Differential Linear Logic and Bounded Time Complexity

Riley Moriss

April 23, 2023

Contents

Linear Logic

History and Philosophy

Why study logic? What is the study of logic? Logic is the "science" of reasoning. It seems there are many ways to take this seriously. Maybe the most successful was "classical logic", there have been successful critiques against it but it seems largely untouched.

Regardless it seems that the most immediate way to study logic is to write one down and then examine its properties. One makes observations about what they think is correct argumentation or the nature that we can "a priori" combine truths or some such thing, and then codefies them as formal systems. By nature we want to abstract forms from arguments or truths to see what they have in common and deduce logic. Now once someone has written such a thing down it seems that if you take it seriously as capturing the essence of truth and its manipulations then studying it and its properties is quite important. What can and cant be proved in the system, how can such and so be proved. What form does a correct argument take, are there equivilent arguments for the same thing. etc. From this point of view. One then becomes interested in proofs as the bearers of truths. One wants to know how complicated a proof of something will be etc. You get the domains of logic and proof theory. This process somewhat lead to two main logics "classical" and "intuitionistic". LL in a sense contains them both, moreover it allows more delicate control over the structural rules of them. In this way LL can be seen as a tool for proving things about the logics that exist inside of it and that people for some reason care about (they capture the one true logical method or something).

Another point of veiw may be that Linear Logic is itself the one true logic (or a better attempt at it), that it better mirrors the essence of human reason, or the reason of god (i.e. the logic that is latent in the universe whatever that means). So LL is itself the thing we ought to study, not merely a tool.

One can consider evolutionarily how logic may have came to be, why we beleive in implication and conjunction and negation etc. The thing that LL initially was emphasissed for was its ability to represent the finite vs infinite. To represent causal implication and logical implication. This may be good justification for beleiving LL is closer to the strictly correct notion of reasoning because there are many rules of thought that are finitistic (causal), while there are the "logical" (mathematical) laws that are infinite that are also captured by LL.

If logic is supposed to capture how we think, or how truth in an abstract sense "works" then we can ask the meta-quesiton. How do we compare different logics. What makes such and such logic better or more true than another. This is obviously a subtle and meaningless question. Regardless it seems important. One answer is utility. A logic is correct iff I get the things I want to be true being true from it. There are empiricist criterions, a rule is logically valid in the limit of some inductive argument. One suggested by Dan that perhaps one should take seriously the idea of symmetry as a logical asset, the more symetry a system seems to have the better. What one means by symmmetry (delimiting) and why an inconsistent system is not the ultimate in symetry is not clear but its a concept that can be applied just as validly as empiricism or the more honest blind preference.

Well anyway one might claim that LL has more symetry than classical and for this reason is to be prefered because it also has the same utility etc.

It occurs to me that maybe I could think of conjuction, implication and negation as trancendental logical rules in the Kantian sense (necessary for the possibility of [LOGICAL] thought).

Intuitionistic Turn

UNDER CONSTRUCTION

What is meant by intuitionistic or constructive logics. Brouwer and Heyting were the spearheads.

The synthesis of the movement seems to be the proofs as programs paradigm. If there is a Curry-Howard type isomorphism, a deterministic cut elimination process or you restrict to single formulas on the right of the turnstile.

Why is this related to double negation contradiction etc. There is also the point that Choice is non-constructive but that is somewhat seperate.

Why is the last thing constructive, precissely because it corresponds to the other two.

REFERENCE AND PROOF FOR THIS. THE UNDERSTANDING IS ALWAYS IN THE PROOF

Well the point is that Girard "discovered" LL in a semantics for intuitionistic logic and it was somewhat of a selling point in the original work that his logic was "constructive with an involutive negation". I kind of want to understand the idea of constructive more deeply here.

Quick Summary of Linear Logic

[1], [4], [5] as well as the original paper and the appendix of "Proofs and Types" by Lafont.

Linear logic appears to capture the logical behaviour of things like causality, question and answer, action and reaction, once, infinitely many times, both and either. These interpretations all focus on the role of the "bang" and "why not" modalities, which formally replace the structural rules.

Classical logic has the property that all proofs of a sequent must be equivilent under cut elimination (See Lafonts introduction for an explination why); this means that there can be no non-trivial denotational semantics and no Curry-Howard like correspondence to computation (all programs that compute the same thing are identified). Linear Logic refines the structural rules of classical logic to remidie this, i.e. it gives a classical setting in which the Curry-Howard type correspondence will hold. This is the sense in which it is constructive. It is nicer that intuitionistic logic because it maintains the symmetries of classical logic (involutive negation) while claiming the correspondence to programs still. The deduction rules, using a one sided sequent caqlculus presentation for its compacness, are below:

Identity group:

$$\frac{}{\vdash A, A^{\perp}}\ \text{axiom} \qquad \frac{\vdash \Gamma, A \qquad \vdash A^{\perp}, \Delta}{\vdash \Gamma, \Delta}\ \text{cut}$$

Structural Rule:

$$\frac{\vdash \Gamma}{\vdash \Gamma'}\ \text{exchange (Γ' is a permutation of Γ)}$$

Logical Rules:

$$\frac{}{\vdash 1}\ \text{one} \qquad \frac{\vdash \Gamma}{\vdash \Gamma, \perp}\ \text{false}$$

$$\frac{\vdash \Gamma, A \qquad \vdash B, \Delta}{\vdash \Gamma, A \otimes B, \Delta}\ \text{times} \qquad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \parr B}\ \text{par}$$

$$\frac{}{\vdash \Gamma, \top}\ \text{true} \qquad \text{(no rule for zero)}$$

$$\frac{\vdash \Gamma, A \qquad \vdash \Gamma, B}{\vdash \Gamma, A \& B}\ \text{with} \qquad \frac{\vdash \Gamma, A}{\vdash \Gamma, A \oplus B}\ \text{left plus} \qquad \frac{\vdash \Gamma, B}{\vdash \Gamma, A \oplus B}\ \text{right plus}$$

$$\frac{\vdash ?\Gamma, A}{\vdash ?\Gamma, !A}\ \text{of course} \qquad \frac{\vdash \Gamma}{\vdash \Gamma, ?A}\ \text{weak}$$

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, ?A}\ \text{deriliction} \qquad \frac{\vdash \Gamma, ?A, ?A}{\vdash \Gamma, ?A}\ \text{contraction}$$

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, \forall x.A}\ \text{for all (x not free in Γ)} \qquad \frac{\vdash \Gamma, A[t/x]}{\vdash \Gamma, \exists x.A}\ \text{exists}$$

The language is given over some set of atomic propositions and their negations $\{p, p^{\perp}, q, q^{\perp}, ...\}$. There are four constants $1, \perp, \top, 0$ for the connectives $\otimes, \parr, \&, \oplus$ respectively as well as the modalities $!, ?$ and the quantifiers $\forall, \exists$. Apart from the atomi negations the negation operation is meta-notation (defined) for the following:

$$1^\perp \equiv \perp \qquad \perp^\perp \equiv 1$$

$$\top^\perp \equiv 0 \qquad 0^\perp \equiv \top$$

$$(p)^\perp \equiv p^\perp \qquad (p\perp)^\perp \equiv p$$

$$(A \otimes B)^b ot \equiv A^\perp ⅋ B^\perp \qquad (A ⅋ B)^\perp \equiv A^b ot \otimes B^\perp$$

$$(A\&B)^\perp \equiv A^\perp \oplus B^\perp \qquad (A \oplus B)^\perp \equiv A^\perp \& B^\perp$$

$$(!A)^\perp \equiv ?A^\perp \qquad (?A)^\perp \equiv !A\perp$$

$$(\forall x.A)^\perp \equiv \exists x.A^\perp \qquad (\exists x.A)^\perp \equiv \forall x.A^\perp$$

Likewise implication is meta-notation for the following

$$A \multimap B \equiv A^\perp ⅋ B$$

Its interesting to notes that the deriliction rule is equivilent to adding either of the axioms $B \multimap ?B$ or $!B \multimap B$.

Inclusions of Other Logics

There is an embedding intuitionistic $LJ \hookrightarrow LL$

Theorem. *A sequent in the propositional (no quantifiers) fragment of Gentzens LJ is provable iff its image is provable in LL under the translation*

$$A \wedge B \mapsto A\&B$$

$$A \vee B \mapsto !A \oplus !B$$

$$A \implies B \mapsto !A \multimap B$$

$$\neg A \mapsto !A \multimap 0$$

Proof.

Proof? Extend to full generality of quantifiers, I imagine its identity map.

Translation and similar theorom for classical (LK)

Cut Elimination

[6], [2].

Want to connect this up to the nontrivial denotational models as well as how it replaces classical logic. Connect to the Curry-Howard correspondence for LL.

Proof Nets Formalism

UNDER CONSTRUCTION

This is the notation for equivilence classes of proofs. It can be slick or quite complicated. Its nice for the multiplicative fragment, however things get out of hand with the exponentials and "boxes". I dont yet understand this.

You can build up the proof nets inductively or you can define them as those proof structures that satisfy the long trip condition. This is effectively saying something about the fact that you can embed a circle into the graph in a way that goes through each node exactly twice..

Theorem. *A proof structure is a proof net iff it has a long trip.*

Differential Linear Logic

The Differential Lambda Calculus

2.1.1 Linearity

In λ-calculus linearity means that an argument is used exactly once. This is naturally connected to the concept of head reduction which is a reduction strategy evaluating the subterms in linear position.

There is a lot more on head reduction I think to look into

In algebra linearity means commutation with sums and scalars. The goal of DiLC is to connect these senses of linearity.

2.1.2 Differentials

In differential geometry we think of the differential of a map $f : M \to N$ between smooth manifolds as a smooth map linear map between vector spaces $Df_p : T_pM \to T_{f(p)}N$ or in more generality

$$Df : M \to Hom(T_pM, T_{f(p)}N)$$

with $Df(p)(v)$ sometimes denoted as $Df_p \cdot v$.

In ordinary vector calculus the directional derivative of a map $f : \mathbb{R}^n \to R$ along a vector v is given by

$$D_v f(x) = \lim_{h\to 0} \frac{f(x-hv) - f(x)}{h}$$

Alernative notation is that you dot product the derivative vector with the vetor along which you are taking the derivative.

write this out, I think this will make their notation more transparent?

2.1.3 The Calculus

The terms of the calculus over some commutative unital semi-ring $R = \{a, b, ...\}$ are given by a countable set of term variables $\{x, y, ...\}$ and

$$s, t ::== x|\lambda x.s|(s)t|D_i s.(t_1, ..., t_n)|0|as + bt$$

The following identities are then made on terms:

- α-equivilence
- $(as + bt)u = a(s)u + b(t)u$
- $\lambda x.(as + bt) = a\lambda x.s + b\lambda x.t$
- $D_{i_1,...,i_n} s = D_{i_{\sigma(1)},...,i_{\sigma(n)}} s$ where σ is a permutation
- $D_i(D_{i_1,...,i_n} x \cdot (u_1, ..., u_n)) \cdot u = D_{i,i_1,...,i_n} x \cdot (u, u_1, ..., u_n)$
- $$D_i(D_1^n \lambda xt \cdot (u_1, ..., u_n)) \cdot u = \begin{cases} D_1^{n+1} \lambda xt \cdot (u, u_1, ..., u_n), i = 1 \\ D_1^n \lambda x(D_{i-1} t \cdot u) \cdot (u_1, ..., u_n), i > 1 \end{cases}$$
- $D_i(t)v \cdot (u_1, ..., u_n) = (D_{i+1} t \cdot (u_1, ..., u_n))v$
- $D_i(\sum a_s s) \cdot (\sum b_u u) = \sum a_s b_u D_i s \cdot u$

A technical remark from [7] is that a single differential Ds suffices where originally ER had defined one piecewise $D_i s$. ER set this up a little differently, likely for the purposes of certain proofs, as well as making the definition as just the free R module generated on some set of terms. We want the differential to mimick the behaviour of the normal derivative, intuitivley reading $D_i s \cdot t$ as the derivative of s along t with respect to the i^{th} variable.

Lemma. $D_i(D_j s \cdot u) \cdot v = D_j(D_i s \cdot v) \cdot u$

Analogous to the interchange of partial derivatives on smooth functions. If we let $u, v \in \mathbb{R}^n$ and denote $v_i = (0, ..., 0, v_i, 0, ..., 0)$ we get that

$$D_{v_i} D_{u_j} f(x) = D_{u_j} D_{v_i} f(x)$$

Substitution is then defined inductively in the natural way

- $D_{i_1,...,i_n} y \cdot (u_1, ..., u_n)[t/x] = D_{i_1,...,i_n} y[t/x] \cdot (u_1[t/x], ..., u_n[t/x])$
- $D_1^n \lambda y v \cdot (u_1, ..., u_n)[t/x] = D_1^n \lambda y(v[t/x]) \cdot (u_1[t/x], ..., u_n[t/x])$
- $(v)w[t/x] = (v[t/x])w[t/x]$
- $(\sum a_v v)[t/x] = \sum a_v v[t/x]$

with some restrictions on free variables etc.

Linear substitution is then defined as follows: We denote the linear substitution as the partial derivative of s with respect to x along u as $\frac{\partial s}{\partial x} \cdot u$ and is given inductively

why is this not just the directional derivative?

- $$\frac{\partial D_{i_1,...,i_n} y \cdot (u_1, ..., u_n)}{\partial x} \cdot v = \delta_{x=y} D_{i_1,...,i_n} v \cdot (u_1, ..., u_n) + \sum_{i=1}^{n} D_{i_1,...,i_n} y \cdot (u_1, ..., \frac{\partial}{\partial x} u_i, ..., u_n)$$
- $$\frac{\partial D_1^n \lambda y v \cdot (u_1, ..., u_n)}{\partial x} \cdot t = D_1^n \lambda y(\frac{\partial v}{\partial x} \cdot t) \cdot (u_1, ..., u_n) + \sum_{i=1}^{n} D_1^n \lambda y v \cdot (u_1, ..., \frac{\partial}{\partial x} u_i \cdot t, ..., u_n)$$
- $$\frac{\partial (v)w}{\partial x} \cdot u = (\frac{\partial v}{\partial x} \cdot u)w + (D_1 v \cdot (\frac{\partial w}{\partial x} \cdot u))w$$
- $$\frac{\partial}{\partial x}(\sum a_v v) \cdot u = \sum a_v \frac{\partial v}{\partial x} \cdot u$$

There is a passing resemblance to the chain rule here in the application case.

The operation wants to be linear, so in the case of the application the v is in a linear position so we can take its partial derivative without problem. The w is not in linear position however so applying the operation to it must take two steps, replacing the $(v)w$ application with $(D_1 v \cdot w)w$ to get a "linear copy of w" then applying the partial derivative.

why is this a linear copy?

Lemma.

$$\frac{\partial D_i t \cdot u}{\partial x} \cdot v = D_i(\frac{\partial t}{\partial x} \cdot v) \cdot u + D_i t \cdot (\frac{\partial u}{\partial x} \cdot v)$$

Lemma. *If x is not free in t then*

$$\frac{\partial t}{\partial x} \cdot u = 0$$

Obvious parallel to the partial derivative of a constant (with respect to some variable) being zero

Lemma. *If y is not free in u*

$$\frac{\partial}{\partial x}(\frac{\partial t}{\partial y} \cdot v) \cdot u = \frac{\partial}{\partial y}(\frac{\partial t}{\partial x} \cdot u) \cdot v + \frac{\partial t}{\partial y} \cdot (\frac{\partial v}{\partial x} \cdot u)$$

Combining the last two we get

Lemma. *When y is not free in u and x is not free in v*

$$\frac{\partial}{\partial x}(\frac{\partial t}{\partial y} \cdot v) \cdot u = \frac{\partial}{\partial y}(\frac{\partial t}{\partial x} \cdot u) \cdot v$$

Interchange of second order partial derivatives.

The substitutions work well together

Lemma. *If x and y are distinct and y is not free in u or v then*

$$\frac{\partial t[v/y]}{\partial x} \cdot u = (\frac{\partial t}{\partial x} \cdot u)[v/y] + (\frac{\partial t}{\partial y} \cdot (\frac{\partial v}{\partial x} \cdot u))[v/y]$$

Lemma. *If x is not free in v and y is distinct from x we have*

$$(\frac{\partial t}{\partial x} \cdot u)[v/y] = \frac{\partial t[v/y]}{\partial x} \cdot (u[v/y])$$

Finally we can decompose the derivatives into finite sums of simpler terms (towards a Taylor expansion):

Lemma.

Lemma.

2.1.4 Confluence

The last thing to add to the calculus is a reduction rule. We extend beta reduction to include the following reductions:

- $(\lambda x.s)t \rightsquigarrow s[t/x]$
- $D_1 \lambda x.s \cdot u \rightsquigarrow \lambda(\frac{\partial s}{\partial x} \cdot u)$

It takes quite some work to then show that this is well defined etc.

Theorem. *This relation is confluent.*

Theorem. *If two ordinary lambda terms are beta equivalent in the differential lambda calculus then they are are beta equivalent in ordinary lambda calculus.*

2.1.5 Strong Normalization

We introduce types to the system with a collection of atomic types and then given two types A, B then $A \to B$ is a type. The normal lambda typing rules from lambda calculus carry over with some new typing rules:

$$\frac{\Gamma \vdash s : A_1, ..., A_i \to B \qquad \Gamma \vdash u : A_i}{\Gamma \vdash D_i s \cdot u : A_1, ..., A_i \to B} \text{ (Differential Application)}$$

$$\frac{\Gamma \vdash s : A \qquad \Gamma \vdash t : A}{\Gamma \vdash as + bt : A} \text{ (Linear Combination)}$$

$$\frac{}{\Gamma \vdash 0 : A} \text{ (Linear Combination)}$$

Then if the semi-ring overwhich we have defined the calculus has the following properties

- $ab = 0 \implies a = 0 \vee b = 0$
- $a + b = 0 \implies a = b = 0$ (positivity)
- $\forall a \in R$ there are only finitely many $b, c \in R$ with $a = b + c$

then we can prove that this is a strongly normalising calculus. With only positivity then we still have weak normalisation.

2.1.6 Taylor Expansion

Theorem (Leibniz Rule). *For terms t and u, and distinct variables x and y, such that y is not free in u we have that*

$$\frac{\partial t[x/y]}{\partial x} \cdot u = (\frac{\partial t}{\partial x} \cdot u)[x/y] + (\frac{\partial t}{\partial y} \cdot u)[x/y]$$

contrast

$$\frac{\partial (uv)}{\partial x} = u\frac{\partial v}{\partial x} + v\frac{\partial u}{\partial x}$$

im not really seeing it ...

"clear logical meaning, expressing how derivation behaves when interacting with a contraction in cut elimination"?

Theorem (Leibniz Formula). *Under the same restrictions*

$$\frac{\partial^n t[x/y]}{\partial x^n} \cdot u^n = \sum_{p=0}^{n} \binom{n}{p} \left(\frac{\partial^n t}{\partial x^p \partial y^{n-p}} \cdot u^n\right)[x/y]$$

Lemma (Deriving Applications). *Let $t = t_1, ..., t_k$ be a sequence of terms, x a variable and u a simple term. y a distinct variable to x not free in t or u. x is not free in u. Then if $n \geq 1$ we have*

$$\frac{\partial^n (x)t}{\partial x^n} \cdot u^n = n\frac{\partial^{n-1} (u)t}{\partial x^{n-1}} \cdot u^{n-1} + (\frac{\partial^n (y)t}{\partial x^n} \cdot u^n)[x/y]$$

Which has the special case

$$(\frac{\partial^n (x)t}{\partial x^n} \cdot u^n)[0/x] = n(\frac{\partial^{n-1} (u)t}{\partial x^{n-1}} \cdot u^{n-1})[0/x]$$

Theorem (Taylors Theorem). *If s and u are terms of the ordinary lambda calculus and ξ is a distinguished variable such that $(s)u \simeq_\beta \xi$ then there is a unique $n \in \mathbb{Z}$ with*

$$(D_1^n s \cdot u^n)0 \not\simeq_\beta 0$$

Moreover for this n we have that

$$(D_1^n s \cdot u^n)0 \simeq_\beta n!\xi$$

Hence the Taylors formula holds

$$s(u) = \sum_{n \geq 0} \frac{1}{n!}(D_1^n s \cdot u^n)0$$

s(u) instead of (s)u now wtf, its not even in the syntax

Proof. Sketch.

Argue from the fact that (s)u reduces to ξ that it must be of a certain form, assuming that it is in head normal form without loss of generality (all terms are beta equivilent to a unique head normal form term).

Notice that form has a recursive structure.

Define a number that is related to the number of beta reductions to induct on. It counts the number of substitutions of successive head variables of s in the linear head reduction of (s)u.

Using the previous Leibniz formula and deriving application lemma to compute $(D_1^n s \cdot u^n)0 \simeq_\beta n(D_1^{n-1} s' \cdot u^{n-1})0$ where s' is part of the recursive structure.

The result then follows.

The example of the self application of $\lambda x.xx$ is calculuated to have taylor expansion 0.

That makes no sense? So this term is identically 0? They use equality in the thm between the application and the Taylor series.. What do they mean?

Differential LL

2.2.1 Motivation

The notion of linearity can be seen in linear logic as well. A proof can be called linear in a hypothesis if that hypothesis is used exactly once during cut elimination i.e. not duplicated or eliminated.

In everyday mathematics derivations take a nonlinear map (between manifolds) and give a linear map (between vector spaces), the new structural rule of coderiliction mimic this behaviour of taking a nonlinear proof and making it linear (in the above sense). This is dual to deriliction which take a linear proof and makes it nonlinear. Erhardt stresses that this coderiliction gives a $!A$ without making it duplicable, this makes this fragment have some nice properties (strong normalisation, all proofs are linear combinations of simple proofs) .

but its not linear like MLL is so what are they talking about.

> “ *one really needs to take the point of view of computational trinitarianism in order to understand the transition from linear logic to differential linear logic. It is more difficult to understand naively the proofs of differential linear logic than the ones of linear logic.* ”
>
> **- nlab**

The idea of computational trinitarianism is that computation, logic and category theory are three sides of the same coin. They are all talking about the same object in different ways. Taking this really seriously means that if you observe something in a computation it must have a logical meaning, and if you see some categorical structure it must have a computational and logical meaning.

2.2.2 Syntax

In outline it is full LL with some new deduction rules

$$\frac{\Gamma, !A \vdash B}{\Gamma, A \vdash B}\ \text{(Coderiliction)}$$

$$\frac{\Gamma, !A \vdash B}{\Gamma, !A, !A \vdash B}\ \text{(Cocontraction)}$$

$$\frac{\Gamma, !A \vdash B}{\Gamma \vdash B}\ \text{(Coweakening)}$$

$$\frac{\vdash \Gamma \quad \dots \quad \vdash \Gamma}{\vdash \Gamma}\ \text{(sum)}$$

What is given in [3] is a bigger term calculus and full cut elimination transformations. The term calculus has more information in typing and contexts which make its semantics in categories maybe a little clearer, maybe not. The sum rule is written

$$\frac{\forall i \in [n] \quad \Phi \vdash p_i : \Gamma \text{ and } \mu_i \in R}{\Phi \vdash \sum \mu_i p_i : \Gamma}\ \text{(sum)}$$

so the linear combination information is in the context not the type.

This is interesting to me because the type to me is the logical part and the term is the program. No but I think the context is the actual proof, this is a term calculus for the proof, so the left of the : can be seen as the lambda term or the actual proof itself and the right is the type or the thing that is proved. This explains the use of the structural rules and dereliction rules in the context.

2.2.3 Basic Results

Yes the system is confluent and has normalization.

2.2.4 What does this have to do with differentiation

`https://www.pls-lab.org/en/Differential_Linear_Logic` has a nice little paragraph. Basically in some categorical models we have a derivation map that takes a function $f : !X \to Y$ (nonlinear because of the bang)

$$D(f) : !X \to (X \multimap Y)$$

i.e. a map that takes a point in the domain of f and gives a linear (no bang) map at that point, intuitively a linear approximation to the function there.

Stratified Linear Logic

Complexity Theory

3.1.1 Recall Acceptance

We say that a TM, M, accepts w iff there is a sequence of configurations starting at w and ending at the accept state. The language of a TM, $L(M)$, is the set of all accepted strings. A language is Turing recognisable iff there is a TM that enumerates it.

3.1.2 Complexity Classes

Notes on the "Advanced Theoretical Computer Science" subject slides.

> ***Definition: Big O Notation*** *We say that $g \in O(f)$ or sometimes $g = O(f)$ iff there is some $n_0, c \in \mathbb{N}$ such that for every $n > n_0$ we have that*
> $$g(n) \leq cf(n)$$
> *Asymptotically g is bounded by f.*

A function $t : \mathbb{N} \to \mathbb{N}$ is polynomial iff $\exists r \in \mathbb{N}$ such that $t \in O(n^r)$. The function hierarchy looks like the following

$$1 < log(n) < n < n^c < exp((log(n))^c(log(log(n)))^{1-c}) < c^n < n! < c^{b^n} < \text{ Ackerman functions etc}$$

Formally an algorthm is a Turing Machine. Running the algorithm on some data is having that data encoded on the tape of the TM when it is run. The runtime of the algorithm is then the number of steps that it takes for the TM to halt (enter its accept state). The idea in complexity is to bound the runtime as a function of the length of the input.

> ***Definition: Time Complexity of TM*** *The time complexity of given TM, M, is*
> $$t_M(n) = max_{w \in \Sigma^n}\{m : M(w) \text{ halts after } m \text{ steps }\}$$
> *So the longest time for the machine to halt on any given length n input.*

The complexity class of a function is then the collection of all languages that are decided in less than the time of the function i.e.

$$TIME(t) = \{L : L \text{ is a language that is decided by some TM M with } t_M \in O(t)\}$$

Theorem. *Every linear time language is regular.*

Different types of TM can have slightly different complexity properties (there are algorithms which can be given faster implimentations on two tape vs one tape machines). They are however closed under certain classes.

> ***Definition: Polynomial Time*** *P is the class of languages decidable by a deterministic TM in polynomial time*
> $$P = \bigcup_k TIME(n^k)$$

Any (reasonable) deterministic model of a TM will have the same class P.
A verifier for a language A is a deterministic TM (algorithm) M such that

$$A = \{w : \exists c, \text{ M accepts } \langle w, c\rangle\}$$

Then a language is polynomial time verifiable if this algorithm is polynomial time in the length of w.

Definition: NP

$$NP = \{\textit{ languages with a polynomial verifier}\}$$
$$= \bigcup_k NTIME(n^k)$$

so the collection of all languages decidable by a non-deterministic TM in polynomial time.

The runtime of a non-deterministic TM (NTM) is the maximum over all branches of the computation. Again the class NP is robust against changes to model of non-deterministic computation.

3.1.3 Hardness

Definition: *A language $\mathcal{A}$ reduces in polynomial time to a language B, denoted $\mathcal{A} \leq_p B$ iff there is an algorithm f such that*

$$w \in \mathcal{A} \iff f(w) \in B$$

B is NP hard iff for any $A \in NP$ we have that $A \leq_p B$. B is NP complete (NPC) if in addition $B \in NP$. There are several interesting theorems about this

- $A \leq_p B, B \in P \implies A \in P$
- $B \in NPC, B \in P \implies NP = P$
- $B \in NP - hard, B \leq_p C \implies C \in NP - hard$
- $P \neq NP \implies NPI = NP \setminus (P \cup NPC) \neq \emptyset$ i.e. if P is not NP then there are non-polynomial non-NP complete problems.

P is closed under compliment however it is not known whether the same is true of NP. The hypothesised class is called co-NP i.e.

$$co - NP = \{\mathscr{L} : \mathscr{L}^C \in NP\}$$

Stratified LL: $LL_\S$

3.2.1 Motivation

There have been two successful strategies for creating subsystems of LL with bounded complexity properties. The idiosyncratic soft logic of Lafont [?] and a collection of systems from Girard, Terui, Baillot, Mazza and others that utilise a system of stratification or levels. Similar to type theory the idea is to label sequents with a "level" and then disallow the communication of levels (through the deduction rules only allowing cuts and other operations to happen on the same level). After sever years of refinement [?] introduced a general framework in which to understand these logics using a notion of stratification. The logic is called Stratified Linear Logic and abbreviated to $LL_\S$

The general insight is that when stratification is tied to the exponentials complexity properties emerge because those are the operations in LL that are responsible for the computational power.

3.2.2 The Calculus

The calculus has all the formulas of LL with one new modality $A ::== A|\S A$, that is self dual (negation defined as) $(\S A)^\perp \equiv \S A^\perp$. The sequent calculus is now a 2-sequent system, which just means that each formula has a label of level attached.

Identity group:

$$\frac{}{\vdash A^i, A^{\perp^i}} \text{ axiom} \qquad\qquad \frac{\vdash \Gamma, A^i \qquad \vdash A^{\perp^i}, \Delta}{\vdash \Gamma, \Delta} \text{ cut}$$

Structural Rule:

$$\frac{\vdash \Gamma}{\vdash \Gamma'} \text{ exchange } (\Gamma' \text{ is a permutation of } \Gamma)$$

Logical Rules:

$$\frac{}{\vdash 1^i}\ \text{one}$$

$$\frac{\vdash \Gamma, A^i \qquad \vdash B^i, \Delta}{\vdash \Gamma, (A \otimes B)^i, \Delta}\ \text{times}$$

$$\frac{}{\vdash \Gamma, \top^i}\ \text{true}$$

$$\frac{\vdash \Gamma, A^i \qquad \vdash \Gamma, B^i}{\vdash \Gamma, (A \& B)^i}\ \text{with}$$

$$\frac{\vdash ?\Gamma, A^i}{\vdash ?\Gamma, !A^i}\ \text{promotion}$$

$$\frac{\vdash \Gamma, A^i}{\vdash \Gamma, ?A^i}\ \text{deriliction}$$

$$\frac{\vdash \Gamma, A^i}{\vdash \Gamma, \forall x.A^i}\ \text{for all (x not free in } \Gamma)$$

$$\frac{\vdash \Gamma}{\vdash \Gamma, \bot^i}\ \text{false}$$

$$\frac{\vdash \Gamma, A^i, B^i}{\vdash \Gamma, (A \parr B)^i}\ \text{par}$$

(no rule for zero)

$$\frac{\vdash \Gamma, A^i}{\vdash \Gamma, (A \oplus B)^i}\ \text{left plus}$$

$$\frac{\vdash \Gamma, B^i}{\vdash \Gamma, (A \oplus B)^i}\ \text{right plus}$$

$$\frac{\vdash \Gamma}{\vdash \Gamma, ?A^i}\ \text{weak}$$

$$\frac{\vdash \Gamma, ?A^i, ?A^i}{\vdash \Gamma, ?A^i}\ \text{contraction}$$

$$\frac{\vdash \Gamma, A^i[t/x]}{\vdash \Gamma, \exists x.A^i}\ \text{exists}$$

$$\frac{\Gamma, A^{i+1}}{\Gamma, \S A^i}\ \text{paragraph}$$

Some things to notice: The cut, as well as other logical rules must take place between two formulas of the same level. The paragraph modality can decrease the depth, and is the only rule that changes the index of a formula.

There is a new cut elimination transformation to deal with the new modality

$$\frac{\dfrac{\Gamma, A^{i+1}}{\Gamma, \S A^i}\ \text{paragraph} \qquad \dfrac{\Delta, A^{\perp^{i+1}}}{\Delta, \S A^{\perp^i}}\ \text{paragraph}}{\Gamma, \Delta}\ \text{cut} \quad \rightsquigarrow \quad \frac{\Gamma, A^{i+1} \qquad \Delta, A^{\perp^{i+1}}}{\Gamma, \Delta}\ \text{cut}$$

One last modification is to restrict the collection of valid proofs to only those that have all formulas of the same level in the conclusion, this ensures that $\vdash A, B$ is provable iff $\vdash A \parr B$ is provable.

3.2.3 Basic Results

The system enjoys cut elimination.

There is a $(\mathbb{Z}, +)$ action on formulas $k \bullet A^i \mapsto A^{i+k}$ and proofs $k \bullet \pi \mapsto$ the same proof with all formulas acted on by k.

Lemma. $\vdash \Gamma$ *is derivable iff* $\vdash k \bullet \Gamma$ *is derivable*

Hence it is not the level that matters but the relative level within a proof.

There is a proof net formalism with all the usual properties and there is a categorical construction that can be given to any model of LL to make it a model of $LL_\S$

Inclusion of Other Logics

There are several other LL variants that can be easily shown to sit inside $LL_\S$. A list of the ones mentioned in [**?**] are LLL, ELL, L^3, L^4 and obviously full second order LL.

3.3.1 LL

Simply remove the paragraph deduction rule and you have second order LL.

3.3.2 LLL

Light Linear Logic, Girards logic of polytime, is given when we restric to the fragment where stratification corresponds to exponential depth (where depth of exponentials is explained in [?]).

Make this explicit and precise. Connect to Asperti and Teruis LALL and Mazzas L^4 which are all polytime too

3.3.3 ELL

Girards system for Elementary (towers of exponentials) time computations. This is given by restricting to sequents of the form $\vdash \S\Delta^i, \Gamma^{i+1}$, where there are no paragraphs in Γ.

3.3.4 L^3

The subsystem of Baillot and Mazzas Linear Logic by Levels that captures elementary functions. We simply require that every exponential is preceeded by a §. The paper then goes on to give several different classifications of L^3 [?].

Can we extend them to L4?

Interactive

Geometric

Semantic

Denotational Semantics

Algebraic Geometry I Guess

I never thought it would come to this.

4.1.1 Functors

Algebraic geometry terms are all functors on categories in their absolute generality. Moreover these encapsulate all the information that the concrete maps do.

4.1.2 Algebras and Coalgebras

give the definitions

Some interesting facts to prove to get my head around the concept. Maybe try to understand some stuff in this post `https://mathoverflow.net/questions/76509/what-is-a-coalgebra-intuitively` Go through an example of a bialgebra and check all the diagrams...

- coporduct is a homomorphism iff product is a cohomomorphism
- Sym as the limit of finite subspaces or whatever

Todo list

References

[1] Torben Braüner. Introduction to Linear Logic.

[2] Torben Braüner and Valeria De Paiva. Cut-Elimination for Full Intuitionistic Linear Logic. *BRICS Report Series*, 3(10), January 1996.

[3] Thomas Ehrhard. An introduction to Differential Linear Logic: proof-nets, models and antiderivatives. 2016.

[4] J.-Y. Girard. Linear Logic: its syntax and semantics. In Jean-Yves Girard, Yves Lafont, and Laurent Regnier, editors, *Advances in Linear Logic*, pages 1–42. Cambridge University Press, 1 edition, June 1995.

[5] Jean-Yves Girard. LINEAR LOGIC AND PARALLELISM. Technical report.

[6] Michele Pagani and Lorenzo Tortora de Falco. Strong normalization property for second order linear logic. *Theoretical Computer Science*, 411(2):410–444, January 2010.

[7] Lionel Vaux. The differential $\lambda\mu$-calculus. *Theoretical Computer Science*, 379(1-2):166–209, June 2007.