A Unified Compilation Style Labelled Deductive System for Modal and Substructural Logic using Natural Deduction.

Krysia Broda   Alessandra Russo

Department of Computing
Imperial College of Science, Technology and Medicine
180 Queen's Gate, London SW7 2BZ
email: {kb,ar3}@doc.ic.ac.uk

Technical Report DoC. 97/10

October 1997

Abstract

This paper describes a proof theoretic and semantic approach in which logics belonging to different families can be given common notions of derivability relation and semantic entailment. This approach builds upon Gabbay's methodology of Labelled Deductive Systems (LDS) and it is called the *compilation approach* for labelled deductive systems (CLDS). Two different logics are here considered, (i) the modal logic of elsewhere (known also as the logic of inequality) and (ii) the moltiplicative fragment of substructural linear logic. A general natural deduction style proof system is given, in which the notion of a theory is defined as a (possibly singleton) structure of points, called a *configuration*, and a "general" model-theoretic semantic approach is described using a translation technique based on first-order logic. Then it is shown how both this proof theory and semantics can be directly applied to the logic of elsewhere and to linear logic, illustrating also that the same technique for proving soundness and completeness can be adopted in both logics. Finally, the proof systems for the logic of elsewhere and for linear logic are proved to correspond, under certain conditions, to standard Hilbert axiomatisation and standard sequent calculus respectively. Such results prove that the natural deduction proof systems described in this paper are *proper generalisations* of any proof system already developed for these two logics.

1 Introduction

The use of logics in areas such as computer science and artificial intelligence has led to the proliferation of a large number of various logical systems often characterised by different notions of derivability relation, different sets of logical connectives as well as different underlying semantics. Logics within the same family often differ in "small" variations either in their proof theory or in their semantics. For example, normal modal logics differ from each other only in the set of properties of their related semantic accessibility relation [Fit83, HC68]. A new logical approach, called *Labelled Deductive System* has been proposed by Gabbay [Gab92] which, taking into account this observation, facilitates logics of the same family to have a common formalisation. Results in [DG94, BFR97] have already shown that uniform labelled proof systems can be developed for a family of substructural logics, using respectively tableaux and natural deduction proof style. This paper takes a further step. It provides a logical approach, based on Labelled Deductive System, in which logics belonging to different families, and thus characterised by different notions of derivability relation and semantic entailment, can find a uniform presentation.

Two sample logics are considered, the modal logic of inequality and the multiplicative fragment of substructural linear logic. Substructural logics are logics whose derivability relations are often described as relations between *sequences* of formulae (or assumptions) and single formulae [Dôs93]. According to the type of substructural logic, assumptions can be used only in certain specific order and a certain number of times (e.g., in linear logic assumptions are all be used in the given order and only once, whereas in relevance logic assumptions are all used at least once). On the other hand, classical logic and its extensions, such as the modal logic of elsewhere [dR92], present their notion of derivability relations in terms of a relation between *sets* of formulae (or assumptions) and single formula. In these cases, assumptions can be used in any order and an arbitrary (possibly none) number of times. The approach developed in this paper, called *compilation approach to LDS* (CLDS), provides a general presentation of derivability relation which is equally applicable to both substructural and modal logics.

In the CLDS, a logical theory written in a given *logical language* is combined with a *labelling algebra* written in a first-order *labelling language*. This is defined as a first-order theory axiomatising the properties – semantic or proof theoretic – that uniquely identify the underlying logic. In the case of the logic of elsewhere, the labelling algebra is a binary first-order theory that axiomatises the Kripke semantic accessibility relation as the inequality relation between "possible worlds". In the case of substructural logic, the labelling algebra is a binary first-order theory axiomatising standard structural rules of the underlying logic in terms of properties on the labels. The two languages (logical language and labelling language) are combined via the LDS's notion of *declarative unit*. The declarative unit $\alpha : \lambda$ expresses that the formula α is true or verified at the label (i.e. point) λ. Depending on the logic, labels are interpreted in different way. In modal logic, labels are interpreted as possible worlds, whereas in substructural logic they are interpreted as (combination of) resources. Inference rules are defined to act on both the syntactic components of the declarative units, logical formulae and labels, according to the desired properties of the connectives and of the labelling algebra.

For the logic of elsewhere, this combined approach of the $\mathrm{E}_{\mathrm{CLDS}}$ system retains the advantages of both implicit (e.g. [Fit83]) and explicit (e.g. [Ohl91]) traditional formalisations. Statements such as "necessary α" can be captured succinctly, using the modal operator $\Box$, by simply writing the single declarative unit $S_a : \Box\alpha$ (where S_a is the labelling algebra representation of the actual world). Like the explicit approach, the language is rich enough to allow explicit syntactic reference to particular possible worlds and to specific inequality or equality relationships between possible worlds. As for the substructural logic, the L_{CLDS} system facilitates an "object-level" formalisation of both the operational and structural features of the proof theory, the former by means of the logical operators and the latter by means of the labelling algebra. Label conditions expressed in the rules, together with the labelling algebra properties for handling labels, provide the proof theory with the same features as the standard structural rules of substructural logic ([Dôs93],[DG94]), but facilitating a presentation of the derivability relation in terms of a relation between sets of formulae and formulae, as in the case of modal logic.

The combined feature of $\mathrm{E}_{\mathrm{CLDS}}$ and L_{CLDS} systems has facilitated a *generalisation* of the standard formalisms of substructural and modal logics. The CLDS systems are generalisations of modal and substructural logics in that they facilitate reasoning about what is true or verified at different points in a (possibly singleton) structure of respectively actual worlds and actual resources. Information about points in a structure are expressed by *R-literals*. These are of the form $R(\lambda_i, \lambda_j)$ or $\neg R(\lambda_i, \lambda_j)$, where λ_i and λ_j are labels. In the case of modal logic they express which worlds are in relation with each other and which are not, whereas in substructural logic they express which resources are "included" with each other and which are not, where the notion of inclusion is with respect to interpretation of labels as the resources needed to "verify" a formula. A CLDS theory, called *configuration* is thus a set of declarative units and R-literals, where the R-literals specify a given structure of points (actual worlds or resources) and the declarative units describe which formulae are assumed to be verified at each point in the structure.

As mentioned before the proof system of a CLDS is a *uniform* natural deduction system in that (i) most of the inference rules have a common unique format for both the logic of elsewhere and the substructural logic – additional rules are mainly needed for the specific additional connectives $\Box$ and $\otimes$ –, (ii) for each logic the complete set of natural deduction rules can be equally applied to other logics belonging to the same family. So, for instance the set of inference rules for the logic of elsewhere can also be used for any other normal modal logic, and the rules for substructural logic are equally applicable to relevance, linear and intuitionistic[1] logics. The difference between one modal logic and another or between one substructural logic and another is captured entirely by the labelling algebra.

The paper is organised as follows. In Section 2 the language and syntax of CLDSs is defined together with the notion of a *configuration* – a CLDS system's equivalent to a theory. A basic general natural deduction style proof system for CLDSs is given in which inference rules are applied to configurations and a basic general model–theoretic semantics, based on a translation method into classical logic, is described together with the notion of a semantic entailment. In Section 3, this system is refined for the logic of "elsewhere". Additional rules are added for the specific modal operators, and for the $\wedge$ and $\vee$ operators, and the semantics refined to capture the meaning of the modal operators. Soundness and completeness results of the "Elsewhere system"($\mathrm{E}_{\mathrm{CLDS}}$) with respect to the refined semantics are proved, and it is shown that $\mathrm{E}_{\mathrm{CLDS}}$ is equivalent to standard Hilbert systems for the logic of elsewhere [dR92] whenever the initial configuration is a singleton structure. Section 4 refines the proof system and semantics given in Section 2 for the case of multiplicative linear logic. Additional rules are included for the $\otimes$ operator, as well as additional first-order axioms to capture its meaning. Soundness and completeness results of this proof system with respect to the refined semantics are proved, using the same Henkin methodology adopted for the logic of elsewhere in Section 3. Thus, a correspondence theorem is shown which states the conditions on the initial configurations under which $\mathrm{L}_{\mathrm{CLDS}}$ is equivalent to a standard sequent calculus for the fragment of linear logic considered [Dôs93]. The paper ends with a general discussion in Section 5.

Some remarks may be helpful regarding notation. Throughout the paper constant and predicate symbols begin with an upper-case letter, whereas variables and function symbols begin with a lower-case letter. Greek letters meta-variables are used to refer in general to terms and expressions in the system. Larger entities such as structures, sets, theories and

[1]The classical ($\neg\neg$) rule has to be dropped in Intuitionistic Logic.

languages are symbolised in calligraphic font, $\mathcal{A}, \mathcal{B}, \mathcal{C}$, etc.. The power set of a given set $\mathcal{A}$ is denoted by $PW(\mathcal{A})$. The general system will be referred to as the CLDS system whereas its refined versions are referred to as the $\mathrm{E}_{\mathrm{CLDS}}$ system for the logic of elsewhere and the $\mathrm{L}_{\mathrm{CLDS}}$ system for the linear logic.

2 The CLDS Approach

In this section the CLDS approach is described formally. Basic definitions of the CLDS language and syntax are given together with the notion of a *configuration* – the CLDS system's equivalent to a modal or substructural theory.

2.1 Languages and Syntax

A CLDS language is defined as an ordered pair $\langle \mathcal{L}_P, \mathcal{L}_L \rangle$, where $\mathcal{L}_L$ is a *labelling language* and $\mathcal{L}_P$ is a *propositional language* composed of a countable set of propositional letters, $\{p, q, r, \ldots\}$, a set of unary connectives $\{\sharp_1, \sharp_2, \ldots\}$ and a set of binary connectives $\{\flat_1, \flat_2, \ldots\}$. The labelling language $\mathcal{L}_L$ is a binary fragment of a first-order language composed of a countable set of constant symbols $\{s_0, s_1, s_2, \ldots\}$ a countable set of variables $\{x, y, z, \ldots\}$, a binary predicate R, a (possibly empty) finite set of function symbols $\{f_1, f_2, \ldots\}$, the set of logical connectives $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$, and the quantifiers $\forall$ and $\exists$. The first-order language $Func(\mathcal{L}_P, \mathcal{L}_L)$ is an extension of $\mathcal{L}_L$ defined as follows.

Definition 2.1 Let $\mathcal{L}_P$ be a propositional language and $\{\alpha_1, \alpha_2, \ldots\}$ be the set of all wffs of $\mathcal{L}_P$. The semi-extended labelling language $Func(\mathcal{L}_P, \mathcal{L}_L)$ is defined as the language $\mathcal{L}_L$ extended with a set of *skolem* function symbols $\{sk^n_{\alpha_1}, sk^n_{\alpha_2}, \ldots\}$, where $n \geq 0$.

The ground terms of $Func(\mathcal{L}_P, \mathcal{L}_L)$, called *labels*, are interpreted differently according to the family of logics which is under consideration. In the case of modal logics, they refer to possible worlds, whereas in the case of substructural logics they denote "resources". Analogously for the binary predicate R. R represents, in the case of modal logics, the accessibility relation between possible worlds, and in the case of substructural logics, a partial ordering of "inclusion" between resources. Labels constructed using skolem function symbols have specific roles in the CLDS proof system. As shown in Section 3, the skolem symbols of $Func(\mathcal{L}_P, \mathcal{L}_L)$ are unary function symbols of the form f_α and box_α used to denote specific possible worlds. For each wff α and possible world (label) λ, the ground term $f_\alpha(\lambda)$ names a *particular* possible world specifically associated with α which formalises the Kripke semantic notion "there exists a possible world...". In contrast, ground terms of the form box_α can be thought of as referring to any *arbitrary* world specifically associated with α. These terms will be used to express Kripke semantic notions of the form "for all possible worlds...". In the case of the $\mathrm{L}_{\mathrm{CLDS}}$ system, as shown in Section 4, the skolem symbols of $Func(\mathcal{L}_P, \mathcal{L}_L)$ are instead constant symbols, called *parameters*, of the form c_α. For each wff α of $\mathcal{L}_P$, the parameter c_α denotes the smallest resource needed to verify the formula α, and it is sometimes referred to as the *characteristic* label of α.

To capture different classes of logics within the CLDS framework an appropriate first-order theory, written in the language $\mathcal{L}_L$, called *labelling algebra* and denoted by $\mathcal{A}$, needs to be defined. For example, a standard normal modal logic can be captured by defining the labelling algebra of the CLDS system as the first-order theory axiomatising the accessibility relation [vB83]. (Examples of such CLDS systems are largely described in [Rus96].) For the logic of elsewhere, the notion of "elsewhere" expresses that worlds are accessible from each another if and only if they are different. This notion is captured in the $\mathrm{E_{CLDS}}$ system by defining the labelling algebra as a binary first-order theory with equality where the binary predicate R is the inequality relation – i.e. $\forall x, y(R(x, y) \leftrightarrow x \neq y)$. In the case of linear logic, the labelling algebra is a binary first-order theory which axiomatises (i) the binary predicate R as a pre-ordering relation and (ii) two properties, called *identity* and *order preserving* of a function symbol $\circ$ of the labelling language of the $\mathrm{L_{CLDS}}$ system. Sections 3 and 4 provide respectively formal definitions of the $\mathrm{E_{CLDS}}$ and $\mathrm{L_{CLDS}}$ labelling algebra.

Syntax. The CLDS language facilitates the formalisation of two types of information, (i) what holds at particular points and (ii) which points are in relation with each other and which are not. These two types of information are captured within the syntax of a CLDS system by two different types of syntactic entities, the *declarative units* and the *R-literals*. A declarative unit is defined as a pair *"formula:label"* expressing that a formula "holds" at a point. The label component is a ground term of the semi-extended labelling language $Func(\mathcal{L}_P, \mathcal{L}_L)$ and the formula is a wff of the language $\mathcal{L}_P$. An R-literal is any ground literal in the semi-extended labelling language of the form $R(\lambda_1, \lambda_2)$ and $\neg R(\lambda_1, \lambda_2)$, where λ_1 and λ_2 are labels, expressing that λ_2 is or is not related to λ_1. In the $\mathrm{E_{CLDS}}$ system "related to" is interpreted as "not equal to", whereas in the $\mathrm{L_{CLDS}}$ system "related to" is defined as "less than or equal to". For each R-literal Δ, the *conjugate* of Δ, written $\overline{\Delta}$, is the opposite in sign of Δ (i.e. $\neg R(\lambda_1, \lambda_2)$ if $\Delta = R(\lambda_1, \lambda_2)$ and $R(\lambda_1, \lambda_2)$ if $\Delta = \neg R(\lambda_1, \lambda_2)$).

This combined aspect of the CLDS syntax yields a definition of a CLDS theory more general than the traditional notion of a modal or substructural theory ([HC68], [Dôs93]). Informally, a CLDS theory, called a *configuration*, is composed of two sets, a set of R-literals and a set of declarative units. An example of a $\mathrm{E_{CLDS}}$ theory is the pair of sets $\{R(S_0, S_1), R(S_0, S_2), \neg R(S_1, f_p(S_1))\}$ and $\{\Box(p \rightarrow q : S_0, \Box r : S_0, \Diamond p : S_1, p : f_p(S_1), q : S_2\}$, whereas an example of a $\mathrm{L_{CLDS}}$ configuration is the pair of sets $\{R(\circ_1(c_p, c_q), S_1\}$ and $\{p \otimes q : S_1, q : c_q, p : c_p\}$. The formal definition of a configuration is as follows.

Definition 2.2 Given a CLDS language, a configuration is a tuple $\langle \mathcal{D}, \mathcal{F} \rangle$ where $\mathcal{D}$ is a finite set of R-literals and $\mathcal{F}$ is a function from the set of ground terms of *Func*$(\mathcal{L}_P, \mathcal{L}_L)$ to the set PW(wff$(\mathcal{L}_P)$) of sets of wffs of $\mathcal{L}_P$.

The $\mathcal{D}$ component of a configuration $\mathcal{C} = \langle \mathcal{D}, \mathcal{F} \rangle$ will sometimes be referred to as a *diagram* and set membership statements of the form $A \in \mathcal{F}(\lambda)$ will usually be written as $A : \lambda \in \mathcal{C}$. In the next section, a "basic" natural deduction style proof system for an arbitrary CLDS is given, in which inference rules and the notion of a derivability relation are defined between configurations. A set $\mathcal{R}$ of such inference rules, together with a CLDS language $\langle \mathcal{L}_P, \mathcal{L}_L \rangle$ and a labelling algebra $\mathcal{A}$, uniquely define a CLDS system (i.e. for any CLDS system S, $S = \langle \langle \mathcal{L}_P, \mathcal{L}_L \rangle, \mathcal{A}_S, \mathcal{R}_S \rangle$).

2.2 A "basic" natural deduction system

The "structural" aspect of a CLDS theory has stimulated the idea of defining deductive processes that describe how configurations can "evolve" by reasoning within and between the local theories associated with each point in the configuration or by reasoning about the diagram of the configuration. Inference rules and derivability relation are defined between configurations. An inference rule of a CLDS is generally defined as follows.

Definition 2.3 An inference rule $\mathcal{I}$ is a set of pairs of configurations, where each such pair is written as $\mathcal{C}/\mathcal{C}'$. If $\mathcal{C}/\mathcal{C}' \in \mathcal{I}$ then we say $\mathcal{C}$ is an *antecedent configuration* of $\mathcal{I}$, and $\mathcal{C}'$ is an *inferred (or consequence) configuration of $\mathcal{I}$ with respect to $\mathcal{C}$.*

All the rules except one have the effect of expanding the antecedent configuration. These rules can extend an antecedent configuration $\mathcal{C}$ with either a declarative unit, or with an R-literal or with both. However, configurations equal or smaller than the antecedent one can also be inferred. This is facilitated by an inference rule called the $\mathcal{C}$-Reduction (C-R) rule. Tables 1 and 2 provide a schematic representation of the inference rules for the connectives $\rightarrow$ and $\neg$, and for the R-literals respectively, and, because of space limitation, mathematical definitions are given only for some of these rules. These rules have the same format in both the $\mathrm{E_{CLDS}}$ and the $\mathrm{L_{CLDS}}$ systems, since the standard semantic difference between the classical $\rightarrow$ and $\neg$ and the substructural $\rightarrow$ and $\neg$ is captured by the labels included in the rules. In Sections 3.1 and 4.2 this set of rules is extended to include elimination and introduction rules for the modal operators $\Box$ and $\Diamond$ and elimination and introduction rules for the substructural operator $\otimes$ respectively. For the R-literals three additional rules are defined. One of these (the (R-A) rule) facilitates first-order derivations of relationships between points in the configuration using the labelling algebra $\mathcal{A}$. For logics of the same family (i.e. different substructural logics or different modal logics), this rule captures entirely the difference between one CLDS system and another, allowing all other inference rules to be equally applicable to any CLDS system. For logics belonging to different families, such as the logic of elsewhere and linear logic, the (R-A) rule allows different reasoning on the R-literals which capture the different specific semantics of the logic. Most of the mathematical formalisation of the $\mathrm{E_{CLDS}}$ system is described in [Rus96] and [BR97a] includes a full mathematical formalisation of the additions for $\mathrm{E_{CLDS}}$ and $\mathrm{L_{CLDS}}$ systems.

Informally, a proof is a non empty sequence of configurations, $\mathcal{C}_0, \ldots, \mathcal{C}_n$, where, for each $0 < i \leq n$, $\mathcal{C}_i$ is obtained from $\mathcal{C}_{i-1}$ by the application of an inference rule. A configuration $\mathcal{C}'$ is said to be *derivable* from a configuration $\mathcal{C}$, written $\mathcal{C} \vdash_{\mathrm{CLDS}} \mathcal{C}'$, if and only if there exists a proof $\mathcal{C}, \ldots, \mathcal{C}'$. This is formally defined below.

Definition 2.4 Given a CLDS system S, a *proof* is a pair $\langle \mathcal{P}, m \rangle$, where $\mathcal{P}$ is a sequence of configurations $\{\mathcal{C}_0, \ldots, \mathcal{C}_n\}$, with $n > 0$, and m is a mapping from the set $\{0, \ldots, n-1\}$ to $\mathcal{R}_S$ such that for each i, $0 \leq i < n$, $\mathcal{C}_i/\mathcal{C}_{i+1} \in m(i)$.

Definition 2.5 [Derivability of a CLDS system]
Given a CLDS system S, and two configurations $\mathcal{C}$ and $\mathcal{C}'$, $\mathcal{C}'$ is *derivable* from $\mathcal{C}$ in S, written $\mathcal{C} \vdash_{\mathrm{S}} \mathcal{C}'$, if there exists a proof $\langle \{\mathcal{C}, \ldots, \mathcal{C}'\}, m \rangle$.

Notation 2.1 *Let $\mathcal{C} = \langle \mathcal{D}, \mathcal{F} \rangle$ be a configuration and π be either a declarative unit or an R-literal, $\mathcal{C} \vdash_{\mathrm{CLDS}} \pi$ if there exists a configuration $\mathcal{C}'$ such that $\mathcal{C} \vdash_{\mathrm{CLDS}} \mathcal{C}'$ and $\pi \in \mathcal{C}'$. Moreover, if π is a declarative unit of the form $\alpha : \lambda$ then $\mathcal{C} + [\alpha : \lambda]$ is the configuration $\langle \mathcal{D}, \mathcal{F}' \rangle$, such that $\mathcal{F}'(\lambda) = \mathcal{F}(\lambda) \cup \{\alpha\}$ and for any λ' different from λ, $\mathcal{F}'(\lambda') = \mathcal{F}(\lambda')$. If π is an R-literal Δ, then $\mathcal{C} + [\Delta]$ is the configuration $\langle \mathcal{D}', \mathcal{F} \rangle$ such that $\mathcal{D}' = \mathcal{D} \cup \{\Delta\}$.*

It is easy to show that the derivability relation $\vdash_{\mathrm{S}}$ of a CLDS system S is reflexive, transitive and monotonic, (for a proof see [Rus96]). Notation 2.1 captures the standard notion of a derivability relation between theories (configurations) and formulae (declarative units or R-literals) in terms of the more general derivability relation given in Definition 2.5. A "vice-versa" characterisation can be shown – a configuration $\mathcal{C}'$ is derivable from a configuration $\mathcal{C}$ if each unit of information (declarative units and R-literals) of $\mathcal{C}'$ is derivable from $\mathcal{C}$. This is proved in the following lemma.

Lemma 2.1 Let $\mathcal{C}$ and $\mathcal{C}'$ be two configurations of a CLDS system S, such that $\mathcal{C}' - \mathcal{C}$ is finite. $\mathcal{C} \vdash_{\mathrm{S}} \mathcal{C}'$ if and only if for each $\pi \in \mathcal{C}' - \mathcal{C}$, $\mathcal{C} \vdash_{\mathrm{S}} \pi$, where π is a declarative unit or an R-literal.

Proof: The "only if" part is trivial, whereas the "if" part is proved by induction on the size of $\mathcal{C}' - \mathcal{C}$. A formal description of this proof is given in [Rus96].

□

Two examples of formally defined inference rules for a CLDS system are now given.

Definition 2.6 For all configurations $\mathcal{C}$, terms λ_1,λ_2 and λ_3, and wffs α and β, $\mathcal{C}/\mathcal{C} + [\alpha \rightarrow \beta : \lambda_3]$ is a member of the inference rule $\rightarrow$-Introduction (sometimes written $\rightarrow\mathcal{I}$) if $\mathcal{C} + [\alpha : \lambda_1] \vdash_{\mathrm{CLDS}} \beta : \lambda_2$.

Definition 2.7 For all configurations $\mathcal{C}$, terms λ_1,λ_2 and λ_3 and wff α, $\mathcal{C}/\mathcal{C} + [\neg\alpha : \lambda_3]$ is a member of the inference rule $\neg$-Introduction (sometimes written $\neg\mathcal{I}$) if $\mathcal{C} + [\alpha : \lambda_1] \vdash_{\mathrm{CLDS}} \perp : \lambda_2$.

Before giving the schematic representation of the inference rules, some remarks are essential to clarify such representation. For any configuration $\mathcal{C}$, the informal notation $\mathcal{C}\langle\alpha : \lambda\rangle$ (respectively $\mathcal{C}\langle\Delta\rangle$) denotes that $\mathcal{C}$ includes a declarative unit $\alpha{:}\lambda$ (respectively R–literal Δ). Declarative units and R–literals contained in square brackets (see e.g. the $\rightarrow\mathcal{I}$ rule) are assumptions introduced within a derivation that are subsequently discharged (sometimes called *temporary assumptions*). The notation $\mathcal{C}'\langle\psi\rangle$ represents that the inferred configuration $\mathcal{C}'$ is $\mathcal{C}$ extended with the declarative unit or R–literal ψ. In the introduction rules, the $\tilde{\mathcal{C}}$ and $\overline{\mathcal{C}}$ are the configurations derived in subderivations after adding to the antecedent configuration $\mathcal{C}$ the temporary assumptions.

According to the type of logic the labels λ_1, λ_2 and λ_3 that appear in the rules have different characteristics. In the $\mathrm{E_{CLDS}}$ system, they are defined to be the same labels, i.e. $\lambda_1 = \lambda_2 = \lambda_3 = \lambda$, for any arbitrary term λ of the extended labelling language, whereas in the $\mathrm{L_{CLDS}}$ system they are defined to be of a particular form. This is formally defined in Section 4.1. Analogously for the symbol $\perp$. The notion of contradiction (or inconsistency) in the CLDS framework strictly depends on the type of logic. Modal logics (and therefore the logic of elsewhere) include a classical notion of inconsistency. Thus, in the ($\neg\mathcal{I}$) rule of the $\mathrm{E_{CLDS}}$ system, the symbol $\perp$ is a short-hand for any wff of $\mathcal{L}_P$ of the form $\alpha \wedge \neg\alpha$. Substructural logics instead respect a different notion of contradiction, according to which the declarative $\perp : \lambda$ leads to an inconsistency only when the label λ denotes a "consistent resource". This is further explained in Section 4.4.

In the ($\rightarrow\mathcal{I}$) rule, $\tilde{\mathcal{C}}$ is the configuration derived after adding the assumption $\alpha : \lambda_1$ to the antecedent configuration, to get the declarative unit $\beta : \lambda_3$. In the $\neg\mathcal{I}$ rule, $\tilde{\mathcal{C}}$ is the configuration derived after adding the assumption $\alpha : \lambda_1$ to $\mathcal{C}$ to get a contradiction – the declarative unit $\perp{:}\lambda_2$.

Table 1: Natural deduction rules for $\rightarrow$ and $\neg$ connectives.

$\dfrac{\mathcal{C}\langle \alpha\rightarrow\beta:\lambda_1, \alpha:\lambda_2\rangle}{\mathcal{C}'\langle \beta:\lambda_3\rangle}$	$(\rightarrow\mathcal{E})$	$\dfrac{\begin{array}{c}\mathcal{C}\langle[\alpha:\lambda_1]\rangle\\ \vdots \\ \tilde{\mathcal{C}}\langle \beta:\lambda_2\rangle\end{array}}{\mathcal{C}'\langle \alpha\rightarrow\beta:\lambda_3\rangle}$	$(\rightarrow\mathcal{I})$
$\dfrac{\mathcal{C}\langle \neg\neg\alpha:\lambda\rangle}{\mathcal{C}'\langle \alpha:\lambda\rangle}$	$(\neg\neg)$	$\dfrac{\begin{array}{c}\mathcal{C}\langle[\alpha:\lambda_1]\rangle\\ \vdots \\ \tilde{\mathcal{C}}\langle \bot:\lambda_2\rangle\end{array}}{\mathcal{C}'\langle \neg\alpha:\lambda_3\rangle}$	$(\neg\mathcal{I})$

To allow reasoning about arbitrary configurations and to capture the different CLDS systems (given by the different labelling algebra), a second set of inference rules is needed to be included as part of a propositional CLDS system. These rules facilitate reasoning about the diagram of a configuration, using the particular labelling algebra $\mathcal{A}$ under consideration, and infer R-literals and declarative units which are not implied by the logical connectives. A schematic representation of these rules is given in Table 2. In Section 1, it has been emphasised that different labelling algebras define different propositional CLDS systems. Proof theoretically, these differences are imposed by the (R-A) rule. This rule facilitates the inference of new R-literals according to the properties of the underlying logic axiomatised by the particular first-order labelling algebra $\mathcal{A}$. In [Rus96] examples derivations are given for the different labelling algebrae associated with the different normal modal logics and it is shown how the CLDS approach (called in [Rus96] MLDS) facilitates the development of a uniform proof system for any normal modal logic. In [BFR97] a similar approach, also based on LDS, is described in which a common set of natural deduction rules are developed for a given family of substructural logics. The difference between one logic and another in [BFR97], is not

Table 2: Rules for the R-literals

$(\bot\mathcal{E})$	$\dfrac{\mathcal{C}\langle \Delta, \overline{\Delta}\rangle}{\mathcal{C}'\langle \alpha:\lambda\rangle}$	(C-R)	$\dfrac{\mathcal{C}}{\mathcal{C}'}$ where $\mathcal{C}' \subseteq \mathcal{C}$
$(R\mathcal{I})$	$\dfrac{\begin{array}{c}\mathcal{C}\langle[\ \overline{\Delta}\ ]\rangle\\ \vdots \\ \tilde{\mathcal{C}}\langle \bot:\lambda\rangle\end{array}}{\mathcal{C}'\langle \Delta\rangle}$	(R-A)	$\dfrac{\mathcal{C}}{\mathcal{C}'\langle \Delta\rangle}$ if $\mathcal{A}\cup\mathcal{D}\vdash_{FOL}\Delta$

embedded explicitly in the derivation process in terms of inference rules, but it is captured by algorithmic processes which solve labels constraints generated by the derivation taking into account labels properties defined by the associated labelling algebra.

In the E_{CLDS} system, where the labelling algebra axiomatises the symmetry property of the R predicate, the R-A rule allows, for instance, the inference of R-literals of the form $R(\lambda_2, \lambda_1)$ whenever the antecedent configuration includes R-literals of the form $R(\lambda_1, \lambda_2)$, thus embedding the symmetry property of the accessibility relation in the derivation process. This enables the derivation of declarative units of the form $\alpha \rightarrow \Box\Diamond\alpha : \lambda$, for any arbitrary label λ. In the L_{CLDS} system, the R-A rule allows, for instance, the inference of R-literals of the form $R(\circ(\lambda_1, \lambda_2), \circ(\lambda_2, \lambda_1))$ where the symbol $\circ$ is a binary function symbol of the language $\mathcal{L}_L$ in L_{CLDS}, denoting the "composition" of resources, thus embedding the "commutativity" property on resources [DG94, BFR97] in the derivation process. Such a property enables for instance the derivations of declarative units of the form $\alpha \otimes \beta \rightarrow \beta \otimes \alpha : \lambda$, for the particular label $\lambda = 1$, which is a theorem of linear logic.

The rules $(R\mathcal{I})$ and $(\bot\mathcal{E})$ express additional forms of interactions between the R-literals and the declarative units. The $(\bot\mathcal{E})$ rule allows the inference of falsity (i.e. $\bot : \lambda$) whenever R-literals and its negations are present in a configuration. This is necessary because since no compound classical formulae with R-literals can be inferred in a configuration, inconsistency of this form would not otherwise be captured. The (R-I) rule enables the derivation of R-literals in the presence of a logical inconsistency. It is the analogue of the $(\neg\mathcal{I})$ rule for R-literals. The structural rules described above and the rules given in Table 1 all have the effect of expanding their antecedent configurations. The (C-R) rule is then included in every propositional CLDS system to simply allow the derivation of any configuration contained in the antecedent one.

2.3 Semantics

A propositional CLDS can be considered to be a "semi-translated" approach to a given logic – in the case of E_{CLDS} system for instance, a Kripke-like accessibility relation is syntactically expressed, but without requiring the full translation of modal formulae into first-order sentences. In the L_{CLDS} system on the other hand, the CLDS facilitates the "meta-level" features of the underlying logic to be formalised as part of the object-level proof system. This also can be seen as a semi-translated approach to linear logic, which still preserves the concise aspect of the logical substructural language. Therefore, a model-theoretic semantics could be equally given in terms of the traditional semantics of the underlying logic (i.e. Kripke semantics [HC68] for the logic of elsewhere and algebraic semantics [Dôs93, DG94, BDR97] for linear logic) or in terms of a first-order semantics using a translation method. The latter enables the development of model-theoretic approach which is equally applicable to any logics, also belonging to different families, whose operators have a semantics which can be expressed into a first-order theory.

This second approach has been chosen here. In this section, a translation method of a CLDS system into first-order logic is defined and the notions of model, satisfiability of a configuration and semantic entailment are then given in terms of classical semantics.

As already pointed out above, a declarative unit $\alpha : \lambda$ represents that the formula is verified (or holds) at the point λ, whose interpretation is strictly related to the type of underlying logic. In what follows, these notions are expressed in terms of first-order statements of the form $[\alpha]^*(\lambda)$, where $[\alpha]^*$ is a predicate symbol. The relationships between these predicate symbols are constrained by a set of first-order axiom schemas which capture the satisfiability conditions of each type[2] of formula α. The *extended labelling algebra* $Mon(\mathcal{L}_P, \mathcal{L}_L)$ is an extension of the language $Func(\mathcal{L}_P, \mathcal{L}_L)$ given by adding a monadic predicate symbol $[\alpha]^*$ for each wff α of $\mathcal{L}_P$. This is formally defined below.

Definition 2.8 Let $Func(\mathcal{L}_P, \mathcal{L}_L)$ be a semi-extended labelling language. Let $\alpha_1, \ldots, \alpha_n, \ldots,$ be the ordered set of wffs of $\mathcal{L}_P$. The *extended labelling language* $Mon(\mathcal{L}_P, \mathcal{L}_L)$ is defined as the language $Func(\mathcal{L}_P, \mathcal{L}_L)$ extended with the following set of unary predicate symbols

$$\{[\alpha_1]^*, \ldots, [\alpha_n]^*, \ldots\}$$

An *extended algebra* $\mathcal{A}^+$ is a first-order theory written in $Mon(\mathcal{L}_P, \mathcal{L}_L)$ which extends a labelling algebra $\mathcal{A}$ with axiom schemas on the monadic predicates. These schemas strictly depend on the underlying logic. For example, in the $\mathrm{E_{CLDS}}$ system the extended labelling algebra $\mathcal{A}^+$, includes the two axiom schemas $\forall x([\Box\alpha]^*(x) \rightarrow (\forall y(R(x,y) \rightarrow [\alpha]^*(y))))$ and $\forall x((R(x, box_\alpha(x)) \rightarrow [\alpha]^*(box_\alpha(x))) \rightarrow [\Box\alpha]^*(x))$ which together with the axiom $\forall x, s(R(x,s) \leftrightarrow x \neq s)$ capture the Kripke semantic meaning of the elsewhere $\Box$ operator. Formal definitions of the extended labelling algebras $\mathcal{A}^+$ for the $\mathrm{E_{CLDS}}$ and $\mathrm{L_{CLDS}}$ systems are given in Sections 3 and 4 respectively.

However, the notions of satisfiability and semantic entailment are common to any CLDS. These are based on a translation method which associates syntactic expressions of the CLDS system with sentences of the first-order language $Mon(\mathcal{L}_P, \mathcal{L}_L)$, and hence associates theories (configurations) with first-order theories in the language $Mon(\mathcal{L}_P, \mathcal{L}_L)$. Each declarative unit $\alpha:\lambda$ is translated into the sentence $[\alpha]^*(\lambda)$, and R-literals are translated as themselves. Therefore, the first-order translation of a configuration is a first-order theory including the R-literals, which are present in the diagram of the configuration, and the set of monadic formulae $[\alpha]^*(\lambda)$ that correspond to the declarative units present in the configuration. A formal definition is given below.

Definition 2.9 Let $\mathcal{C} = \langle \mathcal{D}, \mathcal{F} \rangle$ be a configuration. The *first-order translation* of $\mathcal{C}$, written $FOT(\mathcal{C})$, is a theory written in $Mon(\mathcal{L}_P, \mathcal{L}_L)$ and defined by the expression:

$$FOT(\mathcal{C}) = \mathcal{D} \cup \mathcal{DU}$$

where $\mathcal{DU} = \{[\alpha]^*(\lambda) \mid \alpha \in \mathcal{F}(\lambda),\ \lambda \text{ is a ground term of } Func(\mathcal{L}_P, \mathcal{L}_L)\}$.

Note that since labels can only be ground terms of the language $Func(\mathcal{L}_P, \mathcal{L}_L)$, the first-order translation of a configuration is a set of *ground literals* of the language $Mon(\mathcal{L}_P, \mathcal{L}_L)$. Notions of model, satisfiability and semantic entailment are given in terms of classical semantics using the above definition, as follows (where "$\mathcal{M} \Vdash_{FOL} \psi$" signifies that the classical formula ψ is true in the classical model $\mathcal{M}$, according to the standard definition).

Definition 2.10 Given a CLDS system S, the associated extended algebra $\mathcal{A}^+_S$, a declarative unit $\alpha:\lambda$ and a R–literal Δ,

$$\mathcal{M} \text{ is a semantic structure of } S \;\;\Leftrightarrow_{\text{def}}\;\; \mathcal{M} \text{ is a model of } \mathcal{A}^+_S \tag{1}$$

$$\mathcal{M} \Vdash_S \alpha:\lambda \;\;\Leftrightarrow_{\text{def}}\;\; \mathcal{M} \Vdash_{FOL} [\alpha]^*(\lambda) \tag{2}$$

$$\mathcal{M} \Vdash_S \Delta \;\;\Leftrightarrow_{\text{def}}\;\; \mathcal{M} \Vdash_{FOL} \Delta \tag{3}$$

[2]The type of a wff is given by the main connective of the wff itself, e.g., the wff $\Diamond(p \rightarrow q)$ is a $\Diamond$-formula, whereas the formulae $\alpha \otimes (\beta \rightarrow \gamma)$ is a $\otimes$-formula.

In the above definition, (1) defines the class of models of a CLDS system S in terms of models of the extended algebra $\mathcal{A}_S^+$ associated with S. (2) and (3) define the satisfiability of declarative units and R–literals in terms of classical satisfiability of their associated first–order translations. A semantic structure $\mathcal{M}$ satisfies a configuration $\mathcal{C}$, written $\mathcal{M} \Vdash_S \mathcal{C}$, if and only if for each $\pi \in \mathcal{C}$ (where π may be a declarative unit or an R–literal), $\mathcal{M} \Vdash_S \pi$. The notion of semantic entailment in a CLDS system is given here as a relation between configurations. It is formally defined as follows.

Definition 2.11 Let $S = \langle\langle \mathcal{L}_P, \mathcal{L}_L, \rangle, \mathcal{A}, \mathcal{R}\rangle$ be a CLDS and let $\mathcal{A}^+$ be the extended algebra of S. Let $\mathcal{C} = \langle \mathcal{D}, \mathcal{F}\rangle$ and $\mathcal{C}' = \langle \mathcal{D}', \mathcal{F}'\rangle$ be two configurations of S and $FOT(\mathcal{C}) = \mathcal{D} \cup \mathcal{DU}$ and $FOT(\mathcal{C}') = \mathcal{D}' \cup \mathcal{DU}'$ their respective first-order translations. The configuration $\mathcal{C}$ semantically entails $\mathcal{C}'$, written $\mathcal{C} \models_S \mathcal{C}'$, iff for each $\Delta \in \mathcal{D}'$, $\mathcal{A}^+ \cup FOT(\mathcal{C}) \models_{FOL} \Delta$, and for each $[\alpha]^*(\lambda) \in \mathcal{DU}'$, $\mathcal{A}^+ \cup FOT(\mathcal{C}) \models_{FOL} [\alpha]^*(\lambda)$.

In Sections 3.1 and 4.2 the above definition expresses the notion of semantic entailment for the $\mathrm{E_{CLDS}}$ and L_{CLDS} systems, which are respectively denoted by $\models_{\mathrm{E}}$ and $\models_{\mathrm{L}}$ respectively. Soundness and completeness proofs will also be given for the specific $\mathrm{E_{CLDS}}$ and $\mathrm{L_{CLDS}}$ system with respect to these two notions of semantic entailment. Although the extended labelling algebrae associated with these two systems are different from each other (because of the different axiom schemas) the same methodology is used to prove the above results. A general description of this methodology is given here.

The soundness and completeness properties of the proof system of a given CLDS system S with respect to the above semantics consists in showing that the derivability relation $\vdash_S$ is equivalent to the semantic entailment $\models_S$. This means to prove that whenever there exists a natural deduction proof of a configuration $\mathcal{C}'$ from a configuration $\mathcal{C}$ then $\mathcal{C}$ semantically entails $\mathcal{C}'$, and vice-versa.

Proving soundness. The soundness theorem is proved by using a technique that differs from the standard technique used in the literature for proving soundness of a natural deduction proof system. In general (see [Fit83] for an example of standard soundness proof for natural deduction proof systems) such property is proved by induction on the number of inference steps in the given derivation, taking into account the specific *context* of each inference rule. In this paper instead we define the notions of size of an inference rule and size of a proof, and apply induction on the size of a given derivation. In this way there is no difference (apart from the size) between the inference rules that introduce new assumptions and those which do not introduce new assumptions. These notions are formally defined below, using the following additional notation.

Notation 2.2 *Given a CLDS system S, its set $\mathcal{R}$ of inference rules is classified into four categories. The first category denoted with $\mathcal{I}^{00}$ includes just the C-R rule, as being the only rule which does not infer new declarative units or new R-literals. The second category, denoted by $\mathcal{I}^0$, consists of the inference rules that infer new declarative units and/or new R-literals without using any subderivations as conditions. The third category, denoted by $\mathcal{I}^+$, is the set of inference rules that require one subderivation as a condition. Finally, the fourth category, denoted by $\mathcal{I}^{++}$, is the set of inference rules that use two subderivations as conditions.*

◁

In both the E_{CLDS} and L_{CLDS} systems the $(\rightarrow\mathcal{E})$ and $(\neg\neg)$ rules are examples of members of the $\mathcal{I}^0$ category, whereas $(\rightarrow\mathcal{I})$ and $(\neg\mathcal{I})$ rules are examples of members of $\mathcal{I}^+$ category. The $\mathcal{I}^{++}$ category instead is composed only of the $(\vee\mathcal{E})$ rule in the E_{CLDS} system and it is empty in the L_{CLDS} system.

Definition 2.12 Let S be a CLDS system, let $\mathcal{I}_i \in \mathcal{R}_S$ and let $\mathcal{C}/\mathcal{C}' \in \mathcal{I}_i$. The *size* of $\mathcal{C}/\mathcal{C}'$ with respect to $\mathcal{I}_i$, written $l(\mathcal{C}/\mathcal{C}', \mathcal{I}_i)$, is defined as follows:

- If $\mathcal{I}_i \in \mathcal{I}^{00}$ then $l(\mathcal{C}/\mathcal{C}', \mathcal{I}_i) = 0$.
- If $\mathcal{I}_i \in \mathcal{I}^{0}$ then $l(\mathcal{C}/\mathcal{C}', \mathcal{I}_i) = 1$.
- If $\mathcal{I}_i \in \mathcal{I}^{+}$ then $l(\mathcal{C}/\mathcal{C}', \mathcal{I}_i) = 1 + l_1$, where l_1 is the smallest of the sizes of all subderivations that can be used as condition of the rule.
- If $\mathcal{I}_i \in \mathcal{I}^{++}$ then $l(\mathcal{C}/\mathcal{C}', \mathcal{I}_i) = 1 + l_1 + l_2$, where l_1 and l_2 are the smallest of the sizes of all the two subderivations that can be used as conditions of the rule.

Definition 2.13 Let S be a CLDS system, the *size of a proof* $\langle\{\mathcal{C} \ldots \mathcal{C}_n\}, m\rangle$, written $l(\langle\{\mathcal{C} \ldots \mathcal{C}_n\}, m\rangle)$, is defined as follows

$$l(\langle\{\mathcal{C}_0, \ldots, \mathcal{C}_n\}, m\rangle) = \sum_{k=0}^{n-1} l(\mathcal{C}_k/\mathcal{C}_{k+1}, m(k))$$

Given that the semantics is based on a first-order translation method, the proof of the soundness property of the $\vdash_{\text{CLDS}}$ is based on the soundness property of the first-order classical derivability relation $\vdash_{FOL}$. A diagrammatic representation of the soundness theorem of a CLDS system S is given in Figure 1.

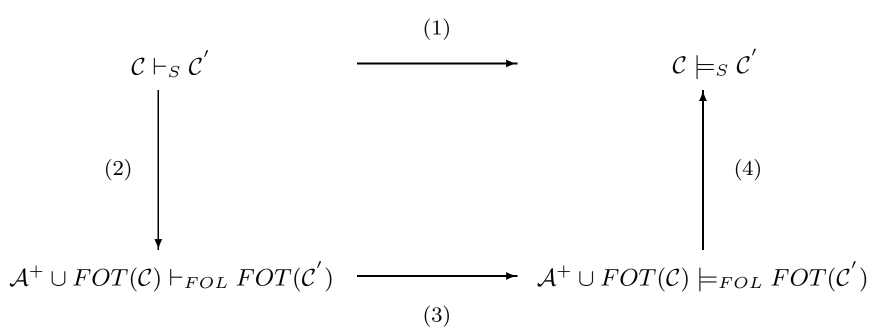

Figure 1: Proof of the soundness property of a CLDS system $\mathcal{C}$.

The soundness statement, which corresponds to the arrow labelled with (1), is proved by the composition of three main steps, arrows (2), (3) and (4) respectively. The first step (arrow (2)) proves that the hypothesis, $\mathcal{C} \vdash_S \mathcal{C}'$ for a CLDS system S, implies that $\mathcal{A}^+ \cup FOT(\mathcal{C}) \vdash_{FOL} FOT(\mathcal{C}')$. This trivially implies (by soundness of first-order logic) that $\mathcal{A}^+ \cup FOT(\mathcal{C}) \models_{FOL} FOT(\mathcal{C}')$, which gives the second step of the proof (arrow (3)). Arrow (4) is given by the definition of the semantic entailment between configurations given in Definition 2.11. Note that this methodology is generally applicable to any CLDS system. The first step is the only

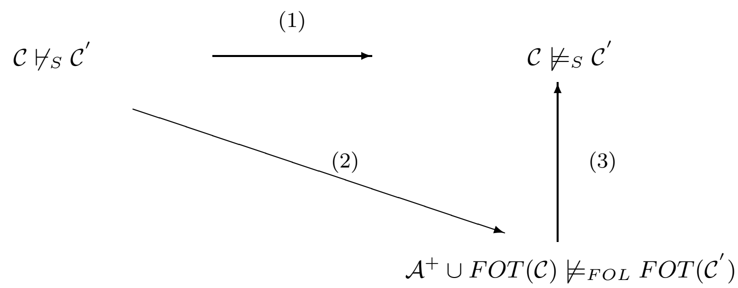

Figure 2: Proof of the completeness property of a CLDS system $\mathcal{C}$.

one that needs to be proved for each specific logic formalised in CLDS. Soundness of the $\mathrm{E}_{\mathrm{CLDS}}$ and $\mathrm{L}_{\mathrm{CLDS}}$ systems are proved respectively in Sections 3.2 and 4.3 on the basis of this methodology.

Proving completeness. The completeness property of a CLDS system with respect to the semantics described in Section 2.3 can be proved using standard Henkin-style methodology [HC68]. The theorem states that, given a CLDS system S and two configurations $\mathcal{C}$ and $\mathcal{C}'$ such that $\mathcal{C}' - \mathcal{C}$ is a finite[3], if $\mathcal{C}'$ is semantically entailed from $\mathcal{C}$ then $\mathcal{C}'$ is also derived from $\mathcal{C}$. $\mathcal{C}' - \mathcal{C}$ (formally defined in [Rus96]) is basically the set of declarative units and R-literals if $\mathcal{C}'$ but not in $\mathcal{C}$. The methodology adopted to prove the completeness of a CLDS system is diagrammatically represented in Figure 2 and it can be informally described as follows. The proof is of the contrapositive statement (arrow (1)), which states that, given a CLDS S system and two configurations $\mathcal{C}$ and $\mathcal{C}'$ such that $\mathcal{C}' - \mathcal{C}$ is a finite, if $\mathcal{C} \nvdash_{\mathrm{CLDS}} \mathcal{C}'$ then $\mathcal{C} \not\models_{\mathrm{CLDS}} \mathcal{C}'$. This is proved by the composition of two main steps, arrows (2) and (3). Arrow (3) is already given by Definition 2.11, while arrow (2) represents the main part of the theorem.

The proof of arrow (2) is based on the statement *if $\mathcal{C}$ is a consistent configuration then $\mathcal{C}$ is satisfiable*, known as the "Model Existence Lemma" and consists of the following reasoning steps. Note that the definition of a consistent configuration strictly depends on the CLDS system.

- The hypothesis that $\mathcal{C}'$ is not derivable from $\mathcal{C}$, $\mathcal{C} \nvdash_{\mathrm{CLDS}} \mathcal{C}'$, implies that there exists a $\pi \in \mathcal{C}' - \mathcal{C}$ (where π is a declarative unit or an R-literal) such that $\mathcal{C} \nvdash_{\mathrm{CLDS}} \pi$. This is shown in Lemma 2.1.
- The above step implies that the configuration $\mathcal{C}$ extended with $\neg\pi$ (written $\mathcal{C} + [\neg\pi]$) is a consistent configuration. This is shown by Proposition 3.1 and Lemma 4.2 for the $\mathrm{E}_{\mathrm{CLDS}}$ and $\mathrm{L}_{\mathrm{CLDS}}$ system respectively.
- The second step implies then that the configuration $\mathcal{C} + [\neg\pi]$ is satisfiable. This is proved by means of Lemmas 3.5 and 4.12 which are respectively the model existence lemma of the $\mathrm{E}_{\mathrm{CLDS}}$ system and the model existence lemma of the $\mathrm{L}_{\mathrm{CLDS}}$ system. Therefore, there exists a semantic structure $\mathcal{M}$ of the CLDS system S which satisfies $\mathcal{C}$ and that also satisfies $\neg\pi$. It is then shown that $\mathcal{M}$ does not satisfy π. Thus, since $\pi \in \mathcal{C}'$, by definition

[3]Obviously, if the configuration difference $\mathcal{C}' - \mathcal{C}$ were infinite, an infinite proof sequence would be required to prove $\mathcal{C}'$ from $\mathcal{C}$.

of satisfiability of a configuration, $\mathcal{M}$ does not satisfy $\mathcal{C}'$. Hence $\mathcal{A}^+ \cup FOT(\mathcal{C}) \not\models_{\text{FOL}} FOT(\mathcal{C}')$.

3 The E_{CLDS} system

In this section the E_{CLDS} system is formally described, on the basis of the CLDS approach defined in Section 2. The E_{CLDS} language is defined as the ordered pair $\langle \mathcal{L}_P, \mathcal{L}_L \rangle$, where $\mathcal{L}_P$ is a propositional language composed of a countable set of propositional letters $\{p, q, r, \ldots\}$, the set of classical connectives $\{\vee, \wedge, \neg, \rightarrow\}$ and the set of modal operators $\{\Box, \Diamond\}$. The labelling language $\mathcal{L}_L$ is a first-order language composed of a countable set of constant symbols $\{S_0, S_1, S_2, \ldots\}$, a countable set of variables $\{x, y, z, \ldots\}$, the binary predicates R and $=$ and the set of logical connectives and quantifiers.

The labelling language $\mathcal{L}_L$ is extended into a new language, the semi-extended abelling language $Func(\mathcal{L}_P, \mathcal{L}_L)$, to include two sets of special unary function symbols.

Definition 3.1 Let $\mathcal{L}_P$ be the E_{CLDS} propositional language and let $\{\alpha_1, \alpha_2, \ldots\}$ be the set of all wffs of $\mathcal{L}_P$. The first–order language $Func(\mathcal{L}_P, \mathcal{L}_L)$ is defined as the language $\mathcal{L}_L$ extended with the sets of unary function symbols $\{f_{\alpha_1}, f_{\alpha_2}, \ldots\}$ and $\{box_{\alpha_1}, box_{\alpha_2}, \ldots\}$.

As mentioned in Section 2, terms of the form $f_\alpha(\lambda)$ will be used to express Kripke semantic notions of the form "there exists a possible world ...", whereas terms of the form $box_\alpha(\lambda)$ will be used to expressed Kripke semantic notions of the form "for all possible worlds ...". However, formally speaking $f_\alpha(\lambda)$ and $box_\alpha(\lambda)$ are just terms of $\mathcal{L}_L$ and within a particular model might even refer to the same possible world. The whole set of ground terms of $Func(\mathcal{L}_P, \mathcal{L}_L)$ defines the set of *labels* of the E_{CLDS} system. Since they denote actual and accessible worlds, as explained in Section 2, the expressions "labels" and "possible worlds" will be used interchangeably throughout this section.

Syntax. The predicate $=$ is introduced in the labelling language $\mathcal{L}_L$ in order to capture the meaning of the Kripke semantic accessibility relation of the logic of elsewhere. Within this logic, possible worlds are accessible from each other if and only if they are not equal [dR92, Dem96]. This notion of accessibility relation is formalised by the labelling algebra $\mathcal{A}$ defined later in this section. Syntactically, the E_{CLDS} language facilitates the formalisation of three types of information, (i) what holds at particular possible worlds, (ii) which worlds are in relation with each other and which are not and (iii) which worlds are equal to each other and which are not. Whereas the first type of information is captured by the *declarative units* (defined in Section 2), the last two types of information are captured by the following extended definition of R-literals.

Definition 3.2 [R-literals] Let $\langle \mathcal{L}_P, \mathcal{L}_L \rangle$ be the E_{CLDS} language. An R-literal is any literal of the form $R(\lambda_1, \lambda_2)$, $\neg R(\lambda_1, \lambda_2)$, $= (\lambda_1, \lambda_2)$ and $\neg(= (\lambda_1, \lambda_2))$, where λ_1 and λ_2 are ground terms of the language $Func(\mathcal{L}_P, \mathcal{L}_L)$. The last two types of literals are often written in their infix form $\lambda_1 = \lambda_2$ and $\lambda_1 \neq \lambda_2$, where the expression $\neq (\lambda_1, \lambda_2)$ is a shorthand for the wff $\neg(= (\lambda_1, \lambda_2))$.

The notion of *conjugate* of an R-literal, introduced in Section 2 is also extended to the R-literals constructed from the $=$ predicate. So, for any R-literal Δ of the form $\lambda_1 = \lambda_2$, the conjugate $\overline{\Delta}$ is equal to $\lambda_1 \neq \lambda_2$ and vice-versa, for any R-literal Δ of the form $\lambda_1 \neq \lambda_2$, $\overline{\Delta}$ is equal to $\lambda_1 = \lambda_2$.

The syntax of the E_{CLDS} system allows arbitrary sets of modal formulae to be associated with (different) labels, describing not only one initial set of local assumptions (as in the implicit approach of the logic of elsewhere [dR92]) but allowing for several (distinct) local initial modal theories to be specified. With the addition of R-literals, these local theories can be stated to be related to the same possible world or to different possible worlds and therefore interacting with each other. This yields a definition of a E_{CLDS} theory more general than the traditional notion of a modal theory given in [HC68, Fit83, dR92]. This feature contributes towards the long term aim of this work of providing a modal formalism closer to the needs of application (see discussions in Section 5).

A E_{CLDS} configuration, or theory, is composed of two sets of information, (i) a set of R-literals and (ii) a set of declarative units. Sets of declarative units having the same labels denote local modal theories associated with that label (possible world), whereas declarative units having different labels express modal formulae belonging to possibly different local actual worlds. By means of the R-literals based on the $=$ predicate, it is possible to specify within an E_{CLDS} theory which local actual worlds are different or equal with each other. In the specific case of the logic of elsewhere this information is semantically equivalent to the information of two possible worlds being accessible or not accessible from each other, which is instead syntactically formalised by the R-literals constructed from the binary predicate R. The semantic equivalence between these two types of R-literals, typical of the logic of elsewhere, is syntactically captured by the labelling algebra $\mathcal{A}$ of the E_{CLDS} system, which allows the inference of inequalities between possible worlds stated to be in relation with each other, and vice-versa, the inference of relations between possible worlds that are not equal. The two types of R-literals are here introduced mainly to preserve the generality of the CLDS approach (see discussion in Section 5).

So far, notions of E_{CLDS} language, syntax and theory have been defined. These are "basic" concepts in the sense that they are common to all CLDS propositional modal systems. (See [Rus96] for the CLDS systems corresponding to the most well known normal modal logics.) However, as shown in Section 2, there is an essential component which needs to be defined to uniquely characterise the E_{CLDS} system, the *labelling algebra*. The class of Kripke frames associated with the logic of elsewhere is declaratively formalised by a first-order axiomatisation, called *labelling algebra*, written in the language $Func(\mathcal{L}_P, \mathcal{L}_L)$ including the standard first-order logic equality theory [CL73]. This is composed of three main schemas, namely the reflexivity axiom $\forall x(x = x)$ and the two equality substitution axioms for terms and predicate respectively. For the specific cases of terms constructed from the function symbol f_α and of the binary predicate symbol R the last two axiom schemas are respectively of the form (i) $\forall x, y[x = y \rightarrow (f_\alpha(x) = f_\alpha(y))]$, and (ii) $\forall x, y, z[x = y \rightarrow (R(x, z) \rightarrow R(y, z))]$. It is easy to show that the reflexivity axiom together with equality substitution axiom for the binary predicate $=$ proves the symmetry and transitivity properties of the $=$ predicate. The labelling algebra of the E_{CLDS} system is formally defined as follows.

Definition 3.3 [Labelling algebra $\mathcal{A}_{\text{E}}$] The *labelling algebra* $\mathcal{A}_{\text{E}}$ is the first-order theory, written in language $Func(\mathcal{L}_P, \mathcal{L}_L)$, given by the standard equality theory extended with the following axiom:

$$\forall x, y(R(x, x) \leftrightarrow (x \neq y)) \qquad (\mathbf{E})$$

Axiom (**E**) expresses the meaning of the Kripke accessibility relation in the specific case of the logic of elsewhere, for which only possible worlds which are different from a given possible world are accessible from it. The $\mathrm{E}_{\mathrm{CLDS}}$ system is then defined by the tuple $\langle\langle\mathcal{L}_P, \mathcal{L}_L\rangle, \mathcal{A}_{\mathrm{E}}, \mathcal{R}_{\mathrm{E}}\rangle$ where the set of inference rules $\mathcal{R}_{\mathrm{E}}$ is defined in the following section.

3.1 Proof theory and Semantics of the $\mathbf{E}_{\mathrm{CLDS}}$

To give a full definition of the $\mathrm{E}_{\mathrm{CLDS}}$ system it is necessary to specify the set of inference rules for the classical and modal operators of the language $\mathcal{L}_P$, as well as the inference rules for reasoning about relationships, equality and inequality between possible worlds.

In the $\mathrm{E}_{\mathrm{CLDS}}$ system, given an antecedent configuration $\mathcal{C}$ three types of reasoning step can occur. Those of the first type are "classical", and occur within any particular local modal theory included in $\mathcal{C}$, respecting standard notions of inference for classical connectives. A specialised version of the introduction and elimination rules for the $\rightarrow$ and $\neg$ operators given in Table 1 are rules for the classical operators $\rightarrow$ and $\neg$ of the $\mathrm{E}_{\mathrm{CLDS}}$ system. This specialised version requires that in each of these rules the labels λ_1, λ_2 and λ_3 are all the same. The set of rules for the $\vee$ and $\wedge$ classical connectives is given in Table 3, together with a special rule called $\mathcal{I}_{Sub}$. The $\mathcal{I}_{Sub}$ expresses a specific form of interaction between the R-literals constructed from the $=$ predicate and the declarative units included in a configuration. This interaction is similar to an equality substitution property for declarative units. Any modal CLDS systems whose configurations include equality (and inequality) literals ought to have this rule as part of their natural deduction system. (See [Rus96] for examples of other CLDS systems of this kind.) Note that, with respect to the notion of length of an inference rule given n Section 2, the $\mathcal{I}_{Sub}$ is a inference rule with length equal to 1 – i.e. $\mathcal{I}_{Sub} \in \mathcal{I}^0$.

Table 3: Natural deduction rules for classical connectives.

$$\frac{\mathcal{C}\langle\alpha\wedge\beta:\lambda\rangle}{\mathcal{C}'\langle\alpha:\lambda\rangle}\ (\wedge\mathcal{E}) \qquad \frac{\mathcal{C}\langle\alpha:\lambda, \beta:\lambda\rangle}{\mathcal{C}'\langle\alpha\wedge\beta:\lambda\rangle}\ (\wedge\mathcal{I})$$

$$\frac{\mathcal{C}\langle\alpha\vee\beta:\lambda\rangle \quad \begin{matrix}\mathcal{C}\langle[\alpha:\lambda]\rangle \\ \vdots \\ \tilde{\mathcal{C}}\langle\gamma:\lambda\rangle\end{matrix} \quad \begin{matrix}\mathcal{C}\langle[\beta:\lambda]\rangle \\ \vdots \\ \overline{\mathcal{C}}\langle\gamma:\lambda\rangle\end{matrix}}{\mathcal{C}'\langle\gamma:\lambda\rangle}\ (\vee\mathcal{E}) \qquad \frac{\mathcal{C}\langle\alpha:\lambda\rangle}{\mathcal{C}'\langle\alpha\vee\beta:\lambda\rangle}\ (\vee\mathcal{I})$$

$$\frac{\mathcal{C}\langle\alpha:\lambda, \lambda=\lambda'\rangle}{\mathcal{C}'\langle\alpha:\lambda'\rangle}\ (\mathcal{I}_{Sub})$$

Rules of the second type are "modal" and concern the interaction between different modal theories in $\mathcal{C}$. These are given in Table 4. The $\Diamond\mathcal{E}$ rule can be seen (informally) as a "skolemization" of the existential quantifier over possible worlds which is semantically implied

Table 4: Natural deduction rules for modal operators.

$\dfrac{\mathcal{C}\langle\lambda:\Diamond\alpha\rangle}{\mathcal{C}^{'}\langle f_\alpha(\lambda):\alpha, R(\lambda, f_\alpha(\lambda))\rangle}$	$(\Diamond\mathcal{E})$	$\dfrac{\mathcal{C}\langle\lambda_2:\alpha, R(\lambda_1,\lambda_2)\rangle}{\mathcal{C}^{'}\langle\lambda_1:\Diamond\alpha\rangle}$	$(\Diamond\mathcal{I})$
$\dfrac{\mathcal{C}\langle\lambda_1:\Box\alpha, R(\lambda_1,\lambda_2)\rangle}{\mathcal{C}^{'}\langle\lambda_2:\alpha\rangle}$	$(\Box\mathcal{E})$	$\dfrac{\begin{array}{c}\mathcal{C}\langle[R(\lambda, box_\alpha(\lambda))]\rangle\\ \vdots\\ \tilde{\mathcal{C}}\langle box_\alpha(\lambda):\alpha\rangle\end{array}}{\mathcal{C}^{'}\langle\lambda:\Box\alpha\rangle}$	$(\Box\mathcal{I})$

by the formula $\Diamond\alpha$ in the premise. The term $f_\alpha(\lambda)$ defines a particular possible world uniquely associated with the formula α, and inferred to be accessible from the possible world λ (i.e. $R(\lambda, f_\alpha(\lambda))$). It is clear from the definition that this rule has the effect of expanding both the components (diagram and set of declarative units) of the antecedent configuration. In the $\Box\mathcal{I}$ rule, the temporary assumption $R(\lambda, box_\alpha(\lambda))$ should be read as "given an arbitrary world accessible from λ", using then the term $box_\alpha(\lambda)$ not to name particular objects (possible worlds), as normally done, but to refer to an arbitrary object. This role of $box_\alpha(\lambda)$ will become clearer when the semantics of the E_{CLDS} system is defined. Because of the semantic equivalence between the predicates R and $\neq$, stated in the labelling algebra $\mathcal{A}_{\text{E}}$, each R-literal of the form $R(\lambda_i, \lambda_j)$ that appears in the rules for modal operators can be equally read as "λ_i different from λ_j". So for instance, the $(\Box\mathcal{E})$ rule allows any formula α to be inferred from a declarative unit $\Box\alpha:\lambda_1$ at any possible world λ_2 different from λ_1. This reflects the semantic meaning of the $\Box$ operator in the logic of elsewhere[4].

Both classical and modal reasoning steps are based on the logical (classical and modal) information (wffs) incorporated in the declarative units that belong to $\mathcal{C}$. The third type of reasoning step is instead related to the diagram information in $\mathcal{C}$ and to the "interaction" between the diagram and the declarative units. In this case, inferred configurations are often "structural expansions" of (i.e. additions of R-literals to) the antecedent configurations. These are identical to those given in Table 2, but with the meta-variables Δ and $\overline{\Delta}$ referring respectively to the notions of R-literal given in Definition 3.2 and its conjugate. The rule R-A in particular, facilitates reasoning about the diagram of a configuration, using the specific labelling algebra $\mathcal{A}_{\text{E}}$. For instance, from the R-A rule and the (**E**) axiom of the labelling algebra $\mathcal{A}_{\text{E}}$, it is possible to prove the symmetry property of the accessibility relation R using the symmetry property of the $\neq$ predicate, itself given by the symmetry of the equality predicate $=$ included in $\mathcal{A}_{\text{E}}$. A example graphical representation of a derivation is given in Figure 3 which shows the proof of the modal theorem $\alpha \rightarrow \Box\Diamond\alpha$, at the initial world S_0.

Semantics. The semantics of the E_{CLDS} system is based on the model theoretic semantics defined in Section 2.3 for a general CLDS system. As pointed out in Section 2.3, in the E_{CLDS} system a declarative unit $\alpha : \lambda$ represents that the formula α holds at the possible world

[4]The syntactic inference of λ_2 being different from λ_1 (i.e. $\lambda_1 \neq \lambda_2$) is given by an application of the (R-A) rule, given in Table 2, using the (**E**) axiom of the labelling algebra $\mathcal{A}_{\text{E}}$.

$\mathcal{C}_\emptyset\langle\rangle$

$\mathcal{C}\langle[\alpha : S_0]\rangle$ (assumption)

$\mathcal{C}\langle[R(S_0, box_{\diamond\alpha}(S_0))]\rangle$ (assumption)

$\mathcal{C}_1\langle S_0 \neq box_{\diamond\alpha}(S_0)\rangle$ (R-A)

$\mathcal{C}_2\langle box_{\diamond\alpha}(S_0) \neq S_0\rangle$ (R-A)

$\mathcal{C}_3\langle R(box_{\diamond\alpha}(S_0), S_0)\rangle$ (R-A)

$\mathcal{C}_4\langle \diamond\alpha : box_{\diamond\alpha}(S_0)\rangle$ ($\diamond\mathcal{I}$)

$\mathcal{C}_5\langle \Box\diamond\alpha : S_0\rangle$ ($\Box\mathcal{I}$)

$\mathcal{C}'\langle \alpha \rightarrow \Box\diamond\alpha : S_0\rangle$ ($\rightarrow\mathcal{I}$)

Figure 3: Example derivation in the $\mathrm{E_{CLDS}}$ system.

λ. This is expressed in terms of first-order statements of the form $[\alpha]^*(\lambda)$, where $[\alpha]^*$ is a unary predicate symbol. Here, the language $Func(\mathcal{L}_P, \mathcal{L}_L)$ is further extended to include the predicate symbol $[\alpha]^*$ for each wff α of $\mathcal{L}_P$. The resulting language is called *extended labelling language* and denoted with $Mon(\mathcal{L}_P, \mathcal{L}_L)$.

The relationships between these unary predicates $[\alpha]^*$ are constrained by a set of first-order axiom schemas which capture the satisfiability conditions for each type of formula α. These axiom schemas extend the labelling algebra $\mathcal{A}_{\mathrm{E}}$ of the $\mathrm{E_{CLDS}}$ system into a first-order theory called an *extended algebra*. A formal definition is given below. Note that, in the extended algebra the equality substitution schema is extended to each predicate symbol $[\alpha]^*$.

Definition 3.4 [Extended algebra $\mathcal{A}_{\mathrm{E}}^{+}$] Given the extended labelling language $Mon(\mathcal{L}_P, \mathcal{L}_L)$ and the labelling algebra $\mathcal{A}^+$ of the $\mathrm{E_{CLDS}}$ system, the *extended algebra* $\mathcal{A}_{\mathrm{E}}^{+}$ is the first-order theory in $Mon(\mathcal{L}_P, \mathcal{L}_L)$ consisting of the following axiom schemas (Ax1)-(Ax8) together with the axiom (**E**) and the standard equality axioms of $\mathcal{A}_{\mathrm{E}}$ extended to the monadic predicates $[\alpha]^*$. For any wffs α and β of $\mathcal{L}_P$:

$$\forall x([\alpha \wedge \beta]^*(x) \leftrightarrow ([\alpha]^*(x) \wedge [\beta]^*(x))) \qquad \text{(Ax1)}$$

$$\forall x([\neg\alpha]^*(x) \leftrightarrow \neg[\alpha]^*(x)) \qquad \text{(Ax2)}$$

$$\forall x([\alpha \vee \beta]^*(x) \leftrightarrow ([\alpha]^*(x) \vee [\beta]^*(x))) \qquad \text{(Ax3)}$$

$$\forall x([\alpha \rightarrow \beta]^*(x) \leftrightarrow ([\alpha]^*(x) \rightarrow [\beta]^*(x))) \qquad \text{(Ax4)}$$

$$\forall x([\Diamond\alpha]^*(x) \rightarrow (R(x, f_\alpha(x)) \wedge [\alpha]^*(f_\alpha(x)))) \qquad \text{(Ax5)}$$

$$\forall x(\exists y(R(x, y) \wedge [\alpha]^*(y)) \rightarrow [\Diamond\alpha]^*(x)) \qquad \text{(Ax6)}$$

$$\forall x((R(x, box_\alpha(x)) \rightarrow [\alpha]^*(box_\alpha(x))) \rightarrow [\Box\alpha]^*(x)) \qquad \text{(Ax7)}$$

$$\forall x([\Box\alpha]^*(x) \rightarrow (\forall y(R(x, y) \rightarrow [\alpha]^*(y)))) \qquad \text{(Ax8)}$$

The first four axiom schemas express the distributive properties of the logical connectives among the monadic predicates of $Mon(\mathcal{L}_P, \mathcal{L}_L)$. (Ax5) and (Ax8) force the accessibility relation R on labels generated by the application of the function symbols f_α and box_α of $Mon(\mathcal{L}_L, \mathcal{L}_M)$ whenever their respective antecedents hold. Clearly, all the axiom schemas (Ax1)–(Ax8) reflect the traditional Kripke semantic definition of satisfiability of modal wffs[5]. The axiom schemas (Ax5)–(Ax8) together with the axiom (**E**) of the labelling algebra $\mathcal{A}_{\mathrm{E}}$ express the *specific* semantic meaning of the modal operators $\Diamond$ and $\Box$ for the logic of elsewhere. Axiom schemas (Ax5),(Ax6) and (**E**) together express the semantic definition of $\Diamond$ operator for which a formula $\Diamond\alpha$ is true at a possible world λ if and only if there exists a possible world $\lambda' \neq \lambda$ where α is true. Similarly, axiom schemas (Ax7),(Ax8) and (**E**) express the semantic definition of $\Box$ operator, for which a formula $\Box\alpha$ is true at a possible world λ if and only if for all possible worlds $\lambda' \neq \lambda$, α is true at λ'.

The notions of satisfiability and semantic entailment of the $\mathrm{E}_{\mathrm{CLDS}}$ system are as specified in Definitions 2.10 and 2.11, but based on the extended algebra $\mathcal{A}^+_{\mathrm{E}}$ defined above. These notions are based on the translation method defined in Section 2 which translates a configuration of the $\mathrm{E}_{\mathrm{CLDS}}$ system into a first-order theory including the R-literals present in the diagram of the configuration, and the set of monadic formulae $[\alpha]^*(\lambda)$ that correspond to the declarative units $\alpha:\lambda$ present in the configuration.

3.2 Main results about the $\mathrm{E}_{\mathrm{CLDS}}$ system.

In this section it is shown that the natural deduction proof system of $\mathrm{E}_{\mathrm{CLDS}}$ is sound and complete with respect to the semantics given in Section 3.1, and that, *under certain restrictions*, corresponds to the standard axiomatisation of the logic of elsewhere given in [dR92].

Soundness and Completeness results.

The soundness and completeness proofs , based respectively on the two methodologies described in Section 2.3, take advantage of the soundness and completeness of first-order logic. Most of the theorems, lemmas and propositions used to prove the soundness and completeness properties extend those given in [Rus96] for the class of CLDS normal modal logics systems, with the additional cases corresponding to the equality and inequality between possible worlds and to the special rule $\mathcal{I}_{Sub}$ for equality substitution in declarative units. Hence, the proofs described in this paper will consider only these extended cases, referring the reader to [Rus96] for the remaining full proofs.

[5] This is easily seen by interpreting the truth of $[\alpha]^*(x)$ as the truth of the modal formula α in the possible world x.

Theorem 3.1 [Soundness of the $\mathrm{E_{CLDS}}$ system]
Let $E = \langle\langle \mathcal{L}_P, \mathcal{L}_L\rangle, \mathcal{A}^+_{\mathrm{E}}, \mathcal{R}_{\mathrm{E}}\rangle$ be a the $\mathrm{E_{CLDS}}$ system and let $\mathcal{C}$ and $\mathcal{C}'$ be two configurations. If $\mathcal{C} \vdash_{\mathrm{E}} \mathcal{C}'$ then $\mathcal{C} \models_{\mathrm{E}} \mathcal{C}'$.

The proof of Theorem 3.1 is represented diagrammatically in Figure 1, and by exploiting the soundness property of first-order logic, it suffices to prove that for any pair of configurations $\mathcal{C}$ and $\mathcal{C}'$, if $\mathcal{C} \vdash_{\mathrm{E}} \mathcal{C}'$ then $\mathcal{A}^+_{\mathrm{E}}, FOT(\mathcal{C}) \vdash_{\mathrm{FOL}} FOT(\mathcal{C}')$. This is captured by the following lemma.

Lemma 3.1 [Soundness with respect to translation] Let $\mathcal{A}^+_{\mathrm{E}}$ be the extended algebra of the $\mathrm{E_{CLDS}}$ system. Let $\mathcal{C}$ and $\mathcal{C}'$ be two configurations and let $FOT(\mathcal{C}$ and $FOT(\mathcal{C}^+)$ be their respective first-order translations. If $\mathcal{C} \vdash_{\mathrm{E}} \mathcal{C}'$ then $\mathcal{A}^+_{\mathrm{E}}, FOT(\mathcal{C}) \vdash_{\mathrm{FOL}} FOT(\mathcal{C}')$.

Proof: The proof is by induction of the smallest size of derivations of the form $\langle\{\mathcal{C}_0, \ldots, \mathcal{C}_n\}, \overline{m}\rangle$, where $\mathcal{C}_0 = \mathcal{C}$ and $\mathcal{C}_n = \mathcal{C}'$. In what follows $\langle\{\mathcal{C}_0, \ldots, \mathcal{C}_n\}, m\rangle$ is a proof of this smallest size with length $l \geq 0$. The base case is when $l = 0$. This means by Definition 2.13 that $\mathcal{C}' \subseteq \mathcal{C}$ from which the theorem trivially follows. The inductive step is proved by cases on the last inference rule of the derivation (assuming that this is different from the $\mathcal{C}$-R rule[6], since the theorem holds by inductive hypothesis for the remaining first part of the derivation. Hence, it is sufficient to show that $\mathcal{A}^+_{\mathrm{E}}, FOT(\mathcal{C}_{n-1}) \vdash_{\mathrm{FOL}} FOT(\mathcal{C}_n)$.

For Cases 1–15, corresponding to the inference rules given in Tables 1, 2, 3 and 4 except the $\mathcal{I}_{Sub}$, the reader is referred to [Rus96].

Case 16: $\mathcal{I}_{Sub}$.
In this case $\mathcal{C}_{n-1}/\mathcal{C}_n \in \mathcal{I}_{Sub}$. Then there exist a declarative unit $\alpha : \lambda$ and an R-literal of the form $\lambda = \lambda'$ such that $\{\alpha : \lambda, \lambda = \lambda'\} \subseteq \mathcal{C}$. Therefore, the set $\{[\alpha]^*(\lambda), \lambda = \lambda'\} \subseteq FOT(\mathcal{C})$. This implies, by applying the equality substitution axiom of $\mathcal{A}^+_{\mathrm{E}}$ to the predicate $[\alpha]^*$, that $\mathcal{A}^+_{\mathrm{E}}, FOT(\mathcal{C}_{n-1}) \vdash_{\mathrm{FOL}} [\alpha]^*(\lambda')$. Since $\mathcal{C}_n = \mathcal{C}_{n-1} + [\alpha : \lambda']$, $\mathcal{A}^+_{\mathrm{E}}, FOT(\mathcal{C}_{n-1}) \vdash_{\mathrm{FOL}} FOT(\mathcal{C}_n)$.
□

Proof of Theorem 3.1:
By hypothesis $\mathcal{C} \vdash_{\mathrm{E}} \mathcal{C}'$. By Lemma 3.1 $\mathcal{A}^+_{\mathrm{E}}, FOT(\mathcal{C}) \vdash_{\mathrm{FOL}} FOT(\mathcal{C}')$. By soundness of first-order logic $\mathcal{A}^+_{\mathrm{E}}, FOT(\mathcal{C}) \models_{\mathrm{FOL}} FOT(\mathcal{C}')$. Hence, by definition of semantic entailment, $\mathcal{C} \models_{\mathrm{E}} \mathcal{C}')$.

Completeness. The completeness theorem states that, given the $\mathrm{E_{CLDS}}$ system and two configurations $\mathcal{C}$ and $\mathcal{C}'$, if $\mathcal{C}'$ is semantically entailed from $\mathcal{C}$ then $\mathcal{C}'$ is also derived from $\mathcal{C}$. This is formally defined below.

Theorem 3.2 [Completeness of the $\mathrm{E_{CLDS}}$ system]
Let $E = \langle\langle \mathcal{L}_P, \mathcal{L}_L\rangle, \mathcal{A}^+_{\mathrm{E}}, \mathcal{R}_{\mathrm{E}}\rangle$ be a $\mathrm{E_{CLDS}}$ system and let $\mathcal{C}$ and $\mathcal{C}'$ be two configurations such that $\mathcal{C}' - \mathcal{C}$ is a finite configuration. $\mathcal{C} \vdash_{\mathrm{E}} \mathcal{C}'$ then $\mathcal{C} \models_{\mathrm{E}} \mathcal{C}'$.

[6]Note that the lemma can trivially be proved to hold for proofs obtained by extending the one considered here with a $\mathcal{C}$-R rule on the last step.

As discussed in Section 2.3, the Henkin-style methodology is here adopted to prove Theorem 3.2. The basic notions in this methodology are that of *consistent* and *maximal consistent* configurations.

Definition 3.5 [Consistent configuration] Given the E_{CLDS} system and a configuration $\mathcal{C}$, $\mathcal{C}$ is *inconsistent* if $\mathcal{C} \vdash_{\text{E}} \bot : \lambda$ for some ground term λ of $Func(\mathcal{L}_P, \mathcal{L}_L)$. $\mathcal{C}$ is *consistent* if it is not inconsistent.

Definition 3.6 [Maximal Consistent Configuration] Given the E_{CLDS} system, a configuration $\mathcal{C}_{\text{mcc}}$ is a *maximally consistent* configuration of E_{CLDS}, if (i) it is consistent and if (ii) for any $\pi \notin \mathcal{C}_{\text{mcc}}$ (where π is a declarative unit or an R-literal), the configuration $\mathcal{C}_{\text{mcc}} + [\pi]$ is not consistent.

The proof of Theorem 3.2 is then based on constructing a *canonical* semantic structure of the E_{CLDS} system to show the contrapositive statement that

$$\mathcal{C} \not\vdash_{\text{E}} \mathcal{C}' \text{ implies } \mathcal{C} \not\models_{\text{E}} \mathcal{C}' \tag{4}$$

In standard Henkin-style proofs of completeness for modal logics, the canonical model is obtained by progressively building maximal consistent sets (see for example [HC68]) where consistency is locally checked according to the properties of the underlying accessibility relation. In our approach, the explicit declarative representation of possible worlds and of relationships between possible worlds facilitates the construction of a canonical model for the E_{CLDS} system by simply extending a given consistent configuration into a single maximal consistent configuration where consistency is then checked globally. This is shown in the following lemma.

Lemma 3.2 Given the E_{CLDS} system, every consistent configuration $\mathcal{C}$ can be extended to a maximal consistent configuration $\mathcal{C}_{\text{mcc}}$.

Proof: Let $\mathcal{C}$ be a consistent configuration and let $\pi_1, \pi_2, \pi_3, \ldots, \pi_n, \ldots$ be an ordering on the set of all declarative units and R-literals of $\langle \mathcal{C}_P, \mathcal{L}_L \rangle$. Starting from $\mathcal{C}_0 = \mathcal{C}$ a sequence of consistent configurations $\mathcal{C}_i$ is constructed by inductively defining, for each element π, $\mathcal{C}_i$ to be

$$\begin{array}{ll} \mathcal{C}_i = \mathcal{C}_{i-1} + [\pi] & \text{if } \mathcal{C}_{i-1} + [\pi_i] \text{ consistent} \\ \mathcal{C}_i = \mathcal{C}_{i-1} & \text{otherwise} \end{array}$$

It is easy to show that for each $i \geq 0$, $\mathcal{C}_i$ is a consistent configuration. Now, let $\mathcal{C}_{\text{mcc}}$ be the configuration obtained from the union of all the $\mathcal{C}_i$, $\mathcal{C}_{\text{mcc}} = \cup_{i \geq 0} \mathcal{C}_i$. It is easy to show that $\mathcal{C}_{\text{mcc}}$ is maximal and consistent (see Proposition 3.8 in [Rus96] for a formal proof). Hence $\mathcal{C}_{\text{mcc}}$ is a maximal consistent configuration.

□

Maximal consistent configurations are configurations whose declarative units and R-literals satisfy particular properties. These are listed in the following lemma where only the properties related to R-literals constructed from the = predicate are proved, referring the reader to [Rus96] for a formal proof of the remaining cases.

Lemma 3.3 Let $\mathcal{C}_{\text{mcc}}$ be a maximal consistent configuration of E_{CLDS}. Then for any π (where π is a declarative unit or an R-literal) and for any wffs α and β,

1. π and $\overline{\pi}$ are not both in $\mathcal{C}_{\text{mcc}}$.
2. Either $\pi \in \mathcal{C}_{\text{mcc}}$ or $\overline{\pi} \in \mathcal{C}_{\text{mcc}}$.
3. $\alpha \wedge \beta : \lambda \in \mathcal{C}_{\text{mcc}}$ if and only if $\alpha : \lambda \in \mathcal{C}_{\text{mcc}}$ and $\beta : \lambda \in \mathcal{C}_{\text{mcc}}$.
4. $\alpha \vee \beta : \lambda \in \mathcal{C}_{\text{mcc}}$ if and only if $\alpha : \lambda \in \mathcal{C}_{\text{mcc}}$ or $\beta : \lambda \in \mathcal{C}_{\text{mcc}}$.
5. $\alpha \rightarrow \beta : \lambda \in \mathcal{C}_{\text{mcc}}$ if and only if if $\alpha : \lambda \in \mathcal{C}_{\text{mcc}}$ then $\beta : \lambda \in \mathcal{C}_{\text{mcc}}$.
6. If $\Box\alpha : \lambda \in \mathcal{C}_{\text{mcc}}$ and $R(\lambda, \lambda^{'}) \in \mathcal{C}_{\text{mcc}}$ then $\alpha : \lambda^{'} \in \mathcal{C}_{\text{mcc}}$.
7. If $\neg R(\lambda, box_\alpha(\lambda)) \in \mathcal{C}_{\text{mcc}}$ or $\alpha : box_\alpha(\lambda) \in \mathcal{C}_{\text{mcc}}$, then $\Box\alpha : \lambda \in \mathcal{C}_{\text{mcc}}$.
8. If $\Diamond\alpha : \lambda \in \mathcal{C}_{\text{mcc}}$ then $R(\lambda, f_\alpha(\lambda)) \in \mathcal{C}_{\text{mcc}}$ and $\alpha : f_\alpha(\lambda) \in \mathcal{C}_{\text{mcc}}$.
9. If $R(\lambda, \lambda^{'}) \in \mathcal{C}_{\text{mcc}}$ and $\alpha : \lambda^{'} \in \mathcal{C}_{\text{mcc}}$, then $\Diamond\alpha : \lambda \in \mathcal{C}_{\text{mcc}}$.
10. $R(\lambda, \lambda^{'}) \in \mathcal{C}_{\text{mcc}}$ if and only if $\lambda \neq \lambda^{'} \in \mathcal{C}_{\text{mcc}}$.
11. If $\lambda = \lambda_1 \in \mathcal{C}_{\text{mcc}}$ and $R(\lambda, \lambda^{'}) \in \mathcal{C}_{\text{mcc}}$ then $R(\lambda_1, \lambda^{'}) \in \mathcal{C}_{\text{mcc}}$.
12. If $\lambda = \lambda_1 \in \mathcal{C}_{\text{mcc}}$ and $\lambda = \lambda^{'} \in \mathcal{C}_{\text{mcc}}$ then $\lambda_1 = \lambda^{'} \in \mathcal{C}_{\text{mcc}}$.
13. If $\alpha : \lambda \in \mathcal{C}_{\text{mcc}}$ and $\lambda = \lambda^{'} \in \mathcal{C}_{\text{mcc}}$ then $\alpha : \lambda^{'} \in \mathcal{C}_{\text{mcc}}$.

Proof:
Property (1) is proved only for the case of π equal to $\lambda = \lambda^{'}$ for arbitrary labels λ, $\lambda^{'}$. Suppose that both $\lambda = \lambda^{'}$ and $\lambda \neq \lambda^{'}$ are in $\mathcal{C}_{\text{mcc}}$. Then by definition of the $\bot\mathcal{E}$ rule $\mathcal{C}_{\text{mcc}} \vdash_{\text{E}} \bot : \lambda$ which constradicts the hypothesis $\mathcal{C}_{\text{mcc}}$ being a maximal consistent configuration.

Property (2) is also proved only for the case of π equal to $\lambda = \lambda^{'}$ for arbitrary labels λ, $\lambda^{'}$. Suppose that neither $\lambda = \lambda^{'}$ nor $\lambda \neq \lambda^{'}$ is in $\mathcal{C}_{\text{mcc}}$. Then by definition of maximality $\mathcal{C}_{\text{mcc}} + [\lambda = \lambda^{'}] \vdash_{\text{E}} \bot : \lambda$ and $\mathcal{C}_{\text{mcc}} + [\lambda \neq \lambda^{'}] \vdash_{\text{E}} \bot : \lambda^{'}$. Then it is easy to show (see [Rus96]) that there exist two configurations $\mathcal{C}_1$ and $\mathcal{C}_2$ such that $\mathcal{C}_1 \subseteq \mathcal{C}_{\text{mcc}}$, $\mathcal{C}_1 + [\lambda = \lambda^{'}] \vdash_{\text{E}} \bot : \lambda$, $\mathcal{C}_2 \subseteq \mathcal{C}_{\text{mcc}}$ and $\mathcal{C}_2 + [\lambda \neq \lambda^{'}] \vdash_{\text{E}} \bot : \lambda^{'}$. By monotonicity of the E_{CLDS} derivability relation, also the configuration $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2$ is such that $\mathcal{C} \subseteq \mathcal{C}_{\text{mcc}}$, $\mathcal{C} + [\lambda = \lambda^{'}] \vdash_{\text{E}} \bot : \lambda$ and $\mathcal{C} + [\lambda \neq \lambda^{'}] \vdash_{\text{E}} \bot : \lambda^{'}$. This implies that there exists a derivation which shows that $\mathcal{C} \vdash_{\text{E}} \bot : \lambda^{'}$, which, by the monotonicity of $\vdash_{\text{E}}$ contradicts the hypothesis that $\mathcal{C}_{\text{mcc}}$ is consistent.

Property (11) and (12), that express the deduction closure of the maximal consistent configuration with respect the equality substitution in R-literals are also proved by contradiction using the (R-A) rule. Similarly for Property (10). Property (13) is proved by contradiction using $\mathcal{I}_{Sub}$ rule. For all the other cases, the reader is referred to [Rus96].

□

To prove the maximal consistent lemma it is essential to define the notion of a canonical semantic structure. This is given with respect to a maximal consistent configuration and a notion of a canonical interpretation.

Definition 3.7 [Canonical Interpretation] Let $\mathcal{C}_{\text{mcc}} = \langle \mathcal{D}_{\text{mcc}}, \mathcal{F}_{\text{mcc}} \rangle$. be a maximal consistent configuration of E_{CLDS} and let $FOT(\mathcal{C}_{\text{mcc}})$ be its first-order translation. A canonical semantic structure of E_{CLDS} is the pair $(\mathcal{U}, \mathcal{I}_{\text{mcc}})$, where $\mathcal{U}$ is the Herbrand universe of the language $Mon(\mathcal{L}_P, \mathcal{L}_L)$ and $\mathcal{I}_{\text{mcc}}$ is an interpretation function on the language $Mon(\mathcal{L}_P, \mathcal{L}_L)$ defined as follows.

- For each ground term λ,
  $\parallel \lambda \parallel_{I_{\text{mcc}})} = \lambda$.

- For the binary predicate R,
  $\| R \|_{I_{\text{mcc})}} = \{\langle\lambda_i, \lambda_j\rangle \mid R(\lambda_i, \lambda_j) \in FOT(\mathcal{C}_{\text{mcc}})\}$ [7].
- For the binary predicate $=$,
  $\|{=}\|_{I_{\text{mcc})}} = \{\langle\lambda_i, \lambda_j\rangle \mid \lambda_i = \lambda_j \in FOT(\mathcal{C}_{\text{mcc}})\}$
- For each monadic predicate $[\alpha]^*$,
  $\| [\alpha]^* \|_{I_{\text{mcc}}} = \{\lambda_i \mid [\alpha]^*(\lambda_i) \in FOT(\mathcal{C}_{mcc})\}$

The following lemma shows that the canonical interpretation constructed in the above definition is a canonical semantic structure of the E_{CLDS} system

Lemma 3.4 [Canonical Semantic Structure] Let $\mathcal{C}_{\text{mcc}}$ be a maximal consistent configuration of E_{CLDS} and let $(\mathcal{U}, \mathcal{I}_{\text{mcc}})$ be a canonical interpretation. Then $(\mathcal{U}, \mathcal{I}_{\text{mcc}})$ is a canonical semantic structure of E_{CLDS}.
Proof: To show that $(\mathcal{U}, \mathcal{I}_{\text{mcc}})$ is a canonical semantic structure of E_{CLDS} it is needed to show, by Definition 2.10, that $(\mathcal{U}, \mathcal{I}_{\text{mcc}})$ is a model of the extended algebra $\mathcal{A}^+_{\text{E}}$. This means to show that $(\mathcal{U}, \mathcal{I}_{\text{mcc}})$ is a model of the axioms (Ax1)–(Ax8) as well as of the axioms of the equality theory included in $\mathcal{A}^+_{\text{E}}$. This is easy to prove by using Lemma 3.3 and the consideration that, by definition of the canonical interpretation, for each declarative unit and R-literal π, $(\mathcal{U}, \mathcal{I}_{\text{mcc}}) \Vdash FOT(\pi)$ if and only if $FOT(\pi) \in FOT(\mathcal{C}_{\text{mcc}})$.

□

It is now possible to prove the *Model Existence Lemma* for the E_{CLDS} system.

Lemma 3.5 [Model Existence Lemma] Let $\mathcal{C}_{\text{mcc}}$ be a maximal consistent configuration, and let $\mathcal{M}_{\text{mcc}} = (\mathcal{U}, \mathcal{I}_{\text{mcc}})$ be a canonical semantic structure of E_{CLDS}. Then for any π (where π is a declarative unit or an R-literal) of E_{CLDS}, $\mathcal{M}_{\text{mcc}} \models_{\text{E}} \pi$ if and only if $\pi \in \mathcal{C}_{\text{mcc}}$.

Proof: Only the case of π equal to an R-literal constructed from the $=$ predicate is considered here. For the other cases, the reader is referred to [Rus96]. Let π be of the form $\lambda = \lambda'$. If $\lambda = \lambda' \in \mathcal{C}_{\text{mcc}}$ then $\lambda = \lambda' \in FOT(\mathcal{C}_{\text{mcc}})$. This implies by Definition 3.7 that $\mathcal{M}_{\text{mcc}} \models_{\text{E}} \lambda = \lambda'$. If $\lambda = \lambda' \notin \mathcal{C}_{\text{mcc}}$ then by Lemma 3.3 $\lambda \neq \lambda' \in \mathcal{C}_{\text{mcc}}$, which implies that $\mathcal{M}_{\text{mcc}} \models_{\text{E}} \lambda \neq \lambda'$ and hence $\mathcal{M}_{\text{mcc}} \not\models_{\text{E}} \lambda = \lambda'$.

□

Corollary 3.1 *Let $\mathcal{C}$ be a consistent configuration of the E_{CLDS} system. Then $\mathcal{C}$ is satisfiable.*

Proof: The proof trivially follows from Lemmas 3.2 and 3.5.

□

The following Proposition is the final result needed to prove the completeness theorem.

Proposition 3.1 Let $\mathcal{C}$ be a configuration of the E_{CLDS} system and let π be a declarative unit or an R-literal such that $\pi \notin \mathcal{C}$. If $\mathcal{C} \not\vdash_{\text{E}} \pi$ then $\mathcal{C} + [\pi]$ is a consistent configuration.

Proof: Only the case of π equal to a R-literal of the form $\lambda = \lambda'$ is considered. For the other cases, the reader is referred to [Rus96]. Let π be of the form $\lambda = \lambda'$. The contrapositive of the

[7] Notice that $FOT(\mathcal{C}_{\text{mcc}})$ contains only ground literals

proposition is proved. Suppose that $\mathcal{C}+[\lambda \neq \lambda']$ is not consistent. Then $\mathcal{C}+[\lambda \neq \lambda'] \vdash_{\mathrm{E}} \bot : \lambda_1$. By definition of the $R\mathcal{I}$, the configuration $\mathcal{C}' = \mathcal{C} + [\lambda = \lambda']$ is derivable from $\mathcal{C}$. Hence, $\mathcal{C} \vdash_{\mathrm{E}} \lambda = \lambda'$.

□

The proof of Theorem 3.2 can now be given.

Proof of Theorem 3.2:

The proof is by contrapositive. Assume that $\mathcal{C} \not\vdash_{\mathrm{E}} \mathcal{C}'$. Then by Lemma 2.1 there exists a $\pi \in \mathcal{C}' - \mathcal{C}$, where π is a declarative unit or an R–literal, such that $\mathcal{C} \not\vdash_{\mathrm{E}} \pi$. Then by Proposition 3.1, $\mathcal{C}+[\neg\pi]$ is a consistent configuration. By Corollary 3.1, $\mathcal{C}+[\neg\pi]$ is satisfiable. Let $\mathcal{M}$ be the canonical semantic structure that satisfies the configuration $\mathcal{C} + [\neg\pi]$. So $\mathcal{M} \models_{\mathrm{E}} \mathcal{C}$ and $\mathcal{M} \models_{\mathrm{E}} \neg\pi$. There are three cases to consider, according to the form of π. Only the case of π equal to an R-literal of the form $\lambda = \lambda'$ is considered here. For the other two cases the reader is referred to [Rus96]. Let π be then of the form $\lambda = \lambda'$. By Definition 2.10, $\mathcal{M} \Vdash_{\mathrm{FOL}} \lambda \neq \lambda'$, which implies that $\mathcal{M} \not\Vdash_{\mathrm{FOL}} \lambda = \lambda'$. Then by Definitions 2.10 and 2.11 $\mathcal{A}^+, FOT(\mathcal{C}) \not\Vdash_{\mathrm{FOL}} \lambda = \lambda'$. Hence $\mathcal{C} \not\models_{\mathrm{E}} \mathcal{C}'$.

Correspondence result. In Section 1, it has been stated that the $\mathrm{E}_{\mathrm{CLDS}}$ system is a *generalisation* of the standard implicit formalisation of the modal logic of elsewhere, in that it facilitates reasoning about structures of actual worlds, which may or may not be singleton structures. This claim is substantiated here by showing (i) that there exists a correspondence between the $\mathrm{E}_{\mathrm{CLDS}}$ system and the Hilbert system for the logic of elsewhere, whenever certain restrictions are imposed on initial configurations, and (ii) that the correspondence clearly fails if no restriction is imposed. As far as the first result is concerned, the restriction consists of identifying a particular constant symbol in $\mathcal{L}_L$, say S_0, and allowing initial configurations only of the form $\mathcal{C}_i = \langle\{\}, \mathcal{F}_i\rangle$ (i.e. no R-literals belong to $\mathcal{C}_i$) where for any label λ, $\lambda \neq S_0$, $\mathcal{F}_i(\lambda) = \{\}$. With this restriction the only initial assumptions (if any) are modal formulae associated with the label S_0. This corresponds to the traditional notion of local assumptions in modal logic. In particular, the following two theorems shown that any declarative unit of the form $\alpha : S_0$ can be derived from an (empty) initial configuration of the form $\mathcal{C}_i$ if and only if its formula α is derivable, within the sound and complete axiomatic system for modal logic given in [dR92], from the (empty) set of formulae that appear in $\mathcal{C}i$.

A definition of the axiomatic system taken into consideration is first given.

Definition 3.8 Let $\mathcal{L}_P$ be the propositional modal logic considered in the $\mathrm{E}_{\mathrm{CLDS}}$ system. The axiomatic system for the logic of elsewhere, written $\mathcal{E}_{Ax}$, is defined as a standard propositional logic axiomatisation [HC68] extended with the following schemas:

$$\Box(\alpha \to \beta) \to (\Box\alpha \to \Box\beta) \qquad \text{(E1)}$$

$$\alpha \to \Box\Diamond\alpha \qquad \text{(E2)}$$

$$\Diamond\Diamond\alpha \to (\alpha \vee \Diamond\alpha) \qquad \text{(E3)}$$

together with the (MP) and (Nec) rules:

$$\text{If } \vdash_{\mathcal{E}_{Ax}} \alpha \text{ then } \vdash_{\mathcal{E}_{Ax}} \Box\alpha \qquad \text{(Nec)}$$

Theorem 3.3 [Simple correspondence] Consider the $\mathrm{E_{CLDS}}$ system, the axiomatic system $\mathcal{E}_{Ax}$ for the logic of elsewhere, and the initial empty configuration $\mathcal{C}_{\{\}} = \langle\{\}, \mathcal{F}\rangle$, given by $\mathcal{F}(\lambda) = \{\}$, for any label λ. Let α be a formula of $\mathcal{L}_P$. Then:

$$\vdash_{\mathcal{E}_{Ax}} \alpha$$
$$\text{if and only if}$$
$$\text{for all ground terms } \lambda \in Mon(\mathcal{L}_P, \mathcal{L}_L)\ \ \mathcal{C}_{\{\}} \vdash_{\mathrm{E}} \alpha : \lambda$$

Proof: ("Only if") part: The proof is by induction on the number of steps of the shortest derivation of α proving $\vdash_{\mathcal{E}_{Ax}} \alpha$. The formal proof is given in [Rus96] but with the following three extra cases on the base case of the induction. The base case is when there are zero number of steps, i.e. α is an instantiation of the schemas given in Definition 3.8. It is sufficient to prove that $\mathcal{C}_{\{\}} \vdash_{\mathrm{E}} \lambda : \alpha$ for α equal to (E1), (E2) and (E3). Only the case of the (E3) schema is considered here, since the proof of the (E2) is already given in Figure 3.1 and the proof of (E1) schema is fully given in [Rus96].

Case (E3):
The contrapositive schema is considered instead, $(\Box\alpha \wedge \alpha) \rightarrow \Box\Box\alpha$. Let λ be a ground term of $Mon(\mathcal{L}_P, \mathcal{L}_L)$. Then $\mathcal{C}_{\{\}} \vdash_{\mathrm{E}} (\Box\alpha \wedge \alpha) \rightarrow \Box\Box\alpha$. See Figure 4.

("If") part: The proof is by showing that the contrapositive statement holds. Given the soundness and completeness of both the systems under consideration this means to show that if $\not\models_{\mathcal{E}_{Ax}} \alpha$ then there exists a ground term λ such that $\mathcal{C}_{\{\}} \not\models_{\mathrm{E}} \alpha : \lambda$. The formal proof is given in [Rus96] and it informally consists of constructing a classical interpretation $\mathcal{M}$ from the Kripke countermodel of α and showing that $\mathcal{M}$ is a model of the $\mathrm{E_{CLDS}}$ system (i.e. it satisfies the schemas of the extended algebra $\mathcal{A}_{\mathrm{E}}^{+}$). This implies that there exists a ground term λ, specifically the one corresponding to the Kripke possible world where α is false, such that $\mathcal{M} \not\models_{\mathrm{FOL}} [\alpha]^*(\lambda)$[8]. Hence $\mathcal{M} \not\models_{\mathrm{E}} \alpha : (\lambda)$.

$\Box$

It is easy to show that the above theorem can be generalised to global and local assumptions of the logic of elsewhere using the notation introduced by Fitting in [Fit83]. (i.e. $T \models_{\mathcal{E}_{Ax}} U \Rightarrow \alpha$ denote that the formula α is derivable from the global assumptions T and the local assumptions U.) This is achieved by considering initial configurations of the form $\mathcal{C}_{TU} = \langle\{\}, \mathcal{F}_{TU}\rangle$, where $\mathcal{F}_{TU}(S_0) = T \cup U$ and for each label $\lambda \neq S_0$, $\mathcal{F}(\lambda) = T$.

The above results effectively provide a translation method from a modal theory $\langle T, U\rangle$ of the logic of elsewhere into an equivalent $\mathrm{E_{CLDS}}$ configuration, which preserves derivability and semantic entailment. However, it is clear that many initial configurations are not the translation of any modal theory. (For example, any configuration whose diagram $\mathcal{D}$ is not

[8]Note that $[\alpha]^*(\lambda)$ is proved to be false in $\mathcal{M}$ by the construction of $\mathcal{M}$ using the fact that α is false at the Kripke possible world that corresponds to λ.

$$\mathcal{C}_\emptyset\langle\rangle$$

$$\mathcal{C}_1\langle[\Box\alpha\wedge\alpha:\lambda]\rangle \quad \text{(assumption)}$$

$$\mathcal{C}_2\langle\Box\alpha:\lambda,\alpha:\lambda\rangle \quad (\wedge\mathcal{E})$$

$$\mathcal{C}_3\langle[R(\lambda,box_{\Box\alpha}(\lambda))]\rangle \quad \text{(assumption)}$$

$$\mathcal{C}_4\langle[R(box_{\Box\alpha}(\lambda),box_{\alpha}(box_{\Box\alpha}(\lambda)))]\rangle \quad \text{(assumption)}$$

$$\mathcal{C}_5\langle[\neg\alpha:box_{\alpha}(box_{\Box\alpha}(\lambda))]\rangle \quad \text{(assumption)}$$

$$\tilde{\mathcal{C}}\langle[R(\lambda,box_{\alpha}(box_{\Box\alpha}(\lambda)))]\rangle \quad \text{(assumption)}$$

$$\tilde{\mathcal{C}}_1\langle\alpha:box_{\alpha}(box_{\Box\alpha}(\lambda))\rangle \quad (\Box\mathcal{E})$$

$$\tilde{\mathcal{C}}_2\langle\bot:box_{\alpha}(box_{\Box\alpha}(\lambda))\rangle \quad (\wedge\mathcal{I})$$

$$\mathcal{C}_6\langle\neg R(\lambda,box_{\alpha}(box_{\Box\alpha}(\lambda)))\rangle \quad (R\text{-I})$$

$$\mathcal{C}_7\langle\lambda=box_{\alpha}(box_{\Box\alpha}(\lambda))\rangle \quad (R\text{-A})$$

$$\mathcal{C}_8\langle\alpha:box_{\alpha}(box_{\Box\alpha}(\lambda))\rangle \quad (\mathcal{I}_{Sub})$$

$$\mathcal{C}_9\langle\bot:box_{\alpha}(box_{\Box\alpha}(\lambda))\rangle \quad (\wedge\mathcal{I})$$

$$\mathcal{C}_{10}\langle\alpha:box_{\alpha}(box_{\Box\alpha}(\lambda))\rangle \quad (\neg\mathcal{I})$$

$$\mathcal{C}_{11}\langle\Box\alpha:box_{\Box\alpha}(\lambda)\rangle \quad (\neg\mathcal{I})$$

$$\mathcal{C}_{12}\langle\Box\Box\alpha:\lambda\rangle \quad (\neg\mathcal{I})$$

$$\mathcal{C}_{13}\langle(\Box\alpha\wedge\alpha)\rightarrow\Box\Box\alpha:\lambda\rangle \quad (\rightarrow\mathcal{I})$$

Figure 4: $\mathrm{E}_{\mathrm{CLDS}}$ derivation of the (E3) axiom

empty or whose $\mathcal{F}$ differs at more than one label.) Hence, the information that such configurations encode cannot be represented within the standard logic of elsewhere, making the $\mathrm{E}_{\mathrm{CLDS}}$ system strictly more general than the standard Hilbert system.

4 The $\mathcal{L}_{\mathrm{CLDS}}$ System

In this section the L_{CLDS} system is defined. Additional inference rules are defined for dealing with the $\otimes$ operator and additional interactions between labels and formulas. The extended labelling algebra $\mathcal{A}_{\mathcal{L}}^{+}$ is defined, which together with the notion of semantic entailment and model given in Definitions 2.11 and 2.10 provides the model theoretic semantics for L_{CLDS}. The soundness and completeness of the proof system with respect to this semantics is then proved. The correspondence with a standard sequent calculus presentation for linear logic is demonstrated, showing the L_{CLDS} system to be more general than standard presentations. Other proof systems based on LDS for substructural logics have been described in [DG94, Gab92] in which a tableau proof system was presented, and in [BFR97], in which a natural deduction system was given. This case study is restricted to the multiplicative operators $\otimes$, $\rightarrow$ and $\neg$ of linear logic and extensions to other substructural logics can be developed. In Section 5 the extensions necessary for additive operators of linear logic are indicated.

Language of L_{CLDS}. The language of the L_{CLDS} system is the pair $\langle \mathcal{L}_{\mathcal{P}}, \mathcal{L}_L \rangle$, where $\mathcal{L}_{\mathcal{P}}$ is a standard propositional language restricted to the substructural operators $\{\rightarrow, \otimes, \neg\}$ and including the two propositions $\top$ and $\bot$. The proposition $\top$ is the identity of $\otimes$ and $\bot$ is equivalent[9] to $\neg\top$. The labelling language $\mathcal{L}_L$ is composed of a countable set of symbols $\{a, b, \ldots, f, a_1, b_1, \ldots, f_1, \ldots\}$ called *constants*, a countable set of variables $\{x, y, z \ldots\}$, a binary function symbol $\circ$ and a binary relation $\preceq$ both usually written in infix form. The language $\mathcal{L}_L$ is extended into $\mathit{Func}(\mathcal{L}_P, \mathcal{L}_L)$ by adding for each wff α of $\mathcal{L}_{\mathcal{P}}$ different from $\top$ the symbol c_α called *parameter*. For the wff $\top$ the parameter 1 is included. Terms of the semi-extended labelling language $\mathit{Func}(\mathcal{L}_P, \mathcal{L}_L)$ are defined inductively, as consisting of 1, constants, parameters and variables, together with expressions of the form $x \circ y$ where x and y are terms. Ground terms are called labels. Note that all parameters have a special role in the proof theory and semantics, especially $c_\bot$. As mentioned in Section 2 parameters c_α represent the smallest label verifying α. The declarative unit $\bot : x \circ y$ is introduced into a proof as a result of deriving any pair of declarative units of the form $\alpha : x$ and $\neg\alpha : y$. The label $c_\bot$ is therefore smaller than any label $x \circ y$ such that $\alpha : x$ and $\neg\alpha : y$ hold for any α.

In a (substructural) configuration $\mathcal{C}$ the R-literals in the diagram are referred to as constraints. Any pair of constraints Δ and $\overline{\Delta}$ in a configuration will be denoted by the shorthand notation $\bot$. Note that the two uses of this symbol (i.e. $\bot$ in the language $\mathcal{L}_{\mathcal{P}}$ and $\bot$ in configurations are distinguishable by the presence of a label in the first case.

The labelling algebra $\mathcal{A}$ is a set of FOL axioms, which express the properties of the function symbol $\circ$ with respect to the pre-ordering relation $\preceq$.

Definition 4.1 (*Labelling Algebra $\mathcal{A}_{\mathcal{L}}$*)

The labelling algebra $\mathcal{A}_{\mathcal{L}}$ written in $\mathit{Func}((\mathcal{L}_P, \mathcal{L}_L))$ is the first order theory given by the following axioms

[9] This is easy to show using the ND rules for L_{CLDS} .

1. (identity) $\forall x[1 \circ x \preceq x \wedge x \preceq 1 \circ x]$

2. (order-preserving) $\forall x, y, z[x \preceq y \rightarrow x \circ z \preceq y \circ z \wedge z \circ x \preceq z \circ y]$

3. (pre-ordering) $\preceq$ is reflexive and transitive:
$\forall x[x \preceq x]$ and $\forall x, y, z[x \preceq y \wedge y \preceq z \rightarrow x \preceq z]$

4. (commutativity) $\forall x, y[x \circ y \preceq y \circ x]$

5. (associativity) $\forall x, y, z[(x \circ y) \circ z \preceq x \circ (y \circ z)]$

□

A L_{CLDS} system consists of a labelling algebra $\mathcal{A}$ and a set of inference rules to generate one configuration from another.

4.1 A Natural Deduction System for $\mathcal{L}_{\text{CLDS}}$

The operator rules ($\rightarrow\mathcal{E}$), ($\rightarrow\mathcal{I}$), ($\otimes\mathcal{E}$), ($\otimes\mathcal{I}$), ($\neg\mathcal{I}$), ($\neg\mathcal{E}$), ($\neg\neg$), together with various rules describing the interaction between constraints and declarative units are introduced next. They are all defined mathematically. Note that the rules ($\otimes\mathcal{E}$) and ($\otimes\mathcal{I}$) belong to the category $\mathcal{I}^0$. Each of the operator rules incorporates the idea of combining resources in order to derive new declarative units. For example, the $\rightarrow\mathcal{I}$ rule expresses how resources λ and λ', verifying $\alpha \rightarrow \beta$ and α respectively, are combined into $\lambda \circ \lambda'$ to verify β.

The characteristic rule (ch) and unit rule (unit) respectively reflect the notions that there is a smallest resource verifying a formula and that any increase of resource maintains verifiability. In linear logic two combinations of resources are only related if they comprise the same resources, but possibly combined in different orders, or if one resource is the smallest that verifies a particular formula. The constraint rule (R-A) in Table 2 is used to derive relationships between labels. The (unit) rule often interacts with the (R-A) rule in the following way: first the (R-A) rule is used to deduce that a resource λ' is greater than another λ, and then the (unit) rule is used to deduce $\alpha : \lambda'$ from $\alpha : \lambda$. These three rules are all in category $\mathcal{I}^0$. The effect of the (ch) rule is seen in the fact that a derivation from an initially empty configuration contains declarative units with characteristic labels only. Initial configurations are discussed later in this section.

An additional derived rule, the (PC) rule, which is equivalent to the double negation ($\neg\neg$) rule could also be included as it is so often used in proofs and can make them slightly shorter. However, for simplicity it is omitted here. Example derivations in L_{CLDS} using the rules are given in Figures 5 and 6, where a derivation of the ($\neg\neg$) rule from the (PC) rule and vice versa is shown, as well as a derivation of $A \otimes B \rightarrow \neg(A \rightarrow \neg B) : 1$. A slightly different presentation of the natural deduction system was first described in [BDR97] and developed in [BFR97]. In both of those papers the semantics was in terms of a particular lattice framework. In this paper the necessary properties are expressed in FOL.

The L_{CLDS} Rules Let $\mathcal{C} = \langle \mathcal{D}, \mathcal{F} \rangle$ be a (substructural) configuration. The set of natural deduction rules is now defined as follows:

($\rightarrow\mathcal{E}$) If $\alpha \rightarrow \beta : \lambda \in \mathcal{C}$ and $\alpha : \lambda' \in \mathcal{C}$ then $\mathcal{C}' = \mathcal{C} + [\beta : \lambda \circ \lambda']$.

($\rightarrow\mathcal{I}$) If $\mathcal{C} + [\alpha : c_\alpha] \vdash_{\text{L}} \beta : \lambda \circ c_\alpha$ then $\mathcal{C}' = \mathcal{C} + [\alpha \rightarrow \beta : \lambda]$.

($\otimes\mathcal{E}$) If $\alpha \otimes \beta : \lambda \in \mathcal{C}$ then $\mathcal{C}' = \mathcal{C} + [\alpha : c_\alpha] + [\beta : c_\beta] + [c_\alpha \circ c_\beta \preceq \lambda]$.

($\otimes\mathcal{I}$) If $\alpha : \lambda_1 \in \mathcal{C}$ and $\beta : \lambda_2 \in \mathcal{C}$, then $\mathcal{C}' = \mathcal{C} + [\alpha \otimes \beta : \lambda_1 \circ \lambda_2]$.

($\neg\mathcal{I}$) If $\mathcal{C} + [\alpha : c_\alpha] \vdash_{\mathrm{L}} \bot : \lambda \circ c_\alpha$ then $\mathcal{C}' = \mathcal{C} + [\neg\alpha : \lambda]$.

($\neg\mathcal{E}$) If $\alpha : \lambda_1 \in \mathcal{C}$ and $\neg\alpha : \lambda_2 \in \mathcal{C}$, then $\mathcal{C}' = \mathcal{C} + [\bot : \lambda_1 \circ \lambda_2]$.

($\neg\neg$) If $\neg\neg\alpha : \lambda \in \mathcal{C}$ then $\mathcal{C}' = \mathcal{C} + [\alpha : \lambda]$.

characteristic rule (ch) If $\alpha : \lambda \in \mathcal{C}$ then $\mathcal{C}' = \mathcal{C} + [\alpha : c_\alpha] + [c_\alpha \preceq \lambda]$,

new unit rule (unit) If $\alpha : \lambda \in \mathcal{C}$ and $\lambda \preceq \lambda' \in \mathcal{C}$. then $\mathcal{C}' = \mathcal{C} + [\alpha : \lambda']$.

rule for $\bot$ (base) The declarative unit $\bot : c_\bot$ may be introduced into any configuration, that is, $\mathcal{C}' = \mathcal{C} + [\bot : c_\bot]$.

The next four rules are from Table 2 adapted for L_{CLDS}.

constraint rule (R-A) If $\mathcal{C},\ \mathcal{A} \vdash_{\mathrm{FOL}} \Delta$, where Δ is a constraint then $\mathcal{C}' = \mathcal{C} + [\Delta]$.

reduce rule ($\mathcal{C}$-R) If $\mathcal{C}' \subseteq \mathcal{C}$ then $\mathcal{C}'$ can be derived from $\mathcal{C}$.

contradiction rule ($\bot\mathcal{I}$) If $\Delta \in \mathcal{C}$ and $\neg\Delta \in \mathcal{C}$ then $\mathcal{C}' = \mathcal{C} + [\alpha : \lambda]$, where Δ is a constraint and α and λ are any wff and label.

$R\mathcal{I}$ If $\mathcal{C} + [\overline{\Delta} \vdash_{\mathrm{L}} \bot$ then $\mathcal{C}' = \mathcal{C} + [\Delta]$.

1	$\mathcal{C}_1\langle[\neg\alpha : c_{\neg\alpha}]\rangle$	
2	$\vdots$	(proof P)
3	$\mathcal{C}_n\langle[\bot : c_{\neg\alpha} \circ \lambda]\rangle$	
4	$\mathcal{C}_{n+1}\langle[\neg\neg\alpha : \lambda]\rangle$	$\neg\mathcal{I}$ (1-3)
5	$\mathcal{C}'\langle[\alpha : \lambda]\rangle$	$\neg\neg$

1	$\mathcal{C}\langle[\neg\neg\alpha : \lambda]\rangle$	(given)
2	$\mathcal{C}_1\langle[\neg\alpha : c_{\neg\alpha}]\rangle$	
3	$\mathcal{C}_n\langle[\bot : c_{\neg\alpha} \circ \lambda]\rangle$	$\neg\mathcal{E}$ (1,2)
4	$\mathcal{C}'\langle[\alpha : \lambda]\rangle$	PC (2-3)

Figure 5: Equivalence of the (PC) and ($\neg\neg$) rules

In Figure 5 it is shown how the (PC) rule can be derived given the ($\neg\neg$) rule, and also how, given $\neg\neg\alpha : \lambda$ and the (PC) rule the ($\neg\neg$) rule can be derived. Henceforth, only the ($\neg\neg$) rule will be considered.

1	$\mathcal{C}_1\langle[\alpha \otimes \beta : c_{\alpha\otimes\beta)}]\rangle$	(assumption for $\to\mathcal{I}$ line 10)
2	$\mathcal{C}_2\langle[\alpha \to \neg\beta : c_{\alpha\to\neg\beta}]\rangle$	(assumption for $\neg\mathcal{I}$ line 9)
3	$\mathcal{C}_3\langle[\alpha : c_\alpha]\rangle$	($\otimes\mathcal{E}$ (1))
4	$\mathcal{C}_4\langle[\beta : c_\beta]\rangle$	($\otimes\mathcal{E}$ (1))
5	$\mathcal{C}_5\langle[c_\alpha \circ c_\beta \preceq c_{\alpha\otimes\beta}]\rangle$	($\otimes\mathcal{E}$ (1))
6	$\mathcal{C}_6\langle[\neg\beta : c_{\alpha\to\neg\beta} \circ c_\alpha]\rangle$	$\to\mathcal{E}$ (2,3)
7	$\mathcal{C}_7\langle[\bot : c_{\alpha\to\neg\beta} \circ c_\alpha \circ c_\beta]\rangle$	$\neg\mathcal{E}$ (4,6)
8	$\mathcal{C}_8\langle[\bot : c_{\alpha\to\neg\beta} \circ c_{\alpha\otimes\beta}]\rangle$	(unit) rule (7)
9	$\mathcal{C}_9\langle[\neg(\alpha \to \neg\beta) : c_{\alpha\otimes\beta}]\rangle$	$\neg\mathcal{I}$ (2-8)
10	$\mathcal{C}'\langle[\alpha \otimes \beta \to \neg(\alpha \to \neg\beta) : 1]\rangle$	$\to\mathcal{I}$ (2- 9)

Figure 6: Derivation of $\alpha \otimes \beta \to \neg(\alpha \to \neg\beta) : 1$

4.2 A First Order Semantics for L_{CLDS}

The extended labelling language $Mon(\mathcal{L}_P, \mathcal{L}_L)$, is defined by adding to $Func(\mathcal{L}_P, \mathcal{L}_L)$ monadic predicates $[\alpha]^*$, for each wff α in $\mathcal{L}_P$. The atom $[\alpha]^*(x)$ can be read as x verifies α. The extended algebra $\mathcal{A}^+_{\mathcal{L}}$ for L_{CLDS}, written in $Mon(\mathcal{L}_P, \mathcal{L}_L)$ expresses relationships between the monadic predicates and constraints according to the semantic meaning of the substructural operators and the (unit), (ch) and (base) rules.

The Axiom Schema for L_{CLDS} The axiom schema for L_{CLDS} are Skolemised versions of the basic axiom schema given in Table 5. The first axiom (Ax1) characterises the (unit) rule and the second axiom (Ax2) characterises the (ch) rule. The others, (Ax3) - (Ax5) characterise the operators $\rightarrow$, $\neg$ and $\otimes$ respectively, whilst axiom (Ax6) characterises the ($\neg\neg$) rule. Axiom (Ax7) characterises the (base) rule and (Ax8) the ($R\mathcal{I}$) rule.

Table 5: Basic Axioms for L_{CLDS}

Ax1:	$\forall x \forall y (x \preceq y \wedge [\alpha]^*(x) \rightarrow [\alpha]^*(y))$
Ax2:	$\forall x ([\alpha]^*(x)) \rightarrow \exists y ([\alpha]^*(y) \wedge \forall z ([\alpha]^*(z) \rightarrow y \preceq z)))$
Ax3:	$\forall x ([\alpha \rightarrow \beta]^*(x) \leftrightarrow \forall y ([\alpha]^*(y) \rightarrow [\beta]^*(x \circ y)))$
Ax4:	$\forall x ([\neg\alpha]^*(x) \leftrightarrow \forall y ([\alpha]^*(y) \rightarrow \bot : x \circ y))$
Ax5:	$\forall w ([\alpha \otimes \beta]^*(w) \leftrightarrow \exists u \exists v ([\alpha]^*(u) \wedge [\beta]^*(v) \wedge (u \circ v \preceq w)))$
Ax6:	$\forall x ([\neg\neg\alpha]^*(x) \rightarrow [\alpha]^*(x))$
Ax7:	$[\bot]^*(c_\bot)$

Table 6: The Extended Algebra $\mathcal{A}^+_{\mathcal{L}}$

Ax1:	$\forall x \forall y (x \preceq y \wedge [\alpha]^*(x) \rightarrow [\alpha]^*(y))$
Ax2:	$\forall x ([\alpha]^*(x) \rightarrow [\alpha]^*(c_\alpha) \wedge c_\alpha \preceq x)$
Ax3a:	$\forall x \forall y ([\alpha \rightarrow \beta]^*(x) \wedge [\alpha]^*(y) \rightarrow [\beta]^*(x \circ y)))$
Ax3b:	$\forall x ([\alpha \rightarrow \beta]^*(x) \leftarrow ([\alpha]^*(c_\alpha) \rightarrow [\beta]^*(x \circ c_\alpha)))$
Ax4a:	$\forall x \forall y ([\neg\alpha]^*(x) \wedge [\alpha]^*(y) \rightarrow \bot : x \circ y))$
Ax4b:	$\forall x ([\neg\alpha]^*(x) \leftarrow ([\alpha]^*(c_\alpha) \rightarrow \bot : x \circ c_\alpha))$
Ax5a:	$\forall x ([\alpha \otimes \beta]^*(x) \rightarrow ([\alpha]^*(c_\alpha) \wedge [\beta]^*(c_\beta) \wedge c_\alpha \circ c_\beta \preceq x))$
Ax5b:	$\forall u \forall v ([\alpha]^*(u) \wedge [\beta]^*(v) \rightarrow [\alpha \otimes \beta]^*(u \circ v))$
Ax6:	$\forall x ([\neg\neg\alpha]^*(x) \rightarrow [\alpha]^*(x))$
Ax7:	$[\bot]^*(c_\bot)$

The extended algebra $\mathcal{A}^+_{\mathcal{L}}$ is the first order theory given by the set of axioms shown in Table 6. These are sub-axioms derived from the basic axioms which are used to form a theory corresponding to a given initial structure and those structures derived from it. The sub-axioms, given in Table 6, are derived from the basic axioms by taking each half of the $\leftrightarrow$ in turn. The axioms (Ax1) and Ax2) relate declarative units to constraints. Several of the sub-axioms can be simplified by use of parameters and the characteristic and new unit rules (effectively applying Skolemisation). Declarative units including a parameter in the label can only arise during a derivation in this application; they will never be part of an initial configuration. The parameters play the same role as Skolem constants and functions in FOL and are the semantic counterpart of a syntactic wff.

In (Ax2), for example, the label y is specified to be the smallest label that verifies α. (Notice that, if $\leftrightarrow$ is used in place of the first $\rightarrow$ then (Ax1) follows from the (if)-half as a theorem.) (Ax2) can be Skolemised to give

$$\forall x([\alpha]^*(x) \rightarrow ([\alpha]^*(c_\alpha) \wedge \forall w([\alpha]^*(w) \rightarrow c_\alpha \preceq w)))$$

The parameter c_α is associated with the formula α and is also called the *α-characteristic* label. It is the smallest label to verify α. (The parameter 1 is the $\top$-characteristic label and the parameter $c_\perp$ is the $\perp$ characteristic label.) (Ax2) can also be simplified to the equivalent version[10] (also called (Ax2))

$$\forall x([\alpha]^*(x) \rightarrow ([\alpha]^*(c_\alpha) \wedge c_\alpha \preceq x))$$

As an example of derivations of other sub-axioms, consider (Ax3). To derive (Ax3b) from (Ax3), first of all

$$\forall x[\alpha \rightarrow \beta]^*(x) \vee \exists y[\alpha]^*(y)$$

is obtained from the $\leftarrow$ direction of (Ax3). It is easy to show, using (Ax2), that this is equivalent to $\forall x[\alpha \rightarrow \beta]^*(x) \vee [\alpha]^*(c_\alpha))$. Similar considerations apply to the other axioms. Note that the axioms in Table 6 are of quite simple forms; they can be reduced further to yield either Horn clauses or simple disjunctions, in which one disjunct is always ground, having the form $[\alpha]^*(c_\alpha)$. This feature will be discussed further in Section 5.

Initial Substructural Configuration There are several possible assumptions that may be made about an initial configuration S_i. Some examples of simple derivations from possible initial configurations are given next. First, suppose $c_\beta \preceq c_\alpha$ and $\beta : \lambda$ hold in a configuration, then $\vdash_{\text{L}} \alpha \rightarrow \beta : 1$. For suppose $\alpha : c_\alpha$ then it is required to show that $\beta : c_\alpha$, or $\beta : c_\beta$, since $c_\beta \preceq c_\alpha$. But $\beta : \lambda$ implies $\beta : c_\beta$ by the (ch) rule. Second, suppose the two declarative units $\alpha : c_\beta$ and $\beta : c_\alpha$ were present in an initial configuration. Then, $\vdash_{\text{L}} \alpha \rightarrow \beta : 1$ and $\vdash_{\text{L}} \beta \rightarrow \alpha : 1$, so α and β would be considered equivalent to one another. Since α and β may be any sentences in the language this additional implication may not reflect the standard linear logic. This is because the L_{CLDS} is a proper extension of standard linear logic. As shown in the correspondence theorem the set of standard linear logic theorems are provable in L_{CLDS} only under the condition that the initial configuration is empty.

Notation 4.1 (Empty Initial Configuration) *The term $\mathcal{C}_\emptyset$ denotes a particular initial configuration in which there are no declarative units or constraints.*

$\triangleleft$

Note that declarative units $\alpha : 1$ derivable from an empty configuration are called *theorems*.

In Sections 4.3 and 4.4 it is shown the natural deduction rules are sound and complete with respect to the semantics. In Section 4.5 it is also demonstrated that the rules correspond to a standard presentation of substructural logics.

[10]Note that in this form it cannot be used to derive (Ax1).

4.3 Soundness Result

Recall from Section 2.3 that $\mathcal{C} \models_{\text{L}} \mathcal{C}'$ iff $\mathcal{A}^+_{\mathcal{L}}$, FOT($\mathcal{C}$) $\models$ FOT($\mathcal{C}'$). The soundness property that is proved in this section is the following: for configurations $\mathcal{C}$ and $\mathcal{C}'$, if $\mathcal{C} \vdash_{\text{L}} \mathcal{C}'$, then $\mathcal{C} \models_{\text{L}} \mathcal{C}'$ and the proof follows the pattern used for $\mathcal{E}_{\text{CLDS}}$. The proof is in three steps: first of all the faithfulness of the translation is shown, namely if $\mathcal{C} \vdash_{\text{L}} \mathcal{C}'$ then $\mathcal{A}^+, FOT(\mathcal{C}) \vdash_{\text{FOL}} FOT(\mathcal{C}')$. Next the soundness of first order logic is used to derive $\mathcal{A}^+, FOT(\mathcal{C}) \models_{\text{FOL}} FOT(\mathcal{C}')$ and finally, the definition of semantic entailment is used to obtain $\mathcal{C} \models_{\text{L}} \mathcal{C}'$. The soundness property then follows by transitivity of implication.

The first proof (faithfulness of the translation) is made using induction on the length of natural deduction proofs.

Lemma 4.1 (*Faithfulness of the Translation*)

Let $\mathcal{C}$ and $\mathcal{C}'$ be configurations, then if $\mathcal{C} \vdash_{\text{L}} \mathcal{C}'$ then $\mathcal{A}^+$, $FOT(\mathcal{C}) \vdash_{\text{FOL}} FOT(\mathcal{C}')$.
Proof:
Base Case:

Suppose the proof has length 0. The only step is therefore one that reiterates an element of $\mathcal{C}$. Clearly the property of the theorem holds in this case.
Induction step:

Suppose that the statement of the theorem holds for all proofs of length k, $k \geq 0$, and consider a proof of length $k+1$ and the last step in such a proof, from $\mathcal{C}_k$ to $\mathcal{C}_{k+1}$, where $\mathcal{C}' = \mathcal{C}_{k+1}$.

Each possibility for this step is considered in turn and for each case it is shown that $\mathcal{A}^+_{\mathcal{L}}$, FOT($\mathcal{C}_k$) $\vdash_{\text{FOL}}$ FOT($\mathcal{C}_{k+1}$). By inductive hypothesis $\mathcal{A}^+_{\mathcal{L}}$, FOT($\mathcal{C}$) $\vdash_{\text{FOL}}$ FOT ($\mathcal{C}_k$), which implies that $\mathcal{A}^+_{\mathcal{L}}$, FOT($\mathcal{C}$) $\vdash_{\text{FOL}}$ $\mathcal{A}^+_{\mathcal{L}}$, FOT ($\mathcal{C}_k$). Hence, by transitivity of $\vdash$, $\mathcal{A}^+_{\mathcal{L}}$, FOT($\mathcal{C}$) $\vdash_{\text{FOL}}$ FOT ($\mathcal{C}'$).

(unit) rule: $\lambda \preceq \lambda' \in \mathcal{C}_k$ and $\alpha : \lambda \in \mathcal{C}_k$, then $[\alpha]^*(\lambda) \in$ FOT($\mathcal{C}_k$) and $\lambda \preceq \lambda' \in$ FOT($\mathcal{C}_k$) and, by (Ax1), $\mathcal{A}^+_{\mathcal{L}}$, FOT($\mathcal{C}_k$) $\vdash_{\text{FOL}} [\alpha]^*(\lambda')$.

($\mathcal{C}$-R) rule: Since $\mathcal{C}_{k+1} \subseteq \mathcal{C}_k$, FOT($\mathcal{C}_{k+1}$) $\subseteq$ FOT($\mathcal{C}_k$) and hence $\mathcal{A}^+_{\mathcal{L}}$, FOT($\mathcal{C}_k$) $\vdash_{\text{FOL}}$ FOT($\mathcal{C}_{k+1}$).

(R-A) rule: The result is immediate since the steps are already first order and use the axioms of $\mathcal{A}$ which are included in the $\mathcal{A}^+_{\mathcal{L}}$ algebra.

(ch) rule $\alpha : \lambda \in \mathcal{C}_k$, then $[\alpha]^*(\lambda) \in$ FOT($\mathcal{C}_k$). Hence, by using (Ax2), $\mathcal{A}^+_{\mathcal{L}}$, FOT($\mathcal{C}_k$) $\vdash_{\text{FOL}} [\alpha]^*(c_\alpha)$ and $\mathcal{A}^+_{\mathcal{L}}$, FOT($\mathcal{C}_{k+1}$) $\vdash_{\text{FOL}} c_\alpha \preceq \lambda$ by Ax2.

($\to\mathcal{E}$): There exist $\alpha : \lambda$ and $\alpha \to \beta : \lambda'$ in $\mathcal{C}_k$ and $\mathcal{C}_{k+1} = \mathcal{C}_k + [\beta : \lambda' \circ \lambda]$. Hence FOT($\mathcal{C}_k$) $\supseteq \{[\alpha]^*(\lambda), [\alpha \to \beta]^*(\lambda')\}$ and FOT ($\mathcal{C}_{k+1}$) = FOT($\mathcal{C}_k$) $\cup\{[\beta]^*(\lambda' \circ \lambda)\}$. It remains to show that $\mathcal{A}^+_{\mathcal{L}}$, FOT($\mathcal{C}_k$) $\vdash_{\text{FOL}} [\beta]^*(\lambda' \circ \lambda)$. Using (Ax3a) this follows straightforwardly.

($\to\mathcal{I}$): There exists a proof of $\beta : \lambda \circ c_\alpha$ from $\mathcal{C}_k + [\alpha : c_\alpha]$. By the induction hypothesis $\mathcal{A}^+_{\mathcal{L}}$, FOT($\mathcal{C}_k$)$\cup\{[\alpha]^*(c_\alpha)\} \vdash_{\text{FOL}} [\beta]^*(\lambda \circ c_\alpha)$. Using (Ax3b) and the deduction theorem of FOL $[\alpha \to \beta]^*(\lambda)$ can be derived from $\mathcal{A}^+_{\mathcal{L}}$, FOT($\mathcal{C}_k$).

($\otimes\mathcal{E}$): $\alpha \otimes \beta : \lambda \in \mathcal{C}_k$. So $[\alpha \otimes \beta]^*(\lambda) \in$ FOT($\mathcal{C}_k$). Using (Ax5a) all of the new elements $[\alpha]^*(c_\alpha)$, $[\beta]^*(c_\beta)$ and $c_\alpha \circ c_\beta \preceq \lambda$ of FOT($\mathcal{C}_{k+1}$) are derivable from $\mathcal{A}^+_{\mathcal{L}}$, FOT($\mathcal{C}_k$).

($\otimes\mathcal{I}$): $\mathcal{C}_{k+1} = \mathcal{C}_k + [\alpha \otimes \beta : \lambda \circ \lambda'])]$. $\alpha : \lambda$ and $\beta : \lambda'$ are in $\mathcal{C}_k$, then FOT($\mathcal{C}_k$) $\supseteq$ $\{[\alpha]^*(\lambda), [\beta]^*(\lambda')\}$. Using (Ax5b) $[\alpha \otimes \beta]^*(\lambda \circ \lambda')$ can be derived from $\mathcal{A}^+_{\mathcal{L}}$, FOT($\mathcal{C}_k$).

$\neg\mathcal{E}$**:** There exist $\alpha : \lambda$ and $\neg\alpha : \lambda'$ in $\mathcal{C}_k$ and $\mathcal{C}_{k+1} = \mathcal{C}_k + [\bot : \lambda' \circ \lambda]$. Hence FOT($\mathcal{C}_k$) $\supseteq \{[\alpha]^*(\lambda), [\neg\alpha]^*(\lambda')\}$ and FOT ($\mathcal{C}_{k+1}$) = FOT($\mathcal{C}_k$) $\cup\{[\bot]^*(\lambda' \circ \lambda)\}$. It remains to show that $\mathcal{A}^+_{\mathcal{L}}$, FOT($\mathcal{C}_k$) $\vdash_{\text{FOL}} [\bot]^*(\lambda' \circ \lambda)$. Using (Ax4a) this follows straightforwardly.

($\neg\mathcal{I}$**):** There exists a proof of $\bot : \lambda \circ c_\alpha$ from $\mathcal{C}_k + [\alpha : c_\alpha]$. By the induction hypothesis $\mathcal{A}^+_{\mathcal{L}}$, FOT($\mathcal{C}_k$) $\cup \{[\alpha]^*(c_\alpha)\} \vdash_{\text{FOL}} [\bot]^*(\lambda \circ c_\alpha)$. Using (Ax4b) and the deduction theorem of FOL $[\neg\alpha]^*(\lambda)$ can be derived from $\mathcal{A}^+_{\mathcal{L}}$, FOT($\mathcal{C}_k$).

($\neg\neg$**):** This is proved using (Ax6).

($\bot\mathcal{I}$**):** $\Delta \in \mathcal{C}_k$ and $\neg\Delta \in \mathcal{C}_k$, then FOT($\mathcal{C}_k$) $\vdash_{\text{FOL}} \alpha : \lambda$ and hence $[\alpha]^*(\lambda)$.

($R\mathcal{I}$**):** There exists a proof of $\bot$ from $\mathcal{C}_k + [\overline{\Delta}]$. By the induction hypothesis $\mathcal{A}^+$, FOT($\mathcal{C}_k$) $\cup$ $\{\overline{\Delta}) \vdash_{\text{FOL}} \bot$. By FOL Δ can be derived.

(base): Ax7 proves the soundness of the rule immediately.

□

Together, this suffices to prove Theorem 4.1.

Theorem 4.1 (*Soundness of* L_{CLDS}) Let $\langle\langle\mathcal{L}_P, \mathcal{L}_L\rangle, \mathcal{A}^+_{\mathcal{L}}, \mathcal{R}_{\mathcal{L}}\rangle$ be a L_{CLDS} system and let $\mathcal{C}$ and $\mathcal{C}'$ be configurations such that $\mathcal{C}' - \mathcal{C}$ is finite, then, if $\mathcal{C} \vdash_{\text{L}} \mathcal{C}'$ then $\mathcal{C} \models_{\text{L}} \mathcal{C}'$.

4.4 Completeness Result

In this section it is shown that, for configurations $\mathcal{C}$ and $\mathcal{C}'$, if $\mathcal{C} \models_{\text{L}} \mathcal{C}'$ then $\mathcal{C} \vdash_{\text{L}} \mathcal{C}'$. In fact, as for the $\mathcal{E}_{\text{CLDS}}$ case, it is the contrapositive statement that is proved, namely, if $\mathcal{C} \nvdash_{\text{L}} \mathcal{C}'$ then $\mathcal{C} \not\models_{\text{L}} \mathcal{C}'$. Because of the definition of semantic entailment between configurations, it suffices to show that if $\mathcal{C} \nvdash_{\text{L}} \mathcal{C}'$ then $\mathcal{A}^+_{\mathcal{L}}$, FOT($\mathcal{C}$) $\not\models$ FOT($\mathcal{C}'$). In other words, given $\mathcal{C} \nvdash_{\text{L}} \mathcal{C}'$, it is enough to construct a model of $\mathcal{A}^+_{\mathcal{L}}$ and FOT($\mathcal{C}$) that is not a model of FOT($\mathcal{C}'$).

Definition 4.2 (*Inconsistency of a Substructural Configuration*)

Let $\mathcal{C}$ be a configuration. Then $\mathcal{C}$ is *inconsistent* iff there exists a configuration $\mathcal{C}'$ such that $\mathcal{C} \vdash_{\text{L}} \mathcal{C}'$ and $\{\Delta, \overline{\Delta}\} \subseteq \mathcal{C}'$. This will be written in a shorthand notation as $\mathcal{C} \vdash_{\text{L}} \bot$. A configuration $\mathcal{C}$ is *consistent* if it is not inconsistent.

□

Notice that the following configuration,

$$\{\alpha : \lambda, \neg\alpha : \lambda', \alpha \otimes \neg\alpha : \lambda \circ \lambda', \bot : \lambda \circ \lambda', c_\bot \preceq \lambda \circ \lambda'\}$$

is not inconsistent. It is only so in the presence of the additional constraint $c_\bot \npreceq \lambda \circ \lambda'$. This is to be expected since in linear logic the formula $\alpha \otimes \neg\alpha$ does not allow a derivation of all formulas as in classical logic and so the configuration above should not be considered inconsistent.

The proof of completeness has the following, now familiar, overall structure. First it is shown how a consistent configuration can be expanded into a maximally consistent configuration ($\mathcal{I}_{\text{MCC}}$) and various useful properties of such a structure are proved. Next is described how a Herbrand model for any $\mathcal{I}_{\text{MCC}}$ can be given that is also a model of the axioms $\mathcal{A}^+_{\mathcal{L}}$.

Lemma 4.2 (*Consistency Lemma*) Let $\mathcal{C}$ be a consistent configuration, then, if $\mathcal{C} \nvdash_{\text{L}} \alpha : \lambda$ then $\mathcal{C} + [c_\bot \npreceq \lambda \circ c_{\neg\alpha}] + [\neg\alpha : c_{\neg\alpha}]$ is consistent, and if $\mathcal{C} \nvdash_{\text{L}} \lambda \preceq \lambda'$ then $\mathcal{C} + [\neg\lambda \preceq \lambda']$ is consistent.

Proof:

Suppose $\mathcal{C} + [c_\bot \not\preceq \lambda \circ c_{\neg\alpha}] + [\neg\alpha : c_{\neg\alpha}]$ is not consistent, then $\mathcal{C} + [\neg\alpha : c_{\neg\alpha}] + [c_\bot \not\preceq \lambda \circ c_{\neg\alpha}] \vdash_{\mathrm{L}} \bot$ and hence, $\mathcal{C} + [\neg\alpha : c_{\neg\alpha}] \vdash_{\mathrm{L}} c_\bot \preceq \lambda \circ c_{\neg\alpha}$. By the (base) rule, $\bot : c_\bot$, and then by the (unit) rule $\bot : \lambda \circ c_{\neg\alpha}$. But then $\mathcal{C} \vdash_{\mathrm{L}} \alpha : \lambda$ by the $(\neg\mathcal{I})$ rule, which contradicts the initial assumption $\mathcal{C} \nvdash_{\mathrm{L}} \alpha : c_\alpha$.
Suppose next that $\mathcal{C} + [\lambda \not\preceq \lambda'] \vdash_{\mathrm{L}} \bot$ then $\mathcal{C} \vdash_{\mathrm{L}} \lambda \preceq \lambda'$, by $(R\mathcal{I})$, a contradiction.

□

Definition 4.3 (*Maximally Consistent Configuration ($\mathcal{I}_{\mathrm{MCC}}$)*) A configuration $\mathcal{I}_{\mathrm{MCC}}$ is a *maximally consistent configuration* if it is consistent and if for any $\pi \notin \mathcal{I}_{\mathrm{MCC}}$, where π is a declarative unit or constraint, the configuration $\mathcal{I}_{\mathrm{MCC}} + [\pi]$ is not consistent.

Let $\mathcal{C}$ be a consistent configuration. A maximally consistent configuration is constructed in the same way as described for $\mathrm{E}_{\mathrm{CLDS}}$ in Lemma 3.2. The configuration $\mathcal{I}_{\mathrm{MCC}}$ has various properties as stated and proven next. These properties will be used to show that a particular Herbrand model constructed from $\mathcal{I}_{\mathrm{MCC}}$ satisfies the axioms $\mathcal{A}^{+}_{\mathcal{L}}$. In all Lemmas 4.3 – 4.11 $\mathcal{I}_{\mathrm{MCC}}$ will name a maximally consistent substructural configuration.

Note that most of the lemmas below use the following reasoning steps. Firstly, if $\pi \notin \mathcal{I}_{\mathrm{MCC}}$ for some declarative unit π, then $\mathcal{I}_{\mathrm{MCC}} + [\pi]$ is not consistent and $\mathcal{I}_{\mathrm{MCC}} + [\pi] \vdash_{\mathrm{L}} \bot$. If it is then shown that $\mathcal{I}_{\mathrm{MCC}} \vdash_{\mathrm{L}} \mathcal{I}_{\mathrm{MCC}} + [\pi]$ then $\mathcal{I}_{\mathrm{MCC}} \vdash_{\mathrm{L}} \bot$. This is usually used to derive a contradiction from the assumption that $\mathcal{I}_{\mathrm{MCC}}$ is consistent. Secondly, the monotonicity property of derivations if a maximal configuration $\mathcal{I}_{\mathrm{MCC}}$ satisfies $\mathcal{I}_{\mathrm{MCC}} \vdash_{\mathrm{L}} \alpha : \lambda$ then there is a configuration $\mathcal{C}_1 \subseteq \mathcal{I}_{\mathrm{MCC}}$ such that $\mathcal{C}_1 \vdash_{\mathrm{L}} \alpha : \lambda$. Moreover, if $\mathcal{C}_1 \subseteq \mathcal{I}_{\mathrm{MCC}}$ and $\mathcal{C}_1 \vdash_{\mathrm{L}} \alpha : \lambda$ then $\mathcal{I}_{\mathrm{MCC}} \vdash_{\mathrm{L}} \alpha : \lambda$. The transitivity of the entailment relation $\vdash_{\mathrm{L}}$ is also used throughout the proofs.

Lemma 4.3 (*Properties of $\mathcal{I}_{\mathrm{MCC}}$ w.r.t. the Labelling Algebra $\mathcal{A}_{\mathcal{L}}$*)
If $\lambda \preceq \lambda' \in \mathcal{I}_{\mathrm{MCC}}$ and $\lambda' \preceq \lambda'' \in \mathcal{I}_{\mathrm{MCC}}$ then $\lambda \preceq \lambda'' \in \mathcal{I}_{\mathrm{MCC}}$.

Proof:

If $\lambda \preceq \lambda'' \notin \mathcal{I}_{\mathrm{MCC}}$ then $\mathcal{I}_{\mathrm{MCC}} + \lambda \preceq \lambda'' \vdash_{\mathrm{L}} \bot$. By the ($R$-A) rule using the transitivity property of $\preceq$, $\mathcal{I}_{\mathrm{MCC}} \vdash_{\mathrm{L}} \mathcal{I}_{\mathrm{MCC}} + \lambda \preceq \lambda''$ and so $\mathcal{I}_{\mathrm{MCC}}$ is inconsistent, a contradiction.

□

Similar properties of $\mathcal{I}_{\mathrm{MCC}}$ are proved in a similar way for the remaining axioms of $\mathcal{A}_{\mathcal{L}}$.

Lemma 4.4 (*Characteristic property for $\mathcal{I}_{\mathrm{MCC}}$*) If $\alpha : \lambda \in \mathcal{I}_{\mathrm{MCC}}$ then $\alpha : c_\alpha \in \mathcal{I}_{\mathrm{MCC}}$ and $c_\alpha \preceq \lambda \in \mathcal{I}_{\mathrm{MCC}}$.
If $\lambda \preceq \lambda' \in \mathcal{I}_{\mathrm{MCC}}$ and $\alpha : \lambda \in \mathcal{I}_{\mathrm{MCC}}$ then $\alpha : \lambda' \in \mathcal{I}_{\mathrm{MCC}}$.
$\bot : c_\bot \in \mathcal{I}_{\mathrm{MCC}}$.
$\top : 1 \in \mathcal{I}_{\mathrm{MCC}}$.

Proof: Suppose $\alpha : \lambda \in \mathcal{I}_{\mathrm{MCC}}$ for an arbitrary label λ, but $\alpha : c_\alpha \notin \mathcal{I}_{\mathrm{MCC}}$. Then $\mathcal{I}_{\mathrm{MCC}} + [\alpha : c_\alpha]$ is inconsistent. But then, since $\mathcal{I}_{\mathrm{MCC}} \vdash_{\mathrm{L}} \mathcal{I}_{\mathrm{MCC}} + [\alpha : c_\alpha]$, $\mathcal{I}_{\mathrm{MCC}}$ is inconsistent, a contradiction. Similarly, $c_\alpha \preceq \lambda \in \mathcal{I}_{\mathrm{MCC}}$. On the other hand, suppose $\alpha : \lambda \in \mathcal{I}_{\mathrm{MCC}}$ and $\lambda \preceq \lambda' \in \mathcal{I}_{\mathrm{MCC}}$, but $\alpha : \lambda' \notin \mathcal{I}_{\mathrm{MCC}}$. Then $\mathcal{I}_{\mathrm{MCC}} + [\alpha : \lambda']$ is inconsistent. Since $\mathcal{I}_{\mathrm{MCC}} \vdash_{\mathrm{L}} \alpha : \lambda'$ this means $\mathcal{I}_{\mathrm{MCC}}$ is inconsistent, a contradiction. So $\alpha : \lambda' \in \mathcal{I}_{\mathrm{MCC}}$

If $\bot : c_\bot$ were not in $\mathcal{I}_{\mathrm{MCC}}$ then $\mathcal{I}_{\mathrm{MCC}} + [\bot : c_\bot] \vdash_{\mathrm{L}} \bot$, but from the (base) rule this would imply $\mathcal{I}_{\mathrm{MCC}}$ was inconsistent, a contradiction.

$\top : 1 \in \mathcal{I}_{\text{MCC}}$ follows trivially from the fact that $\top$ is equivalent to $\neg\bot$ and the $(\neg\mathcal{I})$ rule.

□

The characteristic property will be used many times in the following lemmas and will not be explicitly given every time.

Lemma 4.5 (*A Consistency property of* $\mathcal{I}_{\text{MCC}}$)
For any constraint $x \preceq y$, either $x \preceq y \notin \mathcal{I}_{\text{MCC}}$ or $x \not\preceq y \notin \mathcal{I}_{\text{MCC}}$.
Let α be a wff, then, if $c_\bot \not\preceq c_\alpha \circ c_{\neg\alpha} \in \mathcal{I}_{\text{MCC}}$, then either, for each label λ, $\alpha : \lambda \notin \mathcal{I}_{\text{MCC}}$, or for each label λ, $\neg\alpha : \lambda \notin \mathcal{I}_{\text{MCC}}$.
Let α be a wff, then, if $c_\alpha \not\preceq \lambda \in \mathcal{I}_{\text{MCC}}$ then $\alpha : \lambda \notin \mathcal{I}_{\text{MCC}}$

Proof: In the first place, if $x \preceq y \in \mathcal{I}_{\text{MCC}}$ and $x \not\preceq y \in \mathcal{I}_{\text{MCC}}$, then by the ($R$-A) rule $\mathcal{I}_{\text{MCC}} \vdash_{\text{L}} \bot$ and is inconsistent.
Secondly, suppose $c_\bot \not\preceq c_\alpha \circ c_{\neg\alpha}$ and that the conclusion is not the case, then $\exists\lambda'', \lambda'$ such that $\alpha : \lambda'' \in \mathcal{I}_{\text{MCC}}$ and $\neg\alpha : \lambda' \in \mathcal{I}_{\text{MCC}}$. Then $\alpha : c_\alpha \in \mathcal{I}_{\text{MCC}}$ and $\neg\alpha : c_{\neg\alpha} \in \mathcal{I}_{\text{MCC}}$ (by Lemma 4.4). Hence by the $(\neg\mathcal{E})$ rule $\mathcal{I}_{\text{MCC}} \vdash_{\text{L}} \bot : c_\alpha \circ c_{\neg\alpha}$ and hence $c_\bot \preceq c_\alpha \circ c_{\neg\alpha}$. But then $\mathcal{I}_{\text{MCC}} \vdash_{\text{L}} \bot$ by the (R-A) rule (since $c_\bot \not\preceq c_\alpha \circ c_{\neg\alpha}$) and $\mathcal{I}_{\text{MCC}}$ is inconsistent, a contradiction. In particular, $\alpha : c_\alpha \notin \mathcal{I}_{\text{MCC}}$ or $\neg\alpha : c_{\neg\alpha} \notin \mathcal{I}_{\text{MCC}}$ for every α, so if $\alpha : c_\alpha \in \mathcal{I}_{\text{MCC}}$, then $\neg\alpha : c_{\neg\alpha} \notin \mathcal{I}_{\text{MCC}}$ (under the condition $c_\bot \not\preceq c_\alpha \circ c_{\neg\alpha}$). Also, if $\alpha : \lambda \in \mathcal{I}_{\text{MCC}}$ then $\forall y.\neg\alpha : y \notin \mathcal{I}_{\text{MCC}}$ as $\forall x.\alpha : x \notin \mathcal{I}_{\text{MCC}}$ must be false.
Finally, suppose $\alpha : \lambda \in \mathcal{I}_{\text{MCC}}$, then $c_\alpha \preceq \lambda \in \mathcal{I}_{\text{MCC}}$ and hence $\mathcal{I}_{\text{MCC}}$ would be inconsistent, a contradiction.

□

Lemma 4.6 (*Maximal nature of* $\mathcal{I}_{\text{MCC}}$) For any wff α, either $\alpha : c_\alpha \in \mathcal{I}_{\text{MCC}}$ or $\neg\alpha : c_{\neg\alpha} \in \mathcal{I}_{\text{MCC}}$.
For any constraint either $\lambda \preceq \lambda' \in \mathcal{I}_{\text{MCC}}$ or $\lambda \not\preceq \lambda' \in \mathcal{I}_{\text{MCC}}$.

Proof: Suppose that $\alpha : c_\alpha \notin \mathcal{I}_{\text{MCC}}$ and $\neg\alpha : c_{\neg\alpha} \notin \mathcal{I}_{\text{MCC}}$ then $\mathcal{I}_{\text{MCC}} + [\alpha : c_\alpha] \vdash_{\text{L}} \bot$ and $\mathcal{I}_{\text{MCC}} + [\neg\alpha : c_{\neg\alpha}] \vdash_{\text{L}} \bot$. Hence $\mathcal{I}_{\text{MCC}} + [\alpha : c_\alpha] \vdash_{\text{L}} c_\bot : c_\alpha \circ c_{\neg\alpha}$ (by $(\bot\mathcal{I})$ rule) and and $\mathcal{I}_{\text{MCC}} \vdash_{\text{L}} \neg\alpha : c_{\neg\alpha}$ by the $(\neg\mathcal{I})$ rule. Hence $\mathcal{I}_{\text{MCC}}$ is inconsistent, a contradiction. Notice that as a consequence, if $\alpha : c_\alpha \notin \mathcal{I}_{\text{MCC}}$ then $\neg\alpha : c_{\neg\alpha} \in \mathcal{I}_{\text{MCC}}$.
Similarly, for the case of constraints.

□

There are several lemmas proved next that will be used in proving that a particular model of $\mathcal{I}_{\text{MCC}}$ satisfies the axioms in $\mathcal{A}^+_{\mathcal{L}}$. Use is made of the facts proved in Lemmas 4.4 and 4.6, that if $\alpha : \lambda \in \mathcal{I}_{\text{MCC}}$ then $\alpha : c_\alpha \in \mathcal{I}_{\text{MCC}}$ and if $\alpha : c_\alpha \notin \mathcal{I}_{\text{MCC}}$ then $\neg\alpha : c_{\neg\alpha} \in \mathcal{I}_{\text{MCC}}$.

Lemma 4.7 (*Property of* $\otimes$)
$\alpha \otimes \beta : \lambda \in \mathcal{I}_{\text{MCC}}$ iff $\alpha : c_\alpha \in \mathcal{I}_{\text{MCC}}$, $\beta : c_\beta \in \mathcal{I}_{\text{MCC}}$ and $c_\alpha \circ c_\beta \preceq \lambda$.

Proof: Suppose $\alpha \otimes \beta : \lambda \in \mathcal{I}_{\text{MCC}}$ but $\alpha : c_\alpha \notin \mathcal{I}_{\text{MCC}}$. Then $\mathcal{I}_{\text{MCC}} + [\alpha : c_\alpha] \vdash_{\text{L}} \bot$ and hence, since $\mathcal{I}_{\text{MCC}} \vdash_{\text{L}} \mathcal{I}_{\text{MCC}} + [\alpha : c_\alpha]$, $\mathcal{I}_{\text{MCC}}$ is inconsistent, a contradiction. Similarly for $\beta : c_\beta$ and $c_\alpha \circ c_{\neg\alpha} \preceq \lambda$.

For the other direction suppose $\alpha : c_\alpha \in \mathcal{I}_{\text{MCC}}$, $\beta : c_\beta \in \mathcal{I}_{\text{MCC}}$ and $c_\alpha \circ c_\beta \preceq \lambda \in \mathcal{I}_{\text{MCC}}$. Suppose also that $\alpha \otimes \beta : c_\alpha \circ c_\beta \notin \mathcal{I}_{\text{MCC}}$. Then $\mathcal{I}_{\text{MCC}} + [\alpha \otimes \beta : c_\alpha \circ c_\beta] \vdash_{\text{L}} \bot$. But since $\mathcal{I}_{\text{MCC}} \vdash_{\text{L}} \alpha \otimes \beta : c_\alpha \circ c_\beta$, this implies $\mathcal{I}_{\text{MCC}}$ is inconsistent, a contradiction. Thus $\alpha \otimes \beta : c_\alpha \circ c_\beta \in \mathcal{I}_{\text{MCC}}$ and $\alpha \otimes \beta : \lambda \in \mathcal{I}_{\text{MCC}}$ since $c_\alpha \circ c_\beta \preceq \lambda \in \mathcal{I}_{\text{MCC}}$ (by Lemma 4.4).

□

Lemma 4.8 (*Property of → (part (i))*)
If $\alpha \to \beta : \lambda \in \mathcal{I}_{\text{MCC}}$ and $\alpha : \lambda' \in \mathcal{I}_{\text{MCC}}$ then $\beta : \lambda \circ \lambda' \in \mathcal{I}_{\text{MCC}}$.

Suppose $\beta : \lambda \circ \lambda' \notin \mathcal{I}_{\text{MCC}}$, then $\mathcal{I}_{\text{MCC}} + [\beta : \lambda \circ \lambda']$ is inconsistent and by the $(\to\mathcal{E})$ rule $\mathcal{I}_{\text{MCC}} \vdash_{\text{L}} \beta : \lambda \circ \lambda'$, so $\mathcal{I}_{\text{MCC}}$ is inconsistent, a contradiction. Hence $\beta : \lambda \circ \lambda' \in \mathcal{I}_{\text{MCC}}$.

□

Lemma 4.9 (*Property of → (part (ii))*)
If $\alpha : c_\alpha \in \mathcal{I}_{\text{MCC}}$ implies that $\to \beta : \lambda \circ c_\alpha \in \mathcal{I}_{\text{MCC}}$, then $\alpha \to \beta : \lambda \in \mathcal{I}_{\text{MCC}}$

Proof: Suppose that either $\alpha : c_\alpha \notin \mathcal{I}_{\text{MCC}}$ or $\beta : \lambda \circ c_\alpha \in \mathcal{I}_{\text{MCC}}$. and that $\beta : \lambda \circ c_\alpha \notin \mathcal{I}_{\text{MCC}}$ and so $\mathcal{I}_{\text{MCC}} + [\alpha \to \beta : \lambda]$ is inconsistent.

First, if $\beta : \lambda \circ c_\alpha \in \mathcal{I}_{\text{MCC}}$ then $\mathcal{I}_{\text{MCC}} \vdash_{\text{L}} \alpha \to \beta : \lambda$ by the $(\to\mathcal{I})$ rule, $\mathcal{I}_{\text{MCC}}$ is inconsistent, a contradiction.

On the other hand, suppose $\alpha : c_\alpha \notin \mathcal{I}_{\text{MCC}}$. Then $\mathcal{I}_{\text{MCC}} + [\alpha : c_{\neg\alpha}] \vdash_{\text{L}} \bot$. Hence, by the $(\bot\mathcal{I})$ rule, $\mathcal{I}_{\text{MCC}} + [\alpha : c_\alpha \vdash_{\text{L}} \beta : \lambda \circ c_\alpha$, and $\mathcal{I}_{\text{MCC}} \vdash_{\text{L}} \alpha \to \beta : \lambda$ and $\mathcal{I}_{\text{MCC}}$ is again inconsistent.

□

Together, Lemmas 4.8 and 4.9 show that $\alpha \to \beta : \lambda \in \mathcal{I}_{\text{MCC}}$ iff forall x if $\alpha : x \in \mathcal{I}_{\text{MCC}}$ then $\beta : \lambda x \in \mathcal{I}_{\text{MCC}}$.

Lemma 4.10 (*Property of* ¬¬)

Proof: Suppose $\neg\neg\alpha : \lambda \in \mathcal{I}_{\text{MCC}}$, but $\alpha : \lambda \notin \mathcal{I}_{\text{MCC}}$. Then $\mathcal{I}_{\text{MCC}} + [\alpha : \lambda]$ is inconsistent. By the $(\neg\neg)$ rule $\mathcal{I}_{\text{MCC}}$ is inconsistent.

□

Lemma 4.11 (*Property of* ¬) If $\neg\alpha : \lambda \in \mathcal{I}_{\text{MCC}}$ and $\alpha : \lambda' \in \mathcal{I}_{\text{MCC}}$ then $\bot : \lambda \circ \lambda' \in \mathcal{I}_{\text{MCC}}$ and if $\alpha : c_\alpha \in \mathcal{I}_{\text{MCC}}$ implies that $\bot : c_\alpha \circ \lambda \in \mathcal{I}_{\text{MCC}}$, then $\neg\alpha : \lambda \in \mathcal{I}_{\text{MCC}}$.

Proof: First, if $\bot : \lambda \circ \lambda' \notin \mathcal{I}_{\text{MCC}}$, then $\mathcal{I}_{\text{MCC}} + \bot : \lambda \circ \lambda' \vdash_{\text{L}} \bot$. But if $\neg\alpha : \lambda \in \mathcal{I}_{\text{MCC}}$ and $\alpha : \lambda' \in \mathcal{I}_{\text{MCC}}$ then $\mathcal{I}_{\text{MCC}} \vdash_{\text{L}} \bot : \lambda \circ \lambda'$. But then $\mathcal{I}_{\text{MCC}}$ is inconsistent.
Second, suppose either $\alpha : c_\alpha \notin \mathcal{I}_{\text{MCC}}$ or $\bot : \lambda \circ c_\alpha \in \mathcal{I}_{\text{MCC}}$ and $\neg\alpha : \lambda \notin \mathcal{I}_{\text{MCC}}$, so $\mathcal{I}_{\text{MCC}} + [\neg\alpha : \lambda] \vdash_{\text{L}} \bot$. There are then two subcases: either $\bot : \lambda \circ c_\alpha \in \mathcal{I}_{\text{MCC}}$ or $\alpha : c_\alpha \notin \mathcal{I}_{\text{MCC}}$. If $\bot : \lambda \circ c_\alpha \in \mathcal{I}_{\text{MCC}}$, then $\mathcal{I}_{\text{MCC}} \vdash_{\text{L}} \neg\alpha : \lambda$ and $\mathcal{I}_{\text{MCC}}$ is inconsistent, a contradiction. If $\alpha : c_\alpha \notin \mathcal{I}_{\text{MCC}}$, then $\mathcal{I}_{\text{MCC}} + [\alpha : c_\alpha] \vdash_{\text{L}} \bot$ and $\mathcal{I}_{\text{MCC}} + [\alpha : c_\alpha] \vdash_{\text{L}} \bot : c_\alpha \circ \lambda$. Hence $\mathcal{I}_{\text{MCC}} \vdash_{\text{L}} \neg\alpha : \lambda$ and $\mathcal{I}_{\text{MCC}}$ is inconsistent. Both cases lead to a contradiction.

□

The above lemmas are now used to prove a Model Existence Lemma for L_{CLDS}. If $\mathcal{I}_{\text{MCC}}$ is a maximally consistent set of declarative units and constraints, then a Herbrand interpretation $\mathcal{H}_{\mathcal{A}}$ can be constructed that is a model of the axioms $\mathcal{A}^+_{\mathcal{L}}$. $\mathcal{H}_{\mathcal{A}}$ is defined as in Definition 3.7. According to this definition $[\alpha]^*(x) = T$ iff $\alpha : x \in \mathcal{I}_{\text{MCC}}$. $\mathcal{H}_{\mathcal{A}}$ is shown to be a model of the axioms $\mathcal{A}^+_{\mathcal{L}}$ by utilising the Lemmas 4.4 to 4.11.

Lemma 4.12 (*Model Existence Lemma*) The Herbrand interpretation $\mathcal{H}_{\mathcal{A}}$ is a model of the extended algebra $\mathcal{A}^+_{\mathcal{L}}$.

Proof:

Ax1 This follows directly from Lemma 4.4.

Ax2 This also follows from Lemma 4.4.

Ax3a This follows from Lemma 4.8.

Ax3b This follows from Lemma 4.9.

Ax4a,4 b These follow from Lemma 4.11.

Ax5a, 5b These follow from Lemma 4.7.

Ax6 This follows from Lemma 4.10.

Ax7 This follows from Lemma 4.4.

Properties (1 - 4) of $\circ$ These follow from Lemma 4.3.

□

Theorem 4.2 (*Completeness of* L_{CLDS}) Let $\mathcal{C}$, $\mathcal{C}'$ be two configurations such that $\mathcal{C}' - \mathcal{C}$ is a finite configuration. If $\mathcal{C} \vdash_{\text{L}} \mathcal{C}'$ then $\mathcal{C} \models_{\text{L}} \mathcal{C}$.

Proof: Let $\mathcal{C}$ be a configuration such that $\mathcal{C} \nvdash_{\text{L}} \mathcal{C}'$. Then there is a declarative unit or constraint π, where π is either $\alpha : \lambda$ or $\lambda \preceq \lambda'$, such that $\pi \in \mathcal{C}'$ and $\mathcal{C} \nvdash_{\text{L}} \pi$. There are two cases. In both cases a model for $\mathcal{A}^+$, FOT($\mathcal{C}$) is obtained that is not a model for FOT($\mathcal{C}'$). (Case 1) If π is $\lambda \preceq \lambda'$ then $\mathcal{C} + [\lambda \npreceq \lambda']$ is consistent by Lemma 4.2 and there is a model that makes $\mathcal{A}^+$, FOT(C) true but $\lambda \preceq \lambda'$ false. (Case 2) On the other hand, if π is $\alpha : \lambda$,then $\mathcal{C} + [\neg\alpha : c_{\neg\alpha}] + c_{\bot} \npreceq \lambda \circ c_{\neg\alpha}]$ is consistent and can be expanded into a maximally consistent configuration $\mathcal{I}_{\text{MCC}}$ from which a model $\mathcal{H}_{\mathcal{A}}$ is constructed that makes $\neg\alpha : c_{\neg\alpha}$ and $c_{\bot} \npreceq \lambda \circ c_{\neg\alpha}$ both true. Hence $\mathcal{I}_{\text{MCC}}$ makes $\alpha : \lambda$ false, because if not, and $\alpha : \lambda$ were true, $\bot : \lambda \circ c_{\alpha}$ and $c_{\bot} \preceq \lambda \circ c_{\alpha}$ would be true. But the latter yields a contradiction, hence $\alpha : \lambda$ is false.

□

4.5 Correspondence of L_{CLDS} with a Standard Sequent Calculus Presentation

In order to show that the natural deduction CLDS presented here for linear logic (LL) does indeed correspond to a standard sequent calculus presentation it is necessary to show that theorems in the two systems correspond. It is shown constructively in Theorem 4.4 that theorems in LL obtained from a standard sequent presentation of LL are also theorems in the natural deduction system and in Theorem 4.5 it is shown that theorems in the ND system are also theorems in LL.

A single conclusion presentation of the sequent calculus for LL, presented in Table 7, is the most appropriate version to use in this work since it mirrors the natural deduction rules most closely. See [Gol90] for a study of the relationship between various sequent presentations. The rule for ¬¬ on the right is derivable from the ¬R and ¬L1 rules given The two sides of a sequent $X \Longrightarrow Y$ will be called Left and Right respectively. The upper part(s) of a sequent rule is(are) called the antecedent(s) and the lower part(s) is(are) called the conclusion.

The atoms $\bot$ and $\top$ also play a special role in a sequent, namely, $\Gamma \Longrightarrow \emptyset$ is equivalent to $\Gamma \Longrightarrow \bot$ and $\emptyset \Longrightarrow \Delta$ is equivalent to $\top \Longrightarrow \Delta$. In a single-conclusion sequent calculus for LL the only sequent involving $\top$ or $\bot$ that is not trivial is included in the table as rule (id1). In Theorem 4.5 the (CUT) rule is used in the sequent proof derived from a ND proof. Since (CUT) is admissible in the sequent calculus this is not a problem. Because the exchange rule is present, the cut rule can be written as

$$\frac{\Gamma_1 \Longrightarrow \alpha \quad \alpha, \Gamma_2 \Longrightarrow \beta}{\Gamma_1, \Gamma_2 \Longrightarrow \beta} \quad (\textsc{cut})$$

(id1)	$\frac{\Gamma \Longrightarrow \delta}{\Gamma, \top \Longrightarrow \delta}$	$\frac{\Gamma_1, \beta, \alpha, \Gamma_2 \Longrightarrow \delta}{\Gamma_1, \alpha, \beta, \Gamma_2 \Longrightarrow \delta}$	(exchL)
(ax)	$\frac{}{\alpha \Longrightarrow \alpha}$	$\frac{\Gamma \Longrightarrow \alpha}{\Gamma, \neg\alpha \Longrightarrow \bot}$	($\neg$L1)
($\neg$R)	$\frac{\Gamma, \alpha \Longrightarrow \bot}{\Gamma \Longrightarrow \neg\alpha}$	$\frac{\Gamma, \alpha \Longrightarrow \delta}{\Gamma, \neg\neg\alpha \Longrightarrow \delta}$	($\neg\neg$L)
($\otimes$R)	$\frac{\Gamma_1 \Longrightarrow \alpha \quad \Gamma_2 \Longrightarrow \beta}{\Gamma_1, \Gamma_2 \Longrightarrow \alpha \otimes \beta}$	$\frac{\Gamma, \alpha, \beta \Longrightarrow \delta}{\Gamma, \alpha \otimes \beta \Longrightarrow \delta}$	($\otimes$L)
($\rightarrow$R)	$\frac{\Gamma, \alpha \Longrightarrow \beta}{\Gamma \Longrightarrow \alpha \rightarrow \beta}$	$\frac{\Gamma_1 \Longrightarrow \alpha \quad \Gamma_2, \beta \Longrightarrow \delta}{\Gamma_1, \Gamma_2, \alpha \rightarrow \beta \Longrightarrow \delta}$	($\rightarrow$L)

Table 7: Single Conclusion Sequent Rules for linear logic

The proof of correspondence is similar in spirit to that used in [DG94] to prove Propositions 4 and 5. However, the details are different due to the different approach being taken here with respect to the semantics.

Theorem 4.3 (Translation of Sequents into L_{CLDS}) Let P be a sequent calculus proof in LL of the sequent $\delta_1, \delta_2, \ldots, \delta_n \Longrightarrow \alpha$, $n \geq 0$, then there exists a corresponding ND proof of $\alpha : \lambda_1 \circ \ldots \circ \lambda_n$ from $\delta_1 : \lambda_1, \delta_2 : \lambda_2, \ldots, \delta_n : \lambda_n$, where $\lambda_1, \ldots, \lambda_n$ are arbitrary labels.

Proof:

The proof is by induction on the number of sequents in P. The sequent proof P is assumed to be linearised, such that all sequents required as the antecedent for a step occur before the conclusion of a step. For each type of sequent rule it is shown how to construct a proof of the conclusion sequent from a proof of the antecedent sequent. The notation $\Gamma : \lambda$ is used to represent the data $\delta_1 : \lambda_1, \ldots, \delta_n : \lambda_n$, where $\lambda = \lambda_1 \circ \ldots \circ \lambda_n$. Most of the steps are illustrated in Tables 8 and 9. The notation used for configurations is the same as that used in Section 2. Each configuration $\mathcal{C}_{k+1}$ is the configuration $\mathcal{C}_k$ with the additional identified declarative units or constraints. The notation $\mathcal{C}$ + [declarative unit] indicates an assumption that will be discharged.

Base case A sequent calculus proof with one sequent must consist just of an instance of the (axiom) rule $\alpha \Longrightarrow \alpha$. The corresponding translation uses a special case of the (unit) rule, in particular using the fact that $\lambda \preceq \lambda$, for any λ.

Induction Step Suppose as induction hypothesis that for any sequent proof with m steps, $1 \leq m \leq k$, such that if the last sequent is $\delta_1, \delta_2, \ldots, \delta_n \Longrightarrow \alpha$, $n \geq 0$, then there is a corresponding ND proof of $\alpha : \lambda_1 \circ \ldots \lambda_n$ from $\delta_1 : \lambda_1, \delta_2 : \lambda_2, \ldots, \delta_n : \lambda_n$, where the $\{\lambda_i\}$ are arbitrary labels. It is shown that any sequent proof with $k+1$ steps satisfies the theorem as follows: First, the induction hypothesis is applied to give a ND proof(s) $P1(P2)$ of the antecedent sequent(s); second, some substitutions are possibly made, in a systematic way, to some of the labels in $P1(P2)$ so that $P1$ (and $P2$ if it exists) may be used to give a proof of the conclusion sequent. The substitution stage uses the Substitution Lemma, Lemma 4.13.

	$\neg$R: ND rule is $\neg\mathcal{I}$	
$\mathcal{C}_0\langle\Gamma:\lambda_1\rangle$ $\mathcal{C}_0+[\alpha:\lambda_2]$ $\vdots$ $\mathcal{C}_2\langle\bot:\lambda_1\circ\lambda_2\rangle$	(c_α for λ_2)	$\mathcal{C}_0\langle\Gamma:\lambda_1\rangle$ $\mathcal{C}_0+[\alpha:c_\alpha]$ $\vdots$ $\mathcal{C}_2\langle\bot:\lambda_1\circ c_\alpha\rangle$ $\mathcal{C}_3\langle\neg\alpha:\lambda_1\rangle$ $(\neg\mathcal{I})$
	$\otimes$R: ND rule is $\otimes\mathcal{I}$	
$\mathcal{C}_0\langle\Gamma_1:\lambda_1\rangle$ $\vdots$ $\mathcal{C}_1\langle\alpha:\lambda_1\rangle$ $\mathcal{C}_2\langle\Gamma_2:\lambda_2\rangle$ $\vdots$ $\mathcal{C}_3\langle\beta:\lambda_2\rangle$		$\mathcal{C}_4\langle\Gamma:\lambda_1,\ \Gamma_2:\lambda_2\rangle$ $\vdots$ $\mathcal{C}_5\langle\alpha:\lambda_1\rangle$ $\vdots$ $\mathcal{C}_6\langle\beta:\lambda_2\rangle$ $\mathcal{C}_7\langle\alpha\otimes\beta:\lambda_1\circ\lambda_2\rangle$ $(\otimes\mathcal{I})$
	$\rightarrow$R: ND rule is $\rightarrow\mathcal{I}$	
$\mathcal{C}_0\langle\Gamma:\lambda_1\rangle$ $\mathcal{C}_0+[\alpha:\lambda_2]$ $\vdots$ $\mathcal{C}_2\langle\beta:\lambda_1\circ\lambda_2\rangle$	(c_α for λ_2)	$\mathcal{C}_0\langle\Gamma:\lambda_1\rangle$ $\mathcal{C}_0+[\alpha:c_\alpha]$ $\vdots$ $\mathcal{C}_2\langle\beta:\lambda_1\circ c_\alpha\rangle$ $\mathcal{C}_3\langle\alpha\rightarrow\beta:\lambda_1\rangle$ $(\rightarrow\mathcal{I})$

Table 8: ND translation of sequent rules

Each sequent rule is next considered in turn and the appropriate substitutions (if any) and transformation is given. In most cases Tables 8 and 9 give both of these, in which the derivation on the left represents the proof given by the inductive hypothesis and the derivation on the right represents the proof obtained by means of the label substitution indicated.

id1 Given a ND proof $P1$ of $\delta:\lambda$ from $\Gamma:\lambda$, the addition of the data $\top:1$ will still give a proof of the conclusion sequent since $1\circ\lambda=\lambda$. (Formally, the (unit) rule deduces $\delta:1\circ\lambda$ from $\delta:\lambda$.)

ax The required proof is simply an instance of the (unit) rule, as described for the base case.

exchL The assumption yields a proof from $\Gamma_1:\lambda_1,\beta:\lambda_2,\alpha:\lambda_3,\Gamma_2:\lambda_4$ of $\delta:\lambda_1\circ\lambda_2\circ\lambda_3\circ\lambda_4$. The effect of this rule is an application of the (unit) rule from $\delta:\lambda_1\circ\lambda_2\circ\lambda_3\circ\lambda_4$ to $\delta:\lambda_1\circ\lambda_3\circ\lambda_2\circ\lambda_4$. This is sanctioned by the commutative and associative properties.

¬¬L A proof of $\delta:\lambda_1\circ\lambda_2$ from $\Gamma:\lambda_1$ and $\alpha:\lambda_2$ can be turned into a proof of $\delta:\lambda_1\circ\lambda_2$ from $\Gamma:\lambda_1$ and $\neg\neg\alpha:\lambda_2$ by deriving $\alpha:\lambda_2$ from $\neg\neg\alpha:\lambda_2$ by the $(\neg\neg)$ rule.

¬R, ¬L1, ⊗R, ⊗L, →R, →L These steps appear in Tables 8 and 9.

□

¬L1: uses ¬$\mathcal{E}$

$\mathcal{C}_0\langle \Gamma_1 : \lambda_1 \rangle$		$\mathcal{C}_4\langle \Gamma : \lambda_1,\ \neg\alpha : \lambda_3 \rangle$
$\vdots$		$\vdots$
$\mathcal{C}_1\langle \alpha : \lambda_1 \rangle$		$\mathcal{C}_5\langle \alpha : \lambda_1 \rangle$
		$\mathcal{C}_6\langle \bot : \lambda_3 \circ \lambda_1 \rangle \quad (\neg\mathcal{E})$

⊗L: ND rule is ⊗$\mathcal{E}$

$\mathcal{C}_0\langle \Gamma_1 : \lambda_1 \rangle$		$\mathcal{C}_4\langle \Gamma : \lambda_1,\ \alpha \otimes \beta : \lambda_4 \rangle$
$\mathcal{C}_1\langle \alpha : \lambda_2 \rangle$	$(c_\alpha$ for $\lambda_2)$	$\mathcal{C}_5\langle \alpha : c_\alpha,\ \beta : c_\beta,\ c_\alpha \circ c_\beta \preceq \lambda_4 \rangle \quad (\otimes\mathcal{E})$
$\mathcal{C}_3\langle \beta : \lambda_3 \rangle$	$(c_\beta$ for $\lambda_3)$	$\vdots$
$\vdots$		$\mathcal{C}_6\langle \delta : \lambda_1 \circ c_\alpha \circ c_\beta \rangle$
$\mathcal{C}_3\langle \delta : \lambda_1 \circ \lambda_2 \circ \lambda_3 \rangle$		$\mathcal{C}_7\langle \delta : \lambda_1 \circ \lambda_4 \rangle$ ((unit) rule) $(\lambda_1 \circ c_\alpha \circ c_\beta \preceq \lambda_1 \circ \lambda_4)$

→L: uses →$\mathcal{E}$

$\mathcal{C}_0\langle \Gamma_1 : \lambda_1 \rangle$		
$\vdots$		$\mathcal{C}_4\langle \Gamma_1 : \lambda_1,\ \Gamma_2 : \lambda_2,\ \alpha \rightarrow \beta : \lambda_4 \rangle$
$\mathcal{C}_1\langle \alpha : \lambda_1 \rangle$		$\vdots$
	$(\lambda_4 \circ \lambda_1$ for $\lambda_3)$	$\mathcal{C}_5\langle \alpha : \lambda_1 \rangle$
		$\mathcal{C}_6\langle \beta : \lambda_4 \circ \lambda_1 \rangle \quad (\rightarrow\mathcal{E})$
$\mathcal{C}_2\langle \Gamma_2 : \lambda_2, \beta : \lambda_3 \rangle$		$\vdots$
$\vdots$		$\mathcal{C}_7\langle \delta : \lambda_2 \circ \lambda_4 \circ \lambda_1 \rangle$
$\mathcal{C}_3\langle \delta : \lambda_2 \circ \lambda_3 \rangle$		

Table 9: ND translation of sequent rules (continued)

Lemma 4.13 (*Substitution Lemma*)

If the sequence of configurations P is a L_{CLDS} derivation from a configuration $\mathcal{C}\langle \delta_1 : \lambda_1, \delta_2 : \lambda_2, \ldots, \delta_n : \lambda_n \rangle$, where the $\{\lambda_i\}$ are arbitrary labels then the sequence of configuration P' obtained by making none or more substitutions of the form "substitute λ' for λ_i" in P, where λ_i is not of the form c_{δ_i}, and λ' is any label, then P' is a L_{CLDS} derivation.

Outline Proof

The restriction on the label λ_i, which is replaced by λ', is necessary, for the only allowed label in such a case is $\delta_i : c_{\delta_i}$ so it cannot be replaced. The replacements are made in a systematic way in that all are considered to be made simultaneously and to all uses of the λ_i label to derive P' from P. The ND proof P' is still correct as the changes are syntactic. □

As a special case of Theorem 4.3, when $n = 0$, $\lambda_1 \circ \ldots \circ \lambda_n = 1$ as 1 is the identity element of $\circ$. This proves the following Theorem.

Theorem 4.4 (*Correspondence Part I*) Let P be a sequent calculus proof in LL of the theorem α, i.e. $\Longrightarrow \alpha$, then there exists a corresponding ND proof of $\alpha : 1$ i.e. $\mathcal{C}_\emptyset \vdash_{\text{L}} \alpha : 1$.

In ND the structural rule (exchL) is most easily accommodated by allowing permutations of the atomic labels in a composite label. It is not difficult to show that the structural rules can be accumulated and applied just before an application of the (unit) rule.

When only constructing a ND proof without the initial sequent proof, the particular distribution of data into Γ_1 and Γ_2 (where required) is not known, nor are the required applications of structural rules. To avoid having to guess these steps, free variable ND rules can be formulated. For a full description of a free variable approach see [BFR97], in which an algorithm is given that enables the particular applications of the new unit rule used in place of structural rules to be found.

Theorem 4.5 (*Correspondence Part II*)

If there exists a ND proof in L_{CLDS} of the declarative unit $\alpha : 1$ from an empty initial configuration then there is a LL sequent calculus proof of the theorem $\Longrightarrow \alpha$, that is

$$\Longrightarrow \alpha \text{ if } \mathcal{C}_\emptyset \vdash_{\text{L}} \alpha : 1$$

Proof:
Suppose $\mathcal{C}_\emptyset \vdash_{\text{L}} \alpha : 1$. It is required to show $\Longrightarrow \alpha$. By the soundness of natural deduction $\mathcal{C}_\emptyset \models_{\text{L}} \alpha : 1$, so $\mathcal{A}^+_{\mathcal{L}}$, FOT($\mathcal{C}_\emptyset$) $\models [\alpha]^*(1)$, (or $\mathcal{A}^+_{\mathcal{L}} \models [\alpha]^*(1)$). Hence any model of $\mathcal{A}^+_{\mathcal{L}}$ is also a model of $[\alpha]^*(1)$. If a model of $\mathcal{A}^+_{\mathcal{L}}$ can be constructed such that $[\alpha]^*(1) = T$ iff $\Longrightarrow \alpha$, then this will yield $\Longrightarrow \alpha$ as required. Such a model does exist, it is based on the canonical interpretation first introduced in [DG94]. Lemma 4.14 shows there is a suitable model that is a model of $\mathcal{A}$ and Lemma 4.15 shows it is also a model of $\mathcal{A}^+1$.

Definition 4.4 (*Canonical Interpretation*)

Let $\langle \mathcal{L}_P, \mathcal{L}_{\mathcal{L}} \rangle$ be a L_{CLDS}. A *canonical interpretation* is an interpretation from $Mon(\mathcal{L}_P, \mathcal{L}_L)$ defined as follows:

- each characteristic label c_α is interpreted as $\{z | \alpha \Longrightarrow z\}$.
- each constant label λ is interpreted as $\emptyset$.
- $||\lambda \circ \lambda'|| = \{z | \alpha \otimes \beta \Longrightarrow z,\ \text{where } \alpha \in ||\lambda|| \text{ and } \beta \in ||\lambda'||\}$.
- is interpreted as $\{z | \Longrightarrow z\}$
- $x \preceq y$ is interpreted as $||x|| \subseteq ||y||$
- $||[\alpha]^*|| = \{||x|| : \alpha \in ||x||\}$

□

On the basis of the above interpretation, an atom $[\alpha]^*(x) = T$ iff $\alpha \in ||x||$. This means, in particular, that if $[\alpha]^*(1) = T$ then $\alpha \in ||1||$ and hence $\Longrightarrow \alpha$. In Lemma 4.15 it is shown that the above interpretation is a model of the extended algebra $\mathcal{A}^+_{\mathcal{L}}$.

Notice that $||c_\perp|| = \{z | \perp \Longrightarrow z\}$ and so $||c_\perp|| = \{\perp\}$. Therefore (Ax 7) is satisfied immediately.

In the following lemmas the interpretation notation $||\ ||$ will be omitted for simplicity.

Lemma 4.14 (*General properties of the canonical interpretation)*

The interpretation of $\circ$ as given by the canonical interpretation forms a monoid operation with identity 1 and is order preserving, and $\subseteq$ is a preorder.

Proof:

commutativity and associativity : Taking commutativity first, suppose $\delta \in \lambda \circ \lambda'$ then there is a sequent calculus proof of δ from $\alpha \otimes \beta$, where $\alpha \in \lambda$ and $\beta \in \lambda'$. That is, $\alpha \otimes \beta \Longrightarrow \delta$ Since $\beta \otimes \alpha \Longrightarrow \alpha \otimes \beta$, by the (cut) rule $\beta \otimes \alpha \Longrightarrow \delta$ and $\delta \in \lambda' \circ \lambda$.

The proof for associativity of $\circ$ is similar.

identity : Suppose $\delta \in 1 \circ \lambda$ then $\beta \otimes \alpha \Longrightarrow \delta$, where $\Longrightarrow \beta$ and $\gamma \Longrightarrow \alpha$ for $\gamma \in \lambda$. Using the sequent rule $\otimes$R the sequent $\gamma \Longrightarrow \beta \otimes \alpha$ is derived and hence $\gamma \Longrightarrow \delta$ by the (cut) rule.

For the other direction, suppose $\delta \in \lambda$, so $\gamma \Longrightarrow \delta$ for some $\gamma \in \lambda$. Then $\top \otimes \gamma \Longrightarrow \delta$ by the $\otimes$L rule and (id1) and $\delta \in 1 \circ \lambda$, since $\top \in 1$.

order-preservedness : Suppose $\lambda \subseteq \lambda'$ and $\delta \in \lambda \circ \lambda''$. Then $\alpha \otimes \beta \Longrightarrow \delta$, where $\alpha \in \lambda$ and $\beta \in \lambda''$. Hence $\alpha \in \lambda'$ and $\alpha \otimes \beta \Longrightarrow \delta$ and $\delta \in \lambda \circ \lambda'$.

pre-order This follows immediately as $\subseteq$ is reflexive and transitive.

$\square$

Lemma 4.15 (*Herbrand model*) The Canonical Interpretation given in Definition 4.4 is a model of the axioms in Table 6.

Proof: Let $\langle \mathcal{L}_P, \mathcal{L}_{\mathcal{L}} \rangle$ be a substructural CLDS and $\mathcal{H}_{\mathcal{A}}$ be the Herbrand model based on the canonical interpretation. In each of the cases below the definitions $[\alpha]^*(x) = T$ iff $\alpha \in x$ and $x \preceq y$ iff $x \subseteq y$ are used implicitly. Notice also that if $\alpha \in x$ for some α and x, then $\gamma \Longrightarrow \alpha$ for some $\gamma \in x$. The (exchangeL) rule of sequent calculus is also used freely.

Ax1 Suppose $\alpha \in x$ and $x \subseteq y$, then $\alpha \in y$ as required.

Ax2 Suppose $\alpha \in x$, then $\gamma \Longrightarrow \alpha$ for some $\gamma \in x$. Notice that $\alpha \in c_\alpha$ since $\alpha \Longrightarrow \alpha$. But also, if $\delta \in c_\alpha$ then $\alpha \Longrightarrow \delta$ and hence $\gamma \Longrightarrow \delta$, so $\delta \in x$ and $c_\alpha \preceq x$.

Ax3a Suppose $\alpha \rightarrow \beta \in x$ and $\alpha \in y$, hence $\gamma \Longrightarrow \alpha \rightarrow \beta$ and $\delta \Longrightarrow \alpha$ for some $\gamma \in x$ and $\delta \in y$. Therefore $\gamma, \alpha \Longrightarrow \beta$ and by the (cut) rule $\gamma, \delta \Longrightarrow \beta$, hence $\gamma \otimes \delta \Longrightarrow \beta$ and $\beta \in x \circ y$.

Ax3b Since $\alpha \in c_\alpha$, suppose $\beta \in x \circ c_\alpha$. Then $\gamma \otimes \alpha \Longrightarrow \beta$, where $\gamma \in x$. Hence, since $\gamma, \alpha \Longrightarrow \gamma \otimes \alpha$, by the (cut) rule $\gamma, \alpha \Longrightarrow \beta$ and $\gamma \Longrightarrow \alpha \rightarrow \beta$. Therefore $\alpha \rightarrow \beta \in x$ as required.

Ax4a Suppose $\neg\alpha \in x$ and $\alpha \in y$, then $\gamma \Longrightarrow \neg\alpha$ and $\delta \Longrightarrow \alpha$, where $\gamma \in x$ and $\delta \in y$. Hence the sequents $\delta, \neg\alpha \Longrightarrow \bot$ and then $\gamma, \alpha \Longrightarrow \bot$ can be derived by the rules ($\neg$L1) and (cut) and then also the sequent $\gamma \Longrightarrow \neg\delta$. Therefore $\neg\delta \in x$ and so $\bot \in x \circ y$ since $\delta \otimes \neg\delta \Longrightarrow \bot$.

Ax4b Since $\alpha \in c_\alpha$, suppose $\bot \in x \circ c_\alpha$, that is $\gamma, \alpha \Longrightarrow \bot$ for some $\gamma \in x$ (recall $c_\bot = \{\bot\}$). Hence $\gamma \Longrightarrow \neg\alpha$ and $\neg\alpha \in x$ as required.

Ax5a The proofs are similar to that for Ax2.

Ax5b Suppose $\alpha \in x$ and $\beta \in y$, hence $\gamma_1 \Longrightarrow \alpha$ and $\gamma_2 \Longrightarrow \beta$, where $\gamma_1 \in x$ and $\gamma_2 \in y$. Hence $\gamma_1, \gamma_2 \Longrightarrow \alpha \otimes \beta$ and $\alpha \otimes \beta \in x \circ y$.

Ax6 Suppose $\neg\neg\alpha \in x$. Then $\gamma \Longrightarrow \neg\neg\alpha$ for some $\gamma \in x$ and $\gamma, \neg\alpha \Longrightarrow$ so $\gamma \Longrightarrow \alpha$ and $\alpha \in x$.

$\square$

5 Conclusions

This paper illustrates a new method, based on Labelled Deductive Systems [Gab96] for providing logics, belonging to different families, with a uniform presentation of their derivability relations and semantic entailments. Section 2 provides the basic definitions of a CLDS framework and describes the main features of a CLDS natural deduction system and of a CLDS model theoretic semantics. The notion of a configuration in a CLDS system (and therefore in both the $\mathrm{E_{CLDS}}$ and $\mathrm{L_{CLDS}}$ systems) generalises the standard notion of a theory. Configurations can be used to describe domains containing not just one, but an explicit structure of local theories. Correspondence results shown in Sections 3.2 and 4.5 show that there is a one-way translation of standard (modal and linear) theories into configurations, which preserves both the derivability and semantic entailment relations. The CLDS notions of derivability relation and semantic entailment are *generalised* to relations between structured theories. The CLDS model-theoretic semantics is given in terms of a translation approach into first-order logic. This facilitates a uniform formalisation for standard semantics of different existing logics. Sections 3 and 4 illustrate how the CLDS general approach can be applied to logics of different families, such as the logic of elsewhere and linear logic.

The presentations of the two systems $\mathrm{E_{CLDS}}$ and $\mathrm{L_{CLDS}}$ are both extensions and refinements of the general CLDS system given in Section 2. New inference rules are only included for the specific logical operators. Rules for reasoning about structures of configurations are instead common to both the two logics. The different standard semantics of the logic of elsewhere and of linear logic are captured by appropriately refining the axiomatisation of the extended algebra to the specific meaning of their associated logical operators and by axiomatising the semantic properties of the structures that compose the configurations. For each of these two CLDS systems, $\mathrm{E_{CLDS}}$ and $\mathrm{L_{CLDS}}$, results on the soundness and completeness of their proof theories with respect to their respective semantics are proved. The methodology adopted in these proofs is also uniform to both these two systems. This uniformity makes the CLDS framework an ideal framework not only for facilitating technical studies of existing logics and their combinations, but mainly for providing a technical methodology for the development and investigation of new logics.

From an applicative point of view, the CLDS approach provides a logic with reasoning which is closer to the needs of computing and A.I. These are in fact application areas with an increasing demand for logical systems able to represent and to reason about *structures* of information (see [Gab96]).

As for the logic of elsewhere, Hilbert-style proof systems have alreday been developed [dR92, Seg81] and a first tableaux system has been described in [Dem96]. The $\mathrm{E_{CLDS}}$ is a first example of a natural-deduction proof system for this type of enriched classical modal logics. This natural deduction proof system is uniform also with respect to the natural deduction systems developed for the standard family of modal logics [Rus96]. The set of rules for the elsewhere modal operators, denoted in the literature by $<\neq>$ and $[\neq]$, is identical to the set of rules for the standard normal modal operators. No additional modal rules need to be included to capture the specific semantic meaning of the elsewhere modalities. This is entirely due to the explicit syntactic formalisation of the accessibility relation's properties by means of the labelling algebra, and to its use in reasoning with possible worlds and with relations between possible worlds as part of the modal system. This differs from the tableaux system for the logic of elsewhere described in [Dem96] where specific tableaux rules are introduced for the elsewhere modal operators. Note that a more specific formalisation of the $\mathrm{E}_{\mathrm{CLDS}}$ system could be defined in which R-literals are only of the form $\lambda \neq \lambda'$ and $\lambda = \lambda'$. Modal operators as well as axioms (Ax6)–(Ax8) of the extended algebra would have needed to refer only to these R-literals, making axiom (**E**) irrelevant. This approach has not been chosen here for its lack of generality; generality which is instead preserved by the $\mathrm{E}_{\mathrm{CLDS}}$ system described in Section 3 as discussed above.

The combined approach of the $\mathrm{E}_{\mathrm{CLDS}}$ system (syntactic representation of the possible worlds and accessibility relation) facilitates also an easy extension of the $\mathrm{E}_{\mathrm{CLDS}}$ system to systems which combine the elsewhere operator with other modal operators, such as the "universal" modal operator. Such systems could be achieved by extended the labelling language with binary relations R_i for each modality $\Box_i$ (and $\Diamond_i$) and extending the labelling algebra $\mathcal{A}_{\mathrm{E}}$ with sets of schemas which respectively axiomatise the properties of the added accessibility relatons R_i and then duplicating modal and structural inference rules for each added modality.

The $\mathrm{E}_{\mathrm{CLDS}}$ natural deduction system described in Section 2 is similar to Fitting's prefixed tableaux [Fit83] and to Sympson natural deduction system [Sym93] in that, in both cases, modal formulae are labelled. However, neither of these two systems has ever been extended to the logic of elsewhere. Other work related to the CLDS approach described in this paper can be found in [FS91], [Gen93], [Mas94] and [Bor93]. From an applicative point of view, it seems likely that a message passing system can be represented as configurations, where each constant label is associated with a part of a system and permitted message passing connections are described by R-literals. This is subject of future investigations.

As for linear logic, the $\mathrm{L}_{\mathrm{CLDS}}$ is a proper generalisation of the standard approachs to this logic in that it facilitates explicit assumptions, and reasoning, about relationships between resources. The labelled natural deduction proof system described in Section 4 is in its rules similar, apart from the labels, to the standard sequent calculus for linear logics. Alternative natural deduction rules could have been defined for the $(\neg\mathcal{I})$ and $(\bot\mathcal{I})$ but these would have made the system less related to standard linear logic sequent calculus. Note that the other multiplicative operator (called *par* and denoted with $\wp$) is defined as $\neg\alpha \rightarrow \beta$. Furthermore, the $\mathrm{L}_{\mathrm{CLDS}}$ proof system can be extended to the additive (& and $\vee$) and exponential operators. For example, the semantic meaning of the additive connectives can be captured by including the following axiom schemas in the extended algebra of $\mathrm{L}_{\mathrm{CLDS}}$.

$$\forall x([\alpha \& \beta]^*(x) \leftrightarrow ([\alpha]^*(x) \wedge [\beta]^*(x)$$

$$\forall x([\alpha \vee \beta]^*(x) \leftarrow [\alpha]^*(x)$$

$$\forall x([\alpha \vee \beta]^*(x) \leftarrow [\beta]^*(x)$$

$$\forall x, y([\alpha \vee \beta]^*(x) \rightarrow \\ (([\alpha]^*(c_\alpha) \rightarrow \gamma^*(y \circ c_\alpha)) \wedge ([\beta]^*(c_\beta) \rightarrow \gamma^*(y \circ c_\beta)) \rightarrow [\gamma]^*(x \circ y))$$

One of the benefits of the LDS approach to substructural logic is its uniformity. This is the same for the CLDS system. Only the case of linear logic has been considered here, but other different substructural logics could be equally defined by considering appropriate labelling algebrae. Labelling algebrae for Lambek, Relevance and Intuitionistic logics are defined by incrementally adding the axioms shown in Table 10 to the basic properties (i.e order-preserving, identity and associative) of $\mathcal{A}_L$. The (R-A) rule uses the appropriate labelling algebra to differentiate one logic from another in the uniform $\mathrm{L_{CLDS}}$ proof theory. (See [BFR97] and [DG94] for a full discussion of this issue.) Note that in the case of intuitionistic

1	$x \circ y \preceq y \circ x$	commutativity	Lambek Calculus = { }
2	$x \circ x \preceq x$	contraction	Linear Logic = { 1 }
3	$x \preceq x \circ x$	expansion	Relevance Logic = { 1,2}
4	$x \preceq x \circ y$	monotonicity	Intuitionistic Logic = {1,2,4}

Table 10: Properties of $\circ$ in different logics

logic, the ($\neg\neg$) rule of $\mathrm{L_{CLDS}}$ needs to be replaced by a rule which allows a deduction of $\alpha : \lambda$ from $\bot : \lambda$ for any α and λ, and by not rewriting $\neg A \rightarrow B$ into $A \wp B$.

Other uniform proof systems based on the LDS methodology have been developed for substructural logics. Examples are the LKE system described in [DG94] and the natural deduction system described in [BFR97], where complete lattices are used to implement the labelling algebra of the $\mathrm{L_{CLDS}}$ sytsem. However, as shown in [BFR97, BDR97], a ND system similar to that described in the $\mathrm{L_{CLDS}}$ system yields proofs that are sound and complete with respect to the LKE-system. Another difference between the $\mathrm{L_{CLDS}}$ and the LKE systems regards the rules for the $\neg$ operator. In [DG94] a new lattice operator is introduced, called the "star" operator, which satisfies the basic property that, for any label λ, $\lambda \circ \lambda^* \preceq 1^*$, or equivalently, that $\bot$ is false at $\lambda \circ \lambda^*$. This yields a slightly more uniform system than the $\mathrm{L_{CLDS}}$. In $\mathrm{L_{CLDS}}$ one extra axiom has to replace the $\neg\neg$ axiom (Ax6) in order to capture the intuitionistic logic semantics, namely $\forall x[c_\bot \not\preceq x]$. The extended $\mathrm{L_{CLDS}}$ system could have used the "star" operator approach by adding to the extended algebra the following axiom stating that the star operator exists and is the largest label "consistent" with x:

$$\forall x(c_\bot \not\preceq x \circ x^* \wedge \forall z(c_\bot \not\preceq x \circ z \rightarrow z \preceq x^*))$$

The resulting negation rules in this system can then be shown to be equivalent to the negation rule in LKE.

For the automated theorem proving point of view, the translation method described in Section 3 facilitates the use of first-order therem provers for deriving theorems of the underlying logic. In fact, the first order axioms of a CLDS extended algebra $\mathcal{A}_S^+$ can be translated into clausal form, and so any clausal theorem proving method might be appropriate for using the axioms to automate the process of proving theorems. The clauses resulting from the translation of a particular configuration represent a partial coding of the data. A resolution refutation that simulates the application of natural deduction rules could be developed. Procedural tableau methods such as those described in [Bro91] can be adapted here. By using some form of resolution the choice of labels to provide appropriate instantiations of the axioms, is transferred to the unification algorithm. For example, for LL an appropriate algorithm could be one which uses AC-unification with identity [Sti85]. A free-variable tableau-based theorem prover using this idea is described in [BF95] or a standard Model Elimination prover could be adapted, by incorporating a specialised unification algorithm, for example the one described in [BF95]. Notice also that the clauses obtained from the axioms are quite specific, in that the R-literals are only used together with the properties of the labelling algebra. In the case of the logic of elsewhere, these are the properties of an equality theory and for linear logic these are properties such as order-preserving and transitivity of $\preceq$. A general theorem prover method in which these properties are separated from the main derivation process, such as theory resolution ([Sti85] [Bau92]) would also be appropriate.

In the case of substructural logics, there is another alternative [BR97b]. This is because the clauses of the extended algebra are nearly all Horn clauses (one positive literal at most). The only exceptions are disjunctions with exactly two positive literals. One of these always has the form $[\alpha]^*(c_\alpha)$ for some wff α and the other has the form $[\gamma]^*(x)$. A theorem prover which uses an adaptation of the Davis Putnam method [CL73] has been built in Prolog for the subcase of wffs using just the $\rightarrow$ and $\neg$ operators. Wffs $A \otimes B$ involving $\otimes$ are rewritten as $\neg(A \rightarrow \neg B)$. The inclusion of further operators such as the additive operators and the exponential operator of LL is currently being investigated. In the case of intuitionistic logic the adapted Davis Putnam method turns out to be particularly simple due to the presence of monotonicity. For in this case $1 \preceq x$ for any label x, and so an assumption $[\alpha]^*(x)$ can be replaced by $[\alpha]^*(1)$ making the whole set of clauses ground.

In the case of modal logics, the clauses resulting from the translation of the extended algebra include, in general, terms involving function symbols as well as non-Horn clauses. Adapting Davis Putnam is therefore not appropriate, but the general theorem prover Otter [McC90], which uses hyper-resolution and also paramodulation, can be used. An example of using Otter to prove the axiom $(\Box p \wedge p) \rightarrow \Box\Box p$, is illustrated in Figure 7. In this example, the translation into standard clauses uses the *holds(x,y)* predicate, in which x is a wff and y is a label. The functors i, a and b represent $\rightarrow$, $\wedge$ and $\Box$ respectively. The functor $box(x, y)$ represents the term $box_x(y)$. Clauses (1) and (2) arise from the axiom for implication introduction, clauses (3) and (4) come from the axiom for $\Box$ introduction and (5) and (6) arise from the axiom for conjunction elimination. Clause (7) defines the R relation, whilst (8) is the axiom for $\Box$ elimination. Clause (9) is the negated conclusion. Prolog style variables (capital letters) are used. The proof is made with the *unit-resulting* and *binary resolution* options and the reader can see that it mirrors the proof given in the text very closely and generated 53 clauses. Other options generated slightly different proofs. Further investigation is necessary on the automated theorem provering aspect of the CLDS approach. However the little results obtained from the above initial investigation makes this line of research promising.

6 Acknowledgements

The authors acknowledge Marcello D'Agostino for helpful and insightful discussions.

References

[Bau92] P. Baumgartner. A model elimination calculus with built-in theories. In *Proceedings of GWAI-92, Lecture Note in Artificial Intelligence 671*. Springer Verlag, 1992.

[BDR97] K. Broda, M. D'Agostino, and A. Russo. Transformation methods in LDS. In *Logic, Language and Reasoning. An Essay in Honor of Dov Gabbay.* Kluwer Academic Publishers, To appear 1997.

[BF95] K. Broda and M. Finger. KE-tableaux for a fragment of linear logic. Technical report, 4th Workshop on Theorem Proving with Analytic Tableaux and Related Methods, Ed. Peter Baumgartner, University of Koblenz, 1995.

[BFR97] K. Broda, M. Finger, and A. Russo. LDS-Natural Deduction for Substructural Logics. Technical Report DOC. 97/11, Imperial College, Department of Computing, 1997.

```
list(usable).
1 [] holds(A,X) | holds(i(A,B),X).
2 [] -holds(B,X) | holds(i(A,B),X).
3 [] r(X,box(A,X)) | holds(b(A),X).
4 [] -holds(A,box(A,X)) | holds(b(A),X).
5 [] -holds(a(A,B),X) | holds(A,X).
6 [] -holds(a(A,B),X) | holds(B,X).
7 [] r(X,Y) | (X = Y).
8 [] -holds(b(A),X) | -r(X,Z) | holds(A,Z).
end_of_list.

list(sos).
9 [] -holds(i(a(b(p),p),b(b(p))),s).
end_of_list.

---------------- PROOF ----------------

12 [ur,9,2] -holds(b(b(p)),s).
13 [ur,9,1] holds(a(b(p),p),s).
17 [ur,12,4] -holds(b(p),box(b(p),s)).
20 [ur,13,6] holds(p,s).
21 [ur,13,5] holds(b(p),s).
37 [ur,17,4] -holds(p,box(p,box(b(p),s))).
45 [ur,37,8,21] -r(s,box(p,box(b(p),s))).
49 [ur,45,7] (box(p,box(b(p),s)) = s).
52 [para_from,49,37] -holds(p,s).
53 [binary,52,20] .
```

Figure 7: An automated proof using Otter

[Bor93] T. Borghuis. Interpreting modal natural deduction in type theory. In Maarten de Rijke, editor, *Diamonds and Defaults*, pages 67–102. Kluwer Academic Publishers, 1993.

[BR97a] K. Broda and A. Russo. A compilation style LDS in a natural deduction framework. Technical report, Imperial College, Department of Computing, 1997. Available by ftp from ftp.theory.papers.Broda.SLDS.ps.Z.

[BR97b] Krysia Broda and Alessandra Russo. Theorem proving in LDS - a compilation approach. Proceedings of Workshop on Automated Reasoning, Manchester, 1997.

[Bro91] K. Broda. *The Application of Semantic Tableaux with Unification to Autmated Deduction*. PhD thesis, Department of Computing. Imperial College of Science, Technology and Medicine., 1991.

[CL73] C. Chang and R. Lee. *Symbolic Logic and Mechanical Theorem Proving*. Academic Press, 1973.

[Dem96] S. Demri. A simple tableaux system for the logic of elsewhere. In *LNAI 1071, Tableaux 96, Palermo, Italy*, pages 177–192, 1996.

[DG94] M. D'Agostino and D. Gabbay. A generalization of analytic deduction via labelled deductive systems.Part I: Basic substructural logics. *Journal of Automated Reasoning*, 13:243–281, 1994.

[Dôs93] K. Dôsen. A historical introduction to substructural logics. In P. Schroeder Heister and Kosta Dôsen, editors, *Substructural Logics*, pages 1–31. Oxford University Press, 1993.

[dR92] M. de Rijke. The modal logic of inequality. *Journal of Symbolic Logic*, 57(2), 1992.

[Fit83] M. Fitting. *Proof methods for modal and intuitionistic logics*. Dordrecht, Reidel, 1983.

[FS91] A. M. Frisch and R. B. Scherl. A general framework for modal deduction. In *2nd Conference on Principles of Knoledge Representation and Reasoning*. Morgan-Kaufmann, 1991.

[Gab92] D. Gabbay. How to Construct a Logic for Your Application. In *Proceedings of German AI Conference, GWAI'92, Lecture Notes on AI*, volume 671, pages 1–30. Springer, 1992.

[Gab96] D. Gabbay. *Labelled Deductive Systems, Volume 1 - Foundations*. Oxford University Press, 1996.

[Gen93] I. Gent. Theory matrices (for modal logics) using alphabetical monotonicity. *Studia Logica*, 52(2):233–257, 1993.

[Gol90] D. Goldson. *Techniques for Theorem Proving and Proof Transformation*. PhD thesis, QMW, University of London, 1990.

[HC68] G. Hughes and M. Cresswell. *An Introduction to Modal Logics*. Methuen, London, 1968.

[Mas94] F. Massacci. Strong analytic tableaux for normal modal logics. In *Proceedings of CADE-12, Lecture Note in Artificial Intelligence 814*. Springer Verlag, 1994.

[McC90] W. McCune. Otter 2.0 user guide. tech report ANL -90/9. Technical report, Argonne National Laboratory, Argonne, IL, 1990.

[Ohl91] H.J. Ohlbach. Semantics–based translation methods for modal logics. *Journal of Logic and Computation*, 1(5):691–746, 1991.

[Rus96] A. Russo. *Modal Logics as Labelled Deductive Systems*. PhD thesis, Department of Computing. Imperial College of Science, Technology and Medicine., 1996.

[Seg81] K. Segerberg. A note on the logic of elsewhere. *Theoria*, 47:183–187, 1981.

[Sti85] M. Stickel. Automated Deduction by Theory Resolution. *JAR*, 1(4), 1985.

[Sym93] A. Sympson. *The Proof Theory and Semantics of Intuitionistic Modal Logics*. PhD thesis, University of Edinburgh, 1993.

[vB83] J. van Benthem. Correspondence theory. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic - Extensions of Classical Logics*, volume II. D. Reidel Publishing Company, 1983.