Lecture Notes on
Linear Logic

15-814: Types and Programming Languages
Frank Pfenning

Lecture 18
Tue Nov 11, 2025

1 Introduction

In this lecture we introduce linear logic in its intuitionistic variant in the context of this course. In the next lecture we discuss linear functional programming and its connection to garbage collection. We are thus moving in the opposite direction from the beginning in the course where we started with types and recognized the underlying logic. This time we start with the logic and then interpret linear propositions as linear types.

Linear logic is due to Girard [1987], including an intuitionistic variant somewhat different from ours Girard and Lafont [1987]. One closer to ours here with some additional materials (including a connection to the classical case) can be found in Chang et al. [2003].

Linear and other substructural cousins like affine logic have recently attracted attention in programming because the related type system form the basis for Rust [Klabnik et al., 2026], Linear Haskell [Bernardy et al., 2018], Oxidized OCaml [Lorenzen et al., 2024], Koka [Lorenzen et al., 2023], and Snax [Jang et al., 2024]. Separation logic [Reynolds, 2002], based on similar principles, and its implementation in Iris [Jung et al., 2015], have been used to reason about imperative programs using a heap, sometimes also using linear types directly as in the Verus system [Lattuada et al., 2023].

The fundamental idea underlying linear logic is that enforce that hypotheses are used *exactly once* in a proof. What this means precisely is the subject of this lecture. As a logic, this would be too weak to be interesting on its own, so we reintroduce the possibility of using some hypotheses multiple times (just as in ordinary logic), but in a controlled way using a modality $!A$ (pronounced *of course* A or *bang* A).

2 Localizing Hypotheses

Because we want to remain close to functional programming, we use *natural deduction*, which is a departure from Girard's choice of the sequent calculus [Gentzen, 1935, Girard, 1987]. Because we view *hypotheses as resources* it makes the rules more understandable if we record for every judgment in a deduction precisely which hypotheses are available. We return to the system of natural deduction from Lecture 17 and annotate each judgment with available hypotheses. For example:

$$\dfrac{\dfrac{\dfrac{\overline{A}^{\,x} \quad \overline{B}^{\,y}}{A \wedge B}\wedge I}{B \supset (A \wedge B)}\supset I^y}{A \supset (B \supset (A \wedge B))}\supset I^x \qquad \dfrac{\dfrac{\dfrac{\overline{x : A, y : B \vdash A}\;x \quad \overline{x : A, y : B \vdash B}\;y}{x : A, y : B \vdash A \wedge B}\wedge I}{x : A \vdash B \supset (A \wedge B)}\supset I^y}{\cdot \vdash A \supset (B \supset (A \wedge B))}\supset I^x$$

Adding the annotations here works bottom-up rather than top-down. That is, we record which hypotheses are available, rather than which are actually used. It should be clear that this annotation can be also be viewed as arising from the typing judgment for proof terms by erasing the proof terms. However, we retain the names of the hypotheses because we would still like to distinguish between the two (obvious) proofs of $A \supset (A \supset A)$.

In the rules we use Γ for the context of available hypotheses, just as in a typing judgment. For example, here are the rules for implication introduction and elimination.

$$\dfrac{\Gamma, x : A \vdash B}{\Gamma \vdash A \supset B}\supset I^x \qquad \dfrac{\Gamma \vdash A \supset B \quad \Gamma \vdash A}{\Gamma \vdash B}\supset E$$

3 Hypotheses as Resources

One intuition about linear logic is that hypotheses should be treated as resources. And that each resource should be used exactly once. The succedent in the judgment $\Gamma \vdash A$ *true* functions as a goal we want to achieve using the resources given in Γ. As customary, we omit "*true*" for brevity.

The hypothesis rule requires us to have exactly one resource.

$$\overline{x : A \vdash A}\;x$$

The proposition $A \otimes B$ (which is the linear analogue of $A \times B$) means that we have resources A and B. So when we want to prove $A \otimes B$ we have to split our resources, devoting some to achieving A and others to achieving B.

$$\dfrac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}\otimes I$$

The notation Γ, Δ represents the *disjoint union* of the two contexts. As always, there should not be any duplicate variables in Γ, Δ so that references to them are unique. Since the hypotheses are independent of each other, the order does not matter.

The corresponding elimination rule devotes some resources (Γ) to achieving $A \otimes B$ and then decomposes that to obtain A and B separately, together with the remaining resources.

$$\dfrac{\Gamma \vdash A \otimes B \quad \Delta, x : A, y : B \vdash C}{\Gamma, \Delta \vdash C}\otimes E^{x,y}$$

It is easy to check the soundness of this elimination rule if we are careful about substitution.

$$\dfrac{\dfrac{\begin{matrix} M \\ \Gamma_1 \vdash A\end{matrix} \quad \begin{matrix} N \\ \Gamma_2 \vdash B\end{matrix}}{\Gamma_1, \Gamma_2 \vdash A \otimes B}\otimes I \quad \begin{matrix} P \\ \Delta, x : A, y : B \vdash C\end{matrix}}{\Gamma_1, \Gamma_2, \Delta \vdash C}\otimes E^{x,y} \quad \longrightarrow_R \quad \begin{matrix} [M/x][N/y]P \\ \Gamma_1, \Gamma_2, \Delta \vdash C\end{matrix}$$

The substitution property has to account for proper use of resources.

Property 1 (Linear Substitution) *If* $\begin{array}{c} M \\ \Gamma \vdash A \end{array}$ *and* $\begin{array}{c} N \\ \Delta, x : A \vdash C \end{array}$ *then* $\begin{array}{c} [M/x]N \\ \Gamma, \Delta \vdash C \end{array}$

One should think of this as the defining property for a *linear hypothetical judgment*. The new proof $[M/x]N$ of C is obtained by substituting the proof M for every use of the linear hypothesis x in N.

As might be expected, we also have a form of implication, usually called *linear implication* $A \multimap B$. We reason as if we had A as a resource in order to prove B. Here, and in the remainder of the lecture, we remove the variable annotations on the introduction rules since the hypothetical judgment itself contains the new variable name.

$$\frac{\Gamma, x : A \vdash B}{\Gamma \vdash A \multimap B} \multimap I$$

We can think of a proof of $A \multimap B$ as a way to transform resource A into resource B. This is formalized in the elimination rule, again taking care to split resources between the two premises.

$$\frac{\Gamma \vdash A \multimap B \quad \Delta \vdash A}{\Gamma, \Delta \vdash B} \multimap E$$

We now have sufficient constructs to write out a small linear proof.

$$\begin{array}{c} \vdots \\ \cdot \vdash A \multimap (B \multimap (B \otimes A)) \end{array}$$

We start with two implication introductions, followed a tensor introduction. The split between the hypotheses for the latter is very easy to see.

$$\cfrac{\cfrac{\cfrac{\cfrac{}{y : B \vdash B}\, y \quad \cfrac{}{x : A \vdash A}\, x}{x : A, y : B \vdash B \otimes A} \otimes I}{x : A \vdash B \multimap (B \otimes A)} \multimap I}{\cdot \vdash A \multimap (B \multimap (B \otimes A))} \multimap I$$

4 Example: A Coin Exchange

To illustrate the interpretation of hypotheses as resources, we write out a small example of a coin exchange between quarters (q, worth 25 cents), dimes (d, worth 10 cents) and nickles (n, worth 5 cents). We have the following *axioms* describing fair exchanges:

$$\begin{array}{lcl} a_1 & : & q \multimap d \otimes d \otimes n \\ a_2 & : & d \otimes d \otimes n \multimap q \\ a_3 & : & d \multimap n \otimes n \\ a_4 & : & n \otimes n \multimap d \end{array}$$

With these axioms we can prove that a quarter and a nickel can be turned into three dimes:

$$\cdot \vdash q \otimes n \multimap d \otimes d \otimes d$$

After a few steps we arrive at

$$
\dfrac{\dfrac{\dfrac{}{x_1 : q \otimes n \vdash q \otimes n}\;x_1 \qquad \begin{array}{c}\vdots\\ x_2 : q, x_3 : n \vdash d \otimes d \otimes d\end{array}}{x_1 : q \otimes n \vdash d \otimes d \otimes d}\;\otimes E}{\cdot \vdash q \otimes n \multimap d \otimes d \otimes d}\;{\multimap}I
$$

At this point we cannot use the $\otimes I$ rule since we cannot split the hypotheses to prove d and $d \otimes d$ separately. Instead, we have to use the axiom a_1 to exchange the quarter for two dimes and a nickel.

$$
\dfrac{\dfrac{\dfrac{}{x_1 : q \otimes n \vdash q \otimes n}\;x_1 \qquad \dfrac{\dfrac{\dfrac{}{\cdot \vdash q \multimap d \otimes d \otimes n}\;a_1 \quad \dfrac{}{x_2 : q \vdash q}\;x_2}{x_2 : q \vdash d \otimes d \otimes n}\;{\multimap}E \times 2 \qquad \begin{array}{c}\vdots\\ x_3 : n, x_4 : d, x_5 : d, x_6 : n \vdash d \otimes d \otimes d\end{array}}{x_2 : q, x_3 : n \vdash d \otimes d \otimes d}\;\otimes E}{x_1 : q \otimes n \vdash d \otimes d \otimes d}\;\otimes E}{\cdot \vdash q \otimes n \multimap d \otimes d \otimes d}\;{\multimap}I
$$

Now we can use $\otimes I$ two times because a fair split between the resources is possible.

$$
\dfrac{\begin{array}{c}\vdots\\ x_3 : n, x_6 : n \vdash d\end{array} \quad \dfrac{}{x_4 : d \vdash d}\;x_4 \quad \dfrac{}{x_5 : d \vdash d}\;x_5}{x_3 : n, x_4 : d, x_5 : d, x_6 : n \vdash d \otimes d \otimes d}\;\otimes I \times 2
$$

The last unproven goal is easy established using axiom a_4.

$$
\dfrac{\dfrac{\dfrac{}{\cdot \vdash n \otimes n \multimap d}\;a_4 \quad \dfrac{\dfrac{}{x_3 : n \vdash n}\;x_3 \quad \dfrac{}{x_6 : n \vdash n}\;x_6}{x_3 : n, x_6 : n \vdash n \otimes n}\;\otimes I}{x_3 : n, x_6 : n \vdash d}\;{\multimap}E \quad \dfrac{}{x_4 : d \vdash d}\;x_4 \quad \dfrac{}{x_5 : d \vdash d}\;x_5}{x_3 : n, x_4 : d, x_5 : d, x_6 : n \vdash d \otimes d \otimes d}\;\otimes I \times 2
$$

A few notes on this proof. It is a bit awkward, but that's often the case if we limit ourselves to the primitive rules of inference. A more efficient way would be to turn the axioms into derived rules of inference and then use these. Another point is that we need axioms that can be used arbitrarily often (including not at all), while the hypotheses must be used exactly once. We return to this point when introducing $!A$. Finally, one can prove as a property of the system of rules we have that linear implications between combinations of coins

$$c_1 \otimes \ldots \otimes c_n \multimap c'_1 \otimes \ldots \otimes c'_k$$

can be derived precisely if the total value of the antecedent is equal to the total value of the succedent.

Before we move on, we can derive the rules for the nullary version of $A \otimes B$, written 1. While $A \otimes B$ represents two resources, 1 represents no resources. So we have:

$$
\dfrac{}{\cdot \vdash 1}\;1I \qquad\qquad \dfrac{\Gamma \vdash 1 \quad \Delta \vdash C}{\Gamma, \Delta \vdash C}\;1E
$$

5 Choice

The connectives so far, $A \otimes B$, 1, and $A \multimap B$ are called *multiplicative*. There is a different class of *additive connectives* that represent a kind of choice. For examples, with two dimes and a nickel I can obtain a quarter, but I can also obtain five nickels. That is,

$$\begin{array}{rcl} x_1 : d, x_2 : d, x_3 : n & \vdash & q \\ x_1 : d, x_2 : d, x_3 : n & \vdash & n \otimes n \otimes n \otimes n \otimes n \end{array}$$

So I can achieve both q and also n^5 (which means it is a form of conjunction), but not at the same time. We call it *alternative conjunction*, *additive conjunction*, or *external choice* and write $A \mathbin{\&} B$.

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \mathbin{\&} B}\ \& I$$

In the example above we would conclude:

$$x_1 : d, x_2 : d, x_3 : n \vdash q \mathbin{\&} (n \otimes n \otimes n \otimes n \otimes n)$$

On the face of it this would seem to violate linearity, because in the premise of $\& I$ we write Γ in both branches. To make sure this is correct a user of resource $A \mathbin{\&} B$ has to choose between between A and B and cannot get both. That is:

$$\frac{\Gamma \vdash A \mathbin{\&} B}{\Gamma \vdash A}\ \& E_1 \qquad \frac{\Gamma \vdash A \mathbin{\&} B}{\Gamma \vdash B}\ \& E_2$$

These rules are indeed sound. Writing out the first of two local reductions:

$$\frac{\dfrac{\begin{array}{c} M \\ \Gamma \vdash A \end{array} \quad \begin{array}{c} N \\ \Gamma \vdash B \end{array}}{\Gamma \vdash A \mathbin{\&} B}\ \& I}{\Gamma \vdash A}\ \& E_1 \quad \longrightarrow_R \quad \begin{array}{c} M \\ \Gamma \vdash A \end{array}$$

We can see that it is important that Γ is propagated unchanged to both premises of $\wedge I$. The resources in Γ can hypothetically be used to achieve A and also to achieve B, but only one of these two constructions can actually ever be used.

We also observe that the introduction and elimination rules look exactly the same as those for $A \wedge B$ in intuitionistic natural deduction. But there is still a hidden difference in that the turnstile here represents a linear hypothetical judgment, while in intuitionstic natural deduction it represents a (ordinary, that is, nonlinear) hypothetical judgment.

We also have the nullary additive conjunctions $\top$ with no elimination rule, reflecting that the binary additive conjunction has two eliminations.

$$\frac{}{\Gamma \vdash \top}\ \top I \qquad \text{(no } \top E \text{ rule)}$$

There is also an *additive disjunction* which is also called an *internal choice* and written as $A \oplus B$. In this case, we have two introduction rules.

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \oplus B}\ \oplus I_1 \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \oplus B}\ \oplus I_2$$

It is called *internal choice* because as we construct this proof we can make the choice between the two disjuncts. Conversely, if we have a resource $A \oplus B$ we have to be prepared for either option, exactly one of which will be true.

$$\frac{\Gamma \vdash A \oplus B \quad \Delta, x : A \vdash C \quad \Delta, y : B \vdash C}{\Gamma, \Delta \vdash C} \oplus E$$

Because only one of A or B will be true, the second and third premise share the same Δ. It is an instructive exercise to write out the local reduction to convince yourself that this sharing, and also the splitting between Γ and Δ are correct and this rule is sound.

Finally, there is a nullary disjunction written as 0.

$$\text{(no } 0I \text{ rule)} \qquad \frac{\Gamma \vdash 0}{\Gamma, \Delta \vdash C} \, 0E$$

This rule again looks strange, but properly reflects 0 as the internal choice between zero alternatives.

6 Of Course!

In our coin exchange example, the axioms define the rules of the exchange and as such can be used arbitrarily rather then be subject to the "must-be-used-exactly-once" discipline. We can reflect this back into the logic by a new form of proposition $!A$ pronounced "*of course* A" or "*bang* A". This should be true if producing A requires no resources, that is, if we can prove A from the empty context. Thus, in the first approximation, we have

$$\frac{\cdot \vdash A}{\cdot \vdash !A} \, !I$$

It shares this rule with modal logic where it is called *necessitation* and allows us to conclude $\Box A$ if we can prove A from no assumption. In modal logic, the distinction is between the possibly contingent truth of A and the universal validity of A. A typical example would be $\Box(A \supset A)$. Here, similarly, we have $!(A \multimap A)$ because the proof of $A \multimap A$ does not require any resources.

We get the cleanest and most flexible metatheory if we treat A *valid* as a first-class judgment and distinguish two forms of hypotheses: the valid ones and the merely true ones. We write

$$\underbrace{\Gamma}_{\textit{valid}} \; ; \underbrace{\Delta}_{\textit{true}} \vdash \underbrace{A}_{\textit{true}}$$

Because valid assumptions require no resources, they are allowed in the $!I$ rule.

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

The corresponding elimination rule adds a validity assumption on A.

$$\frac{\Gamma \, ; \Delta_1 \vdash !A \quad \Gamma, u : A \, ; \Delta_2 \vdash C}{\Gamma \, ; \Delta_1, \Delta_2 \vdash C} \, !E$$

The valid hypotheses can be used arbitrarily and are therefore propagated to all premises of all rules. We can use hypotheses from either of the two contexts.

$$\frac{}{\Gamma\,;\,x : A \vdash A}\,x \qquad \frac{u : A \in \Gamma}{\Gamma\,;\,\cdot \vdash A}\,u$$

This formulation is often called *dual intuitionstic linear logic* because of the two contexts [Barber, 1996, Chang et al., 2003].

Now we can prove some characteristic axioms of necessity, translated to the linear setting.

$$\begin{array}{lcl} \cdot\,;\,\cdot & \vdash & !(A \multimap B) \multimap\, !A \multimap\, !B \\ \cdot\,;\,\cdot & \vdash & !A \multimap A \\ \cdot\,;\,\cdot & \vdash & !A \multimap\, !!A \end{array}$$

Critical is that we **can not** prove in general

$$\cdot\,;\,\cdot \vdash A \multimap\, !A$$

although it may hold for some particular propositions A. Such a theorem would say that if we had resource A we could arbitrarily replicate it. Clearly, we can't and shouldn't be able to.

7 The Structural Rules

Girard [1987] took a somewhat different approach to $!A$ which he called the *exponential modality*. He didn't distinguish two contexts, but allowed certain structural rules to be directly applied to propositions of the form $!A$. His system was a classical sequent calculus, so it doesn't exactly match our form of natural deduction. Transposed into an intuitionistic sequent calculus, we would have the following rules:

$$\frac{!\Gamma \vdash A}{!\Gamma \vdash\, !A}\,!R \qquad \frac{\Gamma, A \vdash C}{\Gamma, !A \vdash C}\,!L$$

$$\frac{\Gamma \vdash C}{\Gamma, !A \vdash C}\,\mathsf{weakening} \qquad \frac{\Gamma, !A, !A \vdash C}{\Gamma, !A \vdash C}\,\mathsf{contraction}$$

A significant design decision here is to keep all hypothesis linear, except that those are explicitly marked as valid may be dropped (rule weakening) or duplicated (rule contraction). The rules are called weakening and contraction because the name expresses how they would be read from the premise to the conclusion.

We call these rules *structural rules* because they are relevant to the structure of context and apply uniformly, independently of the form of the proposition A. Purely linear logic (without the exponential) has no structural rules. *Affine logic* allows weakening, but not contraction. That is, even unmarked hypotheses can be dropped, but they may not be duplicated. Hypotheses can be used **at most once**. In *strict logic* hypotheses must be used **at least once**. This means we have the general rule of contraction but not the rule of weakening. We can recover ordinary intuitionistic logic by allowing both weakening and contraction. The rules then have a different form compared to the ones we presented in Lecture 17. As observed by Girard and Lafont [1987] for a different formulation, by making structural rules explicit we can interpret weakening as deallocation and contraction as aliasing. We will sharpen these observation in the next lecture.

8 Example: A Set Menu

As a last example that illustrates the various connectives, we translate a set menu into (linear) logical axiom.

$\$100 \multimap$		for $100 you get
	$(\mathsf{soup} \mathbin{\&} \mathsf{salad})$	soup or salad, your choice
$\otimes$	$(\mathsf{chicken} \mathbin{\&} \mathsf{beef})$	chicken or beef, your choice
$\otimes$	$(\mathsf{chanterelles} \oplus \mathsf{asparagus})$	chanterelles or asparagus, chef's choice
$\otimes$	$(\mathbf{1} \mathbin{\&} (\$20 \multimap \mathsf{cake}))$	optionally, cake for an additional $20
$\otimes$	$!\mathsf{coffee}$	and unlimited coffee

References

Andrew Barber. Dual intuitionistic linear logic. Technical Report ECS-LFCS-96-347, Department of Computer Science, University of Edinburgh, September 1996.

Jean-Philippe Bernardy, Mathieu Boespflug, Ryan R. Newton, Simon Peyton Jones, and Arnaud Spiwack. Linear haskell. In *Proceedings of the ACM on Programming Languages*, volume 2 (POPL 2018), page pp. 36, 2018.

Bor-Yuh Evan Chang, Kaustuv Chaudhuri, and Frank Pfenning. A judgmental analysis of linear logic. Technical Report CMU-CS-03-131R, Carnegie Mellon University, Department of Computer Science, December 2003.

Gerhard Gentzen. Untersuchungen über das logische Schließen. *Mathematische Zeitschrift*, 39:176–210, 405–431, 1935. English translation in M. E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, pages 68–131, North-Holland, 1969.

Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

Jean-Yves Girard and Yves Lafont. Linear logic and lazy computation. In H. Ehrig, R. Kowalski, G. Levi, and U. Montanari, editors, *Proceedings of the International Joint Conference on Theory and Practice of Software Development*, volume 2, pages 52–66, Pisa, Italy, March 1987. Springer-Verlag LNCS 250.

Junyoung Jang, Sophia Roshal, Frank Pfenning, and Brigitte Pientka. Adjoint natural deduction. In Jakob Rehof, editor, *9th International Conference on Formal Structures for Computation and Deduction (FSCD 2024)*, pages 15:1–15:23, Tallinn, Estonia, July 2024. LIPIcs 299. Extended version available as https://arxiv.org/abs/2402.01428.

Ralf Jung, David Swasey, Filip Sieczkowski, Kasper Svendsen, Aaron Turon, Lars Birkedal, and Derek Dreyer. Iris: Monoids and invariants as an orthogonal basis for concurrent reasoning. In *POPL'15*, pages 637–650, Mumbai, India, January 2015. ACM.

Steve Klabnik, Carol Nichols, and Chris Krycho. *The Rust Programming Language*. No Starch Press, 3rd edition edition, February 2026.

Andrea Lattuada, Travis Hance, Chanhee Cho, Matthias Brun, Isitha Subasinghe, Yi Zhou, Jon Howell, Bryan Parno, and Chris Hawblitzel. Verus: Verifying Rust programs using linear ghost types. In *Proceedings of the ACM on Programming Languages*, volume 7 (OOPSLA 2023), pages 286–315, April 2023. Extended version available as https://arxiv.org/abs/2303.05491.

Anton Lorenzen, Daan Leijen, and Wouter Swierstra. FP^2: Fully in-place functional programming. In *Proceedings of the ACM on Programming Languages*, volume 7 (ICFP), pages 275–304, 2023.

Anton Lorenzen, Leo White, Stephen Dolan, Richard A. Eisenberg, and Sam Lindley. Oxidizing OCaml with modal memory management. In *Proceedings of the ACM on Programming Languages*, volume 8 (ICFP), pages 485–514, 2024.

John C. Reynolds. Separation logic: A logic for shared mutable data structures. In *Proceedings of the 17th Symposium on Logic in Computer Science*, pages 55–74, Copenhagen, Denmark, July 2002. IEEE Computer Society.