# Relational tableaux for tree models, language models and information networks


Wendy MacCaull*
Department of Mathematics, Computing and Information Systems,
St.Francis Xavier University,
PO Box 5000, Antigonish, NS, B2G 2W5, CANADA
and, currently on sabbatical at
Department of Computer Science,
215 Lindley Hall,
Indiana University,
Bloomington, IN, 47405, USA
e-mail: wmaccaul@cs.indiana.edu; Telephone:(812)855-3668


Preliminary Version of Work in Progress

18 December, 1997


*Supported by the Natural Sciences and Engineering Research Council of Canada and by the St. Francis Xavier University Council for Research

**Relational tableaux for tree models, language models and information networks**

**Abstract**

A proof calculus is given for formalisms for tree models and language models, which rests on an Orlowska-style relational semantics and a Rasiowa/Sikorski-style deduction method for the Lambek calculus. The extension of this method to Barwise's logics for information, which formalizes the flow of information through an information network, is outlined. The relationship of the Orlowska-stlye relational semantics to other relational semantics (such as the arrow logics of van Benthem/Venema and Vakarelov, and the gaggle theory of Dunn) is clarified. Some comparisons to Gabbay's labelled deduction method are made. This paper is part of the general program to find effective, modular proof systems for nonclassical logics, based on relational semantics.

**Keywords:** Lambek calculus, proof theory, relational semantics, tableaux, categorial grammar, substructural logics

# 1 Introduction

Lambek calculus, and its extensions via the addition of various structural rules and modal operators are currently of interest to several diverse communities. Resource-sensitive or substructural logics in general are under intensive investigation, because they allow one to pay attention to properties of a proof (see [D'A&G], [G] and [Ro]). By thinking of premises as resources, and by adding or deleting different structural rules, we can make a logic dependent on (among other things) which resources are used, how many times each resource is used and the order in which the resources are used. (Associative) Lambek calculus has long been the predominant formalism for categorial grammars (see, for example, [Bu], [L-F] and [Ro]). Recently Pentus (see [P]) showed that associative Lambek calculus is not only sound but also complete with respect to the interpretation in language models (without the empty word). Associative Lambek calculus is sound (but not complete) with respect to language models with the empty word. The most basic substructural logic, nonassociative Lambek calculus, has recently captured the attention of the linguistics community, because it provides a formalism to describe tree models, thus giving a categorial framework for headedness and discontinuous constituency (see, for example, [Mo], [Mo&Oe] and [Ve1]). Nonassociative Lambek calculus is sound with respect to the class of tree frames, and complete with respect to a superclass of it. Yet another Lambek-like calculus is Barwise's logic of information flow (see [BGH]). This calculus models perfect reasoning about the flow of information through an information network. It has a great number of motivating examples including some of those considered here. Dunn and Meyer (see [Du&M]) have recently shown how combinatory logic may be viewed as a substructural logic.

Finding an effective deductive apparatus for each of these logics is another question. With the proliferation of logical formalisms modeling a huge diversity of phenomena, it is becoming increasingly important to build a spectrum of logical systems for which proof theoretic problems may be approached in a modular fashion. Then minor modifications to a base logic would allow one to deal with each logic in the spectrum. Relational logic in the sense of Orlowska [Or2], which uses a deduction method developed by Rasiowa and Sikorski (see [R&S]), is one such approach. Sound and constructively complete tableaux-style deductive methods have been developed for many nonclassical logics. (By constructive completeness, we mean that if a branch of a tableau does not close, we can build a countermodel for the formula under consideration.) With the exception of [Or3], where ternary relations were used, the relational logics have been based on a semantics where in the formulas are interpreted as binary relations. In [MacC1] we developed a tableaux-style theorem prover for a $FL$, the full (associative) Lambek calculus (a calculus with constants $0, 1, \top$ and $\bot$, and connectives $\backslash, /$ and $\circ$, (denoted in [MacC1] by $\supset, \supset'$ and $\bullet$) $\vee$, and $\wedge$. When the algorithm terminates with all branches closed, the formula is a theorem of the logic; when it terminates with an open branch, information on the open branch of the tableau allows us to construct a countermodel, so the formula is not a theorem of the logic. In [MacC2] we showed how to extend relational deduction to variety of intuitionistic substructural logics. By adding one (or a combination of) condition(s) we can add in one (or more) of the structural properties of exchange, weakening, contraction and expansion. By adding new accessibility relations, subject to a set of rules (relational analogues of those in [O]) we extended $FL$ to deal with Linear Logic with the modal operators ! and ?. $FL$ can be further extended by adding an operator to deal with involution, and thence the logic of quantales with involution, as discussed in [Mu&P], can be dealt with. The last logic is of importance in describing the specification of programs in the case where the behaviour of the machine is quantized.

The strategy is to begin with Kripke-style semantics for each logic under consideration and from this develop relational semantics for the logics. Basic motivation for this approach may be found in [Or2] and [Or3]. The deduction method which accompanies the relational semantics is a generalization of the Rasiowa /Sikorski-style deduction method for classical logic, which yields tableaux-style proofs (that is, proofs are trees, called decomposition trees). In relational semantics, formulas are interpreted as relations; interpretations as ternary relations are used for substructural logics. Each connective gives rise to two deduction rules, called *decomposition rules.* Other deduction rules called *specific rules* are required, depending on the particular substructural logic under consideration.

In this paper we are interested in dealing with formalisms which are weaker than $FL$. We look first at a calculus for tree models and suggest that this approach may answer the question posed in [Ve1]: *Can we find a nice calculus recursively enumerating all sequents* $\Gamma$ *for which* $T_f \models \Gamma$*?* Then we give relational deductive systems for language models (both with and without the empty word). Constructive completeness holds for the tableaux-system for language models without the empty word. Next, we outline a relational calculus for Barwise's logic of information flow (see [BGH]), which must be distinguished from Orlowska's information logics (see [D&Or]), which are modal formalisms both for the representation of and the reasoning about concepts derived from data that describe an application domain. Sections two, three and four are technical results which hopefully will convince the reader of the applicability of the relational deductive method to a wide class of logical formalisms all based on Lambek calculus and with a variety of applications. If details are too sketchy, the reader will find full details of this method applied to other substructural logics in [MacC2] and [Or3]. Our references will point the reader to the applicability of relational tableaux to a great number of other nonclassical formalisms. In the final section, we make a number of comments and comparisons about relational methods in general as pertaining to the whole class of substructural logics. We strive to clarify the connections among various so-called relational semantics. We discuss the connection of relational logic to the arrow logics of van Benthem/Venema, and Vakarelov, addressing questions in [Ve1] and [BGH]. We compare relational semantics and Dunn's gaggle theory (see [Du]) and comment briefly on combinatory logic (see [Du&M]). We comment briefly on Gabbay's labelled deduction system, denoted LDS, (see [D'A&G] and [G]) and the hybrid approach of Basin et al. (see [BaMV]), wherein labelled deduction is combined with a logical frameworks approach based on natural deduction. We outline a number of unsolved problems and further directions for research.

## 2 A calculus for tree models

We begin with some definitions from [Ve1]:

**Definition 1** *Given a set $Pr$ of primitive types, the set $Tp(Pr)$ of types is found by closing $Pr$ under the binary connectives $\circ$, $\backslash$, and $/$. The set of terms is the closure of $Ty(Pr)$ under the structural connective $(.,.)$. A sequent is an expression of the form $X \rightarrow A$, where $X$ is a term and $A$ is a type.*

Consider now a set $L$ of elements called **leaves**. $Tree(L)$, the set of (ordered) **trees over** $L$, is defined as follows: any leaf is a tree and if $s$ and $t$ are trees, then so is $(st)$. A **finite-tree model** is a pair M= $((Tree(L), V)$, where $V$ is an interpretation mapping basic types to subsets of $Tree(L)$. $V$ can be extended to types and terms as follows:

$$V(A \circ B) = \{(st) \in Tree(L) \mid s \in V(A) \text{ and } t \in V(B)\};$$
$$V(A \backslash B) = \{s \in Tree(L) \mid (ts) \in V(B) \text{ for all trees } t \text{ with } t \in V(A)\};$$
$$V(A/B) = \{s \in Tree(L) \mid (st) \in V(A) \text{ for all trees } t \text{ with } t \in V(B)\};$$
$$V(X,Y) = \{(st) \in Tree(L) \mid s \in V(X) \text{ and } t \in V(Y)\}.$$

Denote $s \in V(A)$ by M,$s \models A$, or, more simply, by $s \models A$. A sequent $X \rightarrow A$ holds in a model M iff $V(X) \subseteq V(A)$; it is valid in the class of finite tree frames, denoted $T_f \models X \rightarrow A$, iff it holds in every finite tree-model. (Finite tree-models are known by a variety of names, including free groupoids and bracketed strings.)

The question posed in [Ve1] is:

*can we find a nice calculus recursively enumerating all sequences $\Gamma$ such that $T_f \models \Gamma$?*

To answer this question, Venema first looked to nonassociative Lambek calculus, $NL$. This is the most basic of the substructural logics. In the following formulation of the derivation rules, $X[B]$ denotes a term $X$ with a distinguished occurrance of the type $B$; when used in the same rule, $X[Y]$ denotes the term $X$ with $Y$ substituted for the distinguished occurrance of the type $B$.

**Definition 2** *The sequent derivation system of $NL$ is given by the following rules:*

*(Id)* $\dfrac{}{A \rightarrow A}$ *(Cut)* $\dfrac{X \rightarrow A \quad Y[A] \rightarrow B}{Y[X] \rightarrow B}$

$(NL\circ r)$ $\frac{X\to A \quad Y\to B}{(X,Y)\to A\circ B}$ $(NL\circ l)$ $\frac{X[A,B]\to C}{X[A\circ B]\to C}$

$(NL\backslash r)$ $\frac{(A,X)\to B}{X\to A\backslash B}$ $(NL\backslash l)$ $\frac{X[B]\to C \quad Y\to A}{X[Y,A\backslash B]\to C}$

$(NL/r)$ $\frac{(X,A)\to B}{X\to B/A}$ $(NL/l)$ $\frac{X[B]\to C \quad Y\to A}{X[B/A,Y]\to C}$

The following semantics gives completeness results for $NL$ and is the basis for further investigations for tree models.

**Definition 3** *A* **(relational) frame** *is a pair $F=(W,R)$ where $W$ is a set and $R$ is a* **ternary accessibility relation** *on $W$. A* **(relational) model** *is a pair $M=(F,V)$, where $F$ is a frame and $V$ is an* **interpretation***: $V : Pr\to P(W)$ (where $P(W)$ denotes the powerset of $W$). For any $x\in W$, say $A$ is true at $x$, and write $x\models A$ iff $x\in V(A)$.* **Truth** *of types and terms is defined as follows:*

$$x\models A\circ B \text{ iff } \exists y,z \text{ such that } Rxyz \text{ and } y\models A \text{ and } z\models B;$$
$$x\models A\backslash B \text{ iff } \forall y,z(Ryzx \text{ and } z\models A \text{ implies } y\models B);$$
$$x\models A/B \text{ iff } \forall y,z(Ryxz \text{ and } z\models B \text{ implies } y\models A);$$
$$x\models (Y,Z) \text{ iff } \exists y,z \text{ such that } Rxyz \text{ and } y\models Y \text{ and } z\models Z.$$

*A sequent $X\to A$ holds in an RF-model $M$ iff for all $x$, if $x\models X$ then $x\models A$. A sequent $X\to A$ is valid if it holds in all RF-models.*

The class of relational models is denoted by $RF$-models.

**Theorem 1** *A sequent $X\to A$ is provable in $NL$ iff it is valid (in the class of RF-models).*

By putting $Rstu$ iff $s=(tu)$ we see that $NL$ is sound with respect to $T_f$ and complete with respect to a superclass of it.

Venema [Ve1] investigated some of the sequents that are valid in tree frames but are not valid in $NL$, and formulated a labelled deduction system for $T_f$. While this deductive system is sound (and complete with respect to infinite tree frames), Venema suggested that the introduction of labels in a categorial grammar causes as many problems as it solves. Here we follow his recommendation to bridge the gap betweem $RF$ and $T_f$. We suggest relational semantics and the associated relational tableaux may give the desired calculus. This strategy has been successful in giving a sound and constructively complete deduction method for a whole spectrum of substructural logics (see [MacC1] and [MacC2]).

We first define relational deduction for $NL$. The idea behind relational deduction is to interpret all types and terms as ternary relations. We begin by defining the logic $RNL$. The language of $RNL$ consists of variables $x,y,z,t,u$, etc., a set of constants $C$, a distinguished constant $R$, binary connectives $\circ,\backslash$ and $/$, the structural connective $(.,.)$, and the unary connective $\neg$. The set of relational expressions consists of the constants in $C$, together with the closure of $C$ under the the binary connectives $\circ,\backslash$ and $/$, (call these relational types), the closure of the resulting set under $(.,.)$ (call these relational terms) and the closure of that set under $\neg$. Formulas are of the form $Extu, Rxtu$ or $\neg Rxtu$ where $E$ is a relational expression. An $RNL$-sequent is an expression of the form $X \to A$, where $X$ is a relational term and $A$ is a relational type. (Clearly $NL$- and $RNL$-sequents are of exactly the same form.)

In order to understand the relational deduction we look at the intended models. We need relational operations to correspond to the logical connectives, so that each formula is interpreted as a ternary relation.

**Definition 4** *Let $W$ be a nonempty set, and let $R$ be a ternary relation on $W$, and let $A$, $B$, $X$ and $Y$ be ternary relations. Define relational operations on the family of ternary relations on $W$ as follows (we write $Rxyz$, ($\neg Rxyz$, $Axtz$ and, respectively, $\neg Axyz$), to denote $(x,y,z) \in R$, ($(x,y,z) \notin R$, $(x,y,z) \in A$ and, respectively, $(x,y,z) \notin A$)):*

$$A \circ B = \{(x,u,v) \mid \exists y,z(\ Rxyz \text{ and } Ayuv \text{ and } Bzuv)\};$$
$$A \backslash B = \{(x,u,v) \mid \forall y,z(Ryzx \text{ and } Azuv \text{ imply } Byuv)\};$$
$$A/B = \{(x,u,v) \mid \forall y,z(Ryxz \text{ and } Bzuv \text{ imply } Ayuv)\};$$
$$\neg A = \{(x,y,z) \mid \neg Axyz)\};$$
$$(Y,Z) = \{(x,u,v) \mid \exists y,z(Rxyz \text{ and } Yyuv \text{ and } Zzuv)\}.$$

**Definition 5** *An $RNL$-model is a 3-tuple $M = (W,R,m)$ where $W$ is a nonempty set, $R$ is a distinguished ternary relation on $W$ and $m$ is an interpretation that assigns ternary relations on $W$ to relational expressions in such a way that for all constants $P$, $m(P) = S \times W \times W$ (where $S \subseteq W$) and for other types and terms the interpretation follows as expected ($m(A \circ B) = m(A) \circ m(B)$, etc.). An $RNL$-sequent $X \to A$ holds in an $RNL$-model iff $m(X) \subseteq m(A)$; that is, for all $(x,t,u)$, either $(x,t,u) \notin m(X)$ or $(x,t,u) \in m(A)$. A $RNL$-sequent $X \to A$ is valid in the class of $RNL$-models iff it holds in all $RNL$-models.*

We shall present deduction rules using a shorthand notation: $\Gamma$ denotes a formula, $J$ and $L$ represent finite sequences of formulas and we write: $\frac{\Gamma_1}{\Gamma_2}$ to represent $\frac{J,\Gamma_1,L}{J,\Gamma_2,L}$, and we write: $\frac{\Gamma_1}{\Gamma_{21} \quad \Gamma_{22} \quad ... \ \Gamma_{2k}}$ to represent the $k$-fold branching rule (the node above the line gives rise to $k$ nodes below the line) $\frac{J,\Gamma_1,L}{J,\Gamma_{21},L \quad J,\Gamma_{22},L \quad ... \ J,\Gamma_{2k},L}$.

**Definition 6** *A* **sequence of formulas is valid** *if at least one of the formulas in the sequence is valid. A rule of the form $\frac{K}{H}$ is said to be* **admissible** *if the sequence $K$ is valid if and only if the sequence $H$ is valid. A rule of the form $\frac{K}{H_1 \quad H_2}$ is said to be* **admissible** *if the sequence $K$ is valid if and only if both the sequences $H_1$ and $H_2$ are valid. A similar definition follows for admissibility of a rule of the form $\frac{K}{H_1 \quad H_2 \quad ... \quad H_j}$.*

The admissibility of the following decomposition rules is clear from Definition 4.

**Decomposition Rules**

$$(\circ)^* \ \frac{(A \circ B)xtu}{Rxyz, (A \circ B)xtu \quad Aytu, (A \circ B)xtu \quad Bztu, (A \circ B)xtu} \qquad (\neg\circ) \ \frac{\neg(A \circ B)xtu}{\neg Rxyz, \neg Aytu, \neg Bztu}$$

$(\backslash)\ \frac{(A\backslash B)xtu}{\neg Ryzx, \neg Aztu, Bytu}$ $\qquad$ $(\neg\backslash)^*\ \frac{\neg(A\backslash B)xtu}{Ryzx, \neg(A\backslash B)xtu \quad Aztu, \neg(A\backslash B)xtu \quad \neg Bytu, \neg(A\backslash B)xtu}$

$(/)\ \frac{(A/B)xtu}{\neg Ryxz, \neg Bztu, Aytu}$ $\qquad$ $(\neg/)^*\ \frac{\neg(A/B)xtu}{Ryxz, \neg(A/B)xtu \quad Bztu, \neg(A/B)xtu \quad \neg Aytu, \neg(A/B)xtu}$

$((Y,Z))^*\ \frac{(Y,Z)xtu}{Rxyz, (Y,Z)xtu \quad Aytu, (Y,Z)xtu \quad Bztu, (Y,Z)xtu}$ $\qquad$ $(\neg(Y,Z))\ \frac{\neg(Y,Z)xtu}{\neg Rxyz, \neg Yytu, \neg Zztu}$

**Specific Rule**

$\frac{Axtu}{Axyz}$

Rules superscripted with * will be called tagged rules. Tagged rules may need to be applied more than once, as explained below in Definition 8. The role of the specific rule is clarified in Section 5. In the specific rule and the tagged rules, the $y$ and $z$ are arbitrary variables. In the other decomposition rules, the $y$ and $z$ are variables not occurring in formulas above the line.

**Definition 7** *A sequence of formulas is said to be* **fundamental** *if it contains formulas* $Axtu, \neg Axtu$ *for any formula* $Axtu$. *It is easy to see that fundamental sequences are valid. A* **decomposition tree** *for an RNL-sequent* $X \to A$ *is a tree growing downward with the sequence* $\neg Xxtu, Axtu$ *at its apex and branches generated by applying decomposition rules or specific rules, one at time. A branch is* **closed** *iff it contains a fundamental sequence.*

Since all decomposition and specific rules are admissible and fundamental sequences are valid, it is easy to prove:

**Proposition 1** *If all the branches of a decomposition tree for* $X \to A$ *are closed, then* $X \to A$ *is valid in the class of RNL-models.*

**Definition 8** *A decomposition tree is* **complete** *iff either (a) all branches are closed or (b) for each vertex v of the tree (1) all decomposition rules have been applied to every formula* $Axtu$ *occurring in v; if a tagged decomposition rule is applicable to* $Axtu$*, for every branch br containing v, for all variables y and z occurring in the formulas of br, formulas obtained from* $Axtu$ *by application of the tagged rule with y and z as new variables occur in some vertex of br; and (2) the specific rules have been applied to the vertex v for all variables y and z.*

**Theorem 2** *If a complete decomposition tree of a sequent* $X \to A$ *is not closed, then there is an RNL-model in which* $X \to A$ *is not valid.*

**Proof** Use an open branch $br$ to generate an $RNL$-model $M_{br} = (W_{br}, R_{br}, m_{br})$ as follows. $W_{br}$ is the set of variables from the language and $R_{br}$ is defined as: $\{(x,y,z) \in R_{br}$ if and only if $Rxyz$ is not on the branch $br\}$. Then $m_{br}(R) = R_{br}$, for all $P \in C$, $m_{br}(P) = \{(x,y,z) \mid Pxyz$ is not on $br\}$, $m_{br}(A \circ B) = m_{br}A \circ m_{br}B$, and similarly for $\backslash, /, (.,.)$ and $\neg$. The specific rule ensures $m_{br}(P) \subseteq S \times W_{br} \times W_{br}$. To show $X \to A$ is not valid in $M_{br}$, argue by contradiction (see [Or3] for a similar argument).

The following correspondence theorem is obvious from Definitions 3 and 5.

**Theorem 3** *(a)For each RF-model M, there is an RNL-model M' such that the (NL-)sequent $X \to A$ holds in M iff the (RNL-)sequent $X \to A$ holds in M'.*

*(b)For each RNR-model M, there is an RF-model M' such that the (RNL-)sequent $X \to A$ holds in M iff the (NL-)sequent $X \to A$ holds in M'.*

Consequently: the $NL$-sequent $X \to A$ is valid in the class of $RF$-models iff the $RNL$-sequent $X \to A$ is valid in the class of $RNL$-models.

**Theorem 4** *A sequent $X \to A$ of NL is provable in NL if there is a closed decomposition tree for $X \to A$. If a complete decomposition tree for $X \to A$ is not closed, then there is an RNL-model (and consequently an RF-model) in which $X \to A$ does not hold.*

With this as a base, let us look now at tree models. Venema considered a number of extensions to $NL$ to obtain a completeness result for tree models. None of those considered sufficed: for each extension he considered, he found a sequent valid in tree models which was not provable in that extension to $NL$ (some of the sequents were: $A, A\backslash(B \circ C) \to A \circ C$, $A\backslash(B \circ C), A \to C \circ A$ and $A\backslash(B \circ C), D), A \to (C \circ D) \circ A$). Venema noted that the essential difficulty seemed to be that while the proposition

$t \models A\backslash(B \circ C)$ implies $t \models C$ provided that somewhere in the frame there is a tree $s$ with $s \models A$,

holds for tree models, the sequent form of the calculus $NL$ does not seem adequate to express *the somewhere* concisely. This problem is easily solved using $RNL$; we just need to add the following condition on relations:

$(A\backslash(B \circ C))ztu$ and $Astu$ implies $Cztu$;

this gives us the following specific rule which we must add to the deduction rules of $RNL$:

$$\frac{\neg(A\backslash B\circ C)ztu, \neg Astu,}{\neg(A\backslash B\circ C)ztu, \neg Astu, \neg Cztu}\ (**)$$

Call the new system $RNL_{**}$. Each of the three problem sequents in [Ve1]: $A, A\backslash(B \circ C) \to A \circ C$, $A\backslash(B \circ C), A \to C \circ A$ and $A\backslash(B \circ C), D), A \to (C \circ D) \circ A$ has a decomposition tree which close in $RNL_{**}$. We illustrate with one example.

**EXAMPLE:** $p, p\backslash(q \circ r) \to p \circ r$

$$\cfrac{\neg(p, p\backslash(q \circ r))xtu, (p \circ r)xtu}{\cfrac{\neg Rxyz, \neg pytu, \neg(p\backslash(q \circ r))ztu, (p \circ r)xtu}{\Gamma_1 \qquad\qquad \Gamma_2 \qquad\qquad \Gamma_3}}$$

where $\Gamma_1$ is the sequence: $\neg Rxyz, \neg pytu, \neg(p\backslash(q \circ r))ztu, Rxyz, (p \circ r)xtu$, and therefore closes, $\Gamma_2$ is the sequence: $\neg Rxyz, \neg pytu, \neg(p\backslash(q \circ r))ztu, pytu, (p \circ r)xtu$, and therefore closes, and $\Gamma_3$ is the sequence: $\neg Rxyz, \neg pytu, \neg(p\backslash(q \circ r))ztu, rztu, (p \circ r)xtu$ which closes with the application of the rule $(**)$. (In the decomposition tree we first applied $(\neg(Y, Z))$ and then $(\circ)^*$.)

We now extend $RNL_{**}$ by adding more specific rules and fundamental sequences to incorporate the axioms discussed in [Ve1]. We first consider groupoids. A groupoid is a set $G$ with a binary operation. A groupoid can be seen as a special kind of relational frame by putting $Rstu$ iff $s = tu$. However we need to express the fact that the operation is well defined; that is, if $s = tu$ and $s' = tu$ then $s = s'$. To do this one might introduce a distinguished element $e$ into our groupoid and require that it satisfy the conditions: $Rxxe$ and $Rxex$ and then say $Rstu$ and $Rs'tu$ implies $Rses'$. But this is a bad strategy because a tree with $t$ as a left branch and $u$ as a right branch is different from a tree with $u$ as a left branch and $t$ as a right branch; consequently we need the following condition on $R$:

(1) $Rstu$ implies $\neg Rsut$.

What we can do is introduce a second distinguished ternary relation, $I$, subject to the conditions:

(2) $Ixxy$;
(3) $Ixx'y$ if and only if $Ix'xy$;
(4) $Ixx'y$ and $Ix'x''y$ implies $Ixx''y$;
(5) $Ixx'y$ if and only if $Ixx'z$.

To define a groupoid frame we need a set with two ternary relations, $R$ and $I$, satisfying conditions (1) – (5), above, as well as:

(6) $\forall t, u \exists s\ (Rstu)$ (functionality),

and

(7) $Rstu$ and $Rs'tu$ implies $Iss'u$ (well-definedness).

Then, using the definition of [Ve1], a tree frame is a groupoid frame satisfying **unique splittability**; that is:

$Rwuv$ and $Rwu'v'$ imply $u = u'$ and $v = v'$,

which may be expressed in our relational language as:

(8) $Rwuv$ and $Rwu'v'$ imply $Iuu'y$ and $Ivv'y$;

and **acyclicity**, which is expressed in [Ve1] as:

for no distinct $x_0, ..., x_n$ do we have $x_0Ax_1...x_{n-1}Ax_nAx_0$.

(Here $xAy$ stands for "$x$ is a daughter of $y$ or $y$ is a daughter of $x$", which can easily be seen to be expressible as $\exists z(Ryxz$ or $Ryzx$ or $Rxyz$ or $Rxzy)$, and $xAyAz$ is read: $xAy$ and $yAz$.) The task of writing out a sequence of conditions on $R$ and $I$ (one for every positive integer) expressing the conditions for acyclicity is rather daunting. Let us just give the condition for three variables, using the abbreviation $xAy$:

($9_2$) $x_0Ax_1$ and $x_1Ax_2$ and $x_2Ax_0$ implies $Ix_0x_1y$ or $Ix_1x_2y$ or $Ix_2x_0$.

To axiomatize infinite trees Venema added the condition:

(10) $\forall s \exists t, u(Rstu)$.

To axiomatize finite trees, Venema used the condition that there were no paths of finite length. If we use the expression $x_iDx_{i+1}$ as notation for $x_{i+1}$ is a daughter of $x_i$, it will be an abbreviation for $\exists z_i(Rx_ix_{i+1}z_i$ or $Rx_iz_ix_{i+1})$. Letting $B_n$ be the expression $\neg(x_0Dx_1Dx_2...x_{n-1}Dx_n)$, then $B_n$ states that there is no path of length $n$. The condition we need is then:

(11) $\bigvee_{n \in N} B_n$.

Conditions $(3)-(5),(7)-(9_n)$ and $(11)$ require new specific rules. The other 4 conditions require new fundamental sequences, i.e., branch closing sequences. We present some of the specific rules and fundamential sequences here: the others are analogous. For example, the specific rule corresponding to condition (3) is:

$$\frac{Ixx'y}{Ix'xy},$$

while that corresponding to condition (4) is:

$$\frac{\neg Ixx'y, \neg Ix'x''y}{\neg Ixx'y, \neg Ix'x''y, \neg Ixx''y}.$$

The fundamental sequence corresponding to condition (1) is $\neg Rstu, \neg Rsut$, and that corresponding to condition (2) is $Ixxy$. Conditions (6) and (10) give conditional fundamental sequences. Consider, for example, condition (6) (we leave the specific rule for condition (10) as an exercise for the reader); the corresponding conditional branch closing condition is: if a tree is complete and a branch has a sequence $Rstu, Rytu, Rztu, Rwtu, \ldots..$, where the first variable ranges over all labels on the branch, then the branch is declared closed.

We are currently considering the completeness of a relational tableaux method with respect to tree-models. Venema expresses some reservations to trying to find a complete set of axioms for tree models, suggesting such must necessarily be an infinite set; indeed, this problem manifests itself in conditions $(9_n)$ and the infinitary condition (11). Perhaps we should be content with soundness and drop some of the conditions. We hope to be able axiomatize the system using specific rules involving $R$ and $I$ and do without the specific rule (**), above (that is, show that we can get specific rule $(**)$ as a derived rule).

# 3 Language models

**Definition 9** *(Language models without the empty word) A* **language** *is a set of finite (nonempty) sequences. A family of languages is a set* $\{L_i : i \in I\}$*, where* $L_i$ *is a set of finite (nonempty) sequences (words) over a finite alphabet. A language model is a family of languages enriched with the following operations:*

$$L_a \circ L_b = \{xy \mid x \in L_a \text{ and } y \in L_b\};$$
$$L_a \backslash L_b = \{x \mid (\forall y \in L_a)\ yx \in L_b\};$$
$$L_a / L_b = \{x \mid (\forall y \in L_a)\ xy \in L_b\}.$$

Pentus (see [P]) recently showed that associative $NL$, denoted here by $AssocNL$, is complete with respect to language models without the empty word. The Genzen rules for associativity are:

$$\frac{X[(Y_1,Y_2),Y_3] \to A}{X[Y_1,(Y_2,Y_3)] \to A} \qquad\qquad \frac{X[Y_1,(Y_2,Y_3)] \to A}{X[(Y_1,Y_2),Y_3] \to A}.$$

We must add the conditions:

$Rxyz$ and $Rztu$ implies $\exists w(Rxwu$ and $Rwyt)$

and

$Rxyz$ and $Rytu$ implies $\exists w(Rxtw$ and $Rwuz)$

on the ternary relation $R$ in the definition of $RNL$-model, to define $AssocRNL$-models. Clearly, language models without the empty word and $AssocRNL$-models are the same thing. The addition of the specific rules corresponding to the two conditions above (which in the first case is:

$$\frac{\neg Rxyz, \neg Rztu}{\neg Rxyz, \neg Rztu, \neg Rxwu, \neg Rwyt})$$

to the definition of $RNL$ rules, gives us $AssocRNL$-logic. In order to prove constructive completeness, the following cut rule must be included in the specific rules of $AssocRNL$-logic:

$$\frac{}{Rxyz \qquad\qquad \neg Rxyz}$$

where $x$ is a variable occurring in the formulas above the line. (This rule is used to prove the validity of the conditions for associativity in a countermodel constructed from an open branch). The proof of soundness and completeness of $AssocNL$ with respect to $AssocRNL$-models is straightforward. Then we may prove the following:

**Theorem 5** *A sequent $X \to A$ of AssocNL is provable in AssocNL if there is a closed decomposition tree for $X \to A$. If a complete decomposition tree for $X \to A$ is not closed, then there is an AssocRNL-model (and consequently an AssocRF-model) for which $X \to A$ does not hold.*

One could augment $NL$ with other structural rules. The Gentzen form of these rules may be found in [Do]; each Gentzen rule gives rise to conditions on the ternary relation $R$, which in turn give new specific rules. We leave this to the reader interested the other structural rules. Full details of the conditions required for a variety of structural augmentations to $FL$ (full Lambek calculus) may be found in [MacC2].

We now look at another version of associative Lambek calculus, denoted $LC$. This logic is the original Lambek calculus as a grammar (where the product $A \circ B$ of expressions is mentioned but not incorporated into the system). This logic has an interpretation in language models with the empty word. However it is not weakly complete with respect to these models. The language of LC may be defined as follows: given a denumerable set $P$ of primitive symbols, close this under the operations of $\backslash$, $/$ and $\circ$ to get the formulas. The set of *sequents* is the set of all expressions of the form $A_1, ..., A_n \to A_0$ , where the $A_i$ are formulas. For the axioms and rules of inference $A, B, C$ stand for formulas and $X, Y, Z$ stand for finite sequences of formula including the empty sequence, unless otherwise indicated.

**Definition 10** *The sequent derivation system of LC is given as follows:*

*Axiom:*

$$(LCO)\ A \to A$$

*Rules of Inference:*

$$(LC1) \quad \frac{X \to A \quad A \to B}{X \to B} \quad X \text{ nonempty}$$

$(LC \circ r)$ $\frac{X \to A \quad Y \to B}{X, Y \to A \circ B}$ $X, Y$ *nonempty* $\qquad$ $(LC \circ l)$ $\frac{X, A, B, Y \to C}{X, A \circ B, Y \to C}$

$(LC\backslash r)$ $\frac{A, X \to B}{X \to A \backslash B}$ $X$ *nonempty* $\qquad$ $(LC\backslash l)$ $\frac{X \to A \quad Y, B, Z \to C}{Y, X, A \backslash B \to C, Z}$ $X$ *nonempty*

$(LC/r)$ $\frac{X, A \to B}{X \to B/A}$ $X$ *nonempty* $\qquad$ $(LC/l)$ $\frac{X \to A \quad Y, B, Z \to C}{Y, B/A, X, Z \to C}$ $X$ *nonempty*

**Definition 11** *By a Kripke-model for LC we mean a triple $M = (W, R, V)$ where $W$ is a set, $R$ is a ternary relation on $W$, and $V$ is a valuation of the primitive symbols; i.e., $V : P \rightarrow P(W)$, and $R$ satisfies the following conditions:*

$$Ru_1u_2w \text{ and } Rv_1v_2u_2 \text{ implies } \exists z(Rzv_2w \text{ and } Ru_1v_1z);$$

$$Ru_1u_2w \text{ and } Rv_1v_2u_1 \text{ and } Rt_1t_2v_1 \text{ implies } \exists z_1, z_2(Rt_2v_2z_1 \text{ and } Rt_1z_1z_2 \text{ and } Rz_2u_2w).$$

*Truth is defined as follows: for any $x \in W$,*

$x \models p$ *iff* $x \in V(p)$;
$x \models A \circ B$ *iff there are* $y, z \in W$ *such that* $Ryzx$, *and* $y \models A$ *and* $z \models B$;
$x \models A\backslash B$ *iff for all* $y, z \in W$ *such that* $Ryxz$, *and* $y \models A$ *we have* $z \models B$;
$x \models A/B$ *iff for all* $y, z \in W$ *such that* $Rxyz$, *and* $y \models B$ *we have* $z \models A$;
$x \models A_1, ..., A_n \rightarrow A_0$ *iff* $x \models ((A_1 \circ A_2) ... \circ A_n)$ *implies* $x \models A_0$.

*A sequent $\phi$ of LC holds in the model M iff for all $x \in W$, $x \models \phi$.*

These semantics are sound with respect to $LC$. Completeness is an immediate consequence of the fact that $LC$ is complete with respect to the semantics in [An&M], called *RelSem*. In *RelSem*, worlds are pairs from a transitive binary relation. Starting with a model in *RelSem*, if we define the ternary relation $R$ by $Rxyz$ iff $y = (a, b)$, $z = (b, c)$ and $x = (a, c)$ for some $a, b, c$, we get a *Kripke*-model which satisfies the above two conditions. Given the *Kripke*-models, we can quickly define the relational logic, $RLC$. The relational operations $\circ, \backslash, /$ and $\neg$ are almost identical to those for $RNL$ - the one difference is that the order of the variables in $R$ change. $RLC$-models are as expected and the tableaux-style deductive system has decomposition rules almost identical to those for $RNL$ (including the cut rule, and incorporating the changes to the variables in $R$). We also need specific rules corresponding to the conditions in Definition 11 - we leave it to the reader to write these out. Validity in the class of Kripke-models for $LC$ is equivalent to validity in the class of $RLC$-models, so if a decomposition tree with $\neg((A_1 \circ A_2) \circ ... \circ A_n)xtu, Axtu$ at its apex is closed, then the sequent $A_1, A_2, ..., A_n \rightarrow A$ is $LC$-deducible, and so valid in language models with the empty word. If the complete decomposition tree has an open branch, we may construct a countermodel - which is a Kripke model for $LC$. (We can say nothing with respect to language models because we lack the completeness of language models with respect to $LC$.) Of course, the definition of $RLC$-model may be augmented by more conditions if we wish to consider structural additions to $LC$. The corresponding specific rules for $RLC$ are simple to determine from the added conditions.

## 4 Information Networks

In this section we briefly outline how the relational deduction may be applied to perfect reasoning about the flow of information through an information network. We refer the reader to [BGH].

**Definition 12** *An information frame is a triple $F = (S, C, \sim>)$ where $S$ is a set of objects called sites, $C$ is a set of objects called channels and $\sim>$ is a relation on $S \times C \times S$. The (signaling) relation $s \overset{c}{\sim>} t$ is read $c$ is a channel from $s$ to $t$. An information network is a* *pair N= (F, $\circ$), where F is an information frame and $\circ$ is an associative binary operation, such that for all channels $a, b$,*

$$\forall s, t (s \overset{a \circ b}{\leadsto} t \text{ iff } \exists r \mid s \overset{a}{\leadsto} r \text{ and } r \overset{b}{\leadsto} t).$$

*The language has a collection, AtExp, of atomic types. An information model consists of an information network N together with a function f assigning to each $A \in AtExp$ some sites and channels.*

**Definition 13** *Given an information model (N, f), the* **of type** *relation $\models$ is defined as follows:*

*for $A \in AtExp$, and $s \in S \cup C$, $s \models A$ iff $s \in f(A)$;*
*$c \models (A \circ B)$ iff $\exists c_1 c_2$ ($c = c_1 \circ c_2$ and $c_1 \models A$ and $c_2 \models B$);*
*$c \models (A \backslash B)$ iff $\forall s, t$ ($s \models A$ and $s \overset{c}{\leadsto} t$ then $t \models B$);*
*$s \models (A/B)$ iff $\forall c, t$ ($c \models B$ and $s \overset{c}{\leadsto} t$ then $t \models A$);*
*$t \models (A \downarrow B)$ iff $\exists s, c$ ($s \models A$ and $c \models B$ and $s \overset{c}{\leadsto} t$).*

[BGH] (and many others) use $\rightarrow$, respectively, $\leftarrow$ where we use $\backslash$, respectively, $/$.

[BGH] distinguished between two kinds of sequents, *s-sequents*, denoted here with $\rightarrow$ (and in [BGH], denoted by $\vdash$), and *c-sequents*, denoted here with $\sim$ (and in [BGH], denoted by $|\!\sim$). Say an s-sequent $A, C_1, C_2, ..., C_n \rightarrow B$ holds in a model M iff for every information chain, $s \overset{c_1}{\leadsto} t_1 \overset{c_2}{\leadsto} t_2 \leadsto \ ... \overset{c_n}{\leadsto} t_n$, if $s \models A$ and $c_i \models C_i$ for each $i$, then $t_n \models B$. A c-sequent $C_1, C_2, ..., C_n \sim C$ holds in M iff for every sequence $c_1, c_2, ..., c_n$ of channels, if $c = ((c_1 \circ c_2) \circ ... \circ c_n)$, and $c_i \models C_i$, for each $i$, then $c \models C$. Barwise et al. consider two situations - the two sorted case (sorts are sites and channels) and the one sorted case (channels are considered as sites).

The rules of $L$ are much like the Lambek rules, except in some places $\rightarrow$ is used and in other places $\sim$ is used. We will not give all definitions here; however, we will briefly get the reader started on the program of developing relational deduction for the language of information flow. We shall restrict ourselves to the one sorted case (channels are considered as sites); the other poses the technical problem of distinguishing between sites and channels. To define a relational calculus, we start with a language $RIF$ which has two ternary relations, $R_1$ and $R_2$, and the connectives $\circ, \backslash, /$ and $\downarrow$. (We shall use $R_1 c c_1 c_2$ to articulate $c = c_1 \circ c_2$ and $R_2 sct$ to articulate $s \overset{c}{\leadsto} t$.) In order to interpret formulas as ternary relations, we need relational operations to correspond to the logical connectives. For example, the operation corresponding to the connective $\downarrow$ is:

$$A \downarrow B = \{(x, t, u) \mid \exists x, y \ (Axtu \text{ and } Bytu \text{ and } R_2 zyx)\}.$$

The conditions on $R_1$ and $R_2$ are not difficult to work out - just work through the Gentzen rules given in [BGH], to determine the conditions on $R_1$ and $R_2$ required to show soundness and add those conditions (note some of the Gentzen rules do not need any conditions). This gives $RIF$-models. The decomposition rules and specific rules are worked out from the definitions of relational operations and the conditions on the relational operators.

## 5 Connections to other work and some open questions

The system $RFL$ of relational deduction for full Lambek calculus found in [MacC1] has many more specific rules than $RLC$. However the language for $RFL$ has the full expressive power of full Lambek calculus (with connectives $\vee$, $\wedge$ and the constants 0, 1, $\top$ and $\perp$). A second ternary relation, $I$, is required to take care of the connective $\vee$. Moreover, a hereditary condition is part of the Kripke structure on which the relational logic was built, and this necessitates extra conditions on the ternary relations. To my knowledge, $RFL$ and its extensions to other substructural logics was the first constructively complete deductive system for the additive as well as the multiplicative fragment of $FL$ and its structural extensions. (We have not yet seen a completeness proof for the LDS of [G&D'A] for the additive fragment of the substructural logics.) It is still an important open question whether or not the van Benthem semantics $RelSem$, (see [An&M]) is complete with respect to the Lambek calculus $LC$ *with axioms for* $\vee$ *adjoined.* If it were, one could extend $RLC$ and obtain a completeness result corresponding to that given at the end of Section 3. In that case, one would hope for a relational models with fewer specific rules than $RFL$-models. Note that in the literature, the van Benthem semantics are referred to as relational semanatics - the reason being that validity with respect to $RelSem$ is equivalent to validity with respect to a semantics called $RRS$ (representable relativized relational structures), wherein formulas are interpreted as binary relations, which are subrelations of a binary relation $W$ (to use the jargon, the binary operations are *relativized to* W). The advantage of $RRS$ is that its equational theory is decidable.

The question of the connection of the various semantics which use ternary relations to arrow logic of van Benthem has arisen in several place. Arrow logic (summarizing [Ve2]): unifies insights from computer science, linguistics and philosophy by focussing on the notion of dynamic semantics. Because relation algebras emerge as the modal or complex algebras of arrow logic, arrow logic is intimately connected to the program of algebraic logic. (Indeed in [Ve2] we see that one of the motivations for developing arrow logic is to investigate the theory of Relation Algebras.) We see the same phenomenon in the work by Orlowska [Or2]: models of each of the logics under investigation are sets of binary relations which together with the associated operations constitute algebras of binary relations; that is, relation algebras. Thence formalisms can be studied via algebraic tools using algebras which are well understood; formulas and inferences become equations and quasi-equations. This is not the case in [MacC1] and [MacC2]; there, (the relations we use are ternary and more importantly,) the operations do not all satisfy the axioms of a relation algebra. The problem is the operation corresponding to the logical connective $\vee$ is not the union operation. We believe, however, that these algebras of ternary relations, though not relation algebras, can be studied and valuable representation and duality theorems can be developed. Thus the program of algebraic logic remains to be developed for much of the substructural spectrum. It is to this end that we include the specific rule in the system $RNL$: because of this rule the relations are all ideal relations - an important consideration in the algebrizing of the many nonclassical logics considered by Orlowska. A discussion of arrow logic and $RelSem$ is found in [An&M]. The algebras of relations which we use to give semantics for the $LC$ and $AssocNL$ are reducts of relation algebras. We emphasize however, that in contrast to arrow logic, the Orlowska-style relational semantics is always accompanied by tableaux-style deductive system. Moreover, with appropriate cut rules, constructive completeness is assurred. I will say that as a general rule we can define (Orlowska-style) relational semantics and the associated deduction system to correspond to any $Kripke$-style semantics.

A second, somewhat different, understanding of arrow logic may be found in [V], where the main motivation was a logic which had interpretations in models such as directed graphs, categories, vectors, and others. The connection of Vakarelov's arrow logic and that of van Benthem is articulated in [V]; in fact, the three relations from van Benthem's arrow frames may be defined in terms of the four binary incidence relations of Vakarelov. (The introduction to [V] also summarizes the connections between completeness, representability and decidability.)

Frias and Orlowska (see [F&Or]) point out that the existing (Orlowska-style) relational framework for nonclassical logic has the disadvantage that the class of algebras of binary relations is not finitely axiomatizable (so the equational theory is not decidable). The class of fork algebras is both finitely axiomatizable and representable and they suggest that the fork algebra formalism may be a better candidate for relational formalization of nonclassical logics. They justify this claim by examples from a wide spectrum of logics. In particular, they show that validity in *relevant fork models* is equivalent to validity in relational models for a relevant logic. It is an open question whether one could make a fork algebra formalism for substructural logics weaker than relevant logics or for substructural logics with weakening but no contraction (BCK-logics). It seems to me that the connective $\vee$ will be problematic. Of course, if one were only interested in the associative Lambek calculus, $LC$, it is already known that the equational reasoning in the associated relational algebras, $RRS$ is decidable. In [Dob], Dobrowolska develops a general method for proving decidability of several fragments of Relational Logic, while Konikowska [K] has some decidability results for certain kinds of relational logics.

Another semantical framework, using ternary (as well as binary) relations, for a whole spectrum of substructural logics is given by Dunn's *gaggle theory* (see [Du], [Du&M], [A&Du] and [A&MacC]). Relational deduction methods (tableaux-style) may be developed for these systems (see [A&MacC]), though they have many more deduction rules. In [Du&M], we find a description of combinatory logic as a substructural logic, so methods in relational deduction (resulting from the gaggle approach) may be applied to combinatory logic. The author is currently considering relational deduction for Lukasiewicz logics, which do not fall under the paradigm of gaggle theory but are substructural logics.

In Basin et al., (see [BaMV]) we find a deductive system which is a labelled deduction system combined with a logical framework, which provides a natural deduction system which is implementable for a whole spectrum of modal logics. The logics considered are the base logic $K$ to which various Horn relational theories are added. The claim is made that this hybrid approach is preferable both to the LDS of Gabbay and also to the relational deduction method of Orlowska because of its strong modularity and its strong proof theoretic properties. However, [BaMV] prove that the advantages over systems based on semantic embedding (such as LDS and relational deduction) will be lost for any logic weaker than $K$, and that they depend on the nature of the relational axioms used. Both $NL$ and $LC$ are weaker than $K$.

In [Ro], Roorda embeds (associative) Lambek calculus in a modal logic. Roorda begins with $D$, a dyadic modal logic very much like $K$, but with several modalities, and adds six axioms; the result is the system $DL$, which includes both Lambek calculus and also boolean connectives. The Lambek calculus is faithfully embedded in $DL$. Orlowska considers this system in [Or1]. She uses six different modalities to interpret the Lambek operations $\circ$, $\backslash$ and/, and, using a semantics of binary relations, (models are reducts of relation algebras) presents relational deduction for it. At the time of this writing, the proof of completeness of the relational deduction was still under consideration. It would be interesting to see if the [BaMV]-approach could be applied to $DL$.

Finally, a number of proof theoretic questions and questions regarding practical implementation of $RNL$ and the modifications considered here remain to be answered.

### References


[A&Du] Gerard Allwein and J.Michael Dunn, *Kripke models for linear logic*, Journal of Symbolic Logic, Volume 58 (1993) 514-545.

[A&MacC] __________ and Wendy MacCaull, *Kripke semantics for the logic of Gelfand quantales*, in preparation.

[An&M] Hajnal Andréka and Szabolcs Mikulas, *Lambek calculus and its relational semantics: completeness and incompleteness*, Journal of Logic, Language and Information, Volume 3 (1994) 1-38.

[BGH] Jon Barwise, Dov Gabbay and Chrysafis Hartonas, *On the logic of information flow*, Journal of the Interest Group in Pure and Applied Logic, Volume 3 (1995) 7-49.

[BaMV] David Basin, Séan Matthews and Luca Viganò, *Labelled propositional modal logics: theory and practise*, Preprint, 1996.

[Bu] Wojciech Buszkowski, *Mathematical Linguistics and Proof Theory*, in: Handbook of Logic and Language, eds: J. van Benthem and A. ter Meulen, Elsevier, 1997.

[D'A&G] Marcello D'Agostino and Dov Gabbay, deductive systems I: Basic substructural logics, Journal of Automated Reasoning, Volume 13 (1994) 243-281.

[D&Or] Stéphane Demri and Ewa Orlowska, *Informational Representability of Models for Information Logics*, Warsaw University of Technology Institute of Computer Science Research Report 9/96.

[Dob] Wiera Dobrowolska, *Some decidable classes in relational logic*, Bulletin of the Polish Academy of Sciences, Mathematics, Volume 44 (1996) 87-102.

[Do] Kosta Došen, *Sequent-systems and groupoid models I*, Studia Logica, Volume 47 (1988) 353-385.

[Du] J.Michael Dunn, *Partial gaggle theory applied to logics with restricted structural rules*, in: Substructural Logics, eds: K.Došen and P. Schroeder-Heister, Oxford, 1993.

[Du&M] __________ and Robert K. Meyer, *Combinators and structurally free logic*, Journal of the Interest Group of Pure and Applied Logic, Volume 5 (1997) 505-537.

[G] Dov Gabbay, *A general theory of structured consequence relations*, in: Substructural Logics, eds: K.Došen and P. Schroeder-Heister, Oxford, 1993, 109-152.

[F&Or] Marcelo Frias and Ewa Orlowska, *Equational reasoning in nonclassical logics*, to appear, Journal of Non-Classical Logic.

[L-F] Saturnino Luz-Filho, *Grammar specifications in categorial logics and theorem proving*, preprint, 1996.

[K] Beata Konikowska, Personal communication.

[MacC1] Wendy MacCaull, *Relational semantics and a relational proof system for full Lambek calculus*, to appear, Journal of Symbolic Logic, 1998.

[MacC2] __________ *Relational proof theory for linear and other substructural logics*, Journal of the Interest Group of Pure and Applied Logic, Volume 5 (1997) 673-697.

[Mo] Michael Moortgat, *Categorial Type Logics*, in: Handbook of Logic and Language, eds: J. van Benthem and A. ter Meulen, Elsevier, 1997.

[Mo&Oe] __________ and Dick Oehrle, UCLA Course Slides, 1997.

[Mu&P] Chris Mulvey and Joan Wick Pelletier, *The quantization of the calculus of relations*, Canadian Mathematical Society Conference Proceeding, Volume 13, American Mathematical Society, 1992.

[O] Hiroakira Ono, *Semantics for substructural logics*, in: Substructural Logics, eds: K.Došen and P. Schroeder-Heister, Oxford, 1993, 259-291.

[Or1] Ewa Orlowska *Relational environment for semigroup logics*, in: Logic, Action, and Information, Essays on Logic in Philosophy and Artificial Intelligence, eds: André Fuhrmann and Hans Rott, Walter de Gruyter, 1996, 351-391.

[Or2] __________ *Relational semantics for nonclassical logics: formulas are relations*, in: Philosophical Logic in Poland, ed: Jan Wolenski, Kluwer, 1994, 167-186.

[Or3] __________ *Relational proof system for relevant logics*, Journal of Symbolic Logic, Volume 57 (1992) 1425-1440.

[P] Mati Pentus, *Language completeness of the Lambek calculus*, Proceedings of the Nineth Annual IEEE Symposium on Logic in Computer Science, 487-496, 1994.

[R&S] Helena Rasiowa and R.Sikorski, The Mathematics of Metamathematics, Polish Scientific Publishers, 1963.

[Ro] Dirk Roorda, Resource Logics: Proof Theoretical Investigations, PhD dissertation, University of Amsterdam, 1990.

[V] Dimiter Vakarelov, *Many-dimensional arrow structures. Arrow logics II*, in: Arrow Logic and Multi-modal Logic, Studies in Logic, Language and Information, eds: L.Polos, M.Marx and M.Masuch, CSLI Stanford, 1996, 141-187.

[Ve1] Yde Venema, *Tree models and (labeled) categorial grammar*, Journal of Logic Language and Information, 5 (1996) 253-277.

[Ve2] __________ *A crash course in arrow logic*, in: Arrow Logic and Multi-modal Logic, Studies in Logic, Language and Information, eds: L.Polos, M.Marx and M.Masuch, CSLI Stanford, 1996.