Linear Logic

Alwen Tiu

Australian National University

ANU Logic Summer School
Lecture 2, December 12, 2007

Linear logic

Linear logic (Girard, 1987) departs from intuitionistic and classical logic by treating formula as *resources*. For example, suppose that with \$2 (A) we can buy a chocolate bar (B) and a can of softdrink (C) also costs \$2. This is formalised as follows:

$$(A \supset B) \wedge (A \supset C).$$

In classical or intuitionistic logic we may conclude that with \$2, we can buy both the chocolate and the softdrink ($A \supset B \wedge C$), because the implication

$$(A \supset B) \wedge (A \supset C) \supset (A \supset B \wedge C).$$

holds in classical/intuitionistic logic. Of course, this is certainly not the case in the materialistic world.

Multiplicative and addivitive connectives

- Linear logic refines, or rather splits, the usual classical/intuitionistic connectives into the *multiplicative* parts (which concerns about resource) and the *additive* parts (which concerns about truth).
- Conjunction splits into the multiplicative conjunction $\otimes$ ('tensor') and the additive conjunction & ('with'), disjunction becomes $⅋$ ('par') and $\oplus$ ('plus'). The implication $\supset$ becomes the linear implication $\multimap$.
- When we reason strictly about use of resource, the previous classical formula is interpreted as:

$$(A \multimap B) \otimes (A \multimap C) \multimap (A \multimap B \otimes C)$$

  which is *not* valid in linear logic.
- If we interpret it additively, we have

$$(A \multimap B) \& (A \multimap C) \multimap (A \multimap B \& C)$$

  which is valid in linear logic.

Unbounded resource

Linear logic is intended to be a refinement of both classical and intuitionistic logic. Therefore, one has to be able to recover principles of classical/intuitionistic reasoning. For example, in classical logic, propositions are inexhaustible resource: we can use a theorem as many time as we want without worrying about "running out of theorems"!

The ability to designate some resource (formula) as unbounded is captured in linear logic by the introduction of the modal operators ! and its dual ?. The following hold in linear logic:

$$!A \multimap !A \otimes !A \qquad ?A \parr ?A \multimap ?A$$

We can duplicate $!A$ (in antecedent position) and $?A$ (in succedent position) as many times as we want.

Additive rules

Logical rules for binary connectives are divided into ‘additive’ rules (which copy contexts) and ‘multiplicative’ rules (which split contexts).
Additive rules:

$$\frac{\Delta, B_i \longrightarrow \Gamma}{\Delta, B_1 \& B_2 \longrightarrow \Gamma}\ \&_L \qquad \frac{\Delta \longrightarrow B, \Gamma \quad \Delta \longrightarrow C, \Gamma}{\Delta \longrightarrow B \& C, \Gamma}\ \&_R$$

$$\frac{\Delta \longrightarrow B_i, \Gamma}{\Delta \longrightarrow B_1 \oplus B_2, \Gamma}\ \oplus_R \qquad \frac{\Delta, B \longrightarrow \Gamma \quad \Delta, C \longrightarrow \Gamma}{\Delta, B \oplus C \longrightarrow \Gamma}\ \oplus_L$$

Multiplicative rules

$$\frac{\Delta, B_1, B_2 \longrightarrow \Gamma}{\Delta, B_1 \otimes B_2 \longrightarrow \Gamma}\ \otimes_L \qquad \frac{\Delta_1 \longrightarrow B, \Gamma_1 \quad \Delta_2 \longrightarrow C, \Gamma_2}{\Delta_1, \Delta_2 \longrightarrow B \otimes C, \Gamma_1, \Gamma_2}\ \otimes_R$$

$$\frac{\Delta \longrightarrow B_1, B_2, \Gamma}{\Delta \longrightarrow B_1 \parr B_2, \Gamma}\ \parr_R \qquad \frac{\Delta_1, B \longrightarrow \Gamma_1 \quad \Delta_2, C \longrightarrow \Gamma_2}{\Delta_1, \Delta_2, B \parr C \longrightarrow \Gamma_1, \Gamma_2}\ \parr_R$$

The distinction between additive and multiplicative rules disappears if we allow unrestricted contraction/weakening.

Units

There are four units (analogous to 'true' and 'false' in classical logic): $\top$, $\mathbf{0}$, $\mathbf{1}$ and $\bot$.

$$\frac{}{\Delta \longrightarrow \top, \Gamma}\ \top_R \qquad \frac{\Delta \longrightarrow \Gamma}{\Delta, \mathbf{1} \longrightarrow \Gamma}\ \mathbf{1}_L \qquad \frac{}{\longrightarrow \mathbf{1}}\ \mathbf{1}_R$$

$$\frac{}{\mathbf{0}, \Delta \longrightarrow \Gamma}\ \mathbf{0}_L \qquad \frac{\Delta \longrightarrow \Gamma}{\Delta \longrightarrow \bot, \Gamma}\ \bot_R \qquad \frac{}{\bot \longrightarrow}\ \bot_L$$

One can prove:

$$\mathbf{1} \otimes A \equiv A \qquad \bot \invamp A \equiv A \qquad \top \& A \equiv A \qquad \mathbf{0} \oplus A \equiv A$$

Contraction and weakening

The modal operators ? and ! (also called the 'exponentials') are used to control contraction and weakening.

$$\frac{\Delta \longrightarrow \Gamma}{\Delta, !B \longrightarrow \Gamma}\ !W \qquad \frac{\Delta, !B, !B \longrightarrow \Gamma}{\Delta, !B \longrightarrow \Gamma}\ !C \qquad \frac{\Delta, B \longrightarrow \Gamma}{\Delta, !B \longrightarrow \Gamma}\ !D$$

$$\frac{\Delta \longrightarrow \Gamma}{\Delta \longrightarrow ?B, \Gamma}\ ?W \qquad \frac{\Delta \longrightarrow ?B, ?B, \Gamma}{\Delta \longrightarrow ?B, \Gamma}\ ?C \qquad \frac{\Delta \longrightarrow B, \Gamma}{\Delta \longrightarrow ?B, \Gamma}\ ?D$$

$$\frac{!\Delta \longrightarrow B, ?\Gamma}{!\Delta \longrightarrow !B, ?\Gamma}\ !_R \qquad \frac{!\Delta, B \longrightarrow ?\Gamma}{!\Delta, ?B \longrightarrow ?\Gamma}\ ?_L$$

Linear negation

$$\frac{\Delta \longrightarrow B, \Gamma}{\Delta, B^\perp \longrightarrow \Gamma}\ not_L \qquad \frac{\Delta, B \longrightarrow \Gamma}{\Delta \longrightarrow B^\perp, \Gamma}\ not_R$$

Linear negation obeys the de Morgan duality laws:

$$(A \& B)^\perp \equiv A^\perp \oplus B^\perp \qquad (A \otimes B)^\perp \equiv A^\perp ⅋ B^\perp. \qquad (?A)^\perp \equiv !(A^\perp)$$

We also have $(A^\perp)^\perp \equiv A$, just like in classical logic.

Linear implication

The linear implication, $\multimap$, is a derived connective, i.e.,

$$A \multimap B := A^{\perp} \parr B.$$

Its introduction rules:

$$\frac{\Delta \longrightarrow B, \Gamma \quad \Delta', C \longrightarrow \Gamma'}{\Delta, \Delta', B \multimap C \longrightarrow \Gamma, \Gamma'} \multimap_L \qquad \frac{\Delta, B \longrightarrow C, \Gamma}{\Delta \longrightarrow B \multimap C, \Gamma} \multimap_R$$

We can alternatively take $\multimap$ as primitive and define linear negation as follows:

$$A^{\perp} := A \multimap \perp.$$

One-sided sequent calculus

Since linear negation is involutive, we can transform every formula A into a *negation normal form*, denoted by $\bar{A}$. The sequent rules can then be simplified by considering only the right-introduction rules, e.g.,

$$\frac{}{\longrightarrow A, \bar{A}}\ id \qquad \frac{\longrightarrow A, \Gamma \quad \longrightarrow B, \Delta}{\longrightarrow A \otimes B, \Gamma, \Delta}\ \otimes$$

Every proof of a two-sided sequent $\Gamma \longrightarrow \Delta$ can be translated into a proof of the one-sided sequent

$$\longrightarrow \overline{\Gamma^{\perp}}, \overline{\Delta}.$$

A restaurant analogy

The following analogy is useful to understand the resource-sensitive nature of linear logic:

> *Suppose for a fixed price of $5, a fastfood restaurant will provide a hamburger, a Coke, and as many fries as you like, onion soup or salad (your choice), and pie or ice cream (depending on availability). One may encode this information in the linear formula:*
>
> $$(D \otimes D \otimes D \otimes D \otimes D) \multimap [H \otimes C \otimes !F \otimes (O \& S) \otimes (P \oplus I)]$$

Cut elimination: problematic cases

- Measures for cut reduction: size of cut formulas and height of derivations above cuts.
- Cut reductions for binary connectives are straightforward. The difficult cases involve exponentials, e.g.,

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ !\Gamma_1 \longrightarrow ?\Delta_1, A\end{array}}{!\Gamma_1 \longrightarrow ?\Delta_1, !A}\ !R \qquad \dfrac{\begin{array}{c}\vdots\\ !A, !A, \Gamma_2 \longrightarrow \Delta_2\end{array}}{!A, \Gamma_2 \longrightarrow \Delta_2}\ !C}{!\Gamma_1, \Gamma_2 \longrightarrow ?\Delta_1, \Delta_2}\ cut$$

Generalized cuts

To deal with cases involving exponentials, we make use of two generalized cut rules:

$$\frac{\Gamma \longrightarrow \Delta, !A \quad (!A)^n, \Gamma' \longrightarrow \Delta'}{\Gamma, \Gamma' \longrightarrow \Delta, \Delta'}\ cut!$$

$$\frac{\Gamma \longrightarrow \Delta, (?A)^n \quad ?A, \Gamma' \longrightarrow \Delta'}{\Gamma, \Gamma' \longrightarrow \Delta, \Delta'}\ cut!$$

Here B^n denotes n-copies of B. In both rules, $n > 1$.

Generalized-cut reduction

$$\dfrac{\dfrac{\dfrac{\vdots}{!\Gamma_1 \longrightarrow ?\Delta_1, A}}{!\Gamma_1 \longrightarrow ?\Delta_1, !A}\ !R \qquad \dfrac{\dfrac{\vdots}{(!A)^{n+1}, \Gamma_2 \longrightarrow \Delta_2}}{(!A)^n, \Gamma_2 \longrightarrow \Delta_2}\ !C}{!\Gamma_1, \Gamma_2 \longrightarrow ?\Delta_1, \Delta_2}\ cut$$

reduces to

$$\dfrac{\dfrac{\dfrac{\vdots}{!\Gamma_1 \longrightarrow ?\Delta_1, A}}{!\Gamma_1 \longrightarrow ?\Delta_1, !A}\ !R \qquad \dfrac{\vdots}{(!A)^{n+1}, \Gamma_2 \longrightarrow \Delta_2}}{!\Gamma_1, \Gamma_2 \longrightarrow ?\Delta_1, \Delta_2}\ cut$$

Embedding of intuitionistic logic

$$A^\circ = A, \text{ if } A \text{ is atomic}$$

$$(A \wedge B)^\circ = A^\circ \& B^\circ \qquad (A \vee B)^\circ = (!A^\circ) \oplus (!B^\circ)$$

$$(A \supset B)^\circ = (!A^\circ) \multimap B^\circ \qquad \bot^\circ = \mathbf{0}$$

$$(\forall x.A)^\circ = \forall x.A^\circ \qquad (\exists x.A)^\circ = \exists x.!A^\circ$$

Theorem

$\Gamma \vdash_{IL} A$ *if and only if* $\Gamma^\circ \vdash_{LL} A^\circ$.

Embedding of classical logic

Classical logic can be embedded into linear logic via a detour through intuitionistic embedding. However, there is a more direct embedding.

$$A^+ = A^- = A, \text{ when } A \text{ is atomic}$$

$$(\neg A)^+ = (A^-)^\perp \qquad (\neg A)^- = (A^+)^\perp$$

$$(A \vee B)^+ = A^+ \oplus B^+ \qquad (A \vee B)^- = !(A^-) \parr !(B^-)$$

$$(A \wedge B)^+ = ?(A^+) \otimes ?(B^+) \qquad (A \wedge B)^- = A^- \& B^-$$

$$(\forall x.A)^+ = \forall x.?(A^+) \qquad (\forall x.A)^- = \forall x.A^-$$

$$(\exists x.A)^+ = \exists x.A^+ \qquad (\exists x.A)^- = \exists x.!(A^-)$$

Theorem

$\Gamma \vdash_{CL} \Delta$ *if and only if* $!(\Gamma^-) \vdash_{LL} ?(\Delta^-)$.

Permutation of inference rules

- Given a sequent $\Gamma \longrightarrow \Delta$, there may be more than one applicable rules. Certain orders of rules application may lead to successful proof search, while others may stuck.
- Some of these rules can be applied in any order without destroying provablity.
- We say that a rule R permutes over another rule R' if whenever there is a proof where R appears directly above R', we can find another proof where R appears below R'.
- The study of rule permutation is important for designing proof search strategy, and can be used to reduce non-determinism in proof search.

Invertible rules

- A rule is invertible if provability of its conclusion implies provability of its premises. For example, the $\parr_R$ rule is invertible:

  $\Gamma \longrightarrow A \parr B, \Delta$ *is provable if and only if* $\Gamma \longrightarrow A, B, \Delta$ *is provable.*

- Another characterisation of invertible rules is that we can always apply the rules without losing provability.
- Invertibility of rules can be shown by permutation argument, i.e., showing that it permutes down under any other rule.
- Other invertible rules: $\&_R$, $\oplus_L$, $\forall_R$, $\exists_L$.

Focussed proofs

- Andreoli (1992) came up with a proof search strategy called ‘focussing’ which is complete for linear logic.
- Focussed proofs proceed in two interleaving phases: the *asynchronous* phase and the *synchronous* phase.
- Asynchronous phase corresponds to applications of invertible rules and synchronous the non-invertible ones.
- We need to constraint slightly the sequent rules of linear logic to make the focussing strategy work.

Andreoli's dyadic sequents

Andreoli considered sequents of the form $\longrightarrow \Gamma : \Delta$, whose interpretation (in one-sided sequent) is

$$\longrightarrow ?\Gamma, \Delta.$$

Some rules for dyadic sequents:

$$\frac{}{\longrightarrow \Gamma : A, \bar{A}}\ id \qquad \frac{\longrightarrow \Gamma : \Delta, A \quad \longrightarrow \Gamma : \Delta', \bar{A}}{\longrightarrow \Gamma : \Delta, \Delta'}\ cut$$

$$\frac{\longrightarrow \Gamma, A : \Delta}{\longrightarrow \Gamma : \Delta, ?A}\ ? \qquad \frac{\longrightarrow \Gamma, F : \Delta, F}{\longrightarrow \Gamma, F : \Delta}\ absorb \qquad \frac{\longrightarrow \Gamma : A}{\longrightarrow \Gamma : !A}\ !$$

The introduction rule for ? *is* invertible in this proof system.

Focussed proof search

In dyadic sequent system, we can partition the connectives into two dual sets:

- the *asynchronous connectives*: $\bot$, $⅋, ?, \top, \&, \forall$,
- and the *synchronous connectives*: $\mathbf{1}, \otimes, !, \mathbf{0}, \oplus, \exists$.

Proof search strategy:

- Decompose all asynchronous formulas (in any order).
- When all asynchronous formulas have been decomposed, select any synchronous formula and *focus* on it, i.e., subsequently select its subformula and work on it until asynchronous formula appears.

Proof nets

- As in classical/intuitionistic logic, sequent proofs of linear logic formulas tend to be bureaucratic. *Proof nets* are introduced to reflect better the "essence" of proofs. Proof nets are graph structures representing sequent proofs, which identify some useless permutation of inference rules.
- Proof nets are formula trees with links between dual atomic formulas on the leaves. Example: construct a proof net for $A \otimes B \multimap B \otimes A$.
- The study of proof nets are important for two reasons (among others):
  - It is an attempt to answer the ultimate question of proof theory: what is a proof? Some believe there is an underlying geometrical symmetry in proofs. This is embodied in the "Geometry of Interaction" started by Girard.
  - There is a close connection with computation. Proof nets corresponds to (in a Curry-Howard style) concurrent computations. They are also shown to be useful to study *optimal reduction* in λ-calculus.