Understanding Refute Problem Difficulty for Learners in Indian Computing Education

Viraj Kumar
viraj.kumar.cs@gmail.com
University of New South Wales
Bengaluru, Karnataka, India

S. Sangeetha
sangeetha@psgitech.ac.in
PSG Institute of Technology and Applied Research
Coimbatore, Tamil Nadu, India

Nimisha Agarwal
nimishaagarwal@iitram.ac.in
Institute of Infrastructure, Technology, Research and Management
Ahmedabad, Gujarat, India

Abstract

Refute Problems present students with buggy code for a given task and ask students to demonstrate why the code is buggy. Little is known about what makes a Refute Problem difficult for students. This Working Group (WG) investigates possible causes of difficulty such as (1) comprehending the given implementation (code-level difficulty) and (2) comprehending the given problem specification (task-level difficulty). We hypothesize that task-level difficulty is particularly relevant in the context of Indian computing education, where several students may have limited fluency in the language in which the task is expressed (e.g., English). This WG will limit the scope to Refute Problems in the context of Introductory Programming (CS1), where they were first proposed and have been most extensively studied. Drawing on cognitive load theory, code comprehension, and assessment design, as well as an analysis of existing Refute Problems, instructor reflections, and exploratory studies with undergraduate learners from participating Indian institutions, this WG aims to develop a framework for characterizing Refute Problem difficulty together with design guidelines, annotated examples, and tools to help instructors tailor the difficulty of Refute Problems for their students. This will support the creation of more equitable, scalable, and cognitively calibrated Refute Problems for assessments.

CCS Concepts

• **Social and professional topics → Student assessment**.

Keywords

Refute Problems, Cognitive Difficulty, Linguistic Difficulty, Assessment

ACM Reference Format:
Viraj Kumar, S. Sangeetha, and Nimisha Agarwal. 2026. Understanding Refute Problem Difficulty for Learners in Indian Computing Education. In *Proceedings of ACM COMPUTE (COMPUTE'26).* ACM, New York, NY, USA, 2 pages. https://doi.org/XXXXXXX.XXXXXXX

COMPUTE'26, Coimbatore, TN, India

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

1 Background and Related Work

Prior work has demonstrated that Refute Problems can assess the ability of students to comprehend both programming tasks and code [1, 4, 7]. However, little is known about the factors that influence learner success or failure on such tasks. This Working Group (WG) seeks to address this gap by examining the difficulty of Refute Problems along two key dimensions: comprehending the given code (*code-level* difficulty) and comprehending the given problem statement (*task-level* difficulty). The latter dimension is of particular interest. The ability of students to comprehend problem statements varies substantially [8], and this variance may be exacerbated in the Indian context due to differences in students' comprehension of the language of instruction [5]. The scope of this WG will be limited to CS1 (Introductory Programming, for which Refute Problems have been most frequently studied) using popular programming languages (e.g., Python and C).

2 Goals

This WG aims to develop a research-informed, empirically validated framework for understanding the sources of difficulty that Indian undergraduate students face in solving Refute Problems. This objective translates into two concrete goals:

1. To develop a multidimensional framework for Refute Problem difficulty. Drawing on cognitive load theory [2, 9], program comprehension literature [3], and assessment design research [6], the WG will articulate a framework to explain Refute Problem difficulty. Since Refute Problems require learners to interpret natural-language specifications, the WG will study how wording complexity, abstraction levels, semantic precision, and language accessibility influence task comprehension. This is especially relevant in Indian classrooms where many learners engage with computing education through a second or third language. For the code-level dimensions of difficulty, particular attention will be paid to the complexity of generating candidate counterexample inputs and the number and nature of steps required to trace code execution on these inputs.

2. To generate empirically grounded design guidelines and tools for educators. The WG seeks to support instructors in designing Refute Problems that are pedagogically effective, equitable, and appropriately calibrated for different learner populations. Based on the above framework, the WG will propose (1) *guidelines* to help educators tailor Refute Problem difficulty to their specific learning contexts, and (2) *tools* that help instructors calibrate Refute Problem difficulty.

3 Proposed Methodology

3.1 Phase 1: Jul. 3 – Dec. 2

The primary objective of Phase 1 is to establish the theoretical and empirical foundation necessary to achieve the key target of Phase 2: presenting a preliminary multidimensional framework for Refute Problem difficulty at COMPUTE 2026.

3.1.1 Literature Review, Conceptual Grounding, and Instructor Experience. WG participants will collaboratively conduct a focused review of literature. The review will be used to identify candidate dimensions of Refute Problem difficulty at both the *code level* and the *task level*. WG members will also discuss how they currently design and evaluate Refute Problems, what they perceive as difficult for learners (especially linguistic factors), and strategies they adopt for scaffolding code-level and task-level comprehension.

3.1.2 Developing a Preliminary Refute Problem Corpus. WG members will collaboratively curate a corpus of Refute Problems that systematically vary along hypothesized dimensions of difficulty. The corpus will include problems with varying code complexity, variations in wording and linguistic structure, tasks requiring different levels of abstraction and reasoning, and examples tailored to diverse introductory programming contexts commonly found in Indian undergraduate programs. Parallel versions of problems will be developed to isolate specific dimensions of difficulty (e.g., equivalent natural-language specifications differing in abstraction levels, functionally equivalent code differing in cyclomatic complexity)[1].

3.1.3 Data Collection. A key deliverable of Phase 1 is the completion of empirical data collection involving undergraduate students solving Refute Problems. WG members will conduct pilot and iteratively refined exploratory studies with undergraduate students from participating Indian institutions. These studies will be based on carefully curated Refute Problems, and conducted in classrooms, labs, or online settings. Data collection will include student responses to Refute Problems, correctness and completion rates, time-on-task measures, reflective explanations where feasible, short surveys on perceived difficulty and comprehension, and relevant demographic and language-background information.

3.1.4 Collaborative Analysis and Framework Drafting. Prior to the conference, WG participants will meet regularly in virtual sessions to synthesize findings from the literature review and empirical data. Through iterative discussion and analysis, the WG will identify recurring patterns of success and failure and to refine the hypothesized dimensions of difficulty. This will culminate in a preliminary multidimensional framework describing Refute Problem difficulty, candidate relationships between code-level and task-level factors, and initial hypotheses regarding the role of linguistic accessibility in learner performance.

3.2 Phase 2: Dec. 3 – 5

The draft framework developed in Phase 1 will serve as the central artifact for discussion at COMPUTE 2026. The key goal of this phase is to obtain critical feedback on the draft framework from conference participants.

[1]Example: https://onlinegdb.com/pbt9JZtxs

3.3 Phase 3: Dec.'26 – Mar.'27

Following the conference, the WG will refine and consolidate the framework based on conference feedback and continued analysis of collected data. In particular, the WG will clarify definitions and relationships among dimensions, incorporate additional evidence and examples, address limitations identified during conference discussions, and improve the framework's applicability across instructional contexts missed in earlier phases.

3.3.1 Instructor Guidelines and Tool Development. Building on the refined framework, the WG will develop practical artifacts for educators, including guidelines for designing and calibrating Refute Problems, recommendations for adapting tasks to linguistically diverse classrooms, illustrative examples of difficulty variations, and prototype tools and rubrics to support instructors in estimating Refute Problem difficulty.

3.3.2 Dissemination and Publication. The WG will collaboratively prepare a journal manuscript describing the framework, methodology, and preliminary findings, and develop instructor-oriented resources and tools.

4 Expected Deliverables

The Working Group expects to produce (1) A framework for characterizing Refute Problem difficulty. (2) An annotated corpus of Refute Problems. (3) Design guidelines and tools for educators. (4) Scholarly dissemination of outputs.

5 Human Subjects

IRB approval will be sought for conducting studies involving students in Phase 1.

References

[1] Nimisha Agarwal, Viraj Kumar, Arun Raman, and Amey Karkare. 2023. A Bug's New Life: Creating Refute Questions from Filtered CS1 Student Code Snapshots. In *Proceedings of the ACM Conference on Global Computing Education Vol 1*. 7–14.

[2] Rodrigo Duran, Albina Zavgorodniaia, and Juha Sorva. 2022. Cognitive load theory in computing education research: A review. *ACM Transactions on Computing Education (TOCE)* 22, 4 (2022), 1–27.

[3] Cruz Izu, Carsten Schulte, Ashish Aggarwal, Quintin Cutts, Rodrigo Duran, Mirela Gutica, Birte Heinemann, Eileen Kraemer, Violetta Lonati, Claudio Mirolo, et al. 2019. Program comprehension: Identifying learning trajectories for novice programmers. In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*. 261–262.

[4] Viraj Kumar. 2021. Refute: an alternative to 'explain in plain English'questions. In *Proceedings of the 17th ACM Conference on International Computing Education Research*. 438–440.

[5] Utsav Kumar Nareti, Divyansh Gupta, Chandranath Adak, Soumi Chattopadhyay, Emma Riese, Tanujit Chakraborty, Mayank Agarwal, and Satendra Kumar. 2025. Assessing Engineering Student Perceptions of Introductory CS Courses in an Indian Context. *arXiv preprint arXiv:2508.06563* (2025).

[6] Rajendra Raj, Mihaela Sabin, John Impagliazzo, David Bowers, Mats Daniels, Felienne Hermans, Natalie Kiesler, Amruth N Kumar, Bonnie MacKellar, Renée McCauley, et al. 2021. Professional competencies in computing education: Pedagogies and assessment. In *Proceedings of the 2021 Working Group Reports on Innovation and Technology in Computer Science Education*. 133–161.

[7] Arun Raman and Viraj Kumar. 2022. Programming pedagogy and assessment in the era of AI/ML: A position paper. In *Proceedings of the 15th Annual ACM India Compute Conference*. 29–34.

[8] Leonardo Silva, António José Mendes, Anabela Gomes, and Gabriel Fortes. 2022. Investigating programming students problem comprehension ability and its association with learning performance. *IEEE Transactions on Education* 66, 2 (2022), 156–162.

[9] John Sweller. 2016. Cognitive load theory and computer science education. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. 1–1.