Formalizing decisional and operational roles in legal contracts via term-modal logic

Stef Frijters[1], Matteo Pascucci[2]

[1]KU Leuven, Belgium
[2]Central European University, Austria

Abstract

Translations of legal contracts into formal specifications that can be used for assisted reasoning are currently gaining considerable attention in AI and law. Yet, the conceptual intricacy of some of the normative notions involved in legal contracts continues to provide significant challenges to formalization; in accordance with this, there is a need for developing general logic frameworks which allow for an appropriate analysis of the fundamental components of a contractual situation. In the present work, we focus on the representation of decisional and operational roles played by possibly distinct contracting parties. We provide a flexible framework, which extends term-modal logic, where such roles can be effectively formalized and emphasized.

Keywords

Formal specification of legal contracts, logics in computer science, normative roles, decision and operation, term-modal logic

1. Introduction

The formal analysis of legal contracts is an interdisciplinary area drawing attention from an increasing number of researchers. For a recent and systematic review of approaches offered in the literature, see Soavi et al. [?]. Our discussion will be limited to approaches based primarily on deductive methods, as opposed to those based primarily on statistical methods. The basic ingredients of a deductively-driven analysis of legal contracts can be described as follows: (i) identifying relevant *normative concepts* in contracts (e.g. 'right', 'duty', 'power', 'liability', etc.) (ii) developing a theory of the *semantic relations* among such concepts (e.g. holding that rights and duties are correlatives, in the sense that a party x has a duty towards a party y iff y has a corresponding right against x), (iii) identifying the *range of possible scenarios* to which a type of contract applies (e.g. a sales agreement which concerns some category of goods and involves parties playing some roles), (iv) generating a *formal language* and a *deductive theory* over it that can be assessed with respect to the previous components.

Once one obtains a formal language that tracks all necessary concepts of a contract and a deductive theory that handles both semantic relations and assumptions about scenarios regulated by the contract, one can proceed further by automating reasoning tasks and/or creating a tool for user assistance. As noted in [?], the formal analysis of contracts is part of a broader endeavour to automate aspects of normative reasoning. Thus, in the literature one finds both approaches that are tailored to (certain types of) contracts, such as Montazeri et al. [?] or He et al. [?], and more abstract approaches that provide a flexible framework for normative reasoning, such as Libal and Pascucci [?] or Steen and Fuenmayor [?]. Despite the number and variety of these approaches, the normative domain continues to provide challenges to formalization, due to the conceptual intricacy of some of its core notions. The present work contributes to this research area by addressing the problem of formalizing two fundamental layers of roles in a legal contract: *decisional roles* and *operational roles*. We do this at a general level of analysis, providing a framework that allows for representing different combinations of such roles.

OVERLAY 24: 6th International Workshop on Artificial Intelligence and fOrmal VERification, Logic, Automata, and sYnthesis
**This article results from a joint research work of the two authors.

2. A working example on decisional and operational roles

We can briefly illustrate the difference between decisional roles and operational roles in a contract via an example. Suppose that *Anna*, a student, wants to spend a period of one month at a university in a foreign country and that she looks for an accommodation via a company called *Student Housing*. Such a company happens to have a flat for Anna, which can be booked under the following condition: a guarantor from Anna's home institution is required to pay a deposit of 200€ via online transfer to an account associated with Student Housing at a banking institution called *InterBank*. For our purposes we can neglect the additional condition that Anna will have to pay the rest of the rental fee by a certain date. Anna accepts these conditions, finds a suitable guarantor in *Prof. Benvenuti* and stipulates a legal contract with Student Housing including, among other things, all the information mentioned above.

Now, let us look at the parties emphasized in the example and the roles they play: Anna is the *tenant*, Student Housing is the *flat owner*, Prof. Benvenuti is the *guarantor* and InterBank is the *payment addressee*. The tenant and the flat owner are decisional roles, since the parties playing these roles stipulate the contract at issue. By contrast, the guarantor and the payment addressee play operational roles, since they are involved in the execution, rather than in the stipulation, of the contract.

In principle, the same roles could be played by different parties in similar circumstances (another student interested in renting another flat with another company, etc.). Thus, when we want to design a formal specification for this *type* of contract, it is convenient to keep track of the mentioned roles. More generally, every type of contract brings with it certain decisional and operational roles that are crucial to understand its content and that are invoked when one wants to check whether the contract was fulfilled or violated (e.g. the guarantor did not pay the deposit by the given deadline).

3. Term-modal logic

Here we propose a formal framework that emphasizes the distinction between decisional and operational roles in a contract and show its advantages over a standard representation in first-order (deontic) logic. Our framework is an extension of *term-modal logic*, introduced by Fitting, Thalmann and Voronkov [**?**], applied to normative reasoning by Frijters, Meheus and Van De Putte [**?**] and further developed in that context by Frijters [**? ?**].

Term-modal logic (hereafter, TML) is an extension of first-order logic (hereafter, FOL) by means of *indexed modal operators* of the form $\Box_\sigma$, where σ is a finite list of terms (individual variables or constants). These terms stand for *normative parties* and $\Box$ will here represent a deontic operator of *obligation*. The additional expressiveness of TML in comparison to FOL allows one to formally capture the distinction between quantification over terms that *occur as indices* of a modal operator and quantification over terms that *occur in the scope* of a modal operator. Such a distinction is exemplified by the occurrences of x and y in the TML-formula below:

$$\forall xy((\texttt{Doctor}(y) \land \texttt{PatientOf}(x,y)) \to \Box_{yx}\texttt{Cure}(y,x))$$

We can read the whole formula as "for every x and y, if y is a doctor and x is a patient of y, then it is obligatory for y towards x that y cures x". In other words, every doctor has an obligation towards their patients to cure them. The occurrences of x and y as indices of $\Box$ in the formula at issue convey an *agent-directed obligation*, rendered by the natural language expression "for y towards x" which could not be directly captured in the language of (modal) FOL, as discussed in [**?**]. Clearly, being associated with a direction is a fundamental feature of many obligations. Usually, if we change the direction of an obligation, we give rise to a completely different norm. For instance, replacing $\Box_{yx}$ with $\Box_{xy}$ in the TML-formula above yields the (odd) reading that it is obligatory for patients towards doctors that the latter cure the former.

Our extension of TML will be called *layered term-modal logic* (hereafter, LTML) and is inspired by the following observation, made by Novotná and Pascucci in [**?**]. Norms are often not reducible to *directed obligations*; they rather result from an *agreement* or a *contract*: some parties stipulate that other (possibly different) parties have to behave a certain way. Thus, a norm can conceptually be analysed as a structure involving two layers of parties: one includes those parties that stipulate the agreement (i.e. play *decisional roles*), the other those parties that are expected to act in accordance with the agreement (i.e. play *operational roles*).

4. The formal framework

Let $C = \{a, b, \ldots\}$ be a countable set of individual constants and $V = \{x, y, \ldots\}$ a countable set of individual variables, where $\alpha, \beta, \ldots$ range over C and $\nu, \xi, \ldots$ over V. Let $T = C \cup V$ be the set of *terms* and $\theta, \kappa, \ldots$ the metavariables ranging over it. We use $\sigma_1, \sigma_2, \ldots$ to refer to finite (*possibly empty*) lists of terms. Let Σ_n, for $n \in \mathbb{N}$, denote the set of all lists of terms with cardinality n and Σ the union of all Σ_n. We take $\overline{\nu} = \langle \nu_1, \ldots, \nu_n \rangle$ and $\forall \overline{\nu}$ as an abbreviation for $\forall \nu_1 \ldots \forall \nu_n$. Finally, for $n \in \mathbb{N}$, let $\mathcal{P}^n$ be a countable set of n-ary predicate symbols and $\mathcal{P}$ denote the union of all $\mathcal{P}^n$.

LTML-*formulas* are defined as below, where $P \in \mathcal{P}^n$, $\sigma_1, \sigma_2, \sigma_3, \sigma_4 \in \Sigma$, $\theta, \kappa \in T$, and $\nu \in V$:

$$\varphi ::= P(\theta_1, \ldots, \theta_n) \mid \theta = \kappa \mid \neg\varphi \mid \varphi \vee \varphi \mid (\forall \nu)\varphi \mid \Delta^{(\sigma_1 \Rightarrow \sigma_2)}_{(\sigma_3 \Rightarrow \sigma_4)}\varphi \mid \Delta^{(\sigma_1 \Rightarrow \sigma_2)}_{(\sigma_3 \not\Rightarrow \sigma_4)}\varphi$$

We take symbols $\top$, $\bot$, $\wedge$, $\rightarrow$, $\leftrightarrow$ and $\exists$ to be defined in a standard way. We write $\theta \neq \kappa$ as an abbreviation for $\neg\ \theta = \kappa$. Parentheses are omitted whenever possible. Free and bound variables in formulas are defined as usual, with one addition: the free variables in $\Delta^{(\sigma_1 \Rightarrow \sigma_2)}_{(\sigma_3 \Rightarrow \sigma_4)}$ are all the variables that occur free in φ and all the variables that occur in $\sigma_1, \sigma_2, \sigma_3$ and σ_4. A formula φ is a *sentence* iff all variables in φ are bound. We use $\mathbf{Sent}$ for the set of all LTML-sentences.

The additional expressiveness of LTML in comparison to FOL is given by the following features: for every four lists $\sigma_1, \ldots, \sigma_4$ of terms, we have the operators $\Delta^{(\sigma_1 \Rightarrow \sigma_2)}_{(\sigma_3 \Rightarrow \sigma_4)}$ and $\Delta^{(\sigma_1 \Rightarrow \sigma_2)}_{(\sigma_3 \not\Rightarrow \sigma_4)}$, which take a formula as input and give a formula as output. These operators were introduced in [?] within a propositional framework. They can be regarded as special term-modal operators, given that they are modal operators indexed with a quadruple of lists $(\sigma_1, \ldots, \sigma_4)$ grouped into two layers. In the present context, the upper layer indicates parties playing decisional roles, whereas the lower layer indicates parties playing operational roles. Moreover, each layer has a *direction* thanks to the presence of symbols $\Rightarrow$ and $\not\Rightarrow$. We will call $\Delta^{(\sigma_1 \Rightarrow \sigma_2)}_{(\sigma_3 \Rightarrow \sigma_4)}$ and $\Delta^{(\sigma_1 \Rightarrow \sigma_2)}_{(\sigma_3 \not\Rightarrow \sigma_4)}$ *layered term-modal operators* (or LTML-operators).

The flexible reading of these operators can be explained by examining some schemata of formulas. If none of the lists $\sigma_1, \ldots, \sigma_4$ is empty, we get the interpretation in Table 1:

Formula	**Reading**
$\Delta^{(\sigma_1 \Rightarrow \sigma_2)}_{(\sigma_3 \Rightarrow \sigma_4)}\phi$	on the basis of a stipulation of σ_1 relevant to σ_2, σ_3 must bring about ϕ towards σ_4
$\Delta^{(\sigma_1 \Rightarrow \sigma_2)}_{(\sigma_3 \not\Rightarrow \sigma_4)}\phi$	on the basis of a stipulation of σ_1 relevant to σ_2, σ_3 must avoid bringing about ϕ towards σ_4

Table 1
Reading of layered term-modal operators

This reading indicates that σ_1 and σ_3 (which occur to the left of arrow symbols) are lists of normative parties playing *active roles*, while σ_2 and σ_4 (which occur to the right of arrow symbols) are lists of normative parties playing *passive roles.* Normative parties can also be said to be *legal agents*, where these are not necessarily individual agents (e.g. sometimes a legal agent is an institution like a bank). Notice, however, that the same party can play both an active role and a passive role, as well as occur both in the upper layer (decisional) and in the lower layer (operational). For instance, consider the occurrences of x or of z in the following schema: $\Delta^{(x,y \Rightarrow x,z)}_{(x,w \Rightarrow z)}\phi$. Accordingly, given that most legal contracts are such that all agreeing parties play *both* an active decisional role (being involved in the stipulation) and a passive decisional role (being affected by the fulfilment/violation of the contract), it is often the case that σ_1 and σ_2 are *the same lists of parties*. In those cases, one can adopt a graphical convention and write e.g. $\Delta^{(x,y \Leftrightarrow x,y)}_{(w \Rightarrow z)}\phi$, with a *bidirectional arrow* which emphasizes the reciprocity of the contract at the decisional level. We can use an instance of the latter schema to interpret the working example from section **??**: we assume that x is Anna (tenant), y is Student Union (flat owner), w is Prof. Benvenuti (guarantor), z is InterBank (payment addressee) and let $\phi = \texttt{PayOnline(200€)}$ describe the fact that 200€ are paid via online transfer. The whole can be read in accordance with Table **??**, namely: on the basis of a stipulation between Anna and Student Union, Prof. Benvenuti has to pay 200€ via online transfer to InterBank.

In order to move from the representation of a *specific contract* to the representation of a *type of contract*, we add *role-labels* to the positions occupied by indices of an LTML-operator, so as to obtain an *interpreted pattern of LTML-operator*. In this regard, consider the type of contract instantiated by the working example. We identified four fundamental roles in the working example (tenant, flat owner, guarantor and payment addressee). We can thus define the following conditions on an operator $\Delta^{(\sigma_1 \Rightarrow \sigma_2)}_{(\sigma_3 \Rightarrow \sigma_4)}$: both σ_1 and σ_2 is an ordered pair of terms (the same pair), where the first term plays the role of $tenant$ and the second term plays the role of $flat\ owner$; σ_3 consists of a single term playing the role of $guarantor$; finally, σ_4 consists of a single term playing the role of $payment\ addressee$. The result is an interpreted pattern of LTML-operator where we can use any indices different from x, y, w and z to formalize a new instance of the *same* type of contract mentioned in the working example. Different types of contracts will be associated with other interpreted patterns of LTML-operators. In other words, we can always introduce a new set of role-labels to build interpreted patterns of LTML-operators, depending on the sort of legal contract that we want to analyse.

If some of the lists of terms in an LTML-operator are empty, the reading of the formulas in Table **??** can be changed accordingly. For a systematic analysis of all possible cases, see [**?**]. For instance, if $\sigma_1 = \emptyset$ and $\sigma_2 \neq \emptyset$, we are representing a norm based on the *fulfillment of a condition* (e.g. a norm applicable to the category of self-employed workers), if $\sigma_1 \neq \emptyset$ and $\sigma_2 = \emptyset$ we are representing a *unilateral decision* (e.g. a norm resulting from a judicial decision). If $\sigma_1 = \sigma_2 = \emptyset$, then the decisional layer is vacuously mentioned, and we are ultimately representing a norm without any reference to its source (e.g. the simple command "Carla ought to pay a rental fee to Emma"). An empty list of terms in an LTML-operator will also be denoted by a *blank space*. Moreover, $*$ will denote either $\Rightarrow$ or $\not\Rightarrow$.

The deductive relevance of empty lists of terms in LTML-operators is clarified via the notion of *reference-abstraction*, which indicates how one can safely perform an inference from a norm ψ_1 contained in a legal contract to a simpler norm ψ_2 by removing (some) reference to some of the parties involved in ψ_1, either at the decisional level or at the operational level. Notice that any norm which is inferred by following this procedure applies to *the same legal context as the original norm*. For instance, saying that Prof. Benvenuti ought to pay 200€ to InterBank due to an agreement between Anna and Student Union implies that Prof. Benvenuti ought to pay 200€ due to an agreement between Anna and Student Union. In performing such an inference, the important point to keep in mind is that the latter norm should be interpreted in a *context-relative* way, namely as based on the information available in the particular contract analysed. We cannot apply the inferred norm to arbitrary circumstances. To better illustrate the idea of context-relativity, it is convenient to look at an even simpler norm that we can infer in the working example, namely that Prof. Benvenuti ought to pay 200€. Such a norm says something about the specific legal scenario analysed and should not be confused with the *absolute norm* according to which Prof. Benvenuti ought to pay 200€ *under any circumstances and to an arbitrary party*.

Another noteworthy aspect of operators endowed with indices of normative parties and with a direction, like our operator $\Delta^{(\sigma_1 \Rightarrow \sigma_2)}_{(\sigma_3 \Rightarrow \sigma_4)}$, is that they can be used to express *Hohfeldian concepts* and their relations; this aspect is more extensively discussed in [**?**]. In particular, consider the *correlativity* of rights and duties: x has a duty towards y iff y has a right against x. Under this view, the fact that Prof. Benvenuti ought to pay 200€ to InterBank can be interpreted both as a *duty* of Prof. Benvenuti towards InterBank and as a *right* of InterBank against Prof. Benvenuti. Therefore, as far as the symbolic language is concerned, both these instances of Hohfeldian concepts can be expressed by the same formula.

Definition 1 (Reference-abstraction). *Let* $\psi_1 = \Delta^{(\sigma_1 \Rightarrow \sigma_2)}_{(\sigma_3\, *\, \sigma_4)}\phi$*; a* reference-abstraction *of* ψ_1 *is any LTML-formula* ψ_2 *obtained from* ψ_1 *by removing* some *occurrence of an index in its main LTML-operator.*

The relation of being a reference-abstraction is taken to be *transitive*. Thus, $\Delta^{(y \Rightarrow \;)}_{(w \Rightarrow z)}\phi$ is a reference-abstraction of $\Delta^{(x,y \Rightarrow y)}_{(x,w \Rightarrow z)}\phi$ and $\Delta^{(\sigma_1 \Rightarrow \;)}_{(\sigma_3 \Rightarrow \;)}$ is a reference-abstraction of $\Delta^{(\sigma_1 \Rightarrow \sigma_2)}_{(\sigma_3 \Rightarrow \sigma_4)}$.[1]

5. Semantics

We provide a semantics for LTML based on *neighborhood models*. For a related semantics in the case of TML, see Frijters and Van De Putte [**?**]. We stress that reference-abstraction is addressed by clause 3.1 of the following definition. Given a set X, we take $\mathbf{L}(X)$ to be the set of all finite lists (sequences) over X and $l_1(X) \sqsubseteq l_2(X)$ to mean that $l_1(X)$ is an order-preserving list formed by a subset of the objects in a list $l_2(X)$. In other words, $l_1(X) \sqsubseteq l_2(X)$ means that $l_1(X)$ and $l_2(X)$ are two lists over the same set X and the former is obtained from the latter by possibly removing some objects (normative parties, in the present context).

Definition 2. *An* LTML-model *is a tuple* $M = \langle W, \mathcal{A}, N, I\rangle$*, where:*

1. $W \neq \emptyset$ *is the* world domain *of* M
2. $\mathcal{A} \neq \emptyset$ *is the* agent domain *of* M
3. $N : W \times \mathbf{L}(\mathcal{A}) \times \mathbf{L}(\mathcal{A}) \times \mathbf{L}(\mathcal{A}) \times \mathbf{L}(\mathcal{A}) \to \wp(\wp(W))$ *is the* neighborhood function *of* M*, where:*
   3.1. *if* $X \in N(w, l_1(\mathcal{A}), l_2(\mathcal{A}), l_3(\mathcal{A}), l_4(\mathcal{A}))$, $l'_1(\mathcal{A}) \sqsubseteq l_1(\mathcal{A}), l'_2(\mathcal{A}) \sqsubseteq l_2(\mathcal{A}), l'_3(\mathcal{A}) \sqsubseteq l_3(\mathcal{A})$ *and* $l'_4(\mathcal{A}) \sqsubseteq l_4(\mathcal{A})$, *then* $X \in N(w, l'_1(\mathcal{A}), l'_2(\mathcal{A}), l'_3(\mathcal{A}), l'_4(\mathcal{A}))$
4. I *is an* interpretation function *of* M*, where:*
   4.1. $I : T \to \mathcal{A}$
   4.2. $I : \mathcal{P}^n \times W \to \wp(\mathcal{A}^n)$ *for every* $n \in \mathbb{N}$ *such that* $n \geq 1$.
   4.3. $I : \mathcal{P}^0 \to \wp(W)$

We stress that an n-ary predicate $\mathcal{P}^n$ is interpreted at each world w as a set of n-tuples of normative parties.

Given some model $M = \langle W, \mathcal{A}, N, I\rangle$ and $\sigma = \langle \theta_1, \dots, \theta_n\rangle$, we will henceforth use $I(\sigma)$ as shorthand for the tuple $\langle I(\theta_1), \dots, I(\theta_n)\rangle$. We call $|\varphi|_M = \{w \mid w \in W \text{ and } M, w \models \varphi\}$ the *truth set of* φ. For any $\nu \in V$, $M' = \langle W, \mathcal{A}, N, I'\rangle$ is a ν-alternative of $M = \langle W, \mathcal{A}, N, I\rangle$ iff I' differs at most from I in the member of $\mathcal{A}$ that I' assigns to ν.

Definition 3 (Semantic Clauses). *For any LTML-model* $M = \langle W, \mathcal{A}, N, I\rangle$ *and* $w \in W$*:*

SC1 *If* $P \in \mathcal{P}^n$ *for some* $n \geq 1$*, then* $M, w \models P(\sigma)$ *iff* $I(\sigma) \in I(P, w)$
SC1' *If* $P \in \mathcal{P}^0$*, then* $M, w \models P$ *iff* $w \in I(P)$
SC2 $M, w \models \neg\varphi$ *iff* $M, w \not\models \varphi$
SC3 $M, w \models \varphi \vee \psi$ *iff* $M, w \models \varphi$ *or* $M, w \models \psi$
SC4 $M, w \models \theta = \kappa$ *iff* $I(\theta) = I(\kappa)$
SC5 $M, w \models \Delta^{\sigma_1 \Rightarrow \sigma_2}_{\sigma_3 \Rightarrow \sigma_4}\varphi$ *iff* $|\varphi|_M \in N(w, I(\sigma_1), I(\sigma_2), I(\sigma_3), I(\sigma_4))$
SC6 $M, w \models (\forall\nu)\varphi$ *iff for every* ν*-alternative* M'*:* $M', w \models \varphi$.

Definition 4. *Let* $\Gamma \subseteq \mathbf{Sent}$ *and* $\varphi \in \mathbf{Sent}$*:* φ *is a semantic consequence of* Γ *iff for every LTML-model* $M = \langle W, \mathcal{A}, N, I\rangle$ *and* $w \in W$*: if* $M, w \models \psi$ *for all* $\psi \in \Gamma$*, then* $M, w \models \varphi$*. If* $\Gamma = \emptyset$*, then* ϕ *is valid.*

[1]The TML-operators used in [**?**] correspond to LTML-operators of the form $\Delta^{(\;\Rightarrow\;)}_{(\sigma_3 \Rightarrow\;)}$ or $\Delta^{(\sigma_1 \Rightarrow\;)}_{(\;\Rightarrow\;)}$, namely where only active roles in one layer are mentioned, while those used in [**? ?**] could match LTML-operators having any of the forms $\Delta^{(\sigma_1 \Rightarrow \sigma_2)}_{(\;\Rightarrow\;)}$, $\Delta^{(\;\Rightarrow\;)}_{(\sigma_3 \Rightarrow \sigma_4)}$, $\Delta^{(\sigma_1 \Rightarrow\;)}_{(\;\Rightarrow \sigma_4)}$ and $\Delta^{(\;\Rightarrow \sigma_2)}_{(\sigma_3 \Rightarrow\;)}$, given that they allow for a distinction between active roles and passive roles, although they do not take layers of roles into account.

6. Axiomatic system

An axiomatization of LTML on neighborhood models is obtained by closing an axiomatization of classical propositional logic under the principles in Table **??**.[2] As in [**?**], we use the following notation: $\varphi(\theta/\kappa)$ is the result of replacing all free occurrences of κ in φ by θ, renaming bound variables if necessary to avoid rendering new occurrences of θ bound in $\varphi(\theta/\kappa)$. $\varphi(\theta/\!\!/\kappa)$ is the result of *possibly* replacing some free occurrences of κ in φ by θ, again renaming something if necessary.

A formula $\varphi \in \mathbf{Sent}$ is a theorem of LTML (in symbols, $\vdash \varphi$) iff φ can be derived from the axioms and rules of LTML. $\varphi \in \mathbf{Sent}$ is derivable from $\Gamma \subseteq \mathbf{Sent}$ in LTML (denoted $\Gamma \vdash \varphi$) iff there are $\psi_1, \ldots, \psi_n \in \Gamma$ such that $\vdash (\psi_1 \wedge \ldots \wedge \psi_n) \to \varphi$.

Label	**Axiom Schema**	**Label**	**Rule**
(UI)	$(\forall\nu)\varphi \to \varphi(\alpha/\nu)$	(RE)	if $\vdash \varphi \leftrightarrow \psi$, then $\vdash \Delta^{(\sigma_1 \Rightarrow \sigma_2)}_{(\sigma_3 \,*\, \sigma_4)}\varphi \leftrightarrow \Delta^{(\sigma_1 \Rightarrow \sigma_2)}_{(\sigma_3 \,*\, \sigma_4)}\psi$
(REF)	$\alpha = \alpha$	(MP)	to infer ψ from the assumptions ϕ and $\varphi \to \psi$
(SUB)	$(\alpha = \beta) \to (\varphi \to \varphi(\alpha/\!\!/\beta))$	(UG)	if $\vdash \varphi \to \psi(\alpha/\nu)$ and α neither occurs in φ nor in ψ, then $\vdash \varphi \to (\forall\nu)\psi$.
(ABS)	if ψ is a reference-abstraction of φ, then $\phi \to \psi$		

Table 2
Axiomatization

We emphasize that, in the axiomatic basis provided above, principle ABS gives rise to a form of Modus Ponens which captures the ideas behind reference-abstraction.

7. Final remarks

In this work, we presented an extension of term-modal logic called LTML (layered term-modal logic) and showed that it can properly account for the distinction between decisional roles and operational roles in a contract, which is a crucial component of legal reasoning. Our proposal combines intuitions put forward in [**?**] with a language whose expressiveness is higher than modal FOL. Our future inquiries in this area will be dedicated to the analysis of fragments of LTML that are suitable for an automated representation of selected types of contract and thus for the development of tools for assisted reasoning.

Acknowledgments

The following statement applies to Matteo Pascucci: This research was funded in whole or in part by the Austrian Science Fund (FWF) 10.55776/I6499. For open access purposes, the author has applied a CC BY public copyright license to any author-accepted manuscript version arising from this submission.

[2] Adding other principles such as aggregation or inheritance to the axiomatisation provided is straightforward; see [**?**].