Advances in Modal Logic, Volume 3

F. Wolter, H. Wansing, M. de Rijke, and
M. Zakharyaschev, eds

CENTER FOR THE STUDY
OF LANGUAGE
AND INFORMATION

Contents

1

Algorithmic Correspondence Theory for Substructural Categorial Logic

MARCELO FINGER

ABSTRACT. Substructural categorial connectives can be treated as modalities. Such binary connectives have a possible worlds semantics based on ternary accessibility relations. This modal treatment allows one to explore *categorial correspondence theory*, in analogy to the usual correspondence theory for modal logics. Its aim is to find a first-order restriction over ternary frames corresponding to a categorial sequent.

This paper proposes an algorithmic method that deals with categorial correspondence theory. It proposes a proof theoretical method based on *SLaKE-tableaux* that produces a second-order formula corresponding to a given categorial sequent. When the SLaKE-tableau is finite, a valuation for the propositional atoms is obtained from the tableau; a first-order formula equivalent to the given sequent over ternary frames is thus computed. For infinite tableaux, no first-order formula is computed.

1 Introduction

Since Kripke (Kripke 1963) has proposed a possible worlds semantics for modal logics, it has been noted that the presence of certain modal axioms impose specific restrictions on the binary accessibility relation of Kripke frames. For example, it is well known that the axiom $\Box p \to p$ is true at any frame $(W, <)$ whose binary accessibility relation is reflexive, $\forall x(x < x)$; conversely, models of any system that has such a formula as a theorem must have a reflexive accessibility relation.

In fact, any modal axiom can be translated to a second-order formula. Some of these second-order formulas are equivalent to first-order

Advances in Modal Logic, Volume 3
F. Wolter, H. Wansing, M. de Rijke, and M. Zakharyaschev, eds.

formulas over Kripke frames. At the heart of *modal correspondence theory* lies the identification of which axioms correspond to some first-order restriction over Kripke frames (van Benthem 1984). A special class of modal axioms, known as the Sahlqvist formulas, is guaranteed to generate first-order restrictions over Kripke frames, and such restrictions can be obtained algorithmically (Sahlqvist 1975).

The work of Routley and Meyer (1973) has shown that relevance logics can be treated as a modal logic with a Kripke-style semantics. This was later extended to other logics in the family of substructural logics (Došen 1993, Restall 2000), and in her PhD thesis, Natasha Kurtonina (1994) presented such semantics for a fragment of substructural logics, known as categorial logics, without any structural pressuposition. In analogy to traditional modal logics, categorial logics have three binary connectives and their semantics is based on a *ternary*[1] accessibility relation $R \subseteq W^3$. The connectives generally found in categorial logics (Moortgat 1997, Carpenter 1997) are here[2] represented as $\bullet$ (called *product*, or *fusion*, or *multiplicative conjunction*), / (*slash* or *right-implication*) and \ (*backslash* or *left-implication*).

Categorial logics are just a fragment of substructural logics. In a sequent presentation of substructural logics, it is also known that the presence of certain structural rules correspond to derivability of certain sequents. For example, any system that allows for the structural rule of associativity derives $A/B \vdash (A/C)/(B/C)$.

With a Kripke-style semantics for categorial logics, a semantic connection between derivable sequents and semantic restrictions could be investigated. The idea of a correspondence theory for substructural logics was proposed by Roorda (1991). Kurtonina (1994) later developed several methods to obtain first-order restrictions over ternary Kripke frames corresponding to categorial formulas; each method deals with a different fragment of categorial logics. One of these methods involved translations of formulas into a suitable fragment; another method was more in the modal logic tradition, generating a second-order formula and a valuation of second-order variables that yields a first-order equivalent formula.

This work pursues further the topic of correspondence theory for categorial logics started by Roorda (1991) and Kurtonina (1994). Our approach is algorithmic, pursuing a unifying method for all categorial formulas. We propose a tableaux-based method, called *SLaKE*[3]*-tableaux*,

[1] In general, an n-ary connective is defined in terms of an $(n+1)$-ary accessibility relation.

[2] These connectives are also found in the literature as $\otimes$, $\rightarrow$and $\leftarrow$.

[3] **S**ubstructural **La**belled **KE**-tableaux

to construct a first-order formula corresponding to a categorial sequent. SLaKE-tableaux are based on KE-tableaux (D'Agostino 1992) over T- and F-signed formulas. Furthermore, to deal with its substructural side, formulas are labelled following the Labelled Deductive System discipline (Gabbay 1996, D'Agostino and Gabbay 1994, Broda et al. 1996a). We show that if a SLaKE-tableau is deterministic and finitely saturates (ie, no branches can be further expanded) a first-order formula corresponding to the input sequent can be computed.

The presentation of this tableau-based method for correspondence theory proceeds as follows. Section 2 provides the background for ternary frames and KE-tableaux. This enables the definition of SLaKE-tableaux in Section 3 and the algorithm for constructing a correspondence formula for categorial sequents. We then show several examples of how to construct such a formula in Section 4, followed by a demonstration of the method's correctness in Section 5. We conclude in Section 6 listing the work that still has to be done in substructural correspondence theory.

2 Background

2.1 Ternary Frames

Ternary frames are relational structures that allow us to view the connectives of (substructural) categorial logic as binary modalities. In our case, we will deal with the fragment containing the usual categorial binary connectives, / (slash), \ (backslash) and $\bullet$ (product). The $\bullet$ connective is assumed to be left-associative, that is, $A \bullet B \bullet C \equiv (A \bullet B) \bullet C$. A sequent calculus presentation for such fragment is shown in Figure 1. The antecedents of sequents are binary trees, where $\Gamma[A, B]$ indicates that (A, B) is a subtree of Γ; antecedents are also left associative, that is, $\Gamma, \Delta, \Sigma \equiv (\Gamma, \Delta), \Sigma$. Consequents of sequents are simple formulas. Note that no structural rules are being admitted a priori, and therefore we are in the *non-associative Lambek Calculus* Lambek 1958.

On the semantic side, we define a *ternary frame* as a pair $\mathfrak{F} = (W, R)$, where W is a non-empty set of *possible worlds* and R is a ternary *accessibility relation*. A *ternary model* $\mathcal{M} = (W, R, V)$ consists of a ternary frame plus a *valuation* $V : \mathcal{P} \to 2^W$, mapping propositional letters to sets of worlds.

The semantic interpretation of categorial formulas in the $\{/, \backslash, \bullet\}$-fragment over ternary frames is:

$$\frac{}{A \vdash A}\ (Axiom)$$

$$\frac{\Gamma[A] \vdash C \quad \Delta \vdash B}{\Gamma[A/B, \Delta] \vdash C}\ (/\ \vdash) \qquad \frac{\Gamma, B \vdash A}{\Gamma \vdash A/B}\ (\vdash\ /)$$

$$\frac{\Gamma[A] \vdash C \quad \Delta \vdash B}{\Gamma[\Delta, B\backslash A] \vdash C}\ (\backslash\ \vdash) \qquad \frac{B, \Gamma \vdash A}{\Gamma \vdash B\backslash A}\ (\vdash\ \backslash)$$

$$\frac{\Gamma[A, B] \vdash C}{\Gamma[A \bullet B] \vdash C}\ (\bullet\ \vdash) \qquad \frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \bullet B}\ (\vdash\ \bullet)$$

FIGURE 1 A sequent calculus presentation of the non-associative Lambek Calculus.

$$\begin{array}{lll}
\mathfrak{F}, V, a \models p & \text{iff} & a \in V(p) \\
\mathfrak{F}, V, a \models A/B & \text{iff} & \forall b \forall c (Rcab \wedge \mathfrak{F}, V, b \models B \Rightarrow \mathfrak{F}, V, c \models A) \\
\mathfrak{F}, V, a \models B\backslash A & \text{iff} & \forall b \forall c (Rcba \wedge \mathfrak{F}, V, b \models B \Rightarrow \mathfrak{F}, V, c \models A) \\
\mathfrak{F}, V, a \models B \bullet A & \text{iff} & \exists b \exists c (Rabc \wedge \mathfrak{F}, V, b \models A \wedge \mathfrak{F}, V, c \models B)
\end{array}$$

We also use the abbreviations:

$$\begin{array}{l}
Rab(cd) =_{def} \exists x (Rabx \wedge Rxcd) \\
Ra(bc)d =_{def} \exists y (Rayd \wedge Rybc)
\end{array}$$

Such abbreviation associates to the left: $Ra(bcd)e =_{def} Ra((bc)d)e$, etc. As usual, we write $\mathfrak{F}, a \models A$ when $\mathfrak{F}, V, a \models A$ for any valuation V; if a is omitted, this means that the condition holds for any possible world.

A sequent $A_1, \ldots A_n \vdash C$ holds at a world $a \in W$ in ternary frame $\mathfrak{F}$ (notation: $\mathfrak{F}, a \models (A_1, \ldots A_n \vdash C)$) iff $\mathfrak{F}, V, a \models A_1 \bullet \ldots \bullet A_n$ implies $\mathfrak{F}, V, a \models C$. This is equivalent to saying that if for some $a_1, \ldots, a_n$, $Ra(a_1 \ldots a_{n-1})a_n$ and, for $1 \leq i \leq n$, $\mathfrak{F}, V, a_i \models A_i$, this implies $\mathfrak{F}, V, a \models C$.

Note that $\mathfrak{F}$ can be seen as a first-order model for a language over R and $P_1, \ldots, P_n$, where each P_i is a predicate symbol corresponding to the propositional letter p_i. The notation $(\forall P)\varphi$ indicates the universal closure of all second-order variables in φ, and $(\exists P)\varphi$ the existential one. Following the modal logic tradition, the *standard translation* of a categorial formula A into second-order logic formula is $(\forall P)\forall a ST_a(A)$, where:

$$
\begin{array}{lcl}
ST_a(p_i) & = & P_i(a) \\
ST_a(A \bullet B) & = & \exists bc(Rabc \wedge ST_b(A) \wedge ST_c(B)) \\
ST_a(B/A) & = & \forall bc(Rcab \wedge ST_b(A) \Rightarrow ST_c(B)) \\
ST_a(A\backslash B) & = & \forall bc(Rcba \wedge ST_b(A) \Rightarrow ST_c(B))
\end{array}
$$

The second-order quantification is over the relevant predicate symbols and reflects all relevant valuations in a frame. So every categorial (modal) sequent $A \vdash C$ corresponds to a second-order formula such that:

$$\mathfrak{F} \models (A \vdash C) \text{ \textit{iff} } \mathfrak{F} \models (\forall P)\forall a\ (ST_a(A) \Rightarrow ST_a(C))$$

The crucial point of correspondence theory is to know when such a second-order formula defines a *first-order* frame property, that is, if there is a first-order formula ψ such that

$$\mathfrak{F} \models (A \vdash C) \text{ \textit{iff} } \mathfrak{F} \models \psi$$

It is the computation of such a property, when it exists, that we investigate next by means of KE- and SLaKE-tableaux.

2.2 KE Tableaux

D'Agostino has shown in (D'Agostino 1992) that analytic tableaux, in the style proposed by Smullyan (Smullyan 1968), cannot polynomially simulate truth tables and in some cases perform exponentially worse than them. To avoid such problems in a principled way, KE-tableaux were introduced.[4]

As usual, KE-tableaux deal with *signed formulas*, where each formula is signed with a T (truth) or an F (falsity). The signed formulas $T\ A$ and $F\ A$ are called *opposite formulas*. A tableau branch that contains a pair of opposite formulas is *closed*. A theorem is proved by refutation, trying to close all branches of the tableau. Each connective has a pair of expansion rules, that decompose a signed formula into smaller signed formulas.

Unlike analytic tableaux, all decomposition rules in a KE-tableaux are linear. Some of the rules are single premised, while others take two premises. For instance, we present here the KE-rules for classical implication ($\Rightarrow$): a two-premised rule for positively signed formulas ($T\ B \Rightarrow A$) with a single conclusion, and one single premised rule for negatively

[4] No one seems to know what KE stands for; apparently K stands for "klassisch", used in analogy to Gentzen's LK system; E may stand for "elimination" (of what?). KE was supposed to be just an initial working name, but somehow it stuck.

signed ones ($F\ B \Rightarrow A$) with a double conclusion:

$$\frac{\begin{array}{c} T\ B \Rightarrow A \\ T\ B \end{array}}{T\ A} \qquad \frac{F\ B \Rightarrow A}{\begin{array}{c} T\ B \\ F\ A \end{array}}$$

Similarly, a pair of linear rules can be given to any of the classical boolean connectives. The only branching rule in a KE-tableau is the *Principle of Bivalence* (PB), that states that a formula can be either true or *not true*[5]:

$$\frac{}{T\ A \qquad F\ A}$$

Usually, the branching formula A is chosen to generate a second premise to some linear rule, so it is always a subformula of some existing formula in the tableau. As a consequence, KE-tableaux branch fewer times than a normal analytic tableau, and it can p-simulate a truth table.

3 Substructural Labelled KE Tableaux

Substructural Labelled KE (SLaKE) tableaux are the main proof theoretical equipment we use in the generation of correspondence formulas for substructural sequents.

The use of KE tableaux for substructural logics have been proposed in (D'Agostino and Gabbay 1994), by attaching a label to the signed formula, as in $T\ A : a$. We use this idea, but without following the labelling discipline developed in (D'Agostino and Gabbay 1994, Broda and Finger 1995, Broda et al. 1996b). Instead, we simply add a new label at each new node of the tableau. Formally, let $\mathcal{L}$ be a countable set of labels, let A be a categorial formula; then for every $a \in \mathcal{L}$, the formulas $T\ A : a$ and $F\ A : a$ are *signed labelled formulas.*

Furthermore, each finite SLaKE-tableau is associated with a *correspondence formula.* The original sequent is associated with an initial tableau and with a second-order formula. Each of the tableau linear expansion rules is associated with an expansion of the correspondence formula of the form $\sharp_i := \varphi(R, A_1, \dots, A_n, \sharp_{i+1})$, where R is the ternary accessibility relation, $A_1, \dots, A_n$ are the formulas generated in the expansion, and $\sharp$ is the "substitution place" for next expansion and can be read simply as *truth.* The tableau rules for SLaKE-tableaux are illustrated in Figure 2.

In each linear rule in Figure 2, the formulas above the horizontal line are the *premises* of the rule, and those below it are the *conclusions* of

[5]Note that this is different from *true or false* — the principle of excluded middle — which is not valid for all substructural logics.

SLaKE Expansion	*Formula Expansion*	*Rxyz*
$\dfrac{T\ A/B : a \qquad T\ B : b}{T\ A : c \text{ (new } c)}$	$\sharp_i := \forall c(Rcab \Rightarrow (ST_c(A) \wedge \sharp_{i+1}))$	$Rcab$
$\dfrac{F\ A/B : a}{T\ B : b \text{ (new } b) \qquad F\ A : c \text{ (new } c)}$	$\sharp_i := \exists b \exists c(Rcab \wedge ST_b(B) \wedge \neg ST_c(A) \wedge \sharp_{i+1})$	$Rcab$
$\dfrac{T\ B\backslash A : a \qquad T\ B : b}{T\ A : c \text{ (new } c)}$	$\sharp_i := \forall c(Rcba \Rightarrow (ST_c(A) \wedge \sharp_{i+1}))$	$Rcba$
$\dfrac{F\ B\backslash A : a}{T\ B : b \text{ (new } b) \qquad F\ A : c \text{ (new } c)}$	$\sharp_i := \exists b \exists c(Rcba \wedge ST_b(B) \wedge \neg ST_c(A) \wedge \sharp_{i+1})$	$Rcba$
$\dfrac{T\ A \bullet B : a}{T\ A : b \text{ (new } b) \qquad T\ B : c \text{ (new } c)}$	$\sharp_i := \exists b \exists c(Rabc \wedge ST_b(A) \wedge ST_c(B) \wedge \sharp_{i+1})$	$Rabc$
$\dfrac{F\ A \bullet B : a \qquad T\ A : b}{F\ B : c \text{ (new } c)}$	$\sharp_i := \forall c(Rabc \Rightarrow (\neg ST_c(B) \wedge \sharp_{i+1}))$	$Rabc$
$\dfrac{}{T\ A : x \qquad F\ A : x}$	$\sharp_i := \forall x((ST_x(A) \wedge \sharp^1_{i+1}) \vee (\neg ST_x(A) \wedge \sharp^2_{i+1}))$	—

FIGURE 2 SLaKE rules

the rule. There are one-premised and two-premised rules, but each rule has exactly one premise that is a *compound* formula, which is called the *main premise*; other premises are called *auxiliary*. Two-premised rules are $\forall$-rules and one-premised rules are $\exists$-rules. If either of the conclusions of an $\exists$-rule is present on the current branch, it is not added again with a new label. $\forall$-rules always generate a new conclusion.

The last rule in Figure 2 is the Principle of Bivalence (PB) branching rule. It is only applied for a formula A following the *branching heuristics*:

> PB is used for a formula A that serves as an auxiliary premise for a $\forall$-rule; PB is only applied in a branch when no other linear expansion is possible.

The main premises that trigger the application of PB for A are: $F\ A \bullet B$, $T\ A\backslash B$ and $T\ B/A$. The corresponding $\forall$-rule will be applicable on the

$T\ A$ branch. This heuristics guarantees that only subformulas of the original sequent will be introduced by PB. It introduces two "substitution places" in the correspondence formula, $\sharp^1_{i+1}$ and $\sharp^2_{i+1}$, one for each new branch. A branch that can still be expanded is called *active*. Each active branch in a SLaKE tableau always has exactly one substitution place.

The importance of substitution places is that they guarantee that each formula introduced in the correspondence formula will "see the correct context", that is, it will be in the scope of the correct quantifiers.

A sequent of the form $A_1, \ldots, A_n \vdash C$ is transformed into the initial SLaKE-tableau:

$$\begin{array}{c} T\ A_1 : a_1 \\ \vdots \\ T\ A_n : a_n \\ F\ C : a \end{array}$$

Since the tableau is a refutation method, this induces the correspondence formula:

$$\begin{array}{rl} \neg\varphi = \neg\exists a a_1 \ldots a_n[& ST_{a_1}(A_1) \wedge \ldots \wedge ST_{a_n}(A_n) \wedge \neg ST_a(C) \\ & \wedge Ra(a_1 \ldots a_{n-1})a_n \wedge \sharp] \end{array}$$

A single premised sequent $A \vdash C$ generates the initial tableau containing $T\ A : a$ and $F\ C : a$, with the initial correspondence formula $\neg\varphi_1(\sharp_1) = \neg\exists a(ST_a(A) \wedge \neg ST_a(C) \wedge \sharp_1)$. We could extend the method for sequents with empty antecedents, but we do not pursue this topic here.

The aim of the SLaKE-tableau construction is *not* to close every tableau branch, but to expand each tableau branch until no more expansions are possible. Each expansion step will also give us a new version of the correspondence formula. If we can finitely expand all tableau branches, a valuation for the atomic formula is constructed, so that we obtain a first-order formula by substituting in the final formula the evaluated values. It is also possible that there will be some infinite branches (something that would not happen in simple propositional tableaux), in which case the above method is not applicable.

A *SLaKE-saturated set* Γ is a set of labelled signed formulas and of $Rxyz$ formulas such that, with respect to the rules of Figure 2:

(a) If the premise of a rule is in Γ, each of its consequence is in Γ for some label, and the $Rxyz$ formula from the $Rxyz$-column is also added to Γ.

(b) For each compound formula in Γ that is a main premise of a $\forall$-rule, there must be in Γ either an auxiliary premise or the opposite of it.

The expansion of a tableau aims at constructing branches that are SLaKE-saturated sets. Item (a) corresponds to normal branch expansion. The fact that the consequence of a $\forall$-rule is always added to Γ with a new label rises the possibility of having infinite SLaKE-saturated sets. Item (b) guides the branching heuristics. At the end of a finite expansion, a correspondence formula will be built from a suitable valuation. Let us show it through examples.

Example 3.1 Consider the sequent $p/q \vdash q\backslash p$. Its associated SLaKE tableau is:

1.	$T\ p/q : a$		
2.	$F\ q\backslash p : a$		$\neg\varphi = \neg\exists a(ST_a(p/q) \wedge \neg ST_a(q\backslash p) \wedge \sharp_1)$
3.	$T\ q : b$	from 2	
4.	$F\ p : c$	from 2	$\sharp_1 := \exists b \exists c(Rcba \wedge Q(b) \wedge \neg P(c) \wedge \sharp_2)$
5.	$T\ p : d$	from 1, 3	$\sharp_2 := \forall d(Rdab \Rightarrow P(d) \wedge \sharp_3)$

Initially, we expand line 2, and simultaneously, using the semantics of $q\backslash p$, we expand the correspondence formula substituting $\sharp_1$ into it. We then use lines 1 and 3 for another expansion, obtaining $\sharp_2$. At this point, the tableau is saturated and closed (remember our main goal is not to close a tableau, but to saturate every branch of it). So we make $\sharp_3 := \top$. We have thus built the second-order formula:

$$\begin{aligned}\neg\varphi = \neg\exists a(\quad & ST_a(p/q) \wedge \neg ST_a(q\backslash p) \wedge \\ & \exists b \exists c(Rcba \wedge Q(b) \wedge \neg P(c) \wedge \forall d(Rdab \Rightarrow P(d))))\end{aligned}$$

The formula $\neg\varphi$ is equivalent to the original sequent, so a suitable valuation has to be constructed to obtain a first-order formula. Such a valuation is implicitly built in the following way. If $T\ r : x$ occurs in a branch, then $P(x)$ must hold iff x is different from all worlds y such that $F\ r : y$ occurs in the same branch above it. Similarly, if $F\ r : x$ occurs in a branch, then $\neg P(x)$ must hold iff x is different from all worlds y such that the $T\ r : y$ occurs in the same branch above it. By applying this rule to the tableau above, we get the implicit *canonical valuations* of P and Q:

$$Q(b) \leftrightarrow \top$$
$$(\neg P(c) \leftrightarrow \top) \wedge (P(d) \leftrightarrow d \neq c)$$

Saturation guarantees that the compound formulas in $\neg\varphi$ can be substituted by $\top$. By substituting each second-order variable in $\neg\varphi$ by their canonical valuation, we obtain the first-order formula

$$\neg\exists a \exists b \exists c(\top \wedge \top \wedge Rcba \wedge \forall d(Rdab \Rightarrow d \neq c))$$

which is equivalent to $\forall a \forall b \forall c(Rcba \Rightarrow Rcab)$, meaning that R must have the *commutativity* property for its second and third positions. It is easy to verify that any model satisfying this property also satisfies the original sequent, and vice-versa. $\square$

We will prove the correctness of this method in Section 5. First, let us present the method in detail.

A valuation V', defined over the set of possible worlds W', is said to *extend* V over the set $W \subset W'$ if the two valuations agree on the truth of all atoms at all worlds in W.

The expansion of a tableau is the stepwise construction of a *counter-model* for the input sequent. Each step generates:

- a valuation V_{i+1} extending V_i over the set of labels in the tableau at step i; and
- a second-order formula $\neg\varphi_{i+1}(\sharp_{i+1})$ that is equivalent to $\neg\varphi_i(\sharp_i)$.

Definition 3.2 Let $\mathcal{X} \in \{T, F\}$; define $\bar{\mathcal{X}}$ such that $\bar{T} = F$ and $\bar{F} = T$. For each SLaKE atomic formula $\mathcal{X}\, p : x$ in the tableau, define:

$$O(\mathcal{X}\, p : x) = \{y \mid \bar{\mathcal{X}}\, p : y \text{ occurs "above" } \mathcal{X}\, p : x\}$$

For every atomic p, define the *canonical valuation* of $P(x)$ implicitly as:

$$\bigwedge_{\mathcal{X} p:x} ((P(x) \leftrightarrow \bigwedge_{y \in O(Tp:x)} x \neq y) \wedge (\neg P(x) \leftrightarrow \bigwedge_{y \in O(Fp:x)} x \neq y))$$

The rationale of the canonical valuation is simply that an atom cannot be both true and false at the same world. This is the *minimal* condition that any model must verify. To obtain a first-order formula we need to substitute the *canonical* valuation into $\neg\varphi$. The canonical valuation is defined such that no scope violation is possible, for the substitution introduces no free variables. If the tableau branches, each sub-branch is developed independently and is not affected the labels introduced at other branches. This has the effect of restricting the scope of quantifiers in the construction of the correspondence formula.

If the tableau saturates, such a valuation satisfies all compound formulas (Lemma 5.3). The first-order formula $\neg\psi$ thus obtained is obviously implied by $\neg\varphi$.

Let $(\exists P)\varphi$ be the second order existential closure of φ. If the tableau saturates in finitely many steps, then a countermodel $\mathfrak{F}^*$ must satisfy $\mathfrak{F}^* \models (\exists P)\varphi$. That is, the class of models that validate the initial sequent does not contain $\mathfrak{F}^*$. If there are only finitely many possible refutations of the initial sequent, we obtain finitely many (say m) second-order formulas $\neg\varphi_j$, one for each refutation, that when substituted by the correspondent canonical valuation each generate $\neg\psi_j$ $(1 \leq j \leq m)$, all implied by $\neg\varphi$.

Furthermore, due to saturation, any possible refutation must satisfy one of the φ_j, so a countermodel to the input sequent satisfies the second-order formula $(\exists P) \bigvee \varphi_j$, which is equivalent to $(\exists P)\varphi$. But, with the constructed counter valuations we have a witness for that existential second-order quantification, so each φ_j implies its correspondent ψ_j. And therefore $\neg\psi_j$ implies $\neg\varphi_j$ and $\bigwedge_{j=1}^{m} \neg\psi_j$ implies (and thus is equivalent to) $\neg\varphi_j$.

As a consequence, if there is a single way to finitely refute a sequent, the first-order formula generated $\neg\psi$ is equivalent to the second-order formula $\neg\varphi$, which is equivalent to the validity of the input sequent. This motivates the following definition.

A SLaKE-tableau is *deterministic* if at every point of its extension there is only one applicable rule. From what has just been explained above, finitely-saturated deterministic tableaux generate a first-order formula equivalent to its input sequent. If the tableau is finite and non-deterministic, one has to generate all the possible expansions, and for each one compute its associated formula $\neg\psi_i$; the first-order formula equivalent to the validity of the input sequent is the conjunction of all such formulas.

We now present the full algorithm for computing the correspondence formula.

Algorithm 3.1
Input: a sequent $A_1, \ldots, A_n \vdash C$.
Output: its first-order correspondence formula, if there are only finite refutations.

1. Initialize the tableau for $T\ A_1 : a_1, \ldots, A_n : a_n$ and $F\ C : a$, with initial correspondence formula $\neg\varphi(\sharp)$.
2. Repeat while the tableau is not saturated nor an infinite branch has been detected.
   (a) If there is an applicable rule, expand the tableau and the correspondence formula according to the rules in Figure 2.
   (b) If there are no linear rules applicable but the tableau is not saturated, choose a complex formula over which to apply the branching rule PB and continue expanding both branches.
3. If the tableau has an infinite branch, stop.
4. Otherwise, a second-order formula $\neg\varphi$ was generated. Construct the canonical valuations for every atom p.

5. If the tableau is deterministic, output the first-order formula $\neg\psi$ obtained by substituting all compound formulas by $\top$ and all $P(x)$ by its canonical valuation at the time of introduction. Otherwise, repeat items 2, 3 and 4 for each of the possible refutations, each generating $\neg\psi_j$; output $\bigwedge \neg\psi_j$.

Next we see a few more examples.

4 Examples

Example 4.1 [**Finite Deterministic Non-Branching Tableau**]

Consider the sequent $p/q \vdash (p/r)/(q/r)$; such sequent is not in the format of the "Sahlqvist-van Benthem" Theorem in (Kurtonina 1994) and had to be dealt with by means of a translation method. A SLaKE-tableau constructed for it looks like:

1.	$T\ p/q : a$		$\neg\varphi = \neg\exists a(ST_a(p/q)\wedge$
2.	$F\ (p/r)/(q/r) : a$		$\neg ST_a((p/r)/(q/r)) \wedge \sharp_1)$
3.	$T\ (q/r) : b$	from 2	$\sharp_1 := \exists b\exists c(Rcab\wedge$
4.	$F\ (p/r) : c$	from 2	$ST_b(q/r) \wedge \neg ST_c(p/r) \wedge \sharp_2)$
5.	$T\ r : d$	from 4	
6.	$F\ p : e$	from 4	$\sharp_2 := \exists d\exists e(Recd \wedge R(d) \wedge \neg P(e) \wedge \sharp_3)$
7.	$T\ q : f$	from 3,5	$\sharp_3 := \forall f(Rfbd \Rightarrow (Q(f) \wedge \sharp_4))$
8.	$T\ p : g$	from 1,7	$\sharp_4 := \forall g(Rgaf \Rightarrow (P(g) \wedge \sharp_5))$

The *canonical valuation* for $P(w)$ in $\neg\varphi$ is:

$$\begin{gathered}(P(g) \leftrightarrow g \neq e) \wedge (\neg P(e) \leftrightarrow \top)\\ Q(f) \leftrightarrow \top\\ R(d) \leftrightarrow \top\end{gathered}$$

By substituting the canonical valuation to $\neg\varphi$, which also makes all compound formulas true, we obtain:

$$\neg\exists a\exists b\exists c(Rcab \wedge \exists d\exists e(Recd \wedge \forall f(Rfbd \Rightarrow \forall g(Rgaf \Rightarrow g \neq e)))$$

which is equivalent to

$$\forall a\forall b\forall d\forall e(\quad Re(ab)d \Rightarrow Rea(bd) \quad).$$

That is, the sequent $A/B \vdash (A/C)/(B/C)$ imposes a restriction of *left-associativity* to the second and third positions of the ternary relation R. Note that it is well know from substructural logics that the sequent $A/B \vdash (A/C)/(B/C)$ is provable whenever the structural rule of associativity is accepted (Došen 1993). This shows a remarkable connection between proof-theoretical properties and ternary frame semantics, which holds for other axioms too.

Example 4.2 [Finite Deterministic Branching Tableau]

Now let us see an example with branching. For that, consider Peirce's Axiom $(p\backslash q)\backslash p \vdash p$. When the tableau is initialized, no linear rules are applicable, so the branching heuristics is applied to $p\backslash q$, corresponding to the valid disjunctive statement $\forall b(ST_b(p\backslash q) \vee \neg ST_b(p\backslash q))$:

$$\begin{array}{c}
1.\quad T\ (p\backslash q)\backslash p : a \\
2.\quad F\ p : a \\
\hline
\begin{array}{cc}
3(i).\quad T\ p\backslash q : b & 3(ii).\quad F\ p\backslash q : b \\
\hline
4(i).\quad T\ p : c & 4(ii).\quad T\ p : e \\
\cline{1-1}
5(i).\quad T\ q : d & 5(ii).\quad F\ q : f
\end{array}
\end{array}$$

which has the corresponding second-order expansion:

$$\begin{array}{c}
\neg\varphi = \neg\exists a(ST_a((p\backslash q)\backslash p) \wedge \neg P(a) \wedge \sharp_1) \\
\hline
\sharp_1 := \forall b((ST_b(p\backslash q) \wedge \sharp_2^i) \vee (\neg ST_b(p\backslash q) \wedge \sharp_2^{ii}))) \\
\hline
\begin{array}{ll}
\sharp_2^i = \forall c(Rcba \Rightarrow (P(c) \wedge \sharp_3^i)) & \\
\cline{1-1}
\sharp_3^i = \forall d(Rdcb \Rightarrow (Q(d) \wedge \sharp_4^i)) & \sharp_2^{ii} = \exists e \exists f(Rfeb \wedge P(e) \wedge \neg Q(f) \wedge \sharp_3^{ii})
\end{array}
\end{array}$$

The branches are independently developed. The left branch will cause substitutions in $\sharp_2^i$ and the right branch will cause substitutions in $\sharp_2^{ii}$. Since the substitution occurs in a negative context, the branching will impose a conjunction of constraints, each of which can be computed separately. New labels could be repeated on both branches, since there will never be a quantifier scope confusion, but for clarity reasons we will always use variables new to the entire tableau.

If we were looking for a closed tableau, we could stop the development of the left branch at $4(i)$; however, our aim here is to obtain a *SLaKE-saturated* set at each branch, so we proceed to obtain $5(i)$. The correspondence formula obtained after both branches are saturated and negation is pushed inside is:

$$\begin{array}{ll}
\forall a \exists b[& \neg(ST_a((p\backslash q)\backslash p) \wedge \neg P(a) \wedge ST_b(p\backslash q) \wedge \\
 & \forall c(Rcba \Rightarrow (P(c) \wedge \forall d(Rdcb \Rightarrow Q(d)))))\ \ ] \wedge \\
\forall a \exists b[& \neg(ST_a((p\backslash q)\backslash p) \wedge \neg P(a) \wedge \neg ST_b(p\backslash q) \wedge \\
 & \exists e \exists f(Rfeb \wedge P(e) \wedge \neg Q(f)))\ \ ]
\end{array}$$

The first conjunct corresponds to the left branch, and the second conjunct corresponds to the right one. Since each branch is developed independently, the canonical valuation is developed as before, that is:

$$(\neg P(a) \leftrightarrow \top) \wedge (P(c) \leftrightarrow c \neq a) \wedge (P(e) \leftrightarrow e \neq a)$$
$$(Q(d) \leftrightarrow \top) \wedge (\neg Q(f) \leftrightarrow \top)$$

The first-order correspondence formula thus obtained is:

$$\begin{array}{c} \forall a \exists b[\neg(\forall c(Rcba \Rightarrow (c \neq a \wedge \forall d(Rdcb \Rightarrow \top)))) \\ \wedge \\ \neg(\exists e \exists f(Rfeb \wedge e \neq a \wedge \top))] \end{array}$$

which is equivalent to

$$\forall a \exists b(Raba \wedge \forall ef(Rfeb \Rightarrow e = a))$$

Kurtonina (Kurtonina 1994) has shown that this condition plus associativity (Example 4.1) and commutativity (Example 3.1) makes the ternary model collapse, that is, they imply $\forall abc(Rabc \Rightarrow a = b = c)$.

Example 4.3 [Finite Non-Deterministic Tableau]
All SLaKE-tableaux seen so far have been *deterministic* in the sense that, at every step, there was a single expansion rule to be applied to a branch. Consider now a sequent for which we have more than one applicable expansion at the initial step:

(1) $$A/B, B, A/C, C \vdash A$$

whose initial expansion has two choices: $T\ A/B : a$ with $T\ B : b$, or $T\ A/C : c$ with $T\ C : d$. If the first pair is chosen first, the correspondence formula obtained is

(2) $$\forall abcde(\ Re(abc)d \Rightarrow (Reab \vee (Recd \wedge \exists fRfab))\ ),$$

and if the second pair is chosen first, a different correspondence formula is obtained:

(3) $$\forall abcde(\ Re(abc)d \Rightarrow (Recd \vee (Reab \wedge \exists gRgcd))\ ).$$

Since both formulas constrain the frame, we take the *conjunction* of (2) and (3), thus obtaining

(4) $$\forall abcde(\ Re(abc)d \Rightarrow (Reab \vee Recd)\ ).$$

That is, with non-deterministic SLaKE-tableaux, we have to take the conjunction of the correspondence formulas generated by all possible tableaux. Since we have exhausted *all possible refutations*, this means that a frame validates (1) iff it satisfies (4).

4.1 Infinite Branches

It is not always the case that a tableau branch can be finitely saturated. In those cases we cannot apply the method above, so we do not get a first-order formula. We present next two categorial formulas that generate infinite tableaux. It is known from (Kurtonina 1994) that these formulas have no correspondent first-order formula, so it is quite reassuring that our method does not produce one.

For example, consider the tableau for $A/A \vdash A\backslash A$:

$$
\begin{array}{lll}
1. & T\ A/A : a & \\
2. & F\ A\backslash A : a & \\
\hline
3. & T\ A : b & \text{from } 2 \\
4. & F\ A : c & \text{from } 2 \\
5. & T\ A : d & \text{from } 1 \text{ and } 3 \\
6. & T\ A : e & \text{from } 1 \text{ and } 5 \\
7. & T\ A : f & \text{from } 1 \text{ and } 6 \\
 & \vdots &
\end{array}
$$

Another example is $(A\backslash A)\backslash A \vdash A$:

$$
\begin{array}{ll}
1. & T\ (A\backslash A)\backslash A : a \\
2. & F\ A : a
\end{array}
$$

$$
\begin{array}{ll|ll}
3i. & T\ A\backslash A : b & 3ii. & F\ A\backslash A : b \\
4i. & T\ A : c & 4ii. & T\ A : e \\
5i. & T\ A : d \text{ from } 3i \text{ and } 4i & 5i. & F\ A : f \\
6i. & T\ A : e \text{ from } 3i \text{ and } 5i & & \\
 & \vdots & &
\end{array}
$$

Note that the infinite branches in the tableau above are characterized by cycles of T-signed formulas $T\ A/A$ or $T\ A\backslash A$ and $T\ A$. Such cycles can be of larger size and involve more complex formulas, but as long as a single element of the cycle is derived a second time (with a new label), an infinite branch is guaranteed, for this element will then be regenerated infinitely often.

5 Correctness of the Method

We now prove the correctness of the method developed in this work. We first note the following property of SLaKE expansions.

Lemma 5.1 *Suppose a branch expansion step has transformed* $\neg\varphi_n(\sharp_n)$ *into* $\neg\varphi_{n+1}(\sharp_{n+1})$. *Then* $\neg\varphi_n(\sharp_n)$ *and* $\neg\varphi_{n+1}(\sharp_{n+1})$ *are logically equiva-*

lent over the class of all frames.

Proof. Clearly, $\varphi_{n+1}(\sharp_{n+1}) \models \varphi_n(\sharp_n)$ over the class of all frames. For the other direction, consider first linear expansions. Note that each $\sharp$-expansion in Figure 2 is a logical consequence of the semantic definition of the rule premises. Since the semantic definition of the premises always occur positively in $\varphi_n(\sharp_n)$, it is the case that $\varphi_n(\sharp_n) \models \varphi_{n+1}(\sharp_{n+1})$. Also note that the expansion for PB is logically valid.

So $\varphi_{n+1}(\sharp_{n+1})$ and $\varphi_n(\sharp_n)$ are equivalent over the class of all frames, and so is their negation. $\dashv$

As a consequence, a simple induction on the expansion of $\neg\varphi$ gives us the following.

Lemma 5.2 *If $\neg\varphi$ is the second-order formula computed by the SLaKE-expansion of $A_1, \ldots, A_n \vdash C$, then for every ternary frame $\mathfrak{F}$:*

$$\mathfrak{F} \models A_1, \ldots, A_n \vdash C \text{ iff } \mathfrak{F} \models \neg\varphi$$

So $\neg\varphi$ is equivalent to the input sequent. Any valuation of the second-order variables in $\neg\varphi$ will generate a first-order formula implied by the original sequent. All we need now is to show that the canonical valuation of Definition 3.2 really satisfies all formulas in a branch.

For that, we consider a frame $\mathfrak{F}$, a SLaKE-saturated set Γ and a canonical valuation built on Γ. Consider a mapping of the labels in Γ into the worlds of $\mathfrak{F}$ that respects the restrictions of the canonical valuation (ie the labels that V forces to be different are not mapped into the same world); we say that $\mathfrak{F}, V \models T\ A : a$ iff $\mathfrak{F}, V, a \models A$ and $\mathfrak{F}, V \models F\ A : a$ iff $\mathfrak{F}, V, a \not\models A$.

Lemma 5.3 *Let Γ be a SLaKE saturated set and let $\mathfrak{F}$ be a frame that satisfies all the R-formulas in Γ. Let V be the canonical valuation built from Γ. Then for all signed formula $S \in \Gamma$, $\mathfrak{F}, V \models S$.*

Proof. We prove the lemma by induction on the structure of the formulas in Γ. The canonical valuation satisfies all the atomic formulas, taking care only that no atomic formula is true and false in the same world.

For the compound formulas, the single premises of $\exists$-rules replace a compound formula by its definition, so if the conclusions are satisfied, so is the compound premise. For the main premise of a $\forall$-rule, saturation guarantees that every possible conclusion is added to Γ, so if all such conclusions are satisfied so is the main premise. As a consequence, the frame $\mathfrak{F}$ also satisfies the compound main premise. $\dashv$

By putting together Lemmas 5.2 and 5.3 we obtain the following.

Theorem 5.4 *If $\mathcal{T}$ is a finite saturated SLaKE-tableau for the sequent $A_1, \ldots, A_n \vdash C$ computing a first-order formula $\neg\psi$ then*

$$\mathfrak{F} \models (A_1, \ldots, A_n \vdash C) \Rightarrow \mathfrak{F} \models \neg\psi.$$

For the converse, we first consider deterministic tableaux.

Theorem 5.5 *Let $\mathcal{T}$ be a finite, saturated and deterministic SLaKE-tableau for sequent $A_1, \ldots, A_n \vdash C$ computing a first-order formula $\neg\psi$. Then*

$$\mathfrak{F} \models (A_1, \ldots, A_n \vdash C) \Leftarrow \mathfrak{F} \models \neg\psi.$$

Proof. The construction of $\mathcal{T}$ is a refutation for the input sequent, such that every $\mathfrak{F}$ with $\mathfrak{F} \not\models (A_1, \ldots A_n \vdash C)$ must satisfy the existential closure of the second-order formula φ, that is, for every frame $\mathfrak{F}$ refuting the original sequent, $\mathfrak{F} \models (\exists P)\varphi$.

Since $\mathcal{T}$ is deterministic and saturated, it is associated to a SLaKE saturated set Γ. Any refutation frame $\mathfrak{F}$ must satisfy all the atomic formulas and the R-formulas in Γ. The canonical valuation built from it is such that it satisfies all the atomic formulas.

By Lemma 5.3, $\mathfrak{F}$ with the canonical valuation satisfies all the formulas in $\mathcal{T}$, that is $\mathfrak{F}, V \models \varphi$. So when we apply the canonical valuation to φ obtaining ψ, it follows that:

$$\mathfrak{F} \not\models (A_1, \ldots, A_n \vdash C) \Rightarrow \mathfrak{F} \models \psi,$$

which is equivalent to the desired result. $\dashv$

To eliminate the restriction on deterministic tableaux, we have to consider all possible refutations. This yields the final result.

Theorem 5.6 *Given a sequent $A_1, \ldots, A_n \vdash C$ and all possible finite SLaKE-tableaux refuting it, $\mathcal{T}_1, \ldots, \mathcal{T}_m$ generating, respectively, first-order formulas $\neg\psi_1, \ldots, \neg\psi_m$. Then*

$$\mathfrak{F} \models (A_1, \ldots, A_n \vdash C) \Longleftrightarrow \mathfrak{F} \models \bigwedge_{j=1}^{m} \neg\psi_j.$$

Proof. ($\Rightarrow$) This is Theorem 5.4.

($\Leftarrow$) Suppose there is a frame $\mathfrak{F}$ that refutes the input sequent, such that there is a valuation V and a point a such that $\mathfrak{F}, V, a \not\models (A_1, \ldots, A_n \vdash C)$. The tableaux $\mathcal{T}_1, \ldots, \mathcal{T}_m$ are all the possible refutations of the input sequent, when we consider the second-order formulas $\varphi_1, \ldots, \varphi_m$ generated by each refutation, we get $\mathfrak{F}, a \models \bigvee_{j=1}^{m} (\exists P)\varphi_j$.

Without loss of generality, suppose $\mathfrak{F}$ satisfies one such disjunct, φ_j: $\mathfrak{F} \models (\exists P)\varphi_j$. We claim that there is a *finite* valuation V_f that satisfies φ_j built in the following way:

$$V_f(p) = \{a | T\ p : a \text{ belongs to } \mathcal{T}_j\}$$

It is clear that $V_f(p)$ is finite for each p, for there are only finitely many formulas in T_j.

By induction on the construction of $\mathcal{T}_j$, we show that the satisfaction of φ_j depends on finitely many possible worlds. The base case is the initial tableau, which clearly relies on finitely many possible worlds.

If the expansion of $\mathcal{T}_j$ was done by a single-premised $\exists$-rule, then only two new possible worlds have to be considered, and by induction hypothesis the number of worlds remains finite.

If the expansion of $\mathcal{T}_j$ was done by PB, then no new world needs to be considered (and one extra variable is introduced), so the induction hypothesis guarantees that the number of worlds remains finite.

If the expansion of $\mathcal{T}_j$ was done by a double-premised $\forall$-rule, we need to examine each case, so consider, for example, a main formula of the form $T\ A/B : a$; to satisfy such formula, we need to enforce that:

$$\forall bc(Rcab \wedge T\ B : b \Rightarrow T\ A : c)$$

There may be infinitely many valid formulas of the form $Rcab$, but by induction hypothesis, only finitely many worlds b where the truth of B has been enforced, so the truth of A will be enforced only at finitely many worlds c. [6] The proof is analogous for the other main premises of $\forall$-rules. We have proved our claim.

At the end of the tableau construction, all atomic formulas in the tableau are satisfied by V_f. We have thus shown a *finite model property* for the sequents that have a finitely saturated tableaux. This means that if we have opposite signed formulas of the form $T\ p : x$ and $F\ p : y$, we only have to impose $x \neq y$ for finitely many points. Such a procedure gives us the same substitution as the canonical valuation; it follows that $\mathfrak{F}, V_f, a \models \psi_j$.

We have thus shown:

$$\mathfrak{F} \not\models (A_1, \ldots, A_n \vdash C) \Rightarrow \mathfrak{F} \models \bigvee_{j=1}^{m} \psi_j,$$

which is equivalent to the desired result. $\dashv$

[6] Note that such an argument fails if we had in our language a formula that is true at all worlds; luckily, we do not, not even A/A, as can be verified by its semantics.

6 Conclusion and Further Work

We have proposed an original method to construct a first-order correspondence formula for substructural categorial formulas. This construction is based on the novel SLaKE-tableaux. Our method unifies under an algorithmic approach the generation of first-order correspondence formulas initially treated by (Kurtonina 1994).

We have the following conjecture:

Conjecture 6.1 *A sequent has a first-order equivalent iff it has a finite SLaKE-tableaux.*

The *if*-part has already been proved, but for the *only if*-part all we can say for the moment is that our method does not apply to infinite tableaux. To prove the conjecture, we have to show that it is not possible to have a sequent with an infinite SLaKE-tableau but with a first-order equivalent.

Acknowledgements

The author would like to thank Wamberto Vasconcelos and Angela Weiss for helpful comments on the paper. This work was partly supported by the Brazilian Research Council (CNPq), grant PQ 300597/95-5.

References

Broda, K., and M. Finger. 1995. KE-tableaux for a Fragment of Linear Logic. In *Proceedings of the 4th International Workshop on Analytic Tableaux and Related Methods*. Koblenz, May.

Broda, K., M. Finger, and A. Russo. 1996a. LDS-Natural Deduction for Substructural Logics. In *3rd Workshop on Logic, Language, Information and Computation (WoLLIC96)*, 15–18. Abstract only.

Broda, Krysia, Marcello D'Agostino, and Alessandra Russo. 1996b. Transformation methods in LDS. In *Logic, Language and Reasoning. An Essay in Honor of Dav Gabbay*. Kluwer Academic Publishers.

Carpenter, Bob. 1997. *Type-Logical Semantics*. MIT Press.

D'Agostino, M., and D. Gabbay. 1994. A Generalization of Analytic Deduction via Labelled Tableaux, part I: Basic Substructural Logics. *Journal of Automated Reasoning* 13:243–281.

D'Agostino, Marcello. 1992. Are Tableaux an Improvement on Truth-Tables? — Cut-free Proofs and Bivalence. *Journal of Logic, Language and Information* 1:235–252.

Došen, K. 1993. A Historical Introduction to Substructural Logics. In *Substructural Logics*, ed. P. Schroeder-Heister and K. Došen. 1–31. Oxford University Press.

Gabbay, Dov M. 1996. *Labelled Deductive Systems*. Oxford University Press.

Kripke, Saul. 1963. Semantical Analysis of Modal Logic I. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*. 9(67–96).

Kurtonina, Natasha. 1994. *Frames and Labels — A Modal Analysis of Categorial Inference*. Doctoral dissertation, Research Institute for Language and Speech (OTS), Utrecht, and Institute of Logic, Language and Information (ILLC), Amsterdam.

Lambek, J. 1958. The Mathematics of Sentence Structure. *American Mathematical Monthly* 65:154–169.

Moortgat, M. 1997. Categorial Type Logics. In *Handbook of Logic and Language*, ed. J. Van Benthem and A. ter Meulen. 93–178. Elsevier North-Holland/The MIT Press.

Restall, G. 2000. *An Introduction to Substructural Logics*. Routledge.

Roorda, Dirk. 1991. *Resource Logics: Proof-theoretical Investigations*. Doctoral dissertation, Institute of Logic, Language and Information (ILLC), University of Amsterdam.

Routley, Richard, and Robert K. Meyer. 1973. The Semantics of Entailment. In *Truth, Syntax and Modality*, ed. Hugues Leblanc. Studies in Logic and the Foundations of Mathematics, Vol. 68, 199–243. North-Holland.

Sahlqvist, H.İ975. Completeness and Correspondence in the First- and Second-order Semantics for Modal Logic. In *Proceedings of the 3rd Scandinavian Logic Symposium*.

Smullyan, Raymond M. 1968. *First-Order Logic*. Springer-Verlag.

van Benthem, J. 1984. Correspondence Theory. In *Handbook of Philosophical Logic*. 167–248. D. Reidel Publishing Company.

Marcelo Finger
Department of Computer Science
Institute of Mathematics and Statistics
University of São Paulo
Rua do Matão, 1010
05508-900, São Paulo, SP
Brazil
E-mail: mfinger@ime.usp.br
URL: http://www.ime.usp.br/~mfinger

Index