



ELSA+: AN ASSESSMENT-READY LEARNING PROGRESSION FOR TEACHING RECURSION TO PRE-SERVICE INFORMATICS TEACHERS

Vuong T. PHAM

Abstract: Recursion is a compact idea with an outsized cognitive cost for novices, and it becomes even more consequential in pre-service informatics teacher education, where future teachers must learn to diagnose and respond to students' reasoning. We introduce ELSA+, an assessment-ready learning progression that organizes recursion understanding into five cognitive layers tied to recurring misconceptions, observable indicators, and canonical tasks. An analytic 0–3 rubric supports diagnosis by privileging explanation quality and structural reasoning over trace completion alone. Initial evidence supports the instrument's readiness for use: an expert panel ($n = 8$) indicated excellent relevance ($S\text{-}CVI/Ave_Relevance = 1.00$) and high clarity ($S\text{-}CVI/Ave_Clarity = 0.95$), and a vignette-based calibration with three independent raters showed substantial agreement (Krippendorff's $\alpha = 0.84$; exact agreement = 52%; within-one-level agreement = 96%). Classroom impact remains future work; however, the paper provides a transparent construction method and a practical measurement tool to enable subsequent efficacy studies in teacher-education settings.

Key words: recursion; computing education; teacher education; learning progression; diagnostic assessment

MSC 2010: 97Q70; 97Q60

1. Introduction

This article reports a methodological, instrument-development contribution rather than a classroom efficacy experiment. In recursion teaching, claims about “what works” are difficult to interpret when the field lacks a shared diagnostic construct, a stable family of tasks, and scoring guidance that can be applied consistently. ELSA+ is offered as that measurement infrastructure: a learning progression, a task family, and an analytic rubric, supported by initial evidence of content validity and scoring usability, to make later classroom studies more informative and comparable.

Recursion is conceptually compact yet cognitively demanding. To use recursion competently, learners must coordinate termination (base case), problem reduction (self-reference), and return-time composition (how intermediate results are combined) while maintaining a coherent model of nested calls. A long-standing body of computing education research reports recurring misconceptions and fragmented mental models, including difficulty identifying base cases and reasoning about unwinding. Consequently, novices often fall back on mechanical tracing, which can produce correct outputs without supporting durable structural understanding and transfer.

In teacher education, the challenge is amplified: pre-service teachers must not only solve recursion problems but also diagnose students' reasoning breakdowns and choose appropriate scaffolds. Trace-based instruction and visualization-based tools are useful, but they do not, on their own, provide a shared

Received March 2026.

Cite as: Pham, V. T. (2026). ELSA+: An Assessment-Ready Learning Progression for Teaching Recursion to Pre-Service Informatics Teachers. *Acta Didactica Napocensia*, 19(1), 221-235, <https://doi.org/10.24193/adn.19.1.17>

diagnostic language that links observed errors to targeted instructional moves. Recent reviews suggest that the field benefits from frameworks that are explicit about what counts as evidence of understanding and how competence develops across tasks (McCauley et al., 2015; Rinderknecht, 2014).

This paper proposes ELSA+, a learning progression that operationalizes recursion understanding as five cognitive layers, each defined by observable indicators, canonical tasks, typical misconceptions, and suggested scaffolding/fading moves. The framework is designed as an assessment-ready instrument for teacher education rather than a taxonomy of problem types.

Research questions (RQs) guiding this contribution are:

RQ1: What recurring misconceptions and reasoning breakdowns characterize novice and pre-service teacher learning of recursion?

RQ2: How can these breakdowns be organized into a learning progression with observable criteria to inform instructional decisions?

RQ3: What assessment artifacts and rubric can support diagnostic evaluation of recursion understanding beyond trace correctness?

The remainder of the paper reviews related work, presents the methodology for model construction, specifies the ELSA+ layers and their operationalization, and discusses implications and limitations.

Throughout the paper, “assessment-ready” is used in a practical, design sense: ELSA+ specifies observable criteria, canonical tasks, and scoring guidance that instructors can apply immediately for diagnostic feedback and scaffolding decisions. It does not yet claim full psychometric validation (e.g., reliability and construct/predictive validity on authentic learner artifacts), which is positioned as the next empirical phase.

2. Related Work

2.1. Why Recursion Remains Difficult for Novices

Recursion consistently appears as a challenging concept for novices because it requires a coherent mental model of both forward expansion (recursive calls) and backward return flow (unwinding). Studies of mental models report that learners frequently fail to connect the meaning of the base case, the reduction step, and return-time composition, leading to brittle reasoning and poor transfer. Common breakdowns include ad hoc base cases, confusion across call frames, expectation of immediate results before unwinding, and inability to map recursive structure to the call stack or to data structures such as trees (Götschi et al., 2003; Haberman & Averbuch, 2002). More recent work continues to report persistent comprehension challenges and mental-model gaps, including studies that contrast tail and non-tail recursion and review programmers’ mental models (Baron & Feitelson, 2024; Heinonen et al., 2023).

2.2. Trace-Based and Visualization-Based Instruction

Trace-based instruction makes execution explicit through stepwise traces, stack listings, and manual simulation. This approach can reduce initial mystery and support debugging, but tracing success does not necessarily imply structural understanding; learners may execute procedures without forming schemas that generalize.

Visualization-based approaches externalize recursion’s hidden dynamics by coordinating representations such as recursion trees and call-stack views. For instance, visual representations such as RainbowScheme employ color and structured environments to support reasoning about recursion in functional programs (Tung et al., 2001). In the imperative/algorithmic space, systems such as SRec use coordinated views (e.g., traces, control stack, and recursion tree) to support interactive analysis of recursive algorithms (Velázquez-Iturbide et al., 2009; Velázquez-Iturbide & Pérez-Carrasco, 2010). However, evidence indicates that visualization is most effective when coupled with active engagement and guidance; a meta-study of

algorithm visualization concludes that learning gains depend strongly on how students interact with visualizations (Hundhausen et al., 2002), and broader reviews emphasize the need for careful instructional integration to avoid superficial focus on animation (Sorva et al., 2013).

2.3. Cognitive Load and Learning from Examples

From a cognitive perspective, recursion imposes high intrinsic load because learners must coordinate multiple interacting elements while maintaining nested state. Cognitive Load Theory (CLT) argues that instruction should reduce extraneous load and support schema acquisition (Sweller, 1988). Worked examples and structured example-based supports can be especially helpful for novices, and synthesis work in computing education highlights that examples and tracing can be effective when designed to foreground deep structure and followed by appropriate fading (Paas et al., 2003; Sweller, 1988). For recursion, this implies that examples should explicitly highlight base-case sufficiency, decreasing measures, and return-time composition rather than only enumerating call steps.

2.4. Scaffolding and Learning Progressions

Scaffolding perspectives emphasize contingent support that enables learners to perform beyond unaided capability and is gradually withdrawn as competence grows. The classic formulation by Wood, Bruner, and Ross (1976) aligns with recursion instruction, where learners often need structured prompts, partial representations, and guided explanation to stabilize mental models. Yet many recursion interventions lack assessment-ready learning progressions that connect observed indicators to instructional moves across increasingly complex forms of recursion.

2.5. Survey-Level Synthesis and Gap

Survey-level evidence further clarifies both the persistence of recursion difficulty and the need for coherent progressions. McCauley et al. (2015) synthesize research on teaching and learning recursive programming, highlighting recurring mental-model breakdowns and the limited availability of robust, assessment-oriented instruments. Rinderknecht (2014), published in *Informatics in Education*, surveys approaches and perspectives for recursion teaching, underscoring that instructional framing shapes what learners attend to and what misconceptions persist. These surveys motivate a framework that complements tracing and visualization by specifying observable indicators of understanding and a progression usable for diagnostic assessment in teacher education. Even with recent tutorial-based instructional supports (e.g., RecurTutor and BTRrecurTutor), the literature still offers limited assessment-ready progressions that shift scoring from trace correctness toward explanation quality and transferable reasoning (Hamouda et al., 2019, 2020).

3. Methodology: Theoretical Model Construction

To build ELSA+, we followed a design-oriented model-construction procedure. We (i) assembled and documented an evidence base on recursion misconceptions and instructional approaches, (ii) extracted and coded difficulties into operational indicators and candidate competencies, (iii) defined layer boundaries using explicit rules informed by cognitive load and scaffolding theory, and (iv) conducted initial expert review and rater calibration to test whether the resulting definitions and rubric are interpretable and usable for assessment.

3.1. Evidence base and literature sampling

Search strategy and scope. We performed a focused, reproducible search of education and computing-education literature on recursion learning, misconceptions, and assessment in teacher education. Sources included major indexing services (e.g., Google Scholar and discipline venues in computing education) and backward/forward citation chaining from seminal works. Search strings combined terms such as "recursion"

AND (misconception OR mental model OR novice) AND (teaching OR instruction OR assessment OR learning progression).

Selection criteria. We included peer-reviewed studies and syntheses that (i) reported empirically observed recursion difficulties/misconceptions, (ii) evaluated or described instructional approaches relevant to conceptual understanding (not only programming syntax), or (iii) provided assessment approaches for recursion understanding. We excluded non-peer-reviewed sources and works focusing on optimization-only topics without conceptual recursion reasoning.

Selection flow. Retrieved = 214 records; screened = 187 after deduplication and title/abstract checks; included = 25 for full synthesis.

3.2. Extraction and coding of misconceptions and competencies

Extraction unit. From each included source, we extracted statements describing learner errors, misconceptions, or breakdowns in recursion reasoning (e.g., base case misuse, incorrect reduction, confusion about return-time composition, misunderstanding of mutual recursion, and redundancy in overlapping subproblems). We also extracted proposed instructional moves and assessment cues when available.

Coding scheme. Extracted statements were coded into provisional categories aligned with five cognitive modes. Codes were iteratively refined through constant comparison: new statements were either mapped to an existing code, prompted refinement of indicator wording, or motivated splitting/merging of indicators. Ambiguous cases were resolved by author consensus with explicit notes recorded in a revision log.

3.3. Layer boundary rules

Each layer in ELSA+ is defined by a dominant reasoning demand and the kind of explanatory coordination required: (L1) recognizing and justifying base cases and termination; (L2) expressing self-referential reduction with a decreasing measure and assume-subsolution reasoning; (L3) composing multiple recursive results at return-time (multi-branch composition); (L4) coordinating two or more mutually dependent recursive definitions as a state-transition system; and (L5) controlling redundancy in overlapping subproblems via memoization/tabulation, emphasizing dependency order rather than general algorithmic optimization.

Boundary cases. When an artifact exhibited features spanning adjacent layers, scoring prioritized the highest layer evidenced by explicit explanation quality (e.g., stating a decreasing measure moves beyond L1; articulating return-time composition moves beyond L2).

3.4. Expert-based initial validation (content validity)

Panel. We conducted an expert review with Vietnamese informatics educators (target $n = 8$) balanced across university teacher-education lecturers and secondary-school teachers. Experts were selected for demonstrated experience in teaching programming/algorithms and/or preparing informatics teachers. All experts met eligibility criteria of at least three years of experience teaching recursion or programming/algorithms and/or preparing informatics teachers.

Instrument and procedure. Experts rated ELSA+ items (layer definitions, observable indicators, canonical tasks, and rubric descriptors) on a 4-point scale for Relevance and Clarity. Ratings of 3-4 were treated as acceptable. Open-ended prompts captured suggested revisions, especially for layer boundaries and rubric usability.

Analysis. We computed item-level content validity indices (I-CVI) and scale-level indices (S-CVI/Ave) separately for Relevance and Clarity. Expert panel ratings indicated strong content alignment: S-CVI/Ave_Relevance=1.00, S-CVI/Ave_Clarity=0.95. Lowest clarity items were concentrated in Layer 5 (Optimization), notably Definition (I-CVI=0.63) and Tasks/Rubric (I-CVI=0.75).

3.5. Vignette-based rater calibration and reliability

In the vignette study, three independent raters (R1–R3) scored the 25 standardized vignettes using the 0–3 rubric; vignettes were designed to elicit evidence for a target layer and level.

Materials. To test rubric usability without requiring a full classroom deployment, we developed a standardized vignette set of 25 short student responses (five per layer) covering performance levels 0–3. Each vignette includes a prompt, a brief response, and a target layer for scoring. The vignettes were authored by the research team as standardized, idealized artifacts synthesized from patterns in the literature and aligned to the rubric; they are not raw classroom student submissions.

Raters and procedure. Three raters independently scored all vignettes using the ELSA+ rubric: a university lecturer in algorithms/recursion, an education-focused lecturer with assessment expertise, and a specialized high-school informatics teacher. After an independent first pass, raters discussed discrepancies to calibrate interpretations and refined rubric anchors where needed (recorded in the revision log).

Reliability analysis. We estimated ordinal inter-rater agreement using Krippendorff's alpha (ordinal distance) and reported complementary agreement rates (exact and within-one-level). On the 25-vignette calibration set, agreement was substantial (Krippendorff's alpha = 0.84), with 52% exact agreement and 96% within-one-level agreement.

4. The ELSA+ Framework

4.1. Design Principles

ELSA+ is guided by three design principles. First, it prioritizes explanation quality and transfer over trace correctness: learners should be able to justify why recursion works, not only simulate it. Second, it treats tracing and visualization as supports rather than endpoints, encouraging fading from execution-level simulation to structural reasoning. Third, it is assessment-ready: each layer includes observable evidence and canonical tasks that can be used for diagnosis and feedback.

4.2. Five Layers as Cognitive Modes

ELSA+ defines five layers of recursion understanding:

L1 Base case reasoning: justify termination and boundary correctness.

L2 Self-referential decomposition: specify a decreasing measure and assume a sub-solution for a smaller instance.

L3 Multi-branch composition: explain return-time combination across branches using recursion-tree reasoning.

L4 Mutual recursion: model interdependent definitions as state transitions or alternation with consistent base cases.

L5 Optimization: recognize redundancy and derive memoization/tabulation from recursion structure.

The layers are intended as a progression, but learners may show uneven profiles (e.g., strong L1–L2 but weak L3). The goal is to support diagnosis and targeted scaffolding rather than enforce a rigid sequence.

Scope note: Layer 5 (Optimization) refers specifically to redundancy control in recursive structure (overlapping subproblems and reuse via memoization/tabulation), not to general algorithmic optimization or asymptotic analysis. To avoid misinterpretation: Layers 1–4 constitute the core recursion reasoning progression; Layer 5 is intentionally framed as an advanced/extension layer for contexts where pre-service teachers must reason about redundancy and memoization/dynamic programming.

Figure 1. The ELSA+ assessment-ready learning progression for recursion

L1 Base case reasoning	Operational focus: Termination + boundary correctness Observable evidence: Base-case list + brief justification Instructional support: Boundary-input checklist; fade prompts
L2 Self-referential decomposition	Operational focus: Decreasing measure + assume sub-solution Observable evidence: Reduction statement + invariant note Instructional support: Invariant prompts; fade tracing
L3 Multi-branch composition	Operational focus: Return-time combination across branches Observable evidence: Recursion tree / return-flow explanation Instructional support: Tree/stack mapping; fade visualization
L4 Mutual recursion	Operational focus: Interdependence as state transition/alternation Observable evidence: Alternation rule + state diagram / refactoring Instructional support: State-machine framing; fade diagram
L5 Optimization	Operational focus: Redundancy control (memoization/tabulation) Observable evidence: Overlap marking + memo/DP table order Instructional support: Dependency table/graph; fade hints

Usage in instruction: Diagnose → Scaffold → Fade → Evaluate transfer (rubric 0–3 per layer)

As shown in Figure 1, each ELSA+ layer specifies an operational focus, observable evidence, and an instructional support that can be scaffolded and gradually faded.

Table 1. Comparison of trace-based instruction, visualization-based instruction, and ELSA+.

Note. The comparison highlights the dominant assessment focus of each approach. “Explanation quality” refers to the coherence and correctness of a student’s conceptual account beyond producing running code.

Dimension	Trace-based instruction	Visualization-based instruction	ELSA+ (this work)
Primary goal	Make execution steps explicit via stepwise simulation.	Externalize hidden dynamics (stack/tree) and link representations.	Provide diagnostic progression: what reasoning competence looks like across layers.
Main representations	Trace tables; line-by-line stepping; call-stack listings.	Coordinated views: recursion tree, call stack, animations, graphs.	Layer descriptors + observable indicators + tasks and rubric.
Typical learner activity	Produce traces; follow call order; compute intermediate state.	Interpret visuals; map between stack ↔ tree ↔ code.	Explain base case/reduction/composition; diagnose errors; transfer across forms.

Strengths	Concrete; accessible for beginners; supports basic debugging.	Builds representational fluency; supports structural mapping.	Assessment-ready; supports teacher decision-making; emphasizes explanation and transfer.
Common limitations	Risk of procedural tracing without understanding; weak transfer.	Tool dependence; cognitive overload if unguided.	Requires careful task design; needs empirical validation across contexts.
Assessment operationalization	Trace accuracy; output correctness on familiar examples.	Representation mapping accuracy; interpretation of visualized execution.	Explanation quality + transfer tasks + layer-specific rubric scores (beyond trace/output correctness).

Table 2. ELSA+ layers as an assessment-ready progression: operational definition, observable evidence, canonical tasks, misconceptions, and instructional moves.

Note. ELSA+ layers are ordered from foundational to advanced; Layer 5 is optional/advanced in some curricula. Rubric performance levels are scored on a 0–3 scale (0 = absent/incorrect; 3 = proficient transfer).

Layer	Operational definition	Observable evidence	Canonical task	Typical misconception	Instructor move (scaffold → fade)
L1 Base case	Defines termination + boundary correctness.	• Base case(s) for boundary inputs • Short justification	State base case(s) for $f(n)$ and justify for $n=0/1/\dots$	• Ad hoc base case • Misses edge inputs	Ask: “Which inputs must stop?” “What must be returned?” → fade to student boundary list
L2 Reduction	Decreasing measure + “assume sub-solution”.	• Reduction/recurrence • Measure + invariant noted	Describe what decreases, what stays invariant, and how $f(\text{smaller})$ is used.	• Recursion as repetition • No decreasing measure	Prompt invariant: “What gets smaller? What remains true?” → fade prompts; justify without tracing
L3 Composition	Return-time composition across branches.	• Combine logic correct • Tree/return-flow note	Given two recursive calls, explain how returns are combined and why.	• Can trace, not explain unwinding • Call-order confusion	Use tree: “What does each child return mean?” → fade visualization; require structural explanation
L4 Mutual	Interdependent definitions as state transition.	• Alternation rule • State/grammar sketch	Explain switching logic between f and g ; define base cases; argue termination.	• Functions treated independently • Loses switching rule	Reframe as state machine: “What state after call?” → fade diagram; generalize to new example
L5 Optimization	Redundancy control (memo/DP).	• Marks overlap • Memo key/table order	Mark repeated subcalls and propose memo/DP: key	• Notices slowness, no systematic fix • Inconsistent memoization	Ask: “Which subproblem repeats? What indexes it?” → fade hints;

			and dependency order.		justify memo vs tabulation
--	--	--	-----------------------	--	----------------------------

Note. Each layer is operationalized by observable evidence rather than by problem type alone. Tracing and visualization may be used as supports, but ELSA+ specifies what learners should be able to justify and construct at each stage, enabling diagnostic assessment and principled scaffolding/fading decisions.

5. Instructional Design and Assessment with ELSA+

5.1. Module structure for teacher education

A practical way to use ELSA+ in pre-service teacher education is to structure a recursion module as: (i) diagnostic elicitation, (ii) targeted scaffolding by layer, (iii) guided practice with fading, and (iv) reflective consolidation.

Diagnostic elicitation should collect short artifacts aligned to Table 1 (e.g., base-case justification; reduction statement; return-time composition explanation; alternation rule; redundancy identification). Instructors can then localize the learner's current breakdown and choose supports accordingly (e.g., visualization for L3, state diagrams for L4, dependency tables for L5). For Layer 4 (mutual recursion), a minimal entry example (e.g., `is_even` calling `is_odd` and vice versa) is recommended before progressing to more abstract alternation cases (e.g., grammar/parser-style mutual recursion).

5.2. Assessment artifacts and scoring

ELSA+ supports assessment at two levels. First, formative diagnosis uses brief tasks aligned to each layer, scored with the rubric in Appendix A. Second, summative evaluation can combine rubric scores with transfer tasks that differ in surface features from worked examples. Recommended artifacts include: (a) written explanations, (b) annotated recursion trees or call graphs, (c) code with invariant annotations, and (d) error analyses of flawed recursive solutions.

To reduce subjectivity, scoring should be criterion-referenced (0–3 levels) and anchored with exemplar responses for each layer. If used in research or program evaluation, a second rater can score a subset of artifacts to estimate agreement.

5.3. Initial expert validation results

We summarize expert-panel evidence for content validity using I-CVI and S-CVI/Ave (Relevance and Clarity) and document resulting revisions. Key indices: S-CVI/Ave_Relevance = 1.00; S-CVI/Ave_Clarity = 0.95.

Table 3. Content validity (Clarity) by layer and component (n = 8 experts).

Layer	Definition (Clarity I-CVI)	Indicators (Clarity I-CVI)	Tasks (Clarity I-CVI)	Rubric (Clarity I-CVI)	Notes (main revision theme)
L1	1.00	1.00	1.00	1.00	Minor wording clarifications on base case and termination cues.
L2	1.00	1.00	1.00	1.00	Clarified self-referential decomposition language and prompts for subproblem identification.
L3	1.00	1.00	0.88	1.00	Strengthened return-time composition prompts;

					added transfer-oriented task wording.
L4	1.00	0.88	1.00	1.00	Added simpler mutual recursion example (is_even/is_odd) before abstract parser/grammar cases.
L5	0.63	0.88	0.88	0.88	Clarified optimization scope (recursive-structure optimization via memoization/tabulation), refined boundary wording.

Note. I-CVI is the proportion of experts rating an item as acceptable (3–4) on a 4-point scale. S-CVI/Ave summarizes the average I-CVI across items. Reported indices: S-CVI/Ave_Relevance = 1.00; S-CVI/Ave_Clarity = 0.95.

5.4. Vignette-based rater calibration and inter-rater agreement

Using the standardized 25-vignette set, three raters independently scored each response on the 0-3 scale for the target layer. Agreement was computed using an ordinal reliability coefficient, supplemented by exact and within-one-level agreement. Krippendorff's $\alpha=0.84$; exact agreement=52%; within-one-level=96%.

Illustrative example vignette (excerpt; full set in Appendix B supplementary file):

Prompt: What is the base case for factorial?

Student response: “Base case is $n=1$ return 1. I did not think about $n=0$.”

Assigned score: 1 (Emerging)

Rationale: The response identifies a familiar base case but misses boundary inputs ($n=0$) and provides no sufficiency justification, matching the Layer 1 score-1 descriptor.

Calibration outcomes. Disagreements clustered around boundary cases (e.g., L2 vs L3 when responses mention both reduction and combination, and L5 scope when responses discuss 'optimization' broadly). These cases motivated refinement of rubric anchors and added transfer examples. Examples of expert-driven edits included: (i) delimiting Layer 5 as redundancy control via memoization/tabulation (rather than general algorithm optimization) and adding a transfer-oriented anchor; (ii) adding a concrete mutual-recursion starter (is_even/is_odd) and state-transition phrasing for Layer 4; (iii) tightening the Layer 2 vs Layer 3 boundary by emphasizing reduction/invariant reasoning versus return-time composition. Note on tail recursion: when the recursive call is in tail position (no post-call combination), Layer 3 evidence can be demonstrated by explicitly explaining why no return-time operation remains and how any accumulation/state is carried through parameters.

6. Discussion

ELSA+ complements trace-based and visualization-based instruction by shifting the instructional focus from “seeing recursion run” to “explaining why recursion works” at progressively richer levels. This aligns with a cognitive load perspective: traces and visualizations can reduce extraneous search for novices, but must be coupled with prompts that highlight deep structure (base case, decreasing measure, and return-time composition) to support schema formation.

For teacher education, the framework's primary value is diagnostic. Many classroom difficulties can be reframed as layer-specific breakdowns, enabling targeted feedback (e.g., base-case sufficiency vs. reduction vs. composition). The explicit operationalization also supports the design of lesson plans and assessments that promote transfer rather than reliance on mechanical tracing.

Interpretation of initial validation evidence. The expert-based indices (content validity) and vignette-based rater calibration are intended to warrant ELSA+ as an assessment-ready instrument in a design sense: the progression is operationalized with observable indicators, canonical tasks, and rubric descriptors that can be applied consistently to student explanations. These results delimit current claims to content validity and scoring usability rather than classroom impact; field validation with authentic artifacts and learning outcomes remains necessary.

6.3. Theoretical implications

ELSA+ advances three theoretical contributions to computing education research. First, it operationalizes recursion understanding as layer-specific competencies rather than a binary notion of “understands recursion.” This granularity supports finer-grained hypotheses about non-linear learning trajectories and the design of targeted interventions.

Second, ELSA+ demonstrates an assessment-forward learning progression design in which operational indicators, canonical tasks, and rubric descriptors are developed in parallel with the conceptual model. This contrasts with post-hoc assessment construction and offers a transferable template for other persistently difficult CS topics (e.g., pointers, concurrency, and formal reasoning).

Third, by foregrounding explanation quality over trace correctness, ELSA+ contributes to broader debates on what constitutes evidence of understanding in programming education. The framework aligns with cognitive load and scaffolding theories while providing a concrete operationalization for assessment and instructional design.

6.4. Practical implications for teacher education

Practically, ELSA+ can be used as a diagnostic language in pre-service teacher education. In methods courses, instructors can align lesson objectives and formative checks with the five layers, enabling timely remediation (e.g., addressing base-case misconceptions before multi-branch composition tasks).

For example, when a pre-service teacher observes a student who can correctly trace recursive calls but cannot explain how a parent call composes sub-results (e.g., how `merge()` combines two sorted halves), ELSA+ supports diagnosing this as an L2→L3 boundary issue. The instructor can then scaffold return-time composition using recursion-tree representations and explicit “assume sub-solution” prompts, rather than assigning more tracing exercises.

ELSA+ also supports lesson planning and micro-teaching rehearsals: pre-service teachers can select layer-aligned tasks and anticipate typical failure modes (e.g., over-reliance on step-by-step tracing at the expense of return-time composition). The rubric provides structured feedback that emphasizes justification and structure, helping candidates learn to evaluate student explanations—not merely program outputs.

Finally, the framework can inform assessment design by promoting transfer-oriented prompts (e.g., applying composition reasoning from MergeSort to QuickSort) and by distinguishing conceptual errors (layer boundary issues) from implementation errors.

6.5. Methodological contributions

Methodologically, the study illustrates a pragmatic validation pathway for early-stage instrument development when full classroom deployment is not yet feasible: transparent evidence-base construction, expert content validation (CVI), and vignette-based calibration to check scoring usability and agreement. Together, these steps support bounded claims about readiness for diagnostic use while motivating the next phase of field validation.

Such a pathway is particularly useful in informatics education, where promising instructional constructs often appear before accompanying measurement tools. By packaging a progression, task family, and rubric

alongside initial validity and reliability evidence, ELSA+ aims to shorten that gap and make subsequent studies easier to replicate and compare.

6.6. Why Layer 5 belongs in a recursion learning progression

Layer 5 may appear to overlap with algorithm-design curricula; however, we position it as optimizing recursive structure rather than general complexity analysis. In teacher education and many secondary/introductory contexts, learners frequently encounter recursion tasks whose naive formulations exhibit repeated subproblems. The ability to recognize these repetitions and restructure a recursive solution via memoization or tabulation is a principled extension of recursion understanding: it requires preserving the recursive decomposition while reasoning about subproblem identity and reuse at return time.

Moreover, in assessment-driven settings where recursion tasks are common (e.g., competitive-programming traditions and advanced secondary examinations), teachers are expected to distinguish between “a correct recursive idea” and “a viable recursive solution under realistic constraints.” Layer 5 therefore operationalizes a pragmatic threshold of expertise that is pedagogically meaningful for teachers: it supports diagnosing when students have progressed beyond tracing and explanation toward designing robust recursive solutions.

We acknowledge that Layer 5 may not be required in every course. Accordingly, ELSA+ treats Layer 5 as optional/advanced for many syllabi, while retaining it for contexts where optimization is a curricular expectation. Concretely, we treat Layers 1–4 as the core recursion progression for introductory teaching and assessment, and Layer 5 as an advanced/extension layer that can be included or omitted depending on course goals.

7. Limitations and Future Work

7.1. Scope of current validation

The present study establishes initial content validity and scoring usability but does not evaluate classroom impact or learning outcomes. Expert-based indices indicate that the ELSA+ components are relevant and interpretable to Vietnamese informatics educators, and vignette-based calibration suggests that the rubric can be applied consistently to standardized responses. However, authentic student artifacts in teacher-education courses may introduce additional challenges (e.g., incomplete answers, mixed-layer features, and language-specific reasoning patterns).

The current evidence supports content validity and scoring usability, not pedagogical effectiveness. Accordingly, we treat “assessment-ready” as a design claim (measurement readiness) rather than an outcome claim (learning gains).

7.2. Methodological limitations

First, the expert panel size ($n = 8$) is appropriate for initial CVI estimation but remains modest, and the findings may reflect local curricular contexts. Second, the vignette set—while useful for calibration—does not fully replicate the variability of authentic coursework artifacts. Third, reliability estimates based on three raters may change with broader rater pools and different training protocols.

Generalizability note. The expert panel reflects Vietnamese informatics education, which may limit immediate generalizability. However, the targeted constructs (recursion mental models, decomposition, return-time composition, and mutual recursion) are broadly documented in the international literature. We therefore treat this work as an initial instantiation and invite cross-context replications to examine transportability and potential adaptation needs.

The reported validation indices represent initial, design-oriented findings that informed rubric refinement and scoring guidance; comprehensive validation with authentic classroom artifacts remains a priority for future work.

7.3. Framework scope limitations

ELSA+ emphasizes conceptual structure and explanation quality rather than debugging behaviors or full implementation proficiency. Although the framework is intended to be language-agnostic, its transfer across paradigms (imperative vs. functional) and across languages of instruction warrants direct examination. In addition, Layer 5 (“Optimization”) remains the most boundary-sensitive layer and may require further refinement as field evidence accumulates.

7.4. Future work

Future work will prioritize (i) deployment in teacher-education courses using scored authentic artifacts, (ii) longitudinal tracking of pre-service teachers’ progression across layers, and (iii) transfer-task evaluation to assess whether layer-aligned instruction supports generalization to novel recursive patterns. A cross-context validation program (across institutions and cultures) is also necessary to strengthen generalizability.

Although this contribution is design-oriented, the intended use of ELSA+ is diagnostic assessment in teacher education. The next phase should therefore validate both layer assignment and rubric scoring on authentic student artifacts across course contexts, programming languages, and learner populations, and develop a clearer validity argument (e.g., construct and predictive validity in relation to transfer performance and teaching decisions). In addition, longitudinal studies can examine whether learners progress linearly across layers or exhibit common non-linear profiles that call for alternative instructional sequences.

8. Conclusion

ELSA+ offers a structured way to talk about recursion understanding in pre-service informatics teacher education: five cognitive layers, each tied to observable evidence, canonical tasks, and a 0–3 rubric that foregrounds explanation quality. Used as a shared diagnostic language, it can help instructors and pre-service teachers move beyond tracing toward structural reasoning and transfer. While stronger field validation is still needed, the framework and accompanying instruments are designed to be used immediately for formative diagnosis and to support more rigorous efficacy studies.

References

- Baron, A., & Feitelson, D. G. (2024). Why Is Recursion Hard to Comprehend? An experiment with experienced programmers in Python. In Proceedings of the 2024 Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2024). <https://doi.org/10.1145/3649217.3653636>
- Götschi, T., Sanders, I., & Galpin, V. (2003). Mental models of recursion. In Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '03) (pp. 346–350). Association for Computing Machinery. <https://doi.org/10.1145/611892.612004>
- Haberman, B., & Averbuch, H. (2002). The case of base cases: Why are they so difficult to recognize? Student difficulties with recursion. In Proceedings of the 7th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE '02) (pp. 84–88). Association for Computing Machinery. <https://doi.org/10.1145/544414.544441>
- Hamouda, S., Edwards, S. H., Elmongui, H. G., Ernst, J. V., & Shaffer, C. A. (2019). RecurTutor: An interactive tutorial for learning recursion. *ACM Transactions on Computing Education*, 19(1), Article 1. <https://doi.org/10.1145/3218328>

- Hamouda, S., Edwards, S. H., Elmongui, H. G., Ernst, J. V., & Shaffer, C. A. (2020). BTRecurTutor: A tutorial for practicing recursion in binary trees. *Computer Science Education*, 30(2), 216–248. <https://doi.org/10.1080/08993408.2020.1714533>
- Heinonen, A., Systä, K., & Hellas, A. (2023). Synthesizing research on programmers' mental models of programs, tasks and concepts: A systematic literature review. *Information and Software Technology*, 163, 107300. <https://doi.org/10.1016/j.infsof.2023.107300>
- Hundhausen, C. D., Douglas, S. A., & Stasko, J. T. (2002). A meta-study of algorithm visualization effectiveness. *Journal of Visual Languages & Computing*, 13(3), 259–290. <https://doi.org/10.1006/jvlc.2002.0237>
- McCauley, R., Grissom, S., Fitzgerald, S. C., & Murphy, L. (2015). Teaching and learning recursive programming: A review of the research literature. *Computer Science Education*, 25(1), 37–66. <https://doi.org/10.1080/08993408.2015.1033205>
- Paas, F., Renkl, A., & Sweller, J. (2003). Cognitive load theory and instructional design: Recent developments. *Educational Psychologist*, 38(1), 1–4.
- Rinderknecht, C. (2014). A survey on teaching and learning recursive programming. *Informatics in Education*, 13(1), 87–119.
- Sorva, J., Karavirta, V., & Malmi, L. (2013). A review of generic program visualization systems for introductory programming education. *ACM Transactions on Computing Education*, 13(4), Article 15, 1–64. <https://doi.org/10.1145/2490822>
- Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(2), 257–285.
- Tung, S.-H., Chang, C.-T., Wong, W.-K., & Jehng, J.-C. (2001). A study of learning recursion in a computer programming course. *International Journal of Human-Computer Studies*, 54(3), 285–300. <https://doi.org/10.1006/ijhc.2000.0433>
- Velázquez-Iturbide, J. Á., & Pérez-Carrasco, A. (2010). InfoVis interaction techniques in animation of recursive programs, 3(1), 76–91. <https://doi.org/10.3390/a3010076>
- Velázquez-Iturbide, J. Á., Pérez-Carrasco, A., & Urquiza-Fuentes, J. (2009). Program visualization and algorithm animation in computing education. In *Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE '09)*. Association for Computing Machinery.
- Wood, D., Bruner, J. S., & Ross, G. (1976). The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry*, 17, 89–100.

Author

Vuong T. PHAM, Saigon University, Ho Chi Minh City, Vietnam. E-mail: vuong.pham@sgu.edu.vn. Corresponding author. ORCID: 0000-0002-3879-9677

Acknowledgement

The author thanks the participating secondary-school teachers and university lecturers for their expert feedback during the validation study.

Appendix A. ELSA+ Rubric (0–3)

Scoring convention: 0 = Not demonstrated; 1 = Emerging (trace-dependent or fragmented); 2 = Competent (mostly correct with minor gaps); 3 = Proficient (robust, transferable, well-explained).

Layer 1 — Base case reasoning

- 0: Cannot identify a valid base case; stopping condition does not guarantee termination/correctness.
- 1: Identifies base case for familiar example but misses boundary inputs or provides weak justification.
- 2: States correct base case(s) and explains sufficiency for the target domain (most inputs).
- 3: Handles edge cases systematically; predicts failure modes when base cases are wrong. (e.g., predicting non-termination when the base case is unreachable).

Evidence artifacts: short written justification; boundary-input checklist; annotated code.

Layer 2 — Self-referential decomposition

- 0: Cannot express a reduction step; treats recursion as repetition.
- 1: Expresses reduction inconsistently; depends on tracing; decreasing measure unclear.
- 2: Specifies a correct reduction and articulates the “assume sub-solution” move for linear recursion.
- 3: Explains reduction and invariant cleanly; designs new recursive definitions for structurally similar problems. (e.g., designing a recursive definition for structural list processing beyond the worked example).

Evidence artifacts: reduction statement; invariant annotation; brief explanation without tracing.

Layer 3 — Multi-branch composition

- 0: Cannot combine results from branches; confuses call order and return values.
- 1: Can trace multi-branch calls but struggles to describe how results are combined on return.
- 2: Correctly describes/implements combination logic and explains composition at internal nodes.
- 3: Uses recursion-tree reasoning to justify correctness and transfer to new divide-and-conquer patterns. (e.g., applying QuickSort-style partitioning after learning MergeSort-style divide-and-conquer).

Evidence artifacts: annotated recursion tree; return-flow explanation; transfer task solution.

Layer 4 — Mutual recursion

- 0: Cannot explain interdependence; treats functions independently.
- 1: Follows an example with heavy tracing but cannot generalize the alternation rule.
- 2: Explains mutual dependency as state transition/alternation and implements with correct base cases.
- 3: Generalizes patterns; refactors between mutual recursion and equivalent state-based forms. (e.g., rewriting a mutual-recursion recognizer into an explicit state-machine loop).

Evidence artifacts: state diagram/grammar sketch; alternation rule statement; refactoring explanation.

Layer 5 — Optimization

- 0: Does not recognize redundant subcalls; cannot explain performance degradation.
- 1: Recognizes repetition after seeing traces/trees but cannot propose systematic optimization.
- 2: Identifies overlapping subproblems and applies memoization/tabulation correctly for standard cases.
- 3: Justifies optimization choices; derives DP formulation and explains trade-offs for novel variants. (e.g., deriving a DP table for Fibonacci variants or edit-distance-like recurrences).

Evidence artifacts: marked overlap in recursion tree; memoization key/table; dependency order reasoning; transfer variant.

Appendix B. Standardized vignette set for rater calibration (25 items)

Appendix B (Vignette Set) and Appendix C (Expert Validation Protocol and Revision Log) are provided as supplementary files for this submission.

This appendix summarizes the standardized vignette set used for rubric calibration. Each vignette consists of a prompt and a short student response designed to elicit evidence for one target layer and level on the 0-3 rubric. The full rater-ready pack and scoring sheet can be provided as supplementary material.

Supplementary materials: Vignette pack (DOCX) and scoring sheet (XLSX).

Appendix C. Expert validation protocol and revision log (summary)

This appendix summarizes the expert-validation protocol and the main rubric/framework refinements made after expert feedback. A full protocol, anonymized rating workbook, and detailed revision log are provided as supplementary files.