LinLaPeL: A Linear Logic Programming Language

1 LinLaPeL Abstract Syntax

$$x \in \mathit{Variable} \qquad f \in \mathit{Function} \qquad p \in \mathit{Predicate}$$

$$\begin{aligned}
t \in \mathit{Term} &::= x \mid f(\vec{t}) \\
g \in \mathit{Goal} &::= p(\vec{t}) \mid g_1 \otimes g_2 \mid g_1 \,\&\, g_2 \mid g_1 \oplus g_2 \mid a \multimap g \mid a \supset g \mid !g \mid \exists \vec{x}.g \mid \textsf{output} \\
a \in \mathit{Assumption} &::= p(\vec{t}) \mid p(\vec{t}) \circ\!- g \mid \&\vec{a} \mid \forall \vec{x}.a \\
\mathit{prog} \in \mathit{Program} &::= \vec{a} \vdash g
\end{aligned}$$

We have two forms of hypothetical implication: linear implication using $\multimap$ and nonlinear implication using $\supset$. The difference is that $\multimap$ makes the assumption a linear and $\supset$ makes the assumption a nonlinear. This format is easier to deal with than allowing escaped assumptions $!a$. The **output** goal is guaranteed to appear exactly once at the very end of the initial query. Thus, seeing an **output** goal during execution will signal the end of a proof. This restriction is convenient because we need to ensure at the end of a proof that we've successfully used all of the linear assumptions. We've added a new form of assumption $\&\vec{a}$, which is like $a_1 \,\&\, a_2$ except generalized to allow an arbitrary number of assumptions.

2 Judgements and Axioms

For judgements we'll use the following notation:

- $\Gamma = \vec{a}$ such that all $a \in \vec{a}$ are nonlinear.
- $\Delta = \vec{a}$ such that all $a \in \vec{a}$ are linear.
- $\bullet$ represents an empty sequence of assumptions.

Judgements are now of the form $\Gamma; \Delta \vdash g$. That is, a judgement now explicitly records linear and nonlinear assumptions separately. This change makes keeping track of which assumptions are linear and which assumptions are nonlinear easier. We'll also add the following **Absorption** axiom:

$$\frac{a, \Gamma; a, \Delta \vdash g}{a, \Gamma; \Delta \vdash g} \ \text{(Absorb)}$$

That is, we are allowed to copy an assumption from the nonlinear context to the linear context whenever we want. This axiom is a convenience in the formalism: it allows the inference rules to only look at the linear context because Absorb allows us to copy things from the nonlinear context to the linear context at will. Otherwise we would have to have two copies of each elimination rule, one that uses the linear context and one that uses the nonlinear context. We don't actually implement the Absorb rule in an interpreter, instead when we need to find an assumption to use we just look at the linear assumptions first (the ones in Δ), then the nonlinear assumptions (the ones in Γ).

3 Inference Rules

Goals	**Rules**
$p(\vec{t})$	$\dfrac{}{\Gamma; p(\vec{t}) \vdash p(\vec{t})}\ (\mathrm{Id})$ $\qquad$ $\dfrac{\Gamma; \Delta \vdash g}{\Gamma; p(\vec{t}) \subset g, \Delta \vdash p(\vec{t})}\ (\supset\mathrm{E})$ $\dfrac{\vec{t}_2[\vec{x} \mapsto \vec{t}_3] = \vec{t}_1}{\Gamma; \forall\vec{x}.p(\vec{t}_2) \vdash p(\vec{t}_1)}\ (\forall\mathrm{E}_{\mathrm{Id}})$ $\qquad$ $\dfrac{\vec{t}_2[\vec{x} \mapsto \vec{t}_3] = \vec{t}_1 \qquad \Gamma; \Delta \vdash g[\vec{x} \mapsto \vec{t}_3]}{\Gamma; \forall\vec{x}.p(\vec{t}_2) \subset g, \Delta \vdash p(\vec{t}_1)}\ (\forall\mathrm{E}_{\supset})$ $\dfrac{p(\vec{t}) \in \vec{a}}{\Gamma; \&\vec{a} \vdash p(\vec{t})}\ (\&\mathrm{E}_{\mathrm{Id}})$ $\qquad$ $\dfrac{p(\vec{t}) \subset g \in \vec{a} \qquad \Gamma; \Delta \vdash g}{\Gamma; \&\vec{a}, \Delta \vdash p(\vec{t})}\ (\&\mathrm{E}_{\supset})$ $\dfrac{\forall\vec{x}.p(\vec{t}_2) \in \vec{a} \qquad \vec{t}_2[\vec{x} \mapsto \vec{t}_3] = \vec{t}_1}{\Gamma; \&\vec{a} \vdash p(\vec{t}_1)}\ (\&\mathrm{E}_{\forall\mathrm{Id}})$ $\qquad$ $\dfrac{\forall\vec{x}.p(\vec{t}_2) \subset g \in \vec{a} \qquad \vec{t}_2[\vec{x} \mapsto \vec{t}_3] = \vec{t}_1 \qquad \Gamma; \Delta \vdash g[\vec{x} \mapsto \vec{t}_3]}{\Gamma; \&\vec{a}, \Delta \vdash p(\vec{t}_1)}\ (\&\mathrm{E}_{\forall\supset})$
$g_1 \otimes g_2$	$\dfrac{\Gamma; \Delta_1 \vdash g_1 \qquad \Gamma; \Delta_2 \vdash g_2}{\Gamma; \Delta_1, \Delta_2 \vdash g_1 \otimes g_2}\ (\otimes\mathrm{I})$
$g_1 \,\&\, g_2$	$\dfrac{\Gamma; \Delta \vdash g_1 \qquad \Gamma; \Delta \vdash g_2}{\Gamma; \Delta \vdash g_1 \,\&\, g_2}\ (\&\mathrm{I})$
$g_1 \oplus g_2$	$\dfrac{\Gamma; \Delta \vdash g_i \qquad i \in \{1, 2\}}{\Gamma; \Delta \vdash g_1 \oplus g_2}\ (\oplus\mathrm{I})$
$a \multimap g$	$\dfrac{\Gamma; a, \Delta \vdash g}{\Gamma; \Delta \vdash a \multimap g}\ (\multimap\mathrm{I})$
$a \supset g$	$\dfrac{a, \Gamma; \Delta \vdash g}{\Gamma; \Delta \vdash a \supset g}\ (\supset\mathrm{I})$
$!g$	$\dfrac{\Gamma; \bullet \vdash g}{\Gamma; \bullet \vdash !g}\ (!\mathrm{I})$
$\exists\vec{x}.g$	$\dfrac{\Gamma, \Delta \vdash g[\vec{x} \mapsto \vec{t}]}{\Gamma, \Delta \vdash \exists\vec{x}.g}\ (\exists\mathrm{I})$
output	$\dfrac{}{\Gamma; \bullet \vdash \textbf{output}}\ (\mathrm{Out})$

In the elimination rules for proving a goal $p(\vec{t})$, the assumption we use to prove the goal is removed from Δ when proving any premises—this removal is what enforces linearity. In the rule for proving a goal $a \multimap g$, assumption a is placed in the linear assumptions Δ; in the rule for proving a goal $a \supset g$, assumption a is placed in the nonlinear assumptions Γ. The goal $a \supset g$ is exactly the same as goal $!a \multimap g$, but this way we segregate the linear and nonlinear assumptions rather than mixing them together. The rule for goal $!g$ requires that Δ is empty, i.e., that there are no linear assumptions left to consume. The same is true of the rule for **output**, which signals the end of a proof—there should be no leftover linear assumptions at this point.

4 Handling Nondeterminism

The inference rules in Section 3 have the same problems with nondeterminism that the rules for LaPeL had, and they can be dealt with in the same way. However, there is a new source of nondeterminism in LinLaPeL that we need to deal with: splitting the linear assumptions for proving the goal $g_1 \otimes g_2$. The inference rule for this goal nondeterministically splits the linear assumptions into two pieces: Δ_1 to prove sub-goal g_1 and Δ_2 to prove sub-goal g_2. Having the interpreter nondeterministically split the linear assumptions means that it would have to try an exponential number of possible combinations, which is not tractable.

Solution. The solution for this problem is called "linear assumption passing style". The basic idea is that we pass *all* linear assumptions to the proof of g_1. During that proof we remove any linear assumptions that are consumed, and the proof returns the linear assumptions that were not consumed. Those remaining linear assumptions are then passed to the proof of g_2. Of course, there may be multiple ways to prove g_1 that consume different linear assumptions and thus pass on different linear assumptions to the proof of g_2; the interpreter still needs to explore all of those possible proofs.

This solution requires a fundamental reformulation of the inference rules, and this reformulation effects everything, not just the rule for $g_1 \otimes g_2$. Previously (i.e., in Section 3) proofs never returned anything, and certain rules (e.g., all of the elimination rules and also the $!g$ rule) required that Δ contains either no assumptions or exactly one assumption. These things are no longer true, because we're going to pass all of the possible linear assumptions to the proof and return all of the ones that we didn't need to use. We need to do this for all of the goals because they could contain sub-goals that use $\otimes$. Section 5 gives the new formulation of the inference rules for LinLaPeL that use linear assumption passing style.

5 Inference Rules Redux

We modify judgements to the form $\Gamma; \Delta_1 \vdash g \Rightarrow \Delta_2$ in order to indicate the "linear assumption passing style". Δ_1 is the set of linear assumptions passed into the proof of g and Δ_2 is the subset of those linear assumptions that were *not* consumed in that proof and are being returned. The new inference rules using these modified judgements are given below. There are five rules in particular that influence the implementation of an interpreter:

- $g_1 \otimes g_2$. The premises of this rule require that we pass information from the proof of g_1 to the proof of g_2. This means that Δ acts more like the equivalence relation than like Γ, because Γ is completely independent for each goal but the equivalence relation is passed from goal to goal as it is updated.
- g_1 & g_2. The premises of this rule require that the new Δ resulting from proving g_1 is the same as the new Δ resulting from proving g_2. Remember that linear assumption passing style is supposed to emulate nondeterministic splitting; we can't have split the original Δ in two different ways for proving g_1 and g_2.
- $a \multimap g$. The second premise of this rule requires that the linear assumption a that we added to Δ_1 in order to prove g has been consumed in that proof. This check is complicated by the fact that we can have multiple copies of the same linear assumption.
- $!g$. The premise of this rule requires that we prove g without using any linear assumptions. The conclusion requires that any linear assumptions that we had before doing this proof are passed on.
- **output**. Remember that the **output** goal signals the end of a proof; as such, there is nowhere left to pass any remaining linear assumptions. Therefore in order for this goal to succeed Δ must be empty.

Consider the following depictions of how information must flow in the interpreter for proving the above goals:

$$\square \xrightarrow{\Delta_1} g_1 \otimes g_2 \xrightarrow{\Delta_1} g_1 \cdots \xrightarrow{\Delta_2} g_2 \cdots \xrightarrow{\Delta_3} \square$$

$$\square \xrightarrow{\Delta_1} g_1 \,\&\, g_2 \xrightarrow{\Delta_1} g_1 \cdots \xrightarrow{\Delta_1,\Delta_2} g_2 \cdots \xrightarrow{\Delta_2,\Delta_3} \boxed{\Delta_2 \overset{?}{=} \Delta_3} \xrightarrow{\Delta_2} \square$$

$$\square \xrightarrow{\Delta_1} a \multimap g \xrightarrow{\Delta_2,\#a_{\Delta_1}} g \cdots \xrightarrow{\Delta_3,\#a_{\Delta_1}} \boxed{\#a_{\Delta_1} \overset{?}{\geq} \#a_{\Delta_3}} \xrightarrow{\Delta_3} \square$$

$$\square \xrightarrow{\Delta_1} !g \xrightarrow{\cdot} g \cdots \xrightarrow{\Delta_1} \square$$

Goals	**Rules**
$p(\vec{t})$	$\dfrac{}{\Gamma; p(\vec{t}), \Delta \vdash p(\vec{t}) \Rightarrow \Delta}\ (\mathrm{Id})$ $\qquad$ $\dfrac{\Gamma; \Delta_1 \vdash g \Rightarrow \Delta_2}{\Gamma; p(\vec{t}) \subset g, \Delta_1 \vdash p(\vec{t}) \Rightarrow \Delta_2}\ (\supset\mathrm{E})$ $\dfrac{\vec{t}_2[\vec{x} \mapsto \vec{t}_3] = \vec{t}_1}{\Gamma; \forall \vec{x}.p(\vec{t}_2), \Delta \vdash p(\vec{t}_1) \Rightarrow \Delta}\ (\forall\mathrm{E}_{\mathrm{Id}})$ $\qquad$ $\dfrac{\vec{t}_2[\vec{x} \mapsto \vec{t}_3] = \vec{t}_1 \qquad \Gamma; \Delta_1 \vdash g[\vec{x} \mapsto \vec{t}_3] \Rightarrow \Delta_2}{\Gamma; \forall \vec{x}.p(\vec{t}_2) \subset g, \Delta_1 \vdash p(\vec{t}_1) \Rightarrow \Delta_2}\ (\forall\mathrm{E}_{\supset})$ $\dfrac{p(\vec{t}) \in \vec{a}}{\Gamma; \&\vec{a}, \Delta \vdash p(\vec{t}) \Rightarrow \Delta}\ (\&\mathrm{E}_{\mathrm{Id}})$ $\qquad$ $\dfrac{p(\vec{t}) \subset g \in \vec{a} \qquad \Gamma; \Delta_1 \vdash g \Rightarrow \Delta_2}{\Gamma; \&\vec{a}, \Delta_1 \vdash p(\vec{t}) \Rightarrow \Delta_2}\ (\&\mathrm{E}_{\supset})$ $\dfrac{\forall \vec{x}.p(\vec{t}_2) \in \vec{a} \qquad \vec{t}_2[\vec{x} \mapsto \vec{t}_3] = \vec{t}_1}{\Gamma; \&\vec{a}, \Delta \vdash p(\vec{t}_1) \Rightarrow \Delta}\ (\&\mathrm{E}_{\forall\mathrm{Id}})$ $\qquad$ $\dfrac{\forall \vec{x}.p(\vec{t}_2) \subset g \in \vec{a} \qquad \vec{t}_2[\vec{x} \mapsto \vec{t}_3] = \vec{t}_1 \qquad \Gamma; \Delta_1 \vdash g[\vec{x} \mapsto \vec{t}_3] \Rightarrow \Delta_2}{\Gamma; \&\vec{a}, \Delta_1 \vdash p(\vec{t}_1) \Rightarrow \Delta_2}\ (\&\mathrm{E}_{\forall\supset})$
$g_1 \otimes g_2$	$\dfrac{\Gamma; \Delta_1 \vdash g_1 \Rightarrow \Delta_2 \qquad \Gamma; \Delta_2 \vdash g_2 \Rightarrow \Delta_3}{\Gamma; \Delta_1 \vdash g_1 \otimes g_2 \Rightarrow \Delta_3}\ (\otimes\mathrm{I})$
$g_1 \,\&\, g_2$	$\dfrac{\Gamma; \Delta_1 \vdash g_1 \Rightarrow \Delta_2 \qquad \Gamma; \Delta_1 \vdash g_2 \Rightarrow \Delta_2}{\Gamma; \Delta_1 \vdash g_1 \,\&\, g_2 \Rightarrow \Delta_2}\ (\&\mathrm{I})$
$g_1 \oplus g_2$	$\dfrac{\Gamma; \Delta_1 \vdash g_i \Rightarrow \Delta_2 \qquad i \in \{1, 2\}}{\Gamma; \Delta_1 \vdash g_1 \oplus g_2 \Rightarrow \Delta_2}\ (\oplus\mathrm{I})$
$a \multimap g$	$\dfrac{\Gamma; a, \Delta_1 \vdash g \Rightarrow \Delta_2 \qquad a \notin \Delta_2}{\Gamma; \Delta_1 \vdash a \multimap g \Rightarrow \Delta_2}\ (\multimap\mathrm{I})$
$a \supset g$	$\dfrac{a, \Gamma; \Delta_1 \vdash g \Rightarrow \Delta_2}{\Gamma; \Delta_1 \vdash a \multimap g \Rightarrow \Delta_2}\ (\supset\mathrm{I})$
$!g$	$\dfrac{\Gamma; \bullet \vdash g \Rightarrow \bullet}{\Gamma; \Delta \vdash !g \Rightarrow \Delta}\ (!\mathrm{I})$
$\exists \vec{x}.g$	$\dfrac{\Gamma, \Delta_1 \vdash g[\vec{x} \mapsto \vec{t}] \Rightarrow \Delta_2}{\Gamma, \Delta_1 \vdash \exists \vec{x}.g \Rightarrow \Delta_2}\ (\exists\mathrm{I})$
output	$\dfrac{}{\Gamma; \bullet \vdash \textbf{output} \Rightarrow \bullet}\ (\mathrm{Out})$