Uniform Atomic Ordered Linear Logic

A meta-circular interpreter for Olli

Jeff Polakow
Awake Security
USA

ABSTRACT

The original presentation of uniform ordered linear logic does not permit direct implementation (in Olli) as a meta-circular interpreter. We explain the difficulty and present a new formulation of the logic, called uniform *atomic* ordered linear logic which does allow a direct transcription into a meta-circular interpreter. We prove the new system sound and complete with respect to the old; and we exhibit a meta-circular interpreter for Olli.

ACM Reference format:
Jeff Polakow. 2017. Uniform Atomic Ordered Linear Logic. In *Proceedings of LFMTP '17, Oxford, United Kingdom, September 8, 2017,* 9 pages.
https://doi.org/10.1145/3130261.3130264

1 INTRODUCTION

Ordered linear logic (OLL) [10] is a conservative extension of intuitionistic linear logic (ILL) with an ordered context in which none of the standard three structural rules apply; i.e., ordered hypotheses cannot be exchanged, copied, or deleted. As with ILL, the uniform (focussed and goal-directed) [1, 8] fragment of OLL can be turned into a logic programming language (called Olli) [9], as well as serve as the basis for a logical framework [11].

However, unlike uniform ILL [6], the presentation of uniform OLL (UOLL) in [10] does not allow full logical compilation [2]. While unrestricted and linear hypotheses can be residuated, or logically compiled into new goal formulae, ordered hypotheses were resistant to such transformation. In addition to being an aesthetic wart on the logic, the inability to residuate ordered hypotheses prevents constructing a meta-circular interpreter for Olli which directly uses the ordered context of the logic.

The problem with residuating an ordered hypothesis ultimately boils down to the fact that its position in the context– which ordered formulae are on either side of it– is lost when residuating to a new goal formula. In this paper we overcome the position problem by transforming ordered hypotheses into linear hypotheses whose positions are explicitly represented by atomic placeholder predicates which will occur as ordered hypotheses. The resulting system, uniform atomic ordered linear logic (UAOLL), has a demoted ordered context containing only placeholders.

LFMTP '17, September 8, 2017, Oxford, United Kingdom

ACM ISBN 978-1-4503-5374-8/17/09…$15.00
https://doi.org/10.1145/3130261.3130264

In this paper, we present UAOLL and prove soundness and completeness with respect to UOLL. Since there are no ordered formulae to residuate, we can achieve full logical compilation for UAOLL by using the residuation previously shown for UOLL. Finally, we transcribe the inference rules for residuated UAOLL into Olli to get a meta-circular interpreter.

The rest of this paper proceeds as follows: section 2 gives a brief review of OLL; section 3 presents UOLL; section 4 reviews residuation for unrestricted and linear UOLL hypotheses; section 5 explains why residuation does not work for ordered hypotheses; section 6 describes our solution for not needing to residuate ordered hypotheses; section 7 presents UAOLL; section 8 gives a proof of the correctness of UAOLL with respect to UOLL; section 9 adds residuation to UAOLL; section 10 presents our meta-circular interpreter for Olli; section 11 offers some conclusions and thoughts on further work.

2 ORDERED LINEAR LOGIC

In this section we present a brief overview and reconstruction of OLL; see [10] for an in depth presentation of this material. This section is included for context as the technical starting point for UAOLL (presented in section 7) is UOLL (presented in section 3).

2.1 Purely Ordered Fragment

We shall start with a simple purely ordered logic, whose formulas are:

$$A \ ::= \ P \ | \ A \twoheadrightarrow A \ | \ A \rightarrowtail A$$

where P represents atomic formulae, $\twoheadrightarrow$ is right implication and $\rightarrowtail$ is left implication. There are two implications because there are two well-defined places to put a hypothesis, i.e., at either end of the context.

We specify the valid derivations of the logic with the following judgment:

$$\Omega \vdash A$$

where Ω is a context defined by:

$$\Omega \quad ::= \quad \cdot \ | \ \Omega, A$$

We will abuse , to both add a formula to a context, at either end, and to append two contexts together. The logic is then characterized by the following derivation rules:

$$\frac{}{A \vdash A}\,init$$

$$\frac{\Omega, A \vdash B}{\Omega \vdash A \twoheadrightarrow B}\twoheadrightarrow_R \qquad \frac{\Omega_L, B, \Omega_R \vdash C \qquad \Omega_A \vdash A}{\Omega_L, A \twoheadrightarrow B, \Omega_A, \Omega_R \vdash C}\twoheadrightarrow_L$$

$$\frac{A, \Omega \vdash B}{\Omega \vdash A \rightarrowtail B}\rightarrowtail_R \qquad \frac{\Omega_L, B, \Omega_R \vdash C \qquad \Omega_A \vdash A}{\Omega_L, \Omega_A, A \twoheadrightarrow B, \Omega_R \vdash C}\rightarrowtail_L$$

Note that this logic is a reformulation of Lambek calculus [7].

2.2 Adding Linear Hypotheses

We now add linear hypotheses to the logic by adding a linear context, Δ, which will admit exchange, i.e. the order of hypotheses will not matter. We add linear implications to the formula language:

$$A \ ::= \ P \mid A \twoheadrightarrow A \mid A \rightarrowtail A \mid A \multimap A$$

and a linear context, Δ, to the judgement:

$$\Delta ; \Omega \vdash A$$

We use $\bowtie$ to denote non-deterministic context merge; i.e., $\Delta_L \bowtie \Delta_R$ denotes any merging of contexts Δ_L and Δ_R, we will also write $\Delta \bowtie A$ where A is implicitly lifted to a singleton context.

We now have the following derivation rules:

$$\frac{}{\cdot ; A \vdash A}\,init \qquad \frac{\Delta ; \Omega_L, A, \Omega_R \vdash C}{\Delta \bowtie A ; \Omega_L, \Omega_R \vdash C}\,place$$

The *place* rule, when read bottom up, places a linear hypothesis in the ordered context. Note that we cannot move hypotheses from the ordered to the linear context. These two rules show that, in essence, this judgement represents one ordered context where the positions of the linear hypotheses are not fixed until they are actually used.

$$\frac{\Delta ; \Omega, A \vdash B}{\Delta ; \Omega \vdash A \twoheadrightarrow B}\twoheadrightarrow_R \qquad \frac{\Delta ; \Omega_L, B, \Omega_R \vdash C \qquad \Delta_A ; \Omega_A \vdash A}{\Delta \bowtie \Delta_A ; \Omega_L, A \twoheadrightarrow B, \Omega_A, \Omega_R \vdash C}\twoheadrightarrow_L$$

$$\frac{\Delta ; A, \Omega \vdash B}{\Delta ; \Omega \vdash A \rightarrowtail B}\rightarrowtail_R \qquad \frac{\Delta ; \Omega_L, B, \Omega_R \vdash C \qquad \Delta_A ; \Omega_A \vdash A}{\Delta \bowtie \Delta_A ; \Omega_L, \Omega_A, A \rightarrowtail B, \Omega_R \vdash C}\rightarrowtail_L$$

The ordered implication rules now carry along linear hypotheses and the use of $\bowtie$ ensures that order of linear hypotheses doesn't matter.

$$\frac{\Delta, A ; \Omega \vdash B}{\Delta ; \Omega \vdash A \multimap B}\multimap_R \qquad \frac{\Delta ; \Omega_L, B, \Omega_R \vdash C \qquad \Delta_A ; \cdot \vdash A}{\Delta \bowtie \Delta_A ; \Omega_L, A \multimap B, \Omega_R \vdash C}\multimap_L$$

The restriction on the premise of the $\multimap_L$ rule forbids the linear argument from depending on any ordered hypotheses. Note that the left rules only apply to formulae in the ordered context.

2.3 Adding Unrestricted Hypotheses

Unrestricted hypotheses can also be added to the logic in a manner similar to the previous section. The formula language is extended to include unrestricted implications:

$$A \ ::= \ P \mid A \twoheadrightarrow A \mid A \rightarrowtail A \mid A \multimap A \mid A \rightarrow A$$

and an unrestricted context, Γ, is added to the judgement:

$$\Gamma; \Delta; \Omega \vdash A$$

The derivation rules are similarly modified. We only show the extra rules which directly deal with the unrestricted hypotheses:

$$\frac{\Gamma \bowtie A; \Delta; \Omega_L, A, \Omega_R \vdash C}{\Gamma \bowtie A; \Delta; \Omega_L, \Omega_R \vdash C}\,copy$$

$$\frac{\Gamma, A; \Delta; \Omega \vdash B}{\Gamma; \Delta; \Omega \vdash A \rightarrow B}\rightarrow_R \qquad \frac{\Gamma; \Delta; \Omega_L, B, \Omega_R \vdash C \qquad \Gamma; \cdot; \cdot \vdash A}{\Gamma; \Delta; \Omega_L, A \rightarrow B, \Omega_R \vdash C}\rightarrow_L$$

The *copy* rule, when read bottom up, retains a copy of the unrestricted formula as well as places a copy in the ordered context; additionally, the $\rightarrow_L$ rule forbids unrestricted arguments from depending on both linear and ordered hypotheses.

2.4 Full System

OLL consists of the logic in the previous section extended with the full complement of logical connectives, i.e. two multiplicative conjunctions ($\bullet, \circ$) and their unit (1), additive conjunction ($\&$) and its unit ($\top$), additive disjunction ($\oplus$) and its unit (0), a linear modality (¡), an unrestricted modality (!), as well as universal and existential quantifiers over a standard term language. The derivation rules for multiplicative conjunction and unit are as follows:

$$\frac{}{\Gamma; \cdot; \cdot \vdash 1}1_R$$

$$\frac{\Gamma; \Delta_0; \Omega_0 \vdash A_0 \qquad \Gamma; \Delta_1; \Omega_1 \vdash A_1}{\Gamma; \Delta_0 \bowtie \Delta_1; \Omega_0, \Omega_1 \vdash A_0 \bullet A_1}\bullet_R \qquad \frac{\Gamma; \Delta; \Omega_L, A, B, \Omega_R \vdash C}{\Gamma; \Delta; \Omega_L, A \bullet B, \Omega_R \vdash C}\bullet_L$$

Rather than show the rule for $\circ$, we simply note that $A \circ B \equiv B \bullet A$.

We will elide the remaining derivation rules which are not particularly germane to the rest of the paper, and which can be easily recovered from their counterparts in section 3.

3 UNIFORM ORDERED LINEAR LOGIC

In a slight deviation from [8], we shall define uniform proofs to be goal directed, meaning the proof follows the structure of the goal formula (so the *init* rule is restricted to have an atomic goal), and focussed[1], meaning a hypothesis must be completely used before examining another hypothesis. These constraints allow the interpretation of a goal formula as program instructions, hypotheses as named subroutines, and atomic goals as subroutine calls. UOLL is the fragment of OLL for which uniform proofs are complete.

In order to satisfy the constraints of uniform derivations, we will have to restrict the occurrence of some formulae; e.g. there is no uniform derivation of

$$\Gamma; \cdot; A \bullet B \vdash A \bullet B$$

since there is no proof beginning (reading bottom up) with the $\bullet_R$ rule.

It turns out that we only have to restrict the formulae which can be hypotheses. Towards that end we describe the formula language of UOLL in terms of clause formulae and goal formulae (whose definitions are mutually recursive).

We define clause formulae as:

$$\begin{array}{rcll} D & ::= & P & \mid \forall x.D \\ & \mid & \top & \mid D \,\&\, D \\ & \mid & G \twoheadrightarrow D & \mid G \rightarrowtail D \\ & \mid & G \multimap D & \mid G \rightarrow D \end{array}$$

We define goal formulas as:

$$\begin{array}{rclll} G & ::= & P & \mid \forall x.G & \mid \exists x.G \\ & \mid & \top & \mid G \,\&\, G & \\ & \mid & 0 & \mid G \oplus G & \\ & \mid & 1 & \mid G \bullet G & \mid G \circ G \\ & \mid & D \twoheadrightarrow G & \mid D \rightarrowtail G & \mid\ ¡G \\ & \mid & D \multimap G & \mid\ !G & \mid D \rightarrow G \end{array}$$

In order to make our derivations goal directed and focussed, we will use two separate judgements for UOLL derivations:

$$\Gamma; \Delta; \Omega \vdash G \quad \text{and} \quad \Gamma; \Delta; (\Omega_L; \Omega_R) \vdash D \gg P$$

where all contexts are restricted to only contain clause formulas; we respectively refer to the two judgements as goal directed judgements and focussed judgements.

Derivations of goal directed judgements follow the structure of the goal formula, G; when read bottom up, these derivations break down the goal until it is atomic using the various $-_R$ rules shown below. Once the goal formula is atomic, a clause formula must be chosen, using one of the *choice* rules shown below, and the derivation switches to focussed judgements which keep the "focus" of the derivation on the chosen clause formula until the atomic goal is satisfied.

Derivations of focussed judgements follow the structure of the chosen clause formula, D; when read bottom up, these derivations break down the clause formula until it is atomic using the various $-_L$ rules shown below [1] which also spawn off new goal directed sub-derivations. Focussed judgements have a split ordered context (i.e. Ω_L and Ω_R) in order to maintain the position of the chosen clause in the ordered context; i.e., $\Gamma; \Delta; (\Omega_L; \Omega_R) \vdash D \gg P$ may be thought of as $\Gamma; \Delta; \Omega_L, D, \Omega_R \vdash P$ where left rules may only apply to D.

We now present the UOLL derivation rules. The additive connectives do not require splitting contexts.

$$\frac{}{\Gamma; \Delta; \Omega \vdash \top}\top_R \qquad \frac{\Gamma; \Delta; \Omega \vdash G_0 \qquad \Gamma; \Delta; \Omega \vdash G_1}{\Gamma; \Delta; \Omega \vdash G_0 \,\&\, G_1}\&_R$$

$$\frac{\Gamma; \Delta; \Omega \vdash G_0}{\Gamma; \Delta; \Omega \vdash G_0 \oplus G_1}\oplus_{R0} \qquad \frac{\Gamma; \Delta; \Omega \vdash G_1}{\Gamma; \Delta; \Omega \vdash G_0 \oplus G_1}\oplus_{R1}$$

The multiplicatives do require splitting contexts.

$$\frac{}{\Gamma; \cdot; \cdot \vdash 1}1_R$$

$$\frac{\Gamma; \Delta_0; \Omega_0 \vdash G_0 \qquad \Gamma; \Delta_1; \Omega_1 \vdash G_1}{\Gamma; \Delta_0 \bowtie \Delta_1; \Omega_0, \Omega_1 \vdash G_0 \bullet G_1}\bullet_R$$

$$\frac{\Gamma; \Delta_0; \Omega_0 \vdash G_0 \qquad \Gamma; \Delta_1; \Omega_1 \vdash G_1}{\Gamma; \Delta_0 \bowtie \Delta_1; \Omega_1, \Omega_0 \vdash G_0 \circ G_1}\circ_R$$

The implictions add hypotheses to the appropriate places.

$$\frac{\Gamma; \Delta; \Omega, D \vdash G}{\Gamma; \Delta; \Omega \vdash D \twoheadrightarrow G}\twoheadrightarrow_R \qquad \frac{\Gamma; \Delta; D, \Omega \vdash G}{\Gamma; \Delta; \Omega \vdash D \rightarrowtail G}\rightarrowtail_R$$

$$\frac{\Gamma; \Delta, D; \Omega \vdash G}{\Gamma; \Delta; \Omega \vdash D \twoheadrightarrow G}\multimap_R \qquad \frac{\Gamma, D; \Delta; \Omega \vdash G}{\Gamma; \Delta; \Omega \vdash D \rightarrowtail G}\rightarrow_R$$

The quantifiers are standard.

$$\frac{\Gamma; \Delta; \Omega \vdash G[x := a]}{\Gamma; \Delta; \Omega \vdash \forall x.G}\forall_R(a \text{ new}) \qquad \frac{\Gamma; \Delta; \Omega \vdash G[x := t]}{\Gamma; \Delta; \Omega \vdash \exists x.G}\exists_R$$

The modalities explicitly capture independence from ordered and linear hypotheses.

$$\frac{\Gamma; \Delta; \cdot \vdash G}{\Gamma; \Delta; \cdot \vdash ¡G}¡_R \qquad \frac{\Gamma; \cdot; \cdot \vdash G}{\Gamma; \cdot; \cdot \vdash !G}!_R$$

[1] The symmetry between the behavior of goal directed and focussed derivations informs the idea of residuation discussed in section 4.

As mentioned above, the *choice* rules apply when the goal is atomic and allow the derivation to shift from being goal directed to focussing on a hypothesis; these rules have the functionality of the *place* and *copy* rules built-in and thus require splitting the ordered context.

$$\frac{\Gamma; \Delta; (\Omega_L; \Omega_R) \vdash D \gg P}{\Gamma; \Delta; \Omega_L, D, \Omega_R \vdash P}choice_\Omega$$

$$\frac{\Gamma; \Delta_L, \Delta_R; (\Omega_L; \Omega_R) \vdash D \gg P}{\Gamma; \Delta_L, D, \Delta_R; \Omega_L, \Omega_R \vdash P}choice_\Delta$$

$$\frac{\Gamma \bowtie D; \Delta; (\Omega_L; \Omega_R) \vdash D \gg P}{\Gamma \bowtie D; \Delta; \Omega_L, \Omega_R \vdash P}choice_\Gamma$$

As mentioned above, the *init* rule is restricted to atomic formulae.

$$\frac{}{\Gamma; \cdot; (\cdot; \cdot) \vdash P \gg P}init$$

The remaining rules are just reformulations of the left ($_L$) rules from OLL into the focussing judgment.

$$\frac{\Gamma; \Delta; (\Omega_L; \Omega_R) \vdash D[x := t] \gg P}{\Gamma; \Delta; (\Omega_L; \Omega_R) \vdash \forall x.D \gg P}\forall_L$$

$$\frac{\Gamma; \Delta; (\Omega_L; \Omega_R) \vdash D_0 \gg P}{\Gamma; \Delta; (\Omega_L; \Omega_R) \vdash D_0 \,\&\, D_1 \gg P}\&_{L0}$$

$$\frac{\Gamma; \Delta; (\Omega_L; \Omega_R) \vdash D_1 \gg P}{\Gamma; \Delta; (\Omega_L; \Omega_R) \vdash D_0 \,\&\, D_1 \gg P}\&_{L1}$$

$$\frac{\Gamma; \Delta; (\Omega_L; \Omega_R) \vdash D \gg P \qquad \Gamma; \Delta_G; \Omega_G \vdash G}{\Gamma; \Delta_G \bowtie \Delta; (\Omega_L; \Omega_G, \Omega_R) \vdash G \twoheadrightarrow D \gg P}\twoheadrightarrow_L$$

$$\frac{\Gamma; \Delta; (\Omega_L; \Omega_R) \vdash D \gg P \qquad \Gamma; \Delta_G; \Omega_G \vdash G}{\Gamma; \Delta_G \bowtie \Delta; (\Omega_L, \Omega_G; \Omega_R) \vdash G \rightarrowtail D \gg P}\rightarrowtail_L$$

$$\frac{\Gamma; \Delta; (\Omega_L; \Omega_R) \vdash D \gg P \qquad \Gamma; \Delta_G; \cdot \vdash G}{\Gamma; \Delta_G \bowtie \Delta; (\Omega_L; \Omega_R) \vdash G \multimap D \gg P}\multimap_L$$

$$\frac{\Gamma; \Delta; (\Omega_L; \Omega_R) \vdash D \gg P \qquad \Gamma; \cdot; \cdot \vdash G}{\Gamma; \Delta; (\Omega_L; \Omega_R) \vdash G \rightarrow D \gg P}\rightarrow_L$$

UOLL is sound and complete with respect to OLL (restricted to the same formula language). The proofs are non-trivial but largely follow the techniques used in [1, 8]; the complete details can be found in [10].

3.1 Ordered Linear Logic Progamming

Uniform proofs provide a strong computational intuition and remove a large amount of non-determinism from proof search. For intuitionistic logic, they remove all non-determinism[2] except the choice of clause to focus on; i.e., the only point in the proof search where a choice needs to be made is when an atomic goal arises.

However, for (ordered) linear logic, uniform proofs still contain large amounts of non-determinism related to how contexts are split. Such non-determinism must be removed to get a reasonable logic programming language, at least a reasonably efficient one. The techniques in [4, 6] to lazily split contexts can be generalized, as described in [10], to handle cases where the ordered context needs to be split between two premises.

Lazily splitting contexts does not deal with the context splits in the conclusion of the $choice_\Gamma$ and $choice_\Delta$ rules. The technique

[2] We ignore the choice of term in the $\exists_R$ and $\forall_L$ rules which is usually dealt with by unification.

for removing that non-determinism is called residuation, or logical compilation. As it is central to writing a meta-circular interpreter for Olli, we explore residuation in some detail in section 4.

3.2 Towards a Meta-Circular Intepreter

We would like to write a meta-circluar interpreter for Olli which uses the meta-logic contexts to represent the object-logic contexts, rather than carrying around explicit term-level representations of the contexts. Such an encoding has the pleasant property of allowing the object-logic connectives to be turned directly into meta-logic connectives, e.g. object-logic implication could be implemented by the clause: *goal (implies A B)* ← (*hyp A* → *goal B*).

One immediate obstacle to creating such an encoding for UOLL is the split ordered context in the focussing judgements. Object-logic terms cannot explicitly manipulate meta-logic contexts. As we shall see in section 4, (full) residuation removes the need for left rules and focussing judgements and thus saves us from having to represent split ordered contexts.

4 RESIDUATION

This section motivates and presents residuation, a technique for "logically compiling" clause formulae into new goal formulae. Olli relies on residuation to remove the ordered context splits in the UOLL *choice* rules which are not covered by lazy context splitting.

The ordered context splits in the $choice_\Gamma$ and $choice_\Delta$ rules exist because the unordered hypothesis must be placed somewhere in the ordered context before being used. Somewhat remarkably, this split can be done lazily, or implicitly; the structure of the clause determines where it can be placed in the ordered context. For example, the judgement

$$\cdot; P_1 \twoheadrightarrow P_2 \rightarrowtail P; P_2, P_1 \vdash P$$

has only one UOLL derivation which must place the linear formula $P_1 \twoheadrightarrow P_2 \rightarrowtail P$ (via the $choice_\Delta$ rule) between P_2 and P_1:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\cdot;\cdot;(\cdot;\cdot) \vdash P \gg P}\,init \qquad \cdot;\cdot;P_2 \vdash P_2}{\cdot;\cdot;(P_2;\cdot) \vdash P_2 \rightarrowtail P \gg P}\,{\rightarrowtail_L} \qquad \cdot;\cdot;P_1 \vdash P_1}{\cdot;\cdot;(P_2;P_1) \vdash P_1 \twoheadrightarrow P_2 \rightarrowtail P \gg P}\,{\twoheadrightarrow_L}}{\cdot; P_1 \twoheadrightarrow P_2 \rightarrowtail P; P_2, P_1 \vdash P}\,choice_\Delta$$

Furthermore, this derivation can be turned into something like

$$\dfrac{\dfrac{\cdot;\cdot;P_2 \vdash P_2 \qquad \cdot;\cdot;P_1 \vdash P_1}{\cdot;\cdot;P_2, P_1 \vdash P_2 \bullet P_1}\,\bullet_L}{\cdot; P_1 \twoheadrightarrow P_2 \rightarrowtail P; P_2, P_1 \vdash P}\,choice'_\Delta$$

where the *choice'*$_\Delta$ rule transforms the chosen (linear) clause, $P_1 \twoheadrightarrow P_2 \rightarrowtail P$, into the new goal formula, $P_2 \bullet P_1$, and does not split the ordered context.

This transformation, called residuation, exploits the symmetry between the focussed left rules and the goal directed right rules to remove the need for explicit left rules altogether; i.e., a clause formula is residuated into a new goal formula, providing it's head matches the current goal, whose structure captures the behavior of the erstwhile left rules. Residuation is described for intuitionistic and linear logic in [2] and can be extended to UOLL (for non-ordered clauses) as follows.

We describe residuation with the following judgement

$$G_I \,;\, D \gg P \setminus G_O$$

which may be thought of as a function from G_I, D, P to G_O where G_I is the accumulating residuated goal, D is the clause formula being residuated, P is the atomic goal we need to satisfy, and G_O is the final residual goal. The derivation rules for residuation are as follows:

$$\dfrac{}{G \,;\, P \gg P \setminus G} \qquad \dfrac{G_I \,;\, D \gg P \setminus G_O}{G_I \,;\, \forall x.D \gg P \setminus \exists x.G_O}$$

$$\dfrac{}{G \,;\, \top \gg P \setminus 0} \qquad \dfrac{G_I \,;\, D_0 \gg P \setminus G_0 \qquad G_I \,;\, D_1 \gg P \setminus G_1}{G_I \,;\, D_0 \,\&\, D_1 \gg P \setminus G_0 \oplus G_1}$$

$$\dfrac{G \circ G_I \,;\, D \gg P \setminus G_O}{G_I \,;\, G \twoheadrightarrow D \gg P \setminus G_O} \qquad \dfrac{G \bullet G_I \,;\, D \gg P \setminus G_O}{G_I \,;\, G \rightarrowtail D \gg P \setminus G_O}$$

$$\dfrac{\text{¡}G \bullet G_I \,;\, D \gg P \setminus G_O}{G_I \,;\, G \multimap D \gg P \setminus G_O} \qquad \dfrac{!G \bullet G_I \,;\, D \gg P \setminus G_O}{G_I \,;\, G \rightarrow D \gg P \setminus G_O}$$

Residuation succeeds, returning the accumulated residual goal, when the clause to residuate matches the current atomic goal. The rule for $\twoheadrightarrow$ uses $G \circ G_I$ rather than the equivalent $G_I \bullet G$ in order to preserve the order in which subgoals will be solved; for a clause $G_0 \twoheadrightarrow G_1 \rightarrowtail G_2 \twoheadrightarrow P$, we desire the order of the subgoals to be G_2, G_1, G_0 to get a Prolog-style interpretation of the clause where P is the head and $G_2 \circ (G_1 \bullet G_0)$ is the body.

The *choice* rules using residuation would look like:

$$\dfrac{1 \,;\, D \gg P \setminus G \qquad \Gamma; \Delta; \Omega \vdash G}{\Gamma; \Delta \bowtie D; \Omega \vdash P}\,choiceR_\Delta$$

$$\dfrac{1 \,;\, D \gg P \setminus G \qquad \Gamma \bowtie D; \Delta; \Omega \vdash G}{\Gamma \bowtie D; \Delta; \Omega \vdash P}\,choiceR_\Gamma$$

Note that the ordered context is not split; additionally, the initial residual goal of 1 is fine since $G \bullet 1 \equiv G \circ 1 \equiv G$. Then our transformed proof above becomes exactly

$$\dfrac{\Xi \qquad \dfrac{\cdot;\cdot;P_2 \vdash P_2 \qquad \cdot;\cdot;P_1 \vdash P_1 \circ 1}{\cdot;\cdot;P_2, P_1 \vdash P_2 \bullet P_1 \circ 1}\,\bullet_L}{\cdot; P_1 \twoheadrightarrow P_2 \rightarrowtail P; P_2, P_1 \vdash P}\,choiceR_\Delta$$

where Ξ is

$$\dfrac{\dfrac{\dfrac{}{P_2 \bullet P_1 \circ 1 \,;\, P \gg P \setminus P_2 \bullet P_1 \circ 1}}{P_1 \circ 1 \,;\, P_2 \rightarrowtail P \gg P \setminus P_2 \bullet P_1 \circ 1}}{1 \,;\, P_1 \twoheadrightarrow P_2 \rightarrowtail P \gg P \setminus P_2 \bullet P_1 \circ 1}$$

Full details of residuation for UOLL along with a proof of correctness can be found in [10].

5 FAILURE OF ORDERED RESIDUATION

Residuated UOLL gets rid of the non-deterministic ordered context split in the $choice_\Delta$ and $choice_\Gamma$ rules, but it leaves the $choice_\Omega$ rule unchanged. This is fine from the point-of-view of eliminating non-deterministic splits, since the context split in $choice_\Omega$ is deterministic. However, it would be nice to have all the *choice* rules use residuation and then dispense with the separate judgement for focussing on a clause formula, and the associated $-_L$ rules.

Unfortunately, $choice_\Omega$ cannot be correctly converted to the same form as the other two rules. As mentioned at the beginning of the paper, the problem comes down to the position of the ordered clause not being represented in the residuated goal; to be more concrete, a putative $choiceR_\Omega$ rule

$$\frac{1\,;\,D \gg P \setminus G \qquad \Gamma;\Delta;\Omega_L,\Omega_R \vdash G}{\Gamma;\Delta;\Omega_L,D,\Omega_R \vdash P}choiceR_\Omega$$

would allow both

$$\cdot;\cdot;P_2,P_1 \twoheadrightarrow P_2 \rightarrowtail P, P_1 \vdash P \quad \text{and} \quad \cdot;\cdot;P_2,P_1,P_1 \twoheadrightarrow P_2 \rightarrowtail P \vdash P$$

to be derived.

In addition to leaving an annoying asymmetry in the logic, the lack of residuation for ordered clauses effectively prevents us from writing a nice[3] meta-circular interpreter for Olli directly from the UOLL derivation rules. The actual difficulty lies in the focussed judgement which explicitly splits apart the ordered context; if we use the meta-level contexts to represent the object-level contexts, we have no way of representing the split ordered context. On the other hand, we could directly transcribe the UOLL derivation rules into Olli if we only had the goal directed judgements and a residuation function.

6 REMOVING ORDERED CHOICE

Since the $choice_\Omega$ rule encompasses the diffulties previously described, a solution to the problem would be to not use that rule; i.e., to design a system in which we never focus on a hypothesis in the ordered context. We propose to essentially demote the ordered context to a list of placeholders (just some arbitrary atomic formulae), and to put ordered clauses into the linear context, after translation to a form which can reference those placeholders. For example, the judgement

$$\cdot\,;\,\cdot\,;\,P_2,P_1 \twoheadrightarrow P_2 \rightarrowtail P, P_1 \vdash P$$

could be translated to something like

$$\cdot\,;\,Q_P \twoheadrightarrow P_1 \twoheadrightarrow P_2 \rightarrowtail P\,;\,P_2,Q_P,P_1 \vdash P$$

where the atom Q_P is a placeholder which has been placed into the appropriate position in the ordered context. Furthermore, the (only) derivation of this new judgement would just be:

$$\dfrac{\dfrac{\begin{array}{c}\Xi\\ \cdot;\cdot;(P_2;P_1) \vdash P_1 \twoheadrightarrow P_2 \rightarrowtail P \gg P\end{array} \qquad \cdot;\cdot;Q_P \vdash Q_P}{\cdot;\cdot;(P_2;Q_P,P_1) \vdash Q_P \twoheadrightarrow P_1 \twoheadrightarrow P_2 \rightarrowtail P \gg P}\twoheadrightarrow_L}{\cdot;Q_P \twoheadrightarrow P_1 \twoheadrightarrow P_2 \rightarrowtail P;P_2,Q_P,P_1 \vdash P}choice_\Delta$$

where Ξ is the derivation for the original judgement (without the final $choice_\Delta$ step) at the beginning of section 4. Note that there is only one valid way to split the ordered context, i.e. at the placeholder Q_P, thus the judgement

$$\cdot;Q_P \twoheadrightarrow P_1 \twoheadrightarrow P_2 \rightarrowtail P\,;\,P_2,P_1,Q_P \vdash P$$

would not be derivable since we cannot derive

$$\cdot;\cdot;(P_2,P_1\,;\,\cdot) \vdash P_1 \twoheadrightarrow P_2 \rightarrowtail P \gg P$$

[3] One which doesn't explicitly represent the logical contexts with terms.

7 UNIFORM ATOMIC ORDERED LINEAR LOGIC

This section fleshes out the idea sketched at the end of the previous section to get a full logical system, UAOLL, which is equivalent to UOLL as presented in section 3. Section 8 gives a proof of correctness for UAOLL. The defining characteristic of UAOLL is that no ordered formula can be chosen to focus on; as explained in sections 5 and 6, this will ultimately allow us to have a system which can be directly transcribed into Olli to get a meta-circular interpreter.

In order to prohibit focussing on an ordered formula, UAOLL will demote the ordered context to a list of placeholders, described in section 6. For the rest of this section, we assume a distinguished placeholder predicate Q, taking a single term as an argument, e.g. Q_x. We define demoted ordered contexts as:

$$\omega \;::=\; \cdot \mid \omega, Q_x \qquad \text{where } x \text{ not in } \omega$$

In other words, a demoted ordered context is a list of unique placeholders.

We allow placeholders in the grammar of goal formulae, G, defined in section 3:

$$G \;::=\; Q_x \mid P \mid \ldots$$

However, we do not add placeholders to the grammar of clause formulae, D, defined in section 3. Instead, we introduce modified clause formulae:

$$E \;::=\; D \mid Q_x \twoheadrightarrow D$$

and modified linear contexts, δ, for use in UAOLL:

$$\delta \;::=\; \cdot \mid \delta, E$$

7.1 UAOLL derivations

We use the following two judgements for UAOLL derivations:

$$\Gamma;\delta;\omega \vdash G \quad \text{and} \quad \Gamma;\delta;(\omega_L;\omega_R) \vdash E \gg P$$

The derivation rules follow.

All the non-implication right rules are identical to their UOLL counterparts.

$$\frac{}{\Gamma;\delta;\omega \vdash \top}\top_{R'} \qquad \frac{\Gamma;\delta;\omega \vdash G_0 \qquad \Gamma;\delta;\omega \vdash G_1}{\Gamma;\delta;\omega \vdash G_0 \,\&\, G_1}\&_{R'}$$

$$\frac{\Gamma;\delta;\omega \vdash G_0}{\Gamma;\delta;\omega \vdash G_0 \oplus G_1}\oplus_{R0'} \qquad \frac{\Gamma;\omega;\omega \vdash G_1}{\Gamma;\delta;\omega \vdash G_0 \oplus G_1}\oplus_{R1'}$$

$$\frac{\Gamma;\delta;\omega \vdash G[x := a]}{\Gamma;\delta;\omega \vdash \forall x.G}\forall_{R'}(a \text{ new}) \qquad \frac{\Gamma;\delta;\omega \vdash G[x := t]}{\Gamma;\delta;\omega \vdash \exists x.G}\exists_{R'}$$

$$\frac{\Gamma;\delta;\cdot \vdash G}{\Gamma;\delta;\cdot \vdash {¡}G}{¡}_{R'} \qquad \frac{\Gamma;\cdot;\cdot \vdash G}{\Gamma;\cdot;\cdot \vdash !G}!_{R'} \qquad \frac{}{\Gamma;\cdot;\cdot \vdash 1}1_{R'}$$

$$\frac{\Gamma;\delta_0;\omega_0 \vdash G_0 \qquad \Gamma;\delta_1;\omega_1 \vdash G_1}{\Gamma;\delta_0 \bowtie \delta_1;\omega_0,\omega_1 \vdash G_0 \bullet G_1}\bullet_{R'}$$

$$\frac{\Gamma;\delta_0;\omega_0 \vdash G_0 \qquad \Gamma;\delta_1;\omega_1 \vdash G_1}{\Gamma;\delta_0 \bowtie \delta_1;\omega_1,\omega_0 \vdash G_0 \circ G_1}\circ_{R'}$$

The ordered implication rules add unique placeholders to their hypotheses which are put into the linear context, and only put the placeholders into the ordered context.

$$\frac{\Gamma;\delta,Q_x \twoheadrightarrow D;\omega,Q_x \vdash G}{\Gamma;\delta;\omega \vdash D \twoheadrightarrow G}\twoheadrightarrow_{R'} (x \text{ new})$$

$$\frac{\Gamma;\delta,Q_x \twoheadrightarrow D;Q_x,\omega \vdash G}{\Gamma;\delta;\omega \vdash D \rightarrowtail G}\rightarrowtail_{R'} (x \text{ new})$$

The unordered implications are identical to their UOLL counterparts.

$$\frac{\Gamma; \delta, D; \omega \vdash G}{\Gamma; \delta; \omega \vdash D \multimap G} \multimap_{R'} \qquad \frac{\Gamma, D; \delta; \omega \vdash G}{\Gamma; \delta; \omega \vdash D \to G} \to_{R'}$$

The $choice_\omega$ is much simpler than its UOLL counterpart, $choice_\Omega$ since the ordered context only contains placeholders.

$$\frac{}{\Gamma; \cdot; Q_x \vdash Q_x} choice_\omega$$

The remaining rules are identical to their UOLL counterparts.

$$\frac{\Gamma \bowtie D; \delta; (\omega_L; \omega_R) \vdash D \gg P}{\Gamma \bowtie D; \delta; \omega_L, \omega_R \vdash P} choice_\gamma$$

$$\frac{\Gamma; \delta; (\omega_L; \omega_R) \vdash E \gg P}{\Gamma; \delta \bowtie E; \omega_L, \omega_R \vdash P} choice_\delta$$

$$\frac{}{\Gamma; \cdot; (\cdot; \cdot) \vdash P \gg P} init' \qquad \frac{\Gamma; \delta; (\omega_L; \omega_R) \vdash D[x := t] \gg P}{\Gamma; \delta; (\omega_L; \omega_R) \vdash \forall x.D \gg P} \forall'_L$$

$$\frac{\Gamma; \delta; (\omega_L; \omega_R) \vdash D_0 \gg P}{\Gamma; \delta; (\omega_L; \omega_R) \vdash D_0 \,\&\, D_1 \gg P} \&_{0L'}$$

$$\frac{\Gamma; \delta; (\omega_L; \omega_R) \vdash D_1 \gg P}{\Gamma; \delta; (\omega_L; \omega_R) \vdash D_0 \,\&\, D_1 \gg P} \&_{1L'}$$

$$\frac{\Gamma; \delta; (\omega_L; \omega_R) \vdash D \gg P \qquad \Gamma; \delta_G; \omega_G \vdash G}{\Gamma; \delta_G \bowtie \delta; (\omega_L; \omega_G, \omega_R) \vdash G \twoheadrightarrow D \gg P} \twoheadrightarrow_{L'}$$

$$\frac{\Gamma; \Delta; (\Omega_L; \Omega_R) \vdash D \gg P \qquad \Gamma; \Delta_G; \Omega_G \vdash G}{\Gamma; \Delta_G \bowtie \Delta; (\Omega_L, \Omega_G; \Omega_R) \vdash G \rightarrowtail D \gg P} \rightarrowtail_{L'}$$

$$\frac{\Gamma; \Delta; (\Omega_L; \Omega_R) \vdash D \gg P \qquad \Gamma; \Delta_G; \cdot \vdash G}{\Gamma; \Delta_G \bowtie \Delta; (\Omega_L; \Omega_R) \vdash G \multimap D \gg P} \multimap_{L'}$$

$$\frac{\Gamma; \Delta; (\Omega_L; \Omega_R) \vdash D \gg P \qquad \Gamma; \cdot; \cdot \vdash G}{\Gamma; \Delta; (\Omega_L; \Omega_R) \vdash G \to D \gg P} \to_{L'}$$

We end this section with an example derivation.

$$\cfrac{\cfrac{\cfrac{\Xi \qquad \cfrac{}{\cdot; \cdot; Q_P \vdash Q_P} choice_\omega}{\cdot; Q_2 \twoheadrightarrow P_2, Q_1 \twoheadrightarrow P_1; (Q_2; Q_P, Q_1) \vdash Q_P \twoheadrightarrow D \gg P} \twoheadrightarrow_{L'}}{\cdot; Q_2 \twoheadrightarrow P_2, Q_P \twoheadrightarrow D, Q_1 \twoheadrightarrow P_1; Q_2, Q_P, Q_1 \vdash P} choice_\delta}{\cdot; \cdot; \cdot \vdash P_2 \twoheadrightarrow D \twoheadrightarrow P_1 \twoheadrightarrow P} \twoheadrightarrow_{R'} \times 3$$

where $D = P_1 \twoheadrightarrow P_2 \rightarrowtail P$
and Ξ is

$$\cfrac{\cfrac{\cfrac{}{\cdot; \cdot; (\cdot; \cdot) \vdash P \gg P} init' \qquad \Xi_2}{\cdot; Q_2 \twoheadrightarrow P_2; (Q_2; \cdot) \vdash P_2 \rightarrowtail P \gg P} \twoheadrightarrow_{L'} \qquad \Xi_1}{\cdot; Q_2 \twoheadrightarrow P_2, Q_1 \twoheadrightarrow P_1; (Q_2; Q_1) \vdash P_1 \twoheadrightarrow P_2 \rightarrowtail P \gg P} \twoheadrightarrow_{L'}$$

and Ξ_1 is

$$\cfrac{\cfrac{\cfrac{}{\cdot; \cdot; (\cdot; \cdot) \vdash P_1 \gg P_1} init' \qquad \cfrac{}{\cdot; \cdot; Q_1 \vdash Q_1} choice_\omega}{\cdot; \cdot; (\cdot; Q_1) \vdash Q_1 \twoheadrightarrow P_1 \gg P_1} \twoheadrightarrow_{L'}}{\cdot; Q_1 \twoheadrightarrow P_1; Q_1 \vdash P_1} choice_\delta$$

and Ξ_2 is similar to Ξ_1.

8 CORRECTNESS

In this section, we formally prove the correctness of UAOLL with respect to UOLL. Since the ordered UAOLL implications add to both the linear and ordered contexts, we will need machinery for translating between the linear and ordered contexts of UAOLL and UOLL.

8.1 Translating contexts

The linear context in UAOLL effectively contains both linear and ordered hypotheses, therefore we will need to simultaneously translate linear and ordered contexts between UAOLL and UOLL. We use the judgement $\delta\,;\,\omega \vDash \Delta\,;\,\Omega$ for this translation as follows:

$$\frac{}{\cdot\,;\,\cdot \vDash \cdot\,;\,\cdot} \cdot \qquad \frac{\delta\,;\,\omega \vDash \Delta\,;\,\Omega}{\delta \bowtie D\,;\,\omega \vDash \Delta \bowtie D\,;\,\Omega} lin$$

$$\frac{\delta_L\,;\,\omega_L \vDash \Delta_L\,;\,\Omega_L \qquad \delta_R\,;\,\omega_R \vDash \Delta_R\,;\,\Omega_R}{\delta_L \bowtie \delta_R \bowtie Q_x \twoheadrightarrow D\,;\,\omega_L, Q_x, \omega_R \vDash \Delta_L \bowtie \Delta_R\,;\,\Omega_L, D, \Omega_R} ord$$
(*where* Q_x *not in* ω_L *nor in* ω_R)

We can freely combine translated contexts to get larger translated contexts.

Lemma 1 (Combining Translations).
$\delta_L\,;\,\omega_L \vDash \Delta_L\,;\,\Omega_L$ *and* $\delta_R\,;\,\omega_R \vDash \Delta_R\,;\,\Omega_R$ *implies*
$\delta_L \bowtie \delta_R\,;\,\omega_L, \omega_R \vDash \Delta_L \bowtie \Delta_R\,;\,\Omega_L, \Omega_R$

Proof. By induction on the structure of the given derivations. □

We cannot freely split translated contexts apart since we need to make sure that corresponding pieces stay together. However, we can safely split apart translated contexts based on a given ordered context split.

Lemma 2 (Splitting Translations).

(1) $\delta\,;\,\omega_L, \omega_R \vDash \Delta\,;\,\Omega$ *implies*
$\delta_L\,;\,\omega_L \vDash \Delta_L\,;\,\Omega_L$ *and* $\delta_R\,;\,\omega_R \vDash \Delta_R\,;\,\Omega_R$
where $\delta = \delta_L \bowtie \delta_R$ *and* $\Delta = \Delta_L \bowtie \Delta_R$ *and* $\Omega = \Omega_L, \Omega_R$

(2) $\delta\,;\,\omega \vDash \Delta\,;\,\Omega_L, \Omega_R$ *implies*
$\delta_L\,;\,\omega_L \vDash \Delta_L\,;\,\Omega_L$ *and* $\delta_R\,;\,\omega_R \vDash \Delta_R\,;\,\Omega_R$
where $\Delta = \Delta_L \bowtie \Delta_R$ *and* $\delta = \delta_L \bowtie \delta_R$ *and* $\omega = \omega_L, \omega_R$

Proof. By inversion on *lin* and *ord*, and induction on the structure of the given derivation. □

We note that linear clauses of form D (i.e. not $Q_x \twoheadrightarrow D$) are not associated with anything in the ordered context, and a valid judgement $\delta\,;\,\cdot \vDash \Delta\,;\,\cdot$ will only have such clauses in δ and Δ; thus, we have the following corollary.

Lemma 3 (Inversion on *ord*).

(1) $\delta \bowtie Q_x \twoheadrightarrow D\,;\,\omega, Q_x \vDash \Delta\,;\,\Omega'$ *implies*
$\Omega' = \Omega, D$ *and* $\delta\,;\,\omega \vDash \Delta\,;\,\Omega$.

(2) $\delta'\,;\,\omega' \vDash \Delta\,;\,\Omega, D$ *implies*
$\delta' = \delta \bowtie Q_x \twoheadrightarrow D$ *and* $\omega' = \omega, Q_x$ *and* $\delta\,;\,\omega \vDash \Delta\,;\,\Omega$.

(3) $\delta \bowtie Q_x \twoheadrightarrow D\,;\,Q_x, \omega \vDash \Delta\,;\,\Omega'$ *implies*
$\Omega' = D, \Omega$ *and* $\delta\,;\,\omega \vDash \Delta\,;\,\Omega$.

(4) $\delta'\,;\,\omega' \vDash \Delta\,;\,D, \Omega$ *implies*
$\delta' = \delta \bowtie Q_x \twoheadrightarrow D$ *and* $\omega' = Q_x, \omega$ *and* $\delta\,;\,\omega \vDash \Delta\,;\,\Omega$.

Proof. By using lemma 2 and then using lemma 1 to move any stray linear clauses back to the desired side. □

8.2 Soundness

We now state and prove the soundness of UOALL with respect to UOLL. Specifically, given a UAOLL derivation of a judgement, J, there is a corresponding UOLL judgement, J', the result of translating the UAOLL contexts to UOLL contexts, and a corresponding UOLL derivation of J'.

Theorem 1 (Soundness).

(1) $\Gamma;\delta;\omega \vdash G$ *implies* $\exists\Delta,\Omega\,.\,\delta\,;\,\omega \vDash \Delta\,;\,\Omega$ *and* $\Gamma;\Delta;\Omega \vdash G$

(2) $\Gamma;\delta;(\omega_L;\omega_R) \vdash D \gg P$ *implies* $\exists\delta_L,\delta_R,\Delta_L,\Delta_R,\Omega_L,\Omega_R\,.\,\delta = \delta_L \bowtie \delta_R$ *and* $\delta_L\,;\,\omega_L \vDash \Delta_L\,;\,\Omega_L$ *and* $\delta_R\,;\,\omega_R \vDash \Delta_R\,;\,\Omega_R$ *and* $\Gamma;\Delta_L \bowtie \Delta_R;(\Omega_L;\Omega_R) \vdash D \gg P$

Proof. By induction on the structure of the given derivation. We show some representative cases.

case: $$\dfrac{\Gamma;\delta,Q_x \twoheadrightarrow D;\omega,Q_x \vdash G}{\Gamma;\delta;\omega \vdash D \twoheadrightarrow G}\twoheadrightarrow_{R'}\ (x\ new)$$

$\delta,Q_x \twoheadrightarrow D\,;\,\omega,Q_x \vDash \Delta\,;\,\Omega'$
and $\Gamma;\Delta;\Omega' \vdash G$ by ind. hyp.
$\Omega' = \Omega,D$ and $\delta\,;\,\omega \vDash \Delta\,;\,\Omega$ by lemma 3.
$\Gamma;\Delta;\Omega \vdash D \twoheadrightarrow G$ by $\twoheadrightarrow_R$.

case: $$\dfrac{\Gamma;\delta;(\omega_L;\omega_R) \vdash E \gg P}{\Gamma;\delta \bowtie E;\omega_L,\omega_R \vdash P}choice_\delta$$

sub: $E = D$
$\delta = \delta_L \bowtie \delta_R$ and
$\delta_L\,;\,\omega_L \vDash \Delta_L\,;\,\Omega_L$ and $\delta_R\,;\,\omega_R \vDash \Delta_R\,;\,\Omega_R$ and
$\Gamma;\Delta_L \bowtie \Delta_R;(\Omega_L;\Omega_R) \vdash D \gg P$ by ind. hyp.
$\delta_L \bowtie \delta_R\,;\,\omega_L,\omega_R \vDash \Delta_L \bowtie \Delta_R\,;\,\Omega_L,\Omega_R$ by lemma 1.
$\delta_L \bowtie \delta_R \bowtie D\,;\,\omega_L,\omega_R \vDash \Delta_L \bowtie \Delta_R \bowtie D\,;\,\Omega_L,\Omega_R$ by *lin.*
$\Gamma;\Delta_L \bowtie \Delta_R \bowtie D;\Omega_L,\Omega_R \vdash P$ by $choice_\Delta$.

sub: $E = Q_x \twoheadrightarrow D$
$\Gamma;\delta;(\omega_L;\omega_{R'}) \vdash D \gg P$ and
$\omega_R = Q_x,\omega_{R'}$ by inversion on $\twoheadrightarrow_{L'}$.
$\delta = \delta_L \bowtie \delta_R$ and
$\delta_L\,;\,\omega_L \vDash \Delta_L\,;\,\Omega_L$ and $\delta_R\,;\,\omega_{R'} \vDash \Delta_R\,;\,\Omega_R$ and
$\Gamma;\Delta_L \bowtie \Delta_R;(\Omega_L;\Omega_R) \vdash D \gg P$ by ind. hyp.
$\delta_L \bowtie \delta_R \bowtie Q_x \twoheadrightarrow D\,;\,\omega_L,Q_x,\omega_{R'} \vDash \Delta_L \bowtie \Delta_R\,;\,\Omega_L,D,\Omega_R$ by *ord.*
$\Gamma;\Delta_L \bowtie \Delta_R;\Omega_L,D,\Omega_R \vdash P$ by $choice_\Omega$.

case: $$\dfrac{\Gamma;\delta;(\omega_L;\omega_R) \vdash D \gg P \qquad \Gamma;\delta_G;\omega_G \vdash G}{\Gamma;\delta \bowtie \delta_G;(\omega_L;\omega_G,\omega_R) \vdash G \twoheadrightarrow D \gg P}\twoheadrightarrow_{L'}$$

$\delta = \delta_L \bowtie \delta_R$ and
$\delta_L\,;\,\omega_L \vDash \Delta_L\,;\,\Omega_L$ and $\delta_R\,;\,\omega_R \vDash \Delta_R\,;\,\Omega_R$ and
$\Gamma;\Delta_L \bowtie \Delta_R;(\Omega_L;\Omega_R) \vdash D \gg P$ by ind. hyp.
$\delta_G\,;\,\omega_G \vDash \Delta_G\,;\,\Omega_G$ and $\Gamma;\Delta_G;\Omega_G \vdash G$ by ind hyp.
$\delta_G \bowtie \delta_R\,;\,\omega_G,\omega_R \vDash \Delta_G \bowtie \Delta_R\,;\,\Omega_G,\Omega_R$ by lemma 1.
$\Gamma;\Delta_L \bowtie \Delta_R \bowtie \Delta_G;(\Omega_L;\Omega_G,\Omega_R) \vdash G \twoheadrightarrow D \gg P$ by $\twoheadrightarrow_L$.

□

8.3 Completeness

This section formally states and proves the completeness of UAOLL with respect to UOLL; this theorem and proof are largely symmetric to the those for soundess in section 8.2.

Theorem 2 (Completeness).

(1) $\Gamma;\Delta;\Omega \vdash G$ *implies* $\exists\delta,\omega\,.\,\delta\,;\,\omega \vDash \Delta\,;\,\Omega$ *and* $\Gamma;\delta;\omega \vdash G$

(2) $\Gamma;\Delta;(\Omega_L;\Omega_R) \vdash D \gg P$ *implies* $\exists\Delta_L,\Delta_R,\delta_L,\delta_R,\omega_L,\omega_R\,.\,\Delta = \Delta_L \bowtie \Delta_R$ *and* $\delta_L\,;\,\omega_L \vDash \Delta_L\,;\,\Omega_L$ *and* $\delta_R\,;\,\omega_R \vDash \Delta_R\,;\,\Omega_R$ *and* $\Gamma;\delta_L \bowtie \delta_R;(\omega_L;\omega_R) \vdash D \gg P$

Proof. By induction on the structure of the given derivation. We show some representative cases.

case: $$\dfrac{\Gamma;\Delta;\Omega,D \vdash G}{\Gamma;\Delta;\Omega \vdash D \twoheadrightarrow G}\twoheadrightarrow_R$$

$\delta'\,;\,\omega' \vDash \Delta\,;\,\Omega,D$ and $\Gamma;\delta';\omega' \vdash G$ by ind. hyp.
$\delta' = \delta \bowtie Q_x \twoheadrightarrow D$ and $\omega' = \omega,Q_x$ and
$\delta\,;\,\omega \vDash \Delta\,;\,\Omega$ by lemma 3.
$\Gamma;\delta;\omega \vdash D \twoheadrightarrow G$ by $\twoheadrightarrow_{R'}$.

case: $$\dfrac{\Gamma;\Delta;(\Omega_L;\Omega_R) \vdash D \gg P}{\Gamma;\Delta;\Omega_L,D,\Omega_R \vdash P}choice_\Omega$$

$\Delta = \Delta_L \bowtie \Delta_R$ and
$\delta_L\,;\,\omega_L \vDash \Delta_L\,;\,\Omega_L$ and $\delta_R\,;\,\omega_R \vDash \Delta_R\,;\,\Omega_R$ and
$\Gamma;\delta_L \bowtie \delta_R;(\omega_L;\omega_R) \vdash D \gg P$ by ind. hyp.
$\delta_L \bowtie \delta_R \bowtie Q_x \twoheadrightarrow D\,;\,\omega_L,Q_x,\omega_R \vDash \Delta_L \bowtie \Delta_R\,;\,\Omega_L,D,\Omega_R$ by *ord.*
$\Gamma;\cdot;Q_x \vdash Q_x$ by $choice_\omega$.
$\Gamma;\delta_L \bowtie \delta_R;(\omega_L;Q_x,\omega_R) \vdash Q_x \twoheadrightarrow D \gg P$ by $\twoheadrightarrow_{R'}$.
$\Gamma;\delta_L \bowtie \delta_R \bowtie Q_x \twoheadrightarrow D;\omega_L,Q_x,\omega_R \vdash P$ by $choice_\delta$.

case: $$\dfrac{\Gamma;\Delta;(\Omega_L;\Omega_R) \vdash D \gg P}{\Gamma;\Delta \bowtie D;\Omega_L,\Omega_R \vdash P}choice_\Delta$$

$\Delta = \Delta_L \bowtie \Delta_R$ and
$\delta_L\,;\,\omega_L \vDash \Delta_L\,;\,\Omega_L$ and $\delta_R\,;\,\omega_R \vDash \Delta_R\,;\,\Omega_R$ and
$\Gamma;\delta_L \bowtie \delta_R;(\omega_L;\omega_R) \vdash D \gg P$ by ind. hyp.
$\delta_L \bowtie \delta_R\,;\,\omega_L,\omega_R \vDash \Delta_L \bowtie \Delta_R\,;\,\Omega_L,\Omega_R$ by lemma 1.
$\delta_L \bowtie \delta_R \bowtie D\,;\,\omega_L,\omega_R \vDash \Delta_L \bowtie \Delta_R \bowtie D\,;\,\Omega_L,\Omega_R$ by *lin.*
$\Gamma;\delta_L \bowtie \delta_R \bowtie D;\omega_L,\omega_R \vdash P$ by $choice_\delta$.

case: $$\dfrac{\Gamma;\Delta;(\Omega_L;\Omega_R) \vdash D \gg P \qquad \Gamma;\Delta_G;\Omega_G \vdash G}{\Gamma;\Delta \bowtie \Delta_G;(\Omega_L;\Omega_G,\Omega_R) \vdash G \twoheadrightarrow D \gg P}\twoheadrightarrow_L \text{ and}$$

$\Delta = \Delta_L \bowtie \Delta_R$ and
$\delta_L\,;\,\omega_L \vDash \Delta_L\,;\,\Omega_L$ and $\delta_R\,;\,\omega_R \vDash \Delta_R\,;\,\Omega_R$ and
$\Gamma;\delta_L \bowtie \delta_R;(\omega_L;\omega_R) \vdash D \gg P$ by ind. hyp.
$\delta_G\,;\,\omega_G \vDash \Delta_G\,;\,\Omega_G$ and $\Gamma;\delta_G;\omega_G \vdash G$ by ind hyp.
$\delta_G \bowtie \delta_R\,;\,\omega_G,\omega_R \vDash \Delta_G \bowtie \Delta_R\,;\,\Omega_G,\Omega_R$ by lemma 1.
$\Gamma;\delta_L \bowtie \delta_G \bowtie \delta_R;(\omega_L;\omega_G,\omega_R) \vdash G \twoheadrightarrow D \gg P$ by $\twoheadrightarrow_{L'}$.

□

9 RESIDUATED UAOLL

The presentation of UAOLL in section 7 does not allow focussing on an ordered hypothesis thus overcoming the main obstacle to writing a nice meta-circular interpreter as described in section 5. However, UAOLL is not directly suited for transcription into Olli since its focussing judgement still maintains a split ordered context. In this section we remove the need for focussing judgements by adding residuation to UAOLL.

As it turns out, there is not much to do. UAOLL is close enough to UOLL that we may directly use the residuation machinery described in section 4. We need only modify the $choice_\gamma$ and $choice_\delta$ rules to use residuation, and remove the *init'* rule. The modified *choice* rules are as follows:

$$\frac{1\,;\,D \gg P \setminus G \qquad \Gamma \bowtie D;\Delta;\Omega \vdash G}{\Gamma \bowtie D;\Delta;\Omega \vdash P}\,choiceR_\gamma$$

$$\frac{1\,;\,D \gg P \setminus G \qquad \Gamma;\Delta;\Omega \vdash G}{\Gamma;\Delta \bowtie D;\Omega \vdash P}\,choiceR_\delta$$

The correctness statement and proof for adding residuation to UAOLL is essentially unchanged from the version for UOLL, full details can be found in [10].

We end this section with an example derivation in residuated UAOLL.

$$\dfrac{\dfrac{\Xi \qquad \dfrac{\Xi_2 \qquad \dfrac{\Xi_1 \qquad \dfrac{\dfrac{}{\cdot;\cdot;Q_P \vdash Q_P}\,choice_\omega \qquad \dfrac{}{\cdot;\cdot;\cdot \vdash 1}\,1_{R'}}{\cdot;\cdot;Q_P \vdash Q_P \circ 1}\,\circ_{R'}}{\cdot;Q_1 \twoheadrightarrow P_1;Q_P,Q_1 \vdash P_1 \circ Q_P \circ 1}\,\circ_{R'}}{\cdot;Q_2 \twoheadrightarrow P_2,Q_1 \twoheadrightarrow P_1;Q_2,Q_P,Q_1 \vdash P_2 \bullet P_1 \circ Q_P \circ 1}\,\bullet_{R'}}{\cdot;Q_2 \twoheadrightarrow P_2,Q_P \twoheadrightarrow P_1 \twoheadrightarrow P_2 \rightarrowtail P,Q_1 \twoheadrightarrow P_1;Q_2,Q_P,Q_1 \vdash P}\,choiceR_\delta}{\cdot;\cdot;\cdot \vdash P_2 \twoheadrightarrow (P_1 \twoheadrightarrow P_2 \rightarrowtail P) \twoheadrightarrow P_1 \twoheadrightarrow P}\,\twoheadrightarrow_{R'} \times 3$$

where Ξ is

$$\dfrac{\dfrac{\dfrac{\dfrac{}{P_2 \bullet P_1 \circ Q_P \circ 1\,;\,P \gg P \setminus P_2 \bullet P_1 \circ Q_P \circ 1}}{P_1 \circ Q_P \circ 1\,;\,P_2 \rightarrowtail P \gg P \setminus P_2 \bullet P_1 \circ Q_P \circ 1}}{Q_P \circ 1\,;\,P_1 \twoheadrightarrow P_2 \rightarrowtail P \gg P \setminus P_2 \bullet P_1 \circ Q_P \circ 1}}{1\,;\,Q_P \twoheadrightarrow P_1 \twoheadrightarrow P_2 \rightarrowtail P \gg P \setminus P_2 \bullet P_1 \circ Q_P \circ 1}$$

Ξ_2 is

$$\dfrac{\dfrac{\dfrac{}{Q_2 \circ 1\,;\,P_2 \gg P_2 \setminus Q_2 \circ 1}}{1\,;\,Q_2 \twoheadrightarrow P_2 \gg P_2 \setminus Q_2 \circ 1} \qquad \dfrac{\dfrac{}{\cdot;\cdot;Q_2 \vdash Q_2}\,choice_\omega \qquad \dfrac{}{\cdot;\cdot;\cdot \vdash 1}\,1_{R'}}{\cdot;\cdot;Q_2 \vdash Q_2 \circ 1}\,\circ_{R'}}{\cdot;Q_2 \twoheadrightarrow P_2;Q_2 \vdash P_2}\,choice_\delta$$

and Ξ_1 is similar to Ξ_2.

10 META-CIRCULAR INTERPRETER

This section shows the promised meta-circular interpreter for Olli, which is (almost) a direct transcription of the residuated UAOLL rules, making use of the meta-logic contexts. The main difference between UAOLL and our object language is that we don't formalize the distinction between goal formulae and clause formulae; this makes the correctness of our interpreter rely on the operational semantics of Olli, i.e. attempting to solve for a grammatically incorrect formula will result in failure to find a proof, rather than in a type error.

We use the same syntax for the meta-logic as we've used for UAOLL. We assume a simply typed lambda calculus for the term language. *o* is the (built-in) type of predicates; i.e. $\twoheadrightarrow$ *: o -> o -> o*.

10.1 Brief review of Olli

We write our Olli code in a Prolog style where program clauses generally take the form: *head* $\twoheadleftarrow$ *body* where the whole clause is in the grammar for clauses, *D*, defined in section 3; i.e., *head* is a clause formula, *D*, while *body* is a goal formula, *G*. We use $\twoheadrightarrow$ as the outermost connective for uniformity even though the other implications often also work[4]. Additionally, following Prolog, we use upper case letters for variables universally quantified at the outermost level; e.g. $a\,X \twoheadleftarrow b\,Y$ is equivalent to $\forall x.\forall y.a\,x \twoheadleftarrow b\,y$.

The behavior of goal formulae (i.e. the bodies of clauses) generally follows the bottom-up reading of the corresponding (goal directed) rule for that formula. Universal quantifiers create a fresh parameter guaranteed not to already exist.

Implications dynamically add hypotheses to the context matching the implication; e.g., $a\,X \twoheadleftarrow b\,X \rightarrowtail c\,X \multimap d$. is a clause which takes an argument *X* and adds *b X* to the left end of the ordered context, then adds *c X* to the linear context, and then calls *d*.

Ordered multiplicative conjunctions specify the relative order of the hypotheses used by each conjunct; e.g., $a\,X \twoheadleftarrow b \bullet c\,X \circ d$ is a clause with three subgoals, *b*, *c X*, and *d*, where the hypotheses used by *b* are to the left of those used by the other two subgoals, and the hypotheses used by *c X* are to the right of those used by *d*; i.e. the ordered context consumed by the clause would look like $\omega_b, \omega_d, \omega_c$ where ω_b contains the hypotheses consumed by *b* and similar for *d* and *c*.

10.2 Encoding of UAOLL

Here is the signature for the object language:

trm : type.	*frm : type.*
atom : type	*atm : atom -> frm.*
place : trm -> atm.	*one : frm.*
# : frm -> frm -> frm.	*zero : frm.*
& : frm -> frm -> frm.	*top : frm.*
forall : (trm -> frm) -> frm.	*exists : (trm -> frm) -> frm.*
-» : frm -> frm -> frm.	*>-> : frm -> frm -> frm.*
−o : frm -> frm -> frm.	*–> : frm -> frm -> frm.*
** : frm -> frm -> frm.*	*<> : frm -> frm -> frm.*
gnab : frm -> frm.	*bang : frm -> frm.*

We assume that all of our object level binary operators are infix. *#*, * and <> are meant to represent $\oplus$, $\bullet$ and $\circ$.

The encoding of residuation directly transcribes the residuation rules in section 4; where the judgement

$$G_i\,;\,D \gg P \setminus G_o$$

is represented by the predicate *resid Gi D P Go* defined as follows:

resid : frm -> frm -> atm -> frm -> o.

resid G (atm P) P G.
resid G top P zero.

[4]The exact implication required in a clause is a function of what kinds of hypotheses the clause will need to access and whether the clause itself is ordered, e.g. unrestricted clauses may safely use $\twoheadrightarrow$ or $\rightarrowtail$.

resid G (D0 & D1) P (G0 # G1) ↞
  resid G D0 P G0 • *resid G D1 P G1.*
resid Gi (forall D) P (exists Go) ↞ $\forall y$. *resid Gi (D y) P (Go y).*
resid Gi (G -» D) P Go ↞ *resid (G <> Gi) D P Go.*
resid Gi (G >-> D) P Go ↞ *resid (G * Gi) D P Go.*
resid Gi (G −o D) P Go ↞ *resid (gnab G * Gi) D P Go.*
resid Gi (G −> D) P Go ↞ *resid (bang G * Gi) D P Go.*

Note that *resid* does not depend upon any linear nor ordered hypotheses.

Our encoding of the goal directed rules uses the meta-logic contexts (i.e. Olli's own contexts) to represent the object-logic contexts; i.e. there is no explicit representation of a context. We use the predicate *hyp* to lift object-logic formulas into the meta-logic context. Thus the judgement

$$\Gamma; \delta; \omega \vdash G$$

is simply represented by the predicate *goal G*, and each object hypothesis *e* is captured by a meta-logic hypothesis *hyp e*. The encoding of the rules themselves is then a direct transcription of the rules in sections 7 and 9.

hyp : frm -> o.

goal : frm -> o.

goal top ↞ ⊤.
goal (G0 & G1) ↞ *goal G0* & *goal G1.*
goal (G0 # G1) ↞ *goal G0* ⊕ *goal G1.*
goal (forall G) ↞ $\forall x$. *goal (G x).*
goal (exists G) ↞ *goal (G X).*
goal (D -» G) ↞ $\forall x$.
  hyp (atm (place x) -» D) ⊸ *hyp (atm (place x))* ↠ *goal G.*
goal (D >-> G) ↞ $\forall x$.
  hyp (atm (place x) -» D) ⊸ *hyp (atm (place x))* ↣ *goal G.*
goal (D −o G) ↞ *hyp D* ⊸ *goal G.*
goal (D −> G) ↞ *hyp D* → *goal G.*
goal (gnab G) ↞ ¡ *(goal G).*
goal (bang G) ↞ ! *(goal G).*
goal one ↞ 1.
*goal (G * H)* ↞ *goal G* • *goal H.*
goal (G <> H) ↞ *goal G* ∘ *goal H.*
goal (atm P) ↞ *hyp D* • *resid one D P G* • *goal G.*

The final *atm P* rule above first chooses a clause from the (implicit meta-logical) context, and then calls the residuation predicate on it; since we are using the meta-logic contexts, this one rule implements both $choice_\gamma$ and $choice_\delta$. Furthermore, since we are not distinguishing the *place* atom from other atoms, the *atm P* rule also does the job of the $choice_\omega$ rule where the *resid* clause will do the actual matching and return the new goal *one*.

11 CONCLUSION

We have presented UAOLL, an alternate formulation of UOLL amenable to direct implementation in Olli, and proved its correctness. UAOLL gives a provably correct meta-circular interpreter for Olli which makes use of the meta-logic contexts rather than explicitly representing the object-level contexts as terms.

We end with several directions for further work. There are several optimizations which could be made to UAOLL. We would like to investigate moving residuation to the implication rules (so that residuation only happens once) and recursively residuating subgoals to get a system closer to the logical compilation described in [3]. We would also like to investigate a more aggressive check that the placeholder atom is satisfied, i.e. checking the placeholder could be done at the same time as checking that the head matches the goal; this would remove the need for an explicit $choice_\omega$ rule.

The correctness of UAOLL with respect to UOLL is noteworthy since UAOLL has a demoted ordered context which only contains atoms. It would be interesting to see if all of OLL can be encoded into a corresponding system with a demoted ordered context.

Finally, it would be interesting to translate UAOLL into a more modern focussing system, e.g. [12] or [5].

ACKNOWLEDGMENTS

We thank Alberto Momigliano, Frank Pfenning and the anonymous reviewers for helpful feedback about both the content and presentation of this work.

REFERENCES

[1] Jean-Marc Andreoli. 1992. Logic Programming with Focusing Proofs in Linear Logic. *J. Log. Comput.* 2, 3 (1992), 297–347. https://doi.org/10.1093/logcom/2.3.297
[2] Iliano Cervesato. 1998. Proof-Theoretic Foundation of Compilation in Logic Programming Languages. In *Proceedings of the 1998 Joint International Conference and Symposium on Logic Programming — JICSLP'98*, J. Jaffar (Ed.). MIT Press, Manchester, UK, 115–129. http://www.stanford.edu/~iliano/papers/jicslp98.ps.gz
[3] Iliano Cervesato. 2012. An Improved Proof-theoretic Compilation of Logic Programs. *Theory Pract. Log. Program.* 12, 4-5 (Sept. 2012), 639–657. https://doi.org/10.1017/S1471068412000221
[4] Iliano Cervesato, Joshua S. Hodas, and Frank Pfenning. 1996. Efficient Resource Management for Linear Logic Proof Search. In *Proceedings of the Fifth International Workshop on Extensions of Logic Programming — ELP'96*, R. Dyckhoff, H. Herre, and P. Schroeder-Heister (Eds.). Springer-Verlag LNAI 1050, Leipzig, Germany, 67–81. http://www.cs.cmu.edu/~iliano/papers/elp96.ps.gz
[5] Kaustuv Chaudhuri. 2008. Focusing Strategies in the Sequent Calculus of Synthetic Connectives. In *Proceedings of the 15th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR '08)*. Springer-Verlag, Berlin, Heidelberg, 467–481. https://doi.org/10.1007/978-3-540-89439-1_33
[6] Joshua S. Hodas and Dale Miller. 1994. Logic Programming in a Fragment of Intuitionistic Linear Logic. *Information and Computation* 110, 2 (1994), 327–365. Extended abstract in the Proceedings of the Sixth Annual Symposium on Logic in Computer Science, Amsterdam, July 15–18, 1991.
[7] Joachim Lambek. 1958. The mathematics of sentence structure. *AMERICAN MATHEMATICAL MONTHLY* (1958), 154–170.
[8] Dale Miller, Gopalan Nadathur, Frank Pfenning, and Andre Scedrov. 1991. Uniform Proofs as a Foundation for Logic Programming. *Annals of Pure and Applied Logic* 51, 1–2 (1991), 125–157.
[9] Jeff Polakow. 2000. Linear logic programming with an ordered context. In *PPDP*. 68–79. https://doi.org/10.1145/351268.351277
[10] Jeff Polakow. 2001. *Ordered linear logic and applications.* Ph.D. Dissertation. Carnegie Mellon University.
[11] Jeff Polakow and Kwangkeun Yi. 2001. Proving Syntactic Properties of Exceptions in an Ordered Logical Framework. In *Functional and Logic Programming, 5th International Symposium, FLOPS 2001, Tokyo, Japan, March 7-9, 2001, Proceedings*. 61–77. https://doi.org/10.1007/3-540-44716-4_4
[12] Robert J. Simmons. 2012. *Substructural Logical Specifications.* Ph.D. Dissertation. Pittsburgh, PA, USA. Advisor(s) Pfenning, Frank. AAI3534965.