Proofs, Not Narratives: A Proof-Theoretic Approach to Explainable AI in Education

Simone Cuconato[1]

[1]Department of Physics, University of Calabria, 87036 Rende (CS), Italy

Abstract

In education, explainable AI is more than a technical desideratum: a system that performs without exposing why teaches nothing a learner can check, and in the classical tradition to which the Cl@ssico Digitale project — an innovative curriculum developed by the University of Calabria at the Liceo Classico "G. da Fiore" in Rende (CS) — appeals, to know has always meant being able to give one's reasons. Yet the explanations offered by contemporary systems narrate a decision rather than certify it, and their faithfulness to the underlying computation cannot itself be checked. This paper develops, at the meeting point of mathematical logic and philosophy of AI, a proof-theoretic response to that difficulty. We present a propositional calculus of sequent-style tableaux and the paradigm of Logic-Constrained Prompt Learning, under which a commercial large language model is disciplined by subtraction: every degree of freedom the calculus does not license is removed, so that the model cannot assert a verdict except by exhibiting a derivation whose every step is checkable by hand against seven rules. Because the calculus is analytic, one class of error — the appearance of a formula outside the subformula space of the problem — acquires an exact syntactic signature, and the system becomes explainable by design rather than after the fact, so that verifying its output reduces to proof-checking, a task within reach of a high school student. We further show how the premisses actually consumed by the closures of a derivation can be read off the derivation itself, and delimit precisely what separates this reading from the technical notion of sufficient reason. The framework is illustrated on the Cartesian *cogito*, formalised and mechanically verified in class, and situated within the twofold aim of using AI for education and educating about AI.

Keywords

explainable AI in education, Logic-Constrained Prompt Learning, sequent-style tableaux, neuro-symbolic AI, proof theory

CEUR-WS.org/Vol-4305/9_paper.pdf

1. Introduction

Generative language models have entered the classroom well ahead of any settled understanding of what it means to learn with them, and mathematics education, where the literature is largest, records alongside genuine promise a persistent unease. A scoping survey finds ChatGPT useful for explaining elementary concepts and personalising practice, and isolates over-reliance and the erosion of critical thinking as its characteristic risks [1, 2]. The empirical studies sharpen the diagnosis: pre-service teachers who solved tasks with ChatGPT accepted incorrect answers readily and assessed their own competence in a way that did not correlate with their performance [3], and in a parallel study failed to recognise the model's mistakes at all, recording them as "different approaches" rather than as errors [4]. At the metacognitive level the difficulty is structural: used as a problem-solving tutor, GPT-4 moved directly from orienting to executing, skipping the planning stages through which a human tutor builds reflective control, so that students developed at best a diagnostic use of the tool and never an elaborative one [5]. What these findings share is not that the model reasons badly, but that its explanations *narrate* a solution without exhibiting a structure one could check, leaving the learner to trust or to guess. What matters pedagogically is therefore less access to the model than the form of the interaction with it, and in particular the verifying attitude that the tools themselves neither require nor reward [6].

It is against this background that the Cl@ssico Digitale project takes its stance. Developed by the University of Calabria at the Liceo Classico "Gioacchino da Fiore" of Rende (CS) and now in its fourth

AI-LEAP 2026 - 1st International Workshop on AI Learning, Education And Practice, co-located with AIxIA 2026, University of Perugia, October 2026

simone.cuconato@unical.it (S. Cuconato)

0000-0003-0277-9575 (S. Cuconato)

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-10-28

year [7, 8], it asks what role the classical *lyceum* can play in the age of artificial intelligence, and answers with a notion it places at its centre: *epistemic resistance*, the capacity to preserve the primacy of rational argumentation against the seduction of fluent output. This is no defensive reflex against technology but a positive commitment, drawn from the classical model of knowledge the *lyceum* inherits from the Greek tradition, in which to know is not merely to register that something is the case but to be able to give its reasons — the Greek "invention of the why" [9]. A generative language model, taken as it comes, is an almost perfect antagonist: it returns answers that are plausible, well formed and rhetorically confident, while offering no guarantee that any inferential path connects its premisses to its conclusions. In the vocabulary the students acquire from reading Plato's *Sophist*, it is an *automated sophist*, a producer of discourses that imitate knowledge without possessing its structure. The didactic question is therefore not whether to admit such systems into the classroom, which they have long since entered, but whether their generative freedom can be disciplined so that what they return is an argument rather than the appearance of one.

Logic is the natural place to look for that discipline, and not by accident [10]. The line running from Greek rationalism through the formalisation of deductive reasoning to the theory of computation and thence to artificial intelligence makes logic the one subject at the crossroads of the classical and the computational [7]; and it is there that the ancient demand that knowledge render its *why* meets the contemporary demand, made urgent by opaque models in high-stakes settings, that artificial intelligence be *explainable*. The two demands, we shall argue, are one demand in two idioms. The Cl@ssico Digitale curriculum gives this convergence an institutional home in *Logica e Filosofia dell'IA*, an innovative discipline introduced specifically into the curriculum[1], where formal logic and the philosophy of artificial intelligence are taught together. We take as our domain not mathematics but philosophy — the reading and reconstruction of classical arguments, a native competence of the *lyceum* — and as our instrument not ChatGPT but a constrained configuration of Google's Gemini.

The method was first developed in a clinical–pharmacological setting, where the knowledge base is distilled from official drug leaflets [11]. Three things are new here. The knowledge base is no longer fixed once and for all by an expert, but is the outcome of an interpretive act performed by the students on a source text, which moves the hermeneutic work from the periphery of the exercise to its centre. The construction is then placed within Hilbert's demand for finite, mechanically checkable proof objects, and thereby within a recognisable programme rather than an isolated device. It is connected, finally, to the recent proof-theoretic analysis of explanation from reasons, which allows us to say with precision how much of the word "explanation" the framework earns. This paper accordingly develops the logico-epistemological framework (Section 2), documents a worked classroom instance in which the inferential core of Descartes' *cogito* is formalised, derived and verified line by line (Section 3), and states the protocol under which the approach is to be evaluated (Section 3.4). Section 4 concludes.

2. The logico-epistemological framework: sequent-style tableaux and Logic-Constrained Prompt Learning

This section develops the framework by which the discipline invoked in the Introduction — the constraint that turns a fluent generator into a giver of reasons — is actually imposed. We present first, in a self-contained form, the propositional calculus of *sequent-style tableaux* on which everything rests; we then define the paradigm of *Logic-Constrained Prompt Learning* (LCPL) in the form appropriate to philosophical reasoning, and locate it among the current strategies for endowing language models with logical reasoning; we describe how a commercial model — Gemini — is concretely constrained, by what we call *neuro-symbolic integration by subtraction*, and delimit exactly what the constraint does and does not guarantee. A final subsection situates the framework within the proof-theoretic analysis of explanation from reasons [12, 13].

[1]The second discipline added to the study plan is *Scienza e Tecnologie Digitali*, conceived as a foundational discipline of computer science, in some respects akin to the one offered in the Italian *Liceo Scientifico* with the Applied Sciences option, but with a more theoretical and reflective orientation, consistent with the cultural identity of the Liceo Classico.

2.1. The propositional calculus of sequent-style tableaux

Sequent-style tableaux — block analytic tableaux in sequent notation — graft the sequent notation of Gentzen [14, 15] onto the refutational branching architecture of the analytic tableaux of Beth and Smullyan [16]: each node of the refutation tree carries not a single formula but a finite *block* of formulae, the one-sided counterpart of a sequent whose succedent has been absorbed, in negated form, into the antecedent. The calculus is developed in full in [17], its structural proof theory — the correspondence with cut-free LK, the associated G3- and G0-style sequent calculi and the normalization theorem — in [18], and its first-order extension in [19].[2] We recall here only what Sections 2.3 and 3 require.

Definition 2.1 (Language)**.** The propositional language $\mathscr{L}_0$ is built from a countable set of propositional variables $Var_0 := \{p_1, p_2, p_3, \ldots\}$, the connectives $\{\neg, \wedge, \vee, \rightarrow\}$ and the auxiliary symbols $\{(\,,\,)\}$; the set $Form_0$ of well-formed formulae is defined inductively in the usual way.

In the application that concerns us, each variable denotes a proposition with a determinate philosophical content — "you think", "you know that this is just" — fixed by an interpretation key agreed upon in the formalization phase.

Definition 2.2 (Sequent and associated block)**.** A *sequent* is an expression $\Gamma \Rightarrow \Delta$, where $\Gamma, \Delta \subseteq Form_0$ are finite, possibly empty sets. It is *valid* when every valuation satisfying all of Γ satisfies some member of Δ; the empty antecedent is satisfied by every valuation and the empty succedent by none, so that the definition covers the degenerate cases, which the reading through an associated formula would not, $\mathscr{L}_0$ containing neither $\top$ nor $\bot$. The *block* associated with $\Gamma \Rightarrow \Delta$ is the set

$$\Pi := \Gamma \cup \neg[\Delta], \qquad \neg[\Delta] := \{\neg\psi \mid \psi \in \Delta\},$$

in which antecedent and negated succedent are gathered into a single one-sided list. The sequent is *valid* if its associated formula $\bigwedge \Gamma \rightarrow \bigvee \Delta$ is a tautology.

Proposition 2.3 (Sequent–block correspondence)**.** $\Gamma \Rightarrow \Delta$ *is valid if and only if* $\Pi = \Gamma \cup \neg[\Delta]$ *is unsatisfiable.*

Proof. A valuation refutes the sequent precisely when it verifies every formula of Γ and falsifies every formula of Δ, that is, precisely when it satisfies Π; the claim follows by contraposition. □

It is on blocks that the calculus operates, by decomposition. A block is *closed* (×) if it contains a formula together with its negation; a block to which no rule is applicable and which is not closed is *completed open* (⊙); a tableau is closed if every branch is. The rule repertoire consists of seven schemata, in two families: the α-rules produce a single child block inheriting both formulae of the decomposition, the β-rules produce two child blocks, each inheriting one and each carrying the context unchanged.

– *α-type rules*

$$\frac{\Pi, \neg\neg\varphi}{\Pi, \varphi}\,(\neg\neg) \qquad \frac{\Pi, \varphi \wedge \psi}{\Pi, \varphi, \psi}\,(\wedge) \qquad \frac{\Pi, \neg(\varphi \vee \psi)}{\Pi, \neg\varphi, \neg\psi}\,(\neg\vee) \qquad \frac{\Pi, \neg(\varphi \rightarrow \psi)}{\Pi, \varphi, \neg\psi}\,(\neg\rightarrow)$$

– *β-type rules*

$$\frac{\Pi, \neg(\varphi \wedge \psi)}{\Pi, \neg\varphi \;\mid\; \Pi, \neg\psi}\,(\neg\wedge) \qquad \frac{\Pi, \varphi \vee \psi}{\Pi, \varphi \;\mid\; \Pi, \psi}\,(\vee) \qquad \frac{\Pi, \varphi \rightarrow \psi}{\Pi, \neg\varphi \;\mid\; \Pi, \psi}\,(\rightarrow)$$

In each rule the principal formula is *removed* and replaced by its immediate components, the rest of the block passing unchanged; the rules are in this sense destructive, and no formula is ever retained

[2] The block format is not tied to classical logic: replacing unsigned formulae by signed ones yields a block calculus for intuitionistic propositional logic, in which the whole of intuitionism is carried by the rule for the false-signed implication [20]. The present work stays classical throughout.

alongside its own decomposition. The entire repertoire fits in one display, which is a design decision rather than an accident: the calculus is meant to be internalized by students with no previous logical training, and every step of every derivation must be checkable by hand against these seven schemata. The measure under which the rules diminish what they act on is the following [18].

Definition 2.4 (Weight)**.** The *weight* of φ is the pair $w(\varphi) = (b(\varphi), c(\varphi))$, where $b(\varphi)$ counts the occurrences of the binary connectives $\wedge, \vee, \to$ in φ and $c(\varphi)$ the occurrences of all connectives; weights are compared lexicographically.

Theorem 2.5 (Soundness, completeness, termination)**.** *Every tableau is finite, and every branch terminates in a block that is closed or completed open. There is a closed tableau for* Π *if and only if* Π *is unsatisfiable; validity of sequents is therefore decidable.*

Proof. Termination. Order blocks by the finite multiset of the weights of their members, under the multiset extension of the lexicographic order on weights. Every rule removes its principal formula and puts in its place only immediate components of it, and an immediate component has strictly smaller weight: the six schemata that decompose a binary connective remove that occurrence and introduce none, so b strictly drops, while $(\neg\neg)$ leaves b unchanged and lowers c by two. Each application therefore strictly decreases the multiset, and the multiset extension of a well-founded order is well founded, so no branch is extendable indefinitely. The measure is uniform over the seven schemata; a plain count of connectives is not, since it leaves $(\neg\vee)$ stationary. Because the rules are destructive, no proviso against repeated applications is needed: a principal formula is consumed by its own rule.

Soundness. By induction on the height of the tableau: a closed block is unsatisfiable, since no valuation verifies a formula together with its negation; and if a block is satisfiable then so is at least one of its children, immediately from the truth conditions of the principal formula. Hence a block with a closed tableau is unsatisfiable.

Completeness. Contrapositively, suppose no tableau for Π closes. By termination some branch $\mathscr{R} = \Pi_0, \dots, \Pi_k$ ends in a completed open block. Put $H := \Pi_0 \cup \cdots \cup \Pi_k$, the union of the blocks along the branch; it is H, and not the terminal block Π_k, that carries the information, since destructiveness leaves in Π_k only formulae no rule applies to. Then H is a Hintikka set. No rule has a literal as its principal formula, so a literal persists from the block where it appears down to Π_k; an atom occurring in H together with its negation would therefore put both in Π_k and make it closed, which is the first Hintikka condition and is also what makes the per-block closure criterion adequate. If a formula of type α or β belongs to some Π_j, it cannot survive in the completed block Π_k, where its rule would still apply; by persistence of the context it was therefore principal at some stage between j and k, and that application placed into the branch both of its components in the α case, and the component lying on $\mathscr{R}$ in the β case. Assigning to each variable the value true exactly when it belongs to H satisfies H, and hence $\Pi_0 \subseteq H$. □

For the purposes of this paper, however, the load-bearing property of the calculus is neither soundness nor completeness but its *analyticity*, the property that a tradition running from Bolzano and Gentzen to the contemporary debate on what makes a proof system good has isolated as the mark of a well-designed calculus [21]. Gentzen expressed it by saying that a derivation in normal form is not roundabout: no concepts enter the proof other than those contained in its final result [14]. In the present one-sided, refutational setting the property takes an "up to negation" form.

Definition 2.6 (Negation-closed subformula space)**.** For a block Π_0, let $\mathrm{Sub}(\Pi_0)$ be the set of subformulae of the formulae of Π_0, and $\mathrm{Sub}^{\neg}(\Pi_0) := \mathrm{Sub}(\Pi_0) \cup \{\neg\psi \mid \psi \in \mathrm{Sub}(\Pi_0)\}$.

Proposition 2.7 (Analyticity)**.** *Every formula occurring in any block of any tableau with root* Π_0 *belongs to* $\mathrm{Sub}^{\neg}(\Pi_0)$*. Moreover, the calculus is cut-free by construction: no rule introduces a formula that is neither an immediate component of its principal formula, nor the negation of one, nor inherited unchanged from the parent block.*

Proof. By induction on the height of the tableau, inspecting the seven schemata: each rule removes its principal formula and adds immediate subformulae of it, or negations of immediate subformulae, while the context passes unchanged; and $\mathrm{Sub}^{\neg}(\Pi_0)$ is closed under both operations. No schema has the shape of cut, with a formula present in the premisses and absent from the conclusion. □

The weight of Proposition 2.7 is worth stating plainly. The complete lexicon out of which any legitimate derivation can be built is fixed, and finite, *before the derivation begins*: it is $\mathrm{Sub}^{\neg}(\Pi_0)$, computable by inspection of the initial block. A derivation is therefore *self-contained* in Poggiolesi's sense [21], and a reader examining it is never confronted with material whose provenance must be taken on trust. When the derivation is produced by a generative model the observation acquires a sharper edge: one characteristic failure, the intrusion of a concept belonging to no part of the problem, has an exact syntactic signature, the occurrence in some block of a formula outside $\mathrm{Sub}^{\neg}(\Pi_0)$. Analyticity converts that failure from a statistical risk into a locally detectable violation, recognizable by anyone — a student included — who compares each line against a set computed once from the initial block. It does not, as Section 2.3 insists, convert every failure into one.

2.2. Logic-Constrained Prompt Learning for philosophical reasoning

The calculus becomes an instrument of artificial intelligence education the moment it is imposed on a language model as the model's sole repertoire of admissible generative moves. We call this imposition Logic-Constrained Prompt Learning; the definition adapts to the philosophical domain the clinical formulation of [11].

Definition 2.8 (LCPL, philosophical instance)**.** Let A be an argument extracted from a philosophical source text, reconstructed as a set of premisses and a conclusion and formalized in $\mathscr{L}_0$ — following the method of [22] — as a pair (Γ_A, ψ_A) with $\Gamma_A \subseteq \mathit{Form}_0$ finite and $\psi_A \in \mathit{Form}_0$. A language model $\mathscr{M}$ operates under *Logic-Constrained Prompt Learning* when it is not permitted to assess A by free generation, but is constrained to exhibit a finite tree T of blocks such that: (i) the root of T is $\Pi_0 = \Gamma_A \cup \{\neg\psi_A\}$, the block associated with the sequent $\Gamma_A \Rightarrow \psi_A$; (ii) the children of each non-leaf node are the block, or the two blocks, obtained from it by the explicitly declared application of one of the seven rules of Section 2.1; (iii) every leaf of T is closed or completed open. The verdict on A is read off T: the argument is *valid* if every leaf is closed, *invalid* otherwise, with a countermodel extracted from any completed open leaf.

The division of labour between the two components is strict. The neural component performs the interpretive and heuristic work for which no formal specification exists: reading the argument, choosing at each step which formula to decompose next. The symbolic component fixes the space of admissible moves, and thereby the legitimacy of each step.

Recent work on endowing language models with logical reasoning divides into three families: solver-based methods, which translate the problem into a formal language and delegate the inference to an external engine; prompt-based methods, which shape the model's own generation; and fine-tuning methods, which alter the model itself [23]. LCPL belongs to the second family, and its position within it is what distinguishes it. The solver-based line — Logic-LM [24], and with different engines LINC [25] and SatLM [26] — obtains a guarantee that covers the inference but not its visibility: the solver's internal search is not the exhibited object, and a student cannot inspect it. Prompt-based methods keep the reasoning in view but, as a rule, use logic to *guide* a chain that remains free; here the calculus is not a guide but the whole of the admissible repertoire, and the object exhibited is the certificate itself. The closest competitors in ambition lie elsewhere again, in autoformalisation with proof assistants, where a model drafts a proof that Isabelle or Lean then checks [27]: that route yields machine-checkable output, as ours does, but its certificate is addressed to a proof assistant, whereas seven schemata on half a blackboard are addressed to a sixteen-year-old. For the classroom this difference is not incidental; it is the point.

One further difference from the clinical instance changes the didactic function of the knowledge base. There the base is distilled once and for all from a regulatory text by the human informer, and

the student receives it as given; here Γ_A is the outcome of an interpretive act performed on the source text, sensitive to the translation of natural-language connectives, to the granularity of the atoms, and to genuinely contestable exegetical choices. This hermeneutic layer is half of the exercise, and the half in which the competences of the Liceo Classico are directly engaged.

2.3. Constraining Gemini: neuro-symbolic integration by subtraction

The implementation is a personalized configuration (a *Gem*) of Google's Gemini model, whose generative policy is rigidified, at every conversational turn, by system instructions comprising exactly two components: the seven rule schemata of Section 2.1 and the required layout of the blocks. Both are supplied not as informal paraphrases but as LaTeX source — the very source that typesets them here — and the model must return its derivations in the same notation. The choice is less incidental than it appears: it anchors the model's generative space to the corpus of formal writing in which the calculus lives, serializes two-dimensional proof notation unambiguously, and leaves no translation step between what the machine manipulates, what a checker parses and what the student reads on the blackboard.

We describe the resulting architecture as a neuro-symbolic integration *by subtraction*. Nothing is added to the model: no external solver to which inference is delegated, no verifier agent appended downstream, no retrieval layer interposed. Rather, every degree of freedom that the calculus does not license is *subtracted* from the model's generative policy, so that the model cannot assert a verdict except by traversing, block by declared block, a derivation whose every step instantiates one of the seven schemata. What survives is exactly the heuristic residue the calculus leaves open — the order of decomposition, the reading of the source text — and nothing else.

The epistemological yield is a guarantee whose logical form must be stated exactly, because it is easy to overstate. By the soundness half of Theorem 2.5, *any* well-formed closed tableau — whoever or whatever produced it — establishes that the conclusion follows from the declared premisses; the trustworthiness of the generator is irrelevant to the trustworthiness of the generated proof. What holds is thus a conditional: *if* the output is a well-formed closed tableau, *then* the verdict is certified. Prompting specifies the admissible derivations; it does not by itself establish that the model complies with the specification, and the antecedent of the conditional has to be checked rather than assumed. The check is threefold, and each part is necessary. Each step must instantiate a schema on a formula actually present in the parent block — local rule checking; each leaf must be genuinely closed, or genuinely completed open — without which nothing establishes that the tree is a complete tableau; and no formula may fall outside $\mathrm{Sub}^{\neg}(\Pi_0)$ — the perimeter check. The perimeter check alone is necessary but not sufficient, since a formula inside the perimeter can still be introduced by an illegitimate step, and local rule checking alone does not exclude a tableau left unfinished. Two residual failure modes therefore survive the constraint, and both are false negatives rather than false positives: the model may fail to complete a tableau, or may declare a branch completed open prematurely, so that a valid argument goes unrecognised. What the architecture excludes is the opposite and more dangerous error, an unfounded verdict that passes as founded — and it excludes it not by making the model reliable, but by making non-compliance visible. What remains outside the guarantee altogether is the adequacy of the formalization (Γ_A, ψ_A) to the source text, which no calculus can certify and which stays with the human interpreter.

In the vocabulary of explainable artificial intelligence, the system is *explainable by design* rather than post hoc: since the model is bound to exhibit the derivation leading to its verdict, there is no explanation to be reconstructed after the fact — the explanation *is* the decision procedure, and the documented gap between what a black-box model computed and what its post-hoc rationale narrates [28] cannot open, there being no rationale distinct from the computation exhibited.

This commitment has a precise historical name, and it is worth pronouncing it, because it locates the proposal within a recognisable programme rather than an isolated device. Hilbert asked that mathematics be recast as a formal system in which proofs are finite objects, manipulable and mechanically verifiable, so that the correctness of a demonstration would rest on its form and not on the mathematician's insight into its content [29]. Contemporary explainable AI, for all its technical refinement,

occupies in this respect the position of pre-Hilbertian mathematics: it entrusts itself to intuitions and to narratives — the fluent chain-of-thought of a language model, the post-hoc feature attributions of methods such as LIME and SHAP [30, 31] — whose faithfulness to the underlying computation cannot itself be checked. That this is not a merely theoretical worry has been established experimentally: models can be steered by features they never mention, producing chains of thought that are systematically unfaithful to the process that generated the answer [32]. To bring explanation inside Hilbert's programme is to accept a single, exacting criterion: an explanation is an explanation only if it is a syntactic object that a mechanical verifier can check in finite time. A derivation produced under LCPL is built to satisfy that criterion; a post-hoc rationale, by its nature, cannot.

2.4. Proofs that certify and proofs that explain

We have spoken of a machine constrained to give reasons, and it is fair to ask how much of that phrase the construction redeems. A closed tableau delivers a guarantee of correctness; the classical demand asks for more. Since the *Posterior Analytics* [33], the demonstrative tradition distinguishes knowledge *that* something is the case from knowledge of *why* it is so, reserving the title of *epistéme* for the latter: a demonstration *propter quid* does not merely certify its conclusion, it displays the reasons that make it true. The distinction long appeared refractory to formal treatment, and it is a development of the last few years, due principally to Poggiolesi, that it has received one. In [12] the resources of proof theory are mobilized to formalize explanation from reasons: explanations are analysed as derivations in an *explanatory calculus* whose rules are so designed that their premisses do not merely entail their conclusion but constitute the total reasons why it is true.

The construction met, from an unexpected quarter, the theory of explainable classification. Darwiche and Hirth [34] define a *sufficient reason* for a Boolean classifier's decision as a minimal property of the instance that forces it — a prime implicant of the decision function — and the *complete reason* as the totality of the sufficient ones; Hill and Poggiolesi [13] have proved that the two enterprises coincide, suitably constructed derivations in the explanatory calculus providing all and only the sufficient reasons for a decision. There is thus a *fil rouge* running from Aristotle to machine learning: what contemporary explainable AI calls the reasons behind a decision is, in a mathematically precise sense, what the demonstrative tradition demanded of a proof *propter quid*.

This fixes where LCPL derivations stand, and the placement must be stated with care. A closed tableau is a refutation: it certifies its verdict, and — being analytic — it *localizes* the premisses actually consumed in its closures, so that a reading of the derivation can distinguish what did the inferential work from what was merely carried along. That reading is genuinely informative, and Section 3.2 exhibits it. It is not, however, the technical notion. The set of premisses consumed by the closures of a given closed tableau is relative to that tableau: it depends on the order in which the model chose to decompose, that is, on precisely the heuristic residue the framework leaves free, and it is not invariant across the closed tableaux for one and the same Π_0. Nor is it minimal, and sufficient reasons are minimal by definition, being prime implicants. What the closures deliver is therefore a set of premisses *sufficient for that derivation*, not a sufficient reason in the sense of [34]; obtaining the latter would require either a canonical strategy of decomposition or a subsequent minimisation, in which premisses are removed and the derivation attempted again. The constraint of Section 2.3 thus realizes, of the Aristotelian pair, the *hoti*, and points to the *dioti* as its natural refinement: an explanatory layer for LCPL, in which the model would be bound to rules of the kind introduced in [12], is the most promising direction in which the framework can grow, and we leave it to future work.

3. Descartes under the calculus: a worked classroom instance

Having set out the framework, we document its use in the classroom. The instance presented here was carried out within the modules of *Logica e Filosofia dell'IA* devoted, respectively, to logical methods — syntax and semantics of $\mathscr{L}_0$, analytic tableaux, soundness and completeness — and to the philosophy of artificial intelligence, where passages from philosophical classics are read against the conceptual

questions raised by generative systems [7]. The lesson brought the two modules to convergence on a single object: the inferential core of the *cogito* from Descartes' *Meditations*, formalized and then derived, under the LCPL constraint, by the Gemini Gem of Section 2.3.

3.1. The workflow of the lesson

The lesson was articulated in four phases, and the sequence embodies the division of labour argued for in Section 2.2. The class first read the source passage and reconstructed the argument informally; it then formalized it in $\mathscr{L}_0$, adopting the adapted version in Giordani's textbook [22, p. 127], so that the work consisted in fixing the interpretation key and translating the natural-language connectives. The argument was next recast as a sequent and the initial block Π_0 computed by Definition 2.2. Finally the block was submitted to the constrained model, and the derivation returned was checked by the students against the seven schemata, each line at a time, along the three dimensions of Section 2.3: the declared rule applied to a formula actually present in the parent block, each leaf genuinely closed, and no formula outside $\mathrm{Sub}^{\neg}(\Pi_0)$.

The choice of argument was deliberate. The *cogito* establishes a conclusion — your existence — by an ascent through thinking and knowing, so that the class was examining an argument *about* knowledge by means of a machine whose own epistemic standing the course had put in question. And it is small enough, four atoms and three premisses, that its subformula space fits on half a blackboard: this is what makes the perimeter check a task every student can perform rather than merely conceive.

3.2. The Cartesian argument from the *Meditations*

The argument is an adapted version of the inferential core of the *cogito* [35], as reconstructed by Giordani [22, p. 127]:

> You exist. For you think and, if you think, you know that you think. Moreover, you know that you exist if you know that you think. Finally, if you know that you exist, you exist.

The class fixed the interpretation key

$$\begin{array}{ll} p_1 : \text{you think}, & p_3 : \text{you know that you exist}, \\ p_2 : \text{you know that you think}, & p_4 : \text{you exist}, \end{array}$$

under which the argument has three premisses — $p_1 \wedge (p_1 \to p_2)$, gathering the fact of thinking and the reflexive principle that thought is transparent to itself; $p_2 \to p_3$, the epistemic bridge from knowing that one thinks to knowing that one exists; $p_3 \to p_4$, the factivity of knowledge — and the conclusion p_4. The corresponding sequent is

$$p_1 \wedge (p_1 \to p_2),\ p_2 \to p_3,\ p_3 \to p_4 \ \Rightarrow\ p_4,$$

whose associated block is $\Pi_{\mathrm{C}} = \{p_1 \wedge (p_1 \to p_2),\ p_2 \to p_3,\ p_3 \to p_4,\ \neg p_4\}$. The derivation returned by the constrained model, and checked in class, is the following closed tableau.[3]

$$\dfrac{\dfrac{\dfrac{\dfrac{p_1 \wedge (p_1 \to p_2),\ p_2 \to p_3,\ p_3 \to p_4,\ \neg p_4 \quad (\Pi_{\mathrm{C}})}{p_1,\ p_1 \to p_2,\ p_2 \to p_3,\ p_3 \to p_4,\ \neg p_4}\wedge}{\{\neg p_1,\ p_1,\ p_2 \to p_3,\ p_3 \to p_4,\ \neg p_4\}\times \quad | \quad p_2,\ p_1,\ p_2 \to p_3,\ p_3 \to p_4,\ \neg p_4}\to}{\{\neg p_2,\ p_2,\ p_1,\ p_3 \to p_4,\ \neg p_4\}\times \quad | \quad p_3,\ p_2,\ p_1,\ p_3 \to p_4,\ \neg p_4}\to}{\{\neg p_3,\ p_3,\ p_2,\ p_1,\ \neg p_4\}\times \quad | \quad \{p_4,\ p_3,\ p_2,\ p_1,\ \neg p_4\}\times}\to$$

[3] The derivation is displayed in the linear convention adopted for this calculus [17, 19]: the root block stands at the top, the rule applied is recorded beside each inference line, every node carries the entire current block written out in full, and the two children produced by a β-rule are written on the same line, separated by |, a terminal block being enclosed in braces and marked × when closed. The branching is thus recorded compactly rather than drawn.

Every leaf closes, so by Theorem 2.5 the sequent is valid: the argument is formally impeccable, and whether it is also *sound* depends on whether one accepts its premisses — a distinction, between validity and soundness, that the derivation makes visible rather than presupposing, and on which the class discussion of the epistemic bridge $p_2 \to p_3$ turned. The structure of the tableau rewards attention. After the single α-step that unpacks the first premiss, the derivation is a cascade of three applications of $(\to)$, each opening a left branch that closes at once against a formula already secured — the refutational image, block by block, of three chained *modus ponens* steps carrying p_1 to p_2, p_2 to p_3, p_3 to p_4, where the final closure $\{p_4, \neg p_4\}$ records the collision between the conclusion reached and its assumed negation. And the derivation is visibly self-contained in the sense of Proposition 2.7: every formula on the page is a subformula, or the negation of a subformula, of the cogito's own material. The model could not have imported the veracity of God, the malicious demon, or any other Cartesian doctrine, however salient such material may be in its training corpus: within the perimeter of the proof there is no rule by which it could enter.

The derivation also illustrates, in miniature, the position assigned to LCPL derivations in Section 2.4, and this was the second reading the class was guided through. Consider what each closure *consumes*. The three left-hand closures use, respectively, p_1 against $\neg p_1$, p_2 against $\neg p_2$ and p_3 against $\neg p_3$ — each recording that the escape route offered by a conditional premiss, the falsity of its antecedent, is already barred by material previously secured — while the final closure uses the conclusion reached against its assumed negation. Taken together, the closures localize the premisses that do the inferential work: the atom p_1 and the three conditionals, each activated where its antecedent has just been obtained. Reading them off, one recovers the chain that answers the question *why*: you think; thought, being transparent to itself, yields knowledge of thinking; knowledge of thinking yields knowledge of existing; and knowledge is factive. The distinction was made visible by experiment, adding to Γ_A a redundant premiss and observing that the tableau closes without ever touching it: the premiss appears in every block and in no closure. Two cautions were stated in the same breath. Had the model decomposed in a different order, the closures might have consumed a different set; and nothing guarantees that the set so obtained is minimal. What the derivation offers is a first step from the *hoti* towards the *dioti*, not the arrival.

One interpretive cost of the formalization was itself made explicit in class: $\mathscr{L}_0$ flattens the epistemic operator, so that "you know that you think" is an unstructured atom p_2 rather than a modality applied to p_1, and reflexivity and factivity must be assumed as premisses rather than derived from the logic of knowledge. This is not a defect of the exercise but one of its intended outcomes: it motivates, from within a concrete derivation, the passage to the modal and epistemic extensions of the calculus, which the same block format supports [11] and which the final year of the curriculum introduces.

3.3. What the derivation teaches

The derivation, taken as a classroom object, delivered three things, and it is worth separating what the framework guarantees from what is entrusted to the humans using it.

It was checked, not believed. Each step instantiates one of seven schemata, and the check requires no access to, and no trust in, the underlying model: a closed tableau is a certificate whose validity is independent of its generator. The students performed it line by line, doing at their scale exactly what a mechanical proof checker does; and the intrusion of foreign material was presented to them not as a vague tendency of chatbots to invent, but as the precise syntactic event Proposition 2.7 makes it. They were told, with equal precision, what the check does not settle: that a tableau might have been left unfinished, which is why every leaf must be inspected too.

It located the boundary between what the calculus certifies and what it cannot. The validity of the sequent is guaranteed; the adequacy of the formalization is not. Whether the conditionals render Descartes' text faithfully, whether the epistemic bridge is a premiss he would have granted in this form, whether flattening "you know that" into an atom does violence to the argument — these were argued philologically and philosophically, and the class understood that they *could not* be settled by the machine: the hermeneutic responsibility the architecture deliberately leaves where it belongs [11].

And it gave the students a concrete hold on the distinction, met in the abstract while reading Aristotle, between certifying a conclusion and explaining it. The closure-reading showed where in a derivation the reasons live, and the theorem of [13] allowed us to say that this is no didactic metaphor but the point at which a tradition running from the *Posterior Analytics* through Bolzano meets the current mathematics of explainable classification [34]. The lesson thereby gave *epistemic resistance* an operational content: the students did not merely hear that one should demand reasons from an artificial interlocutor, but interacted with one made structurally incapable of answering without exhibiting them.

3.4. Scope of the evidence, and a protocol for evaluation

What Section 3 reports is a worked instance, not an empirical evaluation, and we state the difference rather than blur it. A single argument, however carefully chosen, establishes that the constrained regime produces an object of the intended kind and that the object is checkable by the intended audience; it does not measure how often the model complies, nor how much students gain. Both are measurable, and the analyticity of the calculus is what makes the first measurable mechanically.

The evaluation we are preparing has two arms. The first is a batch of N arguments from the philosophical curriculum, balanced between valid and invalid, each submitted to the same model under two regimes, constrained and free, the free regime serving as the control the comparison requires. On the constrained arm four quantities are decidable by a checker of a few lines: the proportion of outputs that are well-formed tableaux; the rate of rule violations per step; the rate of perimeter violations, that is, of formulae outside $\mathrm{Sub}^{\neg}(\Pi_0)$; and the rate of false negatives, tableaux left unfinished or branches declared completed open in error. Verdict accuracy is decidable on both arms against the calculus, which for the propositional fragment settles validity by Theorem 2.5. The second arm concerns the students: agreement between their line-by-line checks and the mechanical checker, the time a check takes, and the errors missed, before and after instruction. The two arms answer different questions — whether the machine complies, and whether its compliance is legible to the learner — and neither substitutes for the other.

4. Conclusion

The difficulties surveyed in Section 1 are, at bottom, difficulties of *form*: an output that can only be trusted or doubted invites over-reliance, carries no visible mark when it is wrong, and never exposes the planning and monitoring stages on which reflective control depends [3, 4, 5]. A proof-theoretic constraint addresses each of these where it arises [10]. When the output is a derivation in an analytic calculus it is neither trusted nor doubted but *checked*, and the check is within the learner's reach; a wrong step is not a matter of opinion but a visible violation of one of seven schemata or of the subformula discipline; and the derivation externalises exactly the stages a bare answer conceals, so that following it is itself an exercise in orienting, planning and monitoring. In the terms of that literature, the constrained system is built to convert a replacement use of the tool into a diagnostic and, in time, an elaborative one: the student who checks a derivation is not delegating the reasoning but reconstructing it.

Three boundaries should be restated, lest the proposal claim more than it establishes. The guarantee is conditional and confined to the derivation: it certifies that the conclusion follows from the declared premisses, provided the exhibited tableau is well formed and complete, and says nothing about the faithfulness of the formalisation to the source text — the hermeneutic question, deliberately left where the formative work resides. The premisses a derivation's closures consume are those of that derivation, and not, without further work, the sufficient reasons of the theory of explainable classification. And the calculus is propositional; arguments in which knowledge or necessity are operators rather than atoms call for the modal extensions the same block format supports. Within these limits the instance makes good on the thesis with which we began. The ancient demand that knowledge be able to render its *why* and the modern demand that artificial intelligence be explainable are, we have argued, one demand in two idioms; and a calculus conceived for the first turns out to answer the second, not by adding an

explanation to a decision but by making the decision and its explanation the same object. That is the sense in which the classical inheritance does not merely precede the present but illuminates it — and the sense in which a *lyceum* that teaches logic beside philosophy is, for the age of generative AI, not an anachronism but an unexpectedly exact instrument.

Acknowledgments

I would like to express my sincere gratitude to Francesco Scarcello and Roberto Beneduci of the University of Calabria for conceiving and directing the Cl@ssico Digitale project. I am deeply grateful for their trust, for granting me complete intellectual freedom to develop my own ideas, and for enabling me to expand this initiative into a genuine research domain.

Declaration on Generative AI

No generative AI tools or technologies were used in the conceptualization, drafting, editing, or content generation of this manuscript.

References

[1] B. Pepin, N. Buchholtz, U. Salinas-Hernández, A scoping survey of ChatGPT in mathematics education, Digital Experiences in Mathematics Education 11 (2025) 9–41.
[2] B. Pepin, N. Buchholtz, U. Salinas-Hernández, Mathematics education in the era of ChatGPT: investigating its meaning and use for school and university education. Editorial to special issue, Digital Experiences in Mathematics Education 11 (2025) 1–8.
[3] N. Noster, S. Gerber, H.-S. Siller, Pre-service teachers' approaches in solving mathematics tasks with ChatGPT, Digital Experiences in Mathematics Education (2025).
[4] T. J. Gurl, A. P. Markinson, A. F. Artzt, Using ChatGPT as a lesson planning assistant with preservice secondary mathematics teachers, Digital Experiences in Mathematics Education (2025).
[5] F. Contel, A. Cusi, Investigating the role of ChatGPT in supporting metacognitive processes during problem-solving activities, Digital Experiences in Mathematics Education 11 (2025) 167–191.
[6] Z.-J. Kock, U. Salinas-Hernández, B. Pepin, Students' initial use schemes of ChatGPT as an instrument for learning, Digital Experiences in Mathematics Education (2025).
[7] R. Beneduci, S. Cuconato, F. Scarcello, Il Liceo Classico nell'età dell'Intelligenza Artificiale: la sperimentazione Cl@ssico Digitale, Umanistica Digitale (2027). In press.
[8] S. Cuconato, K. Reale, Logica e Intelligenza Artificiale: pratiche didattiche per il Liceo Classico, in: Quarto Convegno Italiano sulla Didattica dell'Informatica (ITADINFO 2026), 2026. In press.
[9] E. Agazzi, Scientific Objectivity and Its Contexts, Springer, Cham, 2014.
[10] S. Cuconato, F. Scarcello, R. Beneduci, Logical proof in AI education: Epistemological foundations and practical applications for a human-centered perspective, in: SIpEIA Conference 2026, Sapienza Università di Roma, Roma, 2026. In press.
[11] S. Cuconato, Ensuring inferential reliability in clinical LLMs: a proof-theoretic approach via the ISA system, 2025. Preprint, under review.
[12] F. Poggiolesi, (conceptual) explanations in logic, Journal of Logic and Computation (2024) 1–34.
[13] B. Hill, F. Poggiolesi, Explaining with reasons: from Aristotle to machine learning classifiers, The Review of Symbolic Logic 18 (2025) 1068–1089.
[14] G. Gentzen, Untersuchungen über das logische Schließen I–II, Mathematische Zeitschrift 39 (1935) 176–210, 405–431.
[15] S. Negri, J. von Plato, Structural Proof Theory, Cambridge University Press, Cambridge, 2001.
[16] R. M. Smullyan, First-Order Logic, Springer, Berlin–Heidelberg, 1968.
[17] S. Cuconato, Metodi Logici 1. Teoria degli Insiemi, Logica del Primo Ordine, Tableaux Analitici, Calcolo dei Sequenti, Il Sileno Edizioni, Lago (CS), 2025.

[18] S. Cuconato, Proof theory for sequent-style tableaux: G_0- and G_3-style sequent calculi and full normalization, 2026. URL: https://arxiv.org/abs/2607.28555. arXiv:2607.28555, preprint.
[19] S. Cuconato, Sequent-style tableaux for first-order logic: Structural analysis, cut admissibility, and the correspondence with LK, 2026. URL: https://arxiv.org/abs/2607.29193. arXiv:2607.29193, preprint.
[20] S. Cuconato, Sequent-style tableaux for intuitionistic propositional logic, Reports on Mathematical Logic 61 (2026). In press.
[21] F. Poggiolesi, On the importance of being analytic. The paradigmatic case of the logic of proofs, Logique et Analyse 55 (2012) 443–461.
[22] A. Giordani, Introduzione alla logica, EDUCatt Università Cattolica, Milano, 2016.
[23] F. Cheng, H. Li, F. Liu, R. van Rooij, K. Zhang, Z. Lin, Empowering LLMs with logical reasoning: a comprehensive survey, 2025. URL: https://arxiv.org/abs/2502.15652. arXiv:2502.15652, preprint.
[24] L. Pan, A. Albalak, X. Wang, W. Y. Wang, Logic-LM: empowering large language models with symbolic solvers for faithful logical reasoning, in: Findings of the Association for Computational Linguistics: EMNLP 2023, 2023, pp. 3806–3824.
[25] T. X. Olausson, A. Gu, B. Lipkin, C. E. Zhang, A. Solar-Lezama, J. B. Tenenbaum, R. Levy, LINC: a neurosymbolic approach for logical reasoning by combining language models with first-order logic provers, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023), 2023.
[26] X. Ye, Q. Chen, I. Dillig, G. Durrett, SatLM: satisfiability-aided language models using declarative prompting, in: Advances in Neural Information Processing Systems 36 (NeurIPS 2023), 2023.
[27] A. Q. Jiang, S. Welleck, J. P. Zhou, W. Li, J. Liu, M. Jamnik, T. Lacroix, Y. Wu, G. Lample, Draft, sketch, and prove: guiding formal theorem provers with informal proofs, in: International Conference on Learning Representations (ICLR 2023), 2023.
[28] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nature Machine Intelligence 1 (2019) 206–215.
[29] D. Hilbert, Über das Unendliche, Mathematische Annalen 95 (1926) 161–190.
[30] M. T. Ribeiro, S. Singh, C. Guestrin, "Why should I trust you?": explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.
[31] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Advances in Neural Information Processing Systems 30 (NeurIPS 2017), 2017, pp. 4765–4774.
[32] M. Turpin, J. Michael, E. Perez, S. R. Bowman, Language models don't always say what they think: unfaithful explanations in chain-of-thought prompting, in: Advances in Neural Information Processing Systems 36 (NeurIPS 2023), 2023.
[33] Aristotle, Posterior Analytics, 2nd ed., Clarendon Press, Oxford, 1993. Translated with a commentary by Jonathan Barnes.
[34] A. Darwiche, A. Hirth, On the (complete) reasons behind decisions, Journal of Logic, Language and Information 32 (2023) 63–88.
[35] R. Descartes, Meditationes de prima philosophia, Cambridge University Press, Cambridge, 1996. 1st ed. 1641. English translation in: John Cottingham (ed.), *Meditations on First Philosophy*.