Labelling and completeness for categorial resource logics

Natasha Kurtonina
Research Institute for Language and Speech (OTS)
Trans 10, 3512 JK Utrecht

Natasha.Kurtonina@let.ruu.nl

Abstract

An important theme in current categorial research is the shift of emphasis from individual type logics to communicating families of such systems. The reason for this shift is that the individual logics are not expressive enough for realistic grammar development: the grammar writer needs access to the combined inferential capacities of a family of logics. See [Morrill 94], [Moortgat 95] for discussion and motivation. In line with these developments, our objective in this paper is to develop a uniform labelling discipline for the family of resource logics **NL, L, NLP** and **LP** and the generalizations discussed in depth in [K& M 95], and to establish a completeness result for the proposed labelling regime with respect to the general frame semantics for these logics. Our approach uses a generalization of our earlier [Kurtonina 94] completeness result for the 'dynamic' relational semantics of **L** which was based on a labelled deductive presentation of the logic.

1 Introduction: labelling for categorial type systems

Let us situate our approach with respect to related work, before starting with the technicalities. The technique of labelling has been used in the categorial literature before, for various reasons. [Buszkowski 86] used labelling as an auxiliary device to obtain his completeness results for the Lambek calculus. Victor Sanchez in [Sanchez 90] relied on labelling for semantic reasons in his work on a categorially-driven theory of natural language reasoning. [Moortgat 91] added string labelling to type formulas for syntactic reasons, viz. to overcome the expressive limitations of the standard sequent language in capturing discontinuous forms of linguistic composition. In a proof theoretic study of categorial logics, [Roorda 91] introduced labelling to enforce the well-formedness conditions on proof nets. The programmatic introduction of Labelled Deductive Systems as a general framework for the study of structure sensitive consequence relations in [Gabbay 92] made it possible to re-evaluate these scattered earlier proposals. From 1991 on, labelling has been on the categorial agenda on a more systematic level. For recent studies, we refer to [Morrill 94, Morrill 95], [Oehrle 94], [Hepple 94], [Venema 94], to mention just a few.

Consider a consequence relation

$$A_1, \ldots, A_n \Rightarrow B$$

representing the fact that a conclusion B can be derived from a database of assumptions $A_1, \ldots, A_n$. The central idea of the labelled deductive approach is to replace the formula as the basic declarative unit by a pair $x : A$, consisting of a label x and a formula A. Sequents then assume the form

$$x_1 : A_1, \ldots, x_n : A_n \Rightarrow y : B$$

The label is to be thought of as an extra piece of information added to the formula. Rules of inference manipulate not just the formula, but the formula plus its label. We then obtain a whole scala of labelling regimes depending on how we make precise the intuitive notion of an 'extra piece of information added to a formula' — depending on the degree of autonomy between the formula and the label.

At the conservative end of the spectrum one can find semantic lambda term labelling in the sense of the Curry-Howard correspondence. The labels, in this application, simply record the history of the proof — they do not make an independent contribution. At the other end of the spectrum are the labelling systems where in the declarative unit $x : A$ the label x and the formula A each make their own irreducible contribution. Such systems can best be seen as combinations of two logics: the formula logic and the logic governing the labelling algebra.

Our proposals are somewhere in between these two extremes. In the application to linguistic reasoning which is the subject of this chapter, a consequence relation represents a grammaticality judgement: the derivability of an expression of type B from a database of assumptions, i.e. expressions of type $A_1, \ldots, A_n$. Derivability, in this linguistic sense, has to take into account the structure of the assumptions — for example: their linear order, and the way they are grouped into hierarchical units or constituents. We rely on the labelling algebra to make the structure of our linguistic database explicit. In [K& M 95], we have seen how one obtains the systems **L, NLP, LP** from the pure residuation logic **NL** by removing order sensitivity, constituent structure sensitivity or both. In order to develop uniform algorithmic proof theory for this family of type logics we start from the multiset sequent presentation of their common denominator, the Lambek-Van Benthem system **LP**, and impose the extra syntactic fine-tuning in terms of the labelling regime.

Labelling, in this sense, is not uncommon in current categorial work, see for example [Morrill 94, Morrill 95] for descriptive and computational applications. But the labelling systems in use are related to groupoid interpretation, not to the more abstract ternary frame semantics which is the 'common ground' for the general completeness results for the various individual systems in the categorial landscape, and for the mixed multimodal logics. From the perspective of ternary frame semantics, the groupoid models are obtained by specializing the accessibility relation Rx, yz to $x = y + z$, where the binary operation $+$ imposes functionality. The groupoid models become inappropriate when one wants to consider *one-sided* structural postulates (for example: left-associativity without right-associativity), rather than the two-way inferences for which the groupoid equational reasoning is appropriate[1]. In developing labelling for the abstract ternary frame semantics, we can implement the labelling discipline in such a way that we can manipulate the three components of the triples $(xyz) \in R$, rather than assuming that x is determined by $y + z$.

Let us finally insist here on the importance of the completeness result for the labelled calculus. Labels, just like formulas, are pieces of syntax. Even if the rules of a labelled system *look* very much *like* imitating truth conditions we will never have a guarantee that the labelling is really appropriate for the intended interpretation unless we present a completeness proof, which requires an explicit statement of the relation between the labels and the objects of the domain of interpretation. That there are real issues here is shown in the discussion of [Venema 94].

[1] To accommodate one-way structural postulates, one can move to *ordered* groupoids, cf. [Došen 88,89, Došen 92a], but in terms of 'abstractness' these would be on the same level as the general frame semantics adopted here.

The paper is organized as follows. In §2 we introduce uniform labelled sequent presentation for the logics **NL, NLP, L, LP**. We prove completeness with respect to the ternary frame semantics for these logics in §3. Labelled sequent calculus is presented in two formats which we show to be equivalent. The first format decorates formulas with atomic labels and enriches the sequent language with an explicit book-keeping component to record the structure of the database. The second format introduces a language of tree-terms over the atomic labels, and formulates the labelling discipline as a term assignment system. In §5, we discuss a number of generalizations, showing how our results can be applied to multimodal architectures, and to logics extended with one-place multiplicative operators. §6 then moves to concrete labelling. We use the labelling algebra to add sortal refinements to the type formulas: the sort labels filter out theorems that would be derivable in an unlabelled setting.

2 Resource management in labelled sequent calculus

Let us introduce the labelled sequent presentation informally before starting with the definitions. The new declarative unit, as we said before, now is the labelled formula $x : A$, rather than the bare formula A. What kind of labels do we want to use, and how do we want to interpret them? In our first version of the labelled calculus the labels are all taken from some set of *atomic* markers. In a sense to be made precise below, the labels refer to elements of the domain of interpretation W. In order to keep track of the way the labels are configured into a structured database, we add an explicit book-keeping component to the sequent language. Labelled sequents then will be of the general form

$$[\delta];\ a_1 : A_1, a_2 : A_2, \ldots, a_n : A_n \Rightarrow a : A,$$

where each label $a_i (1 \leq i \leq n)$ is a witness of a piece of information attached to A_i and d fixes the configuration of the labels in a tree such that the succedent label a is its root and the antecedent labels a_i the leaves. Because the configuration is fixed in δ the sequent antecedent can be treated simply as a *multiset* of labelled formulae. Now for the definitions.

Definition 2.1. Labels and trees. Let **Lab** be a set of atomic labels. We define by simultaneous induction the set of trees **T** over **Lab** and functions

root: $\mathbf{T} \rightarrow \mathbf{Lab}$

leaves: $\mathbf{T} \rightarrow P(\mathbf{Lab})$

nodes: $\mathbf{T} \rightarrow P(\mathbf{Lab})$

- If $x \in \mathbf{Lab}$, then x is a tree such that

  leaves$(x) = \{x\}$
  root$(x) = x$
  nodes$(x) = \{x\}$

- If $\langle a, bc\rangle \in \mathbf{Lab}^3$ and a, b, c are distinct, then $\langle a, bc\rangle$ is a tree such that

  root$(\langle a, bc\rangle) = a$
  leaves$(\langle a, bc\rangle) = \{b, c\}$

$$\text{nodes}(\langle a, bc\rangle) = \{a, b, c\}$$

- If δ_1 and δ_2 are trees such that

  $\text{root}(\delta_2) \in \text{leaves}(\delta_1)$ and
  $\text{nodes}(\delta_1) \cap \text{nodes}(\delta_2)=\{\text{root}(\delta_2)\}$

  then $\xi=(\delta_1\delta_2)$ is a tree and

  $\text{root}(\xi)=\text{root}(\delta_1)$
  $\text{leaves}(\xi)=(\text{leaves}(\delta_1) \cup \text{leaves}(\delta_2))\text{-}\{\text{root}(\delta_2)\}$
  $\text{nodes}(\xi)=\text{nodes}(\delta_1) \cup \text{nodes}(\delta_2)$

EXAMPLE 2.2. $\xi=((\langle a, bc\rangle\langle c, de\rangle)\langle d, kn\rangle)$ is a tree with

$\text{root}(\xi)=a$

$\text{leaves}(\xi)=\{b, k, n, e\}$

$\text{nodes}(\xi)=\{a, b, c, d, e, k, n\}$

Thus, each tree turns out to be a bracketed string of triangles. As usual we often drop the most external pair of brackets. The size (l) of a tree can be defined as follows:

- $l(a) = 0$
- $l(\langle a, bc\rangle) = 1$
- $l(\xi\chi) = l(\xi) + l(\chi)$

Now we can give precise definitions of a labelled formula and a labelled sequent.

- $a : A$ is a labelled formula if A is a formula and $a \in \textbf{Lab}$;
- $[\delta];\ a_1 : A_1, a_2 : A_2, \ldots, a_n : A_n \Rightarrow a : A$ is a labelled sequent if $a : A, a_i : A_i (1 \leq i \leq n)$ are labelled formulas, and $\delta \in \textbf{T}$ with
  - $\text{root}(\delta)=a$
  - $\text{leaves}(\delta)=\{a_1 \ldots a_n\}$

Note that in a labelled sequent

$$[\delta];\ a_1 : A_1, a_2 : A_2, \ldots, a_n : A_n \Rightarrow a : A$$

$a_1, \ldots, a_n, a$ are distinct since according to the definition all nodes of δ are distinct. Moreover the presence of δ is not conservative, since the rules of the labelled Lambek calculus include not only manipulations with formulas and labels, but also with trees.

We are in a position now to define the labelled sequent presentation first of all for the pure residuation logic **NL**, then for the systems with structural rules.

DEFINITION 2.3. Labelled sequent calculus: the pure residuation logic $\mathbf{NL}^{lab}$. Let α be a labelled formula, X, Y, Z finite multisets of labelled formulas, and let δ_1, δ_2, be members of $\mathbf{T}$. $\mathbf{NL}^{lab}$ has one axiom

$$[a];\ \ a : A \ \Rightarrow\ a : A$$

and the following inference rules, provided that all sequents involved are well-defined:

$$\frac{[\delta_1];\ X \Rightarrow b : A \qquad [\delta_2];\ c : B, Y \Rightarrow \alpha}{[(\delta_2\langle c, ba\rangle)\delta_1];\ a : A \rightarrow B,\ X, Y \Rightarrow \alpha}$$

$$\frac{[\langle c, ba\rangle d]; b : A, X \Rightarrow c : B}{[\delta];\ X \Rightarrow a : A \rightarrow B}$$

$$\frac{[\delta_1];\ X \Rightarrow b : A \qquad [\delta_2];\ c : B, Y \Rightarrow \alpha}{[(\delta_2\langle c, ab\rangle)\delta_1];\ a : A \leftarrow B, X, Y \Rightarrow\ \alpha}$$

$$\frac{[\langle c, ab\rangle d];\ X, b : A \Rightarrow\ c : B}{[\delta];\ X \Rightarrow a : B \leftarrow A}$$

$$\frac{[d\langle a, bc\rangle];\ b : A, c : B, X \Rightarrow \alpha}{[\delta];\ a : A \bullet B, X \Rightarrow \alpha}$$

$$\frac{[\delta_1];\ X_1 \Rightarrow b : A \quad [\delta_2]; X_2\ \Rightarrow\ c : B}{[(\langle a, bc\rangle\delta_1)\delta_2];\ X_1, X_2 \Rightarrow a : A \bullet B}$$

DEFINITION 2.4. Structural rules as manipulations on labels. Structural options for resource management have the general format

$$\frac{[\delta];\ X \Rightarrow \alpha}{[\delta'];\ X \Rightarrow \alpha}$$

where δ' is obtained from δ via an operation matching the structural rule in question. Specifically, $\mathbf{L}^{lab}$ can be obtained from $\mathbf{NL}^{lab}$ by adding the Associativity Rule that δ' is obtained from δ by replacing some subtree $(\langle a, bc\rangle\langle c, de\rangle)$ by a tree $(\langle a, te\rangle\langle t, bd\rangle)$ or vice versa provided that t (resp. c) is fresh. $\mathbf{NLP}^{lab}$ can be obtained from $\mathbf{NL}^{lab}$ by adding the Permutation Rule that δ' is obtained from δ by replacing some subtree $(\langle a, bc\rangle)$ by a tree $(\langle a, cb\rangle)$. Finally, $\mathbf{LP}^{lab}$ is obtained from $\mathbf{NL}^{lab}$ by adding both Associativity and Permutation.

The following sequents give an example of theorems of $\mathbf{NL}^{lab}$.

$$\begin{array}{ll} (i) & [\{\langle x, bc\rangle, \langle d, ax\rangle\}]; a : r \leftarrow q, b : p, c : p \rightarrow q \Rightarrow d : r \\ (ii) & [\{\langle x, bc\rangle, \langle n, ax\rangle\}]; a : r \leftarrow q, b : p, c : p \rightarrow q \Rightarrow n : r \end{array}$$

Note that the derivation of (ii) can be obtained from the derivation of (i) by renaming d for n.

LEMMA 2.5. *Renaming Lemma.* The Lemma consists of two claims.

- Claim 1.
  If a sequent $[\delta];\ x_1 : X_1, x_2 : X_2, \ldots, x_n : X_n \Rightarrow x : X$ is derivable, then $[\delta']; x_1 : X_1, \ldots, \mathbf{y} : X_i, \ldots, x_n : X_n \Rightarrow x : X$ where δ' is obtained from δ by replacing x_i by $\mathbf{y}$ is also derivable.

- Claim 2.
  If a sequent $[\delta];\ x_1 : X_1, x_2 : X_2, \ldots, x_n : X_n \Rightarrow \mathbf{x} : X$ is derivable then $[\delta];\ x_1 : X_1, x_2 : X_2, \ldots, x_n : X_n \Rightarrow \mathbf{y} : X$ where δ' is obtained from δ by replacing x by $\mathbf{y}$ is also derivable.

Both Claims can be proved by straightforward induction on the length of the derivation. □

INTERPRETING LABELLED SEQUENTS. To obtain an interpretation for our labelled sequents we add to a ternary model $\langle W, R, V\rangle$ a function * which assigns exactly one element of W to each $a \in \mathbf{Lab}$.

A sequent $[\delta]$; $a_1 : A_1, a_2 : A_2, \ldots, a_n : A_n \Rightarrow a : A$ is true in M if $a^* \models A$ whenever $a_i^* \models A_i (1 \leq i \leq n)$ and for each triangle $\langle x, yz\rangle$ which occurs in δ, Rx*,y*z* . A sequent ϕ is semantically valid if it is true in all models.

Note that although on the syntactic level the labels on the nodes of each triangle are distinct, we do not impose such distinctness as a semantic requirement. Since * is an arbitrary function it might very well be that in some models the same elements of the domain are assigned to distinct node labels of the syntactic tree. Thus on the semantic level the definition of a ternary frame realizes an *arbitrary* ternary accessibility relation as required.

As usual it is easy to prove soundness, i.e. each sequent derivable in $\mathbf{NL}^{lab}$ is semantically valid. Completeness is the subject of the following section.

3 Labelling and completeness

We establish completeness of the labelling regime first for $\mathbf{NL}^{lab}$, then for the systems with extra resource management properties. The general idea behind the completeness proof can be expressed as follows:

1. Suppose $[\delta]$; $a_1 : A_1, a_2 : A_2, \ldots, a_n : A_n \Rightarrow a : A$ is not derivable in $\mathbf{NL^{lab}}$.
2. Mark all labelled formulas on the left hand side with T and $a : A$ with F. The resulting T-F set is

$$\Delta_0 = \{Ta_1 : A_1, \ldots, Ta_n : A_n, Fa : A\}.$$

   Construct a model such that each a_i^* supports A_i, but a^* does not support A. In other words, extend Δ_0 to Δ and prove that $x^* \models X$ iff $Tx : X \in \Delta$.

To realize this idea of the completeness proof we need to identify some properties of $T - F$ sets, i.e sets of labelled formulas marked with T or F.

PROPERTIES OF T-F SETS. Let Δ be a $T - F$ set, V_Δ the set of all labels that occur in Δ, Δ_R the set of all triangles associated with Δ, and $\delta \in \mathbf{T}$ (i.e. δ is a tree). We loosely say that $\delta \subseteq \Delta_R$ if all triangles that occur in δ are members of Δ_R.

- Δ is *deeply consistent* (d.c.) iff whenever $\delta \subseteq \Delta_R$, $\gamma_1, \ldots, \gamma_k$ are T-members of Δ and $[\delta]$; $\gamma_1, \ldots, \gamma_k \Rightarrow \gamma$ is derivable, then $F\gamma \notin \Delta$.
- Δ is *h-complete* (Henkin complete) iff
  - (i) if $Fa : A \to B \in \Delta$, then there are $x, y \in V_\Delta$ such that $\langle y, xa\rangle \in \Delta_R$, $Tx : A \in \Delta$ and $Fy : B \in \Delta$;
  - (ii) if $Fa : A \leftarrow B \in \Delta$, then there are $x, y \in V_\Delta$ such that $\langle y, ax\rangle \in \Delta_R$, $Tx : A \in \Delta$ and $Fy : B \in \Delta$;
  - (iii) if $Ta : A \bullet B \in \Delta$, then there are $x, y \in V_\Delta$ such that $\langle a, xy\rangle \in \Delta_R$, $Tx : A \in \Delta$ and $Ty : B \in \Delta$.
- Δ is *r-complete* (relatively complete) iff

(i) if $Fa : A \bullet B \in \Delta$ and $\langle a, xy \rangle \in \Delta_R$, then either $Fx : A \in \Delta$ or $Fy : B \in \Delta$;

(ii) if $Ta : A \to B \in \Delta$ and $\langle y, xa \rangle \in \Delta_R$, then either $Fx : A \in \Delta$ or $Ty : B \in \Delta$;

(iii) if $Ta : A \leftarrow B \in \Delta$ and $\langle y, ax \rangle \in \Delta_R$, then either $Fx : A \in \Delta$ or $Ty : B \in \Delta$.

- Δ is *nice* iff it is d.c., h-complete, r-complete
- Let Δ be a T-F set, $x \in V_\Delta$ and A be a non-labelled formula. We say that A and x are *linked* in Δ iff $Tx : A \in \Delta$ or $Fx : A \in \Delta$.

Henkin model.

Lemma 3.1. If Δ is a nice $T - F$ set, then there exists a model $M' = \langle W, R, ^*, V \rangle$ such that if $x \in V_\Delta$ and X are linked in Δ, then $Tx : X \in \Delta$ iff $x^* \models X$ in M'.

proof. Define M' as follows:

$W = V_\Delta$ (the set of all labels that occur in Δ);

Rx, yz iff $\langle x, yz \rangle \in \Delta_R$.

for all $x \in V_\Delta$, $x^* = x$

V is such that $Tx : p \in \Delta$ iff $x^* \models p$ in M'.

The lemma is proved by induction on the length of X. In the atomic case the claim is a direct consequence of the definition. We have to take care of three cases when proving the inductive step.

The first case: $X = A \bullet B$.

1. Suppose $Ta : A \bullet B \in \Delta$.
2. Since Δ is h-complete there are $x, y \in V_\Delta$ such that $\langle a, xy \rangle \in \Delta_R$, $Tx : A \in \Delta$ and $Ty : B \in \Delta$.
3. Therefore $Ra, xy; x \models A$ and $y \models B$ (by inductive assumption) and $a \models A \bullet B$
4. Thus $a^* \models A \bullet B$

1. Suppose $a^* \models A \bullet B$ and therefore $a \models A \bullet B$.
2. Then there are $x, y \in W$ such that $Ra, xy; x \models A$ and $y \models B$.
3. Since a and $A \bullet B$ are linked in Δ, either $Ta : A \bullet B \in \Delta$ or $Fa : A \bullet B \in \Delta$. Let us suppose that $Fa : A \bullet B \in \Delta$.
4. Since Δ is r-complete, $Fa : A \bullet B \in \Delta$ and $\langle a, xy \rangle \in \Delta_R$ (because Ra, xy) imply that either $Fx : A \in \Delta$ or $Fy : B \in \Delta$; in both cases we get a contradiction with [2.].
   Thus $Ta : A \bullet B \in \Delta$.

The second case: $X = A \to B$

1. Suppose $Ta : A \to B \in \Delta$.
2. Suppose Rc, ba and $b \models A$

3. Then $\langle c, ba\rangle \in \Delta_R$ and $Tb : A \in \Delta$
4. Since Δ is r-complete, [1.] [2.] and [3.] imply $Tc : B \in \Delta$, and therefore by the inductive assumption $c \models B$
5. Hence, $a^* \models A \to B$

1. Suppose $a^* \models A \to B$
2. Suppose $Ta : A \to B \notin \Delta$. Then since a and $A \to B$ are linked in Δ, $Fa : A \to B \in \Delta$
3. By h-completeness of Δ, there are $b, c \in V_\Delta$ such that $Tb : A \in \Delta, Fc : B \in \Delta$ and $\langle c, ba\rangle \in \Delta_R$
4. By inductive assumption, get a contradiction. Therefore $Ta : A \to B \in \Delta$

The third case can be left to the reader □

Lemma 3.1 enables one to claim that if a sequent $u : A \Rightarrow u : B$ is not derivable and the corresponding $T - F$ set $\{Tu : A, Fu : B\}$ can be extended to a nice one, then there exists a model where $u : A \Rightarrow u : B$ is falsified. Now we have to define how to make the relevant extensions of d.c. sets. We start with the definition of saturation with h-witnesses and then r-witnesses.

HARMLESS WITNESSES. Formulas of the form $Fa : A \to B$, $Fa : A \leftarrow B$, $Ta : A \bullet B$ will be called *h-formulas* (Henkin formulas). Let Δ be a d.c. set. By *adding h-witnesses* we refer to the following procedure:

(i) if $Fa : A \to B \in \Delta$, then add new labels b and c to V_Δ; $\langle c, ba\rangle$ to Δ_R and new labelled formulas $Tb : A$ and $Fc : B$ to Δ;

(ii) if $Fa : A \leftarrow B \in \Delta$, then add new labels b and c to V_Δ ; $\langle c, ab\rangle$ to Δ_R and new labelled formulas $Tb : A$ and $Fc : B$ to Δ;

(iii) if $Ta : A \bullet B \in \Delta$, then add new labels b and c to V_Δ; $\langle a, bc\rangle$ to Δ_R and new labelled formulas $Tb : A$ and $Tc : B$ to Δ.

We say that in (i)-(iii) the point a *generates* points b and c. The set of successors of a point x (Σ_x) is defined by (iv)-(v):

(iv) if x generates y, then $y \in \Sigma_x$;

(v) if $u \in \Sigma_x$ and u generates w, then $w \in \Sigma_x$.

To saturate some $T - F$ set Δ with r-witnesses perform (i)-(iii)

(i) if $Fa : A \bullet B \in \Delta$ and $\langle a, xy\rangle \in \Delta_R$, then add $Fx : A$ to Δ if it does not disturb d.c. of Δ, otherwise add $Fy : B$

(ii) if $Ta : A \to B \in \Delta$ and $\langle y, xa\rangle \in \Delta_R$, then add $Fx : A$ to Δ if it does not disturb d.c. of Δ, otherwise add $Ty : B$

(iii) if $Ta : A \leftarrow B \in \Delta$ and $\langle y, ax\rangle \in \Delta_R$, then add $Fx : A$ to Δ if it does not disturb d.c. of Δ, otherwise add $Ty : B$.

Formulas of the form $Fa : A \bullet B,\ Ta : A \rightarrow B,\ Ta : A \leftarrow B$ will be called *r-formulas.* We say that an r-formula $Fa : A \bullet B$ (resp. $Ta : A \rightarrow B, Ta : A \leftarrow B$) is *active in* Δ if there are $b, c \in V_\Delta$ such that $\langle a, bc\rangle \in \Delta_R$ (resp. $\langle c, ba\rangle \in \Delta_R,\ \langle c, ab\rangle \in \Delta_R$).

Let Δ be a $T - F$ set, V_Δ be a set of labels that occur in Δ and Δ_R be a set of triangles associated with Δ such that each member of Δ_R is a result of some h-decomposition. Then the following propositions hold.

PROPOSITION 3.2. If $x \in V_\Delta$ and X is the set of all successors of x, then all members of X are distinct.

PROOF. Direct consequence of the definition of adding h-witnesses. □

PROPOSITION 3.3. If $\delta \subseteq \Delta_R$, then δ is generated by a single point.

PROOF. Straightforward induction on the size of δ. Indeed, if $\delta = \langle a, bc\rangle$, then it is generated by a single point according to the definition of adding h-witnesses.

CLAIM
If $\delta = \xi_1\xi_2$, and x is their common point, then either ξ_1 or ξ_2 is generated by x
PROOF. If in both ξ_1 and ξ_2 x is a generated point, then x has to be generated twice: as a daughter and as a root, which is not possible, since by the definition of adding h-witnesses every point in Δ_R is uniquely generated. Thus, at least in one of this trees x is a generator.

Next, suppose $\delta = \xi_1\xi_2$, x is their common point, and ξ_i is generated by x, while ξ_j $(i, j = 1, 2)$ is generated by y. Then either $x = y$ or x is a successor of y or vice versa. In all this cases $\delta = \xi_1\xi_2$, is generated by a single point. □

PROPOSITION 3.4. Let $\delta_1, \delta_2 \subseteq \Delta_R$ with root $(\delta_2) \in$ leaves (δ_1). Then $\delta_1\delta_2 \subseteq \Delta_R$

PROOF. We have to show that $\delta_1\delta_2$ is a well defined tree, or in other words, that δ_1 and δ_2 have no other points in common besides x. Reasoning by analogy with the proof of the Proposition 3.3, conclude, that x generates δ_1 or δ_2 or both of them. If x generates δ_1 and y generates δ_2, then no matter if x is a successor of y or vice versa, the successor would always generate fresh points , therefore δ_1 and δ_2 can not share any points besides x. If x generates both δ_1 and δ_2, then in one case the set of successors would be generated by x as a root (and therefore, by a formula of the form $Ta : A \bullet B$) while the second set would be generated by x as a daughter (and therefore by formula of the form $Fa : A \rightarrow B$ or $Fa : A \leftarrow B$), thus two latter sets can not share any members. □

LEMMA 3.5. Let Δ be a d.c. set, and each member of Δ_R is a result of some h-decomposition. If β is an active r-formula in Δ, then there always exists an r-witness of β which can be added to Δ without disturbing its deep consistency.

PROOF

The first case: $Fa : A \bullet B \in \Delta$

Let β be $Fa : A \bullet B$ and $\langle a, bc\rangle \in \Delta_R$. Suppose that neither $\Delta + Fb : A$, nor $\Delta + Fc : B$ is d.c. Then clearly

(1) there exist $\delta_1 \subseteq \Delta_R$ and T-members of Δ $\gamma_1, \ldots, \gamma_n$ such that

$$(*) \quad [\delta_1];\ \gamma_1, \ldots, \gamma_n \Rightarrow \gamma$$

is derivable in $\mathbf{NL^{lab}}$ and γ is an F-member of $\Delta + Fb : A$. Since Δ is d.c., γ is nothing else but $b : A$;, and $(*)$ has actually the form

$$(*) \qquad [\delta_1];\ \gamma_1, \ldots, \gamma_n \Rightarrow b : a$$

Thus, b is the root of δ_1

(2) there exist $\delta_2 \subseteq \Delta_R$ and T-members of Δ $\alpha_1, \ldots, \alpha_n$ such that

$$(**) \qquad [\delta_2];\ \alpha_1, \ldots, \alpha_n \Rightarrow \alpha$$

is derivable in $\mathbf{NL^{lab}}$ and α is an F-member of $\Delta + Fc : B$. Once again, since Δ is d.c., α is nothing else but $c : B$; whence

$$(**) \qquad [\delta_2];\ \alpha_1, \ldots, \alpha_n \Rightarrow c : B$$

is derivable and therefore c is the root of δ_2

According to Proposition 3.4 $((\langle a, bc\rangle\delta_1)\delta_2$ is a well defined tree, thus by (1) and (2)

$$[((\langle a, bc\rangle\delta_1)\delta_2];\ \gamma_1, \ldots, \gamma_k, \alpha_1, \ldots, \alpha_n \Rightarrow A \bullet B$$

is derivable. Since $((\langle a, bc\rangle\delta_1)\delta_2 \subseteq \Delta_R$ and $\gamma_1, \ldots, \gamma_k, \alpha_1, \ldots, \alpha_n$ are T-members of Δ but $a : A \bullet B$ is an F-member of Δ, Δ can not be d.c..

The second case: $Ta : A \to B \in \Delta$

Let $Ta : A \to B \in \Delta$ and $\langle c, ba\rangle \in \Delta_R$. Suppose neither $\Delta + Fb : A$ nor $\Delta + Tc : B$ is d.c. Then clearly

(1) there exist $\delta_1 \subseteq \Delta_R$ and T-members of Δ $\gamma_1, \ldots, \gamma_k$ such that

$$[\delta_1];\ \gamma_1, \ldots, \gamma_k \Rightarrow \gamma$$

is derivable in $\mathbf{NL^{lab}}$ and γ is an F-member of $\Delta + Fb : A$. Since Δ is d.c., γ is obviously $b : A$;

(2) there exist $\delta_2 \subseteq \Delta_R$ and T-members of $\Delta + Tc : B$ $\alpha_1, \ldots, \alpha_n$ such that

$$[\delta_2];\ \alpha_1, \ldots, \alpha_n \Rightarrow \alpha$$

is derivable in $\mathbf{NL^{lab}}$ and α is an F-member of Δ. Since Δ is d.c., there exists an index i such that α_i coincides with $c : B$;

Clearly c is a leaf of δ_2 and b is a root of δ_1. According to Proposition 3.4 $(\delta_2\langle c, ba\rangle)\delta_1$ is a well defined tree. Therefore the sequent

$$[(\delta_2\langle c, ba\rangle)\delta_1];\ \alpha_1, \ldots, \alpha_{i-1}, \gamma_1, \ldots, \gamma_k, a : A \to B, \alpha_{i+1}, \ldots, \alpha_n \Rightarrow \alpha$$

turns out to be derivable and Δ can not be d.c..

The last case $(\alpha := Tb : A \leftarrow B)$ is left to the reader. □

LEMMA 3.6. Adding h-witnesses does not disturb d.c..

PROOF. Suppose $Fa : A \to B$ belongs to a d.c. set Δ. Suppose that adding $\langle c, ba\rangle$ (where b and c are fresh) to Δ_R and adding $Tb : A$ and $Fc : B$ to Δ does disturb d.c.. Therefore, there exists a derivable sequent

$$(\sharp)\ \ [\delta];\ \gamma_1, \ldots, \gamma_n \Rightarrow \gamma$$

such that $\delta \subseteq \Delta_R \cup \{\langle c, ba\rangle\}$; $\gamma_1, \ldots, \gamma_n$ are T-members of $\Delta + Tb : A$ and γ is an F-member of $\Delta + Fc : B$. Note, that if $\langle c, ba\rangle$ does not occur in δ, then Δ is not d.c.. Thus $\delta = \langle c, ba\rangle\delta_1$, for some $\delta_1 \subseteq \Delta_R$, moreover b and c can occur only once in a tree, generated by δ. Therefore γ coincides with $c : B$ and γ_1 coincides with $b : A$, hence the sequent $(\sharp)$ has actually the form

$$[\langle c, ba\rangle\delta_1];\ b : A, \gamma_2, \ldots, \gamma_n \Rightarrow c : B$$

Thus

$$[\delta_1];\ \gamma_2, \ldots, \gamma_n \Rightarrow a : A \to B$$

is also derivable, and since $\delta_1 \subseteq \Delta_R$, $\gamma_1, \ldots, \gamma_n$ are T-members of Δ, and $a : A \to B$ is an F-member of Δ, Δ can not be d.c.. In the case of adding h-witnesses of $Fa : B \leftarrow A$ or $Ta : A \bullet B$ our argument would not be much different. ☐

COMPLETENESS PROOF: FROM DEEPLY CONSISTENT SETS TO NICE SETS. Recall our initial assumption: $a : A \Rightarrow a : B$ is not derivable in NL^{lab}. Define $\Delta_0, \ldots \Delta_n, \ldots (n \in N)$ as follows:

$\Delta_0 = \{Ta : A, Fa : B\}$.

Δ_{n+1}: add all possible h-witnesses to Δ_n, corresponding triangles to Δ_{R_n}

Δ_{n+2}: add all possible r-witnesses to Δ_{n+1}.

By Lemma 3.5 and Lemma 3.6, $\Delta = \cup\Delta_i (i \in N)$ is nice. Now our Completeness Theorem becomes just a direct consequence of the previous results.

Completeness theorems for $\mathbf{NLP}^{lab}$, $\mathbf{L}^{lab}$ and $\mathbf{LP}^{lab}$ require the additional proof that taking the associative or permutational closure of some d.c. set does not disturb its deep consistency. But this is guaranteed due to the presence of the corresponding structural rule in the sequent presentation of the labelled version of the Lambek Calculus. Note, that the Cut rule was not used in our completeness proof, which means that we have obtained semantical proof of Cut Elimination theorem. For the constructive procedure of Cut Elimination we refer to [Kurtonina 95].

4 Labelling with Kripke tree terms

The labelling preceding subsection decorated the formulas always with atomic labels: the structure of the database was accounted for by adding to the sequent language an explicit representation of the tree configuration of the atomic labels. The alternative labelling regime to be introduced below has a term language to build structured labels out of the atomic labels. The term language now directly captures the structure of the database, so that we can remove the book-keeping component from the sequent language.

DEFINITION 4.1. *Elementary tree terms, tree terms, proper tree terms.*

(i) If $x \in \mathbf{Lab}$, then x is an elementary tree term

(ii) If χ is an elementary tree term, then χ is a tree term;

(iii) If ξ, χ are tree terms and $x \in \mathbf{Lab}$,, then $r(x, \xi, \chi)$ is a tree term.

A tree term t is called *proper* if all its elementary subterms are distinct.
We define the *size* (l) of the tree term as follows:

(i) if x is an elementary tree term, then $l(x) = 0$

(ii) $l(r(x, \xi, \chi)) = l(\xi) + l(\chi) + 1$

Example $r(a, r(x, r(y, bc), d)e)$ is a proper tree term which corresponds to the following tree: $(\langle a, xe\rangle\langle x, y, d\rangle)\langle y, b, c\rangle$.

To prove that each proper tree term corresponds to some tree and vice versa, one proceeds by induction on the length of a tree term for one direction and on the length of a tree for the other one.

DEFINITION 4.2. *Labelled sequents with Kripke tree terms.* Let a, b, c be elementary tree terms, and t, u, v proper tree terms. A labelled sequent is an expression of the form

$$a_1 : A_1, a_2 : A_2, \ldots, a_n : A_n \Rightarrow t : A$$

where each $a_i (1 \leq i \leq n)$ is an elementary tree term and t is a proper tree term.

DEFINITION 4.3. *Labelled sequent calculus.* Let X, Y be finite multisets of formulas labelled with elementary tree terms. And let a, b, c be elementary tree terms, t, u, v proper tree terms as before. We write $t = t'[u/v]$ if t is obtained from the proper tree term t' by replacing the subterm v by u.

The labelled sequent presentation for the basic system $\mathbf{NL}^{\mathbf{lab}}$ has one axiom-scheme and the following inference rules:

$$a : A \;\Rightarrow\; a : A$$

$$\frac{X \Rightarrow t_1 : A \qquad c : B, Y \Rightarrow t_2 : \alpha}{a : A \rightarrow B,\ X, Y \Rightarrow t : \alpha} \quad \text{where } t = t_2[r(c, t_1 a)/c] \qquad\qquad \frac{b : A, X \Rightarrow r(c, bt) : B}{X \Rightarrow t : A \rightarrow B}$$

$$\frac{X \Rightarrow t_1 : A \qquad c : B, Y \Rightarrow t_2 : \alpha}{a : A \leftarrow B, X, Y \Rightarrow\ t : \alpha} \quad \text{where } t = t_2[r(c, at_1)/c] \qquad\qquad \frac{b : A, X \Rightarrow r(c, bt) : B}{X \Rightarrow a : B \leftarrow A}$$

$$\frac{b : A, c : B, X \Rightarrow t_1 : \alpha}{a : A \bullet B, X \Rightarrow t : \alpha} \quad \text{where } t = t_1[a/r(a, bc)] \qquad\qquad \frac{X_1 \Rightarrow t_1 : A \qquad X_2 \;\Rightarrow\; t_2 : B}{X_1, X_2 \Rightarrow r(a, t_1 t_2) : A \bullet B}$$

As before, one obtains labelled presentation for the systems **L, NLP** and **LP** by adding Associativity, Permutation or their combination:

$$\frac{X \Rightarrow t : A}{X \Rightarrow t' : A}$$

where (Associativity) t' can be obtained from t by replacing a subterm $r(a, t_1, r(b, t_2 t_3))$ by a tree term $r(a, r(c, t_1 t_2)t_3)$, provided that c is fresh, or (Permutation) where t' can be obtained from t by replacing a subterm $r(a, t_1 t_2)$ by a tree term $r(a, t_2 t_1)$.

Clearly via translation of proper tree terms into trees and vice versa one can easily prove the equivalence of the two formulations of the Lambek Calculus. A direct consequence of this fact is *soundness* and *completeness* of the Lambek Calculus with tree terms as labels with respect to ternary relation semantics.

5 Generalizations

In the preceding section we have looked at the systems **NL, L, NLP, LP** and presented uniform labelled sequent calculus for these logics, with completeness results for the relevant classes of ternary frames. The methods used are quite general: they can be straightforwardly applied to a number of related systems. Two generalizations seem especially relevant in view of current linguistic applications: the move from unimodal to multimodal architectures, and the introduction of unary multiplicative operators in addition to the familiar binary ones. We discuss these in turn.

5.1 Multimodal architectures

In the preceding paragraphs, we have studied the systems **NL, L, NLP, LP** in isolation: each of these systems characterizes a distinct resource management regime in terms of a package of structural rules — rules for the manipulation of labels in our labelled presentation. It has been argued in the linguistic literature (see for example [Moort. & Morrill 91], [Moort. & Oehrle 94], [Morrill 94] and [Moortgat 95] that for purposes of actual grammar development, one wants to have access to the combined inferential capacity of these various systems.

On the model-theoretic level, such a mixed style of inference requires a move from unimodal frames $\langle W, R\rangle$ to multimodal frames $\langle W, \{R_i\}_{i\in I}\rangle$. We now distinguish a family of accessibility relations: each of these R_i can have its own individual resource management properties, or if R_i and R_j have the same resource management regime they can still be kept distinct in virtue of their indexes i and j. On the syntactic level, we also index the connectives with $i \in I$, so that we can interpret each $\bullet_i$ (and its residual implications $\rightarrow_i$ and $\leftarrow_i$) in terms of its own accessibility relation R_i. Structural postulates, and the corresponding frame conditions, are relativized to the mode indexes. Apart from the standard structural options differentiating **NL, L, NLP** and **LP**, the multimodal architecture supports *mixed* forms, where Associativity or Commutativity apply when two modes are in construction with each other. Such mixed structural principles greatly enhance the linguistic expressivity of the framework. See [Moort. & Oehrle 94], [Morrill 94] for concrete illustrations. We will treat the labelled version of such principles in a moment.

Our framework for labelled deduction directly accommodates the multimodal categorial architecture. We sketch the necessary changes for the tree term labelling. In the definition of tree terms, we now have a family of term constructors r_i instead of the one r of the unimodal setting

DEFINITION 5.1. *Multimodal systems: elementary tree terms, tree terms, proper tree terms.* Let **Lab**, be a set of atomic markers, as before, and **I** a set of resource management mode indices.

(i) If $x \in \mathbf{Lab}$, then x is an elementary tree term

(ii) If χ is an elementary tree term, then χ is a tree term;

(iii) If ξ, χ are tree terms , $x \in \mathbf{Lab}$, and $i \in \mathbf{I}$, then $r_(x, \xi, \chi)$ is a tree term.

Similarly, in the definition of the multimodal labelled sequent calculus, we harmonize the mode information on the connectives and on the associated tree term labels. Below the logical rules for the connectives.

$$\frac{X \Rightarrow t_1 : A \qquad c : B, Y \Rightarrow t_2 : \alpha}{a : A \rightarrow_i B,\ X, Y \Rightarrow t : \alpha} \qquad \frac{b : A, X \Rightarrow r_i(c, bt) : B}{X \Rightarrow t : A \rightarrow_i B}$$
$$\text{where } t = t_2[r_i(c, t_1 a)/c]$$

$$\frac{X \Rightarrow t_1 : A \qquad c : B, Y \Rightarrow t_2 : \alpha}{a : A \leftarrow_i B, X, Y \Rightarrow\ t : \alpha} \qquad \frac{b : A, X \Rightarrow r_i(c, bt) : B}{X \Rightarrow a : B \leftarrow_i A}$$
$$\text{where } t = t_2[r_i(c, at_1)/c]$$

$$\frac{b : A, c : B, X \Rightarrow t_1 : \alpha}{a : A \bullet_i B, X \Rightarrow t : \alpha} \qquad \frac{X_1 \Rightarrow t_1 : A \qquad X_2\ \Rightarrow\ t_2 : B}{X_1, X_2 \Rightarrow r_i(a, t_1 t_2) : A \bullet_i B}$$
$$\text{where } t = t_1[a/r_i(a, bc)]$$

The structural rules, in the multimodal setting, are mode restricted: they refer to specific resource management mode labels. Our earlier versions of Associativity and Permutation would now assume the following form (for obvious mode labels).

$$\frac{X \Rightarrow t : A}{X \Rightarrow t' : A}(\dagger)$$

- Associativity Rule: ($\dagger$) where $t' = t[r_{ass}(a, r_{ass}(c, t_1 t_2) t_3)/r_{ass}(a, t_1, r_{ass}(b, t_2 t_3))]$, provided that c is fresh.
- Permutation Rule: ($\dagger$) where $t' = t[r_{perm}(a, t_2 t_1)/r_{perm}(a, t_1 t_2)]$.

Apart from these familiar unimodal structural options, our language is now expressive enough to also formulate mixed version for situations where different modes are in construction with each other. As an illustration, we present versions of Mixed Associativity and Mixed Commutativity for communication between modes i and j. The structural postulates are as follows.

Mixed Commutativity	$A \bullet_i (B \bullet_j C) \vdash B \bullet_j (A \bullet_i C)$
Mixed Associativity	$A \bullet_i (B \bullet_j C) \vdash (A \bullet_i B) \bullet_j C)$

Translated in the labelling format, these structural postulates become rules for manipulating term labels:

- Mixed Associativity Rule: ($\dagger$) where $t' = t[r_j(a, r_i(c, t_1 t_2) t_3)/r_i(a, t_1, r_j(b, t_2 t_3))]$, provided that c is fresh.
- Mixed Permutation Rule: ($\dagger$) where $t' = t[r_j(a, t_2 r_i(c, t_1 t_3))/r_i(a, t_1, r_j(b, t_2 t_3))]$.

For a linguistic application of this type of communication principle, the reader can turn to [Moort. & Oehrle 94], who give a multimodal analysis of head-adjunction phenomena such as can be found in the Germanic Verb-Raising construction. In a sentence such as

dat Jan (een boek (wil lezen))
(that J. a book wants read) i.e.
that John wants to read a book

the verb 'wil' (want) has to be combined semantically with the combination of the main verb 'lezen' (read) and its direct object 'een boek' (a book). But on the syntactic level, 'wil' does not combine with the phrase 'een boek lezen' but just with its head, viz. 'lezen'. Let the main verb combine with its arguments in mode i, and the modal auxiliary 'wil' in mode j, then the Mixed Commutativity rule makes it possible to proceed from the surface syntactic organization to the configuration required for semantic interpretation.

5.2 Unary multiplicatives

The labelling method and completeness proof of the previous sections was formulated for families of binary residuated connectives and their ternary accessibility relation. Residuation can be generalized to n-ary families of connectives interpreted with respect to $n+1$-ary accessibility relations, [Dunn 91] for an excellent survey, and [Moortgat 95, K& M 95] for the categorial application. Our labelling approach can be applied straightforwardly in the generalized residuation setting.

Especially relevant for the linguistic applications is the case of *unary* residuated operators, interpreted with respect to a binary accessibility relation. The basic residuation pattern for the unary operators assumes the following form:

$$\Diamond A \Rightarrow B \quad \text{iff} \quad A \Rightarrow \Box^{\downarrow} B$$

Semantically, we have the usual truth conditions below:

$$\begin{array}{c} x \models \Diamond A \text{ iff } \exists y(Rxy \ \& \ y \models A) \\ x \models \Box^{\downarrow} A \text{ iff } \forall y(Ryx \Rightarrow y \models A) \end{array}$$

Definition 5.2. *Labelled sequent calculus: unary residuated connectives.* Labelled sequent presentation for unary multiplicatives requires a generalization of the notion of a tree term to include binary tree terms built with the constructor r^2 next to the ternary case we had before: if ξ is a tree terms, $x \in \mathbf{Lab}$, then $r^2(x, \xi)$ is a tree term. The logical rules for the new connectives $\Diamond, \Box^{\downarrow}$ then assume the following form.

$$\frac{b : A, X \Rightarrow t_1 : B}{a : \Box^{\downarrow} A, X \Rightarrow t : B} \qquad \frac{X \Rightarrow r^2(a,t) : A}{X \Rightarrow t : \Box^{\downarrow} A}$$

where $t = t_1[r^2(b,a)/b]$

$$\frac{b : A, X \Rightarrow t_1 : B}{a : \Diamond A, X \Rightarrow t : B} \qquad \frac{X \Rightarrow t : A}{X \Rightarrow r^2(a,t) : \Diamond A}$$

where $t = t_1[a/r^2(a,b)]$

6 Concrete uses of labelling: sortal refinement

The use of labelling in the previous sections is still on the conservative side: we have given a uniform presentation for a family of categorial type logics with different resource management properties by introducing a division of labour between the sequent language and the labelling discipline: the sequent language is kept uniform, and the syntactic fine-tuning is taken care of by the labels.

In this section we want to give a very simple illustration of a form of labelling where the label has acquired a greater degree of independence from the formula, i.e. where the label allows one to incorporate *additional* information relevant to the process of linguistic inference. We show how one can decorate simple type assignment with labels capturing *morphophonological sortal* information. The sortal decoration makes it possible to rule out derivations that would go through if we restricted the attention to unlabelled type assignments.

Compare the adjective modifier **very** and the prefix **un-.** On the syntactic level, they are both functors taking adjectives into adjectives. We could assign them the type $\mathbf{a} \leftarrow \mathbf{a}$ and type $\mathbf{a}$ to **happy.** But given this type assignment both the grammatical **very unhappy** and the ungrammatical **un-very happy** are derivable according to the scheme

$$a \leftarrow a, a \leftarrow a, a \Rightarrow a$$

Can we refine the type assignment in such a way as to take into account the different combinatory possibilities of affix, words, and phrases? To block the undesired derivation and to keep the desired one we decorate the type formulas with sort labels, characterizing **very** and **happy** as syntactic words and **un-** as an affix. Our assignments in labelling format could take the form

very	word	:	$a \leftarrow a$
un-	affix	:	$a \leftarrow a$
happy	word	:	a

Moreover, we have to impose constraints on the composition relation in order to characterize the well-formed combinations on the morphophonological sort level:

(i) $\langle$word, affix word$\rangle$
(ii) $\langle$phrase, word phrase$\rangle$
(iii) $\langle$phrase, word word$\rangle$

Next, to express the fact that a syntactic word can do the duty as a phrasal expression (but not vice versa, of course) we adopt the following rule:

> Let Ψ be a set of constraints and a triangle $\delta \in \Psi$ has a root –**word**, then there exists $\delta' \in \Psi$ with the same leaves as δ and with the root–**phrase**.

In our case that means that (i) adds one more triangle to the set of constraints:

(i') **$\langle$word, affix word$\rangle$**

Now using suitable abbreviations and marks to distinguish tree points we can present a derivation of **very unhappy**:

$$\dfrac{w_1 : a \Rightarrow w_1 : a \qquad \dfrac{ph_1 : a \Rightarrow ph_1 : a \qquad ph_2 : a \Rightarrow ph_2 : a}{[\langle ph_2, w_2 ph_1\rangle] w_2 : a \leftarrow a, ph_1 : a \Rightarrow ph_2 : a}}{[\langle ph_2, w_2 ph_1\rangle\langle ph_1, af w_1\rangle]; w_2 : a \leftarrow a, af : a \leftarrow a, w_1 : a \Rightarrow ph_2 : a}$$

On the other hand **un- very unhappy** turns out to be underivable since our constraints on composition relation do not allow to combine **affix** with **phrase.**

Formally passing from the abstract style of labelling to the concrete style exemplified here we have to specify the definition of a model $\langle W, R, V\rangle$ of interpretation by taking W as a set of sorts and imposing frame constraints on R which are indicated above.

Thus on the level of interpretation, filtering out derivations that would be valid in the non-labelled setting is realized by restricting the class of all ternary models to those satisfying the constraints formulated above.

As we remarked at the beginning of this paragraph, our objective here was just to provide a simple illustration of 'autonomous' forms of labelling. For an elaboration of this style of labelling on a much more fundamental and wide-ranging level, we can refer to Ruth Kempson's work on combining syntactic and semantic inference [Kempson 95].

7 Conclusion

Our main objective in this paper has been to develop a uniform labelling regime for the landscape of categorial resource logics, and to establish completeness of the labelling discipline with respect to the general frame semantics. As remarked at the beginning of this paper, in computational studies of categorial grammar, labelling is often introduced as a tool to obtain efficient parsing strategies. We leave it as a topic for further research how the general labelling method developed in this paper can be combined with the compilation techniques of [Morrill 95] that allow for efficient checking of subproblems of the general labelling problem in terms of optimal data-structures.

References

[Buszkowski 86] W. Buszkowski (1986), 'Completeness Results for Lambek Syntactic Calculus', Zeitschrift fur mathematische Logik und Grundlagen der Mathematik 32, 13-28

[Došen 88,89] K. Došen (1988,1989) "Sequent systems and groupoid models". Studia Logica 47, 353-385, 48, 41-65.

[Došen 92a] K. Došen (1992), "A brief survey of frames for the Lambek calculus". Zeitschr. f. math. Logik und Grundlagen d. Mathematik 38,179-187

[Došen 92b] K. Došen (1992) "Modal translations in substructural logics". Journal of Philosophical Logic 21, 283-336.

[Dunn 91] J. M. Dunn (1991), "Gaggle theory: an abstraction of Galois connections and residuation, with applications to negation, implication, and various logical operators'. In Van Eijck (ed.) Logics in AI. JELIA Proceedings. Springer, Berlin.

[Gabbay 92] D. Gabbay (1992), Labelled Deductive Systems. Summer School of Logic, Language and Information. Essex 1992.

[Hepple 90] M. Hepple (1990) The Grammar and Processing of Order and Dependency: A Categorial Approach. PhD Dissertation, University of Edinburgh.

[Hepple 94] M. Hepple (1994) Labelled Deduction and Discontinuous Constituency. In 'Linear Logic and Lambek Calculus", Proceedings 1993 Rome Workshop

[Kempson 95] R. Kempson (1995), "Ellipsis: a natural deduction perspective". In Kempson (ed) *Language and Deduction*, IGPL Bulletin, Vol 3 (2,3).

[Kurtonina 94] N. Kurtonina (1994), The Lambek Calculus, Relational Semantics and the Method of Labelling. To appear in Studia Logica.

[Kurtonina 95] N. Kurtonina (1995), 'Talking about explicit databases in Categorial Grammar'. Bulletin of the *IGPL*, Vol 3 (2,3), pp 357–370.

[K& M 95] N. Kurtonina and M. Moortgat (1995), "Structural Control". This volume. To appear in Blackburn & de Rijke (eds) *Logic, Structure and Syntax.* Kluwer, Dordrecht.

[Moortgat 91] M. Moortgat (1991), Generalized quantifiers and discontinuous type constructors. Report, OTS, Utrecht University

[Moortgat 88] M. Moortgat (1988), Categorial Investigations. Logical and Linguistic Aspects of the Lambek Calculus. Foris, Dordrecht.

[Moortgat 95] M. Moortgat (1995), "Multimodal linguistic inference", *IGPL* Bulletin, Vol 3 (2,3), p 371–421. Special issue *Deduction and Language* (guest editor: Ruth Kempson). (Revised version to appear in *JoLLI*).

[Moort. & Morrill 91] M. Moortgat and G. Morill (1991), "Heads and phrases. Type calculus for dependency and constituent structure" . Manuscript, OTS Utrecht.

[Moort. & Oehrle 93] M. Moortgat and R.T. Oehrle (1993) Logical Parameters and Linguistic Variations. Lecture Notes on Categorial Grammar. 5th European Summer School in Logic, Language and Information, Lisbon.

[Moort. & Oehrle 94] M. Moortgat and R.T. Oehrle (1994), "Adjacency, dependency and order'. Proceedings 9th Amsterdam Colloquium, pp 447-466.

[Morrill 94] G. Morrill (1994), *Type Logical Grammar. Categorial Logic of Signs.* Kluwer Academic Publishers

[Morrill 95] G. Morrill (1995), 'Higher-order Linear Logic Programming of Categorial Deduction'. Proceedings *EACL95*, Dublin.

[Oehrle 94] R. T. Oehrle (1995), "Term-labeled categorial type systems", Linguistics and Philosophy, 17.633–678.

[Sanchez 89] V. Sanchez (1989), Natural Logic, Generalized Quantifiers and Categorial Grammar. PhD Dissertation, ILLC, University of Amsterdam

[Venema 94] Y. Venema, (1994) Tree Models and labelled Categorial Grammar. Report, OTS, University of Utrecht. To appear in *JoLLI*.