Algebraic Logic
for Deep Learning

by

YKY

A Thesis Submitted to
The Hong Kong University of Science and Technology
in Partial Fulfilment of the Requirements for
the Degree of Master of Philosophy
in Applied Mathematics

August 2024, Hong Kong

Authorization

I hereby declare that I am the sole author of the thesis.

I authorize the Hong Kong University of Science and Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the Hong Kong University of Science and Technology to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

YKY

13 August 2024

Algebraic Logic
for Deep Learning

by

YKY

This is to certify that I have examined the above MPhil thesis
and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by
the thesis examination committee have been made.

Prof. Who, Thesis Supervisor

Prof. Who Else, Thesis Co-supervisor

Prof. Ikari Someone, Head of Department

Department of Maths

13 August 2024

ACKNOWLEDGEMENTS

Thank you, all the Evangelion.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

Algebraic Logic
for Deep Learning

by YKY

Department of Maths

The Hong Kong University of Science and Technology

Abstract

Some text.

CHAPTER 1

INTRODUCTION

1.1 Background

Lambek posited this "trinity":

$$
\begin{tikzcd}
& \text{type theory} \arrow[dl, dash, "\textit{Curry–Howard}"'] \arrow[dr, dash, "\textit{obvious}"] & \\
\text{logic} \arrow[rr, harpoon, shift left, "\text{classifying topos}"] & & \text{category theory} \arrow[ll, harpoon, shift left, "\text{internal language}"]
\end{tikzcd}
\tag{1.1}
$$

where the double arrows at the base can be understood thusly:

I extended some nodes to better see their relations:

$$
\begin{tikzcd}
& & \text{type theory} \arrow[dl, dash] \arrow[dr, dash] & & & \\
\text{algebraic logic} \arrow[r, dash] \arrow[rrrrr, dashed, dash, bend right, "?"] & \text{logic} \arrow[rr, dash] & & \text{category theory} \arrow[r, dash] & \text{topos} \arrow[r, dash] & \text{algebraic geometry}
\end{tikzcd}
\tag{1.2}
$$

I am curious if the two algebras on the left and right are identical?

1.2 Paul Halmos' algebraic logic

Every Boolean algebra $\mathbb{A}$ is isomorphic to the set of all continuous functions from X into $\mathbb{O}$, where X is the dual space of the algebra $\mathbb{A}$, and $\mathbb{O}$ is the Boolean algebra with 2 elements. If there is a homomorphism f between Boolean algebras $\mathbb{A} \to \mathbb{B}$ then there is a dual morphism f^*

between their dual spaces $Y \to X$:

$$\begin{tikzcd}
\mathbb{A} \arrow[r, "f"] \arrow[d, phantom, "\wr\|"] & \mathbb{B} \arrow[d, phantom, "\wr\|"] \\
\left\lceil X \right. \arrow[d] & \left. Y \right\rceil \arrow[l, "f^*"'] \arrow[d] \\
\left\lfloor \mathbb{O} \right\rfloor & \left\lfloor \mathbb{O} \right\rfloor
\end{tikzcd} \tag{1.3}$$

1.3 Yuri Manin and Russians

1.4 Topos and internal language

1.5 Term rewriting and all that

人工智能逻辑 比它更 general

Equational 逻辑，有等号 “=” predicate
专门决定 A =? B 问题 (Ch 4)
这问题又叫 “word problem”

Gröbner bases (用 Buchberger 算法求出) (Ch 8) (1.4)

可以快速解决 ideal membership 问题：
given $f, f_1, \ldots, f_k \in K[X_1, \ldots, X_n]$
ask $f \in \langle f_1, \ldots, f_k \rangle$?

ideal congruence 问题 也可以视为 word problem

- What is the **word problem**?

  is defined for an equational theory E.

  is the problem of deciding whether $s = t$

- why is Gröbner basis equivalent to the word problem

  to ask ideal congruence $f = ?g$ means $f - g \in ?J$

  which is ideal membership problem

  a polynomial can be regarded as a rewrite rule

  because $f = 0$, we can take the “largest monomial” in f as the LHS, and the rest of f as RHS.

  In other words: ideal = set of rules

We ask if a polynomial can be rewritten by the set of rules to another form.

This is similar to logic deduction.

- Here an important question is: polynomial reduction seems unable to handle **logic variables**, it seems only capable of **simple symbolic rewriting**.
- logic is equivalent to what form of polynomials?

  taking the cue that Huet's higher-order unification = Buchberger algorithm, ...

1.6 The set-up

The set of equations F defines an algebraic set = **the world**:

$$F(x) = 0. \tag{1.5}$$

The objective of an intelligent agent is to learn F.

We have the function f performing **prediction** of the immediate future:

$$\boxed{\text{current state}} \quad x_t \overset{f}{\mapsto} x_{t+1} \quad \boxed{\text{next state}}\ . \tag{1.6}$$

In an infinitesimal sense, we can see f as a **differential equation** describing the **world trajectory**:

$$\dot{x} = f(x). \tag{1.7}$$

So F is the **solution** to this differential equation.

It seems that F and f are more or less equivalent ways to describe the world.

Logic can be turned into some form of algebra, and this algebra can be used to express either F or f. Perhaps both ways are feasible, or even mixing the two.

What does it mean to use logic to express F or f?

The following table depicts the main correspondences relevant to our research:

LOGIC	**facts** human(socrates)	**rules** $\forall x.\text{human}(x) \rightarrow \text{mortal}(x)$
ALGEBRA	**element** $p \in \mathbb{A}$	**element** $(p \rightarrow q) \in \mathbb{A}$
WORLD	**states** x_t	**state transitions** $x_t \overset{f}{\mapsto} x_{t+1}$

(1.8)

The relation between LOGIC and WORLD has been elucidated quite thoroughly in the AI literature. Note that the state x_t is made up of a set of facts (logic propositions). A single step of logic inference results in a new conclusion δx which is *added* (as a set element) to the current state x_t to form a new state x_{t+1}. Here t refers to "mental time" which does not necessarily coincide with real time.

1.7 From abstract algebraic logic to concrete computations

There are two main routes to make abstract algebraic logic concrete:

- Find **matrix representations** of the logical algebra
- Implement the logical algebra as the commutative algebra of (classical) **polynomials**

1.8 What does it mean to train the AI?

From the previous section,

$$F(x) = 0 \quad \text{is the solution to} \quad \dot{x} = f(x) \tag{1.9}$$

and the two descriptions (by F or by f) are equivalent.

The sensory data from the AI are a set of "world" points $\{x_i\}$ and we require either:

$$F(x_t) = 0 \quad \text{or} \quad f(x_t) = \delta x = x_{t+1} - x_t \tag{1.10}$$

and F or f can be trained by gradient descent to eliminate errors in the above conditions (equations).

- the x_t's are represented as **logic facts**
- F or f is represented as **logic rules**

and we need to **evaluate** $F(x_t)$ or $f(x_t)$.

Let's do some examples:

Logic formula	Algebraic form
human(socrates)	h(s) = 1
human(socrates) ∧ human(plato)	h(s) · h(p) = 1
human(socrates) → mortal(socrates)	1 + h(s) + h(s) · m(s) = 1
$\forall x$. human(x)	$h(x)$ is a propositional function $\forall_x h(x)$ is a constant function mapping to 1 or 0
$\forall x$. human(x) → mortal(x)	$\forall_x \big(1 + h(x) + h(x) \cdot m(x)\big)$ is a constant function mapping to 1 or 0
$\forall x, y, z$. father(x, y) ∧ father(y, z) → grandfather(x, z)	$\forall_x \forall_y \forall_z \big(1 + f(x,y) \cdot f(y,z) + f(x,y) \cdot f(y,z) \cdot g(x,z)\big)$ $\mapsto$ 0 or 1
general Horn formula: $\forall_{x...} P \wedge Q \wedge R... \rightarrow Z$	$\forall_{x...} \big(1 + P \cdot Q \cdot R.... + P \cdot Q \cdot R... \cdot Z\big)$ $\mapsto$ 0 or 1

(1.11)

Now imagine there are millions of such rules. Number of predicates obviously increases.

Does each equation require new variables, or can variables be re-used? Seems yes, can be re-used.

The **loss function** would be the sum of squared errors over all equations:

$$\mathcal{L} = \sum_{\text{eqns}} \epsilon^2 = \sum_i \big(\phi_i(x...) - 1\big)^2. \tag{1.12}$$

Learning means to perform the **gradient descent** via $\nabla_\Phi \mathcal{L} = \frac{\partial \mathcal{L}}{\partial \Phi}$ where Φ is the set of parameters for the equations.

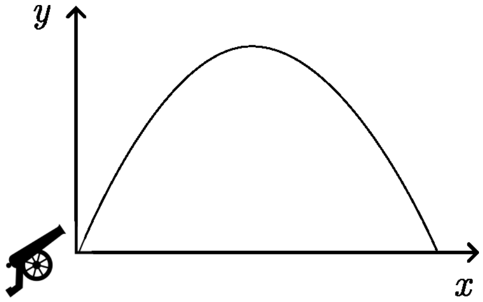
Potential problems:

- How to represent the set of equations efficiently? Matrix of coefficients seems wasteful.
- We have lost the "**deepness**" of deep learning, but there is recent research showing that **shallow learning** may work well too.
- Need to iterate logical inference multiple times using the same set of equations.

How are new conclusions added to the state? What is the state? State = set of facts = set of **grounded** equations.

Inference: how to get from current state to next state? Big problem!! New ground facts have to be read off from satisfaction of all equations. Rather intractable...

Go back to a physics example, the parabolic trajectory of a canon ball:



(1.13)

The parabola is given by the quadratic equation from high school:

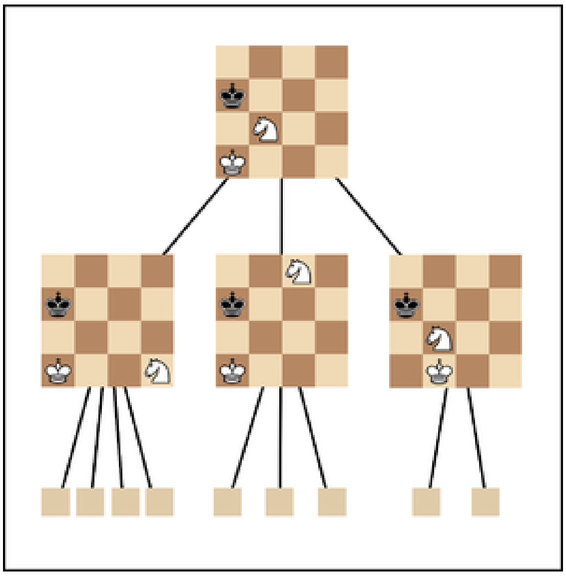
$$F(x) = 0 \qquad F(x) = ax^2 + bx + c \tag{1.14}$$

but the trajectory can also be described by the physics equation parameterized by time t:

$$\dot{\mathbf{x}} = f(\mathbf{x}) = (v_x, v_y) = (v_x^0, -gt + v_y^0). \tag{1.15}$$

This parametric form is not unique. For example, another way is for the point $\mathbf{x}$ to move with uniform speed along the trajectory.

Note that the trajectory above is qualitatively the same as a "thought trajectory" in cognitive space. One intuitive way to visualize cognitive trajectories is via the example of a chess game tree:



(1.16)

$f(x)$ is the functional form we want, as it contains information about the "interestingness" of deduced conclusions.

Are our equations in Table 1.11 describing F or f?

An intuitive idea: state = facts = grounded equations, rules = quantified equations. So the equations are modeling f. This exposed a problem of classical AI that I have not paid too much attention to: the selection of interesting conclusions. It's hard to **enumerate conclusions**, let alone to rate their interestingness.

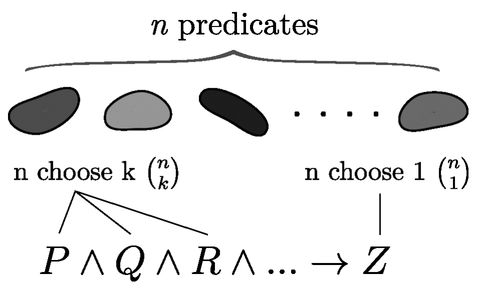
How to enumerate conclusions? The loss function (1.12) can be trained on any data with correlations. But what we have is data in the form of time series. Need extra measures to ensure that equations only model $x_t \to x_{t+\Delta t}$. Now how to enumerate conclusions? From current state, iterate over all equations to generate new states. This is getting very close to the classical logic-based AI inference algorithm.

The **interestingness function** gives a probability distribution over conclusions given the current "context" (which we identify as the current state x_t): $\text{Intg}(\delta x) = \mathbb{P}(\delta x | x_t)$. This function has to be learned. It has an equivariant structure due to the state as a set of propositions with permutation invariance.

One more efficiency problem: iterating through all equations is inefficient, which brings back the necessity of the classical **rete algorithm**: instead of matching rules against the state, we should match the current state against rules. In other words, instead of $\delta x := \bigcup_i \text{rule}_i(x)$, perform $\delta x :=$ compiled-rules(Δx), where Δx is the change of the current state from the previous state, and δx is the change from the current state to the next state.

But we may also avoid the rete algorithm by stacking logical equations into **layers**, thus getting an efficiency advantage similar to deep learning. A "single" step of logic inference would mean going through multiple layers of logic rules (equations).

1.9 "Geometric" logic inference algorithm



(1.17)

- What is a logic fact within a state?
- How does a rule generate a single new fact?

A fact consists of a point (in subject space) and a predicate that contains it. The point itself does not suffice because it can belong to various predicates.

To apply a rule, each **atomic term** in the rule has to be satisfied. For each predicate $Q()$, this is verified by testing if we have any points among the facts contained in Q.

If we have father(john, pete) as a fact then we certainly can satisfy father(x, y). But we already have the point (john, pete) which may satisfy other predicates $Q(x, y)$. So our method is slightly more permissive (and thus more powerful) in rules matching.

How does the rule's RHS generate a new fact? It should also be a (point, predicate) pair. The point has to **match** the premise. How could this be ensured?

Secondly, the output predicate may not cover the point.

The matching process: syntactically, we look at each literal in the rule and see if any fact unifies with the literal. Geometrically, it means taking a point and checking if it lies inside a predicate. If the fact is a (point, predicate) pair then it is given that the point belongs to the predicate, so it is not necessary to check for membership. The result is simply taking the point when the predicates match. But we still need to keep track of which variables the point coordinates are binding to.

The matching of the second literal will also return a point, but its coordinates would be bound to different dimensions.

The binding of coordinates would be the basis of verifying the rule LHS.

RHS: The bound variables (in various dimensions) need to be projected to the output space. It may or may not be covered by the output predicate. If not, the rule does *not* apply. This is in accord with the principle that rules should not change during inference.

CHAPTER 2

CONCLUSIONS

Some conclusion text.

CHAPTER 3

CONCLUSIONS

Some conclusion text.

CHAPTER 4

CONCLUSIONS

Some conclusion text.

CHAPTER 5

CONCLUSIONS

Some conclusion text.

CHAPTER 6

TIC TAC TOE EXPERIMENT

6.1 Representation of states and rules

At any time the state is a set of facts, ie, pairs of (point $\in$ predicate).

There will be N_P predicates and N_R rules.

Each rule is a conjunction of all predicates. If N_P is large, the rules would be cumbersome.

Having a large number of rules, N_R, seems not to have a deleterious effect, if the rules recommender is good at its job.

Each literal in the rule may be negated, how to handle this?

Each literal contains a predicate and its arguments, which can be constants or variables.

It seems that genetic algorithms would be best suited for this kind of search for rules... or unless the rules are represented in such a way that they can continuously vary in a differentiable manifold.

6.2 Rules recommender

The rules recommender would be a set function:

$$\mathsf{Ru} : \{\text{current state}\} \to \{\text{set of rules}\} \tag{6.1}$$

which has equivariant structure on both its input and output. This suggests the **Transformer** architecture is suitable for learning this function.

6.3 Interestingness

This can be implemented implicitly by making the inference algorithm output a probability distribution over all deduced conclusions and then picking the most probable one.

CHAPTER 7

CONCLUSIONS

Some conclusion text.

APPENDIX A

LIST OF PUBLICATIONS

APPENDIX B

FYTGS REQUIREMENTS

The requirements are from the RPG Handbook.

B.1 Components

B.1.1 Order

A thesis should contain the following parts in the order shown:

1. Title page, containing in this order:
    a. Thesis title
    b. Full name of the candidate
    c. Degree for which the thesis is submitted
    d. Name of the University, *i.e.* The Hong Kong University of Science and Technology
    e. Month and year of submission
2. Authorization page
3. Signature page
4. Acknowledgments
5. Table of contents
6. Lists of figures and tables
7. Abstract ($\leq$ 300 words.)
8. Thesis body
9. Bibliography
10. Appendices and other addenda, if any.

B.1.2 Authorization page

On this page, students authorize the University to lend or reproduce the thesis.

1. The copyright of the thesis as a literary work vests in its author (the student).

2. The authorization gives HKUST Library a non-exclusive right to make it available for scholarly research.

B.1.3 Signature page

This page provides signatures of the thesis supervisor(s) and Department Head confirming that the thesis is satisfactory.

B.1.4 Acknowledgments

The student is required to declare, in this section, the extent to which assistance has been given by his/her faculty and staff, fellow students, external bodies or others in the collection of materials and data, the design and construction of apparatus, the performance of experiments, the analysis of data, and the preparation of the thesis (including editorial help). In addition, it is appropriate to recognize the supervision and advice given by the thesis supervisor(s) and members of TSC.

B.1.5 Abstract

Every copy of the thesis must have an English abstract, being a concise summary of the thesis, in 300 words or less.

B.1.6 Bibliography

The list of sources and references used should be presented in a standard format appropriate to the discipline; formatting should be consistent throughout.

Sample pages of both MPhil and PhD theses are provided here (MPhil / PhD), with specific instructions for formatting page content (centering, spacing, etc.).

B.2 Language, Style and Format

B.2.1 Language

Theses should be written in English.

Students in the School of Humanities and Social Science who are pursuing research work in the areas of Chinese Studies, and who can demonstrate a need to use Chinese to write their theses should seek prior approval from the School via their thesis supervisor and the divisional head.

If approval is granted, students are also required to produce a translation of the title page, authorization page, signature page, table of contents and the abstract in English.

B.2.2 Pagination

1. All pages, starting with the Title page should be numbered.
2. All page numbers should be centered, at the bottom of each page.
3. Page numbers of materials preceding the body of the text should be in small Roman numerals.
4. Page numbers of the text, beginning with the first page of the first chapter and continuing through the bibliography, including any pages with tables, maps, figures, photographs, etc., and any subsequent appendices, should be in Arabic numerals.
5. Start a new page after each chapter or section but not after a sub-section.

Note: That means the Title page will be page i; the first page of the first chapter will be page 1.

B.2.3 Format

1. A conventional font, size 12-point, 10 to 12 characters per inch must be used.
2. One-and-a-half line spacing should be used throughout the thesis, except for abstracts, indented quotations or footnotes where single line spacing may be used.
3. All margins—top, bottom, sides—should be consistently 25mm (or no more than 30mm) in width. The same margin should be used throughout a thesis. Exceptionally, margins of a different size may be used when the nature of the thesis requires it.

B.2.4 Footnotes

1. Footnotes may be placed at the bottom of the page, at the end of each chapter or after the end of the thesis body.
2. Like references, footnotes should be presented in a standard format appropriate to the discipline.
3. Both the position and format of footnotes should be consistent throughout the thesis.

B.2.5 Appendices

The format of each appended item should be consistent with the nature of that item, whether text, diagram, figure, etc., and should follow the guidelines for that item as listed here.

B.2.6 Figures, Tables and Illustrations

Figures, tables, graphs, etc., should be positioned according to the scientific publication conventions of the discipline, e.g., interspersed in text or collected at the end of chapters. Charts, graphs, maps, and tables that are larger than a standard page should be provided as appendices.

B.2.7 Photographs/Images

1. High contrast photos should be used because they reproduce well. Photographs with a glossy finish and those with dark backgrounds should be avoided.
2. Images should be dense enough to provide 300 ppi for printing and 72 dpi for viewing.

B.2.8 Additional Materials

Raw files, datasets, media files, and high resolution photographs/images of any format can be included.

Note: Students should get approval from their department head before deviating from any of the above requirements concerning paper size, font, margins, etc.