'-sentences,
- obtaining, for a given I-signature Σ, the category of Σ-models from the corresponding category of Σ'-models.

The direction of the arrows shows how the whole of I is represented by some parts of I'. Institution representations enjoy some interesting properties. For instance, logical consequence is preserved, and, under some conditions, logical consequence is preserved in a conservative way. The interested reader is referred to [5] for further details.

In many cases, in particular those in which the class of models of a signature in the source institution is completely axiomatizable in the language of the target one, Definition 16 can easily be extended to map signatures of one institution to theories of another. This is done so that the class of models of the richer one can be restricted, by means of the addition of axioms (thus the need for theories in the image of the functor γ^{Sign}), in order to be exactly the class of models obtained by translating to it the class of models of the corresponding signature of the poorer one. In the same way, when the previously described extension is possible, we can obtain what Meseguer calls a *map of institutions* [2, definition 27] by reformulating the definition so that the functor between signatures of one institution and theories of the other is $\gamma^{Th} : \mathsf{Th}_0 \to \mathsf{Th}'_0$. This has to be γ^{Sen}-sensible (see definition 5) with respect to the entailment systems induced by the institutions I and I'. Now, if $\langle \Sigma, \Gamma \rangle \in |\mathsf{Th}_0|$, then γ^{Th_0} can be defined as follows: $\gamma^{Th_0}(\langle \Sigma, \Gamma \rangle) = \langle \gamma^{Sign}(\Sigma), \Delta \cup \gamma^{Sen}_{\Sigma}(\Gamma) \rangle$, where $\Delta \subseteq \mathbf{Sen}(\rho^{Sign}(\Sigma))$. Then, it is easy to prove that γ^{Th_0} is γ^{Sen}-simple because it is the γ^{Sen}-extension of γ^{Th_0} to theories, thus being γ^{Sen}-sensible.

The notion of a *map of satisfiability calculi* is the natural extension of a map of institutions in order to consider the more material version of the satisfiability relation. In some sense, if a map of institutions provides a means for representing one satisfiability relation in terms of another in a semantics preserving way, the map of satisfiability calculi provides a means for representing a model construction technique in terms of another. This is done by showing how model construction techniques for richer logics express techniques associated with poorer ones.

Definition 17. *Let* $\mathbb{S} = \langle \mathsf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \{\models^{\Sigma}\}_{\Sigma \in |\mathsf{Sign}|}, \mathbf{M}, \mathbf{Mods}, \mu \rangle$ *and* $\mathbb{S}' = \langle \mathsf{Sign}', \mathbf{Sen}', \mathbf{Mod}', \{\models'^{\Sigma}\}_{\Sigma \in |\mathsf{Sign}'|}, \mathbf{M}', \mathbf{Mods}', \mu' \rangle$ *be satisfiability calculi. Then,* $\langle \rho^{Sign}, \rho^{Sen}, \rho^{Mod}, \gamma \rangle : \mathbb{S} \to \mathbb{S}'$ *is a* map of satisfiability calculi *if and only if:*

1. $\langle \rho^{Sign}, \rho^{Sen}, \rho^{Mod} \rangle : \mathbb{I} \to \mathbb{I}'$ *is a map of institutions, and*
2. $\gamma : \mathbf{models}'^{\mathsf{op}} \circ \rho^{Th_0} \stackrel{.}{\to} \mathbf{models}^{\mathsf{op}}$ *is a natural transformation such that the following equality holds:*

$$\begin{tikzcd}
\mathsf{Th}_0 \arrow[rr, bend left=40, "\mathbf{Mod}"] \arrow[rr, bend right=20, "\mathbf{models}^{\mathsf{op}}"' description] \arrow[dr, bend right, "\rho^{Th_0}"'] \arrow[rr, phantom, shift left=3, "\Uparrow \mu"] & & \mathsf{Cat} & = & \mathsf{Th}_0 \arrow[rr, bend left=40, "\mathbf{Mod}"] \arrow[dr, bend right, "\rho^{Th_0}"'] \arrow[rr, phantom, shift left=3, "\Uparrow \rho^{Mod}_{\mathbf{Mod}'}"] & & \mathsf{Cat} \\
& \mathsf{Th}'_0 \arrow[ur, bend right, "\mathbf{models}'^{\mathsf{op}}"'] \arrow[u, phantom, "\Uparrow \gamma"] & & & & \mathsf{Th}'_0 \arrow[ur, bend left, "\mathbf{Mod}'"] \arrow[ur, bend right=50, "\mathbf{models}'^{\mathsf{op}}"'] \arrow[ur, phantom, "\Nearrow \mu'"] &
\end{tikzcd}$$

Roughly speaking, the 2-cell equality in the definition says that the translation of saturated tableaux is coherent with respect to the mapping of institutions.

Example 3. ***[Mapping Modal Logic to First-Order Logic]*** A simple example of a mapping between satisfiability calculi is the mapping between the tableau method for propositional logic, and the one for first-order logic. It is straightforward since the tableau method for first-order logic is an extension of that of propositional logic.

Let us introduce a more interesting example. We will map the tableau method for modal logic (as presented by Fitting [25]) to the first-order predicate logic tableau method. The mapping between the institutions is given by the standard translation from modal logic to first-order logic. Let us recast here the tableau method for the system K of modal logic. Recall that formulae of standard modal logic are built from boolean operators and the "diamond operator" $\Diamond$. Intuitively, formula $\Diamond\varphi$ says that φ is possibly true in some alternative state of affairs. The semantics for modal logic is given by means of Kripke structures. A Kripke structure is a tuple $\langle W, R, L\rangle$, where W is a set of states, $R \subseteq W \times W$ is a relation between states, and $L : W \to 2^{AP}$ is a labeling function (AP is a set of atomic propositions). Note that a signature in modal logic is given by a set of propositional letters: $\langle \{p_i\}_{i\in\mathcal{I}}\rangle$. The interested reader can consult [29].

In [25] modal formulae are prefixed by labels denoting semantic states. Labeled formulae are then terms of the form $\ell : \varphi$, where φ is a modal formula and ℓ is a sequence of natural numbers $n_0, \ldots, n_k$. The relation R between these labels is then defined in the following way: $\ell R \ell' \equiv \exists n : \ell, n = \ell'$. The new rules are the following:

$$\text{For all } \ell' \text{ such that } \ell R \ell' \text{ and such that } \ell' \text{ appears in } X \; \frac{X \cup \{\ell : \Box\varphi\}}{X \cup \{\ell : \Box\varphi, \ell' : \varphi\}} \, [\Box]$$

$$\text{For } \ell' \text{ such that } \ell R \ell' \; \frac{X \cup \{\ell : \Diamond\varphi\}}{X \cup \{\ell : \Diamond\varphi, \ell' : \varphi\}} \, [\Diamond]$$

The rules for the propositional connectives are the usual ones, obtained by labeling the formulae with a given label. Notice that labels denote states of a Kripke structure. This is related in some way to the tableau method used for first-order predicate logic. Branches, saturated branches and closed branches are defined in the same way as in Example 1, but considering the relations between sets to be also indexed by the relation used at that point. Thus, $s_i \xrightarrow[R_i]{\tau_{\alpha_i}} s_{i+1}$ must be understood as follows: the set s_{i+1} is obtained from s_i by applying rule τ_i to formula $\alpha_i \in s_i$ under the accessibility relation R_i.

Assume $\langle \mathsf{Sign}_{FOL}, \mathbf{Sen}_{FOL}, \mathbf{M}_{FOL}, \mathbf{Mods}_{FOL}, \{\models^{\Sigma}_{FOL}\}_{\Sigma\in|\mathsf{Sign}_{FOL}|}, \mu_{FOL}\rangle$ is the satisfiability calculus for first-order predicate logic, denoted by $\mathbb{SC}_{FOL}$, and $\langle \mathsf{Sign}_K, \mathbf{Sen}_K, \mathbf{M}_K, \mathbf{Mods}_K, \{\models^{\Sigma}_{K}\}_{\Sigma\in|\mathsf{Sign}_K|, \mu_K}\rangle$ is the satisfiability calculus for modal logic, denoted by $\mathbb{SC}_K$. Consider now the standard translation from modal logic to first-order logic. Therefore, the tuple $\langle \rho^{Sign}, \rho^{Sen}, \rho^{Mod}\rangle$ is defined as follows [29]:

Definition 18. *$\rho^{Sign} : \mathsf{Sign}_K \to \mathsf{Sign}_{FOL}$ is defined as $\rho^{Sign}(\langle\{p_i\}_{i\in\mathcal{I}}\rangle) = \langle R, \{p_i\}_{i\in\mathcal{I}}\rangle$ by mapping each propositional variable p_i, for all $i \in \mathcal{I}$, to a first-order unary logic predicate p_i, and adding a binary predicate R, and $\rho^{Sign}(\sigma : \langle\{p_i\}_{i\in\mathcal{I}}\rangle \to \langle\{p'_{i'}\}_{i'\in\mathcal{I}'}\rangle) = \sigma' : \langle R, \{p_i\}_{i\in\mathcal{I}}\rangle \to \langle R', \{p'_{i'}\}_{i'\in\mathcal{I}'}\rangle$ mapping R to R', and p_i to p'_i for all $i \in \mathcal{I}$.*

Lemma 6. *ρ^{Sign} is a functor.*

Definition 19. *Let $\langle\{p_i\}_{i\in\mathcal{I}}\rangle \in |\mathsf{Sign}_K|$. Then $\rho^{Sen}_{\langle\{p_i\}_{i\in\mathcal{I}}\rangle} : \mathbf{Sen}_K(\langle\{p_i\}_{i\in\mathcal{I}}\rangle) \to \rho^{Sign} \circ \mathbf{Sen}_{FOL}(\langle\{p_i\}_{i\in\mathcal{I}}\rangle)$ is defined recursively as $\rho^{Sen}_{\langle\{p_i\}_{i\in\mathcal{I}}\rangle}(\alpha) = T_{\langle\{p_i\}_{i\in\mathcal{I}}\rangle, x}(\alpha)$ where:*

$$\begin{aligned}
T_{\langle\{p_i\}_{i\in\mathcal{I}}\rangle, x}(p_i) &= p'_i(x), \textit{ for all } i \in \mathcal{I}.\\
T_{\langle\{p_i\}_{i\in\mathcal{I}}\rangle, x}(\neg\alpha) &= \neg T_{\langle\{p_i\}_{i\in\mathcal{I}}\rangle, x}(\alpha)\\
T_{\langle\{p_i\}_{i\in\mathcal{I}}\rangle, x}(\alpha \vee \beta) &= T_{\langle\{p_i\}_{i\in\mathcal{I}}\rangle, x}(\alpha) \vee T_{\langle\{p_i\}_{i\in\mathcal{I}}\rangle, x}(\beta)\\
T_{\langle\{p_i\}_{i\in\mathcal{I}}\rangle, x}(\Diamond\alpha) &= (\exists y)(R(x,y) \wedge T_{\langle\{p_i\}_{i\in\mathcal{I}}\rangle, y}(\alpha))
\end{aligned}$$

Lemma 7. *ρ^{Sen} is a natural transformation.*

Definition 20. *Let $\langle\{p_i\}_{i\in\mathcal{I}}\rangle \in |\mathsf{Sign}_K|$. Then we define $\rho^{Mod}_{\langle\{p_i\}_{i\in\mathcal{I}}\rangle} : \rho^{Sign} \circ \mathbf{Mod}_{FOL}(\langle\{p_i\}_{i\in\mathcal{I}}\rangle) \to \mathbf{Mod}_K(\langle\{p_i\}_{i\in\mathcal{I}}\rangle)$ as follows:*

- *for all $\mathcal{M} = \langle S, \overline{R}, \{\overline{p_i}\}_{i\in\mathcal{I}}\rangle \in |\mathbf{Mod}_{FOL}(\langle R, \{p_i\}_{i\in\mathcal{I}}\rangle)|$, $\rho^{Mod}_{\langle\{p_i\}_{i\in\mathcal{I}}\rangle}(\mathcal{M}) = \langle S, \overline{R}, L\rangle$, with $L(p_i) = \{s \in S | \overline{p_i}(s)\}$.*[8]
- *let $\langle\{p_i\}_{i\in\mathcal{I}}\rangle \in |\mathsf{Sign}_K|$; then for all homomorphism $h : \langle S_1, \overline{R_1}, \{\overline{p_{1i}}\}_{i\in\mathcal{I}}\rangle \to \langle S_2, \overline{R_2}, \{\overline{p_{2i}}\}_{i\in\mathcal{I}}\rangle \in ||\mathbf{Mod}_{FOL}(\langle R, \{p_i\}_{i\in\mathcal{I}}\rangle)||$, we define $\rho^{Mod}_{\langle\{p_i\}_{i\in\mathcal{I}}\rangle}(h)$ to be $\widehat{h}$, where $\widehat{h}(s_1) = s_2$ if and only if $h(s_1) = s_2$ for all $s_1 \in S_1$.*

Lemma 8. *Let $\langle\{p_i\}_{i\in\mathcal{I}}\rangle \in |\mathsf{Sign}_K|$. Then $\rho^{Mod}_{\langle\{p_i\}_{i\in\mathcal{I}}\rangle}$ is a functor.*

The proof that this is a mapping between institutions relies on the correctness of the translation presented in [29]. Using this map we can define a mapping between the corresponding satisfiability calculi. The natural transformation: $\gamma : \rho^{Th_0} \circ \mathbf{models}'^{\mathsf{op}} \dot{\to} \mathbf{models}^{\mathsf{op}}$ is defined as follows.

Definition 21. *Let $\langle\langle\{p_i\}_{i\in\mathcal{I}}\rangle, \Gamma\rangle \in |\mathsf{Th}_0^K|$; then*

$$\gamma_{\langle\{p_i\}_{i\in\mathcal{I}}\rangle} : \rho^{Th_0} \circ \mathbf{models}^{\mathsf{op}}_{FOL}(\langle\langle\{p_i\}_{i\in\mathcal{I}}\rangle, \Gamma\rangle) \to \mathbf{models}^{\mathsf{op}}_K(\langle\langle\{p_i\}_{i\in\mathcal{I}}\rangle, \Gamma\rangle)$$

is defined as:

$$\gamma_{\langle\{p_i\}_{i\in\mathcal{I}}\rangle}(\langle\langle R, \{p_i\}_{i\in\mathcal{I}}\rangle, \Delta\rangle) = \langle\langle\{p_i\}_{i\in\mathcal{I}}\rangle, \{\varphi \in |\mathbf{Sen}_K(\langle\{p_i\}_{i\in\mathcal{I}}\rangle)| \mid \rho^{Sen}_{\langle\{p_i\}_{i\in\mathcal{I}}\rangle}(\varphi) \in \Delta\}\rangle$$

Lemma 9. *Let $\langle\{p_i\}_{i\in\mathcal{I}}\rangle \in |\mathsf{Sign}_K|$; then $\gamma_{\langle\{p_i\}_{i\in\mathcal{I}}\rangle}$ is a functor.*

Lemma 10. *$\gamma : \rho^{Th_0} \circ \mathbf{models}^{\mathsf{op}}_{FOL} \to \mathbf{models}^{\mathsf{op}}_K$ is a natural transformation.*

Finally, the following lemma prove the equivalence of the two cells shown in Definition 17.

Lemma 11. *Let $\langle\{p_i\}_{i\in\mathcal{I}}\rangle \in |\mathsf{Sign}_K|$, then*

$$\mu_{K\,\langle\{p_i\}_{i\in\mathcal{I}}\rangle} \circ \gamma_{\langle\{p_i\}_{i\in\mathcal{I}}\rangle} = \rho^{Mod}_{\rho^{Sign}(\langle\{p_i\}_{i\in\mathcal{I}}\rangle)} \circ \mu_{FOL\,\rho^{Sign}(\langle\{p_i\}_{i\in\mathcal{I}}\rangle)} \ .$$

This means that building a tableau using the first-order rules for the translation of a modal theory, then obtaining the corresponding canonical model in modal logic using γ, and therefore obtaining the class of models by using μ, is exactly the same as obtaining the first-order models by μ' and then the corresponding modal models by using ρ^{Mod}.

[8] Notice that $\rho^{Sign}(\langle\{p_i\}_{i\in\mathcal{I}}\rangle) = \langle R, \{p_i\}_{i\in\mathcal{I}}\rangle$, where $\langle\{p_i\}_{i\in\mathcal{I}}\rangle \in |\mathsf{Sign}_K|$.

4 Conclusions and Further work

Methods like resolution and tableaux are strongly related to the semantics of a logic. They are often employed to construct models, a characteristic that is missing in purely deductive methods, such as natural deduction or Hilbert systems, as formalized by Meseguer. In this paper, we provided an abstract characterization of this class of semantics-based tecniques for logical systems. This was accomplished by introducing a categorical characterization of the notion of satisfiability calculus, which covers logical tools such as tableaux, resolution, Gentzen style sequents, etc. Our new characterization of a logical system, that includes the notion of satisfiability calculus, provides both a proof calculus and a satisfiability calculus, which essentially implement the entailment and satisfaction relations, respectively. There clearly exist connections between these calculi that are worth exploring, especially when the underlying structure used in both definitions is the same (see Example 1).

A close analysis of the definitions of proof calculus and satisfiability calculus takes us to observe that the constraints imposed over some elements (e.g., the natural family of functors $\pi_{\langle \Sigma, \Gamma \rangle} : \mathbf{proofs}(\langle \Sigma, \Gamma \rangle) \to \mathbf{Sen}(\langle \Sigma, \Gamma \rangle)$ and $\mu_{\langle \Sigma, \Gamma \rangle} : \mathbf{models}^{\mathsf{op}}(\langle \Sigma, \Gamma \rangle) \to \mathbf{Mod}(\langle \Sigma, \Gamma \rangle)$) may be too restrictive, and working on generalizations of these concepts is part of our further work. In particular, it is worth noticing that partial implementations of both the entailment relation and the satisfiability relation are gaining visibility in the software engineering community. Examples on the syntactic side are the implementation of less expressive calculi with respect to an entailment, as in the case of the finitary definition of the reflexive and transitive closure in the Kleene algebras with tests [30], the case of the implementation of rewriting tools like Maude [31] as a partial implementation of equational logic, etc. Examples on the semantic side are the bounded model checkers and model finders for undecidable languages, such as Alloy [32] for relational logic, the growing family of SMT-solvers [33] for languages including arithmetic, etc. Clearly, allowing for partial implementations of entailment/satisfiability relations would enable us to capture the behaviors of some of the above mentioned logical tools. In addition, functorial relations between partial proof calculi (resp., satisfiability calculi) may provide a measure for how good the method is as an approximation of the ideal entailment relation (resp., satisfaction relation). We plan to explore this possibility, as future work.

Acknowledgements

The authors would like to thank the anonymous referees for their helpful comments. This work was partially supported by the Argentinian Agency for Scientific and Technological Promotion (ANPCyT), through grants PICT PAE 2007 No. 2772, PICT 2010 No. 1690, PICT 2010 No. 2611 and PICT 2010 No. 1745, and by the MEALS project (EU FP7 programme, grant agreement No. 295261). The fourth author gratefully acknowledges the support of the National Science and Engineering Research Council of Canada and McMaster University.

References

1. Goguen, J.A., Burstall, R.M.: Introducing institutions. In Clarke, E.M., Kozen, D., eds.: Proceedings of the Carnegie Mellon Workshop on Logic of Programs. Volume 184 of LNCS., Springer-Verlag (1984) 221–256
2. Meseguer, J.: General logics. In Ebbinghaus, H.D., Fernandez-Prida, J., Garrido, M., Lascar, D., Artalejo, M.R., eds.: Proceedings of the Logic Colloquium ’87. Volume 129., Granada, Spain, North Holland (1989) 275–329
3. Fiadeiro, J.L., Maibaum, T.S.E.: Generalising interpretations between theories in the context of π-institutions. In Burn, G., Gay, D., Ryan, M., eds.: Proceedings of the First Imperial College Department of Computing Workshop on Theory and Formal Methods, London, UK, Springer-Verlag (1993) 126–147
4. Goguen, J.A., Burstall, R.M.: Institutions: abstract model theory for specification and programming. Journal of the ACM **39**(1) (1992) 95–146
5. Tarlecki, A.: Moving between logical systems. In Haveraaen, M., Owe, O., Dahl, O.J., eds.: Selected papers from the 11th Workshop on Specification of Abstract Data Types Joint with the 8th COMPASS Workshop on Recent Trends in Data Type Specification. Volume 1130 of LNCS., Springer-Verlag (1996) 478–502
6. Sannella, D., Tarlecki, A.: Specifications in an arbitrary institution. Information and computation **76**(2–3) (1988) 165–210
7. Tarlecki, A.: Abstract specification theory: an overview. In Broy, M., Pizka, M., eds.: Proceedings of the NATO Advanced Study Institute on Models, Algebras and Logic of Engineering Software. NATO Science Series, Marktoberdorf, Germany, IOS Press (2003) 43–79
8. Mossakowski, T.: Comorphism-based Grothendieck logics. In Diks, K., Rytter, W., eds.: MFCS. Volume 2420 of Lecture Notes in Computer Science., Springer (2002) 593–604
9. Mossakowski, T., Tarlecki, A.: Heterogeneous logical environments for distributed specifications. Volume 5486 of LNCS., Pisa, Italy, Springer-Verlag (2009) 266–289
10. Diaconescu, R., Futatsugi, K.: Logical foundations of CafeOBJ. Theoretical Computer Science **285**(2) (2002) 289–318
11. Diaconescu, R.: Grothendieck institutions. Applied Categorical Structures **10**(4) (2002) 383–402
12. Diaconescu, R., ed.: Institution-independent Model Theory. Volume 2 of Studies in Universal Logic. Birkhäuser (2008)
13. Mossakowski, T., Maeder, C., Luttich, K.: The heterogeneous tool set, Hets. In Grumberg, O., Huth, M., eds.: Proceedings of the 13th. International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2007). Volume 4424 of LNCS., Braga, Portugal, Springer-Verlag (2007) 519–522
14. Tarlecki, A.: Towards heterogeneous specifications. In Gabbay, D., de Rijke, M., eds.: Frontiers of Combining Systems. Volume 2 of Studies in Logic and Computation. Research Studies Press (2000) 337–360
15. Cengarle, M.V., Knapp, A., Tarlecki, A., Wirsing, M.: A heterogeneous approach to UML semantics. In Degano, P., DeNicola, R., Meseguer, J., eds.: Proceedings of Concurrency, graphs and models (Essays dedicated to Ugo Montanari on the occasion of his 65th. birthday). Number 5065 in LNCS, Edinburgh, Scotland, Springer-Verlag (2008) 383–402
16. Beth, E.W.: The Foundations of Mathematics. North Holland (1959)
17. Beth, E.W.: Semantic entailment and formal derivability. In Hintikka, J., ed.: The Philosophy of Mathematics. Oxford University Press (1969) 9–41 Reprinted from [34].

18. Herbrand, J.: Investigation in proof theory. In Goldfarb, W.D., ed.: Logical Writings. Harvard University Press (1969) 44–202 Translated to English from [35].
19. Gentzen, G.: Investigation into logical deduction. In Szabo, M.E., ed.: The Collected Papers of Gerhard Gentzen. North Holland (1969) 68–131 Translated to english from [36].
20. Smullyan, R.M.: First-order Logic. Dover Publishing (1995)
21. Robinson, J.A.: A machine-oriented logic based on the resolution principle. Journal of the ACM **12**(1) (1965) 23–41
22. McLane, S.: Categories for working mathematician. Graduate Texts in Mathematics. Springer-Verlag, Berlin, Germany (1971)
23. Fiadeiro, J.L.: Categories for software engineering. Springer-Verlag (2005)
24. Lopez Pombo, C.G., Castro, P., Aguirre, N.M., Maibaum, T.S.E.: Satisfiability calculus: the semantic counterpart of a proof calculus in general logics. Technical report, McMaster University, Centre for Software Certification (2011)
25. Fitting, M.: Tableau methods of proof for modal logics. Notre Dame Journal of Formal Logic **13**(2) (1972) 237–247 Lehman College.
26. Goguen, J.A., Rosu, G.: Institution morphisms. Formal Asp. Comput. **13**(3-5) (2002) 274–307
27. Castro, P., Aguirre, N.M., Lopez Pombo, C.G., Maibaum, T.S.E.: Towards managing dynamic reconfiguration of software systems in a categorical setting. In Cavalcanti, A., D'eharbe, D., Gaudel, M.C., Woodcock, J., eds.: Proceedings of Theoretical Aspects of Computing - ICTAC 2010, 7th International Colloquium. Volume 6255 of LNCS., Natal, Rio Grande do Norte, Brazil, Springer-Verlag (2010) 306–321
28. Castro, P., Aguirre, N.M., Lopez Pombo, C.G., Maibaum, T.S.E.: A categorical approach to structuring and promoting Z specifications. In Pasareanu, C., Salaün, G., eds.: Proceedings of FACS 2012, 9th International Symposium. Volume 7684 of LNCS., Moutain View, California, USA, Springer-Verlag (2012) 73–88
29. Blackburn, P., de Rijke, M., Venema, Y.: Modal logic. Number 53 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press (2001)
30. Kozen, D.: Kleene algebra with tests. ACM Transactions on Programming Languages and Systems **19**(3) (1997) 427–443
31. Clavel, M., Durán, F., Eker, S., Lincoln, P., Martí-Oliet, N., Meseguer, J., Talcott, C.L., eds.: All About Maude - A High-Performance Logical Framework, How to Specify, Program and Verify Systems in Rewriting Logic. In Clavel, M., Durán, F., Eker, S., Lincoln, P., Martí-Oliet, N., Meseguer, J., Talcott, C.L., eds.: All About Maude. Volume 4350 of Lecture Notes in Computer Science., Springer (2007)
32. Jackson, D.: Alloy: a lightweight object modelling notation. ACM Transactions on Software Engineering and Methodology **11**(2) (2002) 256–290
33. Moura, L.D., Bjørner, N.: Satisfiability modulo theories: introduction and applications. Communications of the ACM **54**(9) (2011) 69–77
34. Beth, E.W.: Semantic entailment and formal derivability. Mededlingen van de Koninklijke Nederlandse Akademie van Wetenschappen, Afdeling Letterkunde **18**(13) (1955) 309–342 Reprinted in [17].
35. Herbrand, J.: Recherches sur la theorie de la demonstration. PhD thesis, Université de Paris (1930) English translation in [18].
36. Gentzen, G.: Untersuchungen tiber das logische schliessen. Mathematische Zeitschrijt **39** (1935) 176–210 and 405–431 English translation in [19].