Fundamental Theorems in Propositional Calculus: Precomputed Knowledge?

Francisco Hernández Quiroz
(some ideas by Nancy Abigail Nuñez)
Facultad de Ciencias, UNAM

20th May 2025

Abstract

- Some very useful and basic classical propositional theorems appear to be very intuitive and easy to grasp.
- But some (like De Morgan's laws) are not easy to prove, but nonetheless they play a role in other more advanced theorems.
- Perhaps they contain precomputed, harder to get, knowledge? In order to test this theory, a suitable measure of hardness is needed.
- Minimal proof length in decidable systems is a good indication of the minimal information needed to prove a theorem.
- This measure is independent of the chosen proof system, provided it meets some natural requirements.
- Moreover, as classical propositional logic is decidable, the measure turns out to be computable (although at a high cost).

Codified Knowledge in Nature

- Genomics has made clear that the basic instructions for building live organism are codified by DNA and possibly by other biochemical means [12, 1].
- But many aspects of living beings' behaviour is also codified by the same means.
- Behaviour can be regarded in a very simplified way as an organism's ability to react to its environment.
- This ability can be very much enhanced if some successful behaviour patterns are included in the codification.
- Therefore codified information is crucial to a living organism survival chances.

Artificial Life: Codified Knowledge in . . . Code

- The field of **Artificial Life** as represented in [8] attempts to reproduce computationally what living organism do.
- An “organism” is a program whose structure or observable behaviour mimics “real” living beings.
- Information enabling the artificial organism to succeed is now codified as . . . computer code.
- Theoretical computational tools could help us to understand how information is stored and processed and what limits can be found in artificial life [6].

What to Compute Beforehand

- Information is not just data to be processed, but very often is also the result of this processing.
- In this case, it can be said that some information has been precomputed.
- But what information is worth precomputing? [13]
- For a living or artificial organism, information that can enhance its survival chances.
- Information retrieval and processing can be very costly and impossible to do just when required, so precomputing it is crucial.
- But an organism may not be able to appreciate the effort that went into this precomputation as the organism may just take it for granted.

Critique of Pure Reason Revisited

- Remember Kant celebrated two dimensional distinction regarding knowledge/judgements [7]:
  - A priori vs a posteriori knowledge. Does knowledge depends on experience or not? A priori knowledge is necessary and universal.
  - Analytic vs synthetic judgements. Is the content of the predicate already contained in the subject? Analytic judgements are true by definition.
- Kant contended that a priori synthetic judgements are possible and are at the core of mathematical knowledge, among other theoretical endeavours.

The scandal of deduction

- If logical deduction allows only to infer conclusions that already are contained in the premises, how can new knowledge be added by this process? [3].
- Although by no means the same problem, there is a parallelism between the idea of acquiring new knowledge by deduction, knowledge that does not depend on experience, and the possibility of having a priori synthetic judgements.
- In the case of deduction a key ingredient is **computation**.

What to pre Compute in Logic

- Computational complexity theory has taught us that deduction is by no means a trivial task of unwrapping conclusions from premises.
- Complexity theory quantifies the cost of deduction in terms of the temporal and spatial resources needed to make an inference.
- If deduction is part of the knowledge package used by successful organisms a promising strategy would be to precompute deductions in some way that reduces the need to incur its costs instantly.
- So let's have a look to what computational complexity can tell us about deduction.

A Measure of Logical Complexity

- The complexity of proofs has been at the centre of attention in computational complexity theory since its beginings.
- Cook and Reckhow proved SAT's **NP**-completeness [4].
- Pudlák and Krajíček have put forward many fundamental results in proof complexity theory [9, 10, 11].
- D'Agostino and Floridi advanced a hierarchy of increasingly complex tautologies [5].
- All these results could be the basis for classifying the difficulty of proving logical theorems, couldn't they?

Not enough fine-grained and too many fundamental open problems

- Computational complexity classifies problems in very broad classes.
- For instance SAT $\in$ **NP** and TAUT $\in$ **coNP**. But obviously not all instances in SAT or in TAUT are equally difficult.
- Besides, fundamental problems remain open (and may remain so): **P** $=$ **NP**?, **NP** $\neq$ **coNP**?, and therefore we are not even sure if these classes are really different.
- As a consequence, many results are of a conditional nature: "if a proof system is polynomially bound then **NP** $=$ **coNP**".
- So it is difficult to use them as a basis for deciding what logical knowledge is useful if precomputed.

Logic and Kolmogorov complexity

- According to Chaitin, formal mathematical theories can be seen as a combination of a **program** that runs in a **computer** and during its execution produces a series of **theorems** [2, p. 10].
- This approach incorporates Algorithmic Information Theory to our understanding of mathematical theories.
- But it also puts all theorems on an equal foot in terms of complexity (an automatic theorem producer may work for all of them).
- Not all theorems are equally informative and definitely some are harder to prove.
- We will recover some ideas of Chaitin's but at the same time we'll produce a theory that can discriminate between theorems through a metric analogous to Kolmogorov complexity.

Kolmogorov Complexity in a Nutshell

Let U be a *Universal* Turing Machine and let α be finite string. Then

$$K_U(\alpha) = |p| \qquad \text{where } p \text{ is the shortest program such that } U(p) = \alpha.$$

It is also posible to measure the complexity of a string given another string

$$K_U(\alpha \mid \beta) = |p| \qquad \text{where } p \text{ is the shortest program such that } U(p, \beta) = \alpha.$$

Let U and U' be two universal Turing machines. Then

$$K_U(\alpha) \leq K_{U'}(\alpha) + c,$$

that is, the Kolmogorov complexity of a string is the same irrespective of the chosen universal computer, up to a constant value c.

Interpreters

- Machines U and U' are *interpreters* of two (Turing-complete) programming languages L and L'.
- Write an interpreter of L in L' (and vice versa). Let us call them I_L and $I_{L'}$ respectively.
- Given a program $p \in L$, build a program $p' \in L'$ doing the same task just by “composing” programs p and I_L and running them in U'.
- The cost in terms of program length is fixed if the composition can be implemented by a fixed program not dependent on p.
- Observe how program p is not compiled (translated) into a new program $p' \in L'$.
- The shortest program in L has to be shorter than the shortest program in L' plus the length of interpreter $I_L{}'$.
- Otherwise, take the shortest program in L' and the interpreter $I_{L'}$ and that’s an even shorter program in L.

Goals and methods of the metrics I

- Our main goal is to have a metric for tautologies that behaves very much as Kolmogorov complexity.
- It will provide a quantifiable means of comparing the information content of tautologies by assigning low values to low complexity and high values to high complexity.
- In an analogous way to Kolmogorov complexity, the minimal proof length should be the same irrespectively of the proof system employed (provided the proof system meets some sufficient conditions).
- It is not based on translations (simulations) between proofs but on “interpreters” of proof systems in other proof systems.

Technical preliminaries I

A very natural definition for our purposes is

$$K_P(\alpha) = \min\{|\pi| \mid P(\pi, \alpha)\}$$

where P is a complete and sound proof system for classical propositional calculus and $\alpha \in \mathrm{TAUT}$.
In order to fulfil our goals, the above metric should meet the following inequality

$$K_P(\alpha) \leq K_Q(\alpha) + C_{P,Q}, \tag{1}$$

for any suitable proof systems P and Q. $C_{P,Q}$ is a value that depends only on P and Q but crucially not on α.

Technical preliminaries II

Definition (Steps in a proof)

$$
\begin{aligned}
\mathbf{k}(\pi) &= \text{number of steps in the proof } \pi \\
\mathbf{k}_P(\alpha) &= \min\{\mathbf{k}(\pi) \mid P(\pi, \alpha)\}.
\end{aligned}
$$

Technical preliminaries III

Definition (Frege rules and systems)

A *Frege rule* is an inference rule of the following form

$$\frac{\alpha_1, \ldots, \alpha_n}{\beta}$$

such that

$$\alpha_1, \ldots, \alpha_n \models \beta.$$

The formulas $\alpha_1, \ldots, \alpha_n$ are the hypotheses/premises and the formula β is the consequence. If $n = 0$ then the rule is an axiom scheme.

A substitution is a function σ that replaces the propositional atoms in a formula with arbitrary propositions.

A **Frege system** F is a finite set of Frege rules in a complete language (that is, with enough logical connective to express any boolean function).

Technical preliminaries IV

An *F-proof* of the formula θ from the formulas $\eta_1, \ldots, \eta_m$ is a sequence of formulas $\pi = \langle \gamma_1, \ldots, \gamma_k \rangle$ such that:

- $\gamma_k = \theta$;
- for all i such that $1 \leq i \leq k$
  - either $\gamma_i \in \{\eta_1, \ldots, \eta_m\}$;
  - or there is a Frege rule

$$\frac{\alpha_1, \ldots, \alpha_n}{\beta}$$

in F and there are numbers $i_1, \ldots, i_n$ and a substitution σ such that

$$\sigma(\alpha_1) = \gamma_{i_1}, \ldots, \sigma(\alpha_n) = \gamma i_n \qquad \sigma(\beta) = \gamma_i.$$

Technical preliminaries V

In that case we will write

$$\pi : \eta_1, \ldots, \eta_m \vdash_F \theta.$$

On the other hand we will write simply

$$\eta_1, \ldots, \eta_m \vdash_F \theta$$

when there is an F-proof π of θ from $\eta_1, \ldots, \eta_m$.

More technical preliminaries I

Definition (Soundess and completeness)

F is *sound and implicationally complete* if and only if

$$\eta_1, \ldots, \eta_m \models \theta \qquad \text{if and only if} \qquad \eta_1, \ldots, \eta_m \vdash_F \theta$$

More technical preliminaries II

Definition (Extended Frege system)

An *extended Frege system* is a Frege system F where the i-th step in the proof of θ from $\eta_1, \ldots, \eta_m$ can also be an *extension axiom*

$$q \equiv \varphi$$

where the atomic formula q abbreviates φ provided

- φ in the language of the system F;
- q does not occur in any $\eta_1, \ldots, \eta_m$;
- q does not occur in $\gamma_1, \ldots, \gamma_{i-1}$;
- q does not occur in φ.

The inclusion of extension atoms can shorten proofs significantly (exactly how much is an open question).

More technical preliminaries III

Definition (Substitution rule)

Let σ be a substitution of atomic formulas for arbitrary formulas. The *substitution rule* is

$$\frac{\alpha}{\sigma(\alpha)}.$$

A **substitution Frege system** is a Frege system supplemented by the substitution rule.

Warning. Applying the substitution rule to hypotheses or to formulas depending on hypotheses renders the system unsound. A trivial example is the substitution $p := \neg p$ applied to the hypothesis p which can produce a “proof” of $p \vdash \neg p$.

Frege Systems and Inferential Complexity

Some sound and complete proof systems can meet the requirements of equation 1. As it turns out, three features suffice for this, and they are quite not extraordinary in any way, as they capture some widespread practices in logic textbooks:

(a) to introduce instances of theorems already proved;

(b) to introduce new connectives by definitions;

(c) to use theorems with hypotheses already proved as if they were new inference rules.

Central theorem of inferential information complexity

Theorem (Universality of inferential complexity)

Let P and Q be two proof systems with features (a)–(c). Then

$$K_P(\alpha) \leq K_Q(\alpha) + C_{P,Q}.$$

Where $C_{P,Q}$ is the length of an interpreter of system P in system Q.

Main (and yet Untested) Hypothesis

Propositional theorems that play a key role in proving other theorems and whose inferential information complexity is very high are worth precomputing.

Future Work

- Gathering empirical evidence in favour of the main hypothesis.
- A direct definitive proof may not be feasible as finding the minimal proof of a theorem is **NP**-hard.
- Instead, an approximation technique where successively shorter proofs are found and used as proxies of the minimal length proof.
- Simultaneously, find statistical evidence that crucial theorems are necessary (or at least useful) in proving other theorems (some evidence is already available for De Morgan's laws).

References I

[1] A. Bird,, “Perceptions of epigenetics”,, in: *Nature* 447 (2007), pp. 396–398, doi: doi.org/10.1038/nature05913.

[2] Gregory J. Chaitin,, “From Philosophy to Program Size. Key Ideas and Methods”,, Lecture Notes on Algorithmic Information Theory. Estonian Winter School in Computer Science, 2003, doi: doi:10.48550/ARXIV.MATH/0303352.

[3] M. Cohen and E. Nagel,, *An introduction to logic and scientific method*,, Routledge and Kegan Paul, 1934.

[4] Stephen A. Cook and Robert A. Reckhow,, “The Relative Efficiency of Propositional Proof Systems”,, in: *The Journal of Symbolic Logic* 44.1 (1979), pp. 36–50, url: http://www.jstor.org/stable/2273702.

References II

[5] M. D'Agostino and L. Floridi,, "The enduring scandal of deduction: Is propositional logic really uninformative?", in: *Synthese* 167 (2009), pp. 271–315.

[6] Santiago Hernández-Orozco, Francisco Hernández-Quiroz and Héctor Zenil,, "Undecidability and Irreducibility Conditions for Open-Ended Evolution and Emergence",, in: *Artificial life* 24.1 (2018), pp. 56–70, doi: 10.1162/ARTL_a_00254.

[7] Immanuel Kant,, *The Critique of Pure Reason*,, Translated by J.M.D. Meiklejohn, 1781.

[8] Maciej Komosinski and Andrew Adamatzky, eds.,, *Artificial Life Models in Software*,, Springer, 2009.

References III

[9] Jan Krajíček, ed.,, *Complexity of Computations and Proofs*,, vol. 13, Quaderni di Matematica, Dipartimento di Matematica della Seconda Università di Napoli, 2004.

[10] Jan Krajíček,, "Information in propositional proofs and algorithmic proof search",, in: *The Journal of Symbolic Logic* 87.2 (2022), pp. 852–869, doi: 10.1017/jsl.2021.75.

[11] Pavel Pudlák,, "The Lengths of Proofs",, in: *Handbook of Proof Theory*, ed. by Samuel R. Buss, vol. 137, Studies in Logic and the Foundations of Mathematics, Elsevier, 1998.

[12] James Watson and Francis Crick,, "Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid",, in: *Nature* 171.4356 (1953), pp. 737–738.

References IV

[13] Hector Zenil,, “What to Compute Before an Apocalypse: Knowledge as Cryptocurrency at the End of Civilisation”,, in: *Post-Apocalyptic Computing*, chap. Chapter 2, pp. 75–92, doi: 10.1142/9789811297144_0002.