The paper titled "Extracting Features for Computational Thinking from Block-Based Code" by Param Damle and Glen Bull from the University of Virginia presents an innovative approach to support undertrained instructors in introductory computer science classes[1]. They propose an automated evaluator (autograder) for Snap!, a block-based programming language that is beginner-friendly[1]. The novelty of their approach lies in its ability to assess the computational thinking (CT) reflected in the structure of a student's submission rather than the accuracy or run-time of its execution[1].

Their method assesses demonstrated knowledge of abstraction and iteration from an XML tree representation of a student's Snap! program[1]. While existing methods involve clustering trees with similar structures together, these methods were found to be too generalized and inadequate at reflecting specific CT elements[1]. Therefore, the authors explore how to tailor their feature extraction to capture such elements, including consecutive repetition and encapsulation of functional blocks[1].

Unlike proprietary autograders, their approach integrates the academic community into the research and development of the optimal feature embedding of Snap! programs[1]. The paper presents both successful and unsuccessful endeavors to inform replications of this work[1]. It also highlights avenues for feature tuning and scalability of the autograding model to larger, more diverse classrooms[1].

Source: Conversation with Copilot, 5/24/2024

(1) Extracting Features for Computational Thinking from Block-Based Code. https://www.kdd.org/kdd2023/wp-content/uploads/2023/08/damle2023extracting.pdf.

(2) KDD '23: Proceedings of the 29th ACM SIGKDD Conference on Knowledge .... https://www.kdd.org/kdd2023/wp-content/uploads/2023/08/toc.html.

(3) Sensor Placement for Learning on Networks - kdd.org. https://www.kdd.org/kdd2023/wp-content/uploads/2023/08/burudgunte2023sensor.pdf.

(4) undefined. https://doi.org/X.