Linear Contextual Modal Type Theory

Anders Schack-Nielsen
IT University of Copenhagen
Copenhagen, Denmark
anderssn@itu.dk

Carsten Schürmann
IT University of Copenhagen
Copenhagen, Denmark
carsten@itu.dk

Abstract

When one develops, implements, and studies type theories based on linear logic, for example, in the context of theorem proving, logic programming, and formal reasoning, one is immediately confronted with questions about their equational theory and how to deal with logic variables. In this paper, we propose linear contextual modal type theory that gives a mathematical account of the nature of logic variables. Our type theory is conservative over intuitionistic contextual modal type theory proposed by Nanevski, Pfenning, and Pientka. As a technical contribution we provide a proof of soundness, and, as a justification for its usefulness, we shed some light on the difficulties working with logic variables in linear logics that contain $\top$.

1 Introduction

Over recent years, linear logic has become increasingly popular as a logic for concurrency, stateful computation, and even security. So far, the idea of resource awareness has had far reaching consequences for the design and implementation of logics and logical frameworks; LLF [CP96], CLF [CPWW02] and even separation logic [Rey02] use ideas borrowed from linear logic at their cores. There are implementations of theorem provers, logic programming languages, and proof assistants that do implement linear logic, as for example, Lolli [HM94], and Celf [SNS08].

All of these implementations depend crucially on the choice of fragment of linear logic and the choice of logic variable. Logic variables stand for still to be proven leaves in a derivation tree or simply holes in a term that is to be instantiated via unification. It is a widely accepted fact that multiplicative connectives are better behaved than, for example, their additive siblings. It is also widely accepted that logic variables are useful.

As a motivating example consider the unification problem

$$c \mathbin{\hat{}} (F \mathbin{\hat{}} x \mathbin{\hat{}} y) = c \mathbin{\hat{}} (F \mathbin{\hat{}} y \mathbin{\hat{}} x)$$

where we write F for the logic variable, x, y for the two resources that need to be consumed exactly once, c for constant symbols, and $\mathbin{\hat{}}$ for linear application. If $\mathbin{\hat{}}$ were intuitionistic application then any instantiation of F with a constant function is a solution. In the multiplicative fragment of linear logic, the problem is not solvable because any instantiation of F will need to mention x and y exactly once on two different rigid paths. Thus the left hand side and the right hand side of the equation above will differ in these two places. If we were to work in linear logic with $\top$, the problem is also solvable by choosing the constant function $F = \widehat{\lambda} x \,.\, \widehat{\lambda} y \,.\, F' \mathbin{\hat{}} \langle\rangle$ where we write $\widehat{\lambda}$ for linear functional abstraction and $\langle\rangle$ for the proof term of $\top$.

This little example illustrates the complex nature of logic variables in linear logic and their role in higher-order linear unification. In the $\top$-free case, every linear resource needs to be consumed by the same term on either side of the equation. If this is impossible, unification fails. In the presence of $\top$ this is no longer the case. Therefore, without a clear understanding of the nature of logic variables from a mathematical point of view, it seems hopeless to try to devise and design algorithms for equality in linear logic.

In this paper we provide such an understanding by the means of linear contextual modal type theory that gives a precise mathematical meaning to logic variables for linear logic, building on ideas from contextual modal type theory by Nanevski, Pfenning, and Pientka [NPP08]. In their paper they work out a modal explanation of contextual validity, which accounts for the contexts that are usually associated with logic variables in the intuitionistic setting [DHKP96]. In this paper we define a contextual modal type theory for linear logic, which accounts for (in part at least) the definition of logic variables used for Cervesato's and Pfenning's linear preunification algorithm [CP97].

The underlying philosophical basis for this work is provided by Martin-Löf's view of logical truth in form of judgments and evidence in form of axioms and inference rules. Using his technique we construct the meaning of *availability*, which corresponds to the multiplicative fragment of linear logic with $\multimap$ as the main connective, *truth*, which cor-

responds to truth in intuitionistic logic with $\rightarrow$ as the main connective, and *contextual validity*, which corresponds to the logic of *logic variables* with $[\Gamma;\Delta]\!\!\rightarrow$ as the main connective (pronounced box arrow Γ, Δ). If we know that $A\ [\Gamma;\Delta]\!\!\rightarrow B$ is true, then a proof of B may mention a logic variable of type $[\Gamma;\Delta]A$, which may refer arbitrarily many times to assumptions in Γ and exactly once to assumptions in Δ. The justification of this construction can be found in Section 2.

Next we show the soundness of linear contextual modal logic in Section 3. To this end we give a sound and complete proof theoretical account of availability, truth, and contextual validity in form of a sequent calculus. Next we prove the admissibility of cut, which guarantees the existence of canonical proofs in linear contextual modal logic. The cut-elimination result of this section is formalized and machine-checked in Twelf [PS99].

Our logic is a bit non-standard because we do not explicitly internalize the judgments for truth and linear contextual validity. That this is not a loss is argued in Section 4, where we define an equivalent logic in terms of a ! modality that restores the intuitionistic implication $A \rightarrow B = !A \multimap B$ and a linear contextual modality $[\Gamma;\Delta]$ that restores the modal box implication $A\ [\Gamma;\Delta]\!\!\rightarrow\ B = [\Gamma;\Delta]A \multimap B$. In addition, this discussion justifies why linear contextual modal logic is in fact a *modal logic*.

In Section 5 we introduce a Curry-Howard correspondence. Every proof rule is endowed with a proof term in the spirit of Bierman [Bie94]. In this section we define all basic operations on logic variables including abstraction, instantiation, and substitution application. The choice of canonical proofs induces the equational theory based on β reduction and η expansion. We show that every term is equivalent to a β-normal η-long form.

Finally, in Section 6 we describe how we use linear contextual modal type theory to help us understand the role of the additive unit $\langle\rangle : \top$. Consider the following unification equation, where F and G are logic variables:

$$c\,\hat{}\,\langle\rangle\,\hat{}\,(F\,\hat{}\,x) = c\,\hat{}\,\langle\rangle\,\hat{}\,(d\,\hat{}\,G)$$

This time, there is only one resource x. On the left the resource x needs to be consumed by the F, but on the right the second argument of c cannot consume x, and it must therefore be consumed by the first argument, the $\langle\rangle$. But now, after decomposition the two resulting unification problems are no longer well-typed! The resource x must to be consumed on the right but not on the left in the first and vice versa in the second. Thus we would expect that there is no solution and wrong again. The solution is $F = \widehat{\lambda}x\,.\ d\,\hat{}\,(G'\,\hat{}\,\langle\rangle)$ and $G = G'\,\hat{}\,\langle\rangle$.

Linear contextual modal type theory presents the theoretical foundation of our implementation of the Celf proof assistant [SNS08].

2 Linear Contextual Modal Logic

The central idea in linear logic [Gir87] is that of a resource. Linear assumptions play the role of a fixed set of *available* resources that must be consumed (exactly once) in a derivation. Therefore, available resources form the philosophical foundation of linear contextual modal logic. The idea of linear logic as a resource oriented logic has rendered it attractive to many application areas. In Petri nets, tokens can be modeled as resources, in programming language theory it is state, and in security simply messages that are being created and consumed.

Traditionally one recovers intuitionistic logic from linear logic by singling out those resources that can be constructed from no other resources. They can be used as often as desired, and thus, constructively speaking, they are *true*.

Finally, we introduce the judgment of *contextual validity*, which will ultimately serve as the logical justification of the existence of logic variables. Usually we say a proposition is valid if it is true in all contexts. But here we refine this idea one step further and refer to the validity of a proposition in a context $\Gamma;\Delta$, where Γ is a collection of true propositions, and Δ is a collection of available resources.

These three judgments can be defined by a set of inference rules and axioms following the ideas of the judgmental reconstruction of modal logic that goes back to Davies and Pfenning [DP01].

Linear Judgments In linear logic, resources are constructed from other resources, all of which are necessarily consumed during the process. We call judgments of this form *linear judgments*. If A is constructed using each linear resource among $A_1 \ldots A_n$ exactly once, we write

$$x_1 : A_1\ \mathsf{avail}, \ldots, x_n : A_n\ \mathsf{avail} \vdash A\ \mathsf{avail}$$

The list of linear resources to the left of the $\vdash$ symbol enjoys among the three structural properties only exchange (and neither weakening nor contraction) and will be abbreviated in the remainder of this paper by the aforementioned Δ. In the remainder of the paper we refer to $x_i : A_i\ \mathsf{avail}$ as a *linear assumption* and to A_i as a *resource*.

In our formulation of the rules that define introduction and elimination forms for the linear implication connective $\multimap$, we use the $\vdash$ symbol as a notational convenience for accounting all resources consumed by the derivation $A\ \mathsf{avail}$.

$$\frac{}{x{:}A\ \mathsf{avail} \vdash A\ \mathsf{avail}}\ \mathsf{lhyp}_x$$

$$\frac{(\Delta, x{:}A\ \mathsf{avail}) \vdash B\ \mathsf{avail}}{\Delta \vdash A \multimap B\ \mathsf{avail}} \multimap \mathsf{I}^x$$

$$\frac{\Delta_1 \vdash A \multimap B\ \mathsf{avail} \quad \Delta_2 \vdash A\ \mathsf{avail}}{(\Delta_1, \Delta_2) \vdash B\ \mathsf{avail}} \multimap \mathsf{E}$$

Theorem 2.1 (Principle of substitution) *If* $\Delta_1 \vdash A$ avail *and* $(\Delta_2, x{:}A \text{ avail}) \vdash B$ avail *then* $(\Delta_1, \Delta_2) \vdash B$ avail.

Proof: by induction on the second assumption. □

Hypothetical Judgments We define the judgment A true to mean that A is always available. This is only possible in the case that the derivation of A avail does not consume any resources. In compliance with the literature, we call the truth judgment a *hypothetical judgment* because it may rely on the assumptions that $x_1 : A_1$ true, $\ldots, x_n : A_n$ true.

$$x_1 : A_1 \text{ true}, \ldots, x_n : A_n \text{ true} \vdash A \text{ true}$$

In contrast to above, this list enjoys all structural properties, i.e. weakening, exchange, and contraction. In the following we will abbreviate the list of assumptions by Γ and refer to $x_i : A_i$ as an *intuitionistic assumption* and to any A_i in Γ as a *hypothesis*.

Using $\Gamma; \Delta \vdash A$ avail as an appropriate notational device for linear judgments, we introduce truth in the following way.

$$\frac{\Gamma; \cdot \vdash A \text{ avail}}{\Gamma \vdash A \text{ true}}$$

Furthermore, we define the rules regarding the introduction and elimination of the intuitionistic implication connective $\rightarrow$ as follows.

$$\frac{}{(\Gamma, x{:}A \text{ true}); \cdot \vdash A \text{ avail}}\ \mathsf{hyp}_x$$

$$\frac{(\Gamma, x{:}A \text{ true}); \Delta \vdash B \text{ avail}}{\Gamma; \Delta \vdash A \rightarrow B \text{ avail}} \rightarrow \mathsf{I}^x$$

$$\frac{\Gamma; \Delta \vdash A \rightarrow B \text{ avail} \quad \Gamma \vdash A \text{ true}}{\Gamma; \Delta \vdash B \text{ avail}} \rightarrow \mathsf{E}$$

Theorem 2.2 (Principle of substitution) *If* $\Gamma; \cdot \vdash A$ avail *and* $(\Gamma, x{:}A \text{ true}); \Delta \vdash B$ avail *then* $\Gamma; \Delta \vdash B$ avail.

Proof: by induction on the second assumption. □

Categorical Judgments We write A valid$[\Gamma; \Delta]$ if A is valid relative to a list of hypothetical judgments Γ and linear judgments Δ. This means that A is available whenever all hypotheses in Γ are true and all resources in Δ are available. Furthermore, we write

$$u_1 :: A_1 \text{ valid}[\Gamma_1; \Delta_1] \ldots u_n :: A_n \text{ valid}[\Gamma_n; \Delta_n] \vdash A \text{ valid}[\Gamma; \Delta]$$

for the judgments of relative validity. As above, the list of assumptions to the left of the $\vdash$ symbol enjoys the full set of structural properties, including exchange, weakening, and contraction. In the interest of clarity, we abbreviate this list by Ψ, and refer to a declaration A_i valid$[\Gamma_i; \Delta_i]$ as a *contextual modal hypothesis*.

$$\frac{\Psi; \Gamma; \Delta \vdash A \text{ avail}}{\Psi \vdash A \text{ valid}[\Gamma; \Delta]}$$

When using a contextual modal hypothesis in $\Psi; \Gamma; \Delta$ of type A valid$[\Gamma'; \Delta']$, we need to make sure that all hypotheses Γ' are true and all resources in Δ' can be provided. More formally, we express this fact by $\Psi; \Gamma; \Delta \vdash \Gamma'; \Delta'$, which is defined as follows: all hypotheses A in Γ' must satisfy $\Psi; \Gamma \vdash A$ true, and for some partition of $\Delta = \Delta_1, \ldots, \Delta_n$ the following holds: $\Psi; \Gamma; \Delta_i \vdash B_i$ avail for all resources B_i in Δ'. The presence of this new kind of contextual assumption gives rise to a new arrow, which is defined by the following introduction and elimination rule.

$$\frac{(\Psi, u :: A \text{ valid}[\Gamma'; \Delta']); \Gamma; \Delta \vdash \Gamma'; \Delta'}{(\Psi, u :: A \text{ valid}[\Gamma'; \Delta']); \Gamma; \Delta \vdash A \text{ avail}}\ \mathsf{mhyp}_u$$

$$\frac{(\Psi, u :: A \text{ valid}[\Gamma'; \Delta']); \Gamma; \Delta \vdash B \text{ avail}}{\Psi; \Gamma; \Delta \vdash A\ [\Gamma'; \Delta'] \!\rightarrowtail B \text{ avail}} \Box\!\rightarrow \mathsf{I}^u$$

$$\frac{\Psi; \Gamma; \Delta \vdash A\ [\Gamma'; \Delta'] \!\rightarrowtail B \text{ avail} \quad \Psi \vdash A \text{ valid}[\Gamma'; \Delta']}{\Psi; \Gamma; \Delta \vdash B \text{ avail}} \Box\!\rightarrow \mathsf{E}$$

Theorem 2.3 (Principle of substitution) *If* $\Psi; \Gamma'; \Delta' \vdash A$ avail *and* $(\Psi, u :: A \text{ valid}[\Gamma'; \Delta']); \Gamma; \Delta \vdash B$ avail *then* $\Psi; \Gamma; \Delta \vdash B$ avail.

Proof: by induction on the second assumption. □

Figure 1 summarizes the rules that define the meaning of the connectives purely in terms of availability. This can always be achieved because truth and contextual validity is defined by one single rule, which is always invertible. We call the resulting logic linear contextual modal logic (LCML).

Returning to our motivating example, contextual modal assumptions are logic variables.

Example 2.4 (Logic Variables) $u :: A$ valid$[\cdot; \cdot]$ is a logical variable that can only be instantiated by closed terms (that may neither refer to intuitionistic or linear assumptions.) $v :: A$ valid$[\cdot; (u{:}B; v{:}C)]$ is a logical variable that must consume the two respective resources u and v exactly once. $w :: A$ valid$[u{:}B; v{:}C]$ is a logical variable that must consume the resource v, but may mention u arbitrary many times.

3 Proof Theory

The rules defining the meaning of the connectives of linear contextual modal logic are sound in the sense that we understand the meaning of a proposition by examining only

$$\frac{}{\Psi; (\Gamma, x{:}A\ \mathsf{true}); \cdot \vdash A\ \mathsf{avail}}\ \mathsf{hyp}_x \quad \frac{}{\Psi; \Gamma; x{:}A\ \mathsf{avail} \vdash A\ \mathsf{avail}}\ \mathsf{lhyp}_x \quad \frac{(\Psi, u{::}A\ \mathsf{valid}[\Gamma'; \Delta']); \Gamma; \Delta \vdash \Gamma'; \Delta'}{(\Psi, u{::}A\ \mathsf{valid}[\Gamma'; \Delta']); \Gamma; \Delta \vdash A\ \mathsf{avail}}\ \mathsf{mhyp}_u$$

$$\frac{\Psi; (\Gamma, x{:}A\ \mathsf{true}); \Delta \vdash B\ \mathsf{avail}}{\Psi; \Gamma; \Delta \vdash A \to B\ \mathsf{avail}} \to \mathsf{I}^x \quad \frac{\Psi; \Gamma; \Delta \vdash A \to B\ \mathsf{avail} \quad \Psi; \Gamma; \cdot \vdash A\ \mathsf{avail}}{\Psi; \Gamma; \Delta \vdash B\ \mathsf{avail}} \to \mathsf{E}$$

$$\frac{\Psi; \Gamma; (\Delta, x{:}A\ \mathsf{avail}) \vdash B\ \mathsf{avail}}{\Psi; \Gamma; \Delta \vdash A \multimap B\ \mathsf{avail}} \multimap \mathsf{I}^x \quad \frac{\Psi; \Gamma; \Delta_1 \vdash A \multimap B\ \mathsf{avail} \quad \Psi; \Gamma; \Delta_2 \vdash A\ \mathsf{avail}}{\Psi; \Gamma; (\Delta_1, \Delta_2) \vdash B\ \mathsf{avail}} \multimap \mathsf{E}$$

$$\frac{(\Psi, u{::}A\ \mathsf{valid}[\Gamma'; \Delta']); \Gamma; \Delta \vdash B\ \mathsf{avail}}{\Psi; \Gamma; \Delta \vdash A\ [\Gamma'; \Delta'] \!\mapsto B\ \mathsf{avail}} \Box\!\!\mapsto \mathsf{I}^u \quad \frac{\Psi; \Gamma; \Delta \vdash A\ [\Gamma'; \Delta'] \!\mapsto B\ \mathsf{avail} \quad \Psi; \Gamma'; \Delta' \vdash A\ \mathsf{avail}}{\Psi; \Gamma; \Delta \vdash B\ \mathsf{avail}} \Box\!\!\mapsto \mathsf{E}$$

....

$$\frac{}{\Psi; \Gamma; \cdot \vdash \cdot; \cdot} \quad \frac{\Psi; \Gamma; \Delta \vdash \Gamma'; \cdot \quad \Psi; \Gamma; \cdot \vdash A\ \mathsf{avail}}{\Psi; \Gamma; \Delta \vdash (\Gamma', x{:}A); \cdot} \quad \frac{\Psi; \Gamma; \Delta_1 \vdash \Gamma'; \Delta' \quad \Psi; \Gamma; \Delta_2 \vdash A\ \mathsf{avail}}{\Psi; \Gamma; (\Delta_1, \Delta_2) \vdash \Gamma'; (\Delta', x{:}A)}$$

Figure 1. Natural Deduction Calculus for LCML

its constituents, in analogy to how global soundness is defined for contextual modal type theory [NPP08].

In pursuit of establishing soundness, we proceed by defining a sequent calculus for our logic for which we then show cut-elimination. Furthermore, we can argue that the sequent calculus is a sound and complete characterization of the rules introduced above.

The defining judgment of the sequent calculus for linear contextual modal logic is written as

$$\Psi; \Gamma; \Delta \Longrightarrow A.$$

The judgment holds if A can be proved from contextual modal hypotheses Ψ, true hypotheses Γ, and available hypotheses Δ.

In analogy to the previous section, we generalize this judgment to lists of true and available hypotheses $\Gamma; \Delta$, for which we write formally

$$\Psi; \Gamma; \Delta \Longrightarrow \Gamma'; \Delta'.$$

The rules defining the sequent calculus are summarized in Figure 2. The initial rule is defined for atomic propositions P only

$$\frac{}{\Psi; \Gamma; P \Longrightarrow P}\ \mathsf{init}.$$

The *identity principle* holds for the sequent calculus. Every proposition A follows from itself as linear hypothesis.

Theorem 3.1 (Identity principle) *For all propositions A, and contexts, Ψ, Γ, Γ', and Δ:*

1. $\Psi; \Gamma; A \Longrightarrow A$

2. $\Psi; \Gamma; \Delta \Longrightarrow \Gamma; \Delta$

3. $\Psi; (\Gamma, \Gamma'); \cdot \Longrightarrow \Gamma; \cdot$

Proof: By an easy mutual induction over the structure of A for 1., Δ for 2., and Γ for 3. ☐

Lemma 3.2 (Weakening)

1. *If* $\Psi; \Gamma; \Delta \Longrightarrow C$ *then* $\Psi; \Gamma, A; \Delta \Longrightarrow C$.
2. *If* $\Psi; \Gamma; \Delta \Longrightarrow C$ *then* $\Psi, A[\Gamma'', \Delta'']; \Gamma; \Delta \Longrightarrow C$.
3. *If* $\Psi; \Gamma; \Delta \Longrightarrow \Gamma'; \Delta'$ *then* $\Psi; \Gamma, A; \Delta \Longrightarrow \Gamma'; \Delta'$.
4. *If* $\Psi; \Gamma; \Delta \Longrightarrow \Gamma'; \Delta'$ *then* $\Psi, A[\Gamma'', \Delta'']; \Gamma; \Delta \Longrightarrow \Gamma'; \Delta'$.

Proof: by a mutual structural induction on the given sequent derivations. ☐

The *cut principle* also holds.

Theorem 3.3 (Cut Principle) *We must consider three different cuts, one for each of the three fragments.*

Linear cut: *If* $\Psi; \Gamma; \Delta_1 \Longrightarrow A$ *and* $\Psi; \Gamma; (\Delta_2, A) \Longrightarrow C$ *then* $\Psi; \Gamma; (\Delta_1, \Delta_2) \Longrightarrow C$.

Hypothetical cut: *If* $\Psi; \Gamma; \cdot \Longrightarrow A$ *and* $\Psi; (\Gamma, A); \Delta \Longrightarrow C$ *then* $\Psi; \Gamma; \Delta \Longrightarrow C$.

Categorical cut: *If* $\Psi; \Gamma'; \Delta' \Longrightarrow A$ *and* $(\Psi, A[\Gamma'; \Delta']); \Gamma; \Delta \Longrightarrow C$ *then* $\Psi; \Gamma; \Delta \Longrightarrow C$.

Proof: By structural induction lexicographically on the cut-formula A, the left, and the right derivation. The proof has been formalized in Twelf, and is accessible on the web at `http://www.itu.dk/~anderssn/lcml.tgz`. ☐

$$\frac{}{\Psi;\Gamma;P \Longrightarrow P}\,\mathsf{init} \qquad \frac{\Psi;(\Gamma,A);(\Delta,A) \Longrightarrow C}{\Psi;(\Gamma,A);\Delta \Longrightarrow C}\,\mathsf{copy} \qquad \frac{(\Psi,A[\Gamma';\Delta']);\Gamma;\Delta_1 \Longrightarrow \Gamma';\Delta' \quad (\Psi,A[\Gamma';\Delta']);\Gamma;(\Delta_2,A) \Longrightarrow C}{(\Psi,A[\Gamma';\Delta']);\Gamma;(\Delta_1,\Delta_2) \Longrightarrow C}\,\mathsf{reflect}$$

$$\frac{\Psi;(\Gamma,A);\Delta \Longrightarrow B}{\Psi;\Gamma;\Delta \Longrightarrow A \to B}\to\mathsf{R} \qquad \frac{\Psi;\Gamma;\cdot \Longrightarrow A \quad \Psi;\Gamma;(\Delta,B) \Longrightarrow C}{\Psi;\Gamma;(\Delta,A\to B) \Longrightarrow C}\to\mathsf{L}$$

$$\frac{\Psi;\Gamma;(\Delta,A) \Longrightarrow B}{\Psi;\Gamma;\Delta \Longrightarrow A \multimap B}\multimap\mathsf{R} \qquad \frac{\Psi;\Gamma;\Delta_1 \Longrightarrow A \quad \Psi;\Gamma;(\Delta_2,B) \Longrightarrow C}{\Psi;\Gamma;(\Delta_1,\Delta_2,A\multimap B) \Longrightarrow C}\multimap\mathsf{L}$$

$$\frac{(\Psi,A[\Gamma';\Delta']);\Gamma;\Delta \Longrightarrow B}{\Psi;\Gamma;\Delta \Longrightarrow A\ [\Gamma';\Delta']\!\!\rightarrowtail B}\Box\!\!\rightarrowtail\mathsf{R} \qquad \frac{\Psi;\Gamma';\Delta' \Longrightarrow A \quad \Psi;\Gamma;(\Delta,B) \Longrightarrow C}{\Psi;\Gamma;(\Delta,A\ [\Gamma';\Delta']\!\!\rightarrowtail B) \Longrightarrow C}\Box\!\!\rightarrowtail\mathsf{L}$$

$$\frac{}{\Psi;\Gamma;\cdot \Longrightarrow \cdot;\cdot}\,\mathsf{id} \qquad \frac{\Psi;\Gamma;\Delta \Longrightarrow \Gamma';\cdot \quad \Psi;\Gamma;\cdot \Longrightarrow A}{\Psi;\Gamma;\Delta \Longrightarrow (\Gamma',A);\cdot}\,\mathsf{dot} \qquad \frac{\Psi;\Gamma;\Delta_1 \Longrightarrow \Gamma';\Delta' \quad \Psi;\Gamma;\Delta_2 \Longrightarrow A}{\Psi;\Gamma;(\Delta_1,\Delta_2) \Longrightarrow \Gamma';(\Delta',A)}\,\mathsf{ldot}$$

Figure 2. Sequent Calculus of LCML

Theorem 3.4 (Soundness and Completeness)

1. $\Psi;\Gamma;\Delta \vdash A$ avail *if and only if* $\Psi;\Gamma;\Delta \Longrightarrow A$.
2. $\Psi;\Gamma;\Delta \vdash \Gamma';\Delta'$ *if and only if* $\Psi;\Gamma;\Delta \Longrightarrow \Gamma';\Delta'$.

Proof: The proofs are by mutual structural induction on the respective derivations. They have been formalized in Twelf, and are accessible on the web at `http://www.itu.dk/~anderssn/lcml.tgz`. □

Example 3.5 (Reflexivity) $\cdot;\cdot;\cdot \Longrightarrow A\ [\cdot;\cdot]\!\!\rightarrowtail A$ is derivable in the sequent calculus.

$A[\cdot;\cdot];\cdot;\cdot \Longrightarrow \cdot;\cdot$	by id
$A[\cdot;\cdot];\cdot;A \Longrightarrow A$	by identity
$A[\cdot;\cdot];\cdot;\cdot \Longrightarrow A$	by reflect
$\cdot;\cdot;\cdot \Longrightarrow A\ [\cdot;\cdot]\!\!\rightarrowtail A$	by $\Box\!\!\rightarrowtail$ R

4 Modalities for Truth and Validity

Our version of linear contextual modal logic is equivalent to one that internalizes the judgments for truth and validity by introducing two modal operators, ! and $[\Gamma;\Delta]$, ultimately justifying the modal nature of the logic. The ! is the standard modality that enables intuitionistic reasoning in linear logic, and $[\Gamma;\Delta]$ is the contextual modal operator that internalizes validity with respect to the intuitionistic context Γ and the linear context Δ.

We obtain the internalized version of linear contextual modal logic by replacing the intuitionistic implication connective and its defining rules by !, !I, and $!\mathsf{E}^x$

$$\frac{\Psi;\Gamma;\cdot \vdash A\ \mathsf{avail}}{\Psi;\Gamma;\cdot \vdash\ !A\ \mathsf{avail}}\,!\mathsf{I}$$

$$\frac{\Psi;\Gamma;\Delta_1 \vdash\ !A\ \mathsf{avail} \quad \Psi;(\Gamma,x{:}A\ \mathsf{true});\Delta_2 \vdash C\ \mathsf{avail}}{\Psi;\Gamma;(\Delta_1,\Delta_2) \vdash C\ \mathsf{avail}}\,!\mathsf{E}^x$$

and the connective for contextual validity and its defining rules by $[\Gamma;\Delta]$, $\Box$I, and $\Box\mathsf{E}^u$. We refer to the resulting logic as LCML^i.

$$\frac{\Psi;\Gamma';\Delta' \vdash A\ \mathsf{avail}}{\Psi;\Gamma;\cdot \vdash [\Gamma';\Delta']A\ \mathsf{avail}}\,\Box\mathsf{I}$$

$$\frac{\begin{array}{l}\Psi;\Gamma;\Delta_1 \vdash [\Gamma';\Delta']A\ \mathsf{avail}\\ (\Psi,u::A\ \mathsf{valid}[\Gamma';\Delta']);\Gamma;\Delta_2 \vdash C\ \mathsf{avail}\end{array}}{\Psi;\Gamma;(\Delta_1,\Delta_2) \vdash C\ \mathsf{avail}}\,\Box\mathsf{E}^u$$

Inspired by the double negation translation and [MS07], we make the relation between ! and $\to$, and $[\Gamma;\Delta]$ and $[\Gamma;\Delta]\!\!\rightarrowtail$ precise. We show that both formulations of the rules prove the same theorems via the following translation and its inverse. The injection $\cdot^i$ translates propositions that use modalities into a version with only implications and $\cdot^e$ is an embedding in the reverse direction. The former translation is the more interesting of the two.

$$\begin{array}{rcl}
P^i & = & P\\
(A \multimap B)^i & = & A^i \multimap B^i\\
(!A)^i & = & (A^i \to p) \multimap p\\
([\Gamma;\Delta]A)^i & = & (A^i\ [\Gamma^i;\Delta^i]\!\!\rightarrowtail p) \multimap p\\
\cdot^i & = & \cdot\\
(\Delta, x{:}A\ \mathsf{avail})^i & = & \Delta^i, x : A^i\ \mathsf{avail}
\end{array}$$

$$
\begin{array}{rcl}
(\Gamma, x{:}A\ \mathsf{true})^i & = & \Gamma^i, x : A^i\ \mathsf{true} \\
(\Psi, u :: A\ \mathsf{valid}[\Gamma;\Delta])^i & = & \Psi^i, u :: A^i\ \mathsf{valid}[\Gamma^i;\Delta^i]
\end{array}
$$

The definition of the translation is parametric in p, which means that for any given translation p can be instantiated by any formula as desired.

Theorem 4.1 (Soundness) *If* $\Psi;\Gamma;\Delta \vdash A\ \mathsf{avail}$ *in LCML*^{i} *then* $\Psi^i;\Gamma^i;\Delta^i \vdash A^i\ \mathsf{avail}$ *in LCML.*

Proof: by induction on the derivation of A. □

The inverse of this translation is a straightforward embedding $\cdot^e$. We only mention the two interesting cases; $A \to B$ is translated into $!A^e \multimap B^e$ and $A\ [\Gamma;\Delta]\!\!\rightarrow B$ into $[\Gamma^e;\Delta^e]A^e \multimap B^e$. The completeness result follows by a simple inductive argument.

Theorem 4.2 (Completeness) *If* $\Psi;\Gamma;\Delta \vdash A\ \mathsf{avail}$ *in LCML then* $\Psi^e;\Gamma^e;\Delta^e \vdash A^e$ *in LCML*^{i}.

Therefore, it is just a question of convenience if one prefers to work with explicit modalities or not. In addition, one can in LCML^i for example show that the introduction and elimination rules of the defined connectives are admissible. We give one example.

Example 4.3 (Admissibility of $\to$ I) We can show that if $\Psi;\Gamma, A\ \mathsf{true};\cdot \vdash B\ \mathsf{avail}$ then $\Psi;\Gamma;\cdot \vdash !A \multimap B\ \mathsf{avail}$. The proof is direct:

$$
\begin{array}{ll}
\Psi;\Gamma;x{:}!A\ \mathsf{avail} \vdash !A\ \mathsf{avail} & \text{by } \mathsf{lhyp}_x \\
\Psi;\Gamma, A\ \mathsf{true};\cdot \vdash B\ \mathsf{avail} & \text{given} \\
\Psi;\Gamma;x{:}!A\ \mathsf{avail} \vdash B\ \mathsf{avail} & \text{by } !\mathsf{E} \\
\Psi;\Gamma;\cdot \vdash !A \multimap B\ \mathsf{avail} & \text{by } \multimap \mathsf{I}
\end{array}
$$

The next example solidifies the relation between logic variables and contextual modal hypotheses. It is taken from a paper by Dowek et al. [DHKP96] describing higher-order pattern unification. Although higher-order unification is undecidable, the higher-order pattern fragment characterizes a fragment that can be reduced to a first-order unification problem. One of the crucial steps is to lower logic variables of higher type to atomic type. For example, a logic variable of type $A \to B \to C$ declared in the empty context should be instantiated by a logic variable of type C in the context containing A and B.

Example 4.4 (Lowering and Raising) We show that lowering and raising for one (linear or intuitionistic) hypothesis is justified. By induction, it also works for many.

1. $\Psi;\Gamma;\Delta \vdash [\Gamma';(\Delta', x : A\ \mathsf{avail})]C\ \mathsf{avail}$ if and only if $\Psi;\Gamma;\Delta \vdash [\Gamma';\Delta'](A \multimap C)\ \mathsf{avail}$.
2. $\Psi;\Gamma;\Delta \vdash [(\Gamma', x : A\ \mathsf{true});\cdot]C\ \mathsf{avail}$ if and only if $\Psi;\Gamma;\Delta \vdash [\Gamma';\cdot](!A \multimap C)\ \mathsf{avail}$.

The proofs are by a straightforward application of the rules. We only show the left to right direction of 1., the remaining three derivations follow the same schema and are omitted in the interest of space. Let $\Psi' = \Psi, u :: C\ \mathsf{valid}[\Gamma';(\Delta', x{:}A\ \mathsf{avail})]$

$$
\begin{array}{ll}
\Psi';\Gamma';(\Delta', x : A\ \mathsf{avail}) \vdash \Gamma';(\Delta', x : A\ \mathsf{avail}) & \\
\qquad \text{by Theorems 3.1 2. and 3.4 2.} & \\
\Psi';\Gamma';(\Delta', x : A\ \mathsf{avail}) \vdash C\ \mathsf{avail} & \text{by } \mathsf{mhyp}_u \\
\Psi';\Gamma';\Delta' \vdash A \multimap C\ \mathsf{avail} & \text{by } \multimap \mathsf{I}^x \\
\Psi';\Gamma;\Delta \vdash [\Gamma';\Delta'](A \multimap C)\ \mathsf{avail} & \text{by } \Box\mathsf{I} \\
\Psi;\Gamma;\Delta \vdash [\Gamma';(\Delta', x{:}A)]C\ \mathsf{avail} & \text{given} \\
\Psi;\Gamma;\Delta \vdash [\Gamma';\Delta'](A \multimap C)\ \mathsf{avail} & \text{by } \Box\mathsf{E}^u
\end{array}
$$

We remark that the left to right direction of 2. relies on the admissibilty of $\to$ I from Example 4.3.

Example 4.5 (Modal laws) The following propositions are tautologies in LCML^i:

1. $!A \multimap A$
2. $!(A \multimap B) \multimap !A \multimap !B$
3. $!A \multimap !!A$
4. $[\cdot;\cdot]A \multimap A$
5. $[\Gamma;\Delta_1](A \multimap B) \multimap [\Gamma;\Delta_2]A \multimap [\Gamma;(\Delta_1,\Delta_2)]B$
6. $[\Gamma;\Delta]A \multimap [\Gamma';\cdot][\Gamma;\Delta]A$

The first three facts are well known from linear logic. The fourth follows from Example 3.5 and Theorems 3.4 1. and 4.2. We omit the proofs in the interest of space.

The following propositions are not provable in LCML^i:

1. $[\Gamma;\Delta]A \multimap A$
2. $[\Gamma;\Delta](A \multimap B) \multimap [\Gamma;\Delta]A \multimap [\Gamma;\Delta]B$
3. $[\Gamma;\Delta]A \multimap [\Gamma;\Delta][\Gamma;\Delta]A$

We therefore conclude that ! satisfies the laws of S5, and so does the box modality $[\Gamma;\Delta]$ but only if the choice of operator respects the laws of linear logic. A further analysis of the modal properties of linear contextual modal logic is beyond the scope of this paper.

In summary, the internalized version presented here is equivalent to the non-internalized version of linear contextual modal logic discussed in the previous two sections. Moreover, it is also a conservative extension of intuitionistic contextual modal logic [NPP08] (ICML) where we need to embed the non-linear part into LCML by introducing spurious empty linear contexts into the modal operators: We define A^n as follows.

$$
\begin{array}{rcl}
A^n & = & \begin{cases} P & \text{if } A = P \\ B^n \to C^n & \text{if } A = B \to C \\ [\Gamma^n;\cdot]B^n & \text{if } A = [\Gamma]B \end{cases} \\
\Gamma^n & = & \begin{cases} \cdot & \text{if } \Gamma = \cdot \\ \Gamma'^n, x{:}A^n\ \mathsf{true} & \text{if } \Gamma = \Gamma', x{:}A\ \mathsf{true} \end{cases} \\
\Psi^n & = & \begin{cases} \cdot & \text{if } \Psi = \cdot \\ \Psi'^n, u :: A^n\ \mathsf{valid}[\Gamma^n;\cdot] & \text{if } \Psi = \Psi', u :: A\ \mathsf{valid}[\Gamma] \end{cases}
\end{array}
$$

Finally, we prove for all A (in ICML) and all Γ' (in ICML) the following by the way of canonical derivations and an easy mutual structural induction.

Theorem 4.6 (Conservative Extension)

1. *If* $\Psi^n; \Gamma^n; \cdot \vdash A^n$ avail *in LCML*^{i} *then* $\Psi; \Gamma \vdash A$ true *in ICML.*
2. *If* $\Psi^n; \Gamma^n; \cdot \vdash \Gamma'^n; \cdot$ *in LCML*^{i} *then* $\Psi; \Gamma \vdash \Gamma'$ *in ICML.*

In the remainder of the paper, we will only work with the non-internalized version since proof terms for LCML^i do not have canonical forms [NPP08].

5 Linear Contextual Modal Type Theory

Constructive logics can be seen as type theories via the well-known Curry-Howard correspondence. Perhaps not very surprisingly, linear contextual modal logic gives rise to linear contextual modal type theory once augmented by proof terms. Traditionally, proof terms are in one-to-one correspondence with the derivations. It is easy to see that the proof terms that we introduce here satisfy this property as well. The situation is less clear for extensions involving, for example, $\top$ as we will discuss in Section 6.

Our version of linear contextual modal type theory (based on the rules introduced in Section 2) is perhaps less interesting for programming but it is the foundation for logical frameworks concerned with the representation of resource based deductive systems, such as LLF [CP96] and CLF [CPWW02], and permits us to study the very nature of logic variables.

The proof term assignment for linear contextual modal logic is based on the rules from Figure 1. For convenience, we drop the judgments of avail, true, and valid, and express the rules in a more traditional form as specified in Figure 3.

Proof Terms	t	$::= x \mid u[\sigma] \mid \lambda x{:}A.\ t \mid t_1\ t_2$ $\mid \widehat{\lambda} x{:}A\,.\,t \mid t_1 \hat{}\, t_2$ $\mid \lambda^{[\Gamma;\Delta]} u : A\,.\,t \mid t_1 \overset{\Gamma;\Delta}{\Box} t_2$
Substitutions	σ	$::= \cdot \mid \sigma, t/x$
Contexts	Γ, Δ	$::= \cdot \mid \Gamma, x : A$
Modal Contexts	Ψ	$::= \cdot \mid \Psi, u :: A[\Gamma; \Delta]$

We distinguish between variables x (intuitionistic or linear) and $u[\sigma]$ (contextual modal), where the contextual modal variables should be thought of as logic variables and σ as a delayed substitution. Every implication (linear, intuitionistic, and contextual modal) provides an abstraction and application term. Intuitionistic and linear application are standard. Therefore we comment only on the two contexts above the contextual modal application $t_1 \overset{\Gamma;\Delta}{\Box} t_2$: all declarations in Γ and Δ are binding occurrences of those intuitionistic and linear declarations that define the *world* the argument term t_2 lives in. The Γ and Δ should be viewed as iterated abstractions of t_2 subject to α-conversion. Their order is therefore important — writing them as contexts is merely a convenient shorthand.

5.1 Substitutions

Linear contextual type theory provides two notions of substitution application. One is for the simultaneous substitutions that witness derivations of the judgment : $\Psi; \Gamma; \Delta \vdash \sigma : \Gamma'; \Delta'$, and the other is for single point substitutions that instantiate contextual modal variables. Recall that contextual modal variables model logic variables.

Definition 5.1 (Simultaneous Substitutions)

$$
\begin{array}{lcll}
[\sigma_1, M/x, \sigma_2](x) & = & x & \\
[\sigma](u[\tau]) & = & u([\sigma]\tau) & \\
[\sigma](\lambda x{:}A.\ t) & = & \lambda x{:}A.\ [\sigma, x/x](t) & \\
[\sigma](t_1\ t_2) & = & [\sigma](t_1)\ [\sigma](t_2) & \\
[\sigma](\widehat{\lambda} x{:}A\,.\,t) & = & \widehat{\lambda} x{:}A\,.\,[\sigma, x/x](t) & \\
[\sigma](t_1 \hat{}\, t_2) & = & [\sigma](t_1) \hat{}\, [\sigma](t_2) & \\
[\sigma](\lambda^{[\Gamma;\Delta]} u{:}A\,.\,t) & = & \lambda^{[\Gamma;\Delta]} u{:}A\,.\,[\sigma](t) & (*) \\
[\sigma](t_1 \overset{\Gamma;\Delta}{\Box} t_2) & = & [\sigma](t_1) \overset{\Gamma;\Delta}{\Box} t_2 & (**) \\
[\sigma](\cdot) & = & \cdot & \\
[\sigma](\sigma', t/x) & = & [\sigma](\sigma'), [\sigma](t)/x &
\end{array}
$$

The definition of substitution application is largely standard, we comment only on the two equations marked by stars. In $(*)$ modal contextual variables never occur in the domain of σ. Therefore σ does not need to be extended by x/x as in the other two abstraction cases. In (**), the argument to a contextual modal term will always live in a world different from the domain of σ, and thus must not be applied to t_2.

Theorem 5.2 (Substitution)

1. *If* $\Psi; \Gamma; \Delta \vdash \sigma : \Gamma'; \Delta'$ *and* $\Psi; \Gamma'; \Delta' \vdash t : A$ *then* $\Psi; \Gamma; \Delta \vdash [\sigma](t) : A$.
2. *If* $\Psi; \Gamma; \Delta \vdash \sigma : \Gamma'; \Delta'$ *and* $\Psi; \Gamma'; \Delta' \vdash \sigma' : \Gamma''; \Delta''$ *then* $\Psi; \Gamma; \Delta \vdash [\sigma](\sigma') : \Gamma''; \Delta''$.

Proof: by a mutual structural induction on the derivation of t in 1. and σ' in 2. □

More interesting is the definition of substitution for contextual modal variables (or logic variables). Here, it is sufficient to define only a single point substitution, which corresponds to the instantiation of several instances of one particular logic variable. The presence of linearity does not

$$\frac{}{\Psi; (\Gamma, x{:}A); \cdot \vdash x : A}\ \mathsf{hyp}_x \quad \frac{}{\Psi; \Gamma; x{:}A \vdash x : A}\ \mathsf{lhyp}_x \quad \frac{(\Psi, u{::}A[\Gamma'; \Delta']); \Gamma; \Delta \vdash \sigma : \Gamma'; \Delta'}{(\Psi, u{::}A[\Gamma'; \Delta']); \Gamma; \Delta \vdash u[\sigma] : A}\ \mathsf{mhyp}_u$$

$$\frac{\Psi; (\Gamma, x{:}A); \Delta \vdash t : B}{\Psi; \Gamma; \Delta \vdash \lambda x{:}A.\, t : A \to B} \to \mathsf{I}^x \quad \frac{\Psi; \Gamma; \Delta \vdash t_1 : A \to B \quad \Psi; \Gamma; \cdot \vdash t_2 : A}{\Psi; \Gamma; \Delta \vdash t_1\ t_2 : B} \to \mathsf{E}$$

$$\frac{\Psi; \Gamma; (\Delta, x{:}A) \vdash t : B}{\Psi; \Gamma; \Delta \vdash \widehat{\lambda} x : A\,.\, t : A \multimap B} \multimap \mathsf{I}^x \quad \frac{\Psi; \Gamma; \Delta_1 \vdash t_1 : A \multimap B \quad \Psi; \Gamma; \Delta_2 \vdash t_2 : A}{\Psi; \Gamma; (\Delta_1, \Delta_2) \vdash t_1 \,\hat{}\, t_2 : B} \multimap \mathsf{E}$$

$$\frac{(\Psi, u{::}A[\Gamma'; \Delta']); \Gamma; \Delta \vdash t : B}{\Psi; \Gamma; \Delta \vdash \lambda^{[\Gamma'; \Delta']} u : A\,.\, t : A\ [\Gamma'; \Delta'] \!\mapsto B} \Box\!\!\mapsto \mathsf{I}^u \quad \frac{\Psi; \Gamma; \Delta \vdash t_1 : A\ [\Gamma'; \Delta'] \!\mapsto B \quad \Psi; \Gamma'; \Delta' \vdash t_2 : A}{\Psi; \Gamma; \Delta \vdash t_1 \overset{\Gamma'; \Delta'}{\Box} t_2 : B} \Box\!\!\mapsto \mathsf{E}$$

$$\frac{}{\Psi; \Gamma; \cdot \vdash \cdot : \cdot; \cdot} \quad \frac{\Psi; \Gamma; \Delta \vdash \sigma : \Gamma'; \cdot \quad \Psi; \Gamma; \cdot \vdash t : A}{\Psi; \Gamma; \Delta \vdash \sigma, t/x : (\Gamma', x{:}A); \cdot} \quad \frac{\Psi; \Gamma; \Delta_1 \vdash \sigma : \Gamma'; \Delta' \quad \Psi; \Gamma; \Delta_2 \vdash t : A}{\Psi; \Gamma; (\Delta_1, \Delta_2) \vdash \sigma, t/x : \Gamma'; (\Delta', x{:}A)}$$

Figure 3. Type Theory LCMTT

lead to any surprises, and the definition of substitution is reminiscent to that described in [NPP08].

Definition 5.3 (Substitution for Contextual Modal Variables) *A substitution for a contextual modal variable is a single point substitution that substitutes a term t', declared in world $\Gamma; \Delta$ for the contextual modal variable u in t. Application written in short as $[\![(\Gamma; \Delta).t'/u]\!](t)$. The rules defining this judgment are depicted in Figure 4.*

Contextual modal substitution application traverses the entire structure of the term and substitutes t' into each occurrence of u. The only two interesting cases are Equations (1) and (2). The former describes the actual substitution step: If we substitute t' for u in $u[\sigma]$, we follow the following steps. First, we replace all occurrences of the contextual modal variable u in the delayed simultaneous substitution σ, written as $\sigma' = [\![(\Gamma'; \Delta').t'/u]\!](\sigma)$. However, before σ can be carried out, its domain must be renamed to match the variables in $\Gamma'; \Delta'$ for which we write $\sigma/(\Gamma'; \Delta')$.

Theorem 5.4 (Contextual Modal Substitution)

1. *If $\Psi; \Gamma'; \Delta' \vdash t' : C$ and $(\Psi, u{::}C[\Gamma'; \Delta']); \Gamma; \Delta \vdash t : A$ then $\Psi; \Gamma; \Delta \vdash [\![(\Gamma'; \Delta').t'/u]\!](t) : A$.*

2. *If $\Psi; \Gamma'; \Delta' \vdash t' : C$ and $(\Psi, u{::}C[\Gamma'; \Delta']); \Gamma; \Delta \vdash \sigma : \Gamma''; \Delta''$ then $\Psi; \Gamma; \Delta \vdash [\![(\Gamma'; \Delta').t'/u]\!](\sigma) : \Gamma''; \Delta''$.*

Proof: by mutual structural induction on t in 1. and σ in 2. □

5.2 Equational Theory

Based on the definition of these two notions of substitution application, every connective in linear contextual modal type theory gives rise to a β- and an η-rule, justifying our choice of definitional equality denoted by $\equiv$.

As a preliminary definition, we define the identity substitution on arbitrary contexts.

$$\begin{array}{lcl} \mathsf{id}_{(\cdot;\cdot)} & = & \cdot \\ \mathsf{id}_{((\Gamma,x:A);\cdot)} & = & \mathsf{id}_{(\Gamma;\cdot)}, x/x \\ \mathsf{id}_{(\Gamma;(\Delta,x:A))} & = & \mathsf{id}_{(\Gamma;\Delta)}, x/x \end{array}$$

Theorem 5.5 (Identity) *For all contexts Ψ, Γ, and Δ,*

$$\Psi; \Gamma; \Delta \vdash \mathsf{id}_{(\Gamma;\Delta)} : \Gamma; \Delta$$

Our choice of definitional equality $\equiv$ is defined as the symmetric, reflexive, and transitive closure of the following β-reduction and η-expansion rules defining $\Longrightarrow$. For the fragment motivated by linear and hypothetical judgments, the rules are standard.

$$\begin{array}{rcl} (\widehat{\lambda} x{:}A\,.\,t_1) \,\hat{}\, t_2 : B & \Longrightarrow & [t_2/x]t_1 \\ t : A \multimap B & \Longrightarrow & \widehat{\lambda} x : A\,.\, t \,\hat{}\, x \\ (\lambda x{:}A.\, t_1)\ t_2 : B & \Longrightarrow & [t_2/x]t_1 \\ t : A \to B & \Longrightarrow & \lambda x : A.\ t\ x \end{array}$$

The fragment that corresponds to the contextual judgment is less standard but nevertheless intuitive.

$$\begin{array}{rcl} (\lambda^{[\Gamma;\Delta]} u{:}A\,.\,t_1) \overset{\Gamma;\Delta}{\Box} t_2 : B & \Longrightarrow & [\![(\Gamma; \Delta).t_2/u]\!]t_1 \\ t : A\ [\Gamma; \Delta] \!\mapsto B & \Longrightarrow & \lambda^{[\Gamma;\Delta]} u : A\,.\, t \overset{\Gamma;\Delta}{\Box} u[\mathsf{id}_{(\Gamma;\Delta)}] \end{array}$$

$$
\begin{array}{rcll}
[\![(\Gamma';\Delta').t'/u]\!](x) & = & x & \\
[\![(\Gamma';\Delta').t'/u]\!](u[\sigma]) & = & t'[([\![(\Gamma';\Delta').t'/u]\!](\sigma))/(\Gamma';\Delta')] & (1)\\
[\![(\Gamma';\Delta').t'/u]\!](v[\sigma]) & = & v[[\![(\Gamma';\Delta').t'/u]\!](\sigma)] \quad \text{where } u \neq v & (2)\\
[\![(\Gamma';\Delta').t'/u]\!](\lambda x{:}A.\ t) & = & \lambda x{:}A.\ [\![(\Gamma';\Delta').t'/u]\!](t) & \\
[\![(\Gamma';\Delta').t'/u]\!](t_1\ t_2) & = & [\![(\Gamma';\Delta').t'/u]\!](t_1)\ [\![(\Gamma';\Delta').t'/u]\!](t_2) & \\
[\![(\Gamma';\Delta').t'/u]\!](\widehat{\lambda} x{:}A\,.\ t) & = & \widehat{\lambda} x{:}A\,.\ [\![(\Gamma';\Delta').t'/u]\!](t) & \\
[\![(\Gamma';\Delta').t'/u]\!](t_1 \mathbin{\widehat{}} t_2) & = & [\![(\Gamma';\Delta').t'/u]\!](t_1) \mathbin{\widehat{}} [\![(\Gamma';\Delta').t'/u]\!](t_2) & \\
[\![(\Gamma';\Delta').t'/u]\!](\lambda^{[\Gamma;\Delta]} x{:}A\,.\ t) & = & \lambda^{[\Gamma;\Delta]} x{:}A\,.\ [\![(\Gamma';\Delta').t'/u]\!](t) & \\
[\![(\Gamma';\Delta').t'/u]\!](t_1 \overset{\Gamma;\Delta}{\square} t_2) & = & [\![(\Gamma';\Delta').t'/u]\!](t_1) \overset{\Gamma;\Delta}{\square} [\![(\Gamma';\Delta').t'/u]\!](t_2)) & \\
[\![(\Gamma';\Delta').t'/u]\!](\cdot) & = & \cdot & \\
[\![(\Gamma';\Delta').t'/u]\!](\sigma, t/x) & = & [\![(\Gamma';\Delta').t'/u]\!](\sigma), [\![(\Gamma';\Delta').t'/u]\!](t)/x &
\end{array}
$$

Figure 4. Substitution for Contextual Modal Variables

We omit the straightforward definition of the congruence rules, which also extends to substitutions. Every equivalence class is represented by a β-normal η-long form, which we also call canonical forms. The idea of equivalence and canonical form generalizes directly to simultaneous substitutions.

5.3 Meta Theory

We can reassure the reader that linear contextual modal type theory behaves as expected. First we show that reduction preserves types.

Theorem 5.6 (Subject Reduction and Expansion) *If* $\Psi;\Gamma;\Delta \vdash t : A$ *and* $t : A \equiv t'$ *then* $\Psi;\Gamma;\Delta \vdash t' : A$.

Proof: To see that β and η rules preserve types, one returns to the interpretation of the typing derivation as a logic derivation. Each β rule is justified by the the argument of local soundness: Introducing a connective followed immediately by elimination is justified by the respective substitution principle. Each η rule is justified by the argument of local completeness. Any derivation of a non-atomic formula followed by an elimination, and reintroduction of the main connective ends in exactly the same derivation of the non-atomic formula we started with. All rules defining $\equiv$ preserve types as well. □

Theorem 5.7 (Canonical forms)

1. *If* $\Psi;\Gamma;\Delta \vdash t : A$ *then there exists a unique term* t' *in canonical form, such that* $\Psi;\Gamma;\Delta \vdash t' : A$ *and* $t \equiv t'$.
2. *If* $\Psi;\Gamma;\Delta \vdash \sigma : \Gamma';\Delta'$ *then there exists a unique substitution* σ' *in canonical form, such that* $\Psi;\Gamma;\Delta \vdash \sigma' : \Gamma';\Delta'$ *and* $\sigma \equiv \sigma'$.

Proof: This theorem follows from the fact that canonical form are in one-to-one correspondence with cut-free derivations in the sequent calculus. □

6 Applications

We believe that linear contextual modal type theory has applications that go beyond the study of the nature of logic variables that we used to motivate this work. For example, we speculate that this logic will be instrumental to help scale the idea of staged computation [DP01] to programming languages with linear types, such as Clean. Furthermore linear contextual modal logic is the logical foundation of the meta linear logical framework MLLF developed by McCreight et al. [MS04]. Both applications, however, lie outside the scope of this paper. We report therefore on our experiences when we tried to add the linear additive connective $\top$ to the logic. This connective is present in Lolli, the linear logical framework LLF, the concurrent logical framework CLF, and virtually every theorem prover built for linear logic [Tam94].

The $\top$ connective in linear logic is the proposition that is available in any context. It can thus consume an arbitrary number of linear resources. It is defined by the following rule.

$$\frac{}{\Psi;\Gamma;\Delta \vdash \top\ \mathsf{avail}}\ \top\mathsf{I}$$

Extending the linear contextual modal logic with $\top$ is without surprises. All proofs above trivially extend for both the natural deduction calculus from Section 2 and the sequent calculus from Section 3.

Interesting insights however can be gained if one tries to add $\top$ to the type theory described in Section 5. Previous presentations of linear type theory have given two possible

proof terms for $\top$, one that is labeled by the names of the resources from Δ it consumes [Bie94] and another that is unlabeled [CP96].

Labeled Unit. Bierman introduces $\langle\rangle_\Delta$ as the proof term for $\top$ with the following typing rule.

$$\frac{}{\Psi;\Gamma;\Delta \vdash \langle\rangle_\Delta : \top}\ \top\mathsf{I}$$

This version of the proof term is not problematic at all. However, substitution application must be generalized to accommodate substitutions on context Δ. Assuming $\Psi;\Gamma';\Delta' \vdash \sigma : \Gamma;\Delta$ we define $[\sigma]\langle\rangle_\Delta = \langle\rangle_{\Delta'}$. The equational theory is subsequently extended with the following η-rule: If $\Psi;\Gamma;\Delta \vdash t : \top$ then $t : \top \Longrightarrow \langle\rangle_\Delta$. There is no β-rule since $\top$ has no elimination/left rule.

The choice of the labeled unit $\langle\rangle_\Delta$ comes also with a few perks: we maintain the one-to-one correspondence between proofs and proof terms. Theorems 5.6 and 5.7 extend seamlessly.

Unlabeled unit. The unlabeled unit $\langle\rangle$ is the proof term of choice for most logical frameworks such as LLF and CLF.

$$\frac{}{\Psi;\Gamma;\Delta \vdash \langle\rangle : \top}\ \top\mathsf{I}$$

The advantage of using $\langle\rangle$ as opposed to $\langle\rangle_\Delta$ is that in any linear logic proof with one or more applications to $\top\mathsf{I}$ we usually do not care which $\top$ consumes what resources. In the case of ambiguity, we happily identify those proofs, which leads to better proof search behavior and also to more succinct encodings of formal systems. For example, there is only one proof term $\widehat{\lambda}x{:}A\,.\,\widehat{\lambda}y{:}\top \multimap \top \multimap B\,.\,y\,\widehat{}\,\langle\rangle\,\widehat{}\,\langle\rangle$ that witnesses the derivation of the proposition $A \multimap (\top \multimap \top \multimap B) \multimap B$ instead of two, if we were to use the labeled unit.

In the context of linear contextual modal type theory, however, the unlabeled unit $\langle\rangle$ causes trouble; consider e.g. the unification problem discussed in Section 1

$$c\,\widehat{}\,\langle\rangle\,\widehat{}\,(F\,\widehat{}\,x) = c\,\widehat{}\,\langle\rangle\,\widehat{}\,(d\,\widehat{}\,G)$$

where simple decomposition fails, i.e. even though the equation has a solution we cannot recurse on subproblems in a type-correct manner.

An alternative way to add $\top$ to linear contextual modal logic is to require that the linear context in $\top\mathsf{I}$ is always empty and to encode the fact $\top$ may consume arbitrary resources by flagging the judgment of derivability to tell whether a derivation may be weakened. This idea is reminiscent of the resource management logics used to express algorithmic type checking in LLF [CHP96]. In this view the $\top\mathsf{I}$ rules above simply enforces that all weakening steps happen implicitly in the leaves of the derivation in one-to-one correspondence with the labeled unit $\langle\rangle_\Delta$. If we instead move the weakening as far towards the corresponding binders as possible, we regain the one-to-one correspondence between proof terms with the unlabeled unit $\langle\rangle$ and derivations. Returning to the example, we can no longer distinguish derivations based on which $\top$ consumes which resource. We call this bit the $\top$-flag, and our judgments thus looks like $\Psi;\Gamma;\Delta \vdash_f t : A$ where f is the $\top$-flag.

Now the cause of the problem is apparent: subject reduction no longer holds in the sense that the $\top$-flag might change. Similarly, the hypothetical and categorical cuts are not admissible in the sequent calculus when we expect the $\top$-flag of the conclusion to be determined immediately by the premises. And since logic variable instantiation during unification corresponds to the admissibility of the categorical cut for hereditary instantiation, it explains the problems with the unification equation above.

This observation seems to suggest that the labeled $\langle\rangle_\Delta$ should be preferred over the unlabeled $\langle\rangle$.

7 Conclusion

In this paper we present linear contextual modal type theory as a mathematical explanation for logic variables in the presence of linear types. We prove the existence of canonical forms by means of cut-elimination, whose proof has been formalized and verified in the proof assistant Twelf.

Linear contextual modal type theory provides us with a deep understanding of the interaction between linear type theory and logic variables that are prevalent in many systems based on linear logic, for example, theorem provers and logic programming languages. In particular, it forms the theoretical foundation for our implementation of the concurrent logical framework CLF [CPWW02] in the Celf system [SNS08]. Hereby it explains the problems that arise when one tries to extend it by the additive connective $\top$, how it affects the underlying notion of equality among terms, and furthermore it is an excellent guide for the design of a unification algorithm.

References

[Bie94] G. Bierman. *On Intuitionistic Linear Logic*. PhD thesis, University of Cambridge, 1994.

[CHP96] Iliano Cervesato, Joshua S. Hodas, and Frank Pfenning. Efficient resource management for linear logic proof search. In R. Dyckhoff, H. Herre, and P. Schroeder-Heister, editors, *Proceedings of the 5th International Workshop on Extensions of Logic Programming*, pages 67–81, Leipzig, Germany, March 1996. Springer-Verlag LNAI 1050.

[CP96] Iliano Cervesato and Frank Pfenning. A linear logical framework. In E. Clarke, editor, *Proceedings of the Eleventh Annual Symposium on Logic in Computer Science*, pages 264–275, New Brunswick, New Jersey, July 1996. IEEE Computer Society Press.

[CP97] Iliano Cervesato and Frank Pfenning. Linear higher-order pre-unification. In Glynn Winskel, editor, *Proceedings of the Twelfth Annual Sumposium on Logic in Computer Science (LICS'97)*, pages 422–433, Warsaw, Poland, June 1997. IEEE Computer Society Press.

[CPWW02] Iliano Cervesato, Frank Pfenning, David Walker, and Kevin Watkins. A concurrent logical framework I: Judgments and properties. Technical Report CMU-CS-02-101, Carnegie Mellon University. Department of Computer Science, 2002.

[DHKP96] Gilles Dowek, Thérèse Hardin, Claude Kirchner, and Frank Pfenning. Unification via explicit substitutions: The case of higher-order patterns. In M. Maher, editor, *Proceedings of the Joint International Conference and Symposium on Logic Programming*, pages 259–273, Bonn, Germany, September 1996. MIT Press.

[DP01] Rowan Davies and Frank Pfenning. A modal analysis of staged computation. *Journal of the ACM*, 48(3):555–604, 2001.

[Gir87] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[HM94] Joshua Hodas and Dale Miller. Logic programming in a fragment of intuitionistic linear logic. *Information and Computation*, 110(2):327–365, 1994. A preliminary version appeared in the Proceedings of the Sixth Annual IEEE Symposium on Logic in Computer Science, pages 32–42, Amsterdam, The Netherlands, July 1991.

[MS04] Andrew McCreight and Carsten Schürmann. A meta linear logical framework. In Carsten Schürmann, editor, *Proceedings of Logical Frameworks and Meta Languages*, volume 199, pages 129–147, Cork, Ireland, July 2004.

[MS07] Rasmus Møgelberg and Alex Simpson. Relational parametricity for computational effects. In *Proceedings of the IEEE Symposium on Logic in Computer Science*. IEEE Computer Society, 2007.

[NPP08] Aleksandar Nanevski, Frank Pfenning, and Brigitte Pientka. Contextual modal type theory. *Transactions on Computational Logic*, 9(3):Article 23, 1–49, June 2008.

[PS99] Frank Pfenning and Carsten Schürmann. System description: Twelf — a meta-logical framework for deductive systems. In H. Ganzinger, editor, *Proceedings of the 16th International Conference on Automated Deduction (CADE-16)*, pages 202–206, Trento, Italy, July 1999. Springer-Verlag LNAI 1632.

[Rey02] John C. Reynolds. Separation logic: A logic for shared mutable data structures. In *Proceedings of the IEEE Symposium on Logic in Computer Science*, pages 55–74. IEEE Computer Society, 2002.

[SNS08] Anders Schack-Nielsen and Carsten Schürmann. System description: Celf – a logical framework for deductive and concurrent systems. In Alessandro Armando, Peter Baumgartner, and Gilles Dowek, editors, *International Joint Conference on Automated Reasoning (IJCAR)*, pages 320–331, Sydney, Australia, 2008.

[Tam94] Tanel Tammet. Proof strategies in linear logic. *Journal of Automated Reasoning*, 12:273–304, 1994.