# LOGIC JOURNAL of the IGPL

Volume 10 Number 2 March 2002

Interest Group in Pure and Applied Logics



**OXFORD UNIVERSITY PRESS**

ISSN 1367-0751

# Subscription Information

Volume 10, 2002 (bimonthly) Full: Europe pounds sterling 275; Rest of World US$ 450. Personal: pounds sterling 138 (US$ 225). Please note that personal rates apply only when copies are sent to a private address and payment is made by a personal cheque or credit card.

## Order Information

Subscriptions can be accepted for complete volumes only. Prices include air-speeded delivery to Australia, Canada, India, Japan, New Zealand, and the USA. Delivery elsewhere is by surface post. Payment is required with all orders and may be made in the following ways:

Cheque (made payable to Oxford University Press)
National Girobank (account 500 1056)
Credit card (Access, Visa, American Express)
UNESCO Coupons
Bankers: Barclays Bank plc, PO Box 333, Oxford, UK. Code 20-65-18, Account 00715654.

Requests for sample copies, subscription enquiries, orders and changes of address should be sent to the Journals Subscriptions Department, Oxford University Press, Great Clarendon Street, Oxford OX2 6DP, UK. Tel: +44 (0) 1865 267907. Fax: +44 (0) 1865 267485. Email: jnl.orders@oup.co.uk

## Advertisements



## Copyright



*Logic Journal of the IGPL* (ISSN 1367-0751) is published bimonthly in January, March, May, July, September and November by Oxford University Press, Oxford, UK. Annual subscription price is US$ 450.00. *Logic Journal of the IGPL* is distributed by M.A.I.L. America, 2323 Randolph Avenue, Avenel, NJ 07001. Periodical postage paid at Rahway, New Jersey, USA and at additional entry points.
US Postmasters: Send address changes to *Logic Journal of the IGPL*, c/o Mercury International, 365 Blair Road, Avenel, NJ 07001, USA.

## Back Issues

The current plus two back volumes are available from Oxford University Press. Previous volumes can be obtained from the Periodicals Service Company, 11 Main Street, Germantown, NY 12526 USA. Tel: +1 (518) 537 4700, Fax: +1 (518) 537 5899.

# Logic Journal of the IGPL

Volume 10, Number 2, March 2002


## Contents




Please visit the journal's World Wide Web site at
http://www.jigpal.oupjournals.org

# Logic Journal of the Interest Group in Pure and Applied Logics



## Scope of the Journal

The *Journal* is the official publication of the International Interest Group in Pure and Applied Logics (IGPL), which is sponsored by The European Foundation for Logic, Language and Information (FoLLI), and currently has a membership of over a thousand researchers in various aspects of logic (symbolic, computational, mathematical, philosophical, etc.) from all over the world.

The *Journal* is published in hardcopy and in electronic form six times per year. Publication is fully electronic: submission, refereeing, revising, typesetting, checking proofs, and publishing, all is done via electronic mailing and electronic publishing.

Papers are invited in all areas of pure and applied logic, including: pure logical systems, proof theory, model theory, recursion theory, type theory, nonclassical logics, nonmonotonic logic, numerical and uncertainty reasoning, logic and AI, foundations of logic programming, logic and computation, logic and language, and logic engineering.

The *Journal* is an attempt to solve a problem in the logic (in particular, IGPL) community:

- Long delays and large backlogs in publication of papers in current journals.
- Very tight time and page number limits on submission.

Papers in the final form should be in LaTeX. The review process is quick, and is made mainly by other IGPL members.

### Submissions

Submissions are made by sending a submission letter to the e-mail address: jigpl@dcs.kcl.ac.uk, giving the title and the abstract of the paper, and informing: of how to obtain the file electronically or, by sending 5 (five) hardcopies of the paper to the Editor-in-Chief.

**URL:** http://www.jigpal.oupjournals.org

# On the Descriptive Complexity of a Simplified Game of Hex

ARGIMIRO ARRATIA, *Departamento de Matemáticas, Universidad Simón Bolívar, Apartado 89000, Caracas 1080-A, Venezuela. E-mail: arratia@ma.usb.ve*

## Abstract

The game Whex is here defined, which is similar to Generalized Hex but the players are restricted to colour vertices adjacent to the vertex last coloured by one of the players. It is shown that the problem of deciding existence of winning strategies for one of the players in this game is complete for **PSPACE**, via quantifier free projections, and that the extension of first order logic with the corresponding generalized quantifier captures **PSPACE** and verifies a normal form. This problem is used to show that the problem of finding a proof in a proof system, like propositional resolution, in which the user is allowed to introduce auxiliary statements in order to help the system reach the theorem that he had set it to prove, is also complete for **PSPACE** via quantifier free projections. Also, it is established the complexity of the game Whex when restricted to graphs of outdegree at most 3, and, as a generalized quantifier, its expressive capabilities in the absence of ordering relation.

*Keywords*: Descriptive complexity; generalized quantifier; quantifier free projection; games; Generalized Hex; polynomial space.

## 1 Introduction

In Descriptive Complexity we are concerned with logical characterizations of the usual Turing machine based notion of a computational complexity class. Among the advantages of the logical approach to Computational Complexity we have: we can establish the complexity of a problem by syntactic measures, like number of quantifiers, variables and other symbols needed for describing the problem, instead of, say, designing an algorithm for it; we can define a notion of reduction among problems (the first order reduction) which is weaker than the traditional log–space; and we can sometimes discern the nature of problems (on logical grounds) that within the traditional Turing machine framework are equivalent (a fact which, often, defies our intuition).

There are various ways of constructing a logic that define exactly the properties that lie within a certain computational complexity class. The method used in this paper consists of extending the expressive power of first order logic with a uniform sequence of generalized quantifiers, corresponding to a problem which is a representative of the complexity class, in the sense of being complete for the class via logspace reducibility. It is also necessary to include some built–in ordering relation, like that of successor on the natural numbers, plus two constants that stand for the first and last element in the ordering. This is the same as saying that our logical interpretations of our complexity classes hold only over ordered finite structures with at least two elements, which seems to be a necessary restriction, at least for logics expressing polynomial time properties and of lesser complexity. The reader can find in the books [5] and [10] further details on generalized quantifiers, and an extensive discussion on the problem of the ordering in the logical characterizations of computational complexity classes.

The computational problem presented in this paper is a variation of the game of Generalized Hex (see [7]), that I have named Whex for reasons to be explained later, which illustrates the facts mentioned in the first paragraph of this introduction. In particular this new game seems easier to play in practice but, as it will be shown, enjoys similar complexity features as Hex in theory.

As an application, the new game problem Whex is used to show that the problem of finding a proof in a (mechanical) proof system with only one rule of inference like, say, propositional resolution, and in which the user is allowed to introduce auxiliary statements in order to help the system reach the theorem that we had set it to prove, is complete for the class of problems that have polynomially space bounded algorithmic solutions, via a very weak kind of reductions, namely *quantifier free projections*. On the way, here I also exhibit some tools for proving problems complete via reductions which are definable in first order logic, and illustrate their use with the particular game problem in consideration. This paper ends exhibiting the complexity of Whex, both computational and logical, when the board is restricted to be directed graphs of outdegree $\leq 3$, and further, it is pointed out the expressive capabilities of Whex, as a generalized quantifier, in the absence of order.

## 2 Background on Descriptive Complexity

Some needed notions from Descriptive Complexity are reviewed in this section. More details can be found in the books [5], [10]. A vocabulary $\tau = \{R_1, \ldots, R_r, C_1, \ldots, C_s\}$ is a finite set of relation and constant symbols, where each relation $R_i$ has arity $n_i$. A finite $\tau$–structure $\mathcal{A} = \langle A, R_1^{\mathcal{A}}, \ldots, R_r^{\mathcal{A}}, C_1^{\mathcal{A}}, \ldots, C_s^{\mathcal{A}} \rangle$ consists of a universe $A = \{0, \ldots, n-1\}$, relations $R_i^{\mathcal{A}} \subseteq A^{n_i}$, and constants $C_j^{\mathcal{A}}$ $(1 \leq i \leq r,\ 1 \leq j \leq s)$. STRUCT($\tau$) denotes the set of all finite $\tau$–structures. A problem over vocabulary $\tau$ is a subset of STRUCT($\tau$) closed under isomorphisms.

A logic $\mathcal{L}$ consists of: 1) all sets $\mathcal{L}(\tau)$ of formulas over each finite vocabulary $\tau$, build up from the symbols in $\tau$, variables, boolean operations $\wedge$, $\vee$, $\neg$, and quantifiers $\forall$, $\exists$, following certain syntactic rules, together with a satisfaction relation $\models$ which establishes the meaning of formulas in (finite) models (possibly extended with interpretations for free variables). In particular FO denotes First Order logic, and $\mathrm{FO}_s$ is First Order logic with built–in symbols $suc(\cdot,\cdot)$, 0 and $max$, which are always interpreted on finite structures as the successor relation over the naturals, the first and the last element. Hence, a sentence in $\mathrm{FO}_s$ describes a property of *ordered structures*.

If $\mathcal{L}$ is a logic and $\phi$ a sentence in $\mathcal{L}(\tau)$, for some vocabulary $\tau$, then $MOD(\phi) := \{\mathcal{A} \in \mathrm{STRUCT}(\tau) : \mathcal{A} \models \phi\}$ denotes the set of finite models that satisfy $\phi$.

Problems as traditionally considered in Computational Complexity are sets of strings, usually of 0's and 1's, so that they can be inputted into Turing machines. To transform finite structures into strings of $\{0,1\}^*$, and viceversa, some forms of encodings are established, and, furthermore, these encodings can be made very efficiently (e.g., first order definable, see [10]). Hence, if $\tau$ is some vocabulary and $\mathcal{A} \in \mathrm{STRUCT}(\tau)$, $e_\tau(\mathcal{A})$ denotes an encoding of $\mathcal{A}$ as a string over the set of symbols $\{0,1\}$, which is first order definable. If $\Omega$ is some problem over $\tau$ (in the sense of being a set of finite $\tau$-structures closed under isomorphism), then the set of strings over $\{0,1\}$ corresponding to encodings of elements of $\Omega$ is denoted $e_\tau(\Omega) := \{e_\tau(\mathcal{A}) : \mathcal{A} \in \Omega\} \subseteq \{0,1\}^*$.

Thus, given a complexity class $\mathcal{C}$, defined in terms of the sets of strings over $\{0,1\}$ accepted by some kind of Turing machines, when it is said that $\mathcal{C}$ *is captured by a logic* $\mathcal{L}$ it is meant formally that

1. for each set of strings $S \in \mathcal{C}$, there is some vocabulary $\tau$ and a sentence $\phi$ in $\mathcal{L}(\tau)$, such that $S = e_\tau(MOD(\phi))$; and
2. for each vocabulary $\tau$ and each sentence $\phi$ in $\mathcal{L}(\tau)$, $e_\tau(MOD(\phi)) \in \mathcal{C}$.

Whenever class $\mathcal{C}$ is captured by logic $\mathcal{L}$ (or $\mathcal{L}$ captures $\mathcal{C}$), it will be denoted $\mathcal{C} = \mathcal{L}$. Moreover, $\mathcal{C} \le \mathcal{L}$ denotes that item 1. above is satisfied, and $\mathcal{L} \le \mathcal{C}$ denotes that item 2. is satisfied. So, $\mathcal{C} = \mathcal{L}$ if and only if $\mathcal{C} \le \mathcal{L}$ and $\mathcal{L} \le \mathcal{C}$. If $\mathcal{L}_1$ and $\mathcal{L}_2$ are two logics then $\mathcal{L}_1 = \mathcal{L}_2$ means that, for all vocabulary $\tau$, every $\mathcal{L}_1(\tau)$-sentence $\phi_1$ is equivalent to an $\mathcal{L}_2(\tau)$-sentence $\phi_2$ and viceversa. (Here, equivalence of sentences $\phi_1$ and $\phi_2$ has the following intended meaning: for every finite $\tau$-structure $\mathcal{A}$, $\mathcal{A} \models \phi_1$ if and only if $\mathcal{A} \models \phi_2$, this is commonly denoted as $\models \phi_1 \longleftrightarrow \phi_2$. )

An important notion in Descriptive Complexity is that of a logical reduction.

**Definition 2.1** Let $\mathcal{L}$ be some logic, let $\tau$ be a vocabulary and $\sigma = \{R_1, \dots, R_r, C_1, \dots, C_s\}$ be another vocabulary, where each $R_i$ is a relation symbol of arity $n_i$ and each $C_j$ is a constant symbol. A set $\Sigma$ of $\mathcal{L}(\tau)$–formulas of the form

$$\Sigma := \{\phi_1(\overline{x}_1), \dots, \phi_r(\overline{x}_r), \psi_1(\overline{y}_1), \dots, \psi_s(\overline{y}_s)\},$$

where $\overline{x}_i$ and $\overline{y}_j$ are vectors of distinct variables with $|\overline{x}_i| = kn_i$ and $|\overline{y}_j| = k$, for some positive integer $k$, $1 \le i \le r$ and $1 \le j \le s$, describes a $(\tau, \sigma)$*–translation of arity* $k$, which is a map sending a $\tau$–structure $\mathcal{A}$ into a $\sigma$–structure $\mathcal{A}_\Sigma$, such that

- the universe of $\mathcal{A}_\Sigma$ is the set $A^k$ of $k$-tuples of $A$;
- for $i = 1, \dots, r$, the relation $R_i$ of $\sigma$ has the interpretation:

$$R_i^{\mathcal{A}_\Sigma} := \{\overline{a} \in A^{kn_i} : \langle \mathcal{A}, \overline{a} \rangle \models \phi_i(\overline{x}_i)\}$$

- for $j = 1, \dots, s$, the constant $C_j$ of $\sigma$ has the interpretation:

$$C_j^{\mathcal{A}_\Sigma} := \overline{u}_j$$

  where $\overline{u}_j$ is the unique element in $A^k$ that satisfies $\psi_j$.

A problem $\Omega_1 \subseteq \text{STRUCT}(\tau)$ $\mathcal{L}$*-reduces to* $\Omega_2 \subseteq \text{STRUCT}(\sigma)$ (denoted $\Omega_1 \le_{\mathcal{L}} \Omega_2$) if there exists a $k > 0$ and a set $\Sigma$ of $\mathcal{L}(\tau)$–formulas which defines a $(\tau, \sigma)$–translation of arity $k$ so that, for all $\mathcal{A} \in \text{STRUCT}(\tau)$,

$$\mathcal{A} \in \Omega_1 \text{ if and only if } \mathcal{A}_\Sigma \in \Omega_2.$$

In particular, if $\mathcal{L}$ in the above definition is FO, then we have a *first order reduction.* If, further, all the formulas in $\Sigma$ are *projections*, then we have a reduction which is a *first order projection (fop).* (A first order formula $\psi$, over some vocabulary $\tau$, is a *projection* if it has the form

$$\alpha_0 \vee (\alpha_1 \wedge \beta_1) \vee \ldots \vee (\alpha_m \wedge \beta_m)$$

where none of the $\alpha_i$ contains a symbol from $\tau$; for $i \neq j$, $\alpha_i$ and $\alpha_j$ are mutually exclusive; and each $\beta_i$ is an atomic or negated atomic formula built up from symbols in $\tau$ only.)

## 3 Capturing complexity classes with generalized quantifiers

We can turn a problem into a generalized quantifier and increase the expressive power of first order logic as follows. Let $\sigma$ be as in the Definition 2.1 and let $\Omega$ be a problem over $\sigma$. Then the extension of FO with the generalized quantifier $\Omega$, which will be denoted $\Omega^*$[FO], is the smallest set $\mathcal{L}$ of formulas such that: $\mathcal{L}$ contains all first order formulas, $\mathcal{L}$ is closed under all logical connectives and first order quantifiers, and for any finite vocabulary $\tau$, if $\Sigma := \{\phi_1, \ldots, \phi_r, \psi_1, \ldots, \psi_s\}$ is a set of $\tau$–formulas in $\mathcal{L}$ that describes a $(\tau, \sigma)$–translation of arity $k$, mapping $\tau$–structure $\mathcal{A}$ into $\sigma$–structure $\mathcal{A}_\Sigma$, then

$$\Phi := \Omega[\overline{x}_1, \ldots, \overline{x}_r, \overline{y}_1, \ldots, \overline{y}_s : \phi_1(\overline{x}_1), \ldots, \phi_r(\overline{x}_r), \psi_1(\overline{y}_1), \ldots, \psi_s(\overline{y}_s)]$$

is a new sentence in $\mathcal{L}$, which is interpreted as follows. Given a $\tau$–structure $\mathcal{A}$, $\mathcal{A} \models \Phi$ if, and only if, the $(\tau, \sigma)$–translation of $\mathcal{A}$ described by $\Sigma$, namely, the $\sigma$–structure $\mathcal{A}_\Sigma$, is such that $\mathcal{A}_\Sigma \in \Omega$.

**Remark 3.1** The formula

$$\Omega[\overline{x}_1, \ldots, \overline{x}_r, y_1, \ldots, y_s : \phi_1(\overline{x}_1), \ldots, \phi_r(\overline{x}_r), y_1 = C_1, \ldots, y_s = C_s]$$

is often abbreviated as

$$\Omega[\overline{x}_1, \ldots, \overline{x}_r : \phi_1(\overline{x}_1), \ldots, \phi_r(\overline{x}_r)](C_1, \ldots, C_s).$$

Some interesting fragments of $\Omega^*$[FO] are:

$\Omega^n$[**FO**] for a positive integer $n$, is such that at most $n$ nested applications of $\Omega$ can appear in a formula.

**pos**$\Omega^*$[**FO**] is the sublogic where no application of $\Omega$ is within the scope of a $\neg$ (i.e., $\Omega$ has only positive occurrences).

**pos**$\Omega^n$[**FO**] for a positive integer $n$ is the language with all occurrences of $\Omega$ positive and at most $n$ nested

$\Omega^*$[**FO**$_s$] is the fragment of $\Omega^*$[FO] with the built–in successor and built–in constants 0 and $max$. In a similar way are defined the fragments $\Omega^n$[FO$_s$], pos$\Omega^*$[FO$_s$] and pos$\Omega^n$[FO$_s$].

**Example 3.2** Consider the vocabulary $\tau_2 = \{E, C, D\}$, where $E$ is a binary relation symbol, $C$ and $D$ are constant symbols, and consider the following two problems over $\tau_2$:

$$\begin{aligned} \text{TC} &:= \{\mathcal{A} \in \text{STRUCT}(\tau_2) : \mathcal{A} \text{ is a directed graph} \\ &\quad\ \text{and there is a path from vertex } C^{\mathcal{A}} \text{ to vertex } D^{\mathcal{A}}\}. \end{aligned}$$

$$\begin{aligned} \text{DTC} &:= \{\mathcal{A} \in \text{STRUCT}(\tau_2) : \mathcal{A} \text{ is a directed graph and there is a} \\ &\quad\ \text{path from } C^{\mathcal{A}} \text{ to } D^{\mathcal{A}} \text{ of all vertices with outdegree 1}\}. \end{aligned}$$

(Recall that the outdegree of a vertex $v$ in a directed graph is the number of edges going out of $v$.)

As it was shown by Neil Immerman in [9], the positive fragment of the extensions of $\mathrm{FO}_s$ with generalized quantifiers corresponding to each one of these problems captures, respectively, the classes nondeterministic logspace (**NL**) and logspace (**L**). Specifically:

$$\mathrm{posTC}^*[\mathrm{FO}_s] = \mathbf{NL} \text{ and } \mathrm{posDTC}^*[\mathrm{FO}_s] = \mathrm{DTC}^*[\mathrm{FO}_s] = \mathbf{L}$$

In general, if $\mathcal{L}$ is any *regular* logic (as defined in [4]), define the extension of $\mathcal{L}$ with the generalized quantifier $\Omega$ in a similar manner as for FO, and obtain the logic $\Omega^*[\mathcal{L}]$ and similar fragments. The next proposition is immediate from the definitions, and it just makes explicit the fact that $\mathcal{L}$–reducibility among problems is equivalent to definability in the extension of $\mathcal{L}$ with the generalized quantifier corresponding to the larger problem.

**Proposition 3.3** Let $\mathcal{L}$ be a regular logic, $\sigma$ and $\tau$ be two vocabularies, with $\tau = \{R_1, \dots, R_r, C_1, \dots, C_c\}$. Let $\Omega_1$ be a $\sigma$–problem and $\Omega_2$ be a $\tau$–problem. Then $\Omega_1 \leq_{\mathcal{L}} \Omega_2$ via a $(\sigma, \tau)$–translation of arity $k$ if, and only if, $\Omega_1 = MOD(\Phi)$ for some sentence $\Phi \in \mathrm{pos}\Omega_2^1[\mathcal{L}(\sigma)]$ of the form $\Omega_2[\overline{x}_1, \dots, \overline{x}_r, \overline{y}_1, \dots, \overline{y}_c : \phi_1, \dots, \phi_r, \psi_1, \dots, \psi_c]$, where $\{\phi_1, \dots, \phi_r, \psi_1, \dots, \psi_c\} \subset \mathcal{L}(\sigma)$ constitute a $(\sigma, \tau)$–translation of arity $k$. ∎

The following important result that links the usual Turing machine based reducibility and the logical reducibility has been proved in [12] (cf. [5, Proposition 10.3.22]).

**Proposition 3.4** Let $\Omega_1 \subseteq \mathrm{STRUCT}(\sigma)$ and $\Omega_2 \subseteq \mathrm{STRUCT}(\tau)$ be problems. Then $e_\sigma(\Omega_1)$ is logspace reducible to $e_\tau(\Omega_2)$ if, and only if, $\Omega_1 \leq_{\mathrm{DTC}^1[\mathrm{FO}_s]} \Omega_2$. ∎

As a consequence of the previous propositions and definitions we have the following *sandwich theorem*, which is a key tool for "almost capturing" complexity classes by extensions of $\mathrm{FO}_s$ with generalized quantifiers. (This result is a generalization of Corollary 3.1 of [12], and it is not difficult to prove.)

**Theorem 3.5** Let $\mathcal{C}$ be a complexity class above **L** and closed under logspace reducibility. Let $\Omega \subseteq \mathrm{STRUCT}(\tau)$ be a problem such that $\mathrm{DTC}^1[\mathrm{FO}_s] \leq \mathrm{pos}\Omega^*[\mathrm{FO}_s]$, and $e_\tau(\Omega)$ is complete in $\mathcal{C}$ via logspace reducibility. Then,

$$\mathrm{pos}\Omega^1[\mathrm{FO}_s] \leq \mathcal{C} \leq \mathrm{pos}\Omega^*[\mathrm{FO}_s]$$

∎

The necessary next step for having an exact description of $\mathcal{C}$ by the logic $\mathrm{pos}\Omega^*[\mathrm{FO}_s]$ is to show that it has a *first order normal form*; that is, every sentence in $\mathrm{pos}\Omega^*[\mathrm{FO}_s]$ is equivalent to one application of the quantifier $\Omega$ to a first order sentence $\psi$. In symbols, that $\mathrm{pos}\Omega^*[\mathrm{FO}_s] = \mathrm{pos}\Omega^1[\mathrm{FO}_s]$ We sometimes get something much better; namely, that the first order sentence $\psi$ is a *quantifier free projection* (qfp), and so it is said that the logic has a *quantifier free projective normal form*.

When there is a normal form then the corresponding problem $\Omega$ is complete for the class via either first order or qfp reductions, according to the nature of the normal form. Either of these reducibilities is computationally much weaker than logspace, since the set of all problems definable by first order sentences is properly contained in the class **L**. As an example, the two logics just mentioned, namely, $\mathrm{posTC}^*[\mathrm{FO}_s]$ and $\mathrm{DTC}^*[\mathrm{FO}_s]$, verify a quantifier free projective normal form and, hence, TC and DTC are complete via qfp reductions in their respective classes (see [9] for the details).

## 4 The WHEX logic

Fix a vocabulary $\tau_2 = \{E, C, D\}$, where $E$ is a binary relation symbol, $C$ and $D$ are constant symbols. A complexity class and a problem that fits in the hypothesis of Theorem 3.5 is **PSPACE** (polynomial space) and the $\tau_2$–problem HEX. An instance of HEX is a graph $G$ with a source $s$ and a sink $t$, and a yes-instance is an instance $(G, s, t)$ where Player 1 has a winning strategy in the game of Hex on $(G, s, t)$. The game of Hex proceeds as follows. Beginning with Player 1, two players take turns in colouring previously uncoloured vertices of $G$, apart from $s$ and $t$, until all vertices are coloured. Player 1 always colours a vertex red and Player 2 always colours a vertex blue. Player 1 wins the game of Hex if in the resulting coloured graph there is a path consisting entirely of red vertices from $s$ to $t$. That the encoding of this problem as strings of 0's and 1's is complete for **PSPACE** via logspace reducibility was shown in [6]; and that the extension of first-order logic using a uniform sequence of generalized quantifiers corresponding to the problem HEX contains DTC–logic (with respect to arbitrary finite structures) is proved in the following theorem.

**Theorem 4.1** $\text{DTC}^1[\text{FO}] \leq \text{posHEX}^*[\text{FO}]$

PROOF. Consider the $\tau_2$–sentence $\psi := \text{DTC}[\overline{x}, \overline{y} : \theta(\overline{x}, \overline{y})](\overline{C}, \overline{D})$, where $\overline{x}$ and $\overline{y}$ are $k$-tuples of distinct variables, for some $k > 0$, $\overline{C}$ and $\overline{D}$ are $k$–tuples with all entries equal to $C$ and to $D$ respectively, and $\theta \in \text{FO}$. Consider the following formula $\phi$ with free variables $\overline{x}$, $u_1$, $u_2$, $\overline{y}$, $v_1$ and $v_2$:

$$\begin{array}{lll}\phi & := & (\overline{x} = \overline{C} \wedge u_1 = u_2 = C \wedge \overline{y} = \overline{C} \\ & & \wedge\ ((v_1 = D \wedge v_2 = C) \vee (v_1 = C \wedge v_2 = D))) \\ & \vee & (\overline{x} = \overline{D} \wedge ((u_1 = D \wedge u_2 = C) \vee (u_1 = C \wedge u_2 = D)) \\ & & \wedge\ \overline{y} = \overline{D} \wedge v_1 = D \wedge v_2 = D) \\ & \vee & (u_1 = v_1 \wedge u_2 = v_2 \wedge \theta(\overline{x}, \overline{y})) \\ & \vee & (u_1 = C \wedge u_2 = D \wedge v_1 = D \wedge v_2 = C \wedge \theta(\overline{x}, \overline{y})) \\ & \vee & (u_1 = D \wedge u_2 = C \wedge v_1 = C \wedge v_2 = D \wedge \theta(\overline{x}, \overline{y}))\end{array}$$

Then

$$\models \psi \ \longleftrightarrow\ \text{HEX}[(\overline{x}, u_1, u_2), (\overline{y}, v_1, v_2) : \phi](\overline{C}, \overline{D}).$$

(Player 1's winning strategy consists of colouring first $(\overline{C}, C, D)$ or $(\overline{C}, D, C)$, and thereafter colour opposite vertex to Player 2's choice; that is, if Player 2 colours $(\overline{x}, u_1, u_2)$ then Player 1 colours $(\overline{x}, u_2, u_1)$.) ∎

In [2] it was shown that the HEX–logic has a quantifier free projective normal form and, hence, captures **PSPACE** applying Theorems 4.1 and 3.5. As an extra bonus we get that the problem HEX is complete for **PSPACE** via quantifier free projections.

I would like to remark that similar results hold for the version of the Hex game where players colour *edges* (as opposed to vertices) of a *directed* graph. The version of the Hex game where players colour edges of an undirected graph is known as the *Shannon switching game* for which polynomial time algorithms are known (see, for example, [15]).

The main concern of this paper is the following variation of the problem HEX. An instance is a graph $G$ with a source $s$ and a sink $t$, and a yes–instance is an instance where Player 1 has a winning strategy in a game which proceeds as the game of Hex but with the following restriction:

> Players can not colour an arbitrary vertex but must proceed as follows: Player 1 begins the game and he must do it by colouring red a vertex adjacent to the source. From this move and on, Player 2 must colour blue an uncoloured vertex adjacent to the vertex last coloured red (i.e. coloured by Player 1), and Player 1 replies by colouring red an uncoloured vertex adjacent to the vertex that he coloured red last.

Thus, Player 1 tries to build a path in a step–by–step fashion and linked to the source, whilst Player 2 tries continually to block Player 1's construction. As in the Hex game, Player 1 wins if he reaches $t$ with a path of red vertices only from $s$, with $s$ and $t$ uncoloured and possibly some other vertices at the end of the game.

The above restriction placed upon the game of Hex makes the construction process of a path local and reduces the space of search for each player (in fact, it makes it linear); for these reasons I named this game *Weak Hex* (abbreviated Whex) and the corresponding decision problem WHEX. However I shall prove below that WHEX is similar to HEX with regards to computational complexity, when both are treated as decision problems, and, on the other hand, they also coincide in expressive power when treated as generalized quantifiers. So, in those two aspects, WHEX is no weaker than HEX.

WHEX is encoded as a class of $\tau_2$-structures as follows:

$$\begin{aligned}\text{WHEX} = \{\mathcal{A} = \langle A, E^{\mathcal{A}}, C^{\mathcal{A}}, D^{\mathcal{A}}\rangle \in \text{STRUCT}(\tau_2) :\ &\text{Player 1 has} \\ &\text{a winning strategy for the game of Whex played on } \mathcal{A}\ \}\end{aligned}$$

It is worth observing that, as classes of structures, WHEX $\neq$ HEX: There are graphs where Player 1 wins the game of Whex but can loose the game of Hex (for example, a graph as constructed in Theorem 4.3 below, corresponding to a satisfiable sentence), and viceversa (for example, the complete bipartite graph $K_{n,n}$, with $n \geq 4$ and where the top two vertices are joined to a vertex $s$ and the bottom two vertices are joined to a vertex $t$, is a yes–instance of HEX but not of WHEX). Furthermore, there are graphs where Player 1 wins both the Hex and the Whex games, as, for example, a ladder with diagonal rungs with the source joined at one end and the sink joined at the other end. One such a graph is constructed in the proof of Theorem 4.1, and, so, we have

**Theorem 4.2** $\text{DTC}^1[\text{FO}] \leq \text{posWHEX}^*[\text{FO}]$ ∎

## 4.1 *WHEX is* **PSPACE**-*complete*

QSAT, or the problem of determining if a quantified boolean formula in conjunctive normal form is true, is the classical example of a **PSPACE**–complete problem [7] and can be regarded as a game as follows (see [11] also): given

$$\Phi := \exists x_1 \forall x_2 \dots Q_{n-1} x_{n-1} Q_n x_n \, \phi$$

where $\phi$ is a boolean formula in conjunctive normal form involving the variables $x_1$, $\ldots$, $x_n$, and the quantifiers $Q_i \in \{\forall, \exists\}$ alternate starting with $\exists$ [1]. We have two players, Player 1 and Player 2, who take turns assigning a value of 0 (false) or 1 (true) to each variable, beginning with $x_1$ and with Player 1 making the first move. So, Player 1 assigns truth values to the variables existentially quantified, whilst Player 2 assigns truth values to the variables universally quantified; for those reasons Player 1 is also known as the $\exists$ player and Player 2 as the $\forall$ player. This game is named Qsat (played on $\Phi$), and one say that Player 1 (or $\exists$) wins the game of Qsat on $\Phi$ if, and only if, after the $n$-th move $\phi$ is true.

To show that WHEX is **PSPACE**–complete I will describe below how to construct, using logarithmic space, an instance, $(G, s, t)$, of WHEX from an instance, $\Phi$, of QSAT.

**Theorem 4.3** QSAT $\leq_{\log}$ WHEX.

Proof. Let $\Phi := \exists x_1 \forall x_2 \ldots Q_n x_n (C_1 \wedge C_2 \wedge \ldots \wedge C_m)$, where each clause $C_i$ is a conjunction of literals. Define the graph $G_\Phi = (V, E)$ as follows:

$$
\begin{aligned}
V \;&=\; \{s, t, y\} \cup \{x_i, \overline{x}_i, u_i, \overline{u}_i, v_i : 1 \leq i \leq n\} \\
&\cup\; \{w_{2i} : 1 \leq i \leq \lceil n/2 \rceil\} \cup \{c_i, z_i : 1 \leq i \leq m\}
\end{aligned}
$$

$$
\begin{aligned}
E \;&=\; \{(s, x_1), (s, \overline{x}_1), (v_n, c_1), (v_n, z_1), (z_m, t), (y, t)\} \\
&\cup\; \{(x_i, u_i), (\overline{x}_i, \overline{u}_i), (u_i, t), (\overline{u}_i, t), (x_i, v_i), (\overline{x}_i, v_i) : 1 \leq i \leq n\} \\
&\cup\; \{(v_i, x_{i+1}), (v_i, \overline{x}_{i+1}) : 1 \leq i \leq n-1\} \\
&\cup\; \{(v_{2i}, w_{2i}), (w_{2i}, t) : 1 \leq i \leq \lceil n/2 \rceil\} \\
&\cup\; \{(z_i, z_{i+1}), (z_i, c_{i+1}) : 1 \leq i \leq m-1\} \\
&\cup\; \{(c_i, y) : 1 \leq i \leq m\} \\
&\cup\; \{(c_i, u_j) : \text{ the literal } \neg x_j \text{ is in clause } C_i\} \\
&\cup\; \{(c_i, \overline{u}_j) : \text{ the literal } x_j \text{ is in clause } C_i\}
\end{aligned}
$$

(Figure 1 illustrates the graph $G_\Phi$ for

$$\Phi := \exists x_1 \forall x_2 \exists x_3 [(\neg x_1 \vee \neg x_2) \wedge (x_2 \vee x_3 \vee \neg x_1) \wedge (x_2 \vee \neg x_3)]\ )$$

First note that the moves of Player 1 and Player 2 on the graph $G_\Phi$ corresponds to moves by $\exists$ and $\forall$ on $\Phi$. The literal $\overline{x}_i$ stands for the negation of $x_i$; hence, Player 1 colouring $x_{2i-1}$ or $\overline{x}_{2i-1}$ red corresponds to $\exists$ deciding to give value 1 or 0 to $x_{2i-1}$. After Player 1's colouring, Player 2 is forced to colour vertex $u_{2i-1}$ or $\overline{u}_{2i-1}$; otherwise Player 1 reaches $t$ in his next move. After the forced colouring of $u_{2i-1}$ or $\overline{u}_{2i-1}$ by Player 2, Player 1's only possible choice is $v_{2i-1}$. Then it is Player 2's turn to decide whether to colour blue vertex $x_{2i}$ or $\overline{x}_{2i}$, which corresponds to $\forall$ making the decision of assigning value 1 or 0 to $x_{2i}$. According to the selection of Player 2, Player 1 must continue with colouring the opposite vertex and, again, Player 2 has a forced

[1] This is no loss of generality since we can always add clauses of the form $x \vee \neg x$ without altering the truth value of $\phi$

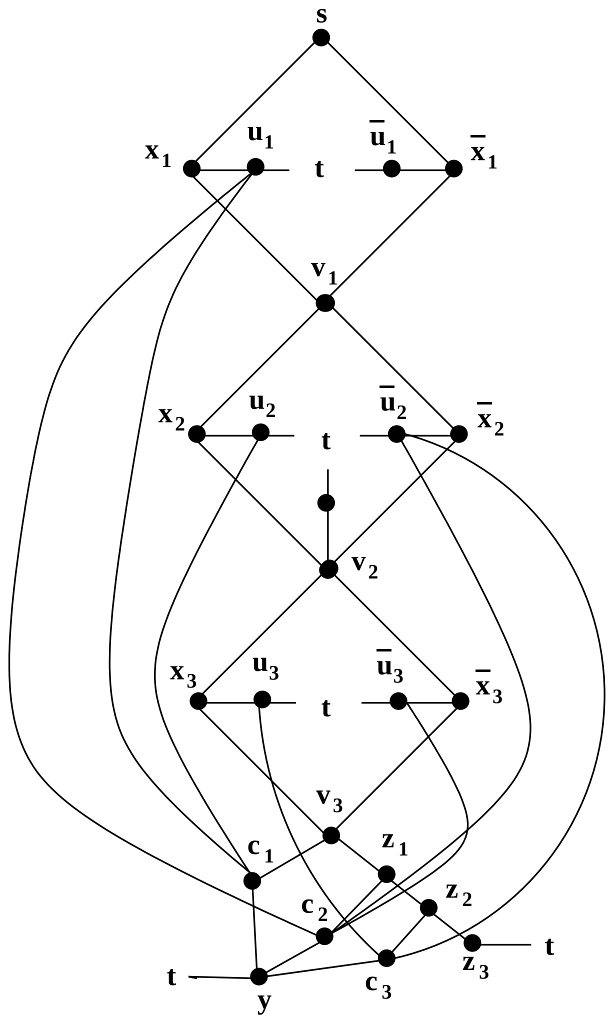


FIG. 1. $G_\Phi$ for $\Phi := \exists x_1 \forall x_2 \exists x_3 [(\neg x_1 \vee \neg x_2) \wedge (x_2 \vee x_3 \vee \neg x_1) \wedge (x_2 \vee \neg x_3)]$.

response, which obliges Player 1 to colour $v_{2i}$. From there, Player 1 either wins or, once again, must choose between $x_{2i+1}$ or $\overline{x}_{2i+1}$, and so on.

Now, suppose $\exists$ has a winning strategy in the game of Qsat on $\Phi$. Then Player 1 has the following winning strategy in the game of Whex on $G_\Phi$: if $\exists$ gives value 1 (respectively, 0) to variable $x_{2i-1}$ then Player 1 colours vertex $x_{2i-1}$ (resp. $\overline{x}_{2i-1}$) red. If, when Player 1 reaches vertex $v_n$, Player 2 subsequently colours all vertices $c_i$ blue, then Player 1 will be left with $z_m$ to colour and wins; otherwise, Player 1 will colour some $c_i$ red. Since all clauses are satisfiable, some literal in $C_i$ is true; therefore there is an edge from $c_i$ to some unvisited vertex $u_j$ or $\overline{u}_j$, besides the edge to $y$, so Player 1 colours the one left free by Player 2 and wins.

For the converse, suppose $\forall$ has a winning strategy in the game of Qsat on $\Phi$. Then

Player 2 wins the game of Whex in the following way: if $\forall$ gives value 1 (resp. 0) to variable $x_{2i}$, then Player 2 colours vertex $\overline{x}_{2i}$ (resp. $x_{2i}$) blue; this forces Player 1 to colour $x_{2i}$ (resp. $\overline{x}_{2i}$) red, and Player 2 to colour $u_{2i}$ (resp. $\overline{u}_{2i}$) blue. When Player 1 reaches $v_n$, there is one clause $C_i$ that is false, and, therefore, Player 2 forces Player 1 to colour vertex $c_i$ by colouring vertex $z_i$ blue. Then Player 2 colours $y$, thus succeeding in blocking Player 1, since all other edges lead to vertices $u_j$ already coloured by Player 2.

Finally it is easy to see that the construction of $G_\Phi$ from $\Phi$ can be done deterministically using logarithmic space. ∎

Next it will be shown that the logic posWHEX$^*$[FO$_s$] has a quantifier–free projective normal form.

**Theorem 4.4** Let $\tau$ be some vocabulary. Every sentence $\phi \in$ posWHEX$^*$[FO$_s(\tau)$] is equivalent to a sentence of the form WHEX[$\overline{x}$, $\overline{y}$ : $\psi$]($\overline{0}, \overline{max}$), where $\psi \in$ FO$_s(\tau)$, $\psi$ is quantifier–free projective and over the distinct $k$-tuples of variables $\overline{x}$ and $\overline{y}$, for some $k \geq 1$, and where $\overline{0}$ (resp. $\overline{max}$) is the constant symbol 0 (resp. $max$) repeated $k$-times.

Proof. (Sketch) The proof runs along the same lines as the proof of the normal form for posHEX$^*$[FO$_s$] in [2], which in turn is inspired by the proof of [9, Theorem 3.3]. I proceed by induction on the complexity of the sentence $\phi$, having to consider essentially five cases; the first one being $\phi$ an atomic or negated atomic sentence which is trivially equivalent to WHEX[$(x_1, y_1), (x_2, y_2) : \phi$]($(0, 0), (max, max)$) with $x_1$, $x_2$, $y_1$, and $y_2$ new variables not occurring in $\phi$.

For the remaining cases I adapted the constructions done in [9, Theorem 3.3], so as to fit in with the combinatorics of the Whex game, by incorporating the following gadget. Given an undirected graph $G = (V, E)$ with source $s$ and sink $t$ in $V$, built a new graph $X(G)$ consisting of a copy of $G$ with two sources, $s_1$ and $s_2$, in place of $s$, and two sinks, $t_1$ and $t_2$, in place of $t$. Draw edges between $s_1$ (respectively $s_2$) and each vertex that forms an edge with $s$ in $G$, and, similarly, draw edges between $t_1$ (respectively $t_2$) and each vertex that forms an edge with $t$ in $G$. Add a new vertex $w$, distinct from all the vertices in $G$, and edges between $w$ and $s_1$, $s_2$, $t_1$ and $t_2$, respectively. The gadget $X(G)$ can be seen in Figure 2.

The idea behind $X(G)$ is to guarantee a winning strategy for Player 1 in the game of Whex played on all the set of vertices $V$, provided that Player 1 has a winning strategy in the game of Whex played on the set of vertices $V - \{s, t\}$. Consider the game of Whex on $X(G)$ as the usual game of Whex on a graph with distinct source and sink, except that Player 1 must begin by colouring one of the two sources, the players continue as in the usual game of Whex from the previous coloured source, and Player 1 wins if he reaches (and colours) one of the two sinks. Suppose Player 1 has a winning strategy in the game of Whex played on $G$; then, Player 1 wins in the game of Whex played on (all the vertices of) $X(G)$ by first taking one of the sources, and, afterwards, if Player 2 doesn't take $w$, Player 1 takes it and wins in his next move; otherwise, Player 1 uses his winning strategy for $G$, which applies *since he is the first one to take a vertex of* $V - \{s, t\}$; he will then reach one of the sinks and win. Conversely, suppose Player 2 has a winning strategy in the game of Whex played on $G$. Then Player 2 wins the game of Whex played on $X(G)$ by taking, in his first

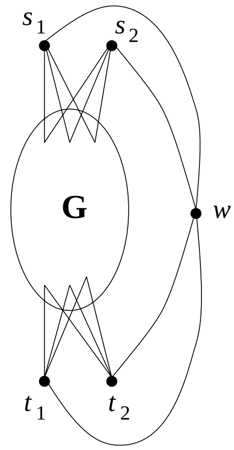


FIG. 2. The gadget $X(G)$.

move, vertex $w$, and then applying his winning strategy for $G$, which is possible since Player 1 is the first one to move in $V - \{s, t\}$.

Now, let me explain how to use $X(G)$ to obtain the formulas in normal form. Say we want to eliminate the existential quantifier in $\phi := \exists z \text{WHEX}[\overline{x}, \overline{y} : \theta(\overline{x}, \overline{y}, z)](\overline{0}, \overline{max})$ where $\theta$ is, by inductive hypothesis, a quantifier free projection and $|\overline{x}| = |\overline{y}| = k$. Proceed to construct, from a given finite $\tau_2$-structure (a graph) $\mathcal{A} = \langle \{0, 1, \ldots, n - 1\}, E^{\mathcal{A}}, C^{\mathcal{A}}, D^{\mathcal{A}} \rangle$, the graphs $X(\mathcal{A}_z)$ for each $z \in \{0, 1, \ldots, n-1\}$, where each $\mathcal{A}_z$ has universe $|\mathcal{A}|^k$ and edge relation determined by $\theta(\overline{x}, \overline{y}, z)$. Then add two new vertices $\mathbf{s}$ and $\mathbf{t}$ and, for each $z \in \{0, \ldots, n - 1\}$, join the sources in $X(\mathcal{A}_z)$ to $\mathbf{s}$, and the sinks to $\mathbf{t}$. In the new graph thus obtained, named $G_\phi$, $\mathbf{s}$ is codified as a $k + 4$ tuple of 0's and $\mathbf{t}$ as a $k + 4$ tuple of $max$'s; the vertices in each $X(\mathcal{A}_z)$ are codified as follows: the sources as $(\overline{0}, max, 0, max, z)$ and $(\overline{0}, 0, 0, max, z)$; the sinks as $(\overline{max}, max, 0, max, z)$ and $(\overline{max}, 0, 0, max, z)$; the $w$ as $(\overline{0}, 0, max, 0, z)$, and the other vertices as $(\overline{x}, max, max, 0, z)$ for $\overline{x} \notin \{\overline{0}, \overline{max}\}$. It is not difficult to describe $G_\phi$ with a quantifier free projection $\psi(\overline{x}, u_1, u_2, u_3, u_4, \overline{y}, v_1, v_2, v_3, v_3)$, and to show that Player 1 has a winning strategy in the game of Whex played on $G_\phi$ if and only if, for some $z \in \{0, \ldots, n - 1\}$, Player 1 has a winning strategy in the game of Whex played on $X(\mathcal{A}_z)$.

To eliminate the universal quantifier place the gadgets in series, and do a similar codification as in the existential case. It is here where the ordering relation is needed to write the appropriate quantifier free formula. The nested case is resolved in a similar manner by incorporating the $X(G)$ gadget to the construction of the nested case for TC in [9, Theorem 3.3], and following that proof the reader can write the appropriate formulas for our present problem. This completes the proof of the theorem. ∎

**Corollary 4.5 PSPACE** $=$ WHEX$^*$[FO$_s$] $=$ posWHEX$^*$[FO$_s$] $=$ posWHEX$^1$[FO$_s$] and WHEX is complete for **PSPACE** via quantifier free projections.

PROOF. Put together Theorems 4.3, 4.2, 3.5 and 4.4. ∎

I would like to remark that the variations of the game of Whex where: 1) players colour vertices of an *undirected* graph; 2) players colour *edges* of a directed or undirected graph, remain complete for **PSPACE** via quantifier free projections. To see the second point, observe that colouring an edge $\{u, v\}$ adjacent to an edge $\{w, u\}$ coloured in a preceding move has the same effect of colouring the vertex $v$ adjacent to the previously coloured vertex $u$; hence, all results for the Whex game on vertices apply with minor fixes to the game of Whex on edges. This contrast with Hex, where playing on edges does make a difference in the computational complexity of the problem, as it was remarked before.

Also, it should be noted that the version of the game of Whex where Player 2 begins the game (by colouring blue a vertex adjacent to the source), and where we ask the same query as before, namely, does Player 1 has a winning strategy?, yields a decision problem, which it will be distinguished by WHEX$'$, with the same computational and logical characteristics than our original WHEX (where Player 1 is the first one to play). The proofs of these facts amounts to slight modifications of the proofs given for the corresponding facts for WHEX. For example, to show WHEX$'$ is complete for **PSPACE**, view QSAT as the same alternating game among $\exists$ and $\forall$, but with $\forall$ beginning the game instead of $\exists$, and the board being formulas with alternating quantifiers beginning with $\forall$. Then the construction in Theorem 4.3 goes through and we have QSAT $\leq_{\log}$ WHEX$'$. We shall find WHEX$'$ more convenient to use for the application of our games given in the next section.

## 5 The complexity of a user–aided proof system

Let $\tau_3 = \{R, C, D\}$ be a vocabulary with a relation symbol $R$ of arity 3 and two constant symbols $C$ and $D$. A $\tau_3$–structure $\mathcal{A}$ of size $n$ can be regarded as a *path system*, that is, a set $A$ of $n$ vertices, a relation $R^{\mathcal{A}} \subseteq A \times A \times A$, a source $C^{\mathcal{A}} \in A$ and a sink $D^{\mathcal{A}} \in A$, where in order to reach a vertex $z$, there must be two other reachable vertices $x$ and $y$ so that $(x, y, z) \in R^{\mathcal{A}}$. This view of structures over $\tau_3$ is suitable for encoding the problem *Path System Accessibility* (see [7]), whose instances are path systems and yes–instances are instances where the sink is accessible from the source, a vertex $z$ being accessible if it is the source of the path system or if $R(x, y, z)$ holds for some accessible vertices $x$ and $y$. The resulting class of $\tau_3$-structures corresponding to Path System Accessibility was denoted PS by Iain Stewart in [13], considered by him as a generalized quantifier and added to first order logic to obtain the extension PS$^*$[FO$_s$], path system logic, which satisfies a projective normal form and which captures the class of polynomial time computable problems, **P**.

**Theorem 5.1 ([13])** PS$^*$[FO$_s$] = posPS$^*$[FO$_s$] = posPS$^1$[FO$_s$] = **P** and the normal form for posPS$^*$[FO$_s$] is quantifier free projective. Therefore, the problem PS is complete for **P** via quantifier free projections. ∎

Alternatively, we can regard a $\tau_3$–structure $\mathcal{A}$ of size $n$ as a *proof system*, by interpreting the elements of $\mathcal{A}$ as statements and the 3–ary relation $R^{\mathcal{A}}$ as a rule of inference, say, for example, one like resolution, which when applied to statements $x$ and $y$ yield the statement $z$, whenever $(x, y, z) \in R^{\mathcal{A}}$. This is the way I like to think of $\tau_3$–structures here, and, under this view, the problem PS is the set of proof systems $\mathcal{A}$ where the statement $D^{\mathcal{A}}$ is *provable* from the statement $C^{\mathcal{A}}$, where *provable* is synonymous with *accessible* and defined likewise.

Considering $\tau_3$–structures as proof systems, one then wonder about the complexity of proving theorems in a proof system where the user can help the system by adding "lemmas" along a proof, which are not necessarily provable from the initial statement $C^{\mathcal{A}}$. The intended meaning of this last statement is formalize in the following definition.

**Definition 5.2** Let $\mathcal{A} = \langle A, R^{\mathcal{A}}, C^{\mathcal{A}}, D^{\mathcal{A}} \rangle$ be a proof system. A statement $z$ in $\mathcal{A}$ is *partially provable* ($p.p$) if it is $C^{\mathcal{A}}$ or it is obtained from some partially provable statement $x$ and an arbitrary, non partially provable statement $y$, by applying the rule $R$ to $x$ and $y$ (i.e. $(x, y, z) \in R^{\mathcal{A}}$).

A statement $w$ is *provable from $u$ by a proof of partially provable statements* if, and only if, there is a sequence of partially provable statements $c_1, c_2, \ldots, c_n$, with $c_n = w$, which witnesses the following set of expressions:

$$\forall b_1 \exists c_1 : (b_1 = w \text{ or } b_1 \text{ is not p.p.) and } (u, b_1, c_1) \in R \tag{5.1}$$

and, for $1 \leq i < n$,

$$\forall b_{i+1} \exists c_{i+1} : (b_{i+1} = w \text{ or } b_{i+1} \text{ is not p.p.) and } (c_i, b_{i+1}, c_{i+1}) \in R. \tag{5.2}$$

Define the problem Partial Proof System (PPS) as the following class of $\tau_3$–structures:

$$\begin{aligned} \text{PPS} \quad := \quad & \{\mathcal{A} \in \text{STRUCT}(\tau_3) : \mathcal{A} \text{ is a proof system, where } D^{\mathcal{A}} \\ & \text{is provable from } C^{\mathcal{A}} \text{ by a proof of p.p. statements}\} \end{aligned}$$

It is not difficult to see that PPS is in **PSPACE**: at the $i$–th step of a proof we need to have in store the last statement $c_i$ partially provable, nondeterministically generate and store a statement $b_{i+1}$, and store a $c_{i+1}$ for which $(c_i, b_i, c_{i+1}) \in R^{\mathcal{A}}$. Furthermore, we have

**Theorem 5.3** WHEX$'$ $\leq_{\log}$ PPS.

PROOF. Given an undirected graph $G = \langle V, E, u, w \rangle$ with source $u$ and sink $w$, define the proof system $\mathcal{A}_G = \langle A, R, s, t \rangle$, with $A = V$, $s = u$, $w = t$ and

$$\begin{aligned} R \quad = \quad & \{(a, b, c) : E(a, c) \wedge E(a, b) \wedge b \neq c\} \\ & \cup \{(a, w, w) : E(a, w)\} \end{aligned}$$

Suppose $G \in$WHEX. Then, for whatever vertex $b_1$ Player 2 starts off the game of Whex on $G$, Player 1 can respond with $c_1$ such that $E(u, b_1)$ and $E(u, c_1)$ holds (and $c_1$ is on a path to $w$). Successively, in the $i$–th move, for whatever vertex $b_i$ Player 2 selects, Player 1 can respond with a $c_i$ such that $E(c_{i-1}, b_i)$ and $E(c_{i-1}, c_i)$ holds (and $c_i$ is on a path to $w$), and so on, until Player 1 reaches a vertex $c_n$ such that $E(c_n, w)$ holds. Then the sequence $c_1, c_2, \ldots, c_n, w$, is made of partially provable statements in $\mathcal{A}_G$, satisfying conditions (5.1) and (5.2). Hence, it constitutes a proof of $t$ from $s$ and, hence, $\mathcal{A}_G \in$PPS.

Conversely, if we assume $\mathcal{A}_G \in$PPS, then there is a proof of $t$ from $s$ by partially provable statements, which satisfies conditions (5.1) and (5.2). These conditions describe a winning strategy for Player 1 in the version of the game of Whex played on $G$ where Player 2 makes the first move. ∎

We have that the problem PPS is complete for **PSPACE** via log–space reducibility. It is argue next why PPS is also complete via quantifier free projections. First, consider PPS as a generalized quantifier and add it to first order logic (with built–in successor), and form the logic $\text{PPS}^*[\text{FO}_s]$. Then observe that the problem DTC is expressible in $\text{posPPS}^*[\text{FO}_s]$: the intuitive idea is that for a given graph $G$, with distinguished vertices $s$ and $t$, a path $\langle s, c_1, c_2, \ldots, c_n, t\rangle$ from $s$ to $t$ can be seen as a proof of $t$ from $s$ by the following sequence of applications of the rule $R$:

$$(s,t,c_1),(c_1,t,c_2),\ldots,(c_i,t,c_{i+1}),\ldots,(c_n,t,t)$$

Next, to show that $\text{posPPS}^*[\text{FO}_s] = \text{posPPS}^1[\text{FO}_s]$ proceed almost identically as in Stewart's proof of the normal form for $\text{PS}^*[\text{FO}_s]$ (Theorem 4.2 of [13]). For example, to eliminate the existential quantifier in the sentence

$$\exists w \text{PPS}[x,y,z:\psi(x,y,z,w)](C,D)$$

consider, for a structure $\mathcal{A}$ of size $n$, a disjoint union of the $n$ proof systems described by $\psi^{\mathcal{A}}(x,y,z,i)$, for $i=0,1,\ldots,n-1$, with a common initial statement and a common final statement. We then have a result analogous to Theorem 5.1.

**Theorem 5.4** $\text{PPS}^*[\text{FO}_s] = \text{posPPS}^*[\text{FO}_s] = \text{posPPS}^1[\text{FO}_s] =$ **PSPACE** and the normal form for $\text{posPPS}^*[\text{FO}_s]$ is quantifier free projective. Therefore, the problem PPS is complete for **PSPACE** via quantifier free projections (with the successor relation). ∎

(An alternative way of proving the completeness of PPS via qfp's follows from observing that the reduction of WHEX′ to PPS in Theorem 5.3 can be described by quantifier free projective formulas. On the other hand, WHEX′ can be shown complete via qfp's by same arguments employed for the completeness of WHEX, and then use that quantifier free projections are transitive [10].)

The above theorem, together with Theorem 5.1, have as consequence a characterization of the **P** versus **PSPACE** problem. **P** = **PSPACE** if and only if PPS is first order reducible to PS. This tells us, informally, that in order for the class **PSPACE** be equal to **P**, it is sufficient to show that any proof system that uses auxiliary statements in its proofs, can be effectively simulated by a proof system where all proofs are constituted by provable statements from the initial one $C^{\mathcal{A}}$.

One more observation about the problem PPS is that it is definable in *partial fixed point logic*, or PFP[PO]. This is the closure of first order logic with the partial fixed point operator PFP and all boolean operations. PFP applies to formulas of the form $\varphi(x_1,\ldots,x_k,R)$, where $R$ is a relational variable of arity $k$, to make up the formula $\text{PFP}[\overline{x},R:\varphi(\overline{x},R)](t_1,\ldots,t_k)$ whose interpretation is as follows: for an appropriate structure $\mathcal{A}$, $\mathcal{A}\models \text{PFP}[\overline{x},R:\varphi(\overline{x},R)](t_1,\ldots,t_k)$ if and only if $(t_1,\ldots,t_k)^{\mathcal{A}}$ is in the fixed point (if it exists) of the sequence of sets

$$\varphi^0_{\mathcal{A}} \quad := \quad \{(a_1,\ldots,a_k)\in A^k : \mathcal{A}\models\varphi(a_1,\ldots,a_k,\emptyset)\}$$

and for $i>0$

$$\varphi^i_{\mathcal{A}} \quad := \quad \{(a_1,\ldots,a_k)\in A^k : \mathcal{A}\models\varphi(a_1,\ldots,a_k,\varphi^{i+1}_{\mathcal{A}})\}$$

(For further details see [5].)

Now, it is not difficult to see that PPS is definable in PFP[PO] by the sentence

$$\exists w(\mathrm{PFP}[z, X : z = C \quad \vee \quad \exists x \forall y(X(x) \wedge (\neg X(y) \vee y = D) \wedge R(x, y, z))](w) \wedge w = D) \tag{5.3}$$

(essentially we put in the set $X$ the partially provable statements, beginning with $C$ and until we reach $D$).

If we consider PFP[PO$_s$], namely, PFP over first order logic with the built–in successor, then using the facts just proved, that PPS is complete for **PSPACE** via qfp's with successor, and that PPS is definable in PFP[PO] (hence in PFP[PO$_s$]), we obtain that every problem computable by Turing machines with a polynomial space bound is definable in PFP[PO$_s$]. Conversely, the set of finite structures that satisfy a given sentence in PFP[PO$_s$] can be decided by a polynomial space bounded algorithm (see [5]). Thus, we have another proof of Abitebuol and Vianu's logical characterization of **PSPACE**:

**Theorem 5.5 ([1]) PSPACE** = PFP[PO$_s$]. ∎

## 6 WHEX on graph with outdegree $\leq 3$.

Let WHEX3 be the set of digraphs of outdegree at most 3 which are yes–instances of WHEX. I shall reduce the problem QSAT to WHEX3 via a quantifier free projection. To do that view QSAT as a problem over the vocabulary $\sigma = \{P, N, U\}$, where $P$ and $N$ are binary relation symbols and $U$ is unary; then, an instance of QSAT is a finite $\sigma$–structure $\mathcal{A} = \langle A, P, N, U\rangle$, where $P(i, j)$ holds in $\mathcal{A}$ if and only if variable $j$ occurs positive in clause $i$, $N(i, j)$ holds in $\mathcal{A}$ if and only if variable $j$ occurs negative in clause $i$, and $U(i)$ holds in $\mathcal{A}$ if and only if variable $i$ is universally quantified. A yes–instance of QSAT is an instance where Player 1 has a winning strategy in the game of Qsat described previously.

**Theorem 6.1** QSAT $\leq_{qfp}$ WHEX3.

Proof. The strategy is as follows: first, a log–space reduction from QSAT to WHEX3 is given, which is essentially the reduction of QSAT to WHEX given in Theorem 4.3 with suitable modifications meant to keep the outdegree below 3. Then it is indicated how to codify the reduction and express it with a quantifier free first order projection.

Let $\Phi := \exists x_1 \forall x_2 \cdots Q_n x_n(C_1 \wedge C_2 \wedge \ldots \wedge C_k)$, where each clause $C_i$ is a conjunction of literals. Observe that $n$ is the number of variables in $\Phi$, and $k$ is the number of clauses. Define the graph $G_\Phi = \langle V, E\rangle$ as follows:

$$\begin{aligned} V &= \{s, t, y\} \cup \{x_i, \overline{x}_i, u_i, \overline{u}_i, v_i : 1 \leq i \leq n\} \\ &\cup \{w_{2i} : 1 \leq i \leq \lceil n/2\rceil\} \cup \{c_i, z_i : 1 \leq i \leq k\} \\ &\cup \{p_i^j, n_i^j, y_i^j : 1 \leq j \leq n, 1 \leq i \leq k\}. \end{aligned}$$

$$\begin{aligned} E &= \{(s, x_1), (s, \overline{x}_1), (v_n, c_1), (v_n, z_1), (z_k, t), (y, t)\} \\ &\cup \{(x_i, u_i), (\overline{x}_i, \overline{u}_i), (u_i, t), (\overline{u}_i, t), (x_i, v_i), (\overline{x}_i, v_i) : 1 \leq i \leq n\} \\ &\cup \{(v_i, x_{i+1}), (v_i, \overline{x}_{i+1}) : 1 \leq i \leq n-1\} \end{aligned}$$

$$
\begin{aligned}
&\cup \quad \{(v_{2i}, w_{2i}), (w_{2i}, t) : 1 \leq i \leq \lceil n/2 \rceil\} \\
&\cup \quad \{(z_i, z_{i+1}), (z_i, c_{i+1}) : 1 \leq i \leq k-1\} \\
&\cup \quad \{(c_i, y), (c_i, p_i^1), (c_i, n_i^1) : 1 \leq i \leq k\} \\
&\cup \quad \{(p_i^j, y_i^j), (n_i^j, y_i^j), (y_i^j, t) : 1 \leq i \leq k, 1 \leq j \leq n\} \\
&\cup \quad \{(p_i^j, p_i^{j+1}), (n_i^j, n_i^{j+1}) : 1 \leq i \leq k, 1 \leq j \leq n-1\} \\
&\cup \quad \{(p_i^j, \ddot{u}_j) : \text{the literal } \ x_j \ \text{ is in clause } \ C_i\} \\
&\cup \quad \{(n_i^j, u_j) : \text{the literal } \ \neg x_j \ \text{ is in clause } \ C_i\}.
\end{aligned}
$$

$G_\Phi$ is a graph with outdegree at most 3, and it is not difficult to see that Player 1 has a winning strategy in the Qsat game played on $\Phi$ if and only if Player 1 has a winning strategy in the Whex game played on $(G_\Phi, s, t)$. In order to write the above reduction as a quantifier free formula, begin by codifying the 14 types of vertices as the following sextuples ($max$ is abbreviated as $m$): $s = (0,0,0,0,0,0)$, $t = (m,m,m,m,m,m)$, $y = (m,m,m,0,0,0)$, $x_i = (i,0,m,m,m,m)$, $\overline{x}_i = (i,m,0,m,m,m)$, $u_i = (i, 0, 0, m, m, m)$, $\overline{u}_i = (i,m,0,0,m,m)$, $v_i = (i,0,m,0,m,m)$, $w_i = (i,m,m,0,m,m)$, $c_i = (i, m, m, m, 0, m)$, $z_i = (i,m,m,0,0,m)$, $p_i^j = (i,j,0,m,0,m)$, $n_i^j = (i,j,0,m,0,0)$, and $y_i^j = (i, j, 0, 0, m, 0)$. Then I proceed to define the edge relation with a formula in the variables $x_1, x_2, \ldots, x_6, y_1, y_2, \ldots, y_6$, consisting of disjunctions of clauses, one for each of the type of edges described above, and most of which can be easily seen to be a quantifier free conjunction. Therefore, I just write down the (possibly) not so trivial quantifier free formulas that describe some of the edges. Take for example an edge of the form $(v_{2i}, w_{2i})$. This is an edge corresponding to variable $i$ universally quantified, and so is equivalent to have $(v_i, w_i) \longleftrightarrow U(i)$ holding in $G_\Phi$; hence, we can express this edge with the formula:

$$(\neg\phi_{vw} \wedge \neg U(x_1)) \vee (\phi_{vw} \wedge U(x_1))$$

where

$$
\begin{aligned}
\phi_{vw} \quad := \quad & x_1 = y_1 \wedge x_2 = x_4 = y_4 = 0 \\
\wedge \quad & x_3 = x_5 = x_6 = y_2 = y_3 = y_5 = y_6 = m
\end{aligned}
$$

Edges such as $(p_i^j, \overline{u}_j)$ are given by the formula

$$
\begin{aligned}
& x_2 = y_1 \wedge x_3 = x_5 = y_3 = y_4 = 0 \\
\wedge \quad & P(x_1, x_2) \wedge x_4 = x_6 = y_2 = y_5 = y_6 = m
\end{aligned}
$$

Edges such as $(n_i^j, u_j)$ are given by the formula

$$
\begin{aligned}
& x_2 = y_1 \wedge x_3 = x_5 = x_6 = y_2 = y_3 = 0 \\
\wedge \quad & N(x_1, x_2) \wedge x_4 = y_4 = y_5 = y_6 = m
\end{aligned}
$$

And for edges such as $(v_i, x_{i+1})$, $(z_i, z_{i+1})$, or $(p_i^j, p_i^{j+1})$ we use the (built–in) successor relation to write up appropriate quantifier free conjunctions. This ends the proof of the theorem. ∎

**Corollary 6.2** WHEX3 is complete for **PSPACE** via first order reductions, and

$$\text{posWHEX3}^1[\text{FO}_s] = \text{WHEX3}^1[\text{FO}_s] = \textbf{PSPACE}.$$

∎

The problem WHEX2 whose instances are digraphs of outdegree at most 2, and yes–instances are instances where Player 1 has a winning strategy for the game of Whex, can be solved in polynomial time. This is immediate, since there is at most one alternative for each player's next move, and so all the moves are forced.

## 7 WHEX on unordered structures

In this section a comment is made on the expressive capabilities of WHEX, as a generalized quantifier, with respect to properties of arbitrary (but finite) structures. If we do not include the built–in successor relation, so that our input structures can be unordered as well as ordered, then the logic WHEX$^*$[FO] has a 0–1 law, since WHEX is a problem closed under extensions and, thus, satisfies the conditions for logics with generalized quantifiers to have asymptotic probability equal to either 0 or 1, as established in [3] for graph problems and later generalized to any problem in [14]. Therefore, WHEX$^*$[FO] does not capture **PSPACE**.

Also, just as it was shown in [2] that HEX$^*$[FO] does not have a normal form using suitable Ehrenfeucht–Fraissé type of games, the same can be shown for the logic WHEX$^*$[FO]:

**Theorem 7.1** There are problems definable in WHEX$^*$[FO] which can not be defined by a sentence of WHEX$^1$[FO] in which the operator WHEX does not appear within the scope of the quantifier $\forall$.

PROOF. The proof is the same as the proof of [2, Proposition 4]. Incidentally, the structures $\mathcal{S}_m$ and $\mathcal{T}_m$ in the proof of that proposition are also in WHEX and not in WHEX, respectively. ∎

## 8 Final Remarks

WHEX illustrates the necessity of developing tools for sharpening the classification of computational problems. It is a problem based on a game where, intuitively, should be easier than the game of Hex to design winning strategies. However I have shown how similar WHEX and HEX are with respect to their computational complexity and logic expressive power. Nonetheless, I believe some distinctive features among these two problems can be obtained by further exploiting their logical characteristics (besides the few structural differences pointed out through this paper). For example, after Theorem 5.4, it was remarked that WHEX$'$, and hence WHEX, can be reduced to PPS via a quantifier free projection; sustituting in formula (5.3) the relation $R$ by its description in the vocabulary $\tau_2 = \{E, C, D\}$, as suggested in the proof of Theorem 5.3, we get a definition of WHEX in the logic PFP[PO] by a formula structurally simple: it contains a second order variable of arity 1 and the quantifiers (PFP and the first order quantifiers) are relativized by atomic formulas. This fit the pattern of formulas in a *guarded fixed point logic* (see [8]), therefore suggesting a finer subclass of **PSPACE** problems (where WHEX and PPS belongs) as those problems definable in some fragment of the logic PFP[PO], possibly a "guarded partial fixed point" logic. I will leave this characterization as an open problem.

Finally, I will end with the following application of WHEX to a problem of communication in networks, suggested to me by Iain Stewart. Given a network with a source

$s$, a sink $t$ and a fixed positive integer $\lambda$, consider each edge between two nodes $i$ and $j$ as having a valuation $t_{ij}$, which might represent the time that a message takes to go from $i$ to $j$. We consider also that, for each node $i$, up to $\lambda$ nodes might fail in processing the information per unit of time $t_{ij}$ among $i$ and any other adjacent node $j$. We wish then to know if a message can be send from $s$ to $t$, and in the afirmative, which is the strategy for transmission that gives the less time possible. Observe that WHEX is a particular instance of this problem: take $\lambda = t_{ij} = 1$; hence, this problem is **PSPACE**–hard.

**Acknowledgement:** I came around to the definition of WHEX, its logical and computational properties, through various conversations with Iain Stewart. I am grateful to him for all his help. Thanks also to EPSRC for a Visiting Fellowship (Grant GR/M 91006) that made possible one of my visits to Stewart's research group at Leicester. I am also grateful to the referees for their many valuable remarks and comments.

## References

[1] S. Abiteboul and V. Vianu. Fixpoint extensions of first–order logic and Datalog–like languages, in: Proc. 4th IEEE Ann. Symp. on Logic in Comp. Sci. (1989) 71–79.

[2] A.A. Arratia-Quesada and I.A. Stewart. Generalized Hex and logical characterizations of polynomial space. Inf. Procs. Letters 63 (1997) 147–152.

[3] A. Dawar and E. Grädel. Generalized quantifiers and 0–1 laws, in: Proc. 10th IEEE Ann. Symp. on Logic in Comp. Sci. (1995) 54–64.

[4] H.D. Ebbinghaus. Extended logics: the general framework, in: Barwise and Feferman, eds., Model Theoretic Logics (Springer–Verlag, 1985).

[5] H.D. Ebbinghaus and J. Flum. Finite Model Theory (Springer-Verlag, 1995).

[6] S. Even and R.E. Tarjan. A combinatorial problem which is complete in polynomial space. J. Assoc. Comput. Mach. 23 (1976) 710–719.

[7] M. R. Garey and D. S. Johnson. Computers and Intractability - A Guide to the Theory of NP-Completeness (W. H. Freeman and Co., 1979).

[8] E. Grädel and I. Walukiewicz. Guarded fixed point logic, in: Proceedings of 14th IEEE Symposium on Logic in Computer Science LICS '99, Trento, pp. 45-54, 1999.

[9] N. Immerman. Languages that capture complexity classes. SIAM J. of Computing 16, 4 (1987) 760–778.

[10] N. Immerman. Descriptive Complexity (Springer, 1998).

[11] T.J. Schaefer. On the complexity of some two-person perfect information games. J. Comp. Syst. Sci. 16 (1978) 185–225.

[12] I.A. Stewart. Comparing the expressibility of languages formed using NP–complete operators. J. Logic Computat. 1 (1991) 305-330.

[13] I.A. Stewart. Logical description of monotone NP problems. J. Logic Computat. 4 (1994) 305-330.

[14] I.A. Stewart. Program schemes, arrays, Lindström quantifiers and zero-one laws, in: J. Flum and M. Rodriguez-Artalejo, eds., Proc. 8th Ann. Conference of the European Association for Computer Science Logic (LNCS Vol. 1683, Springer-Verlag, Berlin, 1999).

[15] L.J. White. An efficient algorithm for the Shannon switching game. Ann. New York Acad. Sci. 175 (1970) 411–412.

Received 21 of March, 2002

# Labelled Natural Deduction for Conditional Logics of Normality

KRYSIA BRODA, *Department of Computing, Imperial College, London SW7 2BZ, E-mail: kb@doc.ic.ac.uk*

DOV M. GABBAY, *Augustus De Morgan Professor of Logic, Department of Computer Science, King's College, London WC2R 2LS, E-mail: dg@dcs.kcl.ac.uk*

LUÍS C. LAMB,[1] *Department of Computing, Imperial College, London SW7 2BZ, E-mail: ldcl@doc.ic.ac.uk*

ALESSANDRA M. RUSSO, *Department of Computing, Imperial College, London SW7 2BZ, E-mail: ar3@doc.ic.ac.uk*

## Abstract

We propose a family of Labelled Deductive Conditional Logic systems (LDCL) by defining a Labelled Deductive formalisation for the propositional conditional logics of normality proposed by Boutilier and Lamarre. By making use of the Compilation approach to Labelled Deductive Systems (CLDS) we define natural deduction rules for conditional logics and prove that our formalisation is a generalisation of the conditional logics of normality.

*Keywords*: Labelled Deductive Systems; Conditional Logic; Natural Deduction

## 1 Introduction

Conditional patterns of inference are ubiquitous in human reasoning. Because of their intrinsic presence in human discourse this subject has been the theme of intense investigation in philosophy, particularly in philosophical logic. Formal analyses of sentences of the form *If* $\alpha$ *then* $\beta$, where $\alpha$ is called the antecedent and $\beta$ is called the consequent, have been extensively studied by philosophers who were interested in explaining under which circumstances conditional sentences hold, see e.g. [1, 11, 13, 18, 24, 26, 29, 35, 40, 45, 46, 47, 49]. In 1968, Stalnaker published a formal semantics and an axiomatic system for conditional logic in a paper that set in motion the modern study of conditional logic [46]. Most of Stalnaker's seminal work, as well as his later joint work with Thomason [47], was concentrated on *counterfactual conditionals*, as opposed to *indicative conditionals* (also studied by philosophers),

[1] Current address: Faculdade de Informática, PUCRS - Av. Ipiranga 6681, Porto Alegre, RS, 90450-030, Brazil. E-mail: lamb@inf.pucrs.br

 

and classical *material conditional* [4]. In particular, Stalnaker proposed a notion of a counterfactual conditional based on a concept of "similarity" between Kripke worlds: a counterfactual conditional $\alpha > \beta$ is true if the consequence $\beta$ is true in the world "most similar" to the world in which $\alpha$ is true [47]. His initial proposal of conditional logic (in 1968) sparked a proliferation of research work, which developed Stalnaker's ideas even further by proposing alternative logical analysis of conditionals under different philosophical perspectives [26]. A number of distinct and competing theories has since been presented in the literature, including, for instance, Boutilier's conditional logic of normality [7], where a conditional $\alpha > \beta$ is interpreted as "if $\alpha$ then normally $\beta$". However, no universally accepted unified theory of conditional logic has ever been proposed or investigated. This paper partly addresses this issue by providing a *unified* theory presentation for the family of conditional logics of normality and extensions given by introducing additional modality operators proposed by Boutilier and Lamarre (see [7] and [32]). However, as discussed in Section 7, this paper acts as the starting point for a broader agenda of research into the development of a unifiying framework for various classes of conditional logics. A full investigation of the applicability of our approach to all existing conditional logics of normality is a topic of future research.

The choice of Boutilier's conditional logics has been driven by our interest towards those conditional logics that can be used to model common sense reasoning in real application problems. As discussed later in this section, one of the advantages of our approach is to provide logical systems with features that make them more suitable for the needs of specific applications, and Boutilier's conditional logics have been shown to be related to issues involved in common sense reasoning. With the development of *non-monotonic* logics (e.g.,[38, 42, 39]), various authors have started investigating the relationships between those logics and conditional logics. For instance, Lehmann and Magidor identified connections between the so called KLM (non-monotonic) systems [31] and conditional logics [34], and further relations between non-monotonic and conditional logics were also presented in [37, 30, 20]. Building on these results some authors have then shown how conditional logics can be useful in modelling applications in artificial intelligence, as for instance in formalising common sense and practical reasoning; see e.g. [2, 15, 37, 13]. Among these, *conditional logics of normality* have received particular attention, as properties like "typicality" and "normality" are of great interest in artificial intelligence. The analysis of normality (or generic) conditionals has been a big research challenge in artificial intelligence, since this type of conditional construction can be seen as a form of common sense reasoning, especially for representing defeasible assertions [7]. However as in philosophical logic, most of the work is centered around semantic approaches, such as preferential models [31, 34], ranked systems [23], and quantitative approaches, especially probabilistic semantics [41]. The approach described in this paper provides conditional logics of normality with a unifying logical representation and uniform natural deduction proof system, which can facilitate their use in real applications.

Existing proof-theoretic presentations of conditional logics are mostly based on axiomatic systems in Hilbert-Frege style, perhaps because of the strong interests of the philosophical community in the area. Even in typical computer science and artificial intelligence literature [15, 7, 20, 32] the presentation of conditional logics has mainly been based on axiomatic systems. Proof procedures for varieties of conditionals have scarcely been proposed so far. In [25, 22, 14, 3] some work on tableaux and Gentzen systems for specific types of conditional logic has been presented, and studies on the complexity problem for conditional logics have been presented in [19]. We believe, however, that natural deduction presentation and formalisation of proofs - as Gentzen and Prawitz have claimed - represents a more natural way of modelling practical reasoning. In placing the emphasis on the relations between premises and conclusions, natural deduction "resembles far more closely the reasoning of ordinary life" [17]. We aim to bridge the gap between the semantics of conditional logics (which have been more extensively studied) and their proof theory presentation. In particular, we propose a *uniform natural deduction* proof system for Boutilier's family of conditional logics of normality and extensions with additional modal operators, providing also a contribution to the proof theoretical study of these existing logics.

The approach developed in this paper builds upon the *Labelled Deductive Systems* (LDS) methodology, proposed by Gabbay in [21], and a related framework called *Compiled Labelled Deductive System* (CLDS), developed by Russo in [44]. The LDS methodology has been initially proposed both for the theoretical study of logics and for the development of logical systems suitable for the needs of specific applications [21]. The basic unit of information of an LDS system is a *labelled formula*, where the label provides useful additional information about a formula, which is not obviously expressed by it. For instance, labels can represent the worlds in which a formula holds (in the case of modal logic), a resource in resource logics, the time at which the formula holds in temporal logics, numerical values in the case of probabilistic, fuzzy or many-valued logics, or information about the type of a formula in the case of the Curry-Howard interpretation of formulas as typed theories. Labels belong to a *labelling algebra*, which syntactically identifies semantic and/or proof theoretic properties of the underlying logics. For example, in the case of modal logic, the labelling algebra can be defined as a binary first-order theory which axiomatises the Kripke semantic notion of an accessibility relation. Derivation rules act on both labels and formulae, according to certain fixed rules of propagation based on the labelling algebra and on the logical meaning of the connectives. The explicit syntactic reference to semantic properties of the underlying logic, allows the development of proof systems that are *uniform* for a given family of logics. For example, in the case of modal logic, the single LDS proof system can be applied to the whole family of normal modal logics by just considering different underlying labelling algebras and leaving the inference rules for the modality operators unchanged [44]. The CLDS framework is a variation of an LDS system, which retains the same features of an LDS system but facilitates, in addition, a *unified semantic* formulation of different logics (possibly belonging to different families), and a *generalisation* of the natural deductive system of these logics [9, 8]. In a CLDS system, the notion of a logical theory is generalised to a notion of *configuration*, i.e. a (possibly singleton) "structure" of local theories. For example, in the case of modal logic, a CLDS theory is a structure of actual worlds, to which arbitrary modal theories can be locally associated [9] The deductive process can be defined with respect to the overall configuration, allowing the inference of what is true at different points of a given structure.

Our approach, called *Labelled Deductive Conditional Logic* (LDCL) makes use of the features and advantages of both LDS and CLDS, to provide a natural deduction proof system for conditional logics of normality that (i) is *uniform* for the whole family of these logics, and (ii) *generalises* the standard formulation of these logics by enabling reasoning on structures of local conditional theories. This is to provide a logic, based on conditional reasoning, able to fulfill the increasing need in some application areas for using not only sets, but structures of information. In particular, in a LDCL system explicit references to structures of arbitrary conditional theories can be made, while retaining the conventional syntax of a conditional logic of normality and a concise proof-theoretical presentation of the underlying conditional logic. This is achieved by combining the conditional logics of normality with an appropriate labelling algebra. The labelling algebra is defined as a binary first-order theory which axiomatises the semantic notion of normal conditional relation between the worlds that are *most similar* or *most normal* compared to each other. The conditional language is given in a traditional way [7], without requiring a full translation into modal logic. The two languages (conditional language and labelling language) are combined via the notion of a *declarative unit*. A declarative unit is a labelled formula of the form $\alpha : w$, which expresses that the conditional formula $\alpha$ is true at the label (i.e. possible world) $w$. This combined language retains the advantage of using a traditional formalisation of conditional logic, and at the same time is rich enough to allow explicit syntactic reference to normal conditional relations between possible worlds. A LDCL theory is a generalisation of standard conditional theories in that it facilitates reasoning about what is true at different points in a (possibly singleton) structure of actual worlds. An example of a LDCL theory is the set $\{R\omega\omega_1, \alpha > \beta : \omega, \Box\alpha : \omega_1, \beta : \omega_1\}$. The proof system is a natural deduction system that is uniform to any of the existing Boutilier's conditional logics of normality, in the sense that the same set of deduction rules can be applied to each of these conditional logics of normality. The difference between one conditional logic and another is captured entirely by the labelling algebra.

To summarise, this paper proposes a family of Labelled Deductive Conditional Logic systems for the existing Boutilier's propositional conditional logics of normality, which contributes to the theoretical study of these logics in the following ways. (1) It provides them with a uniform natural deduction system, which can also be used to investigate extensions of these logics with additional modality operators. (2) It extends the scope of these logics in order to deal with domains containing not just one, but a structure of one or more local or actual conditional theories. A proof is given that existing axiomatisations of Boutilier's propositional conditional logic of normality are subsumed by the LDCL systems described here. (3) It provides an alternative approach to conditional reasoning whose increased syntactic expressivity and deductive power contributes towards the long term aim of constructing logical formalisms more suitable to the needs of applications (in the sense described in [Gab92]).

The paper is organized as follows. In Section 2 a brief summary of Boutilier's conditional logic of normality is given. In Section 3 the language and syntax of a LDCL system is defined together with the notion of a configuration – a LDCL equivalent to a conditional theory. In Section 4, a natural deduction style proof theory for a family of LDCL systems is given in which inference rules are applied to configurations. In Section 5, a model-theoretic semantics, based on a translation method into classical logic, is described together with a notion of semantic entailment. Its equivalence to Boutilier's semantics for conditional logic of normality is also shown whenever the initial configuration is a single point. Soundness and completeness results of the proof system described in Section 4 are proved with respect to this semantics. In Section 6, extensions of Boutilier's conditional logics with additional modalities are formalised and proved to be also sound and complete with respect to the semantics we present. Section 7 ends the paper with a general discussion.

## 2 On Conditional Logics of Normality

As mentioned in the previous section, a (uniform) natural deduction proof presentation for conditional logics is lacking in the literature. Among the very few examples, we can mention Thomason's Fitch-style natural deduction system for Stalnaker's C2 [48], and Chellas's approach based on a full translation of conditional statements $\alpha > \beta$ into modal formulae of the form $[\alpha]\beta$ representing the consequence $\beta$ to be necessary with respect to the antecedent $\alpha$ [12]. On the other hand, among the researchers who have investigated the use of conditional logics in Artificial Intelligence, Boutilier and Lamarre presented an (axiomatic) approach for conditional logics of normality based on standard models for propositional modal logic, as shown in [7, 32]. Our aim is to provide a uniform and generalised natural deduction proof system for these latter logics. Therefore, before describing the details of our system, we briefly summarise Boutilier's propositional conditional logics of normality.

### *2.1 Boutilier's Conditional Logics of Normality*

In [5] a "conditional logic of normality" is proposed in order to provide a logic with enough expressive power to deal with some forms of non-monotonic and defeasible reasoning. A normality conditional, denoted by "$\alpha > \beta$", has been defined as "if $\alpha$ then normally $\beta$" [7]. The models for normality reasoning are essentially Kripke structures. They include a partial ordering between possible worlds where $R\omega\omega'$ denotes that $\omega'$ is "at least as normal" as $\omega$, an idea similar to the intuitive semantics of counterfactual conditional logics based on possible worlds structures. In the conditional logic of normality, the conditional $\alpha > \beta$ is true in a world $\omega$ if for each "$\alpha$-world" there exists a world $\omega_1$ at least as normal as $\omega$ in which both $\alpha$ and $\beta$ hold, i.e. an "$\alpha \wedge \beta$-world" $\omega_1$ such that the material implication $\alpha \supset \beta$ holds at all worlds at least as normal as $\omega_1$ ([7] pp. 103). Following this interpretation of normality, Boutilier's notion of conditional $\alpha > \beta$ is equivalent to the following modal formula [5, 7]:

$$\Box(\alpha \supset \Diamond(\alpha \wedge \Box(\alpha \supset \beta))) \tag{2.1}$$

where, $\Box\alpha$ means that the formula $\alpha$ is true at all normal worlds and $\Diamond\alpha$ means that the formula $\alpha$ holds at some normal world. Within this context, the basic Boutilier's conditional logic is the logic $CT4$, also referred to, in the literature, as mono-modal logic $CT4$. We take this logic as our basic logic as well, and will refer to it as the Boutilier's Conditional Logic (BCL).

The logic $CT4$ and its extensions have been proved to be equivalent to extensions of modal logic $S4$ [32, 5]. Informally, reading the (accessibility) conditional relation $R$ between two possible worlds as the relation "at least as normal as", makes it into a relation that is at least reflexive and transitive, for intuitively, a world is at least as normal as itself. Other conditions can be imposed on the relation $R$, giving rise to extensions of the conditional system $CT4$. For instance, the addition of the seriality property to the conditional relation $R$ gives Boutilier's logic $CT4D$, which is the logic $CT4$ extended with the modal axiom **D** [7]. In [7], $CT4$ has also been extended with additional modalities. These bimodal extensions do not include additional semantic features, but have additional expressive power which allows assessment of information at inaccessible, i.e."less normal", worlds. For instance, the bimodal extension $CT4O$ allows the axiomatisation of the problem of irrelevance and it also makes it easier to express different types of defeasible reasoning, e.g. autoepistemic logic. Boutilier has also showed a correspondence between his conditional logics and other existing conditional systems. In particular, the conditional logics of normality $CT4O$ and $CO$ (bi-modal extension of $CT4$ where the relation $R$ is totally connected) have been shown to correspond to the logics $P$ and $R$ ([34]) whenever both $CT4O$ and $CO$ systems are restricted to their flat fragments (i.e. the ones that do not permit nesting of conditional operators) [7]. Some theorems of the $P$ systems have also been shown to be valid within the basic Boutilier's conditional logic $CT4$. These are $ID, Or, RCM, RT, CM, And$, and we will show in Section 5.2 how these theorems are also captured by our LDCL system.

The syntax of a propositional conditional logic (BCL) is defined over the following language.

**Definition 2.1 (Language $\mathcal{L}_C$)** Let BCL be a normal conditional logic. A *propositional language* $\mathcal{L}_C$ of BCL is given by a countable set of propositional letters denoted by $p, q, r, ...$, the classical connectives $\neg$, $\wedge$, $\vee$, $\supset$, the conditional operator $>$, and the modal operators $\Box$, $\Diamond$. The grouping symbols "(" and ")" are also assumed to be part of $\mathcal{L}_C$.

The semantics of a normal conditional logic BCL builds upon Kripke semantic notions. A model for BCL is essentially a standard $S4$ Kripke structure, whereas the notion of satisfiability extends the standard notions of satisfiability for modal logic with the definition of satisfiability for conditional formulae. These are formally defined below.

**Definition 2.2 (Models for BCL)** Let $\mathcal{L}_C$ be a BCL language. A BCL *model* is a tuple $M = \langle W, R, v\rangle$ where $W$ is a set of possible worlds, $v$ is a mapping that assigns to each propositional letter of $\mathcal{L}_C$ a subset of $W$, and $R$ is a binary relation over $W$ that satisfies reflexivity (i.e. $\forall \omega \in W$, $R\omega\omega$) and transitivity (i.e. $\forall \omega_i, \omega_j, \omega_k \in W$, $R\omega_i\omega_j$ and $R\omega_j\omega_k$ implies $R\omega_i\omega_k$).

A BCL formula is true or false only with respect to a particular possible world. A wff $\alpha$ is said to be *true at a possible world* $\omega$ *of a model* $M$, written $(M, \omega) \models \alpha$, if and only if one of the following satisfiability conditions holds.

**Definition 2.3 (Satisfiability)** Let $\mathcal{L}_C$ be a BCL language, $M = \langle W, R, v\rangle$ be a model, $\omega$ be a possible world, and let $\alpha$ and $\beta$ be two wffs of $\mathcal{L}_C$. The satisfiability relation $\models$ is uniquely defined as follows:

1. $(M, \omega) \models p$ iff $\omega \in v(p)$ (for a propositional letter $p$)

2. $(M, \omega) \models \neg\alpha$ iff $(M, \omega) \not\models \alpha$
3. $(M, \omega) \models \alpha \wedge \beta$ iff $(M, \omega) \models \alpha$ and $(M, \omega) \models \beta$
4. $(M, \omega) \models \alpha \vee \beta$ iff $(M, \omega) \models \alpha$ or $(M, \omega) \models \beta$
5. $(M, \omega) \models \alpha \supset \beta$ iff $(M, \omega) \models \beta$ or $(M, \omega) \not\models \alpha$
6. $(M, \omega) \models \Box\alpha$ iff for all $\omega_1 \in W$, if $R\omega\omega_1$ then $(M, \omega_1) \models \alpha$
7. $(M, \omega) \models \Diamond\alpha$ iff there exists a $\omega_1$ such that $R\omega\omega_1$, and $(M, \omega_1) \models \alpha$
8. $(M, \omega) \models \alpha > \beta$ iff for all $\omega_1 \in W$, if $R\omega\omega_1$ then either (a) or (b) holds:
   (a) there exists $\omega_2 \in W$ such that $R\omega_1\omega_2$ and $(M, \omega_2) \models \alpha$ and for each $\omega_3 \in W$ such that $R\omega_2\omega_3$, $(M, \omega_3) \not\models \alpha$ or $(M, \omega_3) \models \beta$
   (b) for every $\omega_2 \in W$ such that $R\omega_1\omega_2$, $(M, \omega_2) \not\models \alpha$

**Definition 2.4 (Validity)** Let $M = \langle W, R, v\rangle$ be a BCL model and let $\alpha$ be a BCL wff. $\alpha$ is *valid in the model* $M$, written $M \models \alpha$, if for every possible world $\omega$ of $W$, $M, \omega \models \alpha$ (i.e. $\alpha$ is true at every possible world of the model). Moreover, $\alpha$ is *valid*, written $\models \alpha$, if $M \models \alpha$ for every model $M$.

The propositional conditional logic BCL, referred to in the literature as $CT4$ or $C4$ [7, 32], is uniquely defined by the following axiomatisation.

**Definition 2.5 (BCL Axiomatisation)** The propositional conditional logic BCL is the set of formulae that includes, for any arbitrary wffs $\alpha$ and $\beta$, propositional tautologies, all formulae of the form **K**, **T**, **4**, **C** given below, and that is closed under the inference rules **Necessity**, **MP** and **Subs**:

| | |
|---|---|
| **K** | $\Box(\alpha \supset \beta) \supset (\Box\alpha \supset \Box\beta)$ |
| **T** | $\Box\alpha \supset \alpha$ |
| **4** | $\Box\alpha \supset \Box\Box\alpha$ |
| **C** | $(\alpha > \beta) \equiv \Box(\alpha \supset \Diamond(\alpha \wedge \Box(\alpha \supset \beta)))$ |
| **Necessity** | from $\alpha$ infer $\Box\alpha$ |
| **MP** | From $\alpha \supset \beta$ and $\alpha$ infer $\beta$ |
| **Subs** | From $\alpha$ infer $\alpha'$, where $\alpha'$ is a substitution instance of $\alpha$ |

The above BCL logic is also known as the normal conditional logic $CT4$. This is the minimal logic for expressing normality. The formulae given in Table 1 are also well known theorems derivable in BCL using the above axiomatisation. In Section 5.2 we show that these theorems are also provable in our LDCL system.

Table 1. Other theorems of BCL

| | |
|---|---|
| $\alpha > \alpha$ | **ID** |
| $((\alpha > \beta) \wedge (\alpha > \gamma)) \supset (\alpha > (\beta \wedge \gamma))$ | **And** |
| $(\alpha > \beta) \supset (((\alpha \wedge \beta) > \gamma) \supset (\alpha > \gamma))$ | **RT** |
| $((\alpha > \gamma) \wedge (\beta > \gamma)) \supset ((\alpha \vee \beta) > \gamma)$ | **Or** |
| $\Box(\beta \supset \gamma) \supset ((\alpha > \beta) \supset (\alpha > \gamma))$ | **RCM** |
| $((\alpha > \beta) \wedge (\alpha > \gamma)) \supset (\alpha \wedge \beta > \gamma)$ | **CM** |

Note that the rational monotony property (RM), expressed by the formula $((\alpha > \gamma) \wedge (\alpha \wedge \beta \not> \gamma)) \supset (\alpha > \neg\beta)$, and the property $CV$ given by $(\alpha \not> \beta) \supset ((\alpha > \gamma) \supset$

$(\alpha \wedge \neg\beta \not> \gamma))$, are not theorems of BCL, but are valid in some of its extensions, as we explain later on[2].

## 3 Defining a Labelled Deductive Conditional Logic

In this section a Labelled Deductive Conditional Logic system for the conditional logic of normality is described formally. Basic definitions of the LDCL language and syntax are given together with the notion of a configuration – a LDCL's equivalent to a conditional theory.

### 3.1 Languages and Algebra

A *propositional LDCL language* is defined as an ordered pair $\langle \mathcal{L}_L, \mathcal{L}_C \rangle$, where $\mathcal{L}_L$ is a *labelling language* and $\mathcal{L}_C$ is a *propositional BCL language.* As in BCL, $\mathcal{L}_C$ includes both modal and conditional operators and its formal definition is given in Definition 2.1. The labelling language $\mathcal{L}_L$ is a binary fragment of a first-order language defined as follows.

**Definition 3.1 (Labelling Language)** A labelling language $\mathcal{L}_L$ is a first-order language composed of a countable set of constant symbols $W = \{\omega_0, \omega_1, \omega_2, ...\}$, a countable set of variables $\mathcal{V} = \{x, y, z, ...\}$, a binary predicate $R$, the set of connectives $\{\neg, \wedge, \vee, \equiv, \supset\}$ and the quantifiers $\forall, \exists$.

Constants and variables of $\mathcal{L}_L$ denote possible worlds and the binary predicate $R$ formalises the normality relation between possible worlds. For an arbitrary world $\omega$, $R$-literals of the form $R\omega, \omega'$ provides a declarative representation of the worlds $\omega'$ that are at least as normal as $\omega$. The above language allows Kripke semantic notions of possible world structures to be expressed syntactically. Logical information is instead expressed in the propositional conditional language $\mathcal{L}_C$.

For proof-theoretic purposes, the labelling language is extended with additional sets of terms, which will be used only in derivations. Specifically, four sets of "Skolem" function symbols are defined. The resulting language is called *semi-extended labelling language.*

**Definition 3.2 (Semi-extended Labelling Language)** Let $\mathcal{L}_L$ be a labelling language and $\mathcal{L}_C$ be a BCL language. Let $\alpha_1, \alpha_2, ..., \alpha_n$ be a canonically ordered set of the wffs of $\mathcal{L}_C$. The *semi-extended labelling language* $ext(\mathcal{L}_L, \mathcal{L}_C)$ is defined as the language $\mathcal{L}_L$ extended with four sets of unary function symbols $\{f_{\alpha_1}, ..., f_{\alpha_n}, ...\}$, $\{g_{\alpha_1}, ..., g_{\alpha_n}, ...\}$, $\{h_{f_{\alpha_i, \alpha_j}} |$ for all $i, j \geq 1\}$ and $\{h_{g_{\alpha_i, \alpha_j}} |$ for all $i, j \geq 1\}$.

All ground terms of the semi-extended labelling language, also called *labels*, refer to possible worlds. In particular, the terms constructed by using the function symbols of $ext(\mathcal{L}_L, \mathcal{L}_C)$ play particular roles. For each possible world $\omega$ and conditional formula $\alpha$, $f_\alpha(\omega)$ names a *particular* world specifically associated with $\alpha$. Such terms will be

[2] $\alpha \not> \beta$ abbreviates $\neg(\alpha > \beta)$ [7]

used whenever Kripke semantic notions of the form "there exists a possible world ..." need to be formalised. In contrast, terms of the form $g_\alpha(\omega)$ can be thought of as referring to any *arbitrary* world specifically associated with $\alpha$. These terms will be used whenever Kripke semantic notions of the form "for all possible worlds ..." need to be expressed. On the other hand, for each pair of formulae $\alpha$ and $\beta$ and possible world $\omega$, the function symbols $h_{f_{\alpha,\beta}}$ and $h_{g_{\alpha,\beta}}$ refer to a *particular normal* world and an *arbitrary normal* world specifically associated with the conditional $\alpha > \beta$. Intuitively, these two types of function symbols could be thought of as representing a particular composition of possible worlds, named with ground terms using only $f$ and $g$ function symbols, that corresponds to the "unfolded" modal formula equivalent to the conditional formula $\alpha > \beta$.

Different properties of the normality relation $R$ are syntactically formalised in a LDCL system by a first-order axiomatisation called *labelling algebra* written in the language $ext(\mathcal{L}_L, \mathcal{L}_C)$ and denoted by $\mathcal{A}$. We define here the labelling algebra that captures the underlying semantic properties of BCL logic.

**Definition 3.3 (Labelling Algebra)** A labelling algebra $\mathcal{A}$ is a first-order theory written in the language $ext(\mathcal{L}_L, \mathcal{L}_C)$ given by the following axioms:

$$\forall x.Rxx \qquad (\mathbf{T})$$

$$\forall x \forall y \forall z.(Rxy \wedge Ryz) \supset Rxz \qquad (\mathbf{4})$$

As shown in Section 5.2, the above labelling algebra enables our LDCL system to fully capture the existing BCL logic. Different extensions of this logic can also be expressed in our LDCL system by appropriately extending the labelling algebra $\mathcal{A}$ with additional properties on the normal relation $R$. For instance, to capture the logic $CT4D$ (in which RM and CV are valid), our labelling language will have to be extended with the inclusion of a new unary function symbol *succ* and our labelling algebra $\mathcal{A}$ will have to be extended with the seriality axiom $\forall x.Rxsucc(x)$. Some of other relevant additional properties for the normal relation $R$ are listed in Table 2:

TABLE 2. Additional axioms for the labelling algebra $\mathcal{A}$

| | |
|---|---|
| $\forall x \forall y.Rxy \supset Ryx$ | *Symmetry* |
| $\forall x \forall y \forall z.Rxy \wedge Rxz \supset Ryz$ | *Euclidianity* |
| $\forall x \forall y.Rxy \vee Ryx$ | *Connectedness* |

## *3.2 Syntax*

The LDCL language facilitates the formalisation of two types of information, (i) what holds at particular possible worlds and (ii) which worlds are in relation with each other and which are not. Two different syntactic entities are defined to capture these two aspects of the language, called respectively *declarative units* and *R-literals.* A declarative unit is a pair separated by colon of the form *conditional formula:label*, expressing that a conditional formula is true at a possible world. In a very general sense, the symbol ":" between the two components can be regarded as a sort of "Holds" predicate. (This interpretation will be reflected in the semantics of our

LDCL.) The label component is a ground term of the language $ext(\mathcal{L}_L, \mathcal{L}_C)$. This is the only syntactic entity which combines the two entities of the LDCL language, and it is formally defined as follows.

**Definition 3.4 (Declarative Unit)** Given the $LDCL$ language $\langle \mathcal{L}_L, \mathcal{L}_C \rangle$, a *declarative unit* is a pair $\alpha : \omega$ where $\alpha$ is a wff of $\mathcal{L}_C$ and $\omega$ is a ground term of $ext(\mathcal{L}_L, \mathcal{L}_C)$.

An $R$-literal is any ground literal in the language $ext(\mathcal{L}_L, \mathcal{L}_C)$ of the form $R\omega_1\omega_2$ or $\neg R\omega_1\omega_2$, where $\omega_1$ and $\omega_2$ are labels, expressing the fact that $\omega_2$ is, or is not, an accessible world that is at least as normal as $\omega_1$. Examples of $R$-literals are $R\omega_1\omega_2$ and $R\omega_1 g_{\Box\alpha}(\omega_2)$ and $R\omega_0 f_p(\omega_2)$. To distinguish an R-literal from its opposite in sign, the notion of the conjugate of an $R$-literal is also introduced.

**Definition 3.5 ($R$-literal)** Given the $LDCL$ language $\langle \mathcal{L}_L, \mathcal{L}_C \rangle$, an $R$-literal is a literal of the form $R\omega_1\omega_2$ or $\neg R\omega_1\omega_2$, where $\omega_1, \omega_2$ are ground terms of $ext(\mathcal{L}_L, \mathcal{L}_C)$. Let $\mathbb{R}$ be an $R$-literal. The conjugate of $\mathbb{R}$, written $\overline{\mathbb{R}}$, is defined as $\neg R\omega_1\omega_2$ if $\mathbb{R} = R\omega_1\omega_2$ and $R\omega_1\omega_2$ if $\mathbb{R} = \neg R\omega_1\omega_2$.

The syntax of a LDCL system allows arbitrary sets of conditional formulae to be associated with (different) labels, describing not only one initial set of local assumptions (as in the standard conditional logic) but allowing for several (distinct) local initial conditional theories to be specified. With the addition of $R$-literals, these local theories can be stated to be either independent (using negative $R$-literals) or interacting with each other (using positive $R$-literals). This yields a definition of a conditional labelled deductive theory more general than the traditional notion of a conditional theory. A conditional labelled deductive theory, called a *configuration*, 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 conditional theories associated with that label, whereas declarative units with different labels express conditional formulae belonging to (possibly) different local worlds. The first set (i), consisting of $R$-literals, is called a *diagram* and it provides the "structural" information about a conditional labelled deductive theory. For instance, the set $\{R\omega_1\omega_2, R\omega_2\omega_3, \neg R\omega_1\omega_3\}$ is a diagram. Local conditional theories can be assigned to each node of the diagram by adding appropriate declarative units. This is formally defined below.

**Definition 3.6 (Configuration)** Given an $LDCL$ language $\langle \mathcal{L}_L, \mathcal{L}_C \rangle$, a *configuration* is a pair $\langle \mathcal{D}, \mathbf{f} \rangle$ where $\mathcal{D}$ is a (possibly empty) set of $R$-literals in $ext(\mathcal{L}_C, \mathcal{L}_L)$ called *diagram*, and $\mathbf{f}$ is a function from the set of ground terms of $ext(\mathcal{L}_C, \mathcal{L}_L)$ to the set $\wp(wff(\mathcal{L}_C))$ of sets of wffs of $\mathcal{L}_C$.

The function $\mathbf{f}$ is a total function which assigns an empty set to the labels for which there is no information and a non-empty local conditional theory to the other labels. For a given configuration $C = \langle \mathcal{D}, \mathbf{f} \rangle$ an $R$-literal $\mathbb{R}$ is said to be *a member of* that configuration (written $\mathbb{R} \in C$) if $\mathbb{R} \in \mathcal{D}$. A declarative unit $\alpha : \omega$ is said to be *a member of* $C$, written $\alpha : \omega \in C$, if $\alpha \in \mathbf{f}(\omega)$. As mentioned before, function symbols of the language $ext(\mathcal{L}_L, \mathcal{L}_C)$ have been introduced for proof-theoretical reasons. Therefore any user specified conditional labelled deductive theory (or *initial* configuration) will usually (initially) contain only constant symbols of $ext(\mathcal{L}_L, \mathcal{L}_C)$ as labels, whereas configurations containing general ground terms of $ext(\mathcal{L}_L, \mathcal{L}_C)$ will mainly appear in proof-derivations as inferred configurations. This will become clearer in the next section where the description of a natural deduction proof system for our LDCL system is given.

We can now give the definition of an LDCL system.

**Definition 3.7 (Propositional LDCL System)** Given a $LDCL$ language $\langle \mathcal{L}_L, \mathcal{L}_C \rangle$, a *propositional LDCL system* is a tuple $\langle \langle \mathcal{L}_L, \mathcal{L}_C \rangle, \mathcal{A}, \mathcal{I} \rangle$ where $\mathcal{A}$ is a labelling algebra and $\mathcal{I}$ is a set of inference rules which generate configurations from other configurations.

From the above definition it is clear that, given a set of rules $\mathcal{I}$, different propositional LDCL systems can be obtained by considering different labelling algebrae. For the remaining of this paper, we will denote with LDCL the system obtained by the labelling algebra $\mathcal{A}$ given in Definition 3.3 and corresponding to Boutilier's conditional logic $CT4$.

To give a full definition of our LDCL system, we still need to specify the set of inference rules. This is done in the next section.

## 4 A Labelled Natural Deduction System for Conditional Logics of Normality

As illustrated above, a propositional LDCL language is a generalisation of standard conditional logic formalisms, – its syntax facilitates the representation of (possibly singleton) structures of local conditional theories. This main characteristic is also preserved within the LDCL proof theory in the sense that inference rules and the derivability relation are defined *between configurations*. This differs from standard existing conditional logic proof systems for which a derivability relation is either defined axiomatically, or defined between a theory (or a set of formulae) and a single formula. In a propositional LDCL, the deductive process describes how configurations can "evolve" by reasoning within or between local conditional theories or by reasoning about diagrams. An inference rule of a propositional LDCL can be generally defined as follows.

**Definition 4.1 (Inference Rule)** An *inference rule* $\Im$ is a set of pairs of configurations where each pair is written $C \rightsquigarrow C^I$. If $C \rightsquigarrow C^I \in \Im$ then $C$ is said to be an *the antecedent configuration* of $\Im$ and $C^I$ the *inferred configuration* of $\Im$ with respect to $C$. It is also said that $\Im$ *infers* $C^I$ from $C$.

This definition of inference rule was first proposed in [43]. Such inference rules have an advantage over Prawitz's rules (in which there is a distinction between inference rules and improper deduction rules) in the sense that this definition holds also for inference rules that require sub-derivations as their antecedents.

**Definition 4.2 (Proof)** Given a $LDCL$ system $\langle \langle \mathcal{L}_L, \mathcal{L}_C \rangle, \mathcal{A}, \mathcal{I} \rangle$, a *proof* is a pair $(\mathcal{P}, \mu)$ where $\mathcal{P}$ is a sequence of configurations $\langle C_0, ...C_n \rangle$ with $n > 0$, and $\mu$ is a mapping from the set $\{0,\ldots,n-1\}$ to $\mathcal{I}$ such that for each $i$, $0 \le i < n$, $\mu(i)$ infers $C_{i+1}$ from $C_i$, denoted by $C_i \rightsquigarrow C_{i+1} \in \mu(i)$.

**Definition 4.3** Let $C$ and $C^{'}$ be two configurations. A configuration $C^{'}$ is *derivable* from $C$ in a $LDCL$ system (denoted by $C \vdash_{LDCL} C^{'}$) if there is a proof $\langle\{C, ..., C^{'}\}, \mu\rangle$.

Recall that a configuration is a pair of sets, and usual set operations can be defined over them. Thus, we have the following notations. For a configuration $C = \langle\mathcal{D}, \mathbf{f}\rangle$, declarative unit $\alpha : \omega$, and R-literal $\mathbb{R}$, the configuration $C + [\alpha : \omega]$ represents a configuration $C' = \langle\mathcal{D}, \mathbf{f'}\rangle$ where $\mathbf{f'}(w) = \mathbf{f}(\omega) \cup \{\alpha\}$ and $\mathbf{f'}(\omega') = \mathbf{f}(\omega)$ for any $\omega' \neq \omega$ in the language $ext(\mathcal{L}_L, \mathcal{L}_C)$. $C + [\mathbb{R}]$ represents the configuration $C' = \langle\mathcal{D}', \mathbf{f}\rangle$ where $\mathcal{D}' = \mathcal{D} \cup \{\mathbb{R}\}$. Given two configurations $C, C'$, a declarative unit or R-literal $\delta$ we write $C \vdash_{LDCL} \delta$ if there exists a configuration $C'$ such that $C \vdash_{LDCL} C'$, and $\delta \in C'$. Moreover, we say that $C \vdash_{LDCL} \bot : \omega$ if $C \vdash_{LDCL} \gamma \wedge \neg\gamma : \omega$.

As described in Definition 4.1, inference rules of a propositional LDCL system are generally applied to configurations to infer "new" configurations. The main question is, of course, how an inferred configuration is generated. Given an antecedent configuration $\mathcal{C}$, three types of reasoning step can occur. Those of the first type are *local* in the sense that they occur within any particular local conditional theory included in $\mathcal{C}$, respecting standard notions of inference for classical connectives. Those of the second type are *between-theories* and concern the interaction between different local theories in $\mathcal{C}$, according to the modal and/or conditional information (wffs) incorporated in the declarative units that belong to $\mathcal{C}$. In the first two cases inferred configurations are mainly "logical expansions" of (i.e. additions of declarative units to) the antecedent configurations. Those of the third type are *structural*, as they are 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. Four different classes of inference rule have therefore been defined. The first includes classical elimination and introduction rules and are the local type of rules. The second and third classes include elimination and introduction rules for modal and conditional operators, respectively, and constitute the between-theories type of rules, whereas the last class of rules includes rules for manipulating *R*-literals and are therefore structural type of rules. Classical, modal and structural rules are schematically represented in Tables 3–5, where the symbol $\bot$ is used as an abbreviation for any formula of the form $\gamma \wedge \neg\gamma$. Formal descriptions of the conditional rules are instead given in Definitions 4.4 and 4.5.

Some remarks are essential to clarify the informal notation used in both the schematic representation and formal descriptions of the LDCL inference rules. For any configuration $C$, the informal notation $C\langle\alpha : \omega\rangle$ (respectively $C\langle\mathbb{R}\rangle$) denotes that C includes a declarative unit $\alpha : \omega$ (respectively R-literal $\mathbb{R}$). Declarative units and R-literals contained in square brackets (see e.g. the $\vee E$ rule) are assumptions introduced within a derivation that are subsequently discharged. The notation $C_I\langle\delta\rangle$ represents that the inferred configuration $C_I$ is the antecedent configuration $C$ extended with the declarative unit or *R*-literal $\delta$, whereas the notation $C_S$ is used to refer to configurations inferred within subderivations. For instance, in the $(\supset I)$ rule, $C_S$ is the configuration derived in the subderivation after adding the assumption $[\alpha : \omega]$ to the antecedent configuration of the rule in order to obtain the declarative unit $\beta : \omega$.

## 4.1 Classical Rules

The classical rules for classical connectives are schematically described in Table 3. Note that, for the $(\vee I)$ rule, the symmetric rule in which the inferred configuration includes $\beta \vee \alpha : \omega$ instead of $\alpha \vee \beta : \omega$ is implicitly assumed.

TABLE 3. Rules for Classical Connectives

$$\frac{C\langle \alpha \vee \beta : \omega \rangle \quad \begin{array}{c} C\langle [\alpha : \omega] \rangle \\ \vdots \\ C_S\langle \gamma : \omega \rangle \end{array} \quad \begin{array}{c} C\langle [\beta : \omega] \rangle \\ \vdots \\ C_{S'}\langle \gamma : \omega \rangle \end{array}}{C_I\langle \gamma : \omega \rangle} \vee E \qquad \frac{C\langle \alpha : \omega \rangle}{C_I\langle \alpha \vee \beta : \omega \rangle} \vee I$$

$$\frac{C\langle \alpha \wedge \beta : \omega \rangle}{C_I\langle \alpha : \omega, \beta : \omega \rangle} \wedge E \qquad \frac{C\langle \alpha : \omega, \beta : \omega \rangle}{C_I\langle \alpha \wedge \beta : \omega \rangle} \wedge I$$

$$\frac{C\langle \alpha \supset \beta : \omega, \alpha : \omega \rangle}{C_I\langle \beta : \omega \rangle} \supset E \qquad \frac{\begin{array}{c} C\langle [\alpha : \omega] \rangle \\ \vdots \\ C_S\langle \beta : \omega \rangle \end{array}}{C_I\langle \alpha \supset \beta : \omega \rangle} \supset I$$

$$\frac{C\langle \neg\neg\alpha : \omega \rangle}{C_I\langle \alpha : \omega \rangle} \neg E \qquad \frac{\begin{array}{c} C\langle \alpha : \omega \rangle \\ \vdots \\ C_S\langle \bot : \omega' \rangle \end{array}}{C_I\langle \neg\alpha : \omega \rangle} \neg I$$

Each of these rules has the effect of expanding the antecedent configuration with a new declarative unit. With the exception of the $(\neg I)$ rule, both the added declarative unit and the declarative unit(s) involved in the premise refer to the same label. This shows that the reasoning allowed by these rules takes place entirely within the scope of the local theory under consideration. This characteristic is semantically motivated by the fact that the classical fragment of a conditional logic behaves like a classical logic "locally" associated with any particular normal world. But for the $(\neg I)$ rule, this is not always the case. According to the classical interpretation of the $\neg$ connective, the negation of a formula can in general be proved by showing that the assumption of its positive form leads to a contradiction. In conditional reasoning, contradictions can arise in normal worlds which are different from the current actual world. The sub-derivation may involve reasoning about worlds different from the current world in which contradictions may arise. Therefore, in order to capture these cases a different meta-symbol $\omega'$ is used, which may or may not be equal to $\omega$. Note that such world

$\omega'$ does not necessary need to be accessible from the current world $\omega$. This is because, given the more general type of theories (configurations), inconsistency can also arise in local actual worlds which are different and not in relation with the normal world under consideration. In this case, $(\neg I)$ reflects the classical principle "any formula can be inferred from a contradiction". Later it is shown that the same principle holds for inconsistencies caused by $R$-literals.

## *4.2 Modal and Conditional Rules*

The set of natural deduction rules concerning the modal operators $\Box$ and $\Diamond$ of an LDCL language is given in Table 4. Each of these rules describes how sets of information belonging to different worlds in relation with each other can interact.

TABLE 4. Rules for modality operators

$$\frac{\begin{array}{c} C\langle[R(\omega, g_\alpha(\omega))]\rangle \\ \vdots \\ C_S\langle \alpha : g_\alpha(\omega)\rangle \end{array}}{C_I\langle \Box\alpha : \omega\rangle}\Box I \qquad \frac{C\langle \Box\alpha : \omega_1, R\omega_1\omega_2\rangle}{C_I\langle \alpha : \omega_2\rangle}\Box E$$

$$\frac{C\langle \Diamond\alpha : \omega\rangle}{C_I\langle \alpha : f_\alpha(\omega), R\omega f_\alpha(\omega)\rangle}\Diamond E \qquad \frac{C\langle \alpha : \omega_2, R\omega_1\omega_2\rangle}{C_I\langle \Diamond\alpha : \omega_1\rangle}\Diamond I$$

The $(\Diamond E)$ rule can be seen (informally) as a "skolemization" of the existential quantifier over possible worlds which is semantically implied by the formula $\Diamond\alpha$ in the premise. The term $f_\alpha(\omega)$ defines a particular possible world uniquely associated with the formula $\alpha$, and inferred to be accessible from the possible world $\omega$ (i.e. $R\omega f_\alpha(\omega)$). It is clear from the definition that this rule has the effect of expanding both the two components (diagram and set of declarative units) of the antecedent configuration. In the $(\Box I)$ rule, the temporary assumption should be read as "given an arbitrary accessible world $g_\alpha(\omega)$". Although, from a syntactical point of view, the use of a term is often seen as a way of naming particular objects (possible worlds), in this case it is adopted to refer to an arbitrary possible world. This is to ensure that labels are always ground terms. The role of the function symbol $g_\alpha$ will become clearer in Section 5, where the semantics for a LDCL system is given.

The introduction and elimination rules for the conditional operator are formally defined in the following definitions. Intuitively, to prove a conditional declarative unit $\alpha > \beta : \omega$ it is necessary to show that there exists a sub-derivation, which assumes $\alpha$ to be true at an arbitrary world $h_{g_{\alpha,\beta}}(\omega)$ at least as normal as (and accessible from) $\omega$, and to show that $\alpha$ is also true at a world $\omega'$ at least as normal as (and accessible from) $h_{g_{\alpha,\beta}}(\omega)$ and that $\alpha \supset \beta$ is true at any arbitrary world at least as normal as (and accessible from) $h_{g_{\alpha,\beta}}(\omega)$.

**Definition 4.4 (Introduction rule for $>$)** Let $C$ be an arbitrary configuration, let $\omega$ be an arbitrary label and let $\alpha$ and $\beta$ be two well-formed formulae. We say that $C \rightsquigarrow C + [\alpha > \beta : \omega]$ is a member of the inference rule $(> I)$ if $C + [\alpha : h_{g_{\alpha,\beta}}(\omega), R\omega h_{g_{\alpha,\beta}}(\omega)] \vdash_{LDCL} \{\alpha : \omega', \Box(\alpha \supset \beta) : \omega', R h_{g_{\alpha,\beta}}(\omega)\omega'\}$, for some label $\omega'$ The diagrammatic representation is as follows.

$$\frac{\begin{array}{c} C\langle[\alpha : h_{g_{\alpha,\beta}}(\omega), R\omega h_{g_{\alpha,\beta}}(\omega)]\rangle \\ \vdots \\ C_S\langle \alpha : \omega', \Box(\alpha \supset \beta) : \omega', R h_{g_{\alpha,\beta}}(\omega)\omega'\rangle \end{array}}{C_I\langle \alpha > \beta : \omega\rangle} > I$$

**Definition 4.5 (Elimination rule for >)** Let $C$ be an arbitrary configuration, let $\omega$ and $\omega'$ be two arbitrary labels and let $\alpha$ and $\beta$ be two well-formed formulae. We say that $C \rightsquigarrow C + [R\omega' h_{f_{\alpha,\beta}}(\omega'), \alpha : h_{f_{\alpha,\beta}}(\omega'), \Box(\alpha \supset \beta) : h_{f_{\alpha,\beta}}(\omega')]$ is a member of the inference rule $(> E)$ if $\{\alpha > \beta : \omega, R\omega\omega', \alpha : \omega'\} \subset C$. The diagrammatic representation of the rule is as follows.

$$\frac{C\langle \alpha > \beta : \omega, R\omega\omega', \alpha : \omega'\rangle}{C_I\langle R\omega' h_{f_{\alpha,\beta}}(\omega'), \alpha : h_{f_{\alpha,\beta}}(\omega'), \Box(\alpha \supset \beta) : h_{f_{\alpha,\beta}}(\omega')\rangle} > E$$

The definition of the elimination rule for the conditional connective clearly reflects Boutilier's semantic definition of normality, as is also the case for the introduction $(> I)$ rule. From equation 2.1, it is easy to see that $h_{f_{\alpha,\beta}}$ also represents $f_{\alpha \wedge \Box(\alpha \supset \beta)}$, in the same way as the function symbol $h_{g_{\alpha,\beta}}$ represents $g_{\alpha \supset \Diamond(\alpha \wedge \Box(\alpha \supset \beta))}$ in the $(> I)$ rule. Therefore, both $(> I)$ and $(> E)$ rules abstract much of the machinery involved in a "pure" modal logic deductive system, and allows the deduction process to concentrate on the intuition behind the semantics of the conditional connective. The examples given in Section 4.4 illustrate how the conditional rules operate and how the use of the conditional operator instead of a full translation into modal logic facilitates simpler proofs.

## *4.3 Structural Rules*

To allow reasoning about arbitrary configurations and to capture the different LDCL systems, a third set of inference rules needs to be included as part of a propositional LDCL system. These rules facilitate reasoning about the diagram of a configuration, using the particular labelling algebra $\mathcal{A}$ of the LDCL system under consideration, and enable the inference of $R$-literals and declarative units that are not implied by the modal and conditional rules. These structural rules are schematically represented in Table 5. In Section 3 we have mentioned that different labelling algebrae define different propositional LDCL systems. Proof-theoretically these differences are imposed by the $D_{Exp}$ rule. This rule facilitates the inference of new $R$-literals according to the properties of the accessibility relation given by a particular labelling algebra $\mathcal{A}$. So for example, if the labelling algebra $\mathcal{A}$ includes the seriality axiom for the accessibility relation $R$, then the above rule would allow the inference of $R$-literals of the form $R(\omega, succ(\omega))$, for any label $\omega$, thus embedding the seriality property of the accessibility relation in the derivation process and allowing the axiom **D** $\Box(\Box\alpha \supset \beta) \vee \Box(\Box\beta \supset \alpha)) : \omega$ to be proved at any arbitrary label $\omega$, hence capturing the LDCL system CT4D. By using the $D_{Exp}$ rule and the different labelling

TABLE 5. Structural Rules

| | |
|---|---|
| $\dfrac{C}{C_I\langle\mathbb{R}\rangle} D_{Exp} \quad if\ D,\mathcal{A}\vdash_{FOL}\mathbb{R}$ | $\dfrac{C}{C_I} D_{Red},\ where\ C_I \subseteq C$ |
| $\dfrac{\begin{array}{c} C\langle[\sim \mathbb{R}]\rangle \\ \vdots \\ C_S\langle\bot : \omega'\rangle \end{array}}{C_I\langle\mathbb{R}\rangle} R_{Intr}$ | $\dfrac{C\langle\mathbb{R}, \sim \mathbb{R}\rangle}{C_I\langle\alpha : \omega\rangle} \bot_E$ |

algebras, the sets of modal and conditional rules remain unchanged for the whole family of normal conditional logic LDCL, making the system a *uniform* natural deduction system.

The $(\bot_E)$ and $(R_{Intr})$ rules formalise additional forms of interaction between $R$-literals and declarative units. In a LDCL theory contradictory assumptions can be either an $R$-literal and its conjugate, or a declarative unit and its negation[3]. Whereas the latter are captured by the $\neg I$ rule (see previous discussion), the former are identified by the $(R_{Intr})$ rule, which again reflects the classical principle of any formula being derived from a contradiction. The $R_{Intr}$ rules is a kind of $\neg I$ rule for $R$-literals. However, the interesting feature of this rule is that it facilitates a second form of interaction between declarative units and $R$-literals. Specifically, $R$–literals are derived whenever a logical inconsistency arises within a configuration. The standard classical case of deriving an $R$-literal whenever an inconsistency arises among $R$-literals is instead already covered by the $D_{Exp}$ rule.

### *4.4 Example Derivations*

We now show some example derivations using the graphical representation of the natural deduction rules defined above. Note that in these derivations the symbols $h_1, h_2, h_3, h_4$ are abbreviations for labels using the Skolem symbols $h_{g_{\alpha,\beta}}$ and $h_{f_{\alpha,\beta}}$ introduced by the rules for the conditional. In the first example derivation (i.e. the proof of $((\alpha > \beta) \wedge (\alpha > \gamma)) \supset (\alpha > (\beta \wedge \gamma)) : \omega_0)$, $h_1$ stands for $h_{g_{\alpha,(\beta\wedge\gamma)}}(\omega_0)$, $h_2$ for $h_{f_{\alpha,\beta}}(h_1)$, $h_3$ for $h_{f_{\alpha,\gamma}}(h_2)$ and $g_1$ stands for $g_{\alpha\supset(\beta\wedge\gamma)}(h_3)$. In what follows, we don't make distinctions between configurations $C_S$, $C_I$ and intermediate configurations. They are all sequentially indexed with a natural number. Each single derivation step is labelled with the name of the rule and the antecedent configurations on which the rule is applied. For an introduction rule, we also labelled the step of introducing a new assumption and we write the configuration that closes the sub-derivation and where the temporary assumption is discharged.

The first example is a derivation of the axiom (**And**) (also known as $CC$ [12]). Since this is supposed to be a theorem of $CT4$, we start from an empty configuration and we try to derive a configuration which includes $((\alpha > \beta) \wedge (\alpha > \gamma)) \supset (\alpha > (\beta \wedge \gamma))$

[3] The negation of a declarative unit $\alpha : \omega$ is a declarative unit of the form $\neg\alpha : \omega$.

at the initial world $\omega_0$. The derivation of $R$-literals in $C_4$ and $C_8$ is made using the transitivity axiom of the labelling algebra.

$$C_\emptyset$$
$$C_1\langle[\alpha > \beta : \omega_0, \alpha > \gamma : \omega_0]\rangle \quad (Assump, C_{17})$$
$$C_2\langle[\alpha : h_1, R\omega_0 h_1]\rangle \quad (Assump, C_{16})$$
$$C_3\langle \alpha : h_2, \Box(\alpha \supset \beta) : h_2, Rh_1h_2\rangle \quad (> E, C_1, C_2)$$
$$C_4\langle R\omega_0 h_2\rangle \quad (D_{Exp}, C_2, C_3, Trans)$$
$$C_5\langle \alpha : h_3, \Box(\alpha \supset \gamma) : h_3, Rh_2h_3\rangle \quad (> E, C_1, C_3, C_4)$$
$$C_6\langle[Rh_3g_1]\rangle \quad (Assump, C_{14})$$
$$C_7\langle[\alpha : g_1]\rangle \quad (Assump, C_{13})$$
$$C_8\langle \alpha \supset \beta : g_1\rangle \ (\Box E, C_3, D_{Exp}, C_5, C_6, Trans)$$
$$C_9\langle \alpha \supset \gamma : g_1\rangle \ (\Box E, C_5, C_6)$$
$$C_{10}\langle \beta : g_1\rangle \ (\supset E, C_7, C_8)$$
$$C_{11}\langle \gamma : g_1\rangle \ (\supset E, C_7, C_9)$$
$$C_{12}\langle \beta \wedge \gamma : g_1\rangle \ (\wedge I, C_{10}, C_{11})$$
$$C_{13}\langle \alpha \supset (\beta \wedge \gamma) : g_1\rangle \ (\supset I, C_7 - C_{12})$$
$$C_{14}\langle \Box(\alpha \supset (\beta \wedge \gamma)) : h_3\rangle \ (\Box I, C_6 - C_{13})$$
$$C_{15}\langle \alpha : h_3, \Box(\alpha \supset (\beta \wedge \gamma)) : h_3, Rh_1h_3\rangle \ (C_5, C_{14}, D_{Exp}, C_3, C_5, Trans)$$
$$C_{16}\langle \alpha > (\beta \wedge \gamma) : \omega_0\rangle \ (> I, C_2 - C_{15})$$
$$C_{17}\langle((\alpha > \beta) \wedge (\alpha > \gamma)) \supset (\alpha > (\beta \wedge \gamma)) : \omega_0\rangle \ (\supset I, C_1 - C_{16})$$

The next example shows the entailment $\Box\alpha : \omega_0 \vdash_{LDCL} \neg\alpha > \alpha : \omega_0$. In this derivation, the reflexivity property of the labelling algebra is used to derive $C_8$, $h_1$ stays for the label $h_{g_{\neg\alpha,\alpha}}(\omega_0)$, whereas $g_1$ abbreviates $g_{\neg\alpha\supset\alpha}(h_1)$.

$$C_0\langle \Box\alpha : \omega_0\rangle \ (InitialData)$$
$$C_1\langle[\neg\alpha : h_1, R\omega_0 h_1]\rangle \ (Assump, C_9)$$
$$C_2\langle[Rh_1g_1]\rangle \ (Assump, C_7)$$
$$C_3\langle[\neg\alpha : g_1]\rangle \ (Assump, C_6)$$
$$C_4\langle R\omega_0 g_1\rangle \ (D_{Exp}, Trans, C_1, C_2)$$
$$C_5\langle \alpha : g_1\rangle \ (\Box E, C_0, C_4)$$
$$C_6\langle \neg\alpha \supset \alpha : g_1\rangle \ (\supset I, C_3 - C_5)$$
$$C_7\langle \Box(\neg\alpha \supset \alpha) : h_1\rangle \ (\Box I, C_2 - C_6)$$
$$C_8\langle \neg\alpha : h_1, \Box(\neg\alpha \supset \alpha) : h_1, Rh_1h_1\rangle \ (C_1, C_7, D_{Exp}, Refl)$$
$$C_9\langle \neg\alpha > \alpha : \omega_0\rangle \ (> I, C_1 - C_8)$$

The converse entailment $\neg\alpha > \alpha : \omega \vdash_{LDCL} \Box\alpha : \omega$ is proved below, where the labels $g_1$ and $h_1$ abbreviate, respectively, $g_\alpha(\omega_0)$ and $h_{f_{\neg\alpha,\alpha}}(g_1)$.

$$C_0\langle \neg\alpha > \alpha : \omega_0\rangle \ (InitialData)$$
$$C_1\langle[R\omega_0 g_1]\rangle \ (Assump, C_8)$$
$$C_2\langle[\neg\alpha : g_1]\rangle \ (Assump, C_7)$$
$$C_3\langle Rg_1h_1, \neg\alpha : h_1, \Box(\neg\alpha \supset \alpha) : h_1\rangle \ (> E, C_0, C_1, C_2)$$
$$C_4\langle \neg\alpha \supset \alpha : h_1\rangle \ (D_{Exp}, Refl, \Box E, C_3)$$
$$C_5\langle \alpha : h_1\rangle \ (\supset E, C_4, C_2)$$
$$C_6\langle \bot : h_1\rangle \ (\wedge I, C_3, C_5)$$
$$C_7\langle \alpha : g_1\rangle \ (\supset I, C_2 - C_6)$$
$$C_8\langle \Box\alpha : \omega_0\rangle \ (\Box I, C_1 - C_7)$$

Finally, we show here the axiom of the conditional logics of normality called *Weak Modus Ponens* [7], given by the formula $(\alpha \wedge (\alpha > \beta)) > \beta$. The derivation of this axiom illustrates how the LDCL system facilitates simpler derivations than a system where the conditions are fully translated into modal formulae. This simplification is allowed by the use of specific conditional rules. If we were not using the rules for the conditional connective, we would have translated the Weak Modus Ponens axiom into the following modal formula and proved it using only modal operators.

$$\alpha \wedge (\alpha > \beta) > \beta \equiv \Box(\alpha \wedge [\Box(\alpha \supset \Diamond(\alpha \wedge \Box(\alpha \supset \beta)))] \supset \Diamond(\alpha \wedge [\Box(\alpha \supset \Diamond(\alpha \wedge \Box(\alpha \supset \beta)))] \wedge \Box(\alpha \wedge [\Box(\alpha \supset \Diamond(\alpha \wedge \Box(\alpha \supset \beta)))] \supset \beta))).$$

In the derivation given here $h_1$, $h_2$, $h_3$, $g_1$ and $g_2$ are, respectively, abbreviations for $h_{g_{\alpha\wedge(\alpha>\beta),\beta}}(\omega_0)$, $h_{f_{\alpha,\beta}}(h_1)$, $h_{g_{\alpha,\beta}}(h_2)$, $g_{\alpha\wedge(\alpha>\beta)\supset\beta}(h_2)$ and $g_{\alpha\supset\beta}(h_3)$.

$$C_\emptyset$$
$$C_1\langle[\alpha \wedge (\alpha > \beta) : h_1, R\omega_0 h_1]\rangle \ (Assump, C_8)$$
$$C_2\langle \alpha : h_1, \alpha > \beta : h_1\rangle \ (\wedge E, C_1)$$
$$C_3\langle \alpha : h_2, \Box(\alpha \supset \beta) : h_2, Rh_1h_2\rangle \ (> E, Refl, C_2)$$
$$C_4\langle[\alpha : h_3, Rh_2h_3]\rangle \ (Assump, C_9)$$
$$C_5\langle[Rh_3g_2]\rangle \ (Assump, C_8)$$
$$C_6\langle Rh_2g_2, Rh_3h_3\rangle \ (D_{Exp}, Trans, C_4, C_5)$$
$$C_7\langle \alpha \supset \beta : g_2\rangle \ (\Box I, C_3, C_6)$$
$$C_8\langle \Box(\alpha \supset \beta) : h_3\rangle \ (\Box I, C_5 - C_7)$$
$$C_9\langle \alpha > \beta : h_2\rangle \ (> I, D_{Exp}, Refl, C_4 - C_8)$$
$$C_{10}\langle \alpha \wedge (\alpha > \beta) : h_2\rangle \ (\wedge I, C_3, C_9)$$
$$C_{11}\langle[Rh_2g_1]\rangle \ (Assump, C_{17})$$
$$C_{12}\langle[\alpha \wedge (\alpha > \beta) : g_1]\rangle \ (Assump, C_{16})$$
$$C_{13}\langle \alpha : g_1, \alpha > \beta : g_1\rangle \ (\wedge E, C_{12})$$
$$C_{14}\langle \alpha \supset \beta : g_1\rangle \ (\Box E, C_3, C_{11})$$
$$C_{15}\langle \beta : g_1\rangle \ (\supset E, C_{13}, C_{14})$$
$$C_{16}\langle \alpha \wedge (\alpha > \beta) \supset \beta : g_1\rangle \ (\supset I, C_{12} - C_{15})$$
$$C_{17}\langle \Box(\alpha \wedge (\alpha > \beta) \supset \beta) : h_2\rangle \ (\Box I, C_{11} - C_{16})$$
$$C_{18}\langle \Box(\alpha \wedge (\alpha > \beta) \supset \beta) : \wedge \alpha \wedge (\alpha > \beta) : h_2, Rh_1h_2\rangle \ (C_{17}, C_{10}, C_3)$$
$$C_{19}\langle(\alpha \wedge (\alpha > \beta)) > \beta : \omega_0\rangle \ (> I, C_1 - C_{18})$$

## 5 Semantics, Soundness and Completeness

A propositional CLDS can be considered to be a "semi-translated" approach to conditional logic – a Kripke-like accessibility relation is syntactically expressed, but without requiring the full translation of conditional formulae into modal formulae and the latter into first-order sentences. In this section, we define a translation method of a propositional CLDS into a first-order logic, and provide the semantic notion of a consequence relation, $\models_{LDCL}$, as well as the definition of a model and of satisfiability of a configuration, also in terms of classical semantics.

Declarative units $\alpha : \omega$ can be interpreted as *the formula $\alpha$ is true at the possible world $\omega$*. In what follows, such Kripke semantic notions are expressed in terms of first-order statements of the form $[\![\alpha]\!]\omega$, where $[\![\alpha]\!]$ is a unary predicate symbol. Hence the semi-extended labelling language $ext(\mathcal{L}_L, \mathcal{L}_C)$ is further expanded to $ext^+(\mathcal{L}_C, \mathcal{L}_L)$ by adding predicate symbol $[\![\alpha]\!]$ for each wff $\alpha$ of $\mathcal{L}_C$. The relationships between these predicates are constrained by a set of first-order axiom schemas which capture the satisfiability conditions (given in Definition 2.3) of each type[4] of formula $\alpha$. These axiom schemas extend the labelling algebra $\mathcal{A}$ of a propositional CLDS into a first-order theory denoted with $\mathcal{A}^+$ and called an *extended algebra*. Formal definitions are given below.

**Definition 5.1 (Extended Algebra)** Given an extended labelling language $ext^+(\mathcal{L}_C, \mathcal{L}_L)$ and a labelling algebra $\mathcal{A}$, the *extended algebra* $\mathcal{A}^+$ is the first–order theory in $ext^+(\mathcal{L}_C, \mathcal{L}_L)$ consisting of the following axiom schemas (Ax1)–(Ax10), together with the axioms of $\mathcal{A}$:

For any wffs $\alpha$ and $\beta$ of $\mathcal{L}_C$:

$$\forall x([\![\alpha \wedge \beta]\!]x \equiv ([\![\alpha]\!]x \wedge [\![\beta]\!]x)) \tag{Ax1}$$

$$\forall x([\![\neg\alpha]\!]x \equiv \neg[\![\alpha]\!]x) \tag{Ax2}$$

$$\forall x([\![\alpha \vee \beta]\!]x \equiv ([\![\alpha]\!](x) \vee [\![\beta]\!]x)) \tag{Ax3}$$

$$\forall x([\![\alpha \supset \beta]\!]x \equiv [\![\alpha]\!]x \supset [\![\beta]\!]x) \tag{Ax4}$$

$$\forall x([\![\Diamond\alpha]\!]x \supset (Rxf_\alpha(x) \wedge [\![\alpha]\!]f_\alpha(x))) \tag{Ax5}$$

$$\forall x((\exists y(Rxy \wedge [\![\alpha]\!]y) \supset [\![\Diamond\alpha]\!]x) \tag{Ax6}$$

$$\forall x((Rxg_\alpha(x) \supset [\![\alpha]\!]g_\alpha(x)) \supset [\![\Box\alpha]\!]x) \tag{Ax7}$$

$$\forall x([\![\Box\alpha]\!]x \supset (\forall y(Rxy \supset [\![\alpha]\!]y))) \tag{Ax8}$$

$$\begin{aligned}\forall x(((&Rxh_{g_{\alpha,\beta}}(x) \wedge [\![\alpha]\!]h_{g_{\alpha,\beta}}(x)) \supset \\ &\exists y(Rh_{g_{\alpha,\beta}}(x)y \wedge [\![\alpha]\!]y \wedge [\![\Box(\alpha \supset \beta)]\!]y)) \supset [\![\alpha > \beta]\!]x)\end{aligned} \tag{Ax9}$$

$$\begin{aligned}\forall x(&[\![\alpha > \beta]\!]x \supset \\ &\forall y(Rxy \wedge [\![\alpha]\!]y \supset (Ryh_{f_{\alpha,\beta}}(x) \wedge [\![\alpha]\!]h_{f_{\alpha,\beta}}(x) \wedge [\![\Box(\alpha \supset \beta)]\!]h_{f_{\alpha,\beta}}(x))))\end{aligned} \tag{Ax10}$$

The first four axiom schemas express the distributive properties of the logical connectives among the monadic predicates of $ext^+(\mathcal{L}_C, \mathcal{L}_L)$. They cover the Kripke semantic definition of satisfiability of the logical connectives $\wedge$, $\neg$, $\vee$ and $\supset$ respectively. (Ax5) forces the accessibility relation $R$ on the labels generated by the application of function symbols $f_{\alpha_i}$ of $ext^+(\mathcal{L}_C, \mathcal{L}_L)$. Axiom schemas (Ax5)–(Ax6) together cover the Kripke semantic definition of the modal operator $\Diamond$. (i.e. the statement $\forall x(\exists y(R(x, y) \wedge [\![\alpha]\!]y) \equiv [\![\Diamond\alpha]\!]x)$ can be derived from (Ax5)–(Ax6)). Analogously axiom schemas (Ax7)–(Ax8) together cover the Kripke semantic definition of the modal

[4] The type of a wff is given by the main connective of the wff itself.

operator $\Box$. (i.e. the statement $\forall x([\![\Box\alpha]\!]x \equiv \forall y(R(x,y) \supset [\![\alpha]\!]y))$ can be derived from (Ax7)–(Ax8)). Finally, axiom schemas (Ax9) and (Ax10) cover together Boutilier's semantic definition of the normal conditional operator (as given in Definition 2.3).

A translation method is defined next. It associates syntactic expressions of a propositional CLDS with sentences of the first-order language $ext^+(\mathcal{L}_C, \mathcal{L}_L)$, and hence associates theories (configurations) with first-order theories in the language $ext^+(\mathcal{L}_C, \mathcal{L}_L)$. Each declarative unit $\alpha : \omega$ is translated into the sentence $[\![\alpha]\!]\omega$, 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 $[\![\alpha]\!]\omega$ that correspond to the declarative units present in the configuration. A formal definition is given below.

**Definition 5.2 (First-order Translation of a Configuration)** Let $\mathcal{C} = \langle \mathcal{D}, \mathbf{f} \rangle$ be a configuration. The *first-order translation* of $\mathcal{C}$, written $FOT(\mathcal{C})$, is the theory written in $ext^+(\mathcal{L}_C, \mathcal{L}_L)$ and defined by $FOT(\mathcal{C}) = \mathcal{D} \cup \Delta$, where $\Delta = \{[\![\alpha]\!]\omega \mid \alpha \in \mathbf{f}(\omega)\}$.

Note that, since labels can only be ground terms of the language $ext(\mathcal{L}_C, \mathcal{L}_L)$, the first-order translation of a configuration is a set of *ground literals* of the language $ext^+(\mathcal{L}_C, \mathcal{L}_L)$. Notions of models, satisfiability and semantic entailment are given in terms of classical semantics using the following definitions.

**Definition 5.3 (Semantic Structure of a LDCL)** Given an $LDCL$ system and the associated extended algebra $\mathcal{A}^+$, $\mathcal{M}$ is a *semantic structure* of $LDCL$, if $\mathcal{M}$ is a model of $\mathcal{A}^+$.

Note that different classes of semantic structures can be obtained by considering different underlying labelling algebrae. In this paper, we will consider mainly semantic structures for the labelling algebra given in Definition 3.3.

**Definition 5.4 (Satisfiability of Declarative Units and R-literals)** Let $\alpha : \omega$ be a declarative unit. $\alpha : \omega$ *is satisfiable* (with respect to an $LDCL$ system) if there exists a semantic structure $\mathcal{M}$ such that $\mathcal{M} \models_{FOL} [\![\alpha]\!]\omega)$. In this case $\mathcal{M}$ is said to *satisfy* $\alpha : \omega$, written as $\mathcal{M} \models_{LDCL} \alpha : \omega$. Let $\mathbb{R}$ be an $R$-literal. $\mathbb{R}$ *is satisfiable* (with respect to an LDCL system) if there exists a semantic structure $\mathcal{M}$ such that $\mathcal{M} \models_{FOL} \mathbb{R}$. In this case $\mathcal{M}$ is said to *satisfy* $\mathbb{R}$, written as $\mathcal{M} \models_{LDCL} \mathbb{R}$.

**Definition 5.5 (Satisfiability of a Configuration)** Let $C$ be a configuration of a $LDCL$ system. We say that a semantic structure $\mathcal{M}$ satisfies a configuration $C$ denoted by $\mathcal{M} \models_{LDCL} C$, if for each declarative unit or $R$-literal $\delta \in C$, we have that $\mathcal{M} \models_{LDCL} \delta$.

**Definition 5.6 (Semantic Entailment)** Let $\mathcal{A}^+$ be the extended algebra of a $LDCL$ system, let $C = \langle \mathcal{D}, \mathbf{f} \rangle$ and $C^I = \langle \mathcal{D}^I, \mathbf{f}^I \rangle$ be two configurations and let $FOT(C)$ and $FOT(C^I)$ be their first-order translations. We say that $C$ *semantically entails* $C^I$ denoted by $C \models_{LDCL} C^I$ if (i) $\mathcal{A}^+ \cup FOT(C) \models_{FOL} \mathbb{R}$ for each $\mathbb{R} \in \mathcal{D}^I$ and (ii) $\mathcal{A}^+ \cup FOT(C) \models_{FOL} [\![\alpha]\!]\omega$ for each $[\![\alpha]\!]\omega \in \Delta^I$.

## 5.1 Soundness and Completeness

In the previous section, the notions of a syntactic and semantic consequence relation have been defined, so completing the description of a propositional LDCL system as a logical framework. Three important results are proved in this section, which show that (i) the two notions of consequence relation are equivalent and (ii) the propositional LDCL system fully detailed in this paper really is a generalisation of the standard Boutilier's conditional logic $CT4$. To show (i), it is proved that the derivability relation $\vdash_{LDCL}$ is *sound* and *complete* with respect to the semantic entailment relation $\models_{LDCL}$, whereas to show (ii), it is proved that Boutilier's conditional logic $CT4$ is *strictly subsumed* by the corresponding propositional LDCL system. Many of the propositions, lemmas and theorems in this section are proved *reasoning by cases*, but for space limitation only cases relevant to the conditional connective are shown here. The reader is referred to [33] for full detailed proofs.

### 5.1.1 Soundness

The soundness property states that whenever there exists a natural deduction proof of a configuration $\mathcal{C}'$ from a configuration $\mathcal{C}$ then $\mathcal{C}$ semantically entails $\mathcal{C}'$. In general this type of theorem is proved by induction on the number of inference steps of the assumed derivation. However, the natural deduction feature of using temporary new assumptions in a derivation needs particular attention, since it temporarily alters the initial theory. Proofs of soundness described in the literature often adopt a particular technique, which is that of defining for each step of the derivation the notion of a *context* – i.e. the set of all the assumptions which have been introduced and not yet discharged. However a different technique is adopted here. The basic idea is to define the notions of length of an inference rule and of size of a proof, and apply induction on the size of the assumed derivation. In this way there is no difference (apart from the length) between the inference rules that introduce new assumptions and the ones that do not introduce new assumptions. At the inductive step the important consideration is the total size of the subproof under consideration.

Informally, given a LDCL proof, the size of a proof is the sum of the *length* of the inference rules used in the proof. Inference rules of an LDCL system can be grouped into four different categories, depending on the number of their subderivations. The definition of *length of an inference rule*, depends on the category the rule belongs to. The first category includes only the diagram reduction rule $D_{Red}$, which is the only inference rule that does not infer new declarative units or new $R$–literals and does not use any LDCL subderivation as condition. This rule is defined to have length equal to zero. The second category consists of inference rules that infer new declarative units and/or new $R$–literals without using any subderivations as conditions. Rules belonging to this category have length equal to 1. Examples are, for instance, $\vee I$ and $\wedge E$. The third category includes those rules that require only one subderivation, as for instance the $\supset I$ or the $R_{Intr}$ rule. The length of these rules is given by the smallest of the sizes of all subderivations that can be used as a condition of the rule, incremented by 1. Finally, the fourth category includes rules that have two subderivations as conditions. The only rule of this category is the $\vee E$ rule. In this case the length is given by the sum of the lengths of the smallest subderivation that can be used as the two respective conditions of the rule, incremented by 1.

Given that the semantics of an LDCL system is based on a first-order translation method, the proof of the soundness property of $\vdash_{LDCL}$ with respect to $\models_{LDCL}$ is based on the soundness of the first-order classical derivability relation $\vdash_{FOL}$. The formal statement of the theorem is given in Theorem 5.10 and a diagrammatic representation of the proof is given in Figure 1. The soundness statement is proved by the composition of three main steps. The first step proves that the hypothesis, $\mathcal{C} \vdash_{LDCL} \mathcal{C}^I$ implies that $\mathcal{A}^+ \cup FOT(C) \vdash_{FOL} FOT(C^I)$ (see Lemma 5.9). This trivially implies (by soundness of first-order logic) that $\mathcal{A}^+ \cup FOT(C) \models_{FOL} FOT(C^I)$, which gives the second step of the proof. The third step of the proof is given by the definition of the semantic entailment between configurations, which directly gives that $C \models_{LDCL} C^I$ (see Definition 5.6).

**Proposition 5.7 (Classical soundness)** Let $\mathcal{A}^+$ be the algebra of a $LDCL$ and $C, C^I$ two configurations. If $\mathcal{A}^+ \cup FOT(C) \vdash_{FOL} FOT(C^I)$ then $\mathcal{A}^+ \cup FOT(C) \models_{FOL} FOT(C^I)$.

Proof. By hypothesis, $\mathcal{A}^+ \cup FOT(C)$ derives $[\![\alpha]\!]\omega$ for each $[\![\alpha]\!]\omega \in \Delta$ and $\mathcal{A}^+ \cup FOT(C)$ derives $\mathbb{R}$, for each $\mathbb{R} \in \mathcal{D}$. By soundness of first-order logic, $\mathcal{A}^+ \cup FOT(C)$ semantically entails $\alpha : \omega$ and $\mathcal{A}^+ \cup FOT(C)$ semantically entails $\mathbb{R}$ as defined before; thus $\mathcal{A}^+ \cup FOT(C) \models_{FOL} FOT(C^I)$. ∎

**Proposition 5.8** Let $\mathcal{A}^+$ be the algebra of a $LDCL$ system and let $\langle\{C_0, ..., C_k, ..., C_n\}, \mu\rangle$ be a proof where $k \geq 0$ and $n > k$ and $s$ is a map from $\{0, ..., n-1\}$ to $\mathcal{I}$ s.t. $C_i \rightsquigarrow C_{i+1} \in \mathcal{I}$. Let $\mu(j)$ be the inference rule Diagram reduction for all $k \leq j < n$ and let $\mathcal{A}^+ \cup FOT(C_0) \vdash_{FOL} FOT(C_k)$. Then $\mathcal{A}^+ \cup FOT(C_0) \vdash_{FOL} FOT(C_n)$.

Proof. Since the rule of Diagram reduction presupposes that its conclusion - a configuration - is contained in the original configuration we have that $C_n \subseteq C_k$; by reflexivity of the $\vdash_{FOL}$ relation we obtain $\mathcal{A}^+ \cup FOT(C_0) \vdash_{FOL} FOT(C_k)$ and by transitivity of $\vdash_{FOL}$, $\mathcal{A}^+ \cup FOT(C_0) \vdash_{FOL} FOT(C_n)$. ∎

Proposition 5.8 allows us, without loss of generality, to prove, Lemma 5.9 for those derivations that do not apply $D_{Red}$ in the last step of the proof.

**Lemma 5.9 (Soundness with respect to Translations)** Let $\mathcal{A}^+$ be the extended algebra of a $LDCL$, let $C$ and $C^I$ be two configurations and let $FOT(C)$ and $FOT(C^I)$ be their first order translations. If $C \vdash_{LDCL} C^I$ then $\mathcal{A}^+ \cup FOT(C) \vdash_{FOL} FOT(C^I)$

Proof. Given a derivation $\langle\{C_0, ..., C_n\}, \mu\rangle$ where $C_0 = C$ and $C_n = C^I$, we proceed by induction on the length of the smallest derivations. When the length of the derivation is zero, we have that $C_n \subseteq C_0$, and $\mathcal{A}^+ \cup FOT(C_0) \vdash_{FOL} FOT(C_n)$.
Now, for the inductive step, suppose the length of the derivation $length(\langle\{C_0, ..., C_n\}, \mu\rangle) = \lambda, \lambda > 0$. Also, assume that $\mu(n-1)$ is not the rule of diagram reduction. When $n = 1$, $n-1 = 0$ and $length(C_{n-1} \rightsquigarrow C_n, \mu) = \lambda$. For $n > 1$ we

have that $length(C_{n-1} \rightsquigarrow C_n, \mu(n-1)) \in (O, \lambda]$ and $length(\langle\{C_0, ..., C_n\}, \mu'\rangle) \in [0, n)$ where $\mu'(i) = \mu(i)$ for all $i \in [0, n-2]$. Thus, $\mathcal{A}^+ \cup FOT(C_0) \vdash_{FOL} FOT(C_{n-1})$. Now we need to show that $\mathcal{A}^+ \cup FOT(C_{n-1}) \vdash_{FOL} FOT(C_n)$ for any rule $\mu(n-1)$ in $\mathcal{I}$. We only show this for the case when the rule $\mu(n-1)$ is a conditional rule.

• In the case where the inference rule $(> I)$ is applied in the last line a derivation, there exists an $R$-literal of the form $R\omega h_1$, where $h_1$ abbreviates $h_{g_{\alpha,\beta}}(\omega)$ , and a declarative unit $\alpha : h_1$ such that $C_{n-1} + [R\omega h_1, \alpha : h_1] \vdash_{LDCL} \alpha : \omega', \Box(\alpha \supset \beta) : \omega', Rh_1\omega'$ holds for some $\omega'$ and $C_n$ is equal to $C_{n-1} + [\alpha > \beta : \omega]$. Therefore $FOT(C_n) = FOT(C_{n-1}) \cup \{[\![\alpha > \beta]\!]\omega\}$. By reflexivity of $\vdash_{FOL}$, $\mathcal{A}^+ \cup FOT(C_{n-1}) \vdash_{FOL} FOT(C_{n-1})$ holds; but we still need to show that $\mathcal{A}^+ \cup FOT(C_{n-1}) \vdash_{FOL} [\![\alpha > \beta]\!]\omega$.

Let $\langle\{C_{n-1} + [R\omega h_1, \alpha : h_1], ..., C^S\}, \mu'\rangle$ with $\alpha : \omega', \Box(\alpha \supset \beta) : \omega', Rh_1\omega' \in C^S$ be a proof of the smallest length of $C_{n-1} + [R\omega h_1, \alpha : h_1] \vdash_{LDCL} \alpha : \omega', \Box(\alpha \supset \beta) : \omega', Rh_1\omega'$ for some $\omega'$. By hypothesis of inductive step, $0 < l(C_{n-1} \rightsquigarrow C_n, (> I)) = 1 + l_1 \leq \lambda$, where $l_1$ is the smallest length of subderivations that can be used as conditions of the inference rule $(> I)$. Then $l_1 = length(\langle\{C_{n-1} + [R\omega h_1, \alpha : h_1], ..., C^S\}, s'\rangle \in [0, \lambda)$. By inductive hypothesis, $\mathcal{A}^+ \cup FOT(C_{n-1}) \cup \{R\omega h_1, \alpha : h_1\} \vdash_{FO} FOT(C^S)$ and $\mathcal{A}^+ \cup FOT(C_{n-1}) \cup \{R\omega h_1, \alpha : h_1\} \vdash_{FOL} [\![\alpha]\!]\omega' \wedge [\![\Box(\alpha \supset \beta)]\!]\omega' \wedge Rh_1\omega'$.
By the first order logic deduction theorem we have $\mathcal{A}^+ \cup FOT(C_{n-1}) \vdash_{FOL} (R\omega h_1 \wedge [\![\alpha]\!]h_1) \supset \exists y([\![\alpha]\!]y \wedge [\![\Box(\alpha \supset \beta)]\!]y \wedge Rh_1 y)$. Thus by axiom (Ax9), $\mathcal{A}^+, FOT(C_{n-1}) \vdash_{FOL} [\![\alpha > \beta]\!]\omega$.

• In the case where the inference rule $(> E)$ is applied as last rule in a derivation, then there exists a $R$-literal $R\omega\omega' \in C_{n-1}$ and declarative units $\alpha : \omega', \alpha > \beta : \omega \in C_{n-1}$ and $C_n = C_{n-1} + [\alpha : h_2, \Box(\alpha \supset \beta) : h_2, R\omega' h_2]$, where $h_2$ abbreviates $h_{f_{\alpha,\beta}}(\omega')$. Then $\{[\![\alpha > \beta]\!]\omega, [\![\alpha]\!]\omega', R\omega\omega'\} \subseteq FOT(C_{n-1})$ and $FOT(C_n) = FOT(C_{n-1}) \cup \{[\![\alpha]\!]h_2, [\![\Box(\alpha \supset \beta)]\!]h_2, R\omega' h_2\}$. By reflexivity of $\vdash_{FOL}$, $\mathcal{A}^+, FOT(C_{n-1}) \vdash_{FOL} FOT(C_{n-1})$; and by axiom (Ax10) and $FOT(C_{n-1})$, we have $\mathcal{A}^+, FOT(C_{n-1}) \vdash_{FOL} [\![\alpha]\!]h_2; \mathcal{A}^+, FOT(C_{n-1}) \vdash_{FOL} [\![\Box(\alpha \supset \beta)]\!]h_2$ and $\mathcal{A}^+, FOT(C_{n-1}) \vdash_{FOL} R\omega_1 h_2$. ∎

**Theorem 5.10 (Soundness)** If there is a natural deduction proof of a configuration $C^I$ from a configuration $C$ then $C$ semantically entails $C^I$, i.e. if $C \vdash_{LDCL} C^I$ then $C \models_{LDCL} C^I$.

The proof of the theorem is illustrated by the diagram below. The diagram represents a proof technique introduced by Russo in [43] for the case of modal logic described as labelled deductive systems and further developed by Broda and Russo in [10].

Proof. By hypothesis $C \vdash_{LDCL} C^I$; by Lemma 5.9, $\mathcal{A}^+ \cup FOT(C) \vdash_{FOL} FOT(C^I)$. Proposition 5.7 gives us $\mathcal{A}^+ \cup FOT(C) \models_{FOL} FOT(C^I)$, and by definition of semantic entailment we get $C \models_{LDCL} C^I$. ∎

FIG. 1. Diagram of the Soundness Proof

$$\begin{array}{ccc} C \vdash_{LDCL} C^I & \Longrightarrow & C \models_{LDCL} C^I \\ \downarrow & & \uparrow \\ \mathcal{A}^+ \cup FOT(C) \vdash_{FOL} FOT(C^I) & \dashrightarrow & \mathcal{A}^+ \cup FOT(C) \models_{FOL} FOT(C^I) \end{array}$$

### 5.1.2 Completeness Formalisation

The proof of completeness is an adaptation of the Lindenbaum/Henkin [27] classical proof technique of constructing maximal consistent sets, as applied by Makinson in [36] to prove completeness of modal logics with respect to Kripke models and extended by Broda and Russo to labelled modal logics in [10, 44]. The completeness theorem is proved by contraposition. We show that, if $C \nvdash_{LDCL} C^I$ then $C \not\models_{LDCL} C^I$. To do this we provide a definition of maximal consistent configurations and we show relevant properties of such configurations with respect to declarative units and $R$-literals. Omitted proofs can be found in [33].

**Definition 5.11 (Consistent Configuration)** Let $C$ be a configuration of an arbitrary $LDCL$. $C$ is consistent if $C \nvdash_{LDCL} \perp : \omega$ for some ground term $\omega$ of $\text{ext}(\mathcal{L}_L, \mathcal{L}_C)$. We say that $C$ is *inconsistent* if it is not consistent.

**Definition 5.12 (Maximal Consistent Configuration)** The configuration denoted by $C_{max}$ is a *maximal consistent configuration* of $LDCL$ if it is consistent and if for any $\delta$, where $\delta$ is either a declarative unit or an $R$-literal not in $C_{max}$, the configuration $C_{max} + [\delta]$ is inconsistent.

The use of symbols $f_\alpha, g_\alpha, h_{g_{\alpha,\beta}}, h_{f_{\alpha,\beta}}$ is related to the interpretation of the operators associated with modalities and conditionals in the proof procedure. In order to guarantee consistency in the construction of a maximal consistent configuration, it is necessary that these skolem constant symbols satisfy some properties. These are stated below.

**Proposition 5.13** Let $C = \langle \mathcal{D}, \mathbf{f} \rangle$ be a consistent configuration, $\omega$ an arbitrary label and $\alpha$ be a wff of $\mathcal{L}_C$. Then (i) if $\Diamond\alpha \in \mathbf{f}(\omega)$ then $\neg R\omega f_\alpha(\omega) \notin \mathcal{D}$ and $\neg\alpha \notin \mathbf{f}(f_\alpha(\omega))$, (ii) if $\neg\Box\alpha \in \mathbf{f}(\omega)$ then $\neg R\omega g_\alpha(\omega) \notin \mathcal{D}$ and $\alpha \notin \mathbf{f}(g_\alpha(\omega))$, (iii) if $\alpha > \beta \in \mathbf{f}(\omega)$, $R\omega\omega' \in \mathcal{D}$ and $\alpha \in \mathbf{f}(\omega')$, then $\neg R\omega' h_{f_{\alpha,\beta}}(\omega') \notin \mathcal{D}$, $\neg\alpha \notin \mathbf{f}(h_{f_{\alpha,\beta}}(\omega'))$ and $\neg \Box (\alpha \supset \beta) \notin \mathbf{f}(h_{f_{\alpha,\beta}}(\omega'))$, and (iv) if $\neg\alpha > \beta \in \mathbf{f}(\omega)$ then $\neg R\omega h_{g_{\alpha,\beta}}(\omega) \notin \mathcal{D}$, $\neg\alpha \notin \mathbf{f}(h_{g_{\alpha,\beta}}(\omega))$, and for any $\omega'$, either $\alpha \notin \mathbf{f}(\omega')$, $\Box(\alpha \supset \beta) \notin \mathbf{f}(\omega')$ or $Rh_{g_{\alpha,\beta}}(\omega)\omega' \notin \mathcal{D}$.

The proof consists in showing that if the above conditions do not hold, then it is possible to derive an inconsistency within a configuration $C$ and then contradict the hypothesis that $C$ is consistent. For details of the proof the reader is referred to [33].

The next theorem is a characterisation of the derivability relation. It expresses the fact that $C^I$ is derivable from $C$ if each $R$-literal or a declarative unit can be derived from $C$.(For the proofs of the next theorems and propositions, the reader is referred to [33])

**Theorem 5.14 (Characterisation of Derivability)** Let $C, C^I$ be two configurations of an arbitrary $LDCL$ system, let $\delta$ be either a declarative unit or $R$-literal. If $C^I - C$ is finite then $C \vdash_{LDCL} C^I$ if and only if for all $\delta \in C^I - C$ we have $C \vdash_{LDCL} \delta$.

The next proposition and the next theorem are important properties of configurations. The proposition that follows refers to an important property of consistent configurations, namely, consistency of sub-configurations. Theorem 5.16 states the compactness property for an LDCL system.

**Proposition 5.15** Let $C$ be an arbitrary configuration of a $LDCL$ system, $\delta$ be a declarative unit or $R$-literal. Then, if $C + [\delta]$ is consistent then for any configuration $C_i, C_i \subseteq C$ we have $C_i + [\delta]$ is consistent.

**Theorem 5.16 (Finiteness)** Let $C$ be an arbitrary configuration of an arbitrary $LDCL$ system, let $\delta$ be a declarative unit or $R$-literal. If $C \vdash_{LDCL} \delta$ then there exist a finite configuration $C^I, C^I \subseteq C$ and $C^I \vdash_{LDCL} \delta$.

**Corollary 5.17 (Compactness)** Let $C$ be a configuration of a labelled deductive conditional logic system. If for any finite configuration $C^I$, $C^I \subseteq C$, $C^I$ is consistent, then $C$ is consistent.

**Definition 5.18 (Construction of Maximal Consistent Configuration)** Let $LDCL$ be an arbitrary labelled deductive conditional logic system, $\delta_1, \delta_2, ...\delta_n, ...$ be an ordering on the set of declarative units and $R$-literals of $LDCL$. Let $C$ be a consistent configuration; let $C_0 = C$. If $C_0$ is consistent then we construct a sequence of consistent configurations $C_i$ by induction on $i$ and for each $\delta_i$, $C_i$ is defined as: $C_i = C_{i-1} + [\delta_i]$ if $C_{n-1} + [\delta_i]$ is consistent; $C_i = C_{i-1}$, otherwise. Let $C_0, C_1, ..., C_n, ...$ be the sequence of configurations so constructed. Then $C_{max}$ is the configuration which contains all $\delta_i \in C_i$ (for all $i \geq 1$), i.e. $C_{max} = \cup_{i \geq 0} C_i$.

Notice that each $C_i$ included in $C_{max}$ is consistent, by construction and by the assumption that $C_0$ is consistent. Moreover, each $C_i \subseteq C_{max}$ is a consistent configuration. It is therefore easy to show that given a consistent configuration $C$ and a configuration $C_{max}$ constructed as in Definition 5.18, $C_{max}$ is consistent and maximal.

**Proposition 5.19 (Consistency w.r.t Declarative Units/R-literals)** Let $C_{max}$ be a maximal consistent configuration of our system of $LDCL$. Then (i) for any declarative unit $\alpha : \omega$, $\alpha : \omega$ and $\neg\alpha : \omega$ are not both in $C_{max}$; (ii) for any $R$-literal $\mathbb{R}$, $\mathbb{R}$ and $\neg\mathbb{R}$ are not both in $C_{max}$.

**Proposition 5.20 (Maximality on Declarative Units/R-literals)** Let $C_{max}$ be a maximal consistent configuration of an arbitrary $LDCL$. Then (i) for any declarative unit $\alpha : \omega$, either $\alpha : \omega \in C_{max}$ or $\neg\alpha : \omega \in C_{max}$; (ii) for any $R$-literal $\mathbb{R}$, either $\mathbb{R} \in C_{max}$ or $\neg\mathbb{R} \in C_{max}$.

**Proposition 5.21** Let $C_{max}$ be a maximal consistent configuration. Then for any $\delta$, where $\delta$ is a declarative unit or $R$-literal and any wffs $\alpha, \beta$, we have.

1. $\alpha \wedge \beta : \omega \in C_{max} \iff \alpha : \omega \in C_{max}$ and $\beta : \omega \in C_{max}$
2. if $\neg\alpha : \omega \in C_{max}$ or $\beta : \omega \in C_{max}$ then $\alpha \supset \beta : \omega \in C_{max}$
3. $\alpha \vee \beta : \omega \in C_{max} \iff \alpha : \omega \in C_{max}$ or $\beta : \omega \in C_{max}$
4. if $\alpha : \omega \in C_{max}$ and $\alpha \supset \beta : \omega \in C_{max}$ then $\beta : \omega \in C_{max}$
5. if $\Diamond\alpha : \omega \in C_{max}$ then $\alpha : f_\alpha(\omega) \in C_{max}$ and $R\omega f_\alpha(\omega) \in C_{max}$
6. if $R\omega\omega_1 \in C_{max}$ and $\alpha : \omega_1 \in C_{max}$ then $\Diamond\alpha : \omega \in C_{max}$
7. if $\neg R\omega g_\alpha(\omega) \in C_{max}$ or $\alpha : g_\alpha(\omega) \in C_{max}$ then $\Box\alpha : \omega \in C_{max}$
8. if $\Box\alpha : \omega \in C_{max}$ and $R\omega\omega_1 \in C_{max}$ then $\alpha : \omega_1 \in C_{max}$
9. if $\alpha > \beta : \omega \in C_{max}$ and $R\omega\omega' \in C_{max}$ and $\alpha : \omega' \in C_{max}$ then $R\omega' h_{f_{\alpha,\beta}}(\omega') \in C_{max}$ and $\alpha : h_{f_{\alpha,\beta}}(\omega') \in C_{max}$ and $\Box(\alpha \supset \beta) : h_{f_{\alpha,\beta}}(\omega') \in C_{max}$
10. if $\neg\alpha : h_{g_{\alpha,\beta}}(\omega) \in C_{max}$ or $\neg R\omega h_{g_{\alpha,\beta}}(\omega) \in C_{max}$ or $\exists\omega'$ ($\alpha : \omega' \in C_{max}$ and $Rh_{g_{\alpha,\beta}}(\omega') \in C_{max}$ and $\Box(\alpha \supset \beta) : \omega' \in C_{max}$) then $\alpha > \beta : \omega \in C_{max}$

Proof. The cases for (9) and (10) are proved here; for the other cases see [33]. We again use the abbreviations $h_1$ for $h_{g_{\alpha,\beta}}(\omega)$ and $h_2$ for $h_{f_{\alpha,\beta}}(\omega')$. First, assume that $\alpha > \beta : \omega \in C_{max}$, $\alpha : \omega' \in C_{max}$ and $R\omega\omega' \in C_{max}$. We reason by contradiction. Suppose that it is not the case that $\alpha : \omega' \in C_{max}$ and $\Box(\alpha \supset \beta) : \omega' \in C_{max}$ and $Rh_2\omega' \in C_{max}$. Then $\neg\alpha : \omega' \in C_{max}$ or $\neg \Box (\alpha \supset \beta) : \omega' \in C_{max}$ or $\neg Rh_2\omega' \in C_{max}$. We have either $R\omega' h_2 \in C_{max}$ or $\alpha : h_2 \in C_{max}$ or $\Box(\alpha \supset \beta) : h_2 \in C_{max}$ which, using $(> E)$, contradicts the assumption.

Now suppose that $\neg R\omega h_1 \in C_{max}$ or $(\neg\alpha : h_1 \in C_{max})$ or for some $\omega'$, $(Rh_1\omega' \in C_{max} \wedge \alpha : \omega' \in C_{max} \wedge \Box(\alpha \supset \beta) : \omega' \in C_{max})$. Assume by contradiction that $\alpha > \beta : \omega \notin C_{max}$ then by Propositions 5.20 and 5.13 $\neg(\alpha > \beta) : \omega \in C_{max}$. There are three possible cases.
Case (i): Suppose $\neg R\omega h_1 \in C_{max}$. The following derivation leads to a contradiction.

$$\begin{array}{l}
C_{max}\langle \neg R\omega h_1\rangle \ \ (InitialData) \\ \hline
\quad C_1\langle[\alpha : h_1, R\omega h_1]\rangle \ \ (Assump, C_3) \\ \hline
\quad C_2\langle Rh_1h_1, \Box(\alpha \supset \beta) : h_1\rangle \ \ (\bot_E, D_{Exp}, Refl, C_1, C_{max}) \\ \hline
\quad C_3\langle \alpha > \beta : \omega\rangle \ \ (> I, C_1, C_2) \\ \hline
\quad C_4\langle \bot : \omega\rangle \ \ (\wedge E, C_3, C_{max})
\end{array}$$

Case(ii): Suppose $\neg\alpha : h_1 \in C_{max}$. A similar derivation to that used in Case (i) leads to a contradiction.
Case(iii): Suppose $Rh_1\omega' \in C_{max}$ and $\alpha : \omega' \in C_{max}$ and $\Box(\alpha \supset \beta) : \omega' \in C_{max}$ for some $\omega'$. Then $\alpha > \beta : \omega \in C_{max}$ can be derived using $(>I)$, again yielding a contradiction. ∎

**Proposition 5.22** Let $LDCL$ be a labelled deductive conditional logic system such that $\forall x \forall y \forall z (Rxy \wedge Ryz \supset Rxz) \in \mathcal{A}$, let $C_{max}$ be a maximal consistent configuration, let $\omega_1, \omega_2, \omega_3 \in ext(\mathcal{L}_C, \mathcal{L}_L)$ be such that $R\omega_1\omega_2 \in C_{max}$ and $R\omega_2\omega_3 \in C_{max}$. Then $R\omega_1\omega_3 \in C_{max}$.

**Proposition 5.23** Let $LDCL$ be a labelled deductive conditional logic system such that $\forall x Rxx \in \mathcal{A}$, let $C_{max}$ be a maximal consistent configuration. Then for each $\omega \in ext(\mathcal{L}_C, \mathcal{L}_L)$, $R\omega\omega \in C_{max}$.

In order to prove the model existence lemma we need to define the notion of canonical interpretation. In what follows, we define the notion of canonical interpretation with respect to maximal consistent configurations.

**Definition 5.24 (Canonical Interpretation)** Let $C_{max}$ be a maximal consistent configuration relative to $LDCL$, and let $FOT(C_{max}) = \mathcal{D}_{max} \cup \Delta_{max}$ where $\Delta_{max} = \{[\![\alpha]\!]\omega \mid \alpha : \omega \in C_{max}\}$, be its first order translation. Let $H_U$ be the Herbrand universe of the language $ext^+(\mathcal{L}_C, \mathcal{L}_L)$. The *canonical interpretation* of the maximal consistent configuration $C_{max}$ is the pair $MC_{max} = (H_U, \imath_{C_{max}})$ where the interpretation function $\imath_{C_{max}}$ is defined as follows:

**i)** $||\,\omega\,||^{\imath_{C_{max}}} = \omega$ for each ground term $\omega \in ext^+(\mathcal{L}_C, \mathcal{L}_L)$
**ii)** $||\,R\,||^{\imath_{C_{max}}} = \{(\omega_i, \omega_j) | R\omega_i\omega_j \in FOT(C_{max})\}$ for $R \in ext^+(\mathcal{L}_C, \mathcal{L}_L)$
**iii)** $||[\![\alpha]\!]\,||^{\imath_{C_{max}}} = \{\omega_i | [\![\alpha]\!]\omega_i \in FOT(C_{max})\}$ for $[\![\alpha]\!] \in ext^+(\mathcal{L}_C, \mathcal{L}_L)$.

Next we show that given a maximal consistent configuration $C_{max}$ of a LDCL, the canonical interpretation is a semantic structure of the LDCL. To show this property, we have to prove that $MC_{max}$ is a model of the algebra $\mathcal{A}^+$.

**Lemma 5.25 (Model Existence for Extended Algebra)** Let $C_{max}$ be a maximal consistent configuration of our $LDCL$ system, let $\mathcal{A}^+$ be the extended algebra of $LDCL$ and $FOT(C_{max})$ be its first-order translation. Then $MC_{max}$ is a model of $\mathcal{A}^+$.

Proof. For the proof see [33]. ∎

**Lemma 5.26 (Model Existence Lemma)** Let $C_{max}$ be a maximal consistent configuration of our $LDCL$ system. For any $\delta$ as above, there exist a model structure $\mathcal{M}$ such that $\mathcal{M} \models_{LDCL} \delta$ if $\delta \in C_{max}$ and $\mathcal{M} \not\models_{LDCL} \delta$ if $\delta$ is not in $C_{max}$.

Proof. We have to consider the two possibilities ($\delta$ as a declarative unit or as an $R$-literal). If $\delta$ is a declarative unit and $\alpha : \omega \in C_{max}$ then $[\![\alpha]\!]\omega \in FOT(C_{max})$; Then $MC_{max} \models_{FOL} [\![\alpha]\!]\omega$; hence $MC_{max} \models_{LDCL} [\![\alpha]\!]\omega$. If $\alpha : \omega \notin C_{max}$ then $[\![\alpha]\!]\omega \notin FOT(C_{max})$. Therefore $\omega \notin ||\,\alpha\,||^{\imath_{C_{max}}}$ and $MC_{max} \not\models_{FOL} [\![\alpha]\!]\omega$. Hence $MC_{max} \not\models_{LDCL} \alpha : \omega$.
If $\delta$ is an $R$-literal it can be either of the form $R\omega_i\omega_j$ or of its negation. If $\delta = R\omega_i\omega_j$ and $\delta \in C_{max}$ we have that $R\omega_i\omega_j \in FOT(C_{max})$. Thus $MC_{max} \models_{FOL} R\omega_i\omega_j$ and from that $MC_{max} \models_{LDCL} R\omega_i\omega_j$.
If, on the other hand, $R\omega_i\omega_j \notin C_{max}$ then $R\omega_i\omega_j \notin FOT(C_{max})$. Therefore, $MC_{max} \not\models_{FOL} R\omega_i\omega_j$, and $MC_{max} \not\models_{LDCL} R\omega_i\omega_j$.
Now, if $\delta$ is the negation of $R\omega_i\omega_j$, i.e., $\delta = \neg R\omega_i\omega_j$. In this case, if $\delta \in C_{max}$ the $R\omega_i\omega_j \notin C_{max}$. Therefore $R\omega_i\omega_j \notin FOT(C_{max})$ and $MC_{max} \not\models_{FOL} R\omega_i\omega_j$ and $MC_{max} \models_{FOL} \neg R\omega_i\omega_j$. Thus, $MC_{max} \models_{LDCL} \neg R\omega_i\omega_j$ ∎

**Corollary 5.27** Let $C$ be a consistent configuration of a $LDCL$. Then $MC_{max}$ satisfies $C$.

**Proposition 5.28** Let $LDCL$ be a labelled deductive conditional logic system and let $C$ be a configuration of it; let $\delta$ be a declarative unit or $R$-literal such that $\delta \notin C$. If $C \not\vdash_{LDCL} \delta$ then $C + [\neg\delta]$ is a consistent configuration.

**Theorem 5.29 (Completeness)** Let $C$ and $C^I$ be two $LDCL$ configurations such that their difference is finite and let $\mathcal{A}^+$ be the extended algebra of $LDCL$. If $C \models_{LDCL} C^I$ then $C \vdash_{LDCL} C^I$.

PROOF. Assuming that $C \nvdash_{LDCL} C^I$ then by theorem 5.14 there exists a $\delta \in C^I - C$ ($\delta$ is a declarative unit or $R$-literal) such that $C \nvdash_{LDCL} \delta$. Therefore by proposition 5.28 $C + [\delta]$ is a consistent configuration. By the corollary of the model existence lemma, the interpretation of the maximal consistent configuration $\mathcal{M}_i([C + [\neg\delta]])$ satisfies the configuration $C + [\neg\delta]$; thus, by definition, $\mathcal{M}_i \models_{LDCL} C$ and also $\mathcal{M}_i \models_{LDCL} \neg\delta$. When $\delta$ is a declarative unit $\alpha : \omega$, we have $\mathcal{M}_i \models_{FOL} FOT(\neg\delta)$ where $FOT(\neg\delta) = [\![\neg\alpha]\!]\omega$, then $\mathcal{M}_i \models_{FOL} [\![\neg\alpha]\!]\omega$, and by lemma 5.25, $\mathcal{M}_i \models_{FOL} \neg[\![\alpha]\!]\omega$. Therefore, $\mathcal{M}_i \not\models_{FOL} [\![\alpha]\!]\omega$, and as a consequence, $\mathcal{A}^+, FOT(C) \not\models_{FOL} \delta$; thus $C \not\models_{LDCL} C^I$. When $\delta$ is an $R$-literal, $\mathcal{M}_i \models_{FOL} \neg\delta$. By satisfiability of first order logic $\mathcal{M}_i \not\models_{FOL} \delta$ and by definition of satisfiability of configurations $\mathcal{A}^+, FOT(C) \not\models_{FOL} \delta$; thus $C \not\models_{LDCL} C^I$. ∎

## 5.2 Correspondence Results

We show that our logical system is a generalisation of the conditional logic of normality, since it allows reasoning about structures of (possibly singleton) actual worlds. In order to show this we prove that, if we impose appropriate restrictions on initial configurations, then there exists a correspondence between a labelled conditional logic of normality and the corresponding axiomatic presentation of conditional logic systems. This generalisation is possible due to the fact that the correspondence with the BCL logic exists if and only if the initial configuration is empty. In addition we prove that the correspondence does not hold if we do not impose any restrictions. To prove that there is such a correspondence one has to identify a particular constant symbol, for instance $w_0$, in the labelling language $\mathcal{L}_L$ and to allow only initial configurations of the form $C_i = \langle \emptyset, \mathbf{f}_i \rangle$ where for any label $\omega \in ext(\mathcal{L}_C, \mathcal{L}_L)$, $\omega \neq w_0$, we have $\mathbf{f}_i(\omega) = \emptyset$. This restriction imposes that the only initial assumptions are formulae associated to the label $w_0$ and this corresponds to the concept of local assumptions from modal logics. The set $D$ is empty so that the only assumptions at the beginning are formulas of the conditional logic. Next we are going to prove that any declarative unit $\alpha : \omega$ can be derived from an empty initial configuration $C_i$ if and only if the formula $\alpha$ is derivable in the axiomatic system of Normal Conditional Logic presented in Definition 2.5, from the set of formulae that appears in $C_i$.

**Theorem 5.30 (Correspondence for the Normal Conditional Logic $BCL$)** Let $\langle BCL, \vdash_{BCL} \rangle$ be the axiomatic system for the logic BCL given in Def.2.5; let $C_\emptyset = \langle \emptyset, \mathbf{f}_\emptyset \rangle$ be the initial configuration of an $LDCL$, where $\mathbf{f} = \emptyset$ for any label $\omega$, and let $\alpha \in \mathcal{L}_C$. Then $\vdash_{BCL} \alpha \iff \forall \omega \in ext(\mathcal{L}_C, \mathcal{L}_L), C_\emptyset \vdash_{LDCL} \alpha : \omega$.

Proof. The proof is divided into two parts. The first is the "*if*" part, for which we show the contrapositive statement. We prove that, for a given $\alpha \in \mathcal{L}_C$, if $\nvdash_{BCL} \alpha$ then there exists a label $\omega \in ext(\mathcal{L}_C, \mathcal{L}_L)$ such that $C_\emptyset \nvdash_{LDCL} \alpha : \omega$. Since both $\vdash_{BCL}$ and $\vdash_{LDCL}$ are sound and complete, it is sufficient to prove that if $\not\models_{BCL} \alpha$ then there exists an $\omega \in ext(\mathcal{L}_C, \mathcal{L}_L)$ such that $C_\emptyset \not\models_{LDCL} \alpha : \omega$. Now, suppose that $\not\models_{BCL} \alpha$; since $C_\emptyset$ is empty we show that there exists a model structure which satisfies $\neg\alpha : \omega$; by hypothesis and by semantic validity, there exists a Kripke model which satisfies the formula $\neg\alpha$. Now, let $M = \langle \mathbf{W}, \mathbf{R}, \mathbf{v} \rangle$ be this model. Thus there exists a possible world $w \in \mathbf{W}$ such that $M, w \models_{BCL} \neg\alpha$. Let us assume a canonical well-ordering on the set $\mathbf{W}$, let $SAT = \{w_i | w_i \in \mathbf{W}$ and $M, w_i \models_{BCL} \neg\alpha\}$, and let $w_\bot$ be the first element of $SAT$ according to $\mathbf{W}$ canonical well-ordering. Then $M, w_\bot \models_{BCL} \neg\alpha$. Let $\iota$ be an interpretation over the language $ext(\mathcal{L}_C, \mathcal{L}_L)$, where its universe of discourse $\mathbf{W}$ is defined as follows.
(**1**) For constants $w_i$: $\| w_i \|^\iota = w_\bot$.
(**2**) For function symbols:

(a) $\| f_\phi \|^\iota = \mathbf{f}_\phi : \mathbf{W} \to \mathbf{W}$ such that for each $w \in \mathbf{W}$ we define: for a non-empty set $SAT_\phi(w) = \{w_s | w_s \in \mathbf{W}, \mathbf{R}ww_s$ and $M, w_s \models_{BCL} \phi\}$, we define $\mathbf{f}_\phi(w) = w_{\bot_s}$ where $w_{\bot_s}$ is the first element of $SAT_\phi(w)$ with respect to the assumed order of $\mathbf{W}$. Otherwise, $\mathbf{f}_\phi(w) = w_{\bot_\mathbf{W}}$, where $w_{\bot_\mathbf{W}}$ is the first element of $\mathbf{W}$.

(b) $\| g_\phi \|^\iota = \mathbf{g}_\phi : \mathbf{W} \to \mathbf{W}$ such that for each $w \in \mathbf{W}$ we define: if for all $w' \in \mathbf{W}$ not $\mathbf{R}ww'$, then $\mathbf{g}_\phi(w) = w$. If for each $w_i \in ACCESS(w) = \{w' | w' \in \mathbf{W} \wedge \mathbf{R}ww'\}$ we have $M, w_i \models_{BCL} \phi$, then $\mathbf{g}_\phi(w) = w_{\bot_a}$ where $w_{\bot_a}$ is the first element of $ACCESS(w)$ with respect to the assumed order of $\mathbf{W}$. Otherwise, $\mathbf{g}_\phi(w) = w_{\bot_{s'}}$ where $w_{\bot_{s'}}$ is the first element of the non-empty set $SAT_{\neg\phi} = \{w' | w' \in \mathbf{W}, \mathbf{R}ww'$ and $M, w' \models_{BCL} \neg\phi\}$, according to $\mathbf{W}$'s order.

(c)$\| h_{f_{\phi,\psi}} \|^\iota = \mathbf{h}_{\mathbf{f}_{\phi,\psi}} : \mathbf{W} \to \mathbf{W}$ such that for each $w \in \mathbf{W}$ we define: for a non-empty set $SAT_{\phi,\psi}(w) = \{w_s | w_s \in \mathbf{W}, \exists w''(w'' \in \mathbf{W}, \mathbf{R}ww'', \mathbf{R}w''w_s, M, w_s \models_{BCL} \phi$ and $M, w_s \models_{BCL} \Box(\phi \supset \psi)\}$, we define $\mathbf{h}_{\mathbf{f}_\phi}(w) = w_{\bot_s}$ where $w_{\bot_s}$ is the first element of $SAT_{\phi,\psi}(w)$ with respect to the assumed order of $\mathbf{W}$. Otherwise, $\mathbf{h}_{\mathbf{f}_{\phi,\psi}}(w) = w_{\bot_\mathbf{W}}$, where $w_{\bot_\mathbf{W}}$ is the first element of $\mathbf{W}$.

(d) $\| h_{g_{\phi,\psi}} \|^\iota = \mathbf{h}_{\mathbf{g}_{\phi,\psi}} : \mathbf{W} \to \mathbf{W}$ such that for each $w \in \mathbf{W}$ we define: if for all $w' \in \mathbf{W}$ not $\mathbf{R}ww'$, then $\mathbf{h}_{\mathbf{g}_{\phi,\psi}}(w) = w$. If for each $w_i \in Acc(w) = \{w_s | w_s \in \mathbf{W}$ and $\mathbf{R}ww_s$, either $\neg(M, w_s \models_{BCL} \phi)$ or $\exists w'' \in \mathbf{W}(\mathbf{R}w_sw''$, $M, w'' \models_{BCL} \phi$ and $M, w'' \models_{BCL} \Box(\phi \supset \psi))\}$ then $\mathbf{g}_{\mathbf{h}_{\phi,\psi}}(w) = w_{\bot_s}$ where $w_{\bot_s}$ is the first element of $Acc(w)$ with respect to the assumed order of $\mathbf{W}$. Otherwise $\mathbf{g}_{\mathbf{h}_{\phi,\psi}}(w) = w_{\bot_{s'}}$, where $w_{\bot_{s'}}$ is the first element of the non-empty set $Sat' = \{w_{s'} | w \in \mathbf{W}$ and $w_{s'} \notin Acc(w)\}$, according to the assumed order of $\mathbf{W}$.
(**3**) For the predicate $[\![\phi]\!]$, $\| [\![\phi]\!] \|^\iota = \{w | w \in \mathbf{W}$ and $M, w \models_{BCL} \phi\}$.
(**4**) For the binary predicate $R$, $\| R \|^\iota = \mathbf{R}$.

Now we have to show that $\langle \mathbf{W}, \iota \rangle$ is a $LDCL$ semantic structure. We have to prove that $\langle \mathbf{W}, \iota \rangle$ is a classical model of $\mathcal{A}^+$. We proceed by cases on each of the axioms; see [33] for the proof. The cases for axioms Ax9 and Ax10 are shown here.

**(Ax9)** There are two cases to consider. Let $w$ be an arbitrary element of $\mathbf{W}$. First, suppose $\mathbf{R}w\mathbf{h}_{\mathbf{g}_{\phi,\psi}}(w)$ is false. Then for all $w' \in \mathbf{W}$ it is not the case that $\mathbf{R}ww'$ and by the Kripke definition of $\alpha > \beta$, $M, w \models_{BCL} \alpha > \beta$; hence $w \in |||\llbracket\phi > \psi\rrbracket|||^{\iota}$. For the second case, $\exists w''(w'' \in \mathbf{W}$, $\mathbf{R}ww''$, $\mathbf{R}w''w_s$, $M, w_s \models_{BCL} \phi$ and $M, w_s \models_{BCL} \Box(\phi \supset \psi)$. According to the Kripke definition of $\phi > \psi$ these are exactly the required conditions for $M, w \models_{BCL} \alpha > \beta$ and hence $w \in |||\llbracket\phi > \psi\rrbracket|||^{\iota}$.

**(Ax10)** Let $w, w'$ be two arbitrary elements of $\mathbf{W}$ such that $w \in |||\llbracket\phi > \psi\rrbracket|||^{\iota}$, $\mathbf{R}ww'$ and $w' \in |||\llbracket\phi\rrbracket|||^{\iota}$. Then by the Kripke definition of $\phi > \psi$ and the definition of the $\Diamond$ operator, the set $SAT_{\phi,\psi}(w)$ is non-empty and $\exists w''(\mathbf{R}ww''$, $\mathbf{R}w''\ \mathbf{h}_{\mathbf{f}_{\phi,\psi}}(w)$, $\mathbf{h}_{\mathbf{f}_{\phi,\psi}}(w) \in |||\llbracket\phi\rrbracket|||^{\iota}$ and $\mathbf{h}_{\mathbf{f}_{\phi,\psi}}(w) \in |||\llbracket\Box(\phi \supset \psi)\rrbracket|||^{\iota}$.

For the "*only if*" part of the proof, suppose that $\vdash_{BCL}\alpha$ and that $\alpha_1, ..., \alpha_n$ where $n \geq 1$ and $\alpha_n{=}\alpha$ is the shortest derivation of $\alpha$ with length *length*$\geq 1$. The proof that $C_\emptyset \vdash_{LDCL} \alpha : \omega$ is by induction on *length*. The base case is as follows (*length*=1). In this case $\alpha$ is an instance of the axioms of BCL. Then we have to show that $C_\emptyset \vdash_{LDCL} \alpha : \omega$ by cases considering each axiom schema of BCL. Propositional rules are proven by application of the corresponding labelled natural deduction rules. The proofs for $K, T, 4, CC, CM, RCM, RT, Or, Nec$ and $MP$ are shown in [33]. For axiom $C$, $C_\emptyset \vdash_{LDCL} \Box(\alpha \supset \Diamond(\alpha \wedge \Box(\alpha \supset \beta))) \equiv \alpha > \beta : \omega$ as proven below. The abbreviations $h_1$, $h_2$, $f_1$ and $g_1$ are used, respectively, for the labels $h_{g_{\alpha,\beta}}(\omega_0)$, $h_{f_{\alpha,\beta}}(g_1)$, $f_{\alpha\wedge\Box(\alpha\supset\beta)}(h_1)$ and $g_{\alpha\supset\Diamond(\alpha\wedge\Box(\alpha\supset\beta))}(\omega_0)$.

$$C_\emptyset$$
$$C_1\langle[\Box(\alpha \supset \Diamond(\alpha \wedge \Box(\alpha \supset \beta)) : \omega_0]\rangle \ \ (Assump, C_8)$$
$$C_2\langle[\alpha : h_1, R\omega_0 h_1]\rangle \ \ (Assump, C_7)$$
$$C_3\langle\alpha \supset \Diamond(\alpha \wedge \Box(\alpha \supset \beta)) : h_1\rangle \ \ (\Box E, C_1, C_2)$$
$$C_4\langle\Diamond(\alpha \wedge \Box(\alpha \supset \beta)) : h_1\rangle \ \ (\supset E, C_2, C_3)$$
$$C_5\langle\alpha \wedge \Box(\alpha \supset \beta)) : f_1, Rh_1 f_1\rangle \ \ (\Diamond E, C_4)$$
$$C_6\langle\alpha : f_1, \Box(\alpha \supset \beta) : f_1\rangle \ \ (\wedge E, C_5)$$
$$C_7\langle\alpha > \beta : \omega_0\rangle \quad (> I, C_2 - C_6)$$
$$C_8\langle(\Box(\alpha \supset \Diamond(\alpha \wedge \Box(\alpha \supset \beta)))) \supset (\alpha > \beta) : \omega_0\rangle \ \ (\supset I, C_1 - C_7)$$

$$C_\emptyset$$
$$C_1\langle[\alpha > \beta : \omega_0]\rangle \ \ (Assump, C_9)$$
$$C_2\langle[R\omega_0 g_1]\rangle \ \ (Assump, C_8)$$
$$C_3\langle[\alpha : g_1]\rangle \ \ (Assump, C_7)$$
$$C_4\langle Rg_1 h_2, \alpha : h_2, \Box(\alpha \supset \beta) : h_2\rangle \ \ (> E, C_1 - C_3)$$
$$C_5\langle\alpha \wedge \Box(\alpha \supset \beta) : h_2\rangle \ \ (\wedge I, C_4)$$
$$C_6\langle\Diamond(\alpha \wedge \Box(\alpha \supset \beta) : g_1)\rangle \ \ (\Diamond I, C_4, C_5)$$
$$C_7\langle\alpha \supset \Diamond(\alpha \wedge \Box(\alpha \supset \beta) : g_1\rangle \ \ (\supset I, C_3 - C_6)$$
$$C_8\langle\Box(\alpha \supset \Diamond(\alpha \wedge \Box(\alpha \supset \beta))) : \omega_0))\rangle \ \ (\Box I, C_2 - C_7)$$
$$C_9\langle(\alpha > \beta) \supset (\Box(\alpha \supset \Diamond(\alpha \wedge \Box(\alpha \supset \beta)))) : \omega_0\rangle \ \ (\supset I, C_1 - C_8)$$

Now for the inductive step. Assume by inductive hypothesis that for any formula $\phi$ such that $\vdash_{BCL} \phi$ and such that there exists a proof $\phi_1, ..., \phi_n$ where $\phi_n = \phi$, that $n > 0$. Then for any $\omega_i$, $C_\emptyset \vdash_{LDCL} \phi : w_i$. Next, suppose that there exists a shortest proof $\alpha_1, ..., \alpha_{n+1}$ where $\alpha_{n+1} = \alpha$, with $n > 0$ such that $\vdash_{LDCL} \alpha$. By theorem 5.14, the finiteness theorem 5.16 and by inductive hypothesis for any arbitrary $w_i$

there exists a configuration $C_{w_i}$ such that $\alpha_i : w_i \in C_{w_i}$ for all $i \in [1, n]$, such that $C_\emptyset \vdash_{LDCL} C_{w_i}$. Since we are dealing with $n+1$, the formula $\alpha$ is not an axiom of $BCL$, then it can only be obtained using either the rule $MP$ or Necessitation. Therefore, we construct the derivations for the cases when $\alpha$ is derived by an application of *Modus Ponens*(MP), and second when $\alpha$ is derived by an application of the *Necessitation*(Nec) rule. For $MP$, we suppose that the last step in a proof is an application of $MP$; then we have $\alpha_k, (\alpha_k \supset \alpha) \in \alpha_1, ..., \alpha_n$. Thus $\alpha_k : \omega \in C_\omega$ and $\alpha_k \supset \alpha : \omega \in C_\omega$. By definition of the $(\supset E)$ rule, $C_\omega \vdash_{LDCL} \alpha : \omega$; by transitivity of $\vdash_{LDCL}$, $C_\emptyset \vdash_{LDCL} \alpha : \omega$. For $Nec$, we suppose that the last step is an application of $Nec$. Then $\alpha = \Box\alpha_k$ where $\alpha_k \in \alpha_1, ..., \alpha_n$. Thus $\alpha_k : g_{\alpha_k}(\omega) \in C_{g_{\alpha_k}(\omega)}$ and $C_\emptyset \vdash_{LDCL} C_{g_{\alpha_k}(\omega)}$. Reflexivity of the derivation renders $C_{g_{\alpha_k}(\omega)} \vdash_{LDCL} \alpha : g_{\alpha_k}(\omega)$, and transitivity of derivability renders $C_{g_{\alpha_k}(\omega)} + [R\omega g_{\alpha_k}(\omega)] \vdash_{LDCL} \alpha_k : g_{\alpha_k}(\omega)$, which by definition of the $(\Box I)$ rule implies that $C_{g_{\alpha_k}(\omega)} \vdash_{LDCL} \Box\alpha_k : \omega$, and again by transitivity of $\vdash_{LDCL}$, $C_\emptyset \vdash_{LDCL} \Box\alpha_k : \omega$. ∎

## 6 Extending the Labelled Deductive Conditional Logic with Additional Modalities

In this section we introduce a labelled deductive approach for an improved conditional logic of normality. According to Makinson, the normality conditional of Lamarre and Boutilier does not strictly correspond to the intuition underlying normality semantics [37]. The interpretation of the conditional of normality at a world $\omega$ looks only at the worlds that are at least as normal as $\omega$; however, Makinson argues that one should look at all worlds satisfying the antecedent. In order to make its normality semantics closer to the intuition of normality, Boutilier introduced two unary modal operators: one for truth in all worlds less than or equal to the worlds in which the evaluation is effected and another modality to indicate truth in all other possible worlds of the model [6]. Extending the labelled deductive conditional language with an additional modal operator will allow us to express normality properties in another way. This additional unary modal connective denoted by $\overleftarrow{\Box}$ is employed to represent truth at "less normal" or inaccessible worlds. Thus, $\overleftarrow{\Box}\alpha$ denotes that $\alpha$ is true at all inaccessible worlds; since the binary relation between elements of $W$ represents a metric of normality, we say that the syntactic formula $\overleftarrow{\Box}\alpha$ should be read as $\alpha$ holds at all less normal - or incomparable - worlds under the accessibility relation. Here a conditional of normality is represented via a bimodal language. The relation between inaccessible or "less normal" worlds is the complement of $R$ and shall be denoted by $\overline{R}$. Therefore two modal operators are introduced: one for dealing with truth "at all less normal or incomparable worlds" (denoted by $\overleftarrow{\Box}$) and another for dealing with truth "at all equally or more normal worlds" (denoted by $\Box$). Truth at all worlds in the model is the conjunction of these two modalities, defined as $\overleftrightarrow{\Box}\,\alpha \equiv \overleftarrow{\Box}\alpha \wedge \Box\alpha$. We shall define natural deduction rules for these operators later on. By adding new modalities to the language, we can capture in our LDCL system the conditional logic of normality, named by Boutilier as $CT4O$. We shall refer to this logic as $BCL^+$. The new modalities are defined as follows.

(i)$\overleftarrow{\Diamond}\alpha \equiv \neg\overleftarrow{\Box}\neg\alpha$ (ii)$\overleftrightarrow{\Box}\alpha \equiv \Box\alpha \wedge \overleftarrow{\Box}\alpha$ (iii)$\overleftrightarrow{\Diamond}\alpha \equiv \Diamond\alpha \vee \overleftarrow{\Diamond}\alpha$

The models for $BCL^+$ are defined as in BCL (analogously to definition 2.3). However, the new connectives shall have different satisfiability conditions based on the accessibility relation $\overline{R}$ (conjugate of $R$). We extend that definition with the satisfiability conditions for the additional operators as follows.

1. $(M,\omega) \models \overleftarrow{\Box}\alpha \iff$ for each $\omega_1$ such that $\overline{R}\omega\omega_1$, $(M,\omega_1) \models \alpha$
2. $(M,\omega) \models \overleftrightarrow{\Box}\alpha \iff$ for all $\omega_1 \in W$ we have$(M,\omega_1) \models \alpha$
3. $(M,\omega) \models \overleftarrow{\Diamond}\alpha \iff$ for some $\omega_1$ such that $\overline{R}\omega\omega_1, (M,\omega_1) \models \alpha$
4. $(M,\omega) \models \overleftrightarrow{\Diamond}\alpha \iff$ for some $\omega_1 \in W$, $(M,\omega_1) \models \alpha$

Another axiom that is present in the modal logic with reference to inaccessible worlds is an instance of the axiom (*) proposed by Humberstone ([28] p.348) and applied to the logics of normality by Boutilier. In [7] an instance of (*) is renamed **H** : $\overleftrightarrow{\Diamond}(\Box\alpha \wedge \overleftarrow{\Box}\beta) \supset \overleftrightarrow{\Box}(\alpha \vee \beta)$. This axiom represents the property that the accessibility relations are complementary to each other. The models for $BCL^+$ include a relation which is reflexive, transitive and connected (i.e. for all worlds $\omega_i, \omega_j$,$R\omega_i\omega_j$ or $R\omega_j\omega_i$). In [28] it is proved that the modal logic $K + \mathbf{H}$ is sound and complete with respect to the semantics of the modal logic $K^2 + (*)$ (where $K^2 + (*)$ is a bimodal extension of $K$, with an additional relation which is complementary to the standard relation $R$).

**Definition 6.1 ($BCL^+$ Axiomatisation)** The propositional conditional logic $BCL^+$ is the set of formulae that includes, for any arbitrary wffs $\alpha$ and $\beta$, propositional tautologies, all formulae of the form **K**, **T**, **4**, (as in definition 2.5), and **K′**, **S**, **H** given below, and that is closed under the inference rules **Necessity**, **MP** and **Subs** (as in definition 2.5):

**K′** $\overleftarrow{\Box}(\alpha \supset \beta) \supset (\overleftarrow{\Box}\alpha \supset \overleftarrow{\Box}\beta)$

**S** $\alpha \supset \overleftarrow{\Box}\Diamond\alpha$

**H** $\overleftrightarrow{\Diamond}(\Box\alpha \wedge \overleftarrow{\Box}\beta) \supset \overleftrightarrow{\Box}(\alpha \vee \beta)$

The system $BCL^+$, is also proved to be sound and complete wrt the class of $BCL^+$ models [7].

**Theorem 6.2 ([7])** The system $BCL^+$ is sound and complete with respect to the class of $BCL^+$ models, ie $\vdash_{BCL^+} \alpha \iff \models_{BCL^+} \alpha$.

In $BCL^+$, the definition of the conditional operator is slightly different from that given in Section 4 (recall that in $BCL$ models the relation is reflexive and transitive, whereas in $BCL^+$ connectivity is an added property of the relation).

**Definition 6.3** The normality conditional $\alpha > \beta$ for the logic $BCL^+$ is defined as: $[\mathbf{C}^+]$ $\alpha > \beta \equiv \overleftrightarrow{\Box}(\alpha \supset \Diamond(\alpha \wedge \Box(\alpha \supset \beta)))$.

When the relation is totally connected, there is only one chain of normality in the ordering relation. In that case, every world is comparable to each other and $\alpha > \beta$ holds either if $\alpha$ is false, then it holds vacuously, or if the most normal worlds in which

$\alpha$ holds, $\Box(\alpha \supset \beta)$ holds as well. Therefore we would have the following definition for the connective:
[$\mathbf{C}^{++}$] $\alpha > \beta \equiv \overleftrightarrow{\Box}\neg\alpha \vee \overleftrightarrow{\Diamond}(\alpha \wedge \Box(\alpha \supset \beta))$. However, we shall adopt the definition which resembles the definition given in section 4, and is expressed by $\mathbf{C}^{+}$. The adoption of this definition will not affect the logic in any way, since both definitions are equivalent [7]. This equivalence allows us to present natural deduction rules which are in the same spirit as the ones defined for the conditional operator in section 4. In addition to the axiom schemas defined above, some theorems that are provable in $BCL$, namely $ID, CC, CM, RCM, RT, Or$, are also valid in the bimodal extension. Recall that we used the symbols $f_\alpha$ and $g_\alpha$ to refer to worlds in which formulae $\alpha$ hold true. When extending with additional modalities, the symbols $\overleftarrow{f}_\alpha$, $\overleftarrow{g}_\alpha$ are used in a similar way to refer to inaccessible worlds. For instance, $\overleftarrow{\Box}\alpha$ means that $\alpha$ is true at all less normal and incomparable worlds, and we would use $\overleftarrow{g}_\alpha(\omega)$ to refer to an inaccessible possible world associated to $\alpha$.

## 6.1 Rules for Unconventional Modalities

These rules describe the interaction between possible worlds containing information syntactically expressed by the unconventional modalities, and aim to represent the additional modalities defined in the logic $BCL^{+}$. In order to prove that $\overleftrightarrow{\Box}\alpha$ holds at a possible world $\omega$ we have to show that $\alpha$ holds at all worlds in the model, no matter how normal they are. This is expressed by showing that both $\overleftarrow{\Box}\alpha$ and $\Box\alpha$ holds at the world $\omega$. Table 6 presents the diagrammatic representation of these rules. For the rule $\overleftrightarrow{\Diamond}I$, in order to introduce $\overleftrightarrow{\Diamond}\alpha : \omega$ we have to be able to show $\alpha : \omega'$ for some $\omega'$. This is correct, as from $\alpha : \omega'$ we may derive $\Diamond\alpha : \omega$ or $\overleftarrow{\Diamond}\alpha : \omega'$, depending on whether $R\omega\omega'$ or $\overline{R}\omega\omega'$. Since these are complements at least one must hold. A similar simplification allows the derivation of the rule for $\overleftrightarrow{\Box}E$. Two further rules, called *Comp* and *Conn* are given. These allow the use in derivations of disjunctions involving $R$-literals, namely $Rxy \vee \neg Rxy$ and $Rxy \vee Ryx$. The rules are necessary as the $D_{Exp}$ rule only allows for the derivation of $R$-literals, not the arbitrary use of the labelling algebra. Derivations of axioms **H** and **S** are given to illustrate application of the rules in Table 6. For more examples see [33]. For axiom **H**, $C_\emptyset \vdash_{LDCL} \overleftrightarrow{\Diamond}(\Box\alpha \wedge \overleftarrow{\Box}\beta) \supset \overleftrightarrow{\Box}(\alpha \vee \beta)$ as proven below. Similar to before some abbreviations have been used: $f_1$ abbreviates $f_{\Box\alpha\wedge\overleftarrow{\Box}\beta}(\omega_0)$ and $\overleftarrow{f}_2$ abbreviates $\overleftarrow{f}_{\Box\alpha\wedge\overleftarrow{\Box}\beta}(\omega_0)$. The proof also uses a Theorem, called $Th$, which states that $C\langle\Box\alpha : \omega', \overleftarrow{\Box}\beta : \omega'\rangle \vdash_{LDCL} \overleftrightarrow{\Box}(\alpha \vee \beta) : \omega'$. Its derivation is shown below also.

Table 6. Rules for the Additional Modalities

$$\dfrac{\begin{array}{c} C\langle[\overline{R}\omega\overleftarrow{g}_\alpha(\omega)]\rangle \\ \vdots \\ C_S\langle\alpha : \overleftarrow{g}_\alpha(\omega)\rangle \end{array}}{C_I\langle\overleftarrow{\Box}\alpha : \omega\rangle}\ \overleftarrow{\Box}I \qquad \dfrac{C\langle\overleftarrow{\Box}\alpha : \omega_1, \overline{R}\omega_1\omega_2\rangle}{C_I\langle\alpha : \omega_2\rangle}\ \overleftarrow{\Box}E$$

$$\dfrac{C\langle\overline{R}\omega_1\omega_2, \alpha : \omega_2\rangle}{C_I\langle\overleftarrow{\Diamond}\alpha : \omega_1\rangle}\ \overleftarrow{\Diamond}I \qquad \dfrac{C\langle\overleftarrow{\Diamond}\alpha : \omega\rangle}{C_I\langle\alpha : \overleftarrow{f}_\alpha(\omega), \overline{R}\omega\overleftarrow{f}_\alpha(\omega)\rangle}\ \overleftarrow{\Diamond}E$$

$$\dfrac{C\langle\overleftarrow{\Box}\alpha : \omega, \Box\alpha : \omega\rangle}{C_I\langle\overleftrightarrow{\Box}\alpha : \omega\rangle}\ \overleftrightarrow{\Box}I \qquad \dfrac{C\langle\overleftrightarrow{\Box}\alpha : \omega\rangle}{C_I\langle\alpha : \omega'\rangle}\ \overleftrightarrow{\Box}E$$

$$\dfrac{C\langle\overleftrightarrow{\Diamond}\alpha : \omega\rangle}{C_I\langle\Diamond\alpha \vee \overleftarrow{\Diamond}\alpha : \omega\rangle}\ \overleftrightarrow{\Diamond}E \qquad \dfrac{C\langle\alpha : \omega'\rangle}{C_I\langle\overleftrightarrow{\Diamond}\alpha : \omega\rangle}\ \overleftrightarrow{\Diamond}I$$

$$\dfrac{\begin{array}{cc} C\langle[R\omega\omega_0)]\rangle & C\langle[\overline{R}\omega\omega_0]\rangle \\ \vdots & \vdots \\ C_S\langle\gamma : \omega'\rangle & C_{S'}\langle\gamma : \omega'\rangle \end{array}}{C_I\langle\gamma : \omega'\rangle}\ Comp \qquad \dfrac{\begin{array}{cc} C\langle[R\omega\omega_0)]\rangle & C\langle[R\omega_0\omega]\rangle \\ \vdots & \vdots \\ C_S\langle\gamma : \omega'\rangle & C_{S'}\langle\gamma : \omega'\rangle \end{array}}{C_I\langle\gamma : \omega'\rangle}\ Conn$$

$$\begin{array}{l}
\underline{C_\emptyset} \\
\quad \underline{C_1\langle[\overleftrightarrow{\Diamond}(\Box\alpha \wedge \overleftarrow{\Box}\beta) : \omega_0]\rangle \quad (Assump, C_{11})} \\
\qquad \underline{C_2\langle[\Diamond(\Box\alpha \wedge \overleftarrow{\Box}\beta) : \omega_0]\rangle \quad (Assump, C_{10})} \\
\qquad \underline{C_3\langle\Box\alpha \wedge \overleftarrow{\Box}\beta) : f_1, R\omega_0 f_1\rangle \quad (\Diamond E, C_2)} \\
\qquad \underline{C_4\langle\Box\alpha : f_1, \overleftarrow{\Box}\beta) : f_1\rangle \ \ (\wedge E, C_3)} \\
\qquad \underline{C_5\langle\overleftrightarrow{\Box}(\alpha \vee \beta) : \omega_0\rangle \ \ (Th, C_4)} \\
\qquad \underline{C_6\langle[\overleftarrow{\Diamond}(\Box\alpha \wedge \overleftarrow{\Box}\beta) : \omega_0]\rangle \ \ (Assump, C_{10})} \\
\qquad \underline{C_7\langle\Box\alpha \wedge \overleftarrow{\Box}\beta) : \overleftarrow{f}_2, \overline{R}\omega_0\overleftarrow{f}_2\rangle \ \ (\overleftarrow{\Diamond}E, C_6)} \\
\qquad \underline{C_8\langle\Box\alpha : \overleftarrow{f}_2, \overleftarrow{\Box}\beta) : \overleftarrow{f}_2\rangle \ \ (\wedge E, C_7)} \\
\qquad \underline{C_9\langle\overleftrightarrow{\Box}(\alpha \vee \beta) : \omega_0\rangle \ \ (Th, C_8)} \\
\quad \underline{C_{10}\langle\overleftrightarrow{\Box}(\alpha \vee \beta) : \omega_0\rangle \ \ (\overleftrightarrow{\Diamond}E, C_1, C_2 - C_5, C_6 - C_9)} \\
C_{11}\langle\overleftrightarrow{\Diamond}(\Box\alpha \wedge \overleftarrow{\Box}\beta) \supset \overleftrightarrow{\Box}(\alpha \vee \beta) : \omega_0\rangle \quad (\supset I, C_1 - C_{10})
\end{array}$$

The derivation of the aforementioned theorem now follows, in which $g_1$ abbreviates $g_{\alpha\vee\beta}(\omega_0)$ and $\overleftarrow{g}_2$ abbreviates $\overleftarrow{g}_{\alpha\vee\beta}(\omega_0)$.

$$C_0\langle \Box\alpha : \omega_0, \overleftarrow{\Box}\beta : \omega_0\rangle \quad (InitialData)$$
$$C_1\langle [R\omega_0 g_1]\rangle \quad (Assump, C_4)$$
$$C_2\langle [\alpha : g_1]\rangle \quad (\Box E, C_0, C_1)$$
$$C_3\langle \alpha \vee \beta : g_1\rangle \quad (\vee I, C_2)$$
$$C_4\langle \Box(\alpha \vee \beta) : \omega_0\rangle \quad (\Box I, C_1 - C_3)$$
$$C_5\langle [\overline{R}\omega_0 \overleftarrow{g}_2]\rangle \quad (Assump, C_8)$$
$$C_6\langle \beta : \overleftarrow{g}_2\rangle \quad (\Box E, C_0, C_5)$$
$$C_7\langle \alpha \vee \beta : \overleftarrow{g}_2\rangle \quad (\vee I, C_6)$$
$$C_8\langle \overleftarrow{\Box}(\alpha \vee \beta) : \omega_0\rangle \quad (\overleftarrow{\Box} I, C_5 - C_7)$$
$$C_9\langle \overleftrightarrow{\Box}(\alpha \vee \beta) : \omega_0\rangle \quad (\overleftrightarrow{\Box} I, C_4, C_8)$$

The derivation of axiom **S** follows.

$$C_\emptyset$$
$$C_1\langle [\alpha : \omega_0]\rangle \quad (Assump, C_{11})$$
$$C_2\langle [\overline{R}\omega_0 \overleftarrow{g}_{\Diamond\alpha}]\rangle \quad (Assump, C_{10})$$
$$C_3\langle R\overleftarrow{g}_{\Diamond\alpha}\omega_0\rangle \quad (D_{Exp}, C_2, Conn)$$
$$C_4\langle \Diamond\alpha : \overleftarrow{g}_{\Diamond\alpha}\rangle \quad (\Diamond I, C_1, C_3)$$
$$C_5\langle \overleftarrow{\Box}\Diamond\alpha : \omega_0\rangle \quad (\overleftarrow{\Box} I, C_2 - C_4)$$
$$C_6\langle \alpha \supset \overleftarrow{\Box}\Diamond\alpha : \omega_0\rangle \quad (\supset I, C_1 - C_5)$$

## *6.2 Rules for the Additional Conditional Connective*

Recall that the definition of the conditional operator in $BCL^+$ is as follows: $\alpha > \beta \equiv \overleftrightarrow{\Box}(\alpha \supset \Diamond(\alpha \wedge \Box(\alpha \supset \beta)))$. The natural deduction rule has to take into account the fact that worlds related via the co-relation of $R$, represented by $\overline{R}$ should be considered in the scale of normality. The intuition underlying the rules is similar to the intuition behind the rules we defined for the case of the single modality. In addition we have to consider the co-existent relation $\overline{R}$. Recall that, according to Makinson [37], when assessing if $\alpha > \beta$ is true at $\omega$ one should not look only at the worlds that are at least as normal as $\omega$, but to all the worlds satisfying $\alpha$, and then assess the conditional. Therefore the co-relation $\overline{R}$ is introduced to look at the all the other worlds that are not at least as normal as $\omega$. The rules for the conditional operator are defined next. It will be helpful in this section to introduce the notation $\alpha \overleftrightarrow{>} \beta$ in place of $\alpha > \beta$, to remind the reader of the presence of the co-relation $\overline{R}$. This will also prove helpful in the formulation of the $\overleftrightarrow{>} I$ rule, in which a new operator $\overleftarrow{>}$ is used, analogous to $>$, but in which $\overline{R}$ is used in place of $R$. Introduction and elimination rules for $\overleftarrow{>}$ are given next.

**Definition 6.4** Let $C$ be an arbitrary configuration, $\omega$ be an arbitrary ground term and $\alpha, \beta$ well-formed formulae. We say that $C \rightsquigarrow C + [\alpha \overleftarrow{>} \beta : \omega]$ is a member of the inference rule $(\overleftarrow{>} I)$ if $C + [\alpha : h_{g_{\alpha,\beta}}(\omega), \overline{R}\omega h_{g_{\alpha,\beta}}(\omega)] \vdash_{LDCL} [\alpha : \upsilon, \Box(\alpha \supset \beta) : \upsilon, Rh_{g_{\alpha,\beta}}(\omega)\upsilon]$. The diagrammatic representation of the rule is as follows.

$$\dfrac{\begin{array}{c} C\langle[\alpha : h_{g_{\alpha,\beta}}(\omega), \overline{R}\omega h_{g_{\alpha,\beta}}(\omega)]\rangle \\ \vdots \\ C_S\langle \alpha : \upsilon, \Box(\alpha \supset \beta) : \upsilon, Rh_{g_{\alpha,\beta}}(\omega)\upsilon]\rangle \end{array}}{C_I\langle \alpha \overleftarrow{>} \beta : \omega\rangle} (\overleftarrow{>} I)$$

**Definition 6.5** Let $C$ be an arbitrary configuration, $\omega, \omega'$ be arbitrary ground terms and $\alpha, \beta$ well-formed formulae. We say that $C \rightsquigarrow C + [R\omega' h_{f_{\alpha,\beta}}(\omega'), \alpha : h_{f_{\alpha,\beta}}(\omega'), \Box(\alpha \supset \beta) : h_{f_{\alpha,\beta}}(\omega')]$ is a member of the inference rule $(\overleftarrow{>} E)$ if $\alpha > \beta : \omega, \overline{R}\omega\omega', \alpha : \omega' \in C$. The diagrammatic representation of the rule is as follows.

$$\dfrac{C\langle \alpha > \beta : \omega, \overline{R}\omega\omega', \alpha : \omega'\rangle}{C_I\langle R\omega' h_{f_{\alpha,\beta}}(\omega'), \alpha : h_{f_{\alpha,\beta}}(\omega'), \Box(\alpha \supset \beta) : h_{f_{\alpha,\beta}}(\omega')\rangle} (\overleftarrow{>} E)$$

The introduction and elimination rules for $\overleftrightarrow{>}$ are given in terms of $>$ and $\overleftarrow{>}$, since $\alpha \overleftrightarrow{>} \beta$ is defined to be equivalent to $\alpha > \beta \wedge \alpha \overleftarrow{>} \beta$. The graphical representation of each rule is given below.

**Definition 6.6** Let $C$ be an arbitrary configuration, $\omega, \omega'$ be arbitrary ground terms and $\alpha, \beta$ well-formed formulae. We say that $C \rightsquigarrow C + [\alpha \overleftrightarrow{>} \beta : \omega]$ (respectively $C \rightsquigarrow C + [\alpha > \beta \wedge \alpha \overleftarrow{>} \beta : \omega]$) is a member of the inference rule $\overleftrightarrow{>} I$ (resp. $\overleftrightarrow{>} E$) if $\alpha > \beta : \omega, \alpha \overleftarrow{>} \beta : \omega \in C$ (resp. $\alpha \overleftrightarrow{>} \beta : \omega \in C$). The diagrammatic representation of the rules are as follows.

$$\dfrac{C\langle \alpha > \beta : \omega, \alpha \overleftarrow{>} \beta : \omega\rangle}{C_I\langle \alpha \overleftrightarrow{>} \beta : \omega\rangle} \overleftrightarrow{>} I \qquad \dfrac{C\langle \alpha \overleftrightarrow{>} \beta : \omega\rangle}{C_I\langle \alpha > \beta \wedge \alpha \overleftarrow{>} \beta : \omega\rangle} \overleftrightarrow{>} E$$

The use of the new conditional rules is illustrated in the following derivation of $\alpha \overleftrightarrow{>} \beta \supset \overleftrightarrow{\Diamond}\alpha \supset \overleftrightarrow{\Diamond}(\alpha \wedge \Box(\alpha \supset \beta)) : \omega_0$, that proves part of the equivalence of $\alpha \overleftrightarrow{>} \beta$ and $\overleftrightarrow{\Diamond}\alpha \supset \overleftrightarrow{\Diamond}(\alpha \wedge \Box(\alpha \supset \beta))$.

$$C_0\langle \alpha \overleftrightarrow{>} \beta : \omega_0\rangle \;\; (InitialData(Part(i)))$$

$$C_1\langle [\overleftrightarrow{\Diamond}\alpha : \omega_0]\rangle \quad (Assump, C_{18})$$

$$C_2\langle [\Diamond\alpha : \omega_0]\rangle \quad (Assump, C_7)$$

$$C_3\langle \alpha : f_1, R\omega_0 f_1\rangle \quad (\Diamond E, C_2)$$

$$C_4\langle \alpha : h_1, Rf_1h_1, \Box(\alpha \supset \beta) : h_1\rangle \quad (\overleftrightarrow{>} E, \wedge E, > E, C_0, C_3)$$

$$C_5\langle R\omega_0 h_1\rangle \;\; (D_{Exp}, C_3, C_4, Trans)$$

$$C_6\langle \Diamond(\alpha \wedge \Box(\alpha \supset \beta)) : \omega_0\rangle \;\; (\Diamond I, C_4, C_5)$$

$$C_7\langle \overleftrightarrow{\Diamond}(\alpha \wedge \Box(\alpha \supset \beta)) : \omega_0\rangle \;\; (\overleftrightarrow{\Diamond} I, C_6)$$

$$C_8\langle \overleftarrow{\Diamond}\alpha : \omega_0\rangle \;\; (Assump, C_{17})$$

$$C_9\langle \alpha : \overleftarrow{f}_2, \overline{R}\omega_0 \overleftarrow{f}_2\rangle \;\; (\overleftarrow{\Diamond} E, C_8)$$

$$C_{10}\langle \alpha : h_2, R\overleftarrow{f}_2 h_2, \Box(\alpha \supset \beta) : h_2\rangle \;\; (\overleftrightarrow{>} E, \wedge E, \overleftarrow{>} E, C_0, C_9)$$

$$C_{11}\langle [R\omega_0 h_2]\rangle \;\; (Assump, C_{13})$$

$$C_{12}\langle \Diamond(\alpha \wedge \Box(\alpha \supset \beta)) : \omega_0\rangle \;\; (\Diamond I, C_{10}, C_{11})$$

$$C_{13}\langle \overleftrightarrow{\Diamond}(\alpha \wedge \Box(\alpha \supset \beta)) : \omega_0\rangle \;\; (\overleftrightarrow{\Diamond} I, C_{12})$$

$$C_{14}\langle [\overline{R}\omega_0 h_2]\rangle \;\; (Assump, C_{16})$$

$$C_{15}\langle \overleftarrow{\Diamond}(\alpha \wedge \Box(\alpha \supset \beta)) : \omega_0\rangle \quad (\overleftarrow{\Diamond} I, C_{10}, C_{14})$$

$$C_{16}\langle \overleftrightarrow{\Diamond}(\alpha \wedge \Box(\alpha \supset \beta)) : \omega_0\rangle \quad (\overleftrightarrow{\Diamond} I, C_{15})$$

$$C_{17}\langle \overleftrightarrow{\Diamond}(\alpha \wedge \Box(\alpha \supset \beta)) : \omega_0\rangle \quad (Comp, C_{11} - C_{13}, C_{14} - C_{16})$$

$$C_{18}\langle \overleftrightarrow{\Diamond}(\alpha \wedge \Box(\alpha \supset \beta)) : \omega_0\rangle \;\; (\overleftrightarrow{\Diamond} E, C_2 - C_7, C_8 - C_{17})$$

$$C_{19}\langle \overleftrightarrow{\Diamond}\alpha \supset \overleftrightarrow{\Diamond}(\alpha \wedge \Box(\alpha \supset \beta)) : \omega_0\rangle \;\; (\supset I, C_1 - C_{18})$$

## 6.3 Semantics

If we want to express the semantics of the operators based on the bimodal logic we need an additional set of axiom schemas to be included in the extended algebra $\mathcal{A}^+$, namely:

$$\forall x((\overline{R}x\overleftarrow{g}_\alpha(x) \supset [\![\alpha]\!]\overleftarrow{g}_\alpha(x)) \supset [\![\overleftarrow{\Box}\alpha]\!]x) \tag{Ax11}$$

$$\forall x([\![\overleftarrow{\Box}\alpha]\!]x \supset (\forall y(\overline{R}xy \supset [\![\alpha]\!]y))) \tag{Ax12}$$

$$\forall x(\exists y(\overline{R}xy \wedge [\![\alpha]\!]y) \supset [\![\overleftarrow{\Diamond}\alpha]\!]x) \tag{Ax13}$$

$$\forall x([\![\overleftarrow{\Diamond}\alpha]\!]x \supset (\overline{R}x\overleftarrow{f}_\alpha(x) \wedge [\![\alpha]\!]\overleftarrow{f}_\alpha(x))) \tag{Ax14}$$

$$\forall x([\![\overleftrightarrow{\Diamond}\alpha]\!]x \iff \exists y([\![\alpha]\!]y)) \tag{Ax15}$$

$$\forall x([\![\overleftrightarrow{\Box}\alpha]\!]x \iff \forall y([\![\alpha]\!]y)) \tag{Ax16}$$

$$\forall x(((\overline{R}xh_{g_{\alpha,\beta}}(x) \wedge [\![\alpha]\!]h_{g_{\alpha,\beta}}(x)) \supset \exists y(Rh_{g_{\alpha,\beta}}(x)y \wedge [\![\alpha]\!]y \wedge [\![\Box(\alpha \supset \beta)]\!]y)) \supset [\![\alpha \overleftarrow{>} \beta]\!]x) \tag{Ax17}$$

$$\forall x([\![\alpha \overset{\leftarrow}{>} \beta]\!]x \supset \forall y(\overline{R}xy \wedge [\![\alpha]\!]y \supset (Ryh_{f_{\alpha,\beta}}(x) \wedge [\![\alpha]\!]h_{f_{\alpha,\beta}}(x) \wedge [\![\Box(\alpha \supset \beta)]\!]h_{f_{\alpha,\beta}}(x)))) \tag{Ax18}$$

$$\forall x([\![\alpha \overset{\leftrightarrow}{>} \beta]\!]x \iff ([\![\alpha > \beta]\!]x \wedge [\![\alpha \overset{\leftarrow}{>} \beta]\!]x) \tag{Ax19}$$

### *6.4 Soundness, Completeness and Correspondence for BCL+*

The extensions of the soundness and completeness theorems have to take into consideration the new modalities and the new conditional operator. However the proof technique is the same. For details the reader is referred to [33]. Correspondence results can also be obtained using the same proof technique we used before. Here we state the main correspondence theorem.

**Theorem 6.7** (Correspondence for $BCL^+$) Let $LDCL$ be the propositional labelled deductive conditional logic system whose associated labelling algebra $\mathcal{A}^+$ contains the axiom schemas $K, T, 4, C$; let $\langle BCL^+, \vdash_{BCL^+}\rangle$ be the axiomatic system for the logic $BCL^+$, let $C_\emptyset = \langle \emptyset, \mathbf{f}_\emptyset\rangle$ be the initial configuration where $\mathbf{f} = \emptyset$ for any label $\omega$, and let $\alpha \in \mathcal{L}_C$. Then $\vdash_{BCL^+} \alpha \iff \forall \omega \in ext(\mathcal{L}_C, \mathcal{L}_L), C_\emptyset \vdash_{LDCL} \alpha : \omega$.

The theorems concerning correspondence with respect to global and local assumptions are analogous to the proofs of the corresponding theorems stated for the monomodal conditional logic.

## 7 Discussion

We have presented a labelled approach to a propositional conditional logic of normality in natural deduction style. It is important to observe that our methodology is distinct from Boutilier-Lamarre's, as it allows for the reference to specific worlds, being an explicit approach to normality reasoning. Explicity approaches to logics based on possible worlds semantics have the advantage of clearly expressing what is true at possible worlds; moreover it enjoys uniformity, as extensions in the properties of the relation between possible worlds do not alter the rules, i.e. the natural deduction rules are independent from the particular labelling algebra. As far as our knowledge goes, no proof system based on natural deduction for conditional logics of normality had been developed so far. We have extended the Labelled Deductive Conditional Logic System with additional modalities. Labelled natural deduction rules for additional modalities and for a new conditional operator were also proposed. The system presented here is sound and complete with respect to the semantic models proposed, and the correspondence results showed that our system is as powerful as Boutilier's $CT4$ and its extensions.

In addition, our formalisation supports Gabbay's programme of developing LDS's as a general framework for logical systems. Another advantage of using the labelled deductive system methodology is that it not only allows the combination of two existing logics into a new logic, but also shows how to extend a logic in order to generalise it by combining existing logics; in our case, the CLDS methodology made possible to represent and to develop a proof system for conditional reasoning using modal operators in a hybrid system. An automated approach to the conditional logic presented in this paper is currently under investigation. By using the extended algebras one could make use of a classical first-order theorem prover. The extension of our approach to the first-order case was presented in [33]. There, we show how LDS allows for a uniform presentation of a quantified conditional logic which corresponds to the first-order conditional logic presented in [16]. Although the work presented in this paper considers the family of conditional logics of normality that are definable in terms of modalities, we believe that the expressive power of our CLDS system would make it possible to extend the approach to other classes of conditional logics. In particular, our future research includes the investigation of how the CLDS framework could be used in other conditional logics not defined in terms of modalities, such as Adams's approach, in which conditional sentences have associated probabilities [1], Lewis's analysis of counterfactuals using sphere semantics [35], and the conditional logics based on plausibility structures of Friedman et al [20].

## Acknowledgements

We would like to thank Sanjay Modgil and Artur Garcez for the conversations that helped in the improvement of this paper and to two anonymous referees for their constructive comments. Luís Lamb was partially sponsored by the Brazilian agency CAPES. Paul Taylor's LaTeX packages were used in the composition of this document.

## References

[1] Ernest W. Adams. *The Logic of Conditionals.* D. Reidel, Dordrecht, 1975.

[2] Horacio Arlo-Costa and Scott Shapiro. Maps between nonmonotonic and conditional logic. In *Proc. of KR'92*, pages 553–564. Morgan Kaufmann, 1992.

[3] Alberto Artosi and Guido Governatori. A tableau methodology for deontic conditional logic. In *Proc. of the Fourth Internat. Workshop on Deontic Logic in Computer Science*, 1998.

[4] Jonathan Bennett. Classifying conditionals: The traditional way is right. *Mind*, 104:331–354, 1995.

[5] Craig Boutilier. Conditional logics of normality as modal systems. In *Proceedings of AAAI*, pages 594–599, Boston, August 1990.

[6] Craig Boutilier. Inaccessible worlds and irrelevance: Preliminary report. In *Proc. of the IJCAI*, pages 413–418, Sydney, 1991.

[7] Craig Boutilier. Conditional logics of normality: a modal approach. *Artificial Intelligence*, 68:87–154, 1994.

[8] K. Broda, A. Russo, and D. Gabbay. A unified compilation style natural deduction system for modal, substructural and fuzzy logics. In V. Novak and I.Perfileva, editors, *Discovering World with Fuzzy logic: Perspectives and Approaches to Formalization of Human-consistent Logical Systems*. Springer, 2000.

[9] Krysia Broda, Marcelo Finger, and Alessandra Russo. Labelled natural deduction for substructural logics. *Logic Journal of the IGPL*, 7(3):283–318, 1999.

[10] Krysia Broda and Alessandra Russo. A unified compilation style labelled deductive system for modal and substructural logic using natural deduction. Technical report, Dept. of Computing, Imperial College, London, Oct. 1997.

[11] John P. Burgess. Quick completeness proofs for some logics of conditionals. *Notre Dame Journal of Formal Logic*, 22:76–84, 1981.

[12] Brian F. Chellas. Basic conditional logics. *Journal of Philosophical Logic*, 4:133–153, 1975.

[13] G. Crocco, L. Farinas del Cerro, and A. Herzig, editors. *Conditionals: from Philosophy to Computer Science*. Oxford Univ. Press, 1995.

[14] Gabriella Crocco and Luis Farinas del Cerro. Counterfactuals: Foundations for nonmonotonic inferences sequent systems. In Andre Fuhrmann and Hans Rott, editors, *Logic, Action, and Information - Essays on Logic in Philosophy and Artificial Intelligence*. de Gruyter, 1996.

[15] James P. Delgrande. An approach to default reasoning based on a first-order conditional logic: Revised report. *Artificial Intelligence*, 36:63–90, 1988.

[16] James P. Delgrande. On first-order conditional logics. *Artificial Intelligence*, 105:105–137, 1998.

[17] Michael Dummett. Natural deduction. In Ted Honderich, editor, *The Oxford Companion to Philosophy*, page 604. Oxford Univ. Press, 1995.

[18] Dorothy Edgington. On conditionals. *Mind*, 104:235–329, 1995.

[19] Nir Friedman and Joseph Halpern. On the complexity of conditional logics. In *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning KR'94*, 1994.

[20] Nir Friedman, Joseph Y. Halpern, and Daphne Koller. First-order conditional logic for default reasoning revisited. *ACM Transactions on Computational Logic*, 1(2), 2000.

[21] Dov M. Gabbay. *Labelled Deductive Systems, Vol. I.* Clarendom Press, Oxford, 1996. Oxford Logic Guides, Vol. 33.

[22] Ian Gent. A sequent- or tableau-style system for lewis's counterfactual logic vc. *Notre Dame Journal of Formal Logic*, 33(3):369–382, 1992.

[23] Moises Goldszmidt and Judea Pearl. Rank-based systems: a simple approach to belief revision, belief update and reasoning about evidence and actions. In *Proceedings of KR'92*, pages 661–672, 1992.

[24] Nelson Goodman. The problem of counterfactual conditionals. *Journal of Philosophy*, 44:113–128, 1947.

[25] Chris Groeneboer and James P. Delgrande. A general approach to determining the validity of commonsense assertions using conditional logics. *Intl. Journal of Intelligent Systems*, 5:505–520, 1990.

[26] W. L. Harper, Robert Stalnaker, and G. Pearce. *Ifs*. Reidel, Dordrecht, 1981.

[27] Leon Henkin. The discovery of my completeness proofs. *Bulletin of Symbolic Logic*, 2:127–158, 1996.

[28] I. Humberstone. Inaccessible worlds. *Notre Dame Journal of Formal Logic*, 24:346–352, 1983.

[29] Frank Jackson, editor. *Conditionals*. Oxford Univ. Press, 1991.

[30] H. Katsuno and K. Satoh. A unified view of consequence relation, belief revision and conditional logic. In Crocco et al. [13], pages 33–65.

[31] Sarit Kraus, Daniel Lehmann, and Menahem Magidor. Nonmonotonic reasoning, preferential models and cummulative logics. *Artificial Intelligence*, 44:167–208, 1990.

[32] Philippe Lamarre. S4 as the conditional logic of nonmonotonicity. In *Proccedings of KR'91*, pages 357–367, 1991.

[33] Luís da Cunha Lamb. *Labelled Proof Theory for Conditional Logic*. PhD thesis, Imperial College of Science, Technology and Medicine, London, 2000.

[34] Daniel Lehmann and Menachem Magidor. What does a conditional knowledge base entail? *Artificial Intelligence*, 55(1):1–60, May 1992.

[35] David K. Lewis. *Counterfactuals*. Blackwell, Oxford, 1973.

[36] David Makinson. On some completeness theorems in modal logic. *Zeitschrift für math. Logik und Grundlagen der Mathematik*, 12:379–384, 1966.

[37] David Makinson. Five faces of minimality. *Studia Logica*, 52:339–379, 1993.

[38] John McCarthy. Circumscription - a form of non-monotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980.

[39] Drew McDermott. Non-monotonic logic II: Non-monotonic modal theories. *Journal of the ACM*, 29(1):33–57, 1982.

[40] D. H. Mellor. How to believe a conditional. *Journal of Philosophy*, 90:233–248, 1993.

[41] Judea Pearl. Probabilistic semantics for nonmonotonic reasoning:a survey. In R. J. Brachman, H. J. Levesque, and R. Reiter, editors, *Proc. KR'89*, pages 505–516. Morgan Kaufmann, 1989.

[42] Ray Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.

[43] Alessandra M. Russo. Generalising propositional modal logic using labelled deductive systems. In F. Baader and K. U. Schulz, editors, *Frontiers of Combining Systems, Applied Logic Series (APLS),*, volume 3, pages 57–74. Kluwer, 1996.

[44] Alessandra M. Russo. *Modal Logic as Labelled Deductive Systems.* PhD thesis, Imperial College of Science, Technology and Medicine, London, 1996.

[45] David H. Sanford. *If P then Q, Conditionals and the Foundations of Reasoning.* Routledge, London and New York, 1989.

[46] Robert C. Stalnaker. *A Theory of Conditionals*, volume 2 of *American Philosophical Quarterly Monograph Series (Nicholas Rescher, ed.)*, pages 98–112. Blackwell, Oxford, 1968.

[47] Robert C. Stalnaker and Richmond H. Thomason. A semantic analysis of conditional logic. *Theoria*, 36:23–42, 1970.

[48] Richmond H. Thomason. A Fitch-style formulation of conditional logic. *Logique et Analyse*, 52:397–412, 1970.

[49] Michael Woods. Conditionals. In D. Wiggins, editor, *Conditionals.* Clarendon Press, Oxford, 1997. With a Commentary by Dorothy Edgington.

Received February 2002

# The Unrestricted Combination of Temporal Logic Systems

MARCELO FINGER, *Departamento de Ciência da Computação, Instituto de Matemática e Estatística, Universidade de São Paulo, Rua do Matão 1010, 05508-090 São Paulo, SP, Brazil.*
*E-mail: mfinger@ime.usp.br*

M. ANGELA WEISS, *Departamento de Matemática, Instituto de Matemática e Estatística, Universidade de São Paulo, Rua do Matão 1010, 05508-090 São Paulo, SP, Brazil. E-mail: weiss@ime.usp.br*

## Abstract

This paper generalises and complements the work on combining temporal logics started by Finger and Gabbay [11, 10]. We present proofs of transference of soundness, completeness and decidability for the temporalisation of logics $\mathsf{T}(\mathsf{L})$ for any flow of time, eliminating the original restriction that required linear time for the transference of those properties through logic combination. We also generalise such results to the external application of a multi-modal system containing any number of connectives with arbitrary arity, that respect normality.

This generalisation over generic flows of time propagates to other combinations of logics that can be interpreted in terms of temporalisations. In this way, the *independent combination* (also called *fusion*) of temporal logics is studied over generic flows of time. We show the transfer of soundness, completeness and decidability for independent combination of temporal logics. Finally, we also discuss the independent combination of any finite number of normal multi-modal logics.

*Keywords*: Temporal Logics, Combinations of Logical Systems, Completeness of Combination of Logical Systems, Decidability of Combination of Logical Systems.

## 1 Introduction

This paper is concerned with the study of methods for combining temporal logics. In its first part, we extend the study of the *temporalisation* of logic systems introduced by Finger and Gabbay in [11]. There, the temporalisation process was restricted to linear flows of time. Here, we aim to generalise it to any flow of time. We are interested in studying the *transference of properties* from the logic system $\mathsf{L}$ into its temporalised version $\mathsf{T}(\mathsf{L})$. In the case of linear flows of time, temporalisation was shown to be a useful building block in obtaining the *independent combination* of two temporal logics [10]. In the second part of this paper, we show that the same construction is applicable to any class of flows of time.

The logic system $\mathsf{T}(\mathsf{L})$ combines two logics: a temporal logic $\mathsf{T}$, which is applied *externally* to a given logic system $\mathsf{L}$. This combination process, called *temporalisation*, involves the combination of the languages, inference systems, and model structures of $\mathsf{T}$ and $\mathsf{L}$ into a language, inference system and model structure of $\mathsf{T}(\mathsf{L})$. We show that if the logic systems $\mathsf{T}$ and $\mathsf{L}$ are sound, complete or decidable, then $\mathsf{T}(\mathsf{L})$ is also sound, complete or decidable; no constraints are imposed on the nature of the flow of time.

To show the transference of completeness and decidability via temporalisation, we maintain the same general proof strategy of Finger and Gabbay [11]. However, because here we can no longer rely on the linearity of the flow of time in $\mathsf{T}$, the underlying proof construction has to be almost fully reworked in Section 2.1. For that, we introduce a bound associated to the number of steps in "the past" and the number of steps in "the future" one must take to evaluate a given formula $\psi$ in a temporalised model. This construction allows us to select the "relevant" time points in the evaluation of a formula. As is explained in Section 2.2, the set of "relevant" time points may be infinite, but each point can be reached in finitely many steps. This construction allows us to do without the original restriction of linearity. Our approach naturally leads us to decision procedures. In Section 2.3 we show that provided that $\mathsf{L}$ and $\mathsf{T}$ are decidable, so is $\mathsf{T}(\mathsf{L})$.

We then use these transference results as a building block in the transference of similar properties for the *independent combination* of two temporal logics (also called *fusion* of temporal logics) over any class of flows of time. Section 3 shows that the transference of completeness and decidability can be obtained in terms of unions of alternating temporalisation of two temporal logics; furthermore, we show that such transference occurs even in temporal logics containing the highly expressive binary temporal operators "until" and "since". The mere temporalisation of two $\mathsf{US}$-logics gives us a very limited logic, $\mathsf{US}_1(\mathsf{US}_2)$, which does not allow the nesting of $\mathsf{US}_2$-operators inside the temporal operators of $\mathsf{US}_1$; the independent combination $\mathsf{US}_1 \oplus \mathsf{US}_2$ does allow for any nesting of temporal operators. We explore a property that was initially noted in [10], namely that the *independent combination* $\mathsf{US}_1 \oplus \mathsf{US}_2$ can be seen as the infinite union of several temporalisations $\mathsf{US}_1, \mathsf{US}_1(\mathsf{US}_2)$ and $\mathsf{US}_1(\mathsf{US}_2(\mathsf{US}_1)), \ldots$, and thus we show how the temporalisation results can be employed to obtain the transference of soundness, completeness and decidability for $\mathsf{US}_1 \oplus \mathsf{US}_2$ over generic flows of time.

Combination of logics have been previously analysed in the literature. The first property of independently combined modal logics, namely its conservativity, was presented by Thomason in [25]. Fine and Schurz [7] and Kracht and Wolter [22] have studied the transfer properties of systematically combining independently axiomatisable monomodal systems. The work of Fine and Schurz [7] is applicable to more than two independent normal modalities. A generalisation of such results for many-place multi-modal systems is presented by Wolter in [28]; we discuss in more detail some of Wolter's results in Section 3.

Finger and Gabbay [11, 10] were the first to address the issue of combining logics with two-place modalities, $S$ ("since") and $U$ ("until"), and with modalities that were not all independent, for "since" and "until" interact with each other. The results of [11, 10], however, are restricted to the case of linear flows of time and, because non-linear flows, e.g. over trees or over some other partially ordered sets, often appear in Mathematics as well as in Computer Science, our approach is needed.

## 1.1 Applications of combinations of temporal logics

Since it was initially proposed, temporalisation has been applied in several systems. Its initial application was the description of the evolution of a temporal database [8, 9], which needed two temporal references, one external (evolution) and one internal (the actual temporal database). The two-dimensional view of temporal database evolution is detailed in [15].

In temporal databases, time is generally considered to be linear, which explains the initial focusing on linear flows of time only. Also, linearity simplified the proofs of transference of completeness and decidability, for on a linear time one is allowed to express that "a formula $A$ holds at all times".

Another application of linear temporalisation, involving two-dimensional time, is the work on temporal logic programming within the paradigm of *imperative future* [19, 3]. Such paradigm permitted both the specification of formally verifiable programs as well as the execution of such specifications as a temporal logic program. Its original formulation involved only one temporal dimension, which meant that no update on past states could be done, ie no temporal reasoning was carried on the program itself. To deal with temporal programs, the imperative dimension was applied externally to a system, generating a temporalised two-dimensional version of the imperative future in [12, 13].

Besides temporal databases and software specification, temporalisation was applied in the combination of grammar logics in the work of Blackburn *et al.* [4]. Here, however, the limitations of linearity started to show and the use of temporalisation for grammar formalisms lost preference in the face of other formalisms. Still in the realm of grammar formalisms, Blackburn's and Meyer-Viol's *tree logics* [5] is one possible formalism that can be externally applied to other logics with the results below, but not with the linearity restriction.

More recently, the work of Montanari and Franceschet [17, 16] on the study of structures representing time with multiple granularities has shown that temporalisation can be used to generate logics for several classes of granular structures, even with the linear restriction. It would be interesting to see if new logics would arise if we have more flexibility on the structure of the external flow of time.

As for the independent combination (or fusion) of two modal/temporal logics, several applications arise when combining two logics, the most common of which are the combinations of temporal and knowledge logics for the specification of algorithms and protocols, which are best described in [6]. Of course, the logics needed for practical purposes usually demand a stronger interaction between the component logics than that provided by the independent combination. So the logic obtained by the independent combination is in a sense a *minimal* logic and the addition of further properties and stronger interaction has to be analysed separately. This fact has been noted already in the first works of fusion of logics in [7, 22].

## 1.2 The organisation of this paper

This paper addresses several generalisations. As described above, we aim to generalise the notions of temporalisation and independent combination for generic flows of time. Once such generalisations are done, it is normal to ask if these methods also apply to multi-modal modal and temporal logics, where the connectives may have arbitrary arity. It is our aim here to show how the methods applied here can be extended to this generalised case.

The rest of this paper is organised as follows. Extended temporalisations are studied in Section 2. The basic notions are initially described in Section 2.1, and transference of soundness, completeness and decidability is proven in Sections 2.2 and 2.3. In Section 2.4 it is explained why and how those results are applicable to iterated temporalisations, as a prelude to the analysis of the independent combination in Section 3. Those results all concern temporal $US$-logics, and generalising them form multi-modal logics is the aim of Section 2.5.

The study of the independent combination of temporal logics starts with general definitions in 3 and the transference of basic properties in Sections 3.2, 3.3 and 3.4. Finally, the generalisation of the independent combination of any number of multi-modal logics is discussed in 3.5.

Section 4 concludes with a discussion on the relationship between the constructions of Sections 2.1 and 3, and an open problem is reported.

## 2 The temporalised system $\mathsf{T}(\mathsf{L})$

### *2.1 Definition of a temporalised system*

In this section we describe the system $\mathsf{T}(\mathsf{L})$ introduced in [11]. By a *logic system* we mean a *quadruple* $S = (\mathcal{L}_S, \vdash_S, \mathcal{K}_S, \models_S)$, where $\mathcal{L}_S$ is the system's language, $\vdash_S$ is an inference system, $\mathcal{K}_S$ is a class of models for the system and $\models_S \subseteq \mathcal{K}_S \times \mathcal{L}_S$ is a semantic relation such that $M \models_S A$ means that the formula $A \in \mathcal{L}_S$ is satisfied in the model $M \in \mathcal{K}_S$.

### The language of $\mathsf{T}(\mathsf{L})$

The language $\mathcal{L}_{\mathsf{US}}$ of the temporal system $\mathsf{T}$ is built from a denumerable set of atoms $\mathcal{A}$, applying the two-place modalities $U$ (*until) and* $S$*, (*since), and the Boolean connectives $\neg$ (*negation*) and $\wedge$ (*conjunction*).

Very little is required of the internal logic $\mathsf{L}$, except that its language is described from a denumerable set of *atoms* and that it has the classical Boolean connectives $\neg$ and $\wedge$. We also demand that the connectives of $\mathsf{T}$ and $\mathsf{L}$ be disjunct. Apart from that, any other type of modalities or predicates are accepted in the language.

Before we define the language of the *temporalised system* $\mathsf{T}(\mathsf{L})$ we need to introduce a few definitions.

The language $\mathcal{L}_{\mathsf{L}}$ of $\mathsf{L}$ is partitioned into the sets $BC_{\mathsf{L}}$ and $ML_{\mathsf{L}}$, where:

- $BC_{\mathsf{L}}$, the set of *Boolean combinations* consists of the formulas built up from *any* other formulas with the use of the Boolean connectives $\neg$ or $\wedge$;
- $ML_{\mathsf{L}}$, the set of *monolithic formulas* is the complementary set of $BC_{\mathsf{L}}$ in $\mathcal{L}_{\mathsf{L}}$.

If the external logic $\mathsf{L}$ does not contain the classical connectives $\neg$ and $\wedge$, we assume that $ML_{\mathsf{L}} = \mathcal{L}_{\mathsf{L}}$ and $BC_{\mathsf{L}} = \emptyset$, so every formula in $\mathsf{L}$ is considered monolithic.

The set of temporalised formulas, $\mathcal{L}_{\mathsf{T}(\mathsf{L})}$, is defined as the smallest set closed under the rules

1. If $A \in ML_{\mathsf{L}}$, then $A \in \mathcal{L}_{\mathsf{T}(\mathsf{L})}$;
2. If $A, B \in \mathcal{L}_{\mathsf{T}(\mathsf{L})}$, then $\neg A \in \mathcal{L}_{\mathsf{T}(\mathsf{L})}$ and $A \wedge B \in \mathcal{L}_{\mathsf{T}(\mathsf{L})}$;
3. If $A, B \in ML_{\mathsf{L}}$, then $S(A, B) \in \mathcal{L}_{\mathsf{T}(\mathsf{L})}$ and $U(A, B) \in \mathcal{L}_{\mathsf{T}(\mathsf{L})}$.

We say that a formula in $\mathcal{L}_{\mathsf{T}(\mathsf{L})}$ is *monolithic* if it is a formula that is in the language of $\mathsf{L}$ that is monolithic in $\mathsf{L}$.

Note that the atoms of $\mathcal{L}_{\mathsf{US}}$ are not elements of $\mathcal{L}_{\mathsf{T}(\mathsf{L})}$. As an example of a temporalised language, consider the atoms $p, q \in \mathcal{L}_{\mathsf{L}}$ and $\Box$ is a modal symbol in $\mathsf{L}$, then $\Box p$ and $\Box(p \wedge q)$ are monolithic formulas whereas $\neg\Box p$ and $\Box p \wedge \Box q$ are two Boolean combinations.

The *mirror image* of a given formula is given by replacing $U$ by $S$ and vice-versa. We will use the connectives $\vee$ and $\rightarrow$ and the constants $\top$ and $\bot$ in its usual meaning. Also, the formulas $PA$, $FA$, $GA$ and $HA$ abbreviate respectively $S(A, \top)$, $U(A, \top)$, $\neg F(\neg A)$ and $\neg P(\neg A)$. The *complexity* of a formula $A$ is the cardinality of its subformulas.

## The semantics of $\mathsf{T}(\mathsf{L})$

A *flow of time* is a pair $(T, <)$ where $T$ is a set of time points and $<$ is a binary relation on $T$. By imposing restrictions on $<$ we generate *classes of flows of time*, e.g. the class $\mathcal{K}_{lin}$ of all transitive, irreflexive and linear flows of time.

When dealing with a simple temporal logic, a model is a triple $(T, <, h)$, where $(T, <)$ is a flow of time and $h : T \rightarrow 2^{\mathcal{P}}$ is a mapping that associates every time point $t \in T$ with a set of propositions, namely with the set of propositions that are true at that point. If we restrict the class of flows of time to $\mathcal{K}'$, we also restrict the class of models; it is usual practice to also call this class of models $\mathcal{K}'$, leaving the context to disambiguate whether we mean the class of flows of time or the class of models.

This definition of temporal model indicates that every time point is mapped into a classical propositional model, and such a view will be generalised in the temporalised case.

For that, we have to specify some restrictions on the semantic relation $\models_L$ for the logic $\mathsf{L}$, whose class of models will be called $\mathcal{K}_L$. The basic restriction imposes that, for each $\mathcal{M} \in \mathcal{K}_L$ and $A \in \mathcal{L}_L$ we have

$$\text{either } \mathcal{M} \models A \text{ or } \mathcal{M} \models \neg A. \quad (*)$$

This may need some adaptation on the notion of class of model. For instance, if $\mathsf{L}$ is modal logic $S5$, it is not the case that, for each Kripke frame $(W, R)$ where $R$ is an equivalence relation and every modal valuation $V$, either $W, R, V \models A$ or $W, R, V \models \neg A$. However, this problem is solved if we consider as the class of models the set of pairs $\langle \mathcal{M}, w \rangle$, where $\mathcal{M}$ is an $S5$ model $(W, R, V)$ and $w \in W$. For that class of models the property $(*)$ above holds.

Let $(T, <)$ be a flow of time and let $g$ be a mapping from $T$ into $\mathcal{K}_L$, such that $(*)$ holds for $g(t)$, for all $t \in T$. A triple $\mathcal{M}_{\mathsf{T}(\mathsf{L})} = (T, <, g)$ is a *temporalised model* of $\mathsf{T}(\mathsf{L})$. We say that a temporal model $(T, <, g)$ belongs to a class $\mathcal{K}$ iff $(T, <) \in \mathcal{K}$. In general, we will use the term temporalised model to refer to a model of $\mathsf{T}(\mathsf{L})$ and temporal model to refer to a model of $\mathsf{T}$.

The satisfaction relation $\models$ is defined recursively over structure of temporalised formulas:

1. $\mathcal{M}_{\mathsf{T(L)}}, t \models A$, $A \in ML_{\mathsf{L}}$, iff $g(t) = \mathcal{M}_L$ and $\mathcal{M}_L \models A$;
2. $\mathcal{M}_{\mathsf{T(L)}}, t \models \neg A$ iff $\mathcal{M}_{\mathsf{T(L)}}, t \not\models A$;
3. $\mathcal{M}_{\mathsf{T(L)}}, t \models A \wedge B$ iff $\mathcal{M}_{\mathsf{T(L)}}, t \models A$ and $\mathcal{M}_{\mathsf{T(L)}}, t \models B$;
4. $\mathcal{M}_{\mathsf{T(L)}}, t \models S(A, B)$ iff there exists $s < t$ such that $\mathcal{M}_{\mathsf{T(L)}}, s \models A$ and for all $r$, $s < r < t$, $\mathcal{M}_{\mathsf{T(L)}}, r \models B$;
5. $\mathcal{M}_{\mathsf{T(L)}}, t \models U(A, B)$ iff there exists $t < s$ such that $\mathcal{M}_{\mathsf{T(L)}}, s \models A$ and for all $r$, $t < r < s$, $\mathcal{M}_{\mathsf{T(L)}}, r \models B$.

A formula is *valid* in a class $\mathcal{K}$ if it is verified at all times at all models over that class.

### The inference system of $\mathsf{T(L)}$

We assume that an *inference system* for a generic logic system is a mechanism capable of recursively enumerating the set of all provable formulas of the system, here called *theorems* of the logic system.

An inference system is *sound* with respect to a class of models $\mathcal{C}$ if all its theorems are valid over $\mathcal{C}$. Conversely, it is *complete* if all valid formulas are theorems. We assume that $\mathsf{L}$'s inference system is sound and complete.

We will assume that the temporal logic $\mathsf{T}$'s inference system is given in an axiomatic form, consisting of a set of *axioms* and a set of inference rules. For example, consider the possible axiomatisations of $\mathsf{US}$ over several classes of flows of time presented in [29] or in [20]. When a temporal logic $\mathsf{T}$ is sound and complete over the class $\mathcal{K}$ of flows, we write $\mathsf{T}/\mathcal{K}$.

Given $\mathsf{T}/\mathcal{K}$, the inference system of $\mathsf{T(L)}$ is denoted by $\mathsf{T(L)}/\mathcal{K}$ and consists of the following elements:

- The axioms of $\mathsf{T}/\mathcal{K}$;
- The inference rules of $\mathsf{T}/\mathcal{K}$;
- The inference rule *Preserve*: For every formula $\varphi$ in $\mathcal{L}_{\mathsf{L}}$, if $\vdash_L \varphi$ then $\vdash_{T(L)} \varphi$.

In [11] it is shown that if $\mathsf{T}/\mathcal{K}$ and $\mathsf{L}$ have a sound inference system, then the inference system of $\mathsf{T(L)}/\mathcal{K}$ is sound; no extra restrictions are made on the nature of $\mathcal{K}$. Also, in case $\mathsf{L}$ has a complete inference system and $\mathcal{K}_{lin}$ is a class of linear flows of time, then the inference system of if $\mathsf{T(L)}/\mathcal{K}_{lin}$ is complete. We want to eliminate this restriction on linearity.

## *2.2 Completeness of* $\mathsf{T(L)}$

To show the transference of completeness we maintain the same proof strategy of [11], but we introduce a new technique and rework its underlying constructions. In the presence of linearity, one can write a formula that expresses the fact that a formula $A$ "is true everywhere" in a model. This simplifies life a lot, but cannot be reproduced in a generic model. So we introduce a technique that picks up the "relevant" worlds in a model for the evaluation of a given formula, and we construct a formula that forces $A$ to be true over all such relevant worlds.

The strategy of the proof is illustrated in Figure 1. We start with a consistent $\mathcal{L}_{\mathsf{T(L)}}$-formula $\varphi$, translate it into a pure $\mathcal{L}_{\mathsf{US}}$-temporal logic consistent formula $A$; then

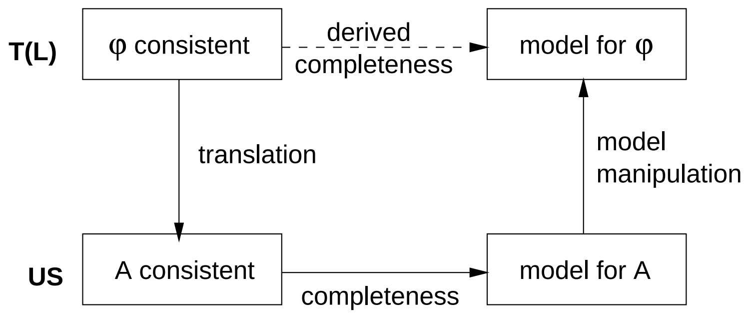


FIG. 1. Completeness proof strategy

completeness of $\mathcal{L}_{\mathsf{US}}/\mathcal{K}$ gives us a model for $A$; after some model manipulation using the completeness of L, we obtain a $\mathsf{T}(\mathsf{L})/\mathcal{K}$-model for $\varphi$, thus deriving the completeness for $\mathsf{T}(\mathsf{L})/\mathcal{K}$. The more sophisticated bit of the proof is the initial translation step, which in the generic case has to deal with the nesting of temporal operators in $\varphi$ instead of the simpler translation used for the linear case. Such initial elaboration allows us later to do a straightforward model manipulation to construct a model for $\varphi$.

To deal with the nesting of temporal operators in a formula, we define the *operator nesting tree* of a temporal or temporalised formula $\psi$, $D_\psi$. A tree is represented here as a set of strings of 0's and 1's, with the symbol $*$ representing concatenation of strings; the empty string is represented by $\varepsilon$. The tree is closed under prefix formation of its strings, that is, if $101 \in D_\psi$, then $\varepsilon, 1, 10 \in D_\psi$ as well. The 0 represents a past operator (a step to the past) and the 1 represents a future operator (or a step to the future).

**Notation 2.1** In the following we will use the Greek letters $\varphi$, $\psi$ and $\chi$ to indicate T(L) formulas, and the letters $A$, $B$ and $C$ to indicate temporal US formulas. We use the Greek letters $\varphi$, $\psi$ and $\chi$ also to refer to either a temporal or temporalised formula.

**Definition 2.2** Given a formula $\psi \in \mathcal{L}_{\mathsf{US}} \cup \mathcal{L}_{\mathsf{T}(\mathsf{L})}$ we build its *operator nesting tree* $D_\psi$ recursively over the structure of $\psi$:

1. If $\psi$ is a literal or monolithic, then $D_\psi = \{\varepsilon\}$;
2. If $\psi = \varphi_1 \wedge \varphi_2$, then $D_\psi = D_{\varphi_1} \cup D_{\varphi_2}$;
3. If $\psi = \neg\varphi$, then $D_\psi = D_\varphi$;
4. If $\psi = S(\varphi_1, \varphi_2)$, then $D_\psi = \{\varepsilon\} \cup \{0 * s | s \in D_{\varphi_1} \cup D_{\varphi_2}\}$;
5. If $\psi = U(\varphi_1, \varphi_2)$, then $D_\psi = \{\varepsilon\} \cup \{1 * s | s \in D_{\varphi_1} \cup D_{\varphi_2}\}$.

This definition implies that $\varepsilon \in D_\psi$ for any $\psi$ and, as a consequence, the prefix of any string in $D_\psi$ will also be a member of $D_\psi$. For example, consider the $US$ formula

$$A = S(U(p, S(p, q)), S(p, p)) \wedge U(\neg U(p, q), S(p, q))$$

It's associated operator nesting tree will be:

$D_A = D_{S(U(p,S(p,q)),S(p,p))} \cup D_{U(\neg U(p,q),S(p,q))}$,
$D_A = \{\varepsilon\} \cup \{0 * s | s \in D_{U(p,S(p,q))} \cup D_{S(p,p)}\} \cup \{1 * r | r \in D_{U(p,q)} \cup D_{S(p,q)}\}$,
$D_A = \{\varepsilon, 0, 1\} \cup \{01 * s' | s' \in D_p \cup D_{S(p,q)}\} \cup \{00 * s'' | s'' \in D_p\} \cup$
$\{11 * r' | r' \in D_p \cup D_q\} \cup \{10 * r'' | r'' \in D_p \cup D_q\}$,
$D_A = \{\varepsilon, 0, 1, 01, 00, 11, 10\} \cup \{010 * s''' | s''' \in \cup D_p \cup D_q\}$ ,
$D_A = \{\varepsilon, 0, 1, 01, 00, 11, 10, 010\}$.

Let $1^m$ represent a string of $m$ 1's, and similarly for $0^m$. Let $0^0$ and $1^0$ represent the empty string. So each string in the nesting operator can be represented as $1^{m_1}0^{m_2}\ldots1^{m_{n-1}}0^{m_n}$, where all $m_i > 0$, except for $m_1$ and $m_n$, that can be 0. Note that $n$ is always an even number.

Each such string is then associated to a temporal operator over $H$ and $G$. Let $H^0\psi = G^0\psi = \psi$; let $G^{n+1}\psi = G(G^n\psi)$; and $H^{n+1}\psi = H(H^n\psi)$. So each string $1^{m_1}0^{m_2}\ldots1^{m_{n-1}}0^{m_n}$ is associated with the temporal operator $G^{m_1}H^{m_2}\ldots G^{m_{n-1}}H^{m_n}$, which we abbreviate as $\Box_{m_1,m_2,\ldots,m_{n-1},m_n}$.

As an example, $\Box_{0,2}(\Box_{0,3,1,0}\psi) \equiv \Box_{0,5,1,0}\psi$ instead of $\Box_{0,2,0,3,1,0}\psi$.

We can now start defining the translation of *consistent* formulas in $\mathsf{T}(\mathsf{L})$ into *consistent* formulas in $\mathsf{US}$. The first step is the *correspondence mapping*.

**Definition 2.3** Let $\{p_1, p_2, \ldots\}$ be an enumeration of the set of atoms of $\mathsf{US}$, and let $\{\psi_1, \psi_2, \ldots\}$ be an enumeration of $ML_\mathsf{L}$, the set of monolithic formulas of $\mathsf{T}(\mathsf{L})$. Define the *correspondence mapping* $\sigma$ from $\mathcal{L}_{\mathsf{T}(\mathsf{L})}$ into $\mathcal{L}_{\mathsf{US}}$, inductively over a formula as:

$$\begin{array}{rcl} (\forall\psi_i \in ML_\mathsf{L})(\sigma(\psi_i)) &=& p_i \\ \sigma(\neg\chi) &=& \neg\sigma(\chi) \\ \sigma(\chi_1 \wedge \chi_2) &=& \sigma(\chi_1) \wedge \sigma(\chi_2) \\ \sigma(S(\chi_1, \chi_2)) &=& S(\sigma(\chi_1), \sigma(\chi_2)) \\ \sigma(U(\chi_1, \chi_2)) &=& U(\sigma(\chi_1), \sigma(\chi_2)) \end{array}$$

The following two lemmas are shown in [11]:

**Lemma 2.4 (The correspondence Lemma)** The correspondence mapping $\sigma$ is a bijection.

**Lemma 2.5** For all $\mathsf{T}(\mathsf{L})$-consistent $\chi \in \mathcal{L}_{\mathsf{T}(\mathsf{L})}$, $\sigma(\chi)$ is $\mathsf{US}$-consistent.

The reverse of Lemma 2.5 is not true, as we can see in this example:

**Example 2.6** In a modal normal logic with the modality $\Box$, for all atoms $\varphi, \psi$,

$$\chi \equiv \Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$$

is a theorem in $\mathsf{L}$. The formulas $\Box(\varphi \to \psi)$, $\Box\varphi$ and $\Box\psi$ are monolithic, so they are mapped by $\sigma$ into some atoms of $\mathsf{US}$, say $p_1$, $p_2$ and $p_3$, respectively.

Thus, $\sigma(\chi) = p_1 \to (p_2 \to p_3)$, that is not a theorem in $\mathsf{T}$.

For the model manipulation in the final part of the proof of completeness, we will need also the converse of Lemma 2.5, that is, $\mathsf{T}(\mathsf{L})$ theorems must be mapped into $\mathsf{US}$ theorems. To achieve that, we define the transformation $\eta(\psi)$, which makes use of the operator nesting tree $D_\psi$, and preserves $\psi$'s consistency; such transformation will guarantee that $\mathsf{T}(\mathsf{L})$-theorems are mapped into $\mathsf{US}$-theorems.

**Definition 2.7** Given two formulas $\varphi, \psi \in \mathcal{L}_{\mathsf{T}(\mathsf{L})}$, define:

1. $Mon(\varphi)$ is the set of monolithic subformulas of $\varphi$.
2. $Lit(\varphi) = Mon(\varphi) \cup \{\neg\psi | \psi \in Mon(\varphi)\}$;
3. $Inc(\varphi) = \{\bigwedge F | F \subseteq Lit(\varphi) \text{ and } F \vdash_{\mathsf{L}} \bot\}$; that is $Inc(\varphi)$ is the set of L-inconsistent formulas that can be built using the monolithic subformulas of $\varphi$;
4. $\Box_\varphi\psi$ is the conjunction of all formulas of the form $\Box_{m_1,\ldots,m_n}\psi$ such that $\Box_{m_1,\ldots,m_n}$ is a temporal operator associated to a string in the operator nesting tree $D_\varphi$;
5. $\eta(\varphi) = \bigwedge\{\Box_\varphi\neg\psi | \psi \in Inc(\varphi)\}$.

**Example 2.8** If $\varphi = S(p,q)$, then $D_\varphi = \{\varepsilon, 0\}$. So, for any formula $\psi$, $\Box_\varphi\psi = \Box_{0,0}\psi \wedge \Box_{0,1}\psi = \psi \wedge H\psi$.

The terminology used in Definition 2.7 was introduced in [11]. The modification for the general case we had to make here is restricted to the definition of $\Box_\varphi\psi$ (used in the definitions of $\eta(\varphi)$).

The following Lemma is an adaptation of [11] for a the case of generic flows of time.

**Lemma 2.9** $\vdash_{\mathsf{T}(\mathsf{L})} \eta(\psi)$.

Proof. Every formula $\varphi$ in $Inc(\psi)$ is a contradiction, and therefore its negation is a theorem of $\mathsf{T}(\mathsf{L})$. Now, if $\neg\varphi$ is a theorem, so are $H\neg\varphi$ and $G\neg\varphi$; by induction we get that $\Box_{m_1,\ldots,m_n}\neg\varphi$ is a theorem too, for any $m_1, \ldots, m_n$. ■

Using Lemmas 2.5 and 2.9, we have that if $\psi$ is $\mathsf{T}(\mathsf{L})$-consistent, then $\sigma(\psi \wedge \eta(\psi))$ is US-consistent. We can apply completeness of $\mathsf{US}/\mathcal{K}$ and obtain a US-model $\mathcal{M}_{\mathsf{US}}$ for $\sigma(\psi \wedge \eta(\psi))$ over $\mathcal{K}$. Furthermore, the theoremhood of the monolithic L-formulas in $\psi$ is captured in $\eta(\psi)$ and will guarantee that its translation will be true in the "relevant part" of $\mathcal{M}_{\mathsf{US}}$. It is this notion of "relevant part" of a temporal model that we define next by associating subflows of time to binary trees (not very surprisingly). At this part of the proof we will be working at the US level.

Let $(T,<) \in \mathcal{K}$ be a flow of time, and let $t, s \in T$. We say that $s$ is 1-related to $t$ if $t < s$ ($s$ is in the future of $t$); similarly, $s$ is 0-related to $t$ if $s < t$ ($s$ is in the past of $t$). Let $t_1, \ldots, t_n \in T$ be a sequence of time points such that each pair $t_i, t_{i+1}$ is 0- or 1-related. Such a sequence can then be associated to a string of 0's and 1's of length $n-1$, where the $i$th position is 1 if $t_i$ and $t_{i+1}$ are 1-related, and 0 otherwise; we represent it as $\mathsf{string}(t_1, \ldots, t_n)$.

The "relevant part" of a flow of time $(T,<)$, with respect to a temporal formula $A$ at a point $t$, is formally defined as the *range of $A$ at $t$ over* $(T,<)$, $Rg(A,t)$:

$$Rg(A,t) = \{t\} \cup \{s \in T \mid \mathsf{string}(t, t_1, \ldots, t_n, s) \in D_A \text{ for some } t_1, \ldots, t_n \in T\}$$

Note that since $D_A = D_{\neg A}$, it follows that $Rg(A,t) = Rg(\neg A, t)$.

It is important to highlight that we are *not* constructing a submodel of a given model generated by $Rg(A,t)$. Our aim is to construct a model that belongs to a class $\mathcal{K}$. If we start in a model over $\mathcal{K}$ and generate a submodel based on $Rg(A,t)$, there is no way to guarantee that the generated submodel belongs to $\mathcal{K}$, and in general it does not. So $Rg(A,t)$ will be used to focus on a relevant part of the model. The satisfaction of a formula $A$ at a point $t$ in a temporal model depends only on the temporal valuation at points in $Rg(A,t)$, as shown below.

**Lemma 2.10** Consider a temporal model $\mathcal{M} = (T, <, g)$, a formula $A \in \mathcal{L}_{\mathsf{US}}$, and a point $t \in T$. Then for any model $\mathcal{M}' = (T, <, g')$ such that $g'(s) = g(s)$ for every $s \in Rg(A, t)$,

$$\mathcal{M}, t \models A \textit{ iff } \mathcal{M}', t \models A.$$

Proof. Initially note that, both $\mathcal{M}$ and $\mathcal{M}'$ are based on the same flow of time, so for every subformula $B$ of $A$ and for every $s \in T$, $Rg(B, s)$ is the same set for both models. We proceed by structural induction over $A$.

- If $A$ is atomic, then $g(t) = g'(t)$.
- If $A = \neg B$, then $Rg(A, t) = Rg(B, t)$, so the induction hypothesis directly gives us the result.
- If $A = B_1 \wedge B_2$, then $Rg(A, t) = Rg(B_1, t) \cup Rg(B_2, t)$, and therefore for every $s \in Rg(B_i, t)$, $g(t) = g'(t)$ $[i = 1, 2]$, so the induction hypothesis applies and gives us that $\mathcal{M}, t \models B_i$ *iff* $\mathcal{M}', t \models B_i$, from which the result follows immediately.
- If $A = S(B, C)$, then $\mathcal{M}, t \models A$ iff there exists a $t' < t$ with $\mathcal{M}, t' \models B$ and for every $t''$ such that $t' < t'' < t$, $\mathcal{M}, t'' \models C$. Note that both $t', t'' \in Rg(A, t)$. Furthermore, because the temporal nesting of $B$ and $C$ is smaller than that of $A$, we have $Rg(B, t') \subseteq Rg(A, t)$ and therefore $g(s) = g(s')$ for every $s \in Rg(B, t')$, so the induction hypothesis applies and yields $\mathcal{M}, t' \models B$ iff $\mathcal{M}', t' \models B$; analogously, we get that for every $t''$ such that $t' < t'' < t$, $\mathcal{M}, t'' \models C$ iff $\mathcal{M}', t' \models C$, and therefore the result follows.
- If $A = U(B, C)$ the reasoning is totally analogous to the previous case, finishing the proof.

∎

The following lemma shows that the definition of $\eta(\psi)$ preserves $\psi$'s truth value over that "relevant part" of a model.

**Lemma 2.11** Let $\mathcal{M}_{\mathsf{US}} = (T, <, g)$ be a temporal model over $\mathcal{K}$ and $\varphi, \psi \in \mathcal{L}_{\mathsf{T(L)}}$. Let $t \in T$ so that $\mathcal{M}_{\mathsf{US}}, t \models \sigma(\Box_\varphi \psi)$. Then for every $s \in Rg(\sigma(\varphi), t)$, $\mathcal{M}_{\mathsf{US}}, s \models \sigma(\psi)$.

Proof. We know that

$$\Box_\varphi \psi = \bigwedge_{1^{m_1} \ldots 0^{m_n} \in D_\varphi} \Box_{m_1, \ldots, m_n} \psi.$$

A simple induction shows that $D_\varphi = D_{\sigma(\varphi)}$, and therefore

$$\sigma(\Box_\varphi \psi) = \bigwedge_{1^{m_1} \ldots 0^{m_n} \in D_{\sigma(\varphi)}} \Box_{m_1, \ldots, m_n} \sigma(\psi).$$

Consider $s \in Rg(\sigma(\varphi), t)$. Then either $s = t$ or there are $t_1, \ldots, t_n \in Rg(\sigma(\varphi), t)$ such that $\mathsf{string}(t, t_1, \ldots, t_n, s) \in D_{\sigma(\varphi)}$. If $s = t$, since $\varepsilon \in D_{\sigma(\varphi)}$, it follows that $\mathcal{M}_{\mathsf{US}}, s \models \sigma(\psi)$. In the latter case, we show the result by induction on $n$.

For $n = 0$, we have that either $s < t$, in which case we have that $\mathcal{M}_{\mathsf{US}}, t \models H\sigma(\psi)$ so $\mathcal{M}_{\mathsf{US}}, s \models \sigma(\psi)$, or $t < s$, in which case we have that $\mathcal{M}_{\mathsf{US}}, t \models G\sigma(\psi)$ so $\mathcal{M}_{\mathsf{US}}, s \models \sigma(\psi)$.

For the inductive case, we have that $\mathsf{string}(t, t_1, \ldots, t_n, s) \in D_{\sigma(\varphi)}$. Again we have two possibilities. If $t_n < s$ then the rightmost operator in $\Box_{m_1,\ldots,m_n}$ is a $G$, and the induction hypothesis gives us that $\mathcal{M}_{\mathsf{US}}, t_n \models G\sigma(\psi)$ so $\mathcal{M}_{\mathsf{US}}, s \models \sigma(\psi)$. If $s < t_n$ then the rightmost operator in $\Box_{m_1,\ldots,m_n}$ is an $H$, and the induction hypothesis gives us that $\mathcal{M}_{\mathsf{US}}, t_n \models H\sigma(\psi)$ so $\mathcal{M}_{\mathsf{US}}, s \models \sigma(\psi)$.

This finishes the induction and the proof. ∎

We can now finally glue the pieces of the completeness proof.

**Theorem 2.12** If the logical system L is complete and US is complete over a class of flows of time $\mathcal{K}$, then the logical system T(L) is complete over $\mathcal{K}$.

Proof. Let $\psi$ be a $\mathsf{T(L)}/\mathcal{K}$-consistent formula. We will construct a T(L)-model for $\psi$ over the class $\mathcal{K}$.

By Lemma 2.9, $\psi \wedge \eta(\psi)$ is also a T(L)-consistent formula. So, by Lemma 2.5, $\sigma(\psi \wedge \eta(\psi))$ is a T-consistent formula. As we assume that $\mathsf{US}/\mathcal{K}$ is complete, then there exists a temporal model $\mathcal{M}_{\mathsf{US}} = (T, <, g)$ with $(T, <) \in \mathcal{K}$ such that for some $t \in T$, $\mathcal{M}_{\mathsf{US}}, t \models \sigma(\psi \wedge \eta(\psi))$. For every $s \in Rg(\psi, t)$, define:

$$G_\psi(s) = \{\varphi \in Lit(\psi) | \mathcal{M}_{\mathsf{US}}, s \models \sigma(\varphi)\}$$

**Claim:** For every $s \in Rg(\psi, t)$, $G_\psi(s)$ is finite and L-consistent.

Indeed, $G_\psi(s)$ is finite because $Lit(\psi)$ is finite. To prove consistency, suppose by absurd that for some $s \in T$, $G_\psi(s)$ is L-inconsistent. Then there exists a subset of $G_\psi(s)$, $\{\varphi_1, \ldots, \varphi_n\}$ such that $\vdash_{\mathsf{L}} \bigwedge_{1 \le i \le n} \varphi_i \to \bot$. Thus $\bigwedge_{1 \le i \le n} \varphi_i \in Inc(\psi)$.

Let $\xi = \Box_\psi \sigma(\neg \bigwedge_{1 \le i \le n} \varphi_i)$. By the definition of $\eta$ and $\sigma$, it follows that $\xi$ is a conjunct of $\sigma(\psi \wedge \eta(\psi))$. From $\mathcal{M}_{\mathsf{US}}, t \models \sigma(\psi \wedge \eta(\psi))$ it follows $\mathcal{M}_{\mathsf{US}}, t \models \xi$, so by Lemma 2.11 $\mathcal{M}_{\mathsf{US}}, s \models \neg\sigma(\bigwedge_{1 \le i \le n} \varphi_i)$. However, by the definition of $G_\psi(s)$ we have that $\mathcal{M}_{\mathsf{US}}, s \models \bigwedge_{1 \le i \le n} \sigma(\varphi_i) = \sigma(\bigwedge_{1 \le i \le n} \varphi_i)$, which is clearly a contradiction.

Therefore $G_\psi(s)$ is always L-consistent, proving the claim.

This claim is then used to build a model for $\psi$ in the following way. By Lemma 2.11, for each $s \in Rg(\psi, t)$, $\mathcal{M}_{\mathsf{US}}, s \models \sigma(G_\psi(t))$. By hypothesis, L is complete, so for each $s \in Rg(\psi, t)$ there exists a model for the L-consistent set $G_\psi(s)$, $\mathcal{M}^s_{\mathsf{L}}$. So, we can define a valuation $h$ as:

$$h(s) = \mathcal{M}^s_{\mathsf{L}}$$

for every $s \in Rg(\psi, t)$; for $s \in T - Rg(\psi, t)$, $h(s)$ can be any model of L.

Consider $\mathcal{M}_{\mathsf{T(L)}} = (T, <, h)$. To obtain completeness, all we have to do is to prove that $\mathcal{M}_{\mathsf{T(L)}}, t \models \psi$. First, note that for every $s \in Rg(\psi, t)$, and every monolithic subformula $B$ of $\psi$, $\mathcal{M}_{\mathsf{T(L)}}, t \models \varphi$ iff $\mathcal{M}_{\mathsf{US}}, t \models \sigma(\varphi)$. Then a straightforward structural induction on $\varphi$ generalises this to show that $\mathcal{M}_{\mathsf{T(L)}}, t \models \psi$ iff $\mathcal{M}_{\mathsf{US}}, t \models \sigma(\psi)$; details omitted.

But since we have that $\mathcal{M}_{\mathsf{US}}, t \models \sigma(\psi)$, it follows that $\mathcal{M}_{\mathsf{T(L)}}$ is a temporalised model for $\psi$ over $\mathcal{K}$, finishing the proof. ∎

The following is a nice consequence of soundness and completeness. Let $Th(\mathsf{L})$ be the set of all theorems of logic L. A logic L′ is an *extension* of logic L if $Th(\mathsf{L}) \subseteq Th(\mathsf{L}')$. Furthermore, such an extension is said to be *conservative* if the language of L′ is a superset of the language of L and for every formula $A$ in the language of L, $A \in Th(\mathsf{L}')$ only if $A \in Th(\mathsf{L})$.

**Lemma 2.13** If the systems US and L are sound and complete, with disjoint sets of connectives, then the temporalised system US(L) is a conservative extension of both US and L.

PROOF. It is obvious from the definition of US(L) that it is an extension of both US and L.

For conservativeness, suppose $A$ is a formula of US that is a theorem of US(L). Suppose for contradiction that $A$ is not a theorem of US, $\not\vdash_{US} A$. Then, since US extends classical logic, we have that $\neg A$ is a consistent formula, and by completeness, we have a model $\mathcal{M}$ for $\neg A$. We construct a temporalised model $\mathcal{M}_{\mathsf{T(L)}} = (T, <, g)$ such that $g(t) = \mathcal{M}$ for every $t \in T$. Such a model is indeed T(L)-countermodel of $A$, contradicting the soundness of T(L). So $A$ is a theorem of US.

If $A$ is a formula of L, because the sets of connectives are disjoint, the only way it can be deduced is via the inference rule of *Preserve*, in which case it clearly is a theorem of L. Which finishes the proof. ∎

**Remark 2.14** Note that the last step of the proof above relies on the fact that the languages of L and US are *disjoint*. If there is a connective of L that also appears in US, the result above would not hold. This seems a vacuous assertion, since we have always assume the disjunction of the languages in this section, but it will be particularly important when we discuss the decomposition of the independent composition in several temporalisations.

## *2.3 Decidability of* T(L)

Transference of decidability is shown in [11] conditioned to the underlying flow of time being linear. We extend here that result to any class of flows of time. Recall that a given system L is *decidable* if for any formula $\psi \in$ L, there exists a procedure that outputs "yes" if $\psi$ is a theorem and "no" otherwise. So, if L is complete then L is decidable if for any formula $\psi \in$ L, it is possible to decide whether $\psi$ is valid or not.

Let us suppose that both the temporal system T and the external system L are decidable. We assume that both T and L are sound and complete. Then, T(L) is also sound and complete, decidability is obtainable if we can decide the validity of a T(L) formula $\psi$ in any temporalised model.

The transference of decidability is obtained through a construction similar to that used for completeness. The definitions of $\eta(\psi)$ and the mapping $\sigma$ are the same. We have:

**Lemma 2.15** Let L and T be sound and complete systems. A formula $\psi$ is T(L)-valid iff $\sigma(\eta(\psi) \to \psi)$ is US-valid.

PROOF. If $\sigma(\eta(\psi) \to \psi)$ US-valid, by US-completeness it is also a theorem, then we can simply mimic the US-proof at the temporalised level, since all US axioms and inference rules are present at T(L), so $\eta(\psi) \to \psi$ is also a T(L)-theorem. And since, by construction, $\eta(\psi)$ is always a theorem, so is $\psi$. By derived soundness, it is also T(L)-valid.

Suppose by contradiction that $\psi$ is T(L)-valid and $\sigma(\eta(\psi) \to \psi)$ is not valid. From Lemmas 2.10 and 2.11 follows that if there was a countermodel for $\sigma(\eta(\psi) \to \psi)$, we would be able to construct a countermodel for $\eta(\psi) \to \psi$, and thus also a countermodel for $\psi$, which contradicts completeness. So $\sigma(\eta(\psi) \to \psi)$ must be US-valid. ∎

It is simple now to see the transference of decidability.

**Theorem 2.16** If $\mathsf{T}$ and $\mathsf{L}$ are sound, complete and decidable, $\mathsf{T}(\mathsf{L})$ is decidable.

PROOF. From the definition of $\eta(\psi)$, if $\mathsf{L}$ is decidable then we have a direct way to construct $\eta(\psi)$. From Lemma 2.15, it follows that the decision of $\psi$ is equivalent to the decision of $\sigma(\eta(\psi) \to \psi)$. Since such a formula is constructible, we can apply the decision procedure of $\mathsf{T}$, thus deciding $\psi$. ∎

It is straightforward to show the following complexity result:

**Lemma 2.17** Let $N$ be the size of a formula, and let $\mathcal{O}(c_{\mathsf{L}}(N))$ and $\mathcal{O}(c_{\mathsf{US}}(N))$ be upper bounds of the complexity of the decision procedures of $\mathsf{L}$ and $\mathsf{T}$, respectively. Then an upper bound of the complexity of the decision procedure for $\mathsf{T}(\mathsf{L})$ is $\mathcal{O}(c_{\mathsf{T}}(2^N) + 2^N \times c_{\mathsf{L}}(N))$.

PROOF. The complexity of the decision procedure for $\mathsf{T}(\mathsf{L})$ is divided in two parts. The first corresponds to the computation of $\eta(\psi)$, where $N$ is the size of $\psi$. This this process consists of computing $Lit(\psi)$, which is $\mathcal{O}(N)$ and then testing all subset of it for inconsistency. Since there is $\mathcal{O}(2^N)$ potential subsets, this part has complexity $\mathcal{O}2^N \times c_{\mathsf{L}}(N))$.

The second part is to apply the decision procedure of $\mathsf{T}$ to $\sigma(\eta(\psi) \to \psi)$. The size of $\eta(\psi)$ is $\mathcal{O}(2^N)$, so this part has complexity $\mathcal{O}(c_{\mathsf{T}}(2^N))$, leading to the overall complexity $\mathcal{O}(c_{\mathsf{T}}(2^N) + 2^N \times c_{\mathsf{L}}(N))$.

∎

## *2.4 Iterated temporalisations*

This section serves as a prelude to the independent combinations of logics to be presented in Section 3. Here we analyse what happens when we apply $\mathsf{US}$ to a logic $\mathsf{L} = \mathsf{T}(\mathsf{US})$, where $\mathsf{T}$ may contain a renaming of $U$ and $S$. This violates the initial assumption that the set of connectives from the external $\mathsf{US}$ and the internal $\mathsf{L}$ are disjoint, for $U$ and $S$, without renaming, now appear in both systems.

To be a little bit more generic, let us consider a temporalisation $\mathsf{US_1}(\mathsf{L_n})$, where $\mathsf{L_n} = \mathsf{US_2}(\mathsf{US_1}(\ldots))$ corresponds to $n$ iterated temporalisations having $\mathsf{US_2}$ as the outermost external application.

It is important to note, however, that the internal and external occurrence of the connectives obey the same logic rules (inference rules and semantics).

We now analyse how the results obtained so far can be brought to $\mathsf{US_1}(\mathsf{L_n})$.

### Completeness and decidability of $\mathsf{US_1}(\mathsf{L_n})$

The main problem here is what is considered a monolithic subformula. For example, if we find a subformula of the format $G_1 p$ in a formula of $\mathsf{US_1}(\mathsf{L_n})$, is it considered monolithic, for it is part of $\mathsf{L_n}$, or is it considered part of $\mathsf{US_1}$, and thus not monolithic?

This question affects the constructions of the proofs of completeness and decidability in that it affects is how to compute $\eta(\psi)$.

The answer to this problem is: for the purpose of computing $\eta(\psi)$ a monolithic subformula is any subformula that is a member of the language of $\mathsf{L_n}$ that is not a

Boolean combination. This is in the spirit of the way $\eta(\psi)$ was defined and preserves its properties for the case of $\mathsf{US_1}(\mathsf{L_n})$.

As defined for a normal temporalisation of $\mathsf{T}(\mathsf{L})$ where $\mathsf{T}$ and $\mathsf{L}$ have disjoint sets of operators,

$$\eta(\psi) = \bigwedge \{ \; \Box_\psi \neg\varphi | \varphi \in Inc(\psi) \; \}$$

where each $\varphi$ consists of a Boolean combination of formulas of $\mathsf{L}$ only.

In $\mathsf{US_1}(\mathsf{L_n})$, $\varphi$ may contain now subformulas that belong to the language of $\mathsf{US_1}$. This, however, does not pose a problem anywhere on the proofs. Let us clarify this point with an example.

Consider the following formula $\psi$ in $\mathsf{US_1}(\mathsf{US_2}(\mathsf{US_1}))$[1].

$$\psi = G_1(p \wedge F_2H_2G_1\neg p)$$

Such a formula is inconsistent if $\mathsf{US_1}$ is complete in transitive flows of time with no end-points. In fact:

- $\vdash FHA \to A$ is a theorem of any temporal logic, so $\psi$ implies $G_1(p \wedge G_1\neg p)$.
- By normality we get $G_1p \wedge G_1G_1\neg p$ and, by transitivity, we get $G_1G_1p \wedge G_1G_1\neg p$.
- By normality, this implies that $G_1G_1(p \wedge \neg p)$. which contradicts the no-endpoints property of $\mathsf{US_1}$.

One may, by mistake, construct a temporalised model for $\psi$, if one considers $Mon(\psi) = \{p, F_2H_2G_1\neg p, H_2G_1\neg p\}$, ignoring $G_1\neg p$. Indeed, in this case, $\eta(\psi) = \top$, so $\sigma(\eta(\psi) \wedge \psi)) = G_1(q_p \wedge q_{F_2H_2G_1\neg p})$, which clearly has a model. And since we can get models to $p \wedge F_2H_2G_1\neg p$, we have constructed a temporalised model to an inconsistent formula!

The problem with the construction above is that what we have to consider now as monolithic subformulas is the set.

$$Mon(\psi) = \{p, F_2H_2G_1\neg p, H_2G_1\neg p, G_1\neg p\}$$

With such $Mon(\psi)$, we see that $F_2H_2G_1\neg p$ and $\neg G_1\neg p$ contradict (because $F_2H_2A \to A$ is a theorem of any temporal logic). We see that a formula of $\mathsf{US_1}$, namely $G_1\neg p$ occurs in $\sigma(\eta(\psi) \wedge \psi))$. So among the conjuncts of $\sigma(\eta(\psi) \wedge \psi))$ we find:

$$G_1(q_p \wedge q_{F_2H_2G_1\neg p}), \qquad q_{F_2H_2G_1\neg p} \to G_1\neg q_p$$

The first is $\sigma(\psi)$ and the second is $\sigma(\Box^0(F_2H_2G_1\neg p \to G_1\neg p))$ These two formulas imply inconsistency, in the same pattern as we derived the inconsistency of $\psi$ above, and by soundness of $\mathsf{US_1}$, no model can be built for it.

With this in mind, the proof of completeness of Section 2.2 applies immediately to $\mathsf{US_1}(\mathsf{L_n})$. This can be seen by an inspection on the proof. We see that nowhere else in that construction was it necessary to use the fact that the set of connectives of the external $\mathsf{US_1}$ and those of internal logic $\mathsf{L_n}$ is disjoint. In particular, any occurrence of a connective of $\mathsf{US_1}$ inside $\mathsf{L_n}$ always occurred within the scope of a $\mathsf{US_2}$ connective, avoiding any interaction between the external and internal occurrences of a $\mathsf{US_1}$ connective.

Similarly, the proof of decidability also applies to $\mathsf{US_1}(\mathsf{L_n})$.

[1] This example was suggested by Massimo Franceschet.

## Conservativeness of $\mathsf{US_1}(\mathsf{L_n})$

The proof of Lemma 2.13, stating that US(L) is a conservative extension of both US and L uses the fact that the sets of connectives are disjoint.

However, in the case of $\mathsf{US_1}(\mathsf{L_n})$, because the internal and external occurrences of connectives of $\mathsf{US_1}$ respect the same semantic rules and inference rules, it is possible to adapt the of Lemma 2.13 to $\mathsf{US_1}(\mathsf{L_n})$. For that, we have to assume that $\mathsf{US_1}$ and $\mathsf{US_2}$ are sound and complete, which gives us the soundness of $\mathsf{L_n}$.

In fact, the interesting case arises when we prove $A$ in $\mathsf{US_1}(\mathsf{L_n})$ and $A$ is a formula of $\mathsf{L_n}$. Then we have to analyse two cases:

- If $A$ is not in the language of $\mathsf{US_1}$, then the only way $A$ could have been derived is by the use of *Preserve* rule. In which case $A$ is also a theorem of $\mathsf{L_n}$.
- Suppose $A$ is in the language of $\mathsf{US_1}$. This means that $A$ is a pure $\mathsf{US_1}$ formula. As in Lemma 2.13, we show that $A$ is a theorem of $\mathsf{US_1}$. Suppose for contradiction that it is not. Then $\neg A$ is a consistent formula, and by completeness of $\mathsf{US_1}$, $\neg A$ has a model $\mathcal{M}_A$. We construct a temporalised model $\mathcal{M}_{\mathsf{T(L)}} = (T, <, g)$ such that, for every $t \in T$, $g(t) = \mathcal{M}_A$, thus building a T(L)-countermodel for $A$, contradicting the soundness of T(L). So $A$ is a theorem of US. But since the temporalisation is an extension of its components, any theorem of $\mathsf{US_1}$ is also a theorem of $\mathsf{L_n}$.

So if $A$ is a theorem in $\mathsf{US_1}(\mathsf{L_n})$ that is in the language of $\mathsf{L_n}$, it is a theorem of $\mathsf{L_n}$. The second item above also showed that if $A$ is in the language of $\mathsf{US_1}$, it is a theorem of $\mathsf{US_1}$. We have thus proved the following.

**Lemma 2.18** If $\mathsf{US_1}$ and $\mathsf{US_2}$ are sound and complete, then $\mathsf{US_1}(\mathsf{L_n})$ is a conservative extension of both $\mathsf{US_1}$ and $\mathsf{L_n}$.

### *2.5 Extension to multimodal, non-temporal logics*

Before we move to the independent combination of logics, we would like to discuss how the results presented so far generalise when in the external logic T the connectives may have any arity, and instead of only two ($U$ and $S$) we may have $n$ connectives $\triangle_1, \ldots, \triangle_n$, such that the arity of $\triangle_i$ is $r_i > 0$.

On the semantical side, we assume that each connective $\triangle_i$ is associated with a binary relation $R_i$. The semantics of formulas is based on a multidimensional frame $(W, R_1, \ldots, R_n)$. We have, however, to impose certain semantic restrictions:

- The semantics of $\triangle_i(p_1, \ldots, p_{r_i})$ is a monadic first-order formula (or connective *truth table* in the sense of [20, Chapter 8]) build from predicates $P_1(\cdot), \ldots, P_{r_i}(\cdot)$, the relational symbols $R_1, \ldots, R_n$, and equality.
- For each relational symbol $R_i$, we must be able to express a derived connective $\Box_i$ such that $\Box_i p$ expresses $\forall x(t\, R_i\, x \Rightarrow P(x))$. Furthermore, the inference system must be able to derive that if $\vdash A$ then $\vdash \Box_i A$, $1 \leq i \leq n$.

This second restriction correspond to the notion of *normality*. Note, however, that the demand of normality made here is weaker than that made in [28], for there it is required that *every argument position* in $\triangle_i(p_1, \ldots, p_{r_i})$ be normal, a requirement that even $U$ and $S$ fail to keep, for they are only normal in the first position and not in the second. A weaker requirement, however, can be found in [2].

Let us call the resulting system a *generalised modal/temporal logic*. The process of applying it externally to a logic $\mathsf{L}$ will be called a generalised *modalisation* of $\mathsf{L}$.

In the case of $US$-temporal logic, each connective has arity 2, $U$ is associates with binary relation $R_1 = <$ and $S$ with $R_2 = >$; also, $\Box_1 = G$ and $\Box_2 = H$ are derivable connectives. The fact that $R_1$ and $R_2$ are related is no limitation for this setting. This setting allows for many well-known modal logics, including the branching time modalities in CTL and CTL$^*$ and multi-arity connectives. The restriction (*) for the internal logics, however, remains.

We can then examine how our proof of completeness can be adapted. The definition of the *operator nesting tree* $D_\psi$ of a formula $\psi$ is simply extended to :

1. If $\psi$ is a literal or monolithic, then $D_\psi = \{\varepsilon\}$;
2. If $\psi = \varphi_1 \wedge \varphi_2$, then $D_\psi = D_{\varphi_1} \cup D_{\varphi_2}$;
3. If $\psi = \neg\varphi$, then $D_\psi = D_\varphi$;
4. If $\psi = \triangle_i(\varphi_1, \ldots, \varphi_{r_i})$, then $D_\psi = \{\varepsilon\} \cup \{i * s | s \in D_{\varphi_i} \cup \ldots \cup D_{\varphi_{r_i}}\}$.

This implies that the strings that compose our strings take as atoms the elements of the interval $[1, i]$ and that each node in the tree can be at most $i$-branching. A temporal operator can then be associated with each string in a straightforward way, that is, each $j \in [1, i]$ is associated with the derived operator $\Box_j$ and a string $j_1 \cdots j_p$ is associated with the string of connectives $\Box_{j_1} \cdots \Box_{j_p}$.

The definition of $\Box_\varphi\psi$ remains the same as before, namely the conjunction of all formulas of the form $\Box_{m_1,\ldots,m_n}\psi$ such that $\Box_{m_1,\ldots,m_n}$ is a temporal operator associated to a string in the operator nesting tree $D_\varphi$.

Given a multi-dimensional frame $(W, R_1, \ldots, R_n)$ and $t_1, \ldots t_m \in W$, such that $t_k$ is related with $t_{k+1}$ by some $R_i$, we represent by $\mathsf{string}(t_1, \ldots t_m)$ the string of length $m - 1$ obtained by a path through all those points.

Finally, the correspondence mapping $\sigma$ can be modified, remaining a homomorphism, so as to deal with generic modalities of the form $\triangle_i(\varphi_1,\ldots,\varphi_{r_i})$:

$$\sigma\left(\ \triangle_i(\varphi_1,\ldots,\varphi_{r_i})\ \right) = \triangle_i\left(\ \sigma(\varphi_1),\ldots,\sigma(\varphi_{r_i})\ \right).$$

The monolithic and Boolean cases remain the same.

Given those constructions all others constructions remain exactly the same. In particular, this way preserves the central notion of $Rg(A, t)$ as the "relevant part" of a multi-dimensional frame $(W, R_1, \ldots, R_n)$ with respect to a formula $A$ at a point $t \in W$, which plays a crucial role in the proof of transference of completeness. With such generalised construction, all lemmas and theorems are straightforwardly generalised and the transference of completeness and decidability follows for the temporalisation/modalisation of a logic with $n$ connectives of arbitrary arity that respect the semantical restrictions above. The reader is invited to verify the details.

What deserves note is the fact that in our construction the fact that $U$ and $S$ are mirror images is taken care by the definition of $\eta(\psi)$. In the same way, in the generalised modalisation, if there is any iteration between the connective and their respective semantical relations, this remains hidden in the construction of $\eta(\psi)$, and the proof generalises smoothly. We can then state the following result.

**Theorem 2.19** The properties of completeness and decidability are transferred via generalised modalisation/temporalisation.

## 3 The independent combination of temporal systems

Once we have generalised the transference results for the unrestricted temporalisation of a logic system over any class of flows of time, the next obvious question is whether such results generalise for the *independent combination* of two temporal logics.

Such an investigation was pursued for the linear case in [10], in which the transference of completeness was obtained by the "unravelling" of the independent combination in a finite number of temporalisations. Here we investigate if such technique is still applicable for the unrestricted case.

The work of Frank Wolter [28] on independent combination of logics (there called *fusion* of logics) is perhaps the work in the literature that more closely relates to the goals of the present work. That work explores the fusion of any number of logics containing any number of operators, of arbitrary arity. One restriction of such work was that each modality had to respect a restriction of normality in every argument, and it turns out the $U$ and $S$ do not respect such condition. Such a restriction was only eliminated as a side effect in a later work [2].

The present work compares with Wolter's in the following ways:

- We present a proof of transfer of decidability for $\mathsf{US}$ over any class of flows of time.
- Wolter's presentation is algebraic, while ours is based on Kripke semantics.
- Our construction shows how the independent combination can be seen as an infinite union of alternating temporalisations.

### *3.1 Definitions*

We now deal with the independent combination of two temporal logic systems, $\mathsf{US}_1$ and $\mathsf{US}_2$. If we temporalise $\mathsf{US}_1$ with $\mathsf{US}_2$, we obtain a very weakly expressive system; in such a system, if $\mathsf{US}_1$ is the internal temporal logic ($F_1$ is a derived connective in $\mathsf{US}_1$), and $\mathsf{US}_2$ is the external one ($F_2$ is also derived in $\mathsf{US}_2$), we cannot express that vertical and horizontal future operators commute,

$$F_1 F_2 A \leftrightarrow F_2 F_1 A.$$

In fact, the subformula $F_1 F_2 A$ is not even in the temporalised language of $\mathsf{US}_2(\mathsf{US}_1)$, nor is the whole formula. In other words, the interplay between the two-dimensions is not expressible in the language of the temporalised $\mathsf{US}_2(\mathsf{US}_1)$.

The idea is then to define a method for combining temporal logics that is symmetric. As usual, we combine the languages, inference systems and classes of models.

**Definition 3.1** Let $Op(\mathsf{T})$ be the set of non-Boolean operators of a generic temporal logic $\mathsf{T}$. Let $\mathsf{T}_1$ and $\mathsf{T}_2$ be two temporal logic systems such that $Op(\mathsf{T}_1) \cap Op(\mathsf{T}_2) = \emptyset$. The *fully combined language* of logic systems $\mathsf{T}_1$ and $\mathsf{T}_2$ over the set of atomic propositions $\mathcal{P}$ is obtained by the union of the respective set of connectives and the union of the formation rules of the languages of both logic systems.

Let the operators $U_1$ and $S_1$ be in the language of $\mathsf{US}_1$ and $U_2$ and $S_2$ be in that of $\mathsf{US}_2$. Their fully combined language over a set of atomic propositions $\mathcal{P}$ is given by

- every atomic proposition is in it;
- if $A, B$ are in it, so are $\neg A$ and $A \wedge B$;
- if $A, B$ are in it, so are $U_1(A, B)$ and $S_1(A, B)$.
- if $A, B$ are in it, so are $U_2(A, B)$ and $S_2(A, B)$.

The two languages taken to be independent of each other and the set of axioms of the two systems are supposed to be disjoint. The following combination method is the *independent combination* of two temporal logics. An axiomatisation is given by a pair $(\Sigma, \mathcal{I})$, where $\Sigma$ is a set of axioms and $\mathcal{I}$ is a set of inference rules.

We have very few limitations on the axiomatisations, namely:

- $\mathsf{US}_1$ and $\mathsf{US}_2$ are extensions of classical logic, so classical manipulations are admissible in the system; ie. if they are not primitive, they can be derived.
- Because the we are assuming a Kripke-style semantics, the logics have to be normal. This means that the axioms of normality (ie, the K-axioms) must be derivable for $G_1, H_1, G_2$ and $H_2$.
- The rule of necessitation has to be admissible: from $\vdash A$ derive $\vdash G_1 A$, $\vdash H_1 A$, $\vdash G_2 A$ and $\vdash H_2 A$.

Note that, since the set of operators of the two logics is disjoint, the set of axioms and inference rules referring to those operators will be disjoint.

**Definition 3.2** Let $\mathsf{US}_1$ and $\mathsf{US}_2$ be two $US$-temporal logic systems defined over the same set $\mathcal{P}$ of propositional atoms such that their languages are independent. The *independent combination* $\mathsf{US}_1 \oplus \mathsf{US}_2$ is given by the following:

- The fully combined language of $\mathsf{US}_1$ and $\mathsf{US}_2$.
- If $(\Sigma_1, \mathcal{I}_1)$ is an axiomatisation for $\mathsf{US}_1$ and $(\Sigma_2, \mathcal{I}_2)$ is an axiomatisation for $\mathsf{US}_2$, then $(\Sigma_1 \cup \Sigma_2, \mathcal{I}_1 \cup \mathcal{I}_2)$ is an axiomatisation for $\mathsf{US}_1 \oplus \mathsf{US}_2$.
- The class of independently combined flows of time is $\mathcal{K}_1 \oplus \mathcal{K}_2$ composed of bi-ordered flows of the form $(T, <_1, <_2)$ where the connected components of $(T, <_1)$ are in $\mathcal{K}_1$ and the connected components of $(T, <_2)$ are in $\mathcal{K}_2$, and $T$ is the (not necessarily disjoint) union of the sets of time points that constitute each connected component.

  A model structure for $\mathsf{US}_1 \oplus \mathsf{US}_2$ over the combined class $\mathcal{K}_1 \oplus \mathcal{K}_2$ is a 4-tuple $(T, <_1, <_2, g)$, where $(T, <_1, <_2) \in \mathcal{K}_1 \oplus \mathcal{K}_2$ and $g$ is an assignment function $g : T \to 2^{\mathcal{P}}$.
- The semantics of a formula $A$ in a model $\mathcal{M} = (T, <_1, <_2, g)$ is defined as the union of the rules defining the semantics of $\mathsf{US}_1/\mathcal{K}_1$ and $\mathsf{US}_2/\mathcal{K}_2$. The expression $\mathcal{M}, t \models A$ reads that the formula $A$ is true in the (combined) model $\mathcal{M}$ at the point $t \in T$. The semantics of formulas is given by induction in the standard way:

$\mathcal{M}, t \models p$ iff $p \in g(t)$ and $p \in \mathcal{P}$.
$\mathcal{M}, t \models \neg A$ iff it is not the case that $\mathcal{M}, t \models A$.
$\mathcal{M}, t \models A \wedge B$ iff $\mathcal{M}, t \models A$ and $\mathcal{M}, t \models B$.
For $i = 1, 2$:
$\mathcal{M}, t \models S_i(A, B)$ iff there exists an $s \in T$ with $s <_i t$ and $\mathcal{M}, s \models A$ and for every $u \in T$, if $s <_i u <_i t$ then $\mathcal{M}, u \models B$.
$\mathcal{M}, t \models U_i(A, B)$ iff there exists an $s \in T$ with $t <_i s$ and $\mathcal{M}, s \models A$ and for every $u \in T$, if $t <_i u <_i s$ then $\mathcal{M}, u \models B$.

The independent combination of two logics also appears in the literature under the names of *fusion* or *join*. The language of such a logic is referred to in the literature as a *two-dimensional* temporal language, even though its semantics is based on the evaluation of formulas at a single point (thus still one dimensional). The topic of two-dimensional modal/temporal languages and logics has been extensively discussed in the literature, e.g. [21, 23, 1, 26, 27, 20, 18].

We now proceed to examine the transference of properties through the independent combination.

## *3.2 Soundness of* $\mathsf{T}_1 \oplus \mathsf{T}_2$

Before we show the transference of soundness, it is worth noting an early result by Thomason [25], which is indeed more general than the independent combination of two US-logics. This result is useful in the proof of both soundness and completeness.

**Proposition 3.3 (Thomason [25])** With respect to the validity of formulas, the independent combination of two modal logics is a conservative extension of the original ones.

In algebraic presentations, Proposition 3.3 is considered a kind of soundness result. However, for our purposes, soundness has to do with the validity of all deductions. We present soundness as a consequence of Proposition 3.3, but it could also be obtained by verifying the validity of axioms and inference rules.

**Theorem 3.4 (Soundness Transference)** If $\mathsf{US}_1/\mathcal{K}_1$ and $\mathsf{US}_2/\mathcal{K}_2$ are sound logic systems, so is $\mathsf{US}_1 \oplus \mathsf{US}_2/\mathcal{K}_1 \oplus \mathcal{K}_2$.

PROOF. By induction of the length of a deduction. For the base case, we have to establish the validity of all axioms, which follows directly from the soundness of $\mathsf{US}_1/\mathcal{K}_1$ and $\mathsf{US}_2/\mathcal{K}_2$ and the fact that by Proposition 3.3, all $\mathsf{US}_1/\mathcal{K}_1$- and $\mathsf{US}_2/\mathcal{K}_2$-valid formulas are valid in the combined system (alternatively, their validity could be verified directly).

For the inductive case, all we are left to do is to verify that the inference rules transform valid formulas into valid formulas, which is a routine, straightforward task. ∎

### *3.3 Completeness*

In the proof of completeness, just as in [10], we will use the temporalisation as an inductive step in the construction of a combined model. However, as discussed in the presentation of the semantics of temporalised logics in Section 2.1, the class of temporal models of the internal logic must also include the evaluation time point, so that a member of the class of models of $\mathsf{US}_1$ or $\mathsf{US}_2$ is a quadruple $(T, <, g, t)$, where $t \in T$.

Let us first define the *degree of alternation* of a $(\mathsf{US}_1 \oplus \mathsf{US}_2)$-formula $A$, $dg(A)$, as the maximum number of alternate times a connective of one of the temporal logics occurs inside a connective of the other temporal logic. In this way, formulas of $\mathsf{US}_1$ and of $\mathsf{US}_2$ all have degree of alternation 0. If we take a temporal formula of $\mathsf{US}_1$, say $F_1 p$ and place it inside a connective of $\mathsf{US}_2$, say $H_2$, the formula $H_2 F_1 p$ has degree 1; similarly, $U_1(H_2 F_1 p, q)$ has degree 2, and so on.

The main idea of the completeness proof is based on the fact that any formula $A$ of $\mathsf{US}_1 \oplus \mathsf{US}_2$ can be seen as a formula of some finite number of alternating temporalisations of the form $\mathsf{US}_1(\mathsf{US}_2(\mathsf{US}_1(\ldots)))$; more precisely, $A$ can be seen as a formula of $\mathsf{US}_1(\mathsf{L_n})$, where $dg(A) = n$, $\mathsf{US}_1(\mathsf{L_0}) = \mathsf{US}_1$, $\mathsf{US}_2(\mathsf{L_0}) = \mathsf{US}_2$, and $\mathsf{L}_{n-2i} = \mathsf{US}_2(\mathsf{L}_{n-2i-1})$, $\mathsf{L}_{n-2i-1} = \mathsf{US}_1(\mathsf{L}_{n-2i-2})$, for $i = 0, 1, \ldots, \lceil \frac{n}{2} \rceil - 1$.

The following Lemma actually allows us to obtain transference of completeness to the independent combination via finite number of alternating temporalisations of $\mathsf{US}_1$ and $\mathsf{US}_2$.

**Lemma 3.5** Let $\mathsf{US}_1$ and $\mathsf{US}_2$ be sound and complete. $A$ is a theorem of $\mathsf{US}_1 \oplus \mathsf{US}_2$ iff it is a theorem of $\mathsf{US}_1(\mathsf{L_n})$, where $dg(A) = n$.

PROOF. If $A$ is a theorem of $\mathsf{US}_1(\mathsf{L_n})$, all the inferences in its deduction can be repeated in $\mathsf{US}_1 \oplus \mathsf{US}_2$, so it is a theorem of $\mathsf{US}_1 \oplus \mathsf{US}_2$.

Suppose $A$ is a theorem of $\mathsf{US}_1 \oplus \mathsf{US}_2$; let $B_1, \ldots, B_m = A$ be a deduction of $A$ in $\mathsf{US}_1 \oplus \mathsf{US}_2$ and let $n' = max\{dg(B_i)\}$, $n' \geq n$. We claim that each $B_i$ is a theorem of $\mathsf{US}_1(\mathsf{L_{n'}})$. In fact, by induction on $m$, if $B_i$ is obtained in the deduction by substituting into an axiom, the same substitution can be done in $\mathsf{US}_1(\mathsf{L_{n'}})$; if $B_i$ is obtained by some inference rule from $B_{j_1}, \ldots, B_{j_k}$, $j_1, \ldots, j_k < i$, then by the induction hypothesis, each $B_{j_\ell}$ is a theorem of $\mathsf{US}_1(\mathsf{L_{n'}})$ and so is $B_i$.

So $A$ is a theorem of $\mathsf{US}_1(\mathsf{L_{n'}})$. It follows from the semantic definitions that the set of valid formulas in $\mathsf{US}_1(\mathsf{L_{n'}})$ is a subset of the valid formulas in $\mathsf{US}_1 \oplus \mathsf{US}_2$. Since $\mathsf{US}_1$ and $\mathsf{US}_2$ are two complete logic systems, by Theorem 2.12 we know that $\mathsf{US}_1(\mathsf{L_{n'}})$ is complete for each $n'$. So Lemma 2.18 yields that each of the alternating temporalisations in $\mathsf{US}_1(\mathsf{L_{n'}})$ is a conservative extension of $\mathsf{L_{n'}}$; it follows that $A$ is a theorem of $\mathsf{US}_1(\mathsf{L_n})$, as desired. ∎

**Theorem 3.6 (Completeness of $\mathsf{US}_1 \oplus \mathsf{US}_2$)** Let $\mathsf{US}_1/\mathcal{K}_1$ and $\mathsf{US}_2/\mathcal{K}_2$ be two sound and complete logic systems. Then their independent combination $\mathsf{US}_1 \oplus \mathsf{US}_2$ is sound and complete over the class $\mathcal{K}_1 \oplus \mathcal{K}_2$.

PROOF. Soundness is given by Theorem 3.4. For completeness, suppose that $A$ is a consistent formula in $\mathsf{US}_1 \oplus \mathsf{US}_2$; by Lemma 3.5, $A$ is consistent in $\mathsf{US}_1(\mathsf{L_n})$, so we construct a temporalised model for it, and we obtain a model $(T^1, <^1_1, g^1, o^1)$, where $o^1 \in T^1$ is the "current time" considered as part of a model to respect the restriction $(*)$ of Section 2.1. We show now how it can be transformed into a model over $\mathcal{K}_1 \oplus \mathcal{K}_2$.

Without loss of generality, suppose that $\mathsf{US}_1$ is the outermost logic system in the multi-layered temporalised system $\mathsf{US}_1(\mathsf{US}_2(\mathsf{US}_1(\ldots)))$, and let $n$ be the number of alternations. The construction is recursive, starting with the outermost logic. Let $i \leq n$ denote the step of the construction; if $i$ is odd, it is a $\mathsf{US}_1$-temporalisation, otherwise it is a $\mathsf{US}_2$-temporalisation. At every step $i$ we construct the sets $T^{i+1}$, $<_1^{i+1}$ and $<_2^{i+1}$ and the function $g^{i+1}$.

We start the construction of the model at step $i = 0$ with the temporalised model $(T^1, <_1^1, g^1, o^1)$ such that $(T^1, <_1^1) \in \mathcal{K}_1$, and we take $<_2^1 = \emptyset$. At step $i < n$, consider the current set of time points $T^i$; according to the construction, each $t \in T^i$ is associated to:

- a temporalised model $g^i(t) = (T^{i+1}(t), <_1^{i+1}(t), g^{i+1}(t), o^{i+1}(t)) \in \mathcal{K}_1$ and take $<_2^{i+1}(t) = \emptyset$, if $i$ is even; or
- a temporalised model $g^i(t) = (T^{i+1}(t), <_2^{i+1}(t), g^{i+1}(t), o^{i+1}(t)) \in \mathcal{K}_2$ and take $<_1^{i+1}(t) = \emptyset$, if $i$ is odd.

The point $t$ is made identical to $o^{i+1}(t) \in T^{i+1}(t)$, so as to add the new model to the current structure; note that this preserves the satisfiability of all formulae at $t$. Let $T^{i+1}$ be the (possibly infinite) union of all $T^{i+1}(t)$ for $t \in T^i$; similarly, $<_1^{i+1}$ and $<_2^{i+1}$ are generated. And finally, for every $t \in T^{i+1}$, the function $g^{i+1}$ is constructed as the union of all $g^{i+1}(t)$ for $t \in T^i$.

Repeating this construction $n$ times, we obtain a combined model over $\mathcal{K}_1 \oplus \mathcal{K}_2$, $\mathcal{M} = (T^n, <_1^n, <_2^n, g^n)$, such that for all $t \in T^n$, $g^n(t) \subseteq \mathcal{P}$. Since satisfiability of formulae is preserved at each step, it follows that $\mathcal{M}$ is a model for $A$, and completeness is proved. ∎

## *3.4 Decidability*

We are going to show the transference of decidability by a recursive application of the temporalisation, generalising the proof of decidability of $\mathsf{T}(\mathsf{L})$ in Section 2.3.

The idea of the recursive proof is to consider a formula $\psi$ of the independent language $\mathsf{US}_1 \oplus \mathsf{US}_2$ of alternation depth $n$ as a temporalised formula $\mathsf{US}(\mathsf{L_n})$. By Lemma 3.5, $\psi$ is a $\mathsf{US}_1 \oplus \mathsf{US}_2$-theorem iff it is a $\mathsf{US}(\mathsf{L_n})$-theorem. Thus the decidability of $\psi$ in $\mathsf{US}_1 \oplus \mathsf{US}_2$ reduces to its decidability in $\mathsf{US}(\mathsf{L_n})$. The following is the basic result in the transference of decidability.

**Lemma 3.7** Let $\mathsf{US}_1/\mathcal{K}_1$ and $\mathsf{US}_2/\mathcal{K}_2$ be two sound, complete and decidable temporal logics. Then for every formula $\psi$ of $\mathsf{US}_1 \oplus \mathsf{US}_2$, there exists a $\mathsf{US}_1$ formula $A$ that is effectively constructible such that $\psi$ is $\mathsf{US}_1 \oplus \mathsf{US}_2$-valid iff $A$ is $\mathsf{US}_1$-valid.

PROOF. Let $\psi$ be a $\mathsf{US}_1 \oplus \mathsf{US}_2$ formula of alternation depth $n$. We propose the following decision procedure, $\mathsf{US}_1 \oplus \mathsf{US}_2$-Decide($\psi$):

> Let $n$ be $\psi$'s alternation degree. If $n = 0$, then $\psi$ is a $\mathsf{US}$-formula and we apply the $\mathsf{US}_1$- or $\mathsf{US}_2$-decision procedure to decide $\psi$, according to which language $\psi$ belongs to.
>
> Otherwise, we construct the formula $\eta(\psi) \to \psi$ in the following way:
>
> - Let $Lit(\psi) = Mon(\psi) \cup \{\neg\phi | \phi \in Mon(\psi)\}$, where $Mon(\psi)$ is the set of monolithic subformulas of $\psi$.

- Let $Inc(\psi)$ be set of inconsistent conjunctions $\phi_i$ in $Lit(\psi)$; *this inconsistency is obtained by a recursive call to* $\mathsf{US}_1 \oplus \mathsf{US}_2$-*Decide*($\phi_i$), where each $\phi_i$ now has alternation degree at most $n-1$.
- Build $\eta(\psi)$ from $Inc(\psi)$ as in Definition 2.7.

Apply $\mathsf{US}_1$-decision procedure to $\sigma(\eta(\psi) \to \psi)$ and return its output.

The recursive construction of $\eta(\psi)$ always terminates, for in each recursive call of the decision process, the degree of alternation decreases, and the procedure stops when it reaches a degree of alternation 0.

The correctness of the procedure is proven by induction on $n$. For $n = 0$ we simply apply the temporal decision procedure of the corresponding temporal logic.

For $n > 0$ we claim that deciding $\phi$ is equivalent to deciding $\sigma(\eta(\psi) \to \psi)$. In fact:

- $\psi$ is $\mathsf{US}_1 \oplus \mathsf{US}_2$-valid iff it is $\mathsf{US}(\mathsf{L_n})$-valid by completeness and Lemma 3.5.
- $\psi$ is $\mathsf{US}(\mathsf{L_n})$-valid iff $\sigma(\eta(\psi) \to \psi)$ is $\mathsf{US}_1$-valid by Lemma 2.15 and $\eta(\psi)$ is constructed deciding the validity of a set of formulas with alternation degree at most $n-1$, so by induction hypothesis $\eta(\psi)$ is constructible.

Thus we have a correct, terminating decision procedure for $\mathsf{US}_1 \oplus \mathsf{US}_2$. ∎

The transference of decidability directly follows from the previous Lemma.

**Theorem 3.8** Let $\mathsf{US}_1/\mathcal{K}_1$ and $\mathsf{US}_2/\mathcal{K}_2$ be two sound, complete and decidable temporal logics. Then $\mathsf{US}_1 \oplus \mathsf{US}_2$ is decidable.

Proof. Let $\psi$ be a $\mathsf{US}_1 \oplus \mathsf{US}_2$-formula. By Lemma 3.7, we construct a $\mathsf{US}_1$-formula whose decision problem is equivalent to $\psi$ and then apply $\mathsf{US}_1$'s decision procedure. ∎

With regards to the complexity of the decision problem, the algorithm outlined above does not give us a good starting point. However, a very detailed analysis of the complexity of such systems was done in [24].

## *3.5 Extension to an arbitrary number of multimodal logics*

In Section 2.5 we showed how the process of applying a logic externally to another could be generalised to modal logics with $n$ connectives of arbitrary arity. In the case of the independent combination, we can go even further. For the temporalisation (or the extended modalisation) is a combination process that involves only two logics: the external $\mathsf{T}$ and the internal $\mathsf{L}$.

However, in the independent combination of logics, this limitation does not hold. For in a *generalised independent combination* any number of logics may be taken as input, each with any number of connectives of arbitrary arity.

Does the transference of properties hold is such a generalised form?

Let us first concentrate on the a combination of two generalised modal logics, $\mathsf{M}_1$ and $\mathsf{M}_2$. We start noting that the format of the combined model $(T, <_1, <_2)$ in the independent combination is basically the same of that of the generalised frame $(W, R_1, \ldots, R_n)$. As usual, we assume that the connectives of $\mathsf{M}_1$ are distinct from those of $\mathsf{M}_2$. Combining the languages and inferences systems poses no problems. In combining the two classes of frames, we would end up with frames of the form $(W, R_1^1, \ldots, R_{n_1}^1, R_1^2, \ldots, R_{n_2}^2)$, which has the same format of a generalised frame.

This means that $\mathsf{M}_1 \oplus \mathsf{M}_2$ has the same format of a generalised modal logic, so the process can be iterated once more. That is we can independently combine $\mathsf{M}_1 \oplus \mathsf{M}_2$ with a generalised modal logic $\mathsf{M}_3$ obtaining yet another generalised modal logic, $(\mathsf{M}_1 \oplus \mathsf{M}_2) \oplus \mathsf{M}_3$. Of course, this process can be iterated any number of time. And its not hard to see that the process, at least on the level of combining language and inference systems, is associative and commutative. On the semantic level, note that we do not distinguish, say, the frames $(W, R_1, R_2)$ from $(W, R_2, R_1)$, so the resulting frame will have a single set of points and the disjoint union of all relations involved which of course is associative and commutative.

This shows that if we can independently combine two generalised modal logics, we can easily independently combine any number of such logics. It remains to be shown that the generalised modalisation/temporalisation can still be used as a building block for the independent combination of $\mathsf{M}_1$ and $\mathsf{M}_2$.

To show that this is indeed the case, we will show how the construction above can be modified for multimodal logics.

The main thing to note here is that, no matter what the modal connectives are, the independent combination of two modal systems can be decomposed in a successive number of modalisations/temporalisations, for a formula of $\mathsf{M}_1 \oplus \mathsf{M}_2$ can always be seen as a formula of some finite number of temporalisations: $\mathsf{M}_1(\mathsf{M}_2(\mathsf{M}_1(\ldots)))$.

The notion of degree of alternation in this case is exactly the same as in the $US$ case. The core of the completeness proof remains the same, namely the proof of the following lemma.

**Lemma 3.9** Let $\mathsf{M}_1$ and $\mathsf{M}_2$ be two sound and complete generalised modal logics. The formula $A$ is a theorem of $\mathsf{M}_1 \oplus \mathsf{M}_2$ iff it is a theorem of some modalised system $\mathsf{M}_1(\mathsf{M}_2(\mathsf{M}_1(\ldots)))$.

The depth of the temporalised system, as before, is bounded by the *degree of alternation* $d$ in $A$ of the nesting modal of $\mathsf{M}_1$ inside $\mathsf{M}_2$ operators, and vice-versa. Such a notion is exactly as it was in the $US$ case. Also, the remarks made in Section 2.4 as to what should be a monolithic formula in $\mathsf{M}_1(\mathsf{M}_2(\mathsf{M}_1(\ldots)))$ also apply here.

With Lemma 3.9 all there is to do now is to mimick the construction of the model in Theorem 3.8. If $A$ is a consistent $\mathsf{M}_1 \oplus \mathsf{M}_2$ formula, by Lemma 3.9 it is also consistent in some temporalised logic $\mathsf{M}_1(\mathsf{M}_2(\mathsf{M}_1(\ldots)))$ with at most $d$ alternations. We apply the generalised modalisation transference of completeness to obtain a modalised model for $A$ and then mimick the steps of Theorem 3.8 to transform such a model into a model of the independent combination. This is straightforward and we ommit the details. This shows that completeness is transferred through independent combination.

To obtain the transference of decidability we hardly have to make any changes to the proof in Section 3.4. There the decision procedure is based on the fact that a formula of $\mathsf{T}_1 \oplus \mathsf{T}_2$ is valid iff it is valid in some temporalised system. But the same result was generalised in Lemma 3.9. So the decision procedure for the generalised case is the same as the decision procedure for the $US$ case, with barely any difference, for we have already shown in Section 2.5 how to extend the mappings $\sigma$ and $\eta$ to the generalised case, which are all that is needed in the decision procedure. So decidability is transferred.

And since soundness is transferred by the result of Thomason [25], we can then conclude the following.

**Theorem 3.10** Let $\mathsf{M}_1$ and $\mathsf{M}_2$ be two sound, complete and decidable extended modal logics. Then $\mathsf{M}_1 \oplus \mathsf{M}_2$ is sound, complete and decidable.

## 4 Conclusion

We have extended the original results on temporalisation of [11] to any class of flows of time, extending the original result for linear classes only. This results was also extended to multi-modal logics with $n$-ary connectives.

Recursive temporalisations were used in [10] to show the transference of completeness and decidability for the independent combination of two linear US-temporal logics. Such construction was shown to generalise to the unrestricted case and was developed inside the traditional Kripke semantics for temporal logics. The same technique could also be applied to the independent combination of arbitrary number of multi-modal logics with $n$-ary connectives.

Recently, the work in [2] has generalized Wolter's algebraic results in [28] for the independent combination of US-logics in the algebraic tradition. That work was developed independently from ours, and did not have in mind US-logics, but was developed for Description Logics; The generalization of Wolter's result for decidability developed in [2] also applies to US-logic. So the relevant points of the results in here are the fact the independent combination was achieved using kripke-style semantics and that we can consistently see any kind of independent combinations as an iterations of modalisations/temporalisations. Note that in all such works, including ours, at least some form of normal behaviour was assumed from the connectives.

It remains an open problem whether the decidability of the logics with arbitrary operators (normal or non-normal) is transferred by their independent combination. The investigation of non-normal temporalisations/modalisations remains a viable way to explore such a question and is a path to be explored in the future.

## Acknowledgements

We would like to thank Massimo Franceschet, Angelo Montanari, Maarten de Rijke and Frank Wolter for helpful discussions. Marcelo Finger was partly supported by the Brazilian Research Council (CNPq), grant PQ 300597/95-5.

## References

[1] Aqvist, L.. A Conjectured Axiomatization of Two-dimensional Reichenbachian Tense Logic. *J. of Philosophical Logic*, 8:1–45, 1979.

[2] Baader, F., Lutz, C., Sturm, H. and FWolter, F.. Fusions of description logics. In F. Baader and U. Sattler, editors, *Proceedings of the 2000 International Workshop on Description Logics*, pages 21–30. Aachener Beitraege zur Informatik, Wissenschaftsverlag Meinz in Aachen, 2000.

[3] Barringer, H., Fisher, M., Gabbay, D., Owens, R. and Reynolds, M. editors. *The Imperative Future — Principles of executable temporal logic.* Johh Wiley & Sons, 1996.

[4] Blackburn, P., Gardent, C. and Meyer-Viol, W.. Talking About Trees. In *6th Conference of the European Chapter of the Association of Computational Linguistics*, pages 21–29, 1993.

[5] Blackburn P. and Meyer-Viol, W.. Linguistics, logic, and finite trees. *Logic Journal of the IGPL*, 2:3–29, 1994.

[6] Fagin, R., Halpern, J. Y., Moses, Y. and Vardi, M.Y.. *Reasoning about Knowledge.* MIT Press, 1995.

[7] Fine, K. and Schurz,G.. Transfer theorems for stratified multimodal logics. To appear in the Proceedings of Arthur Prior Memorial Conference, Christchurch, New Zeland, 1991.

[8] Finger, M.. Handling Database Updates in Two-dimensional Temporal Logic. *J. of Applied Non-Classical Logic*, 2(2):201–224, 1992.

[9] Finger, M.. *Changing the Past: Database Applications of Two-dimensional Temporal Logics*. PhD thesis, Imperial College, Department of Computing, February 1994.

[10] Finger, M. and Gabbay, D. M.. Combining Temporal Logic Systems. *Notre Dame Journal of Formal Logic*, 37(2):204–232, Spring 1996.

[11] Finger, M. and Gabbay, D. M.. Adding a Temporal Dimension to a Logic System. *Journal of Logic Language and Information*, 1:203–233, 1992.

[12] Finger, M. and Reynolds, M.. Two-dimensional executable temporal logic for bitemporal databases. In *2nd International Conference on Temporal Logic (ICTL'97)*, Manchester, July 1997. Kluwer.

[13] Finger, M. and Reynolds, M.. Imperative history: Two-dimensional executable temporal logic. In H.-J. Ohlbach and U. Reyle, editors, *Logic, Language and Reasoning — Essays in Honour of Dov Gabbay*, pages 73–98. Kluwer Academic Publishers, 1999.

[14] Finger, M. and Weiss, M. A.. The unrestricted addition of a temporal dimension to a logic system. In *3rd International Conference on Temporal Logic (ICTL2000)*, Leipzig, Germany, 4–7 October 2000.

[15] Finger, M.. A logical reconstruction of temporal databases. *Journal of Logic and Computation*, 10(6):847–876, 2000.

[16] Franceschet, M.. *Dividing and Conquering the Layered Land*. PhD thesis, University of Udine, 2002. Forthcoming.

[17] Franceschet, M., Montanari, A. andcde Rijke, M.. Model checking for combined logics. In *Proceedings of the 3rd International Conference on Temporal Logics (ICTL2000)*, 2000.

[18] Gabbay, D. M. Reynolds, M. and Finger, M.. *Temporal Logic: Mathematical Foundations and Computational Aspects*, volume volume 2. Oxford University Press, 2000.

[19] Gabbay, D. M. The Declarative Past and the Imperative Future. In B. Banieqbal et al., editors, *Coloquium on Temporal Logic and Specifications — Lecture Notes in Computer Science 389*, Manchester, April 1987. Springer-Verlag.

[20] Gabbay, D. M. , Hodkinson, I. M. and Reynolds, M. A.. *Temporal Logic: Mathematical Foundations and Computational Aspects*, volume 1. Oxford University Press, 1994.

[21] Kamp, H.. Formal Properties of Now. *Theoria*, 35:227–273, 1971.

[22] Kracht, M. and Wolter, F.. Properties of independently axiomatizable bimodal logics. *Journal of Symbolic Logic*, 56(4):1469–1485, 1991.

[23] Segerberg, K.. Two-dimensional Modal Logic. *J. of Philosophical Logic*, 2:77–96, 1973.

[24] Spaan, E.. *Complexity of Modal Logics*. PhD thesis, Free University of Amsterdam, Falculteit Wiskunde en Informatica, Universiteit van Amsterdam, 1993.

[25] Thomason, S. K.. Independent Propositional Modal Logics. *Studia Logica*, 39:143–144, 1980.

[26] Venema, Y.. Expressiveness and Completeness of an Interval Tense Logic. *Notre Dame Journal of Formal Logic*, 31(4), Fall 1990.

[27] Venema, Y.. *Many-Dimensional Modal Logic*. PhD thesis, Department of Mathematics and Computer Science, University of Amsterdam, 1991.

[28] Wolter, F.. Fusions of modal logics revisited. In Marcus Kracht, Maarten de Rijke, Heinrich Wansing, and Michael Zakharyaschev, editors, *Advances in Modal Logic*, volume Volume 1 of *Lecture Notes 87*, pages 361–379. CSLI Publications, Stanford, CA, 1996.

[29] Xu, M.. On some $U, S$-Tense Logics. *Journal of Philosophical Logic*, 17:181–202, 1988.

Received 30 October 2001. Revised 12 March 2002

# A Fibring Semantics for the Semantic-Morphological Interface in Natural Language

RALF NAUMANN, *Seminar für Allgemeine Sprachwissenschaft, University of Düsseldorf, Germany.*
*E-mail: naumann@mail.phil-fak.uni-duesseldorf.de*

## Abstract

*Lexical Decomposition Grammar* assumes that there is a flow of information in grammar that constrains the distribution of linguistic information at the different levels of grammatic representation. In particular, there is a flow of information from the semantic to the morphological level. Particular argument positions correspond to particular cases. This relationship is mediated by intervening constraints at other levels like the theta structure and the syntactic level.

In this paper a formal reconstruction of the flow of information is presented. The grammar is represented by a tuple $< \mathbf{S}_1, ..., \mathbf{S}_n, \gamma_1, ..., \gamma_m >$. Each $\mathbf{S}_i$ is the substructure corresponding to the $i$-th level of linguistic analysis and each $\gamma_j$ is a mapping relating the domains of two substructures. If $\mathbf{S}$ and $\mathbf{S}'$ are related by some $\gamma$, this means that elements of the domain $D_{\mathbf{S}}$ 'constrain' the information associated with objects from the domain $D_{\mathbf{S}'}$ in the sense that if some $d \in D_{\mathbf{S}}$ satisfies the information expressed by $\phi$, then $\gamma(d) \in D_{\mathbf{S}'}$ satisfies the information expressed by $\psi$. Second, appropriate languages (logics) to talk about the structures are defined. The combination of different ontologies is reflected at the syntactic level by *fibring* these languages across each other. This means that instead of building a language $L$ over a base of propositional variables VAR, it is defined over a base which corresponds to the wffs of a second language $L'$. The resulting language $L(L')$, '$L$ layered over $L'$', has two layers: a top layer consisting of the non-atomic formulas and a base layer built by means of the formulas from $L'$.[1]

## 1 Lexical Decomposition Grammar and the Flow of Information

Is there a flow of information in grammar? In Lexical Decomposition Grammar, Wunderlich (1997, 1999), a positive answer to this question is given. It is assumed that, at least normally, the case assigned to an argument of a verb can be predicted from the place that the object denoted by that argument occupies in the dynamic-temporal structure of an event denoted by the verb. E.g. the object that is involved first is denoted by an argument that is linked to the verb by nominative case and the object that is involved last is denoted by an argument that is either linked by accusative case or which is realized as a PP. The empiricial basis for this claim is given by generalizations about the relationship between the default word order and the assignment of case to arguments. Consider the following examples taken from

[1]Full version of a contributed paper presented at the *7th Workshop on Logic, Language, Information and Computation* (*WoLLIC'2000*) (http://www.cin.ufpe.br/~wollic/wollic2000/), August 15–18, 2000, Natal (Rio Grande do Norte), Brazil, organized by Univ. Federal de Pernambuco (UFPE) and Univ. Federal do Rio Grande do Norte (UFRN), with scientific sponsorship by IGPL, FoLLI, ASL, SBC, SBL, and funded by CAPES (grant CDS/PAEP/0099/00-7), CNPq (grant 450994/00-7), FUNPEC.

 

German.

(1) a. Der Mann$_x$ rannte. (The man was running)
b. Der Schnee$_x$ schmolz. (The snow melted)

(2) a. Der Mann$_x$ aß den Apfel$_y$. (The man ate the apple)
b. Die Primadonna$_x$ sang eine Arie$_y$. (The primadonna sang an aria)

(3) a. Der Lehrer$_x$ gab dem Schüler$_y$ ein Buch$_z$. (The teacher gave the pupil a book)
b. Die Frau$_x$ zeigte dem Mann$_y$ ein Bild$_z$. (The woman showed the man a picture)

Each verb in the above examples subcategorizes for a fixed number of arguments. Morphologically, these arguments are related (or linked) to the verb by a particular linker. In German, nominal arguments are linked by cases like nominative, dative or accusative. At the syntactic level, the subcategorized arguments are realized in a particular default (word) order. This order is determined at the level of subordinate clauses; 'daß Hans rannte', 'daß der Mann einen Apfel aß' or 'weil der Lehrer dem Schüler ein Buch gab'. The place of an argument in the default word order and the case it is assigned are not independent of each other. Particular places correspond to particular cases and vice versa. From the data in (1)-(3) the generalizations in (4) and (5) can be derived.

(4) a. Nominative case is assigned only to the first element ($x$ in (1)-(3))
b. Accusative case is assigned only to the last element ($y$ in (2), $z$ in (3))
c. Dative case is assigned only to the intermediate element ($y$ in (3))

(5) a. The first element of the default word order is always assigned nominative case
b. If there are at least two elements, the last gets accusative case
c. If there are more than two elements, the intermediate one is realized by dative case.

The next task consists in relating the semantic interpretation of verbs with the default word order. LDG is based on an event ontology, i.e. $n$-place verbs are interpreted in the lexicon as $n+1$-ary relations on $(E \cup O)^n \times E$ with $E$ the domain of events and $O$ the domain of 'ordinary' objects. It differs from event-based approaches like that of Krifka by defining the relation between an event and its participants not in terms of thematic roles like actor or undergoer. The interpretation of a verb is rather based on a minimal decomposition that reflects the temporal-dynamic structure of events. An example for such a decomposition is given in (6).

(6) (ACT(x) & BEC(POSS(y,z)))(e)

Decompositions like (6) are called *semantic forms* (SF). Usually, a semantic form generalizes over a class of verbs. For instance (6) is the semantic form of all transfer verbs like 'geben' (give) where the transfer is from the actor to some recipient which is not identical to the actor. The intended interpretation of (6) can be paraphrased as 'By acting (in a particular way) (Act(x)) $x$ brings about (&) that $y$ has $z$ (Poss(y,z)). Each conjunct $\alpha_i$ is interpreted as a unary relation on the domain $E$ of events. The connective & expresses a kind of temporal succession in the following sense: $\alpha_1$ & ... & $\alpha_n$ denotes a complex property of events that is true of an event $e$ just in case there are initial stages (prefixes) $e_1$, ..., $e_n$ of $e$ with $\alpha_i$ being true of $e_i$, $1 \leq i \leq n$, and $e_i$ being an initial stage of $e_j$ for $1 \leq i < j \leq n$. If the SF is evaluated with respect to a given event $e$ (and suitable values for the remaining variables), each conjunct $\alpha_i$ is therefore evaluated with respect to a subevent of $e$. The connective & is not commutative. From this non-commutative character it follows that the $\alpha_i$ are linearly ordered. This linear ordering induces a corresponding linear ordering on the free variables of the SF. For 'geben', one gets: $x < y < z$. This ordering is used to define the notion of *theta structure* (TS) in terms of the *Hierarchy Principle* (HP).

(7) *Hierarchy Principle*: The order of arguments in TS is the inverse of the order of arguments in SF.

If $SF_v(x_1, ..., x_n, e)$ is the decomposition of $v$ in the representation language with the free variables $x_1, ..., x_n$ and $e$, and $x_i <_v x_j$ for $i < j$ means that the first occurrence of $x_i$ lies before that of $x_j$, the translation of $v$ in the lexicon that is in accordance with the HP is $\lambda x_n ... \lambda x_1 \lambda e.SF_v(x_1, ..., x_n, e)$. The HP defines a relationship between the two parts of the semantic level that are assumed in LDG. What is missing is the relation between the theta structure and the syntactic level. This relation is defined by the rule in (8).

(8) Mapping between TS and the syntactic component (S)
The order of arguments in TS is the inverse of the default word order, i.e. the arguments are realized inversely to the way they are abstracted over.

When taken together, the HP in (7) and the mapping rule in (8) say that the order of arguments at the SF level is identical to the (default) order in which they are syntactically realized. Thus, there is a 'flow of information' from the level of SF to the morphological component M that is mediated by the level of TS and the syntactic component S.

(9) SF $\xrightarrow{a}$ TS $\xrightarrow{b}$ S $\xrightarrow{c}$ M

$a =$ Hierarchy Principle (7)
$b =$ mapping rule in (8)
$c =$ generalizations in (4) and (5)

Linking rules are the result of composing the mappings $b$ and $c$. In LDG, they directly relate elements of TS to assignments of case. Three examples are given in (10).

(10) a. The highest (= last) element of TS is assigned nominative case.
b. If there are at least two elements in the TS, the lowest (= first) element gets accusative case.
c. If there are more than two elements in TS, the intermediate one is realized by dative case.

As it stands, LDG faces several problems. First, and foremost, the different levels of grammatical representation are not formalized. E.g. the syntactic and the morphological component are not fixed and the logical decomposition predicates *BEC* and & used at the level of semantic form are not formally defined. Consequently, the idea of a 'flow of information' is not made linguistically precise and can only be understood in an intuitive sense. But also for the linking component, where LDG is formulated more explicitly, problems arise. Recall that the theta structure is taken as a finite sequence. In LDG, the elements of this sequence are characterized by two binary features [*hr*] and [*lr*], the intended interpretation of which is 'a higher role' and 'a lower role', respectively. On this interpretation, $[\pm hr]$ ($[\pm lr]$) means 'there is a/no higher role' ('there is a/no lower role'). The basic idea underlying the definition of the cases in LDG, first formulated in Kiparsky (1992), is: what is linguistically relevant with respect to a case (in a particular language) is the range of positions in the theta structure it can occur at. Linguistically, a case is therefore related to a subset of the positions of an arbitrary TS. In LDG, this idea is made precise by defining the cases in terms of the two features [*hr*] and [*lr*]. The definitions are given in (11) ([] means that the case is completely unspecified with respect to both features).

(11) $NOM = []$ $\quad DAT = [+hr, +lr]$ $\quad ACC = [+hr]$

When taken in isolation, these definitions do not comply with the linking rules. E.g. for a transitive verb, both $NOM$ and $ACC$ can be assigned to the lowest (first) position of the theta structure because the structural specification of this position, $[+hr, -lr]$, is compatible with the definitions of both cases (whereas $DAT$ is excluded due to $[+lr]$). The determination of a case is therefore constraint by the *Principle of Secificity* (SP): *The case assigned to an argument is the most specific case that is compatible with the specification of the former.* The SC yields the ranking $NOM < ACC < DAT$. Applying the SC to the first position of the theta structure of a transitive verb excludes $NOM$ because the matching between $ACC = [+hr]$ and $[+hr, -lr]$ is more specific than that between $NOM = [\,]$ and $[+hr, -lr]$. In the table below the linking mechanism is illustrated for a three-place verb.

| $\lambda z$ | $\lambda y$ | $\lambda x.P_v(x)(y)(z)$ |
|---|---|---|
| $[+,-]$ | $[+,+]$ | $[-,+]$ |
| $ACC$, | $DAT$, | $NOM$ |
| $[+hr]$, | $[+hr,+lr]$, | $[\,]$ |

Table 1

A problem arises because the generalizations in (4) and (5), or the corresponding linking rules, are not without exceptions. They only capture what are called the *canonical* patterns of linking. The corresponding verbs are called *canonical verbs*.

| Canonical verb | order according to TS $z > y > x$ |
|---|---|
| ditransitive verb | $ACC$-$DAT$-$NOM$ |
| transitive verb | $ACC$-$NOM$ |
| intransitive verb | $NOM$ |

Table 2

An example for a class of verbs with a non-canonical linking pattern are so-called dative verbs. Examples for this class in German are the verbs 'helfen' (help) and 'folgen' (follow).[2]

[2] Dative verbs are not the only verbs in German that are non-canonical. The other classes are discussed and analyzed in Naumann (2001) using a non-monotonic logic.

(12) a. Hans half einem Freund. (Hans helped a friend)
b. Maria folgte einem Fremden. (Mary followed a stranger)

The internal argument is assigned *DAT* instead of *ACC*. This contradicts the generalizations (4c) and (5b). In Wunderlich (1997:51) this class is analyzed as follows. It is assumed that the lower argument is lexically marked by $[+lr]$. From this it is said to follow that the other argument must be interchanged with respect to the $[lr]$ feature, yielding the assignment of $[-lr]$ to the highest argument. Since the values of the feature $[hr]$ are not interchanged, the lowest argument is specified by $[+hr, +lr]$ and the highest one by $[-hr, -lr]$. Using the *SC*, the former is assigned dative and the latter nominative case (accusative case is excluded for this position because it requires $[+hr]$). This 'explanation' is either inconsistent with the intended meanings of the two features or it assumes that the meaning of these features is not restricted to the purely structural interpretation that has been assumed in the case of canonical patterns of linking. In the second case the meaning is too weak to sustain the inferences drawn from them to capture these canonical patterns. Thus, although LDG has a workable mechanism for canonical linking, there is no comparable strategy for dealing with non-canonical linking.[3]

One solution to the problem that non-canonical patterns of linking pose consists in considering not only structural properties of elements of the theta structure, which are defined using the two features $[hr]$ and $[lr]$, but also non-structural ones. E.g. if the theta structure of a verb is taken as the linearly ordered sequence of thematic roles that are defined for the verb, non-structural properties are the properties of the thematic roles, or, more simply, the thematic roles themselves. Linking rules will then take the form $\alpha \wedge \beta \rightarrow \gamma$ with $\alpha$ expressing a structural and $\beta$ expressing a non-structural property. E.g. instead of one linking rule for the lowest element of the theta structure, one will have two ($\theta_{dat}$ is the thematic role assigned to the internal argument of dative verbs like 'helfen' (help)).

(13) a. If $x$ is the lowest element of a TS with length greater 1 and if $x$ is $\theta_{dat}$, the argument corresponding to $x$ is realized by dative case.
b. If $x$ is the lowest element of a TS with length greater 1 and if $x$ is *not* $\theta_{dat}$, the argument corresponding to $x$ is realized by accusative case.

Non-canonical patterns of linking show that the generalizations in (4) and (5) are true only if they are understood as being relativized to canonical patterns. The relativization can be made explicit by adding information about thematic roles. The general form of a linking rule which covers both canonical and non-canonical patterns is (14).

(14) The argument corresponding to the $i$-th position of a TS (of length $m$) is morphologically realized by
a. case $C_1$ if $i$ satisfies $\phi_1$ or ... or by case $C_n$ if $i$ satisfies $\phi_n$
b. otherwise it is realized by case $C$

Clause (a) covers instances of non-canonical patterns of linking whereas clause (b) captures those instances belonging to canonical patterns. According to this form, canonical patterns are licensed by the *absence* of non-structural information, e.g.

[3] This argument still applies to the OT version of LDG presented in Wunderlich (2001).

about thematic roles, in the sense that the arguments do *not* satisfy particular information. Non-canonical patterns of linking, on the other hand, explicitly require the *presence* of particular non-structural information.[4]

In the second part of the article, the idea of a flow of information will be made formally precise. To this end, it is necessary to formalize each of the four grammatical levels that are involved in this flow. This will be done as follows. Instead of using the minimal decomposition format from LDG, the formalization of the semantic level is based on *Dynamic Event Semantics* (DES) developed in Naumann (1998, 2001) and Naumann and Osswald (1999). The theta structure is defined in terms of the linearly ordered sequence of so-called dynamic roles which are defined in terms of sets of results. The syntactic component is an ordered finite binary tree the nodes of which are decorated with subcategorization information. The morphological component consists of a feature structure for each node of the tree that contains information appropriate for the corresponding categorial information like case, gender and number insofar as this information is already determined by the verb. The flow of information is modeled by defining mappings between the structures formalizing each of the four grammatical levels. At the syntactic level, languages (logics) for talking about the ontologies underlying the structures and for expressing the mappings between the structures are specified.

## 2 Formalizing the Flow of Information

The outline of the basic ideas on which LDG is built has shown that the linguistic analysis of a verb in the lexicon is cast in terms of a composite ontology. There are events and ‘ordinary’ objects (SF), dynamic roles (TS), (tree) nodes (S) and points (in feature structures) (M). Each sort of entity comes equipped with a set of relations, giving rise to a structure in the model-theoretic sense, with the set of entities as the domain that underlies the structure. Suitable structures for the analysis of verbs will therefore have a complex ontology consisting of a *dynamic eventuality* (sub-)structure, *ordered finite binary trees*, *feature structures* and, most importantly from a linguistic point of view, mappings between the different substructures that mutually constrain the relations given in the structures.

### *2.1 LDG Structures*

LDG structures for a verb are tuples $\langle \mathbf{DES}, \mathbf{M}_{TS}, \mathbf{T}_S, \mathbf{M}_M, R_{z_f}, \{R_{\Diamond_v}\}_{v \in \text{VERB}}, R_{z_s}, \gamma, \Delta, \preceq, \{R_{dr^*} \mid dr^* \in DR^*\}\rangle$ such that

- $\mathbf{DES} = \langle \mathbf{E}, \mathbf{S}, \mathbf{O}, \{R_{dr} \mid dr \in DR\}, \{R_{\text{prop}} \mid \text{prop} \in \text{PROP}\}, \alpha, \omega, \tau, \mu\rangle$ is a dynamic eventuality structure
- $\mathbf{M}_{TS}$ is an ordered set together with a set of unary relations on this set
- $\mathbf{T}_S = \langle N, R_{\succ_1}, R_{\succ_2}, root, \Theta\rangle$ is a finite ordered binary tree

[4]The solution to the problem of non-canonical patterns of linking adopted in this article is neither the only possible nor the linguistically most satisfactory one. Alternative approaches are developed in Naumann (2001) and Wunderlich (2001). Whereas the former analyzes linking phenomena as a form of non-monotonic reasoning, the latter develops an analysis using optimality theory. The advantage of the solution adopted here is that it admits to model the flow of information in a simple way.

- $\mathbf{M}_M = \langle \{\mathbf{FS}_n \mid n \in \mathbf{N}_S\}, \delta \rangle$ is a set of feature structures together with a function $\delta$; each $\mathbf{FS}_n = \langle W_n, \{R_f\}_{f \in F}, \{Q_a\}_{a \in A}, w_{0,n} \rangle$ is a point-generated feature structure; for $n \neq n'$, $\mathbf{FS}_n$ and $\mathbf{FS}_{n'}$ are disjoint in the sense that $W_n \cap W_{n'} = \emptyset$
- The $R_{\Diamond_v}$, $R_{z_s}$ and $R_{z_f}$ are mappings which relate $\mathbf{E}$ and $\mathbf{T}_{TS}$, $\mathbf{T}_{TS}$ and $\mathbf{T}_S$ as well as $\mathbf{T}_S$ and $\mathbf{M}_M$, respectively
- $\gamma$, $\Delta$, $\preceq$ and the $R_{dr^*}$ are defined below in Section 2.5.1

Each of the four (sub)structures represents one of the four levels of grammatical representation. The mappings are used to define the linguistically relevant relationships between the various levels. At the logical level, to each mapping corresponds a modality (or a set of modalities) that is used to express constraints between two levels.

## 2.2 Languages for talking about LDG Structures

Recall from Section 1 that constraints between different levels of linguistic analysis are of the form 'If at level $n$ place $\alpha$ contains information $A$, then place $\beta$ (or places $\beta$) at the $m$-th level contains (contain) information $B$'. 'Place' refers to a particular element (or particular elements) of the domain underlying the structure representing a level of grammatical representation. 'Information' must be understood as properties which the elements of the domain have and which can be expressed by a formula of the language that is used to talk about the structure. The constraint, then, relates an element belonging to the domain of the $n$-th structure that satisfies $\phi_A$ to an element (or elements) belonging to the domain of the $m$-th structure that satisfies (satisfy) $\phi_B$. Let $R$ be the relation that maps elements from $D_{\mathbf{S}_n}$, i.e. the domain of the $n$-th structure, to elements of $D_{\mathbf{S}_m}$, i.e. the domain of the $m$-th structure, as required by the constraint. The constraint will then have the form (15).

(15) $\mathbf{S}_n, d \models \phi_A \Rightarrow$ there is a $d'$ s.t. $R(d, d')$ and $\mathbf{S}_m, d' \models \phi_B$

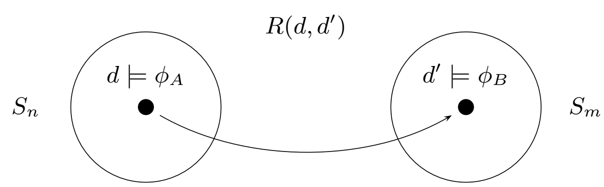


Figure 1

(15) does not express the constraint in the language $L^{\mathbf{S}_n}$. Expressing this constraint in $L^{\mathbf{S}_n}$ requires that it be possible to refer in this language to elements from $D_{\mathbf{S}_m}$. One way of satisfying this requirement consists in using a modality $\bullet$ which is interpreted by $R$. The formula $\langle \bullet \rangle \phi_B$, for $\phi_B$ a formula from $L^{\mathbf{S}_m}$, is then satisfied at an element $d$ from $D_{\mathbf{S}_n}$ just in case an element $d'$ from $D_{\mathbf{S}_m}$ that stands in the relation $R$ to $d$ satisfies $\phi_B$. The constraint can then be expressed by the $L^{\mathbf{S}_n}$ formula $\phi_A \rightarrow \langle \bullet \rangle \phi_B$. Evaluating this formula at $d$ yields (16).

(16) $\mathbf{S}_n, d \models \phi_A \rightarrow \langle \bullet \rangle \phi_B$ iff if $\mathbf{S}_n, d \models \phi_A$, then $\mathbf{S}_n, d \models \langle \bullet \rangle \phi_B$ iff if $\mathbf{S}_n, d \models \phi_A$, then there is a $d'$ s.t. $R(d, d')$ and $\mathbf{S}_m, d' \models \phi_B$

Following Blackburn et al. (1993) and Blackburn and Meyer-Viol (1997), the resulting language, $L^{\mathbf{S}_n}(L^{\mathbf{S}_m})$, will be called the language $L^{\mathbf{S}_n}$ *layered over* the language $L^{\mathbf{S}_m}$ so that a language with two 'layers' is used. The top layer are the formulas from $L^{\mathbf{S}_n}$. The base layer are formulas of the form $\langle\bullet\rangle\,\phi$ with $\bullet$ a modality and $\phi$ a formula from $L^{\mathbf{S}_m}$.

In the following sections the four levels of grammatical representation will be formalized. First the structures for the level in question are defined. In a next step a suitable language for talking about these structures is specified, followed by a definition of the relation which links this structure to another one. Finally, the languages are combined so that it is possible to talk about the combined ontology. Language $L^A$ will be layered over language $L^B$ whenever there is a flow of information from level $A$ to level $B$. Since in LDG the flow is given by SF $\rightarrow$ TS $\rightarrow$ S $\rightarrow$ M, $L^{SF}$ is layered over $L^{TS}$, $L^{TS}$ over $L^S$ and $L^S$ over $L^M$. In addition, since linking rules can be formulated as relating TS to M, $L^{TS}$will also be layered over $L^M$.

## *2.3 The Syntactic Level*

### 2.3.1 The Structures: Finite Ordered Binary Trees

Structures for the syntactic level are *finite ordered binary trees*. We will use the definition given in Blackburn and Meyer-Viol (1997).

**Definition 2.1** A finite ordered binary tree $\mathbf{T}$ is a tuple $\langle N, R_{\succ_1}, R_{\succ_2}, root, \Theta\rangle$ such that

- $N$ is a finite, non-empty set of (tree) nodes
- $R_{\succ_i}$ for $1 \leq i \leq 2$ is a binary relation on $N$ that is a partial function; $R_{\succ_1}(n, n')$ just in case $n'$ is the first daughter of $n$ and $R_{\succ_2}(n, n')$ just in case $n'$ is the second daughter of $n$
- *root* is the (unique) root of the tree
- $\Theta \subseteq N$ is the set of terminal nodes of the tree: $\forall n(n \in \Theta \rightarrow \forall n'(n' \in N \rightarrow \neg R_{\succ_i}(n, n')))$, i.e. terminal nodes do not immediately dominate any other node of the tree

¿From the fact that $\mathbf{T}$ is a binary tree it follows that if $R_{\succ_2}(n, n')$, then there is a unique $n''$ with $R_{\succ_1}(n, n'')$: $\forall n, n'(R_{\succ_2}(n, n') \rightarrow \exists n''(R_{\succ_1}(n, n'') \wedge \forall n'''(R_{\succ_1}(n, n''') \rightarrow n''' = n'')))$. Furthermore, $\forall n, n', n''(R_{\succ_1}(n, n') \wedge R_{\succ_2}(n, n'') \rightarrow n' \neq n'')$. In terms of $R_{\succ_1}$ and $R_{\succ_2}$ the following binary relations are defined.

(17) a. $R_{\succ} = R_{\succ_1} \cup R_{\succ_2}$ (the 'daughter of' relation)

b. $R_{\prec} = (R_{\succ})^c = \{(n, n') \mid (n', n) \in R_{\succ}\}$ (the 'mother of' relation)
$R_{\prec}$ is a partial function that is defined on $N - \{root\}$

c. $R_{\succ^*}$ and $R_{\prec^*}$ are the transitive, reflexive closure of $R_{\succ}$ and $R_{\prec}$, respectively. $R_{\succ^*}$ is the 'dominates' relation whereas $R_{\prec^*}$ is the 'is dominated by' relation; note that due to the reflexive character one has $\forall n(R_{\succ^*}(n, n) \wedge R_{\prec^*}(n, n))$.

d. $R_P = \{(n, n') \mid \exists n_1 \exists n_2 \exists n_3(R_{\succ^*}(n_1, n) \wedge R_{\succ_1}(n_2, n_1) \wedge R_{\succ_2}(n_2, n_3) \wedge R_{\succ^*}(n_3, n'))\}$ (the precedence relation on $N$)

e. $R_F$, the succession relation on $N$, is defined as the converse of $R_F$: $R_P = (R_F)^c$

### 2.3.2 The Language $L^S$

In order to talk about finite ordered binary trees, the tree language $L^S$ of signature $\langle \text{VAR} \rangle$ is defined.

**Definition 2.2** Well-formed formulas of $L^S$: (i) if $p \in \text{VAR}$ , then $p \in L^S_{wff}$; (ii) if $\phi, \psi \in L^S_{wff}$, then $\phi \wedge \psi \in L^S_{wff}$ and $\neg\phi \in L^S_{wff}$, (iii) if $\phi \in L^S_{wff}$, then $\alpha\phi \in L^S_{wff}$ for $\alpha \in \{\downarrow^1, \downarrow^2, \uparrow, \downarrow^*, \uparrow^*, P, F\}$.

In terms of the modalities used in clause (iii) the following modalities and constants are defined.

(18) a. $\downarrow \phi =_{def.} \downarrow^1 \phi \vee \downarrow^2 \phi$
b. $\Downarrow^i \phi =_{def.} \neg \downarrow^i \neg\phi$ for $1 \leq i \leq 2$; $\Downarrow \phi =_{def.} \neg \downarrow \neg\phi$ and $\Uparrow \phi =_{def.} \neg \uparrow \neg\phi$
c. $\Uparrow^* \phi =_{def.} \neg \uparrow^* \neg\phi$ and $\Downarrow^* \phi =_{def.} \neg \downarrow^* \neg\phi$
d. $s_0 =_{def.} \Uparrow \bot$ and $t =_{def.} \Downarrow \bot$

The wffs of $L^S$ are interpreted on finite ordered binary trees. The satisfaction relation is defined as follows (for the base clause see (22) below).

(19) a. $\mathbf{T}_S, n \models \neg\phi$ iff $\mathbf{T}_S, n \not\models \phi$
b. $\mathbf{T}_S, n \models \phi \wedge \psi$ iff $\mathbf{T}_S, n \models \phi$ and $\mathbf{T}_S, n \models \psi$
c. $\mathbf{T}_S, n \models \downarrow^1 \phi$ iff there is an $n'$ s.t. $R_{\succ_1}(n, n')$ and $\mathbf{T}_S, n' \models \phi$
d. $\mathbf{T}_S, n \models \downarrow^2 \phi$ iff there is an $n'$ s.t. $R_{\succ_2}(n, n')$ and $\mathbf{T}_S, n' \models \phi$
e. $\mathbf{T}_S, n \models \uparrow \phi$ iff there is an $n'$ s.t. $R_{\prec}(n, n')$ and $\mathbf{T}_S, n' \models \phi$
f. $\mathbf{T}_S, n \models \downarrow^* \phi$ iff there is an $n'$ s.t. $R_{\succ^*}(n, n')$ and $\mathbf{T}_S, n' \models \phi$
g. $\mathbf{T}_S, n \models \uparrow^* \phi$ iff there is an $n'$ s.t. $R_{\prec^*}(n, n')$ and $\mathbf{T}_S, n' \models \phi$
h. $\mathbf{T}_S, n \models P\phi$ iff there is an $n'$ s.t. $R_P(n, n')$ and $\mathbf{T}_S, n' \models \phi$
i. $\mathbf{T}_S, n \models F\phi$ iff there is an $n'$ s.t. $R_F(n, n')$ and $\mathbf{T}_S, n' \models \phi$

The clauses for the defined constants $s$ and $t$ are given in (20) from which it follows that they are true only at the root node and at terminal nodes, respectively.

(20) a. $\mathbf{T}_S, n \models s_0$ iff $n = root$
b. $\mathbf{T}_S, n \models t$ iff $t \in \Theta$

## *2.4 The Morphological Level M*

### 2.4.1 The Structures: Finite Rooted Feature Structures

Structures for the morphological level M are based on features structures. Blackburn and Meyer-Viol (1997) and Blackburn et al. (1993) define feature structures as labeled decorated directed graphs that are multi-modal Kripke models in which every relation is a partial function.

**Definition 2.3** A finite rooted feature structure **FS** of signature $\langle F, A \rangle$, $F$ the set of *features* and $A$ the set of *attributes* (or *decorations*) is a quadruple $\langle W, \{R_f\}_{f \in F}, \{Q_a\}_{a \in A}, w_0 \rangle$ such that

- $W$ is a non-empty set, the set of (feature structure) points
- each $R_f$, $f \in F$, is a binary relation on $W$ that is required to be a partial function

- each $Q_a$, $a \in A$, is a unary relation on $W$
- $w_0$ is the (unique) root of **FS**, i.e. **FS** is *generated* by $w_0$ in the sense that each point $w \in W$ can be reached from $w_0$ in a finite number of steps

Structures for the morphological level are pairs $\mathbf{M}_M = \langle\{\mathbf{FS}_n \mid n \in \mathbf{N}_S\}, \delta\rangle$ such that

- $\{\mathbf{FS}_n \mid n \in \mathbf{N}_S\}$ is a set of feature structures
- $\delta$ is a function that maps each $\mathbf{FS}_n$ to its root $w_{0,n}$

The set $\{\mathbf{FS}_n \mid n \in \mathbf{N}_S\}$ is the image of the functional and injective relation $R_{z_f}$, which assigns to each node $n \in N$ from the finite ordered binary tree $\mathbf{T}_S$ a feature structure $R_{z_f}(n) = \mathbf{FS}_n$. $\mathbf{T}_S$ therefore is, in effect, a *feature decorated* tree. Since each node has its own 'decorating' feature structure, the morphological level is based on the set of decorating feature structures. The composition of $R_{z_f}$ with the function $\delta$ assigns to each node $n \in N_S$ the root $w_{0,n}$ of the feature structure $\mathbf{FS}_n$ assigned to $n$ by $R_{z_f}$: $[\delta \circ R_{z_f}](n) = w_{0,n}$.

### 2.4.2 The Language $L^M$

The wffs of $L^M$ are defined as follows.

**Definition 2.4** Well-formed formulas of $L^M$ of signature $\langle F, A\rangle$: The set $L^M_{wff}$ is the smallest set such that (i) $a \in L^M_{wff}$ for each $a \in A$ (i.e. the set of attributes functions as the set of propositional variables for $L^M$), (ii) if $\phi$, $\psi \in L^M_{wff}$, then $\neg\phi, \phi\wedge\psi \in L^M_{wff}$ and (iii) if $\phi \in L^M_{wff}$, then $\langle f\rangle\,\phi \in L^M_{wff}$ for each $f \in F$, i.e. the elements of $F$ function as modal operators.

The choice of both $A$ and $F$ depends on the application as well as on the linguistic theory that is used. For present purpose, $A$ will be assumed to contain at least $s$, *np*, $v$ and *vp*, which are used to express purely categorial information about nodes, *nom*, *dat* and *acc*, which are the values of the feature $CASE$ as well as *pl* and *sg*, that are admissible values for the $NUM$ feature. Thus, $A$ can be taken as the union of the label sets $Cat = \{s, np, v, vp\}$, $Case = \{nom, dat, acc\}$ and $Num = \{sg, pl\}$. The set $F$ contains at least the features $CASE$ and $NUM(BER)$. An example of a wff of $L^M$ is $np \;\wedge\; \langle CASE\rangle\, acc \;\wedge\; \langle NUM\rangle\, sg$.

Elements of $L^M_{wff}$ are interpreted on feature structures **FS**. The satisfaction relation is given in (21).

(21) a. $\mathbf{FS}, w \models a$ iff $w \in Q_a$ for $a \in A$
b. $\mathbf{FS}, w \models \neg\phi$ iff $\mathbf{FS}, w \not\models \phi$
c. $\mathbf{FS}, w \models \phi \wedge \psi$ iff $\mathbf{FS}, w \models \phi$ and $\mathbf{FS}, w \models \psi$
d. $\mathbf{FS}, w \models \langle f\rangle\,\phi$ iff there is a $w'$ s.t. $R_f(w, w')$ and $\mathbf{FS}, w' \models \phi$

Furthermore, a feature structure **FS** satisfies an element $\phi$ of $L^M_{wff}$ just in case its root $\delta(\mathbf{FS})$ satisfies $\phi$, (21e).

(21) e. $\mathbf{FS} \models \phi$ iff $\mathbf{FS}, \delta(\mathbf{FS}) \models \phi$

### 2.4.3 Combining $L^S$ and $L^M$

The non-structural properties of a tree node are its syntactic and morphological properties. E.g. a node can be of syntactic category NP and is morphologically marked by dative case, singular number and masculine gender. The morphological properties *refine* the syntactic one: a node is not simply an NP but it is an NP of a particular sort. The sorting is defined by the morphological properties. The morphological level can therefore be said to give internal structure to the atomic categorial information at the syntactic level. Yet, nothing hinges on the fact that the categorial information is defined at the syntactic level. It is equally possible to treat it on a par with the morphological one, i.e. to analyze syntactic and morphological properties both at the morphological level M. Consequently, the non-structural information about a tree node is completely determined by its decorating feature structure.

At the level of structures, $\mathbf{T}_S$ and $\mathbf{M}_M$ are related by the injective function $R_{z_f}$ which assigns to each node $n \in N_S$ its decorating feature structure $R_{z_f}(n) = \mathbf{FS}_n$. Syntactically, this means that instead of building the tree language $L^T$ on top of a set VAR of propositional variables, it is built in terms of a set of structured atomic formulas of the form $\langle z_f \rangle \phi$, where $\phi$ is a formula from $L^M$ and the modality $z_f$ is interpreted by $R_{z_f}$. Thus, VAR is taken to be the set $\left\{ \langle z_f \rangle \phi \mid \phi \in L^M_{wff} \right\}$ so that syntactically the propositional variables $p \in$ VAR in Definition 2.2 are replaced by formulas of the form $\langle z_f \rangle \phi$ as defined above.[5] The base clause in the definition of satisfaction is given in (22).

(22) $\mathbf{T}_S, n \models \langle z_f \rangle \phi$ iff there is a $\mathbf{FS}$ s.t. $R_{z_f}(n, \mathbf{FS})$ and $\mathbf{FS} \models \phi$ iff $\mathbf{FS}_n \models \phi$

The resulting language $L^S(L^M)$ is called $L^S$ *layered over the language* $L^M$. This language consists of two layers. The top layer are formulas from $L^S$ as defined in Definition 2.2(ii) and 2.2(iii) above. They move us around the syntactic tree and are used to talk about structural (or configurational) properties of the tree. The base layer are formulas of the form $\langle z_f \rangle \phi$ for $\phi \in L^M$. These formulas express both syntactic and morphological information about a node $n$: what is the syntactic category of $n$?; what are the values of features like $CASE$ and $NUM(BER)$? They are therefore used to talk about the non-structural properties of nodes.

### 2.4.4 Constraining $L^S(L^M)$

So far no constraints have been imposed on the binary relations $R_i$ in $\mathbf{T}_S$ so that the atomic information can, in effect, be freely distributed at (the decorating feature structures of) the nodes of $\mathbf{T}_S$. For a feature decorated tree to be linguistically admissible, it must satisfy a number of constraints of how this information is distributed. One prominent example of a constraint are phrase structure rules, which constrain the distribution of purely categorial information. A typical example is $S \rightarrow NP\ VP$ which says that a local tree with root node of sort $s$ is admissible if it immediately dominates a left node of sort $vp$ and if it immediately dominates a right node of sort $np$. Formally, these rules impose restrictions on $R_{\succ_1}$ and $R_{\succ_2}$, i.e. the 'immediately

[5] In effect, there is a second type of atomic formulas which are of the form $\left\langle z_s^{-1} \right\rangle \phi$ with $\phi$ a well-formed formula from the language $L^{TS}$ of the TS level. Consequently, VAR is taken to be the set $\left\{ \langle z_f \rangle \phi \mid \phi \in L^M_{wff} \right\} \cup \left\{ \left\langle z_s^{-1} \right\rangle \phi \mid \phi \in L^{TS}_{wff} \right\}$. See Section 2.6.3 below for details.

left (right) dominates' relation. In $L^S(L^M)$ this rule is expressed by the formula in (23).

(23) $\langle z_f \rangle\, s \rightarrow \downarrow^1 \langle z_f \rangle\, np \wedge \downarrow^2 \langle z_f \rangle\, vp$

Examples for other constraints of this sort are (i) each node $n$ must be labeled by exactly one element from *Cat* and (ii) the root node of a tree must be labeled by $s$. These two constraints are expressed by $\bigvee_{p \in Cat} \langle z_f \rangle\, p \ \wedge\ \langle z_f \rangle\, (p \rightarrow \bigwedge_{q \in Cat \setminus \{p\}} \neg q)$ and $s_0 \rightarrow \langle z_f \rangle\, s$, respectively.

These constraints concern the distribution of purely categorial (syntactic) information and are not specific to LDG. The structural information, expressed by pure $L^S$ formulas, plays no role. Constraints on the relation between the syntactic and the morphological component imposed by LDG are of the form $\langle z_f \rangle\, p \wedge \bigwedge \alpha \, \langle z_f \rangle\, p \rightarrow \langle z_f \rangle \, \langle CASE \rangle\, \phi$ ($p \in Cat$, $\bigwedge \alpha\, \langle z_f \rangle\, p$ is a conjunction of formulas of the form $\alpha\, \langle z_f \rangle\, p$ for $\alpha \in \{(\neg)F, (\neg)P\}$). The antecedent expresses both purely categorial information, $\langle z_f \rangle\, p$, and a combination of structural and non-structural information, $\bigwedge \alpha\, \langle z_f \rangle\, p$. The latter formula (or formulas) make a move in the tree to find a node that carries the same categorial information as the node at which the whole formula expressing the constraint is evaluated. The antecedent, therefore, expresses information about the relative position of a node of category $p$ among the nodes in the tree that are of category $p$. The antecedent of the constraint formula requires the evaluating node to satisfy particular morphological information about the case feature of the node. When taken together, a formula of the form $\langle z_f \rangle\, p \wedge \bigwedge \alpha\, \langle z_f \rangle\, p \rightarrow \langle z_f \rangle\, \langle CASE \rangle\, \phi$ therefore imposes the constraint that a node of category $p$ which occupies a particular position among the nodes of that category has a particular case feature. Thus, particular categorial information at one place of the tree $\mathbf{T}_S$ requires some case information to be present at that node too. For a three-place verb like 'geben' the constraints are those in (24). (24a) and (24c) are illustrated in Figure 2.

(24) a. $\langle z_f \rangle\, np\ \wedge \neg P\, \langle z_f \rangle\, np \rightarrow \langle z_f \rangle\, \langle CASE \rangle\, nom$
b. $\langle z_f \rangle\, np\ \wedge\ P\, \langle z_f \rangle\, np \wedge\ F\, \langle z_f \rangle\, np \rightarrow \langle z_f \rangle\, \langle CASE \rangle\, dat$
c. $\langle z_f \rangle\, np\ \wedge \neg F\, \langle z_f \rangle\, np\ \wedge\ P\, \langle z_f \rangle\, np \rightarrow \langle z_f \rangle\, \langle CASE \rangle\, acc$

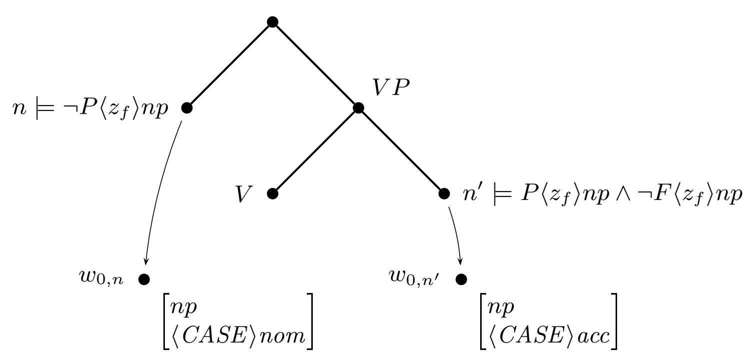


Figure 2

In effect, the constraints in (24) hold for any $n$-place verb that is canonical. If non-structural patterns of linking are taken into account, the constraints must be modified. This issue will be discussed in Section 2.6.3 below.

## 2.5 The SF- and the TS-Level

### 2.5.1 Dynamic Event Semantics

The formalization of the semantic level, consisting of the semantic form and the theta structure, is based on *Dynamic Event Semantics* (DES), Naumann (1998, 1999), Naumann and Osswald (1999, 2001). DES is based on the intuition that non-stative verbs express changes. A change can be conceived of either as an object (event) or as a transformation of state (TS). Events bring about results by transforming a state $s$ at which a result does not hold into a state $s'$ where the result holds. This double perspective is modeled in DES by having both an eventuality structure $\mathbf{E}$ with an underlying domain $E$ of events and a transition structure $\mathbf{S}$ with an underlying domain $S$ of states that is linearly ordered by $<_S$. Results $Q$ are subsets of $S$. Each event $e \in E$ is assigned its execution sequence $\tau(e)$, that is a convex subset of $S$.

The basic idea will be illustrated by two examples. Consider an event $e$ of type 'John give Mary a book'. If such an event occurs, a state in which John does have and Mary does not have the book is transformed into a state in which Mary but not John does have the book, i.e. the book is transferred from John to Mary. There are therefore two results that are brought about by the event which did not hold at its beginning: Mary has the book and John does not have the book. Besides these two results, there are the results which are brought about by the actions of John and which, when taken in isolation, do not constitute an event of giving. An example are his movements towards Mary. The corresponding results do usually not continue to hold after the event and can be evaluated only relative to the beginning point of the event. An event of type 'John eat apple' transforms a state in which the apple exists and is outside of John's stomach into a state where the apple no longer exists and its mass is inside the stomach of John. These results are brought about in stages with corresponding results 'Part of the apple decreased' and 'Part of the apple is in the stomach of John'. Similarly to the first example, there are in addition the results brought about by John which are not events of eating.

Linguistically, there are three aspects of results that are relevant. Results are of different types, they are temporally ordered and they are always brought about with respect to at least one object participating in the event. The first aspect concerns the way a result is evaluated on the execution sequences of events that bring it about. E.g. the results that Mary has but John does not have the book are true only at the end point of the event of giving. The same is true of the result that the mass of the apple is in the stomach of John. The corresponding result that part of the apple is in John's stomach, on the other hand, holds at all initial stages of the event that are eatings. The results corresponding to the actions that are no givings (or eatings) hold during the whole event, except at its beginning point. These distinctions classify results according to their type (relative to an event type, i.e. a set of events). Results that only hold at the end point of an event are called *maximal* (relative to an event type). *Minimality* of a result (relative to an event type) requires that it be true at all initial stages of the event that belong to the event type. A result that holds during the whole occurrence of an event belonging to an event type is *s-minimal* relative to this event type. In terms of these basic types of results further results are defined which exclude each other. The resulting classification is given in Table 3.[6]

[6] There are only five possibilities because s-minimality implies minimality so that the combinations $+-+$ and $--+$

| | maximal | minimal | s-minimal |
|---|---|---|---|
| s-maximal | + | − | − |
| w-maximal | + | + | − |
| w-minimal | − | + | − |
| s*-minimal | − | + | + |
| min-max | + | + | + |

Table 3

E.g. the results that Mary has but John does not have the book are w-maximal. The result that the mass of the apple is in the stomach of John is s-maximal since it is brought about incrementally. The corresponding result that part of the mass of the apple is in John's stomach is w-minimal. Examples for s*-minimal results are those corresponding to the initial actions by John.

The second aspect, the temporal ordering, is closely related to the first: s-minimal results are brought about not later than minimal results which are brought about not later than maximal results. This ordering is captured by the relation $\preceq$ on $\wp(S) \times \wp(S) \times \wp(E)$ which preorders the set of results determined by an event type relative to the event type. If $(Q, Q', P) \in \preceq$, the result $Q$ is brought about on elements of $P$ not later than the result $Q'$, i.e. whenever $Q'$ holds at a point of the execution sequence of an event $e \in P$, $Q$ holds at some point weakly preceding the former point.[7]

Each result is brought about with respect to at least one participant of an event. For an event of type 'John give Mary a book', one gets the assignment in Table 4. The number in brackets indicates the number of results of the type assigned to the participant.

| | s*-min | w-max |
|---|---|---|
| John | + | + (1) |
| Mary | − | + (1) |
| the book | − | + (2) |

Table 4

The above examples show that in DES all participants of an event undergo changes. E.g. although none of the objects in the event of giving is involved incrementally, Mary comes to have the book whereas John ceases to have it. Consequently, all three undergo a change effected by the event. The crucial observation is that each participant is assigned a different set of results. This makes it possible to uniquely pick out participants of events in terms of sets of results. The general idea is the following. The sets of results assigned to participants of events can be characterized by properties of their elements, e.g. the type to which they belong or, more generally, in terms of the preorder $\preceq$. The sets of results assigned to the participants of an event form a cover of the set of results determined by the basic event type to which the event belongs. Using the preorder $\preceq$ on this latter set, it is possible to define a linear

are excluded. The combination $- - -$, finally, is excluded because a result must be of at least one basic type.

[7] Thus, it is possible that $Q$ and $Q'$ are brought about at the same time.

order on this cover. In a second step relations $R_i$ on $E \times O \times \wp(E)$ are defined. A pair $(e, d, P)$ is an element of $R_i$ just in case $d$, which participates in $e \in P$, is assigned the $i$-th element of the cover of the set of results determined for $e$. In a final step, functional relations on $E \times O$, so called *dynamic roles*, are defined in terms of the $R_i$ by existentially quantifying over the event type argument.

Besides the preorder $\preceq$, the construction is based on two primitive relations. The relationship between event types, their elements, objects participating in those elements and results that the latter can bring about with respect to the former is captured by the relation $\Delta$ on $\wp(E) \times E \times O \times \wp(S)$. A quadruple $(P, e, d, Q)$ is an element of $\Delta$ just in case $Q$ is a result that the event $e \in P$ can bring about with respect to its participant $d$. The set $\underline{Q}_{d,e,P} = \{Q \mid Q \in \Delta(P, e, d)\}$ is the set of all results assigned to $d$ by $e$ (relative to $P$) and the set $\underline{Q}_{e,P} = \{Q \mid \exists d : \Delta(P, e, d, Q)\}$ is the set of all results determined by $P$ for $e$. The set of latest results determined for an event relative to an event type and the ordering $\preceq$ is defined by $\text{Latest}(e, P) = \{Q \mid Q \in \underline{Q}_{e,P} \wedge \forall Q'(Q' \in \underline{Q}_{e,P} \rightarrow \preceq (Q', Q, P))\}$. The relation $\Delta^*$ on $\wp(E) \times E \times O \times \wp(S)$ is defined in terms of $\Delta$ as follows. For events that are neither point-like nor the elements of stative event types, $\Delta$ and $\Delta^*$ are identical, i.e. one has: $\Delta(P, e, d) = \Delta^*(P, e, d)$. For point-like events, $\Delta^*(P, e, d)$ is $\Delta(P', e', d)$ with $e' \in P'$ the largest event of which $e$ is a boundary. For events belonging to stative event types, $\Delta^*(P, e, d)$ is $\Delta(P', e', d)$ with $e' \in P'$ the event that brings about the only element of $\underline{Q}_{e,P}$ and of which $e$ is the consequent state (or consequent event).

The function $\gamma$ maps an event $e$ and an event type $P$ to a particular cover of the set of results $\underline{Q}_{e,P}$ assigned to $e$. This cover is verb-dependent although it depends on the verb's aspectual classification. For instantaneous event types, i.e. event types the elements of which are point-like, $\gamma$ is a cover of the set of results brought about by the presupposed event, i.e. a cover of the set of results determined for the event of which $e$ is a boundary. For events denoted by stative event types, the value of $\gamma$ is the set of results determined for the event that brings about the result that is the only element of $\underline{Q}_{e,P}$, i.e. $e'$ is the event of which $e$ is the consequent state. In (26), $\gamma$ is illustrated for the examples in (25). If both s-maximal and w-minimal results are determined, only the former are listed. ($j$ = john, $m$ = mary, $b$ = the book, $d$ = the door, $a$ = the apple, $c$ = the cart, $s$ = the station, $p$ = paul). Be-affected($d$) is used in a twofold sense. It either means that $d$ produces a sound or that $d$ is hurt.

(25) a. John gave Mary the book.
b. John closed the door.
c. John ate the apple.
d. John pushed the cart.
e. John reached the station.
f. John hit Paul.

(26) a. $\gamma(e, P_{give}) = \{\{Act(j), \neg Poss(j, b)\}, \{Poss(m, b)\}, \{Poss(m, b), \neg Poss(j, b)\}\}$
b. $\gamma(e, P_{close}) = \{\{Act(j)\}, \{Closed(d)\}\}$
c. $\gamma(e, P_{eat}) = \{\{Act(j), Be\text{-}in(a, j)\}, \{Be\text{-}in(a, j), \neg Exist(a)\}\}$
d. $\gamma(e, P_{push}) = \{\{Act(j)\}, \{Move(c)\}\}$
e. $\gamma(e, P_{reach}) = \{\{Move(j), Be\text{-}at(j, s)\}, \{Be\text{-}at(j, s)\}\}$
f. $\gamma(e, P_{hit}) = \{\{Act(j)\}, \{Be\text{-}\text{affected}(p)\}\}$

The elements of the cover of the set of results determined for an event can be linearly ordered in terms of the temporal ordering $\preceq$ defined on the set of results. The basic distinction is that between elements of the cover which contain a result that is brought about first, i.e. that is a minimal element relative to $\preceq$ (and a basic event type) and elements which contain a result that is brought about last, i.e. which is a maximal element relative to $\preceq$ (and a basic event type). The exact definitions are given in (27) ($\underline{Q}$ is a set of states and $\underline{\underline{Q}}$ a set of set of states).

(27) a. W-Prec($\underline{Q}, \underline{\underline{Q}}, P$) iff $\underline{Q} \in \underline{\underline{Q}} \wedge \exists Q \in \underline{Q} : \forall \underline{Q'} \in \underline{\underline{Q}} : \forall Q' \in \underline{Q'} : \preceq (Q, Q', P)$

b. W-Suc($\underline{Q}, \underline{\underline{Q}}, P$) iff $\underline{Q} \in \underline{\underline{Q}} \wedge \exists Q \in \underline{Q} : \forall \underline{Q'} \in \underline{\underline{Q}} : \forall Q' \in \underline{Q'} : \preceq (Q', Q, P)$

This distinction does in general not linearly order the elements of the cover in the sense that for a given $\underline{\underline{Q}}$ and $P$ $\underline{Q}$ is uniquely determined. E.g. the elements of the cover in (26a) all contain a result that is brought about last so that they all stand in the relation of weak succession to $\gamma(e, P_{give})$ and $P_{give}$. It is therefore necessary to apply further criteria. An element of a cover can both weakly precede and weakly succeed the other elements of the cover. This is the case for the set $\{Act(j), \neg Poss(j, b)\}$ from (26a). This set can therefore be distinguished from the other elements of this cover by satisfying both relations. The two remaining sets can be discerned from each other in terms of their cardinality. Let $s_1 = \{Act(j), \neg Poss(j, b)\}$, $s_2 = \{Poss(m, b)\}$ and $s_3 = \{Poss(m, b), \neg Poss(j, b)\}$. Using the criteria from above, one gets Table 5.

| | |
|---|---|
| W-Prec | $s_1$ |
| W-Suc | $s_1, s_2, s_3$ |
| ¬W-Prec ∧ W-Suc | $s_2, s_3$ |
| ¬W-Prec ∧ W-Suc ∧ MaxCard | $s_3$ |

Table 5

MaxCard is counted relative to the elements that satisfy ¬W-Prec and W-Suc.[8] Whereas set $s_1$ is the element of the cover that weakly precedes all other, set $s_3$ is the largest set among the sets that weakly succeed but that do not weakly precede all other elements. Set $s_2$, finally, is the set that satisfies none of the criteria picking out the other two sets. This is made precise in the definitions below.

(28) a. First($\underline{Q}, \underline{\underline{Q}}, P$) iff W-Prec($\underline{Q}, \underline{\underline{Q}}, P$) $\wedge \forall \underline{Q'}(\underline{Q'} \in \underline{\underline{Q}} \wedge$ W-Prec($\underline{Q'}, \underline{\underline{Q}}, P$) $\rightarrow |\underline{Q}| \leq |\underline{Q'}|)$

b. Last($\underline{Q}, \underline{\underline{Q}}, P$) iff W-Suc($\underline{Q}, \underline{\underline{Q}}, P$) ∧ ¬W-Prec($\underline{Q}, \underline{\underline{Q}}, P$) ∧ $\forall \underline{Q'}(\underline{Q'} \in \underline{\underline{Q}} \wedge$ W-Suc($\underline{Q'}, \underline{\underline{Q}}, P$) ∧ ¬W-Prec($\underline{Q'}, \underline{\underline{Q}}, P$) $\rightarrow |\underline{Q'}| \leq |\underline{Q}|)$

c. Int($\underline{Q}, \underline{\underline{Q}}, P$) iff $\neg \Big($First($\underline{Q}, \underline{\underline{Q}}, P$) ∨ Last($\underline{Q}, \underline{\underline{Q}}, P$)$\Big)$

Using the relations in (28), each of the three sets can be uniquely determined. Applying the relations to the examples in (25), one gets the table below.

8 In general, MaxCard can be defined in terms of the subset relation.

| | First | Int | Last |
|---|---|---|---|
| give | $\{Act(j), \neg Poss(j,b)\}$ | $\{Poss(m,b)\}$ | $\{Poss(m,b), \neg Poss(j,b)\}$ |
| close | $\{Act(j)\}$ | – | $\{Closed(d)\}$ |
| eat | $\{Act(j), Be\text{-}in(a,j)\}$ | – | $\{Be\text{-}in(a,j), \neg Exist(a)\}$ |
| push | $\{Act(j)\}$ | – | $\{Move(c)\}$ |
| reach | $\{Move(j), Be\text{-}at(j,s)\}$ | – | $\{Be\text{-}at(j,s)\}$ |
| hit | $\{Act(j)\}$ | – | $\{Be\text{-}\mathrm{affected}(p)\}$ |

Table 6

If $\underline{\underline{Q}}$ is the value of $\gamma$ for an event $e$ belonging to the event type $P$, each of its elements is a subset of the set of results assigned to an object participating in $e$. It is therefore possible to define, for each of the relations $R$ in (28), a corresponding relation $R^*$ on $E \times O \times \wp(E)$ that holds between an event $e$, an object $d$ and an event type $P$ just in case $d$ participates in $e$, $e$ belongs to $P$ and the set of results that bears $R$ to $\gamma(e,P)$ and $P$ is a subset of $\Delta^*(P,e,d)$. The relations are defined in (29).

(29) a. First*$(e,d,P)$ iff $\exists \underline{Q}(\underline{Q} \subseteq \Delta^*(P,e,d) \wedge$ First$(\underline{Q}, \gamma(e,P), P))$
b. i Last**$(e,d,P)$ iff $\exists \underline{Q}(\underline{Q} \subseteq \Delta^*(P,e,d) \wedge$ Last$(\underline{Q}, \gamma(e,P), P))$
b. ii Last*$(e,d,P)$ iff Last**$(e,d,P) \wedge \forall d'(d \neq d' \wedge$ Last**$(e,d',P) \rightarrow$ First*$(e,d',P) \vee$ Int*$(e,d',P))$
c. Int*$(e,d,P)$ iff $\exists \underline{Q}(\underline{Q} = \Delta^*(P,e,d) \wedge \neg($First$(\underline{Q}, \gamma(e,P), P) \vee$ Last$(\underline{Q}, \gamma(e,P), P)))$

For First* and Last*, only a subset of the set of results is required to bear the corresponding relation to $\gamma(e,P)$ and $P$ because if a verb is used reflexively there is a single object that is assigned both roles. Since for transfer verbs no reflexive uses are possible, one can set $\underline{Q} = \Delta^*(P,e,d)$. The more complicated definition of Last* is necessary to exclude objects that bear both the relation First* and Last** to an event because a subset of the results assigned to them bears Last to $\gamma(e,P)$ and $P$. For instance for an event $e$ of type 'John reach the station', the set $\{Be{-}at(j,s)\}$ stands in the relation Last to $\gamma(e, P_{reach})$ and $P_{reach}$. It is a proper subset of the results assigned to John and the set of results assigned to the station. Thus, Last**, which is defined analogously to First*, is not functional in its second argument given $e$ and $P_{reach}$. For John, there is an additional result assigned to him that, together with the result $Be\text{-}at(j,s)$, stands in the relation First to $\gamma(e, P_{reach})$ and $P_{reach}$. Consequently, John stands both in the relation First* and Last** to $e$ and $P_{reach}$. He is therefore excluded by the second conjunct of the definition of the relation Last*. An analogous argument applies to events belonging to event types corresponding to transfer verbs of the 'kaufen' and 'legen' type. In this case an object participating in an event of this type can stand both in the relation Last** and Int* to an event and an event type. Verbs of these types are discussed below in Section 2.5.2.

The relations in (29) are functional in the following sense: $\forall e \forall P \forall d \forall d'(R^*(e,d,P) \wedge R^*(e,d',P) \rightarrow d = d')$ for $R^* \in \{$First*, Last*, Int*$\}$. The relations in (29) therefore determine participants of an event (relative to an event type) by requiring that a subset of the set of results assigned to it both be an element of $\gamma(e,P)$ and satisfy one of the properties defined in (28). Since those relations are functional in their first argument for a given $\gamma(e,P)$ and $P$ and since that argument is assigned to a unique participant, a unique participant of events is determined by the relations in (29). Similarly to thematic roles, the relations in (29) are required to be independent of a particular event type, i.e. one has $\forall e \forall P \forall P' \forall d \forall d' (R^*(e,d,P) \wedge R^*(e,d',P') \rightarrow d = d')$. It is therefore possible to define corresponding functional relations on $E \times O$ by existentially quantifying over the event type. This yields (30). The resulting relations are called *basic dynamic roles*.[9]

(30) $R_{dr}(e,d)$ iff $\exists P : R_{dr^*}(e,d,P) \qquad dr \in \{\text{First}, \text{Int}, \text{Last}\}$.

On the dynamic roles an ordering $\ll$ is defined as follows.

(31) $R_{first} \ll R_{int} \ll R_{last}$

The function $\mu$ assigns to each basic event type an ordered set of dynamic roles. E.g. for ditransitive verbs like 'give', $\mu(P_{give}) = \{R_{first}, R_{int}, R_{last}\}$. For transitive verbs like 'eat', $R_{int}$ is not defined so that one has $\mu(P_{eat}) = \{R_{first}, R_{last}\}$. For intransitive verbs, one either sets $\mu(P) = \{R_{first}\}$, or one uses a generalized dynamic role which subsumes both $R_{first}$ and $R_{last}$ (see Naumann and Osswald 2001 for details).

### 2.5.2 Verbs with Prepositional Arguments

All verbs that have been considered so far link their arguments by case. In German, there are also verbs which have prepositional arguments. Two examples are given in (32).

(32) a. Hans kaufte ein Buch von Maria. (Hans bought a book from Mary)
b. Hans legte ein Buch auf den Tisch. (Hans put a book on the table)

Similarly to 'geben' verbs, these verbs express transfers. E.g. the book is transferred from Mary to Peter by the event of buying and it is transferred to the table from some unspecified source by the event of putting. For elements of the 'kaufen' class, it is the source and for elements of the 'legen' class, it is the destination which is realized by a prepositional argument. In (33), the values of $\gamma$ for the different types of transfer verbs are given ($h$ = Hans, $b$ = book, $m$ = Mary and $t$ = table).

(33) a. $\gamma(e, P_{give}) = \{\{Act(h), \neg Poss(h,b)\}, \{Poss(m,b)\}, \{Poss(m,b), \neg Poss(h,b)\}\}$
b. $\gamma(e, P_{\text{buy}}) = \{\{Act(h), Poss(h,b)\}, \{Poss(h,b), \neg Poss(m,b)\}, \{\neg Poss(m,b)\}\}$
c. $\gamma(e, P_{put}) = \{\{Act(h)\}, \{Move(b), Be\text{-}at(b,t)\}, \{Be\text{-}at(b,t)\}\}$[10]

The example of 'kaufen' verbs in (33b) shows that the definition of the Last relation is too simple. For 'kaufen', it picks out the transferred object since it is assigned the largest set which weakly succeeds the elements of the cover $\gamma(e, P_{buy})$. But the transferred object is denoted by the intermediate argument. Similarly, the source is picked out by the relation Int so that it should be realized as the intermediate argument. Yet it is realized as the last argument. The revised definition is given in (34). ($\approx (Q,Q',P) =_{\text{def.}} \prec (Q,Q',P) \wedge \prec (Q',Q,P)$.)

(34) $\text{Last}(\underline{Q}, \underline{\underline{Q}}, P)$ iff $\text{W-Suc}(\underline{Q}, \underline{\underline{Q}}, P) \wedge \neg\text{W-Prec}(\underline{Q}, \underline{\underline{Q}}, P) \wedge \forall \underline{Q}' (\underline{Q}' \in \underline{\underline{Q}} \wedge \text{W-Suc}(\underline{Q}', \underline{\underline{Q}}, P) \wedge \neg\text{W-Prec}(\underline{Q}', \underline{\underline{Q}}, P) \rightarrow (\exists Q \in \underline{Q} \forall Q' \in \underline{Q}' (\prec (Q,Q',P)) \vee (\forall Q \in \underline{Q} \forall Q' \in \underline{Q}' (\approx (Q,Q',P) \wedge \exists Q \in \underline{Q} \forall Q' \in \underline{Q}' (Q' \subseteq Q) \wedge \exists Q' \in \underline{Q}' \forall Q \in \underline{Q} (Q' \subseteq Q)))$

---

[9] The way dynamic roles are defined can be taken as a formalization of Dowty's (1991) notion of a proto role.

[10] Instead of the cover in (33c), the following cover can be used: $\{\{Act(h), \neg Be\text{-}at(b,h)\}, \{Move(b), Be\text{-}at(b,t)\}, \{Be\text{-}at(b,t), \neg Be\text{-}at(b,h)\}\}$. The argument given below applies to both analyses.

The first disjunct in the succedent of the last conjunct is used for verbs of the ‘legen’ class for which the destination is assigned exactly one w-maximal result whereas the transferred object is in addition assigned a non-latest result.

Events of type buying differ from those of type giving in the direction that the transfer takes. For an event of giving, the transfer is always away from the actor whereas for an event of type buying it is always towards the actor. Consequently, for the former type of event the actor is the source of the transfer and the object denoted by the intermediate argument is the destination (or the recipient). For an event of type buying, on the other hand, the actor is the destination of the transfer whereas the source is realized by the prepositional argument. For putting events, the destination is never the actor but the object denoted by the prepositional argument. In addition, the transfer is not or need not be directed away from the actor.

The similarities and differences between the three types of transfer verbs are reflected at the level of result as follows. Independently of the subclass, the transferred object is always assigned all latest results. For elements of the ‘kaufen’ class, these results are derived from the same property of individuals as for the ‘geben’ class, though the assignments to participants is different as shown above. For an element of the ‘legen’ class, there is only one latest result: $y$ is at $z$, with $y$ the transferred object and $z$ the destination. For a verb of the ‘geben’ class, the actor is assigned the logically weaker whereas the recipient is assigned the logically stronger of the two latest results. For an element of the ‘kaufen’ class, the actor is assigned the logically stronger and the source the logically weaker of the latest results. Consider next the three roles defined in (35).

(35) a. $\text{Role}_1 = \{(e, d, P) \mid \forall Q(\Delta(P)(e)(d)(Q) \leftrightarrow (Q \in \text{Latest}(e, P) \wedge \forall Q'(Q' \in \text{Latest}(e, P) \rightarrow Q \subseteq Q')))\}$

b. $\text{Role}_2 = \{(e, d, P) \mid \forall Q(\Delta(P)(e)(d)(Q) \leftrightarrow (Q \in \text{Latest}(e, P)))\}$

c. $\text{Role}_3 = \{(e, d, P) \mid \forall Q(\Delta(P)(e)(d)(Q) \leftrightarrow (Q \in \text{Latest}(e, P) \wedge \forall Q'(Q' \in \text{Latest}(e, P) \rightarrow Q' \subseteq Q)))\}$

$\text{Role}_1$ is assigned to an object that is assigned only the logically strongest of the latest results. Conversely, an object bears $\text{Role}_3$ to an event and an event type if the results assigned to it are the logically weakest of the latest results. $\text{Role}_2$, finally, is assigned to an object if all and only the latest results are brought about with respect to it. $\text{Role}_1$ and $\text{Role}_2$ apply to the intermediate and the lowest arguments of verbs belonging to the ‘geben’ class since the sets of results assigned to them satisfy the conditions in terms of which these roles are defined. $\text{Role}_2$ also applies to the intermediate argument of a verb of the ‘kaufen’ class. $\text{Role}_3$, finally, applies to the source argument of a verb belonging to the ‘kaufen’ class. Since the object denoted by this argument also satisfies $R_{last^*}$, a sufficient condition for an argument of a verb in German to be realized by a PP is (36).

(36) An argument of a verb in German is realized by a PP if the verb semantically determines maximal results and if the argument is linked to the verb by $R_{\text{last}^*}$ and $\text{Role}_3$.

The restriction to verbs that semantically determine latest results of type maximal is necessary to exclude stative and Activity verbs. The internal arguments of Accomplishment verbs are excluded because they are assigned not only latest results (type s-maximal) but in addition also a non-latest result (type w-minimal).

For the prepositional argument of a verb belonging to the 'legen' class, the argument is similar. Since there is only one latest result which is assigned both to the transferred object and the destination, it is not possible to discern the two participants in terms of the latest results. The result distinguishing the two participants is that the transferred object but not the destination is moved. As a consequence, there is an additional (s-minimal) result assigned to the former but not to the latter. From this it follows that none of the three roles in (35) applies to the transferred object because they all require the set of results assigned to an object to be a subset of the latest results. For the destination, on the other hand, all three roles apply because there is exactly one latest result and exactly this result is assigned to it. Since the destination bears in addition $R_{last^*}$ to an event and event type of the 'legen' class, the condition in (36) is satisfied. The argument for 'legen' verbs also applies to verbs like 'werfen' (throw) if they are used ditransitively. Consider an event of type 'Hans throw the ball to Mary'. The cover is identical to that used for 'legen': $\gamma(e, P_{throw}) = \{\{Act(h)\}, \{Move(b), Be\text{-}at(b,t)\}, \{Be\text{-}at(b,t)\}\}$.[11] Consequently, the same argument applies.

So far, no alternations have been considered. In German, alternations are possible for verbs of the 'geben' class: 'Hans brachte der Frau ein Buch' vs. 'Hans brachte ein Buch zu der Frau' (Hans brought a book to the woman). This possibility is captured by the following generalization of (36).

(36') An argument of a verb in German can be realized by a PP if the verb semantically determines maximal results and if an object denoted by the argument is assigned exactly one result which is a latest result.

(36') not only applies to the prepositional arguments discussed so far but also to the recipient of a verb of the 'geben' class. Formally, this additional possibility is accounted for by weakening the condition in the relation Last. The conjunct $\exists Q \in \underline{Q} \forall Q' \in \underline{Q}'(Q' \subseteq Q)$ in the second disjunct must be dropped. This yields (34'). In addition, the definition of the relation Int* must be adapted.

(34') $\text{Last}(\underline{Q}, \underline{\underline{Q}}, P)$ iff $\text{W-Suc}(\underline{Q}, \underline{\underline{Q}}, P) \wedge \neg\text{W-Prec}(\underline{Q}, \underline{\underline{Q}}, P) \wedge \forall \underline{Q}'(\underline{Q}' \in \underline{\underline{Q}} \wedge \text{W-Suc}(\underline{Q}', \underline{\underline{Q}}, P) \wedge \neg\text{W-Prec}(\underline{Q}', \underline{\underline{Q}}, P) \rightarrow (\exists Q \in \underline{Q} \forall Q' \in \underline{Q}'(\prec (Q, Q', P)) \vee (\forall Q \in \underline{Q} \forall Q' \in \underline{Q}'(\approx (Q, Q', P) \wedge \exists Q' \in \underline{Q}' \forall Q \in \underline{Q}(Q' \subseteq Q)))$

(29)c'. $\text{Int}^*(e, d, P)$ iff $\exists \underline{Q}(\underline{Q} = \Delta^*(P, e, d) \wedge (\neg(\text{First}(\underline{Q}, \gamma(e, P), P) \vee \text{Last}(\underline{Q}, \gamma(e, P), P)) \vee (\text{Last}(\underline{Q}, \gamma(e, P), P) \wedge \exists d'(d' \neq d \wedge \underline{Q} = \Delta^*(P, e, d')))))$

Using the above definitions, does not impose an ordering on the recipient and the transferred object for verbs of the 'geben' class. This must be accounted for in the definition of the theta structure and the semantic form. In the sequel, only the basic theta structure of 'geben' verbs is considered, leaving the further analysis of alternations to another occasion.

[11] The difference consists in the way the actor is involved. In a throwing event, he is not involved until the end whereas this is the case for an event of putting.

## 2.6 The Theta Structure TS

### 2.6.1 Structures for TS

In DES, the theta structure $D$ of a verb is defined in terms of the inverse of $\mu(P)$ with $P$ the event type corresponding to the verb. Formally, it is a linearly ordered set of arguments with the latter simply taken as abstract points. The cardinality of $D$ is that of $\mu(P)$ and the $i$-th element of this set is required to be linked to the $i$-th element of the inverse of $\mu(P)$. At the formal level this relationship is expressed as follows. Each element of a TS is characterized by structural and non-structural properties. Structural properties are those properties which an argument has in virtue of being an element of a linearly ordered set like being the first (last) element or having a predecessor (successor). Non-structural properties are those properties an element has in virtue of being related to particular dynamic roles. In models for the TS, these properties correspond to unary relations on the underlying domain of arguments. One has $\mathbf{M}_{TS}, \theta \models \phi_{dr}$ just in case the argument $\theta$ is semantically linked to the dynamic role $R_{dr}$. The relationship between elements of $D$ and the inverse of $\mu(P)$ is then expressed by axioms of the form 'if $\theta$ is the $i$-th element of $D$, then it satisfies $\phi_{dr}$' with $R_{dr}$ the $i$-th element of the inverse of $\mu(P)$.

**Definition 2.5** A model $\mathbf{M}_{TS}$ of signature $\langle DR^* \rangle$ is a pair $\langle D, \{Q_{dr}\}_{dr \in DR^*}, R_{v_{dat}} \rangle$ such that

- $D$ is a set that is linearly ordered by $<$; the elements of $D$ are called arguments
- each $Q_{dr}$ and $R_{v_{dat}}$ are unary relations on $D$; $DR^*$ is a set of dynamic role labels with $DR \subseteq DR^*$. The elements of $DR^* - DR$ are additional dynamic roles like those defined in (35) above. For each element from $DR$ and the label $dat \in DR^*$, $Q_{dr}$ is either a singleton or the empty set.[12] For those $dr \in DR$ for which $R_{dr}$ does not correspond to elements from $D$, $Q_{dr}$ is the empty set. Including labels for all dynamic roles is necessary for the formulation of linking rules. For canonical linking patterns, it must be required that some argument not be linked to a particular dynamic role, independently of whether the latter are defined for $P$ or not.

### 2.6.2 The Language $L^{TS}$

The non-logical part of $L^{TS}$ consists of the elements from $DR^*$ and $\bullet_{dat}$, which function as propositional variables, the modal operators $\uparrow$ and $\downarrow$, in terms of which two further operators, $\perp_\uparrow$ and $\perp_\downarrow$, are defined, the modality $z_s$ and the well-formed formulas from $L^S$.[13] The propositional variables express properties of dynamic roles whereas the modal operators are used to make assertions about the linear ordering on $D$

. The interpretation of the modal operators only depends on the ordering $<$ and not on specific properties of the elements of $D$. The satisfaction clauses for elements

[12] Formally the corresponding $\phi_{dr}$ therefore resemble *nominals* in the sense of Blackburn (1993): they are true of at most one element of the underlying domain. Nominals are required to be true at *exactly* one element of the domain.

[13] In terms of Section 2.4.3 $L^{TS}$ therefore is, in effect, $L^{TS}(L^S)$, $L^{TS}$ layered over $L^S$. Since formulas of the form $\langle z_s \rangle \phi$ for $\phi \in L^S_{wff}$ are only discussed below in Section 2.6.3, only the top layer $L^{TS}$ is defined in this section.

from $DR^*$ and for the modal operators are given in (37).[14, 15]

(37) a. $\mathbf{M}_{TS}, \theta \models \phi_{dr}$ iff $\theta \in Q_{dr}$
b. $\mathbf{M}_{TS}, \theta \models \uparrow$ iff there is a $\theta'$ s.t. $\theta < \theta'$
c. $\mathbf{M}_{TS}, \theta \models \downarrow$ iff there is a $\theta'$ s.t. $\theta' < \theta$
d. $\mathbf{M}_{TS}, \theta \models \bullet_{dat}$ iff $\theta \in R_{v_{dat}}$

An argument is an element of $R_{v_{dat}}$ just in case it is an element of the theta structure of a dative verb like 'helfen' or 'folgen'.[16] In terms of $\uparrow$ and $\downarrow$ two further operators, $\bot_{\uparrow}$ and $\bot_{\downarrow}$, are defined, (38a,b).

(38) a. $\bot_{\uparrow} =_{def.} \uparrow \vee \neg \uparrow$
b. $\bot_{\downarrow} =_{def.} \downarrow \vee \neg \downarrow$

According to (38a,b), $\bot_{\uparrow}$ and $\bot_{\downarrow}$ hold at each position of $D$. The relationship between the features $hr$ and $lr$ in LDG and the modal operators is given in (39).

(39) $\uparrow \equiv +hr \qquad \neg \uparrow \equiv -hr \qquad \downarrow \equiv +lr \qquad \neg \downarrow \equiv -lr$

A model $\mathbf{M}_{TS}$ has to satisfy certain axioms like those in (40).

(40) a. $\uparrow \wedge \downarrow \rightarrow \neg\phi_{dat}$
b. $\neg \uparrow \rightarrow \phi_{first}$
c. $\uparrow \wedge \downarrow \rightarrow \phi_{int}$
d. $\uparrow \wedge \neg \downarrow \rightarrow \phi_{last}$
e. $\uparrow \wedge \neg \downarrow \wedge \bullet_{v_{dat}} \rightarrow \phi_{dat}$

(40a) says that if a position of TS is an intermediate one, i.e. if there is both a lower and a higher position, the dynamic role at that position is *not* $R_{dat}$, i.e. the role assigned by dative verbs like 'helfen' to their internal argument. The axioms in (40b-d) express the relationship between arguments and basic dynamic roles. They can be derived from the axioms expressing the relationship between the SF and the theta structure given below in Section 2.7.4. The axioms in (40a-d) are verb-independent in the sense that the antecedent contains no information that is verb-dependent. In contrast, the antecedent of axiom (40e) expresses non-structural information that is specific to a particular verb class the elements of which are non-canonical. The antecedent determines the arguments of these theta structures that are exceptional in the sense that case is not assigned in accordance with the generalizations in (4) and (5) from Section 1. Similarly to $\bullet_{v_{dat}}$, the propositional variable $\phi_{dat}$ expresses non-structural information that is specific to particular argument positions, namely those arguments of dative verbs which do not comply with the canonical pattern. Thus, non-structural patterns of linking are explained by introducing non-structural information that is specific to particular verb classes. This information can be derived neither from the linear ordering on the set of arguments nor from the verb-independent

---

[14] No distinction between modalities and (boolean combinations of) propositional variables is made at the syntactic level. Modalities and (boolean combinations of) propositional variables can be distinguished by writing $\langle Mod \rangle \top$ instead of $Mod$ for $Mod$ a modality and adjusting the satisfaction clauses accordingly. see also the next footnote.

[15] $\uparrow$ and $\downarrow$ can be defined in TL as follows: $\uparrow =_{def.} F\top$ and $\downarrow =_{def.} P\top$. Since $F\phi$ can be defined as $\langle > \rangle \phi$, where $[\![ > ]\!] = R_{>}$, i.e., $\mathbf{M}, s \models \langle > \rangle \phi$ iff $\exists s'((s, s') \in R_{>})$. Let $R_{<} = (R_{>})^c$, i.e., $(s, s') \in R_{<}$ iff $(s', s) \in R_{>}$, as usual. Then $<$ can be defined as $>^c$, i.e. $\mathbf{M}, s \models \langle < \rangle \phi$ iff $\exists s'((s, s') \in R_{<} \wedge \mathbf{M}, s' \models \phi)$ iff $\exists s'((s', s) \in R_{>} \wedge \mathbf{M}, s' \models \phi)$.

[16] Thus for theta structures of non-dative verbs, $\bullet_{dat}$ is false for each element of $D$.

non-structural information that is derived from the relationship between the semantic form and the theta structure. Formally, this is done by introducing a second sort of formulas, i.e. the $\phi_{dr}$ with $dr \in DR^* - DR$, that express this non-structural information. These variables are interpreted by dynamic roles which are more specific than the three basic ones in terms of which the ordering on the arguments is determined. Three examples for such roles have been given in Section 2.5.2 above.

The above scheme not only applies to verbs with at most three arguments but also to verbs with more arguments. If LDG and other approaches are right that any additional arguments can only be linked non-structurally, this is handled in the present framework by adding further sorts of propositional variables which express the required type of information. Linking rules must then be changed accordingly by adding the required information in the antecedents. If, contrary to what is assumed in LDG, there are languages with verbs that have more than three structural arguments, two ways are open. One either continues to use the Priorian tense logic and uses more complex properties that are expressible in this logic. If the logic is too weak to express the required properties, one has to use a more expressive temporal logic or any other logic in which one can talk about linearly ordered sets that admits to express the linguistically required structural properties.

### 2.6.3 Constraining $L^{TS}(L^S)$

Recall from Section 1 that the relationship between TS and S concerns the notion of argument. The order of arguments at the level of TS is the inverse of the order in which they are realized at the level of word order at S. In the formalization given in this article, this means that there is a (functional) relationship between $D$ and $N$: each element from $D$ is assigned its tree node. The function that maps elements from $D$ to tree nodes will not be surjective. The image of this function is the subset of $N$ that consists of all and only those nodes which are arguments (complements) of the verb, which means that they are of a (maximal) phrasal category. Since in this article only arguments of category np are considered, the subset consists of all and only those nodes that are of category np.[17] The elements of this subset form a linear sequence that corresponds to the word order. It is defined in (41).

(41) $\lambda(N) = \{n \in N \mid \mathbf{T}_S, n \models \langle z_f \rangle\, np\}$

The relationship between $\mathbf{M}_{TS}$ and $\mathbf{T}_S$ is defined in terms of $\lambda(N)$.[18]

(42) $R_{z_s} = \{(\theta, n) \mid \exists\sigma\exists\sigma'(\text{prefix}(\sigma, D) \wedge \text{ last}(\sigma) = \theta \wedge \text{ prefix}(\sigma', \lambda(N)) \wedge \text{ last}(\sigma') = n \wedge \text{ length}(\sigma) = \text{length}(D) - \text{ length}(\sigma') + 1)\}$

$R_{z_s}$ maps the $k$-th element of $D$ to the $n-k+1$-th element of $\lambda(N)$ for $n$ the length of $D$. It is not only functional but, in effect a one-to-one correspondence.

[17] Thus the trees that are considered in this article are those which are determined by the subcategorization frame of the verb. E.g. for a transitive verb, this tree has a root of category s with a left daughter of category np, which is a terminal node, and a right daughter of category vp, which has a left daughter of category v and a right daughter of category np, both of which are terminal nodes. More complex trees can be generated by operations of tree adjoining.

[18] In (42) linearly ordered sets are identified with the corresponding sequences. Alternatively, $R_{z_s}$ can be defined as follows: $(\theta, n) \in R_{z_s}$ iff $\exists M \exists M'(M \subseteq D \wedge M' \subseteq \lambda(N) \wedge \min(D) \in M \wedge \min(\lambda(N)) \in M' \wedge \text{ closed}(M) \wedge \text{ closed}(M') \wedge card(M) = card(D) - card(M') + 1 \wedge \max(M) = \theta \wedge \max(M') = n)$. Given a linearly ordered set $M_{\leq}$, its maximum is defined as $\max(M) = \iota s(s \in M \wedge \forall s'(s' \in M \rightarrow s' \leq s))$. The minimum is defined analogously; 'closed$(M)$' iff $\forall s \forall s'(s \in M \wedge s' \in M \wedge s < s' \rightarrow \forall s''(s < s'' < s' \rightarrow s'' \in M))$.

Constraints between TS and S are expressed in $L^{TS}$ using formulas of the form $\langle z_s \rangle\, \phi$ for $\phi$ a well-formed formula from $L^S$. The satisfaction clause is (43).

(43) $\mathbf{M}_{TS}, \theta \models \langle z_s \rangle\, \phi$ iff there is an $n$ s.t. $R_{z_s}(\theta, n)$ and $\mathbf{T}_S, n \models \phi$

The constraints between the TS and the syntactic level S are of the general form (44a). In (44b-d) the special axioms are given.

(44) a. $\alpha \to \langle z_s \rangle\, \alpha'$
b. $\neg \uparrow \to \langle z_s \rangle\, (\langle z_f \rangle\, np \;\wedge \neg P\, \langle z_f \rangle\, np)$
c. $\uparrow \wedge \downarrow \to \langle z_s \rangle\, (\langle z_f \rangle\, np \;\wedge\; P\, \langle z_f \rangle\, np \wedge\; F\, \langle z_f \rangle\, np)$
d. $\uparrow \wedge \neg \downarrow \to \langle z_s \rangle\, (\langle z_f \rangle\, np \;\wedge \neg F\, \langle z_f \rangle\, np \;\wedge\; P\, \langle z_f \rangle\, np)$

According to (44a), the axioms impose constraints on the distribution of structural information. The constraints (44b) and (44d) require that the last, respective first element of a TS is related to the first, respective last element of the word order. The constraint in (44c) applies to intermediate positions of TSs with length 3. Axiom (44b) is depicted in Figure 3.

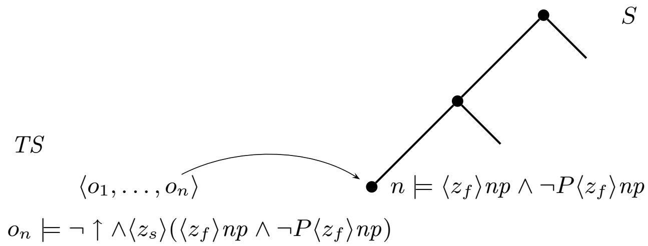


Figure 3

When taken together, the constraints in (44b-d) amount to the requirement that the default word order determined by the verb be the inverse of the order of arguments in TS. There is the following simple algorithm for calculating $\alpha'$ from $\alpha$ (and vice versa): $(\neg) \uparrow \leftrightarrow (\neg)P$ and $(\neg) \downarrow \leftrightarrow (\neg)F$. The antecedents of the constraints in (24) and the argument of $\langle z_s \rangle$ in the consequents of the constraints in (44b-d) are identical. Thus, if a constraint in (44b-d) and the corresponding in (24) are required to hold at a position of $D$, one gets (45).

(45) a. $\neg \uparrow \to \langle z_s \rangle\, (\langle z_f \rangle\, \langle CASE \rangle\, nom)$
b. $\uparrow \wedge \downarrow \to \langle z_s \rangle\, (\langle z_f \rangle\, \langle CASE \rangle\, dat)$
c. $\uparrow \wedge \neg \downarrow \to \langle z_s \rangle\, (\langle z_f \rangle\, \langle CASE \rangle\, acc)$

Linguistically, the constraints in (45) require a position in TS to be related to a particular assignment of case to an argument NP realizing this position. In Figure 4, (45a) is shown.

Similarly to the constraints in (24), they only hold for canonical verbs, i.e. for verbs which link their arguments according to a canonical pattern. In order to capture the distinction between canonical and non-canonical patterns, non-structural information at the level of TS must be taken into consideration. A linking constraint will have one of the two forms in (46).

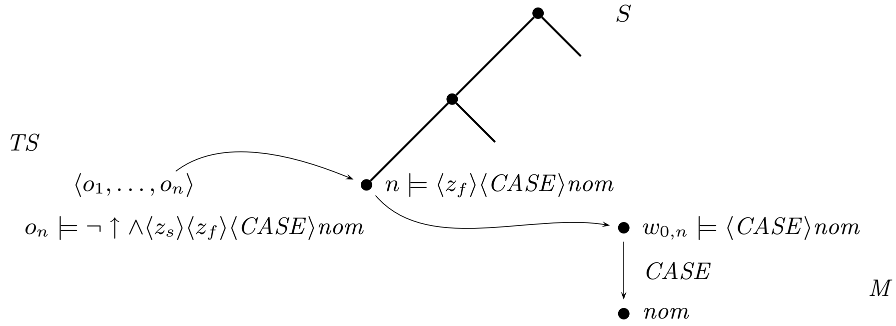


Figure 4

(46) a. $\alpha \wedge \neg\beta \rightarrow \langle z_s \rangle \langle z_f \rangle \langle CASE \rangle \phi$ canonical pattern
b. $\alpha \wedge \beta \rightarrow \langle z_s \rangle \langle z_f \rangle \langle CASE \rangle \phi'$ non-canonical pattern

In (46), $\alpha$ expresses structural and $\beta$ non-structural information (i.e. $\beta \in DR^*$). In order to arrive at constraints of the form in (14) it is not necessary to change the constraints in (44) because they hold for all verbs which realize their arguments by case.[19] Rather, the constraints in (24) must be adapted.

Since $R_{z_s}$ is a one-to-one correspondence, its inverse $R_{z_s}^{-1}$ exists. It is therefore possible to express constraints that concern the level of TS in $L^S$ by using the inverse modality $z_s^{-1}$. This yields (47) as the general scheme.

(47) a. $\psi \wedge \left\langle z_s^{-1} \right\rangle \neg\beta \rightarrow \langle z_f \rangle \langle CASE \rangle \phi$ canonical pattern
b. $\psi \wedge \left\langle z_s^{-1} \right\rangle \beta \rightarrow \langle z_f \rangle \langle CASE \rangle \phi'$ non-canonical pattern

E.g. for the constraint in (24c) one gets the two constraints in (48).

(48) a. $\langle z_f \rangle np \wedge \neg F \langle z_f \rangle np \wedge P \langle z_f \rangle np \wedge \left\langle z_s^{-1} \right\rangle \neg\phi_{dat} \rightarrow \langle z_f \rangle \langle CASE \rangle acc$
b. $\langle z_f \rangle np \wedge \neg F \langle z_f \rangle np \wedge P \langle z_f \rangle np \wedge \left\langle z_s^{-1} \right\rangle \phi_{dat} \rightarrow \langle z_f \rangle \langle CASE \rangle dat$

If these two constraints are combined with those in (46), one gets (49).

(49) a. $\uparrow \wedge \neg \downarrow \wedge \neg\phi_{dat} \rightarrow \langle z_s \rangle \langle z_f \rangle \langle CASE \rangle acc$
b. $\uparrow \wedge \neg \downarrow \wedge \phi_{dat} \rightarrow \langle z_s \rangle \langle z_f \rangle \langle CASE \rangle dat$

(49a) requires the lowest argument of a TS of length greater one to be realized by accusative case if it is not assigned the dynamic role that characterizes the lowest argument of dative verbs like 'helfen'. (49b), on the other hand, captures those cases where the lowest argument is assigned this role and requires the argument to be realized by dative case. (49b) is depicted in Figure 5.

¿From the way the constraint between the two levels is defined it follows that the flow of information between them is bidirectional. It is not only possible to zoom in from TS into S (by $z_s$) but also to zoom out from S into TS. This possibility is due to the way positions of a TS are related to elements of the (default) word order at S. The mapping rule in (8), which requires the order of elements at the level of TS

[19] If verbs with prepositional arguments are taken into account, the constraints in (44) must be changed too. The antecedents must contain non-structural information.

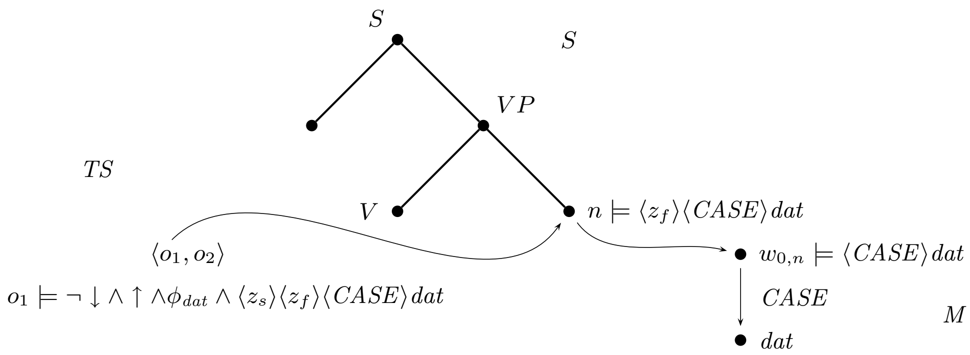


Figure 5

to be the inverse of that at the level of S, can be read in the other direction as well: particular positions of TS correspond to particular positions of the word order at S and vice versa. In the terminology of Finger and Gabbay (1992) this means that $L^{TS}$ and $L^S$ are *fully fibered*.

### 2.6.4 The Definition of the Cases

The analysis of the constraints between TS, S and M given in the preceding sections assumed that information about cases is atomic information at the level of M. This is in contrast to LDG where the cases are defined using the same properties that are used to characterize the TS. Given a position of the TS, the case that is realized at the morphological level is determined by a matching operation between this position and the set of cases. In the preceding sections, an LDG-style definition of the cases was not needed because the constraints were based on the result of the matching operation whereas the matching operation was not accounted for. In this section the cases will be defined in the present framework.

Recall from Section 1 that linking rules are of the form $\alpha \wedge \beta \rightarrow \gamma$, where $\alpha$ expresses structural information, $\beta$ non-structural information and $\gamma$ specifies a case. A $\gamma$ corresponding to a case $C$ can be implied by more than one conjunction $\alpha \wedge \beta$. Let $\bigvee_{1 \leq i \leq n}(\alpha_i \wedge \beta_i)$ be the disjunction of these conjunctions, for $n$ the number of conjuncts that imply $\gamma$. Since in LDG the linguistic function of a case $C$ is exhausted by the possible positions relative to a TS at which it can occur, $C$ can be defined as the disjunction: $C = \bigvee_{1 \leq i \leq n}(\alpha_i \wedge \beta_i)$. If besides canonical patterns also the non-canonical pattern of dative verbs is considered, one gets the relationship between cases, elements of a TS and dynamic roles in Table 7 (AP = argument position).

| Case | positions of TS | dynamic roles |
|---|---|---|
| NOM | highest AP | none |
| DAT | intermediate AP (length of TS > 2) | none |
| | lowest AP (length TS > 1) | $\phi_{dat}$ |
| ACC | lowest AP (length TS > 1) | $\neg\phi_{dat}$ |

Table 7

The rows are to be understood as follows: the argument corresponding to the $i$-th position of a TS (of length $n$) is morphologically realized by case $C$ if they are related to the dynamic roles $R_1, ..., R_n$. E.g. the argument corresponding to the lowest position of a TS (of length greater 1) is morphologically realized (i) by $ACC$ if the dynamic role $R_{dat}$ is not related to that position, and (ii) by $DAT$ if the dynamic role $R_{dat}$ is related to that position.

A sufficient condition for an argument to be realized by a case is given by information both about the position and the presence (or absence) of information about dynamic roles to which that position is related. Canonical linking is characterized by requiring the absence of information about dynamic roles whereas non-canonical linking requires information about a particular dynamic role. Disjuncts which require the absence of information about dynamic roles are structural, otherwise they are non-structural. For structural disjuncts, the structural component is defined using the definitions from (11), i.e. the specificity condition is *not* applied. For non-structural disjuncts, on the other hand, the SC is applied so that the structural component is maximally specified. This has the effect that the structural disjuncts in the definition of a case are less specific than the non-structural ones in the definition of other cases with which the former are compatible. If the SC is applied to the structural component of the structural disjunct in the definition of a case, the structural disjuncts are incompatible with the non-structural ones. The definitions of the cases are given in (50) ($\langle \alpha, \alpha' \rangle$ abbreviates $\alpha \wedge \alpha'$).

(50) a. $NOM =_{def.} \langle \bot_\uparrow, \bot_\downarrow \rangle$
b. $DAT =_{def.} \langle \uparrow, \downarrow \rangle \vee (\langle \neg \uparrow, \downarrow \rangle \wedge \ \phi_{dat})$
c. $ACC =_{def.} \langle \uparrow, \bot_\downarrow \rangle \wedge \ \neg\phi_{dat}$

The linking mechanism is illustrated for a dative verb like 'helfen'.

| TS | $\theta_1$ | $\theta_2$ |
|---|---|---|
| $\alpha$ | $\langle \uparrow, \neg \downarrow \rangle$ | $\langle \neg \uparrow, \downarrow \rangle$ |
| $\beta$ | $\phi_{dat} \wedge \phi_{last}$ | $\phi_{\text{first}}$ |

Table 8

$ACC$ is excluded at $\theta_1$ because the first disjunct requires $\neg\phi_{dat}$, which is not compatible with $\phi_{dat}$ from the non-structural component $\beta$ of $\theta_1$. The disjunct $\langle \uparrow, \neg \downarrow \rangle \wedge \phi_{dat}$ of the definition of $DAT$ is compatible with $\alpha_1 \wedge \beta_1$. Since $\langle \uparrow, \neg \downarrow \rangle$ is more specific than $\langle \bot_\uparrow, \bot_\downarrow \rangle$ (= $NOM$), dative case is assigned to $\theta_1$. For $\theta_2$, only $NOM$ is compatible with the structural specification. Consequently, the argument NP realizing this position is linked by nominative case.

| | |
|---|---|
| $\theta_1$ | $\langle \uparrow, \neg \downarrow \rangle \wedge \ \phi_{dat}$ |
| $NOM$ | $\langle \bot_\uparrow, \bot_\downarrow \rangle$ |
| $DAT$ | $\langle \uparrow, \neg \downarrow \rangle \wedge \phi_{dat}$ |
| $ACC$ | not compatible |

Table 9

## 2.7 The SF-Level

We will start by defining the structures which provide a semantics for the language $L^{SF}$, that is defined in the following section.

### 2.7.1 The Structures: Dynamic Eventuality Structures

Structures for the SF-level are *dynamic eventuality structures*.

**Definition 2.6** A dynamic eventuality structure **DES** is a tuple $\langle \mathbf{E}, \mathbf{S}, \mathbf{O}, \{R_{dr} \mid dr \in DR\}, \{R_{\text{prop}} \mid \text{prop} \in \text{PROP}\}, \alpha, \omega, \tau, \mu \rangle$ such that[20]

- $\mathbf{E} = \langle E, \{P_v \mid v \in \text{VERB}\}, \sqsubseteq_E, C \rangle$ is an eventuality structure with
  - $E$ is (non-empty) set of events
  - $P_v \subseteq E$ is the set of all events of type $v$
  - $\sqsubseteq_E$ is the material part-of relation on $E$, which is a partial order
  - $C = \{(e, e_1, e_2) \mid e_1 \sqsubseteq_E e \wedge e_2 \sqsubseteq_E e \wedge \omega(e_1) = \alpha(e_2) \wedge \alpha(e) = \alpha(e_1) \wedge \omega(e) = \omega(e_2)\}$ is the composition relation, that is required to be associative
- $\mathbf{S} = \langle S, <_S \rangle$ is a transition structure with $S$ a (non-empty) set of states (or time points), that is linearly ordered by $<_S$
- $\mathbf{O} = \langle O, \sqsubseteq_O \rangle$ is an object structure with $O$ a (non-empty) set of objects, that is partially ordered by $\sqsubseteq_O$, the material part-of relation on $O$
- each $R_{dr}$ is a (functional) relation on $E \times O$, which is a dynamic role
- each $R_{\text{prop}}$ is a relation on $O^n \times S$; intuitively, an $n+1$-tuple $\langle d_1, ..., d_n, s \rangle$ is an element of $R_{\text{prop}}$ just in case the property expressed by prop holds between the $n$-tuple $\langle d_1, ..., d_n \rangle$ at $s$
- $\alpha : E \rightarrow S$ and $\omega : E \rightarrow S$ assign to each event $e \in E$ its beginning and end point, respectively
- $\tau(e) = \{s \in S \mid \alpha(e) \leq_S s \leq_S \omega(e)\}$ is the execution sequence of the event $e$
- $\mu$ assigns to each element of $\{P_v \mid v \in \text{VERB}\}$ a set of dynamic roles which is linearly ordered by $<$

### 2.7.2 The Language $L^{SF}$

A suitable language for the SF level is *hybrid Dynamic Modal Arrow Logic* (DMAL$^h$), Naumann and Osswald (1999, 2001). DMAL$^h$ combines Dynamic Modal Logic and Arrow Logic, van Benthem (1991), de Rijke (1993). It is a three-sorted language. For each of the three basic domains of dynamic eventuality structures there is a sort of formula which can be used to talk about elements of the domain. Besides $e$-formulas, that are evaluated with respect to elements of $E$, there are both $s$-formulas and $d$-formulas, that denote sets of states and objects, respectively. The translation of a verb in the lexicon is a complex $e$-formula. This formula must account for three different aspects of the dynamic-temporal structures of events that are linguistically relevant. First, events bring about results. As was shown above in Section 2.5.1, an event usually brings about several results that are of different types. Theoretically, it is

[20] A **DES** for a verb $v$ is based on the set $E$ of all events, and not only on $P_v$, because an event belonging to $P_v$ can have subevents that do not belong to $P_v$.

sufficient to require that the logically strongest of the latest results be brought about since then all other results are brought about too. This follows from the way the results determined for an event are temporally related to each other. Bringing about the logically strongest of the latest results determined for an event requires bringing about all results that strictly precede this result relative to $\preceq$. In addition, the logically weaker of the latest results are brought about too because they are implied by the logically strongest one. The principle that is used in DES is taken over from LDG: the translation of a verb is subject to the Principle of Minimal Decomposition. In DES, this yields (51).

(51) Principle of Minimal Decomposition

In the translation of a verb the logically strongest of the latest results is represented; and the logically weakest of the earliest results is represented if the set of participants to which it is assigned is not identical to that to which the logically strongest of the latest results is assigned.

For a Transfer verb like 'give', the Principle of Minimal Decomposition yields a decomposition that is similar to the one used in LDG in terms of the properties Act and Poss. In DES, $\otimes_{\text{Act}}$ and $\otimes_{\text{Poss}}$ are unary and binary modal operators, respectively. They are interpreted relative to the relations $R_{\text{Act}}$ on $O \times S$ and $R_{\text{Poss}}$ on $O^2 \times S$. For two objects $d$, $d' \in O$, $R_{\text{Act}}(d)$ and $R_{\text{Poss}}(d, d')$ are sets of states and therefore results that can be brought about by events.[21] These results are evaluated on the execution sequences of events of type giving which bring them about in a specific way that is determined by their type. These ways results are evaluated on execution sequences are captured by variants of dynamic modes from Dynamic Modal Logic. In DES a dynamic mode is interpreted as a relation on $\wp(E) \times \wp(S) \times E$, i.e. it maps a set of events and a set of states to a set of events. Since the definition of these modes in DMAL$^h$ is irrelevant for modelling the flow of information in grammar, they are skipped. The interested reader is referred to Naumann and Osswald (1999, 2001).

The second aspect that has to be represented in the translation of verbs is the relationship between events and objects participating in them. This is done in terms of modal operators that syntactically map $d$-formulas to $e$-formulas and that are interpreted relative to the basic dynamic roles defined in Section 2.5.1. The third aspect concerns the relationship between results and participants of events: each result brought about by an event is brought about with respect to at least one object participating in the event. This aspect is captured in terms of the relations on $O^n \times S$ like $R_{Act}$ or $R_{\text{Poss}}$. It is at this point that the need for a hybrid language arises. The arguments of the modal operators corresponding to these relations must be interpreted relative to the same objects that bear particular dynamic roles to the event at which the translation of a verb is evaluated. This cannot be expressed in ordinary modal logics because it requires a mechanism that makes cross-reference possible. A suitable mechanism to achieve such cross-reference are variables. Modal logics with variables are called *hybrid modal logics*.

Hybrid languages share characteristics of both modal and first-order languages (Blackburn and Seligman 1995). With the former they adopt the internal perspective

[21] For $R_{\text{Act}}$, this is a simplification because the corresponding result can only be defined relative to the beginning point of an event that brings it about. Consequently, $R_{\text{Act}}$ should be defined as a relation on $S \times O \times S$. This aspect has been suppressed for the sake of simplicity. See Naumann and Osswald (2001) for details.

whereas they resemble the latter by making explicit use of variables and binding. Syntactically, (basic) hybrid languages result from modal languages by augmenting the latter with a set $X$ of variables. Each element of $X$ functions as a new syntactic atom. The interpretation of modal formulas is relativized to an assignment of values to variables $g$, i.e. $g : X \to D$ assigns to each element from $X$ an element $g(x)$ from the basic domain $D$. The satisfaction relation is defined between a model $\mathbf{M}$, an assignment function $g$, an element from $D$ and a formula from the hybrid language. The base clause for variables is (52).

(52) $\mathbf{M}, g, d \models x$ iff $g(x) = d$

On top of a basic hybrid language further hybrid languages can be built by adding various binding operators and, possibly, further modal operators. A *binding operator* is a binary operator $B$ that takes a variable $x$ and a formula $\phi$ as arguments and returns a formula $Bx.\phi$. $B$ binds all free occurrences of $x$ in $\phi$. The choice of binders depends on the application. In the present context the binder $\exists$ with satisfaction clause (53) will be used.

(53) $\mathbf{M}, g, d \models \exists x.\phi$ iff there is an assignment function $g' \stackrel{x}{=} g$ and $\mathbf{M}, g', d \models \phi$

The binder $\exists$ is a close analogue to the existential quantifier from first-order logic. It non-deterministically changes the assignment function, looking for a satisfying assignment to the variable bound by it. The variables are sorted. There are event variables, which take their values in $E$, and there are object variables, which take their values in $O$. Event variables are referred to by $\varepsilon$ and object variables by $x$, $y$, $z$ (possibly primed).

**Definition 2.7** $e$-formulas, $d$-formulas, $s$-formulas[22]

- $e$-formulas: (i) each event variable $\varepsilon$ is an $e$-formula; (ii) $\sqsubseteq$ and $\sqsubseteq_v$ with $v \in$ VERB are $e$-formulas; (iii) if $\pi$ and $\pi'$ are $e$-formulas, $\pi \bullet \pi'$, $\pi \cap \pi'$ and $\sim \pi$ are $e$-formulas; (iv) if $\phi$ is a $d$-formula, $\Diamond_{dr}\phi$ is an $e$-formula for $dr \in DR$; (v) if $\phi$ is an $s$-formula, $L\phi$ and $R\phi$ are $e$-formulas; (vi) if $\pi$ is an $e$-formula and $x$ an object variable $\exists x.\pi$ is an $e$-formula; (vii) if $\psi$ is an $L^{TS}$ formula, $\langle \Diamond_v \rangle \psi$ is an $e$-formula
- $d$-formulas: (i) each object variable $x$ is a $d$-formula; (ii) if $\phi$ and $\psi$ are $d$-formulas, $\phi \wedge \psi$ and $\neg\phi$ are $d$-formulas
- $s$-formulas: (i) if $\phi_i$, $1 \leq i \leq n$, are $d$-formulas, $\otimes_{\text{prop}}(\phi_1, ..., \phi_n)$ is an $s$-formula for prop $\in$ PROP; (ii) if $\phi$ and $\psi$ are $s$-formulas, $\phi \wedge \psi$ and $\neg\phi$ are $s$-formulas

The satisfaction clauses are given in (54).

(54) a. $\mathbf{DES}, g, e \models \varepsilon$ iff $g(\varepsilon) = e$
b. $\mathbf{DES}, g, e \models \pi \bullet \pi'$ iff there are $e_1$ and $e_2$ s.t. $C(e, e_1, e_2)$, $\mathbf{DES}, g, e_1 \models \pi$ and $\mathbf{DES}, g, e_2 \models \pi'$
c. $\mathbf{DES}, g, e \models \pi \cap \pi'$ iff $\mathbf{DES}, g, e \models \pi$ and $\mathbf{DES}, g, e \models \pi'$
d. $\mathbf{DES}, g, e \models \sim \pi$ iff $\mathbf{DES}, g, e \not\models \pi$
e. $\mathbf{DES}, g, e \models L\phi$ iff $\mathbf{DES}, g, \alpha(e) \models \phi$

[22] According to the definition below, non-atomic $d$- and $s$-formulas are referred to by $\phi$ or $\psi$ (possibly primed). From this no confusion will arise since in the translation of verbs the only $d$-formulas used are object variables. Therefore, $\phi$ and $\psi$ are always $s$-formulas.

f. $\mathbf{DES}, g, e \models R\phi$ iff $\mathbf{DES}, g, \omega(e) \models \phi$
g. $\mathbf{DES}, g, e \models \exists x.\pi$ iff there is a $g'$ s.t. $g' \stackrel{x}{=} g$ and $\mathbf{DES}, g', e \models \pi$
h. $\mathbf{DES}, g, e \models \Diamond_{dr}\phi$ iff $\mathbf{DES}, g, R_{dr}(e) \models \phi$
i. $\mathbf{DES}, g, e \models \sqsubseteq$ iff $e \in E$
j. $\mathbf{DES}, g, e \models \sqsubseteq_v$ iff $e \in P_v$
k. $\mathbf{DES}, g, d \models x$ iff $g(x) = d$
l. $\mathbf{DES}, g, d \models \phi \wedge \psi$ iff $\mathbf{DES}, g, d \models \phi$ and $\mathbf{DES}, g, d \models \psi$
m. $\mathbf{DES}, g, d \models \neg\phi$ iff $\mathbf{DES}, g, d \not\models \phi$
n. $\mathbf{DES}, g, d \models \top$ for all $d \in O$
o. $\mathbf{DES}, g, s \models \otimes_{\text{prop}}(\phi_1, ..., \phi_n)$ iff there are $d_1, ..., d_n$ s.t. $R_{\text{prop}}(d_1, ..., d_n, s)$ and $\mathbf{DES}, g, d_i \models \phi_i$
p. $\mathbf{DES}, g, s \models \neg\phi$ iff $\mathbf{DES}, g, s \not\models \phi$
q. $\mathbf{DES}, g, s \models \phi \wedge \psi$ iff $\mathbf{DES}, g, s \models \phi$ and $\mathbf{DES}, g, s \models \psi$

Each type of formula has a particular function which is related to an aspect of the dynamic-temporal structure of events. The $e$-formulas $\sqsubseteq$ and $\sqsubseteq_v$ express sortal information about events. The relationship between events and results is expressed by $e$-formulas of the form $DM(\sqsubseteq_v, \phi)$, which are abbreviations of complex $e$-formulas. Basically, they are defined in terms of $\bullet$, $L$ and $R$. The latter two are used to impose conditions on the beginning and the end point of an event, respectively. If a formula $L\phi$ or $R\phi$ is evaluated at an event, the evaluation switches from the level of events to that of states. This means that, in effect, the language of $e$-formulas is layered across that of $s$-formulas. The modality $\bullet$ is used to decompose an event so that it is possible to constrain the information that holds at the end points of proper initial stages, again using $R$. The relationship between events and objects is expressed using the modalities $\Diamond_{dr}$ with $dr \in DR$. Note that at the level of semantic form only the basic dynamic roles are used since they are sufficient to uniquely pick out the objects participating in an event that undergo a change relative to an event type. Similarly to formulas of the form $\bigcirc\phi$ with $\bigcirc \in \{L, R\}$, the evaluation of a formula $\Diamond_{dr}\phi$ involves a switch of level. In this case from the level of events to that of objects.

¿From what has been said it follows that the $\Diamond_{dr}$ and the modalities which take an $s$-formula and return an $e$-formula are used to layer one language over another. In effect, DMAL$^h$ can be described as $L^e(L^s, L^d)$, the language of $e$-formulas layered across the languages of $s$-formulas and $d$-formulas. On this perspective, $e$-formulas are used to express properties that concern the type of events as well as their internal structure at the level of the material part-of relation $\sqsubseteq_E$ and the composition relation $C$. The other two types of formulas are used to constrain the linguistically relevant aspects of the dynamic-temporal structure of events. They impose conditions on the subevents that are determined at the level of pure $e$-formulas, i.e. $e$-formulas that are not of the form $\Diamond_{dr}\phi$ or $\bigcirc\phi$, $\bigcirc \in \{L, R\}$.

### 2.7.3 The Translation of Verbs in $L^{DMAL^h}$

The translation of a verb in the lexicon has one of the two forms given in (55).

(55) a. $v \rightsquigarrow \exists x_1 ... \exists x_n (\sqsubseteq_v \cap \Diamond_{dr_1} x_1 \cap ... \cap \Diamond_{dr_n} x_n \cap DM_1(\sqsubseteq, \phi_1) \cap DM_2(\sqsubseteq_v, \phi_2))$
b. $v \rightsquigarrow \exists x_1 ... \exists x_n (\sqsubseteq_v \cap \Diamond_{dr_1} x_1 \cap ... \cap \Diamond_{dr_n} x_n \cap DM(\sqsubseteq_v, \phi))$

(55a) is used if besides the logically strongest of the latest results also the logically weakest of the earliest results is represented in accordance with the Principle of Minimal Decomposition. Otherwise, (55b) is used. The first conjunct $\sqsubseteq_v$ requires an event that satisfies the $e$-formula to be an element of the basic event type corresponding to the verb the formula is the translation of. It therefore expresses sortal information about events. The conjunction $\Diamond_{dr_1} x_1 \cap ... \cap \Diamond_{dr_n} x_n$ expresses the relationship between an event and its participants. The index $n$ is identical to the length of $\mu(P_v)$, i.e. if a basic dynamic role is defined for the verb, it is represented in the translation. The conjuncts $DM(\sqsubseteq, \phi)$ express both the dynamic aspect that events bring about results as well as the relationship between results and participants of events. $DM$ is a dynamic mode. It requires its second argument to be evaluated on the execution sequence of an event satisfying the translation in a particular way. The first argument is used to (possibly) restrict the relevant points of the sequence at which $\phi$ has to be true. Each $s$-formula $\phi$ in the context $DM(\sqsubseteq, \phi)$ is of the form $\otimes_{\mathrm{prop}}(x_1, ..., x_j)$. Each $x_i$, $1 \leq i \leq j$, occurs exactly once as an argument of an $e$-formula $\Diamond_{dr} x$. It is here that the cross-reference mechanism applies. Since $x_i$ is bound by $\exists$ at the outset of the translation, each of its occurrences is interpreted relative to the same object from $O$. This object bears the dynamic role $R_{dr}$ to an event satisfying the translation. As a consequence, $\otimes_{\mathrm{prop}}(x_1, ..., x_j)$ is interpreted relative to an $n$-tuple of objects each element of which participates in the event, as required. If one were only interested in the relationship between an event and its participants and not, in addition, in the relationship between results and participants, one could use $\Diamond_{dr_1} \top \cap ... \cap \Diamond_{dr_n} \top$ instead of $\Diamond_{dr_1} x_1 \cap ... \cap \Diamond_{dr_n} x_n$. This is possible because a dynamic role is a functional relation so that for a given event a unique object is picked out (provided the dynamic role is defined for the former). For $s$-formulas of the form $\otimes_{\mathrm{prop}}(x_1, ..., x_j)$, on the other hand, there is in general more than one $n$-tuple $\langle d_1, ..., d_n \rangle$ of objects for a given state $s$ such that $\langle d_1, ..., d_n, s \rangle$ is an element of $R_{\mathrm{prop}}$. In (56), the translation of Transfer verbs like 'geben' and of the verb 'essen' are given.

(56) a. geben $\rightsquigarrow$ $\exists x \exists y \exists z (\sqsubseteq_{geben} \cap \Diamond_{first} x \cap \Diamond_{int} y \cap \Diamond_{last} z \cap Con\text{-}BEC_S(\sqsubseteq, \otimes_{\mathrm{Act}}(x)) \cap Min\text{-}BEC_S(\sqsubseteq_{geben}, \otimes_{\mathrm{Poss}}(y, z)))$

b. essen $\rightsquigarrow$ $\exists x \exists y \exists z (\sqsubseteq_{essen} \cap \Diamond_{first} x \cap \Diamond_{last} z \cap Con\text{-}BEC_S(\sqsubseteq, \otimes_{\mathrm{Act}}(x)) \cap Min\text{-}BEC_S(\sqsubseteq_{essen}, \otimes_{\mathrm{Be-in}}(y, x)))$

The dynamic mode $Con\text{-}BEC_S$ corresponds to s*-minimal results. It requires its second argument to be true at each point of the execution sequence of an event satisfying $Con\text{-}BEC_S(\sqsubseteq, \phi)$, except at its beginning point. The mode $Min\text{-}BEC_S$ is the dynamic mode which captures the way s- and w-maximal results are evaluated. The second argument is true only at the end point of an event satisfying $Min\text{-}BEC_S(\sqsubseteq_v, \phi)$ (for details, the reader is referred to Naumann and Osswald 1999).

### 2.7.4 The Relationship between SF and TS

Recall that in LDG the relationship between the levels of SF and TS is defined by the Hierarchy Principle: the order of arguments at the theta structure of a verb is the inverse of the ordering at the level of SF. Since the relationship between events and objects participating in events is defined in terms of the three basic dynamic roles, it follows that the notion of argument is explicated in terms of the notion of a basic dynamic role at the SF level. Syntactically, this relation is expressed in terms of the modalities $\Diamond_{dr}$, with $dr \in DR$, which map $d$-formulas to $e$-formulas. The relationship between the two structures must therefore be defined in terms of the domain $E$ of events and the union of the argument structures of the verbs under consideration, which is $\underline{D} = \bigcup_{v \in \mathrm{VERB}} D_v$. For $v \neq v'$, one has $D_v \cap D_{v'} = \emptyset$. Formally, this is done by a family $\{R_{\Diamond_v}\}_{v \in \text{ VERB}}$ of relations. Each $R_{\Diamond_v}$ is a relation on $E \times \underline{D}$. A pair $\langle e, \theta \rangle$ is an element of $R_{\Diamond_v}$ just in case $e \in P_v$ and $\theta$ is an element of the theta structure of the verb $v$, i.e. $\theta \in D_v$.[23] The relationship between the two levels can be expressed in $L^{DMAL^h}$ by using the modalities $\Diamond_v$, which are interpreted by the $R_{\Diamond_v}$, (57).[24]

(57) $\mathbf{DES}, g, e \models \langle \Diamond_v \rangle\, \phi$ iff there is an $\theta \in D$ s.t. $R_{\Diamond_v}(e, \theta)$ and $\underline{\mathbf{M}}_{TS}, \theta \models \phi$

The constraints are of the form (58).

(58) $\sqsubseteq_v \wedge \Diamond_{dr} \top \;\rightarrow \langle \Diamond_v \rangle\, (\psi \wedge \phi_{dr})$

The constraint requires that if an event belonging to the event type corresponding to the verb $v$ satisfies $\Diamond_{dr}\top$, then there is an element of the theta structure of $v$ that satisfies $\phi_{dr}$. According to the HP, the argument is uniquely determined by the place of $R_{dr}$ in $\mu(P_v)$. Thus, the argument can be determined in terms of structural information. In (58), this is done by the $L^{TS}$ formula $\psi$, which expresses structural information about the theta structure. If $\mu(P_v)$ contains at least two elements, one gets the axioms in (59). For intransitive verbs, the axiom in (60) applies.[25]

(59) transitive and ditransitive verbs
a. $\sqsubseteq_v \wedge \Diamond_{first} \top \;\rightarrow \langle \Diamond_v \rangle\, (\neg \uparrow \wedge \downarrow \wedge\, \phi_{first})$
b. $\sqsubseteq_v \wedge \Diamond_{last} \top \;\rightarrow \langle \Diamond_v \rangle\, (\uparrow \wedge \neg \downarrow \wedge\, \phi_{last})$
c. $\sqsubseteq_v \wedge \Diamond_{int} \top \;\rightarrow \langle \Diamond_v \rangle\, (\uparrow \wedge \downarrow \wedge\, \phi_{int})$

(60) intransitive verb
$\sqsubseteq_v \wedge \Diamond_{first} \top \;\rightarrow \langle \Diamond_v \rangle\, (\neg \uparrow \wedge \neg \downarrow \wedge\, \phi_{first})$

¿From (59) and (60) the axioms in (61) about the theta structure can be derived.

(61) a. $\neg \uparrow \rightarrow \phi_{first}$
b. $\uparrow \wedge \downarrow \rightarrow \phi_{int}$
c. $\uparrow \wedge \neg \downarrow \rightarrow \phi_{last}$

Figure 6 illustrates the relationship between the semantic form and the theta structure for the ditransitive verb 'geben'.

The constraint between SF and TS can also be formulated in the other direction. If an element of a theta structure of the verb satisfies $\psi \wedge \phi_{dr}$, then all events that satisfy $\sqsubseteq_v$, also satisfy $\Diamond_{dr}\top$. Thus, it is possible to fully fiber the languages $L^{SF}$ ($= L^{DMAL^h}$) and $L^{TS}$ across each other. When taken together with the fact that $L^{TS}$ and $L^{S}$ are also fully fibered across each other, it follows that one can zoom from SF into S (via TS) and zoom out from S into SF (again via TS) so that there

[23] The dependence on a verb expressed by the index $v$ is necessary because an event can belong to two event types that have different theta structures.

[24] $\underline{\mathbf{M}}_{TS}$ is the structure of the theta structures of all verbs. For $\theta \in D_v$, one has $\underline{\mathbf{M}}_{TS}, \theta \models \phi$ iff $\mathbf{M}_{TS}, \theta \models \phi$ with $\mathbf{M}_{TS}$ the theta structure of the verb $v$.

[25] Here it is assumed that the dynamic role $R_{first}$ is used.

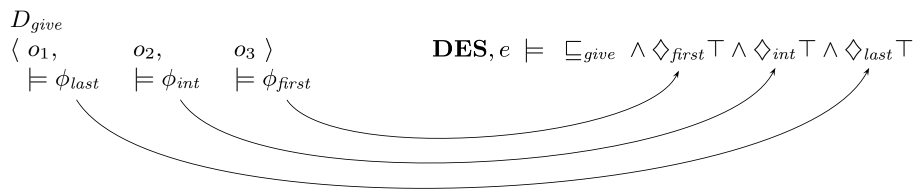


Figure 6

is a bidirectional flow of information from SF to S. From a semantic point of view the direction from SF to S is more important because the decompositional analysis of verbs provides an explanation for the constraints at the other levels.

## 3 Conclusion

On the LDG perspective of grammar, there is a ‘flow of information’ from the level of semantic form to the morphological level that is mediated by the theta structure, which is part of the semantic level, and the syntactic level. This is summarized in the table below.

| | relationship | constraint |
|---|---|---|
| SF - TS | order of arguments in SF | Hierarchy Principle (7) |
| | order of arguments in TS | |
| TS - S | order of arguments in TS | Mapping rules in (8) |
| | default word order | |
| S - M | default word order | canonical patterns in (4) and (5) |
| | assignment of case to NPs | non-canonical patterns |

Table 10

In this article this flow has been explained in terms of the dynamic-temporal structure of events. Events bring about results which are brought about relative to objects that participate in the events. Objects are related to events by dynamic roles which are functional relations on $E \times O$. They are defined in terms of sets of results. Results are brought about in a temporal order. This order is reflected in the way a linear ordering on the dynamic roles is defined.

The theta structure is defined as a linearly ordered set of abstract points, which, following linguistic usage, are called arguments. The relationship to the SF level is established by requiring that at these points particular, non-structural information be true that expresses the link between arguments and particular dynamic roles: this information is satisfied at an argument position just in case it is related to the dynamic role.

At the syntactic level, the order is reversed. This has the effect that the order in which arguments are realized in syntax is identical to the order on the sets of results in terms of which the dynamic role corresponding to the argument is defined. From this it follows that the object participating in the event that is assigned the minimal set of results is denoted by the first argument of the syntactic order. Since this argument is assigned nominative case if the verb is canonical and since this object is involved first in the event, the object that is involved first is normally denoted by the argument which is linked to the verb by nominative case. In the figure below the relationship between the levels of SF, TS and M for a three-place verb with canonical linking pattern (say, 'geben') is depicted.

| TS | | | SF | | |
|---|---|---|---|---|---|
| $\theta_1$ | $\theta_2$ | $\theta_3$ | $e$ | $e$ | $e$ |
| $\models$ | $\models$ | $\models$ | $\models$ | $\models$ | $\models$ |
| $\langle\uparrow,\neg\downarrow\rangle \wedge \phi_{last}$ | $\langle\uparrow,\downarrow\rangle \wedge \phi_{int}$ | $\langle\neg\uparrow,\downarrow\rangle \wedge \phi_{first}$ | $\Diamond_{first}\top$ | $\Diamond_{int}\top$ | $\Diamond_{last}\top$ |
| $\Updownarrow$ | $\Updownarrow$ | $\Updownarrow$ | | | |
| ACC | DAT | NOM | | | |
| M | | | | | |

At the syntactic level the order is reversed.

| | | | | |
|---|---|---|---|---|
| daß | der Lehrer | dem Schüler | ein Buch | gab |
| | NOM | DAT | ACC | |

Each grammatical level is formally represented by a mathematical structure together with a language to talk about elements of this structure. Constraints between different levels are formulated by layering the languages. If level A constrains level B, the constraints are expressed in $L^A$ by using formulas of the form $\bigcirc\phi$ with $\bigcirc$ a modality and $\phi$ a formula from $L^B$. A constraint is of the form $\psi \rightarrow \bigcirc\phi$: if an element from level A satisfies the property expressed by $\psi$, a corresponding element from level B satisfies the property expressed by $\phi$.

## References

[1] van Benthem, J. (1991). *Language in Action*, Amsterdam, North-Holland

[2] Blackburn, P. (1993). Nominal Tense Logic, *Notre Dame Journal of Formal Logic*, 14, 56-83

[3] Blackburn, P. et al. (1993). Talking about Trees, In: *Proceedings 6th Conference of the European Chapter of the ACL*, 21-29

[4] Blackburn, P./Meyer-Viol, W. (1997). Modal Logic and Model-Theoretic Syntax, In: M. de Rijke (ed.). *Advances in Intensional Logic*, Kluwer Academic Publishers, 29-60

[5] Blackburn, P./de Rijke, M. (1997). Zooming In, Zooming Out, *JoLLI*, 6, 5-31

[6] Blackburn, P./Seligman, J. (1995). Hybrid Languages, *JoLLI*, 4, 251-272

[7] Dowty, D. (1979). *Word Meaning and Montague Grammar*, Dordrecht: Reidel

[8] Dowty, D. (1991). Thematic proto-roles and argument selection, *Language*, 67

[9] Finger, M./Gabbay, D. (1992). Adding a temporal Dimension to a Logic System, *JoLLI*, 1, 203-233

[10] Kiparsky, P. (1992). Structural Case, Ms., Institute for Advanced Study, Berlin

[11] Krifka, M. (1992). Nominal Reference, Temporal Constitution and Thematic Relations, In: Sag, I. et al. (eds.). *Lexical Matters*, Stanford, CSLI Publications, 29-53

[12] Naumann, R. (1998). A Dynamic Logic of Events and States for the Analysis of the Interaction between Verb Aspect and Plural Quantification in Natural Language, In: *Proceedings 5th WoLLIC*, Sao Paulo, Brazil, extended version appeared in *Journal of the IGPL*, 7, 591-627

[13] Naumann, R. (1999). Aspects of Changes: A Dynamic Event Semantics, *Journal of Semantics*, 18, 27-81

[14] Naumann, R. (2001). Linking as non-monotonic reasoning, ms. University of Düsseldorf; this is a revised version of an article that appeared in: de Queiroz, R. (ed.). *Proceedings 8th WoLLIC*

[15] Naumann, R./Latrouite A. (2000). Linking and the Dynamic Structure of Verbs: Underspecification at the morphological-semantic Interface, In: Obrst, L. (ed.). *Proceedings of the Workshop on Semantic Approximation, Vagueness and Granularity*, AAAI Publications

[16] Naumann, R./Osswald, R. (1999). A Dynamic Modal Arrow Logic for the Analysis of Aspectual Phenomena in Natural Language, In: *Proceedings 3rd International Tbilisi Symposium on Logic, Language and Computation*

[17] Naumann, R./Osswald, R. (2001). A Dynamic Modal Arrow Logic for the Analysis of Aspectual Phenomena in Natural Language, ms. University of Düsseldorf; extended and revised version of Naumann and Osswald (1999)

[18] de Rijke, M. (1993). *Extending Modal Logic*, PhD. dissertation, University of Amsterdam

[19] Wunderlich, D. (1997). CAUSE and the Structure of Verbs, *Linguistic Inquiry*, 28, 27-68

[20] Wunderlich, D. (1999). Predicate Composition and Argument Extension as general Options, In: Stiebels, B. et al. (eds.). *Proceedings of the Conference 'Lexicon in Focus',* Berlin, Akademie Verlag

[21] Wunderlich, D. (2001). Optimal Case, ms. University of Düsseldorf

Received 2 October 2000. Revised 5 March 2002.

# Acknowledgements

The Editor-in-Chief would like to thank the following colleagues who have helped maintain the standards set for a scientific journal, through their refereeing of the papers that have been submitted.[1]

Torsten Schaub
Anuj Dawar
Erich Graedel
Carsten Lutz
Rainer Osswald
Neil Tennant

[1] The list includes the referees for the papers in this issue, plus the referees of papers rejected meanwhile.

# Interest Group in Pure and Applied Logics (IGPL)

The Interest Group in Pure and Applied Logics (IGPL) is sponsored by The European Association for Logic, Language and Information (FoLLI), and currently has a membership of over a thousand researchers in various aspects of logic (symbolic, mathematical, computational, philosophical, etc.) from all over the world (currently, more than 50 countries). Our main activity is that of a research and information clearing house.

Our activities include:

- Exchanging information about research problems, references and common interest among group members, and among different communities in pure and applied logic.
- Helping to obtain photocopies of papers to colleagues (under the appropriate copyright restrictions), especially where there may be difficulties of access.
- Supplying review copies of books through the journals on which some of us are editors.
- Helping to organise exchange visits and workshops among members.
- Advising on papers for publication.
- Editing and distributing a Newsletter and a Journal (the first scientific journal on logic which is FULLY electronic: submission, refereeing, revising, typesetting, publishing, distribution; first issue: July 1993): the Logic Journal of the Interest Group on Pure and Applied Logics. (For more information on the Logic Journal of the IGPL, see the Web homepage: http://www.jigpal.oupjournals.org)
- Keeping a public archive of papers, abstracts, etc., accessible via ftp.
- Wherever possible, obtaining reductions on group (6 or more) purchases of logic books from publishers.

If you are interested, please send your details (name, postal address, phone, fax, e-mail address, research interests) to:

IGPL Headquarters
c/o Prof. Dov Gabbay
King's College, Dept of Computer Science
Strand
London WC2R 2LS
United Kingdom
e-mail: dg@dcs.kcl.ac.uk

For the organisation, Dov Gabbay, Ruy de Queiroz and Hans Jürgen Ohlbach