education sciences

MDPI

Article

Detecting Credit-Seeking Behavior with Programmed Instruction Framesets in a Formal Languages Course

Yusuf Elnady [1,*], Mohammed Farghally [1,*], Mostafa Mohammed [2] and Clifford A. Shaffer [1,*]

[1] Department of Computer Science, Virginia Tech, Blacksburg, VA 24061, USA
[2] Department of Computer Science and Engineering, University at Buffalo, Buffalo, NY 14260, USA; mkomohmd@buffalo.edu
* Correspondence: yelnady@vt.edu (Y.E.); mfseddik@vt.edu (M.F.); shaffer@vt.edu (C.A.S.)

Abstract: When students use an online eTextbook with content and interactive graded exercises, they often display aspects of two types of behavior: credit-seeking and knowledge-seeking. A student might behave to some degree in either or both ways with given content. In this work, we attempt to detect the degree to which either behavior takes place and investigate relationships with student performance. Our testbed is an eTextbook for teaching Formal Languages, an advanced Computer Science course. This eTextbook uses Programmed Instruction framesets (slideshows with frequent questions interspersed to keep students engaged) to deliver a significant portion of the material. We analyze session interactions to detect credit-seeking incidents in two ways. We start with an unsupervised machine learning model that clusters behavior in work sessions based on sequences of user interactions. Then, we perform a fine-grained analysis where we consider the type of each question presented within the frameset (these can be multi-choice, single-choice, or T/F questions). Our study involves 219 students, 224 framesets, and 15,521 work sessions across three semesters. We find that credit-seeking behavior is correlated with lower learning outcomes for students. We also find that the type of question is a key factor in whether students use credit-seeking behavior. The implications of our research suggest that educational software should be designed to minimize opportunities for credit-seeking behavior and promote genuine engagement with the material.

Keywords: programmed instruction; student engagement; automated assessment; eTextbook; credit-seeking behavior; formal languages

check for updates

Academic Editors: Daniel Muijs and Kendall Hartley

Received: 12 February 2025
Revised: 9 March 2025
Accepted: 27 March 2025
Published: 31 March 2025

Citation: Elnady, Y., Farghally, M., Mohammed, M., & Shaffer, C. A. (2025). Detecting Credit-Seeking Behavior with Programmed Instruction Framesets in a Formal Languages Course. *Education Sciences*, *15*(4), 439. https://doi.org/10.3390/educsci15040439

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

ETextbooks are becoming increasingly popular. Good ones incentivize students to interact with them by including visualizations and exercises Fouh et al. (2012). OpenDSA Fouh et al. (2014) is a widely used example of such a system. It provides content for multiple Computer Science courses, such as Data Structures and Algorithms, Programming Languages, and Formal Languages and Automata (FLA). OpenDSA includes support for the Programmed Instruction (PI) approach. PI is based on learning by repeatedly reading an amount of material then answering a question related to that material before moving on to the next step Lockee et al. (2004). The intent is to engage students continuously with the material. Over the years, much evidence that PI offers a better learning experience for students than traditional approaches has been presented Mohammed (2021); Molenda (2008); Sambasivarao (2020); Wangila et al. (2015).

It is well known that many students will attempt to misuse eTextbooks and similar educational software; this behavior is referred to as "gaming the system" R. Baker et al.

(2004); R. S. Baker et al. (2004); Haig et al. (2009); Koh et al. (2018). Such gaming behavior often hurts students' learning, but sometimes, the system can be redesigned to discourage damaging behavior. We find that some students attempt to "game" PI in different ways. For example, a student might quickly click on all possible solutions in a multiple-choice question of a PI to guess the answer so as to obtain credit without a sufficient understanding of the associated information. We call this "credit-seeking" behavior, as students are obviously "gaming" the system to obtain the corresponding credit and are not actively engaging with the material.

This paper explores the amount of credit-seeking behavior that occurred when an eTextbook with PI was used in a Formal Languages course. The underlying OpenDSA framework recorded all the interactions students made into logs. Leveraging the collected logs, we developed a robust model to discern credit-seeking behavior within student interactions and distinguish it from behavior indicative of genuine, active learning. Our analysis revealed a notable contrast in the overall course performance, specifically in terms of scores, between students exhibiting a "credit-seeking" approach and those utilizing the material for genuine educational engagement. Our research hypothesis is that credit-seeking behavior detrimentally affects students' learning performance, constraining their capacity to achieve high scores in examinations and contributing to lower overall class scores.

Our study makes the following contributions:

- Unlike previous studies that broadly categorize credit-seeking behavior, our research provides a fine-grained analysis of how different types of questions (multi-choice, single-choice, and true/false) influence credit-seeking behavior. We find that multi-choice questions are particularly prone to triggering credit-seeking behavior, which is a significant insight for the design of educational software.
- Our study provides empirical evidence of the negative correlation between credit-seeking behavior and learning outcomes. We showed that students who engage in credit-seeking behavior achieve lower exam scores and overall academic performance compared to their peers. This finding underscores the importance of addressing credit-seeking behavior to improve educational outcomes.
- We used an unsupervised learning technique to cluster work sessions based on student interactions. This approach allowed us to identify patterns of credit-seeking behavior without relying on pre-labeled data, making our model more adaptable to different educational contexts.

2. Background

Formal Languages and Automata (FLA) is an advanced theory course, typically taken at the junior/senior level, and is a standard part of the Computer Science curriculum Computing Curricula and Society (2013). FLA requires students to understand theoretical concepts and proof techniques and apply algorithms to build different finite state machines and formal models capable of representing specific languages.

PI is a systematic approach to presenting material through a graduated series of controlled steps with corresponding activities Skinner (1986). Our implementation is a slideshow that consists of a set of frames, where each frame contains some small amount of information (a sentence or short paragraph) and is often accompanied by a simple question that the student must answer correctly to be able to move to the next frame. These questions assess the student's comprehension in each step. A typical PI frame is shown in Figure 1. Here, a sentence is given, linked to a simple question with an inactive *next* (>) button because the question is not yet solved.

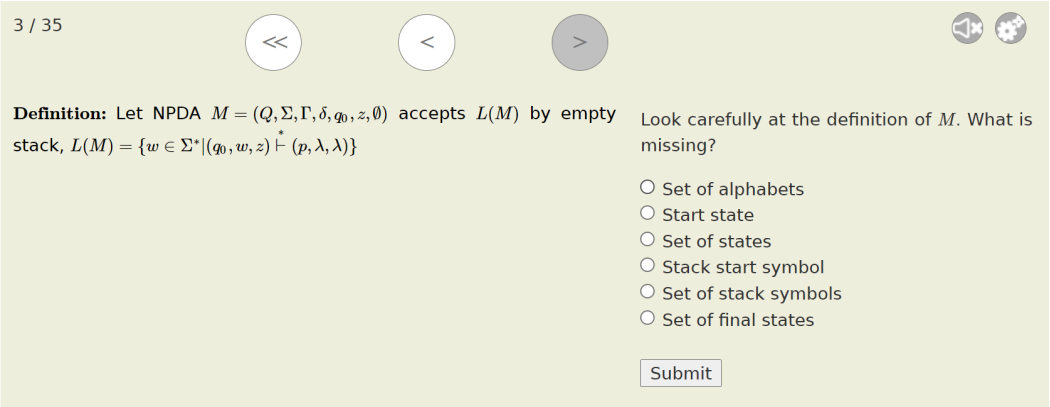


Figure 1. A PI frame consisting of a short sentence associated with a question.

PI typically allows students an unlimited number of attempts to repeat the question until they find the correct answer. Answering the question is not meant to be difficult; it is most critically meant to keep the student engaged and confirm their understanding as they progress through the material. Questions supported by OpenDSA in PI frames consist of single-choice (students can select only one choice from a list of choices, a classic multiple-choice question); multi-choice (students can select one or more choices from a list of choices), or true/false. Some frames do not contain a question but just information.

In multiple-choice questions, solutions grow exponentially. For instance, in Figure 2, with eight choices, potential solutions reach $2^8 - 1 = 255$.

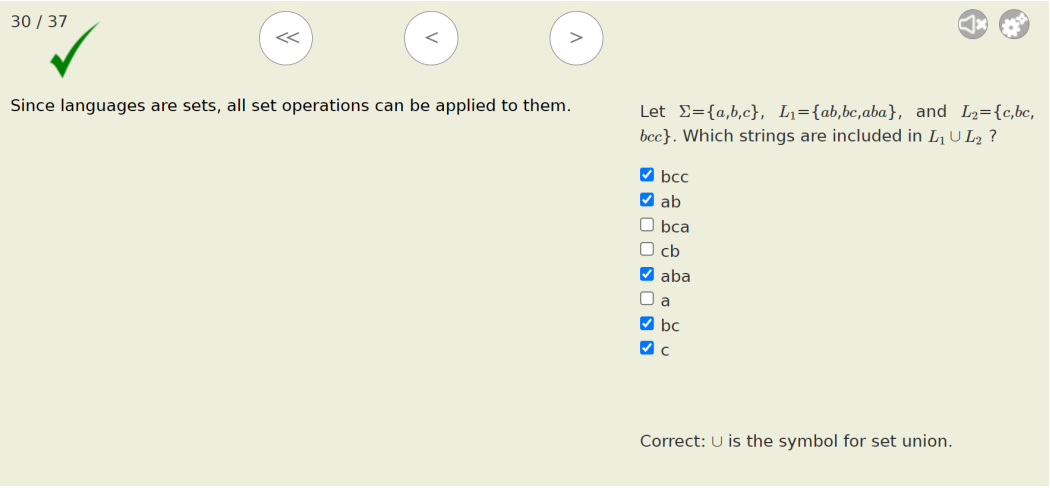


Figure 2. A PI frame with a multi-choice question.

3. Literature Review

The phenomenon of off-task behavior when using educational software was first identified in intelligent tutoring systems, referred to there as "hint abuse" Tait et al. (1973). The term "gaming the system" in this context was first introduced in Cheng and Vassileva (2006), defined as behavior aimed at obtaining correct answers by misusing the software's help system. Usually, the primary incentive for gaming the system is to complete the work and obtain the credit as easily as possible. Often, this is performed to save time, but there are many instances of gaming behavior that saves time but lowers cognitive load.

Prior studies show that credit-seeking behavior in educational software often hurts students' learning R. Baker et al. (2004); R. S. Baker et al. (2004); Haig et al. (2009); Koh et al. (2018). Students might engage in this behavior by misusing the system's feedback and hints to obtain the correct answer Aleven et al. (2006); Peters et al. (2018), systematically

using the tutor's feedback to obtain the answer R. S. Baker et al. (2004), or blindly guessing all possible solutions until hitting the correct answer R. S. Baker et al. (2004). Students who game the system often demonstrate low pre-test understanding and low overall academic achievement and perform significantly lower on post-tests than students who never game the system.

In a follow-up study R. Baker et al. (2004), Baker et al. divided students' behavior in intelligent teaching systems into students who never gamed the system, students who gamed the system but still scored high, and students who gamed the system and scored low. Their model could detect system-gaming behavior that led to impaired learning. The model was generalized to other students it had not seen before who were using the same online teaching system. They then implemented a feature that provided interventions when system gaming was detected. This increased students' learning performance. Other researchers have augmented educational software to detect gaming behavior R. Baker et al. (2010); Beck (2005); Johns and Woolf (2006); Walonoski and Heffernan (2006). Others have developed approaches to prevent and discourage students from this behavior Murray and Vanlehn (2005); Walonoski and Heffernan (2006).

In their 2014 study, Fouh et al. scrutinized student interactions with an eTextbook system Fouh et al. (2014). The authors introduced the term "credit-seeking" to characterize negative behavior, distinct from the behavior associated with "learning". They found that students can perform "credit-seeking" in four different ways: (1) Not reading text associated with exercises and instead jumping directly to the exercises. Only if necessary do they read as much as is required to solve the exercises. (2) "Rushing" behavior where they navigate through slides in visualizations as quickly as they can to obtain the associated credit, without the intention of understanding or reading the material in the slides. (3) Skipping directly to the end of slideshows by clicking on the "fast-forward" button. (4) Using algorithm visualizations that are part of the eTextbook to obtain the output needed to solve proficiency exercises, rather than solving the exercise themselves.

Recent studies continue to explore the impact of different student study behaviors in educational software and its implications on their learning outcomes. Ma et al. (2024) applied clustering analysis to identify reader profiles that differed in performance and progression in an educational literacy app. Another study by Munshi et al. (2023) examined the impact of adaptive scaffolding on support for student self-regulated learning behaviors for middle-school science. The study highlighted the importance of combining context-sensitive inflection points with tailored feedback to support self-regulated learning behaviors and narrow the gap in learning outcomes between high- and low-performing students. A study by Rocha et al. (2024) addressed the challenge of detecting gaming attitudes in novice programming learners, which previous detectors had not fully captured. The study combined knowledge engineering and machine learning algorithms to develop a model for detecting gaming behavior.

4. Materials and Methods

In this section, we discuss how we collected and processed log data relating to students' interactions with the FLA eTextbook, with most content presented using PI Mohammed et al. (2021). We collected data from a senior-level FLA course at our university with sections taught in Fall 2020, Spring 2021, and Spring 2022. All three offerings used the same eTextbook with minor modifications being made on an ongoing basis to improve the PI framesets. In Fall 2020, PI framesets were not part of a student's grade, but in Spring 2021, students who completed all PI framesets earned an additional 5% bonus for the semester. In Spring 2022, students were required to complete the framesets for a total of 20% of the semester grade.

4.1. Interaction Logs Dataset

Fouh et al. (2014) logs correct and incorrect attempts to solve a question within a PI frameset. Each row of the log represents a single interaction, and each column represents a feature of that interaction: *student_id* (a system-generated ID to maintain privacy and anonymity), *frame_name*, *question_number*, *is_correct*, and a timestamp. We processed the logs to assign events to work sessions. A work session was defined as the period during which a student was working on a specific frameset. If the student finished the frameset, jumped to another, or had more than two minutes without interaction, then we considered the work session over. Students took a mean of 15.3 seconds to make their first attempt to answer a given frameset question. Students tended to finish a frameset in one session.

To determine the appropriate duration for when a lack of interactions indicated the end of a session, we analyzed all time gaps in the dataset that happened between any two consecutive interactions that belonged to the same student and the same frameset. A total of 4.77% of the time gaps in the dataset were one minute or longer, while 2.65% were two minutes or longer. Beyond that, the percentage of time gaps longer than a given amount declined slowly. We explored the effect of different thresholds on the results of our analysis and found that any gap beyond two minutes yielded the same results, that is, the results were not sensitive to the exact value of the constant. We chose two minutes as the most reasonable limit to how long a person would take to attempt to answer a question since all frames consisted of one or two sentences with a simple question attached to them. A longer gap, therefore, was considered another session (such as if the user came back from a work break). But, picking any time gap greater than two minutes would not affect the analysis.

In Fall 2020, we had 75 students and 91 framesets with a total of 70,767 interactions among 3895 work sessions. In Spring 2021, we had 70 students and 87 framesets with a total of 126,216 interactions among 5841 work sessions. In Spring 2022, We had 74 students and 46 framesets attempted with a total of 174,721 interactions among 9785 work sessions.

4.2. Dataset Preprocessing

We preprocessed the raw interaction logs into work sessions following these steps:

1. We removed any non-student interactions from the dataset, such as interactions by teaching staff and those automatically generated by the system.
2. We standardized frameset names over the course of the three semesters, as some names were changed as the content was steadily improved.
3. We excluded interactions that occurred after the end of the semester, because they had no association with the concept of credit-seeking as there was no longer any reward.
4. We transformed the initial interactions into work sessions, which consisted of a series of chronologically sequential activities. We identified five types of activities that could appear in a session. Sessions started with *SESSION_START* and ended with *SESSION_END*. A correct answer was labeled *CRRCT*, and an incorrect answer was labeled *X*. The activity *BACK* occurred when the student moved backward in the same frameset and attempted a question they had previously solved. If there was a gap of more than two minutes between any two consecutive interactions, we ended the current session and started a new one that started with the next interaction.
5. We removed sessions with fewer than three interactions, since sessions with only one or two interactions were not useful for studying the differences between credit-seeking and active learning patterns.
6. We aggregated the session activities into a single data point that represented the session, by creating attributes for each session. These attributes represented the final features that we relied on to conduct the clustering process for the coarse-grained analysis. These are explained next.

4.3. Work Session Attributes

Our goal in this stage was to create a model that determined whether a given work session included credit-seeking behavior or not. Since we did not have ground truth, we used unsupervised learning techniques to cluster the work sessions. We chose not to perform the clustering at the student level because a student could have some credit-seeking sessions and some active learning sessions. In a later analysis, we looked at the percentage of credit-seeking or active learning sessions to decide how to classify the student.

We performed session-level clustering to classify the sessions as credit-seeking or active learning based on five attributes. First was the percentage of incorrect attempts that were preceded by an incorrect attempt (that is, the number of interactions that had consecutive wrong attempts) divided by the number of interactions in that session. The idea was that if a student consistently gave incorrect answers, they were more likely to be guessing the answer without paying attention to the material in the PI frame. The 2nd attribute was the percentage of consecutive correct attempts. If the session contained several consecutive correct attempts, it could be an indication of the student's interest in reading the PI frame and then answering the question, which allowed them to be able to answer most questions correctly.

The 3rd and 4th attributes were the percentage of incorrect and correct attempts. Some sessions may have had neither consecutive incorrect attempts nor consecutive correct attempts, leading to no indication of the behavior of these sessions. For example, if a session's interactions were SESSION_START -> Crrct -> X -> Crrct -> X -> Crrct -> SESSION_END, then the attributes of the percentage of consecutive incorrect attempts and the percentage of consecutive correct attempts were not useful. So, we included the percentage of incorrect and correct attempts as well. The 5th attribute was the median time taken between interactions for the session. The rationale was that if a student was randomly guessing the answer and trying to finish the PI frameset as fast as possible, then the time between any two consecutive interactions should have been smaller than when the student was trying to answer questions correctly.

Table 1 shows an example of four sessions with their generated five attributes along with the student ID, session ID, and frameset name columns. For example, Student 5112 in Session 2185 attempted the PDAFS frameset and had 66.7% of their interactions classified as consecutive incorrect interactions and 20% of their interactions as consecutive correct interactions. Student 5112 also had eleven incorrect attempts in total and four correct attempts, and the median time between events was 3.5 s.

Table 1. An example of work session attributes.

Student ID	Session ID	Frameset Name	Percentage of Consecutive Incorrect Attempts	Percentage of Consecutive Correct Attempts	Percentage of Incorrect Attempts	Percentage of Correct Attempts	Median Time Between (s)
812	4	GrammarIntroFS	0.053	0.579	0.211	0.790	19.5
5112	2185	PDAFS	0.667	0.200	0.733	0.267	3.5
6557	3323	ClosureConceptFS	0.000	0.538	0.231	0.770	7.0
6660	4603	RemoveUselessFS	0.541	0.054	0.757	0.243	2.0

5. Results

5.1. Coarse-Grained Analysis

Clustering algorithms struggle with high-dimensional data, even when there are only five features, because many pairs of points have similar distances, resulting in non-meaningful clusters. We standardized (Z-score normalization) the five attributes to have all input features use the same scale. We used Principle Component Analysis to reduce the dimensionality to two features using a randomized single-value decomposition solver with a tolerance for zero values. Since there was a lot of redundancy in the five features

(for example, the percentage of correct and percentage of incorrect scores were actually the same information), PCA increased the various measures for the value of the resulting clusters. The first two components retained 92.6% of the information contained in the original work session attributes. The first (PC1) explained 73.7% of the variance, and the second (PC2) explained 19.0%. The resulting noise variance is 12.3%. Table 2 shows the factor loadings of each of the five work session attributes to the two principal components.

Table 2. Factor loadings of each work session attribute for the two principal components (factors).

	PC1	PC2
% of Incorrect Attempts	−0.993	0.0477
% of Correct Attempts	0.995	−0.0603
% of Consecutive Incorrect Attempts	0.905	0.0846
% of Consecutive Correct Attempts	0.904	−0.0799
Median of Time in Between	0.266	0.964

The output of the PCA was fed to an unsupervised learning algorithm, Fuzzy C-Means clustering, which categorized the work sessions into two clusters that represented credit-seeking and active learning behaviors. Our model was randomly initialized using an initial fuzzy c-partitioned matrix and used a weighting exponent (the fuzzier) of 2. Centroids converged after 28 iterations. We trained our model on the Fall 2020 and Spring 2021 datasets with 8679 data points (work sessions) and tested it on the Spring 2022 dataset (4010 data points) to measure the generalizability of our model to work sessions that had never been seen before in the training step.

We assessed the goodness of our clustering technique with two metrics. The Fuzzy Partition Coefficient (FPC) indicates how cleanly a dataset is described by a certain model Ross (2012). The FPC has a range of 0 to 1, with 1 indicating that the dataset is best described. The Silhouette coefficient measures how similar an object is to its cluster (cohesion) when compared to other clusters (separation) Rousseeuw (1987). The Silhouette score runs from −1 to +1. A score of +1 suggests that the object is well matched to its cluster and poorly matched to other clusters. Values of 0 indicate overlapping clusters. If many data points have a low or negative score, then the clustering is poor, and there may be too many or too few clusters. We took the average Silhouette coefficient over all data points to measure the goodness of our model.

In the training dataset, we achieved an FPC score of 0.824 and an average Silhouette score of 0.549. Both clusters had a close Silhouette score, and no cluster completely fell far from the mean value of 0.549. Cluster 1 had an average Silhouette score of 0.508, and cluster 2 had a Silhouette score of 0.568. By studying the data points that fell into each cluster, we found that points in cluster 1 reflected active learning behavior, while cluster 2 represented credit-seeking behavior. We also noted that the thickness of cluster 2 was greater than that of cluster 1, but this was due to the fact that the percentage of active learning sessions was greater than the percentage of credit-seeking sessions. We found that using subsets of work session attributes may have resulted in a higher Silhouette score, but with most data points in one group and few data points in the other. Thus, the model would perform poorly in distinguishing between credit-seeking and active learning behaviors.

To test the validity of our model, we used the two centroids from Fall 2020 and Spring 2021 to cluster the work sessions of Spring 2022. On the test dataset, we achieved an FPC score of 0.828 and a Silhouette score of 0.557, suggesting that our model generalized well to work sessions that it had not been trained on. Our training dataset (Fall 2020 and Spring 2021) had 64.5% of the work sessions clustered as active learning sessions, and the remaining 35.5% were credit-seeking sessions. Our test dataset (Spring 2022) had

68.3% of the work sessions clustered as active learning and the remaining 31.7% were credit-seeking sessions.

As an additional check, we manually examined and labeled certain work sessions as credit-seeking or active learning sessions and then compared the labeled work sessions to their predicted classifications. We randomly chose 300 different data points from Fall 2020 and Spring 2021, resulting in 196 (65.3%) active learning sessions and 104 (34.7%) credit-seeking sessions. Similarly, for Spring 2022, we chose 300 different data points resulting in 189 (63%) active learning sessions and 111 (37%) credit-seeking sessions. We found that our clustering model reached a perfect accuracy of 100 percent when comparing the 600 randomly selected data points to their equivalent predicted label.

To understand the consequences of credit-seeking behavior on PI framesets, we compared, in each semester, the two groups (credit-seeking and active learning) with respect to the total exam score and the overall score. Our dataset included all PI frameset attempts from Fall 2020, Spring 2021, and Spring 2022. In each semester, the final grade was out of 1000 points, with two midterms worth 100 points each and a final exam worth 150 points. The remaining 650 points were given for various autograded exercises in the eTextbook and written homework, though the details varied across the semesters. In Fall 2020, there was no credit incentive for students to finish the PI framesets. In Spring 2021, students were offered a 5% bonus on the semester points possible for completing all PI framesets of the eTextbook. In Spring 2022, PI frames were a standard part of the homework, with framesets collectively worth 20%.

For each semester, we found a negative correlation between credit-seeking behavior with the PI frames and the total exam score (two midterms and a final). For Fall 2020, $r = -0.416$; for Spring 2021, $r = -0.418$; and for Spring 2022, $r = -0.639$. All p-values were significantly below 0.001. The overall semester score had a non-significant negative correlation with credit-seeking behavior in Fall 2020 ($r = -0.283$; p-value $= 0.126 > 0.01$) but a significant negative correlation in Spring 2021 ($r = -0.400$; p-value $= 0.0013 < 0.01$) and a higher significant negative correlation in Spring 2022 ($r = -0.527$; p-value < 0.00001). Recall that the PI framesets were worth an increasing amount of credit across semesters.

We also took the approach of classifying students as credit-seeking or actively learning based on the number of sessions that were similarly classified. We specified a threshold of $t = 0.5$ such that if a student's percentage of credit-seeking sessions exceeded t, they were categorized as credit-seekers; otherwise, they were active learners. The analysis results might have varied depending on different thresholds, but we found similar results for a range of thresholds. With $t = 0.5$, in Fall 2020, we had 11 credit-seeking students and 56 actively learning students. In Spring 2021, we had 23 credit-seeking students and 41 actively learning students. In Spring 2022, we have 10 credit-seeking students and 58 actively learning students.

We used Welch's Two-Sample Unpooled T-Test to test the hypothesis that the average score in credit-seeking students was lower than the average score in actively learning students. Table 3 shows that in Spring 2021 and Spring 2022, the mean scores of credit-seeking students were significantly lower than the mean scores of actively learning students ($p < 0.01$). In Fall 2020, when no credit was involved, there was significant difference in the total exam scores.

While statistical significance demonstrated the existence of an impact, the effect size was used as an indicator of whether the effect was large enough to be important. Table 3 shows that for the total exam score, in Fall 2020, we had a large effect size of 0.79 (a difference of 24.3 actual points out of 350), and in Spring 2021, we also had a large effect size of 0.78 (a difference of 28.95 actual points out of 350). In Spring 2022, we had an effect size of 0.68 (a difference of 34.17 actual points out of 350). The difference in the total exam

scores in Spring 2022 was much larger than it was in Fall 2020 and Spring 2021. In part, this could be because the increasing use of PI framesets allowed us to better distinguish between credit-seeking and active learning behaviors.

For the overall score, in Spring 2021, we had a large effect size of 0.77 (a difference of 7.3 actual points out of 100), and in Spring 2022, we had an effect size of 0.45 (a difference of 3.26 out of 100). The difference in Spring 2022 overall score was not so high as the total exam score since it included homework and exercises along with exams, which were less impacted by the PI frameset knowledge.

Collectively, these results indicate that there was a practical difference in scores between students who were seeking credit and students who were not.

Table 3. Comparison between average score of credit-seeking students and actively learning students. Exam scores are out of 350; overall scores are out of 100.

		Credit-Seeking			Actively Learning			*p*-	Effect
	Semester	Mean	std	Count	Mean	Std	Count	Value	Size
Total Exam Score	Fall 2020	265.55	26.71	11	289.85	31.58	56	0.0083	0.79
Total Exam Score	Spring 2021	268.77	45.04	23	297.72	32.30	41	0.0051	0.78
Total Exam Score	Spring 2022	234.9	51.50	10	269.07	49.79	58	0.037	0.68
Overall Score	Fall 2020	82.81	7.24	11	85.59	8.12	56	0.14	0.35
Overall Score	Spring 2021	78.14	10.26	23	85.44	8.94	41	0.0034	0.77
Overall Score	Spring 2022	79.96	6.77	10	83.22	7.36	58	0.094	0.45

5.2. Fine-Grained Analysis

Our PI frameset library currently supports three categories of questions: true/false and similar (such as yes/no) where the student selects one of two choices; single-choice questions, where the student selects exactly one from a set of choices (see Figure 1); and multi-choice questions, where the correct answer will be some subset of a collection of choices (see Figure 2). We analyzed patterns of credit-seeking behavior on each of these question types.

Recall that the philosophy of PI does not require or expect that questions be difficult to answer but rather that they maintain engagement. But, from a perspective of blind guessing, multi-choice questions clearly are the most difficult since the number of possible responses is $2^n - 1$ when there are n choices.

We found that 82.2% of work sessions contained at least one multi-choice question attempt: 61.4% of them were clustered as active learning sessions, and 38.6% were clustered as credit-seeking. Credit-seekers took more time to solve multi-choice questions than actively learning students. On average, in credit-seeking sessions, students spent 5.04 minutes solving multi-choice questions, while actively learning students spent only 3.36 minutes solving the same number of multi-choice questions, a statistically significant difference ($p < 0.000001$). Credit-seeking students attempted to answer multi-choice questions quickly with an average time gap of 5.4 seconds between two consecutive attempts, compared to 16.8 seconds for active learners, a statistically significant difference ($p < 0.000001$).

We also found that the median number of choices for multi-choice questions correlated with the student's behavior in a session. As shown in Figure 3, having a median of two or three choices per multi-choice question resulted in fewer credit-seeking sessions than active learning sessions, while it was the opposite when there were four choices or more. Figure 4 shows that for credit-seeking students, as the number of choices increased, the time it took the students to reach the correct solution also increased. Naturally, we also found that for credit-seekers, the higher the number of choices was, the higher the percentage of consecutive incorrect attempts was. For actively learning students, more choices did not necessarily increase the solving time or the number of attempts.

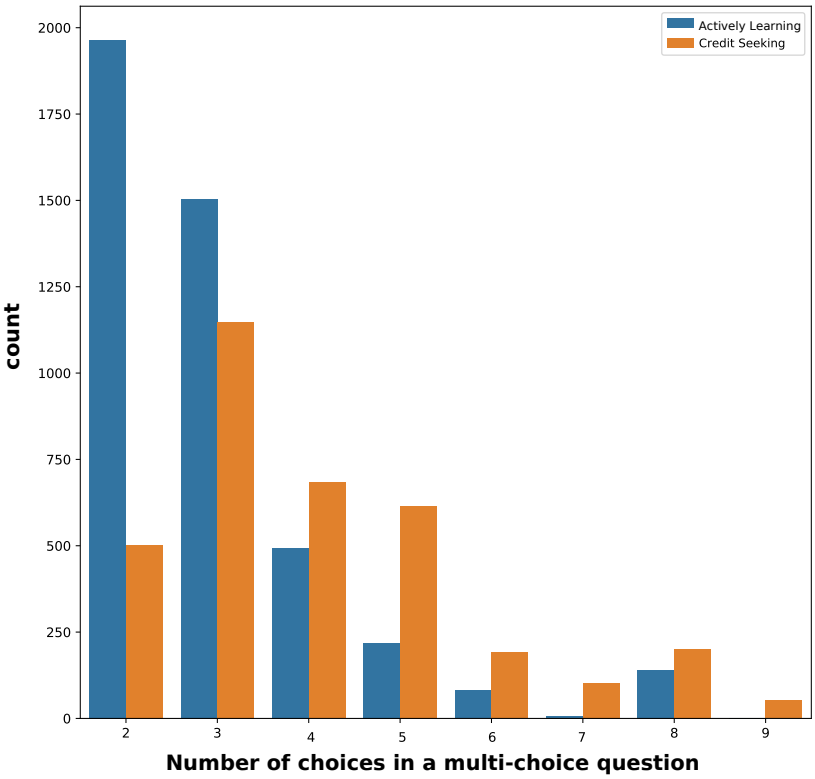


Figure 3. The number of active learning or credit-seeking sessions based on the median number of choices for all multi-choice questions in a given session.

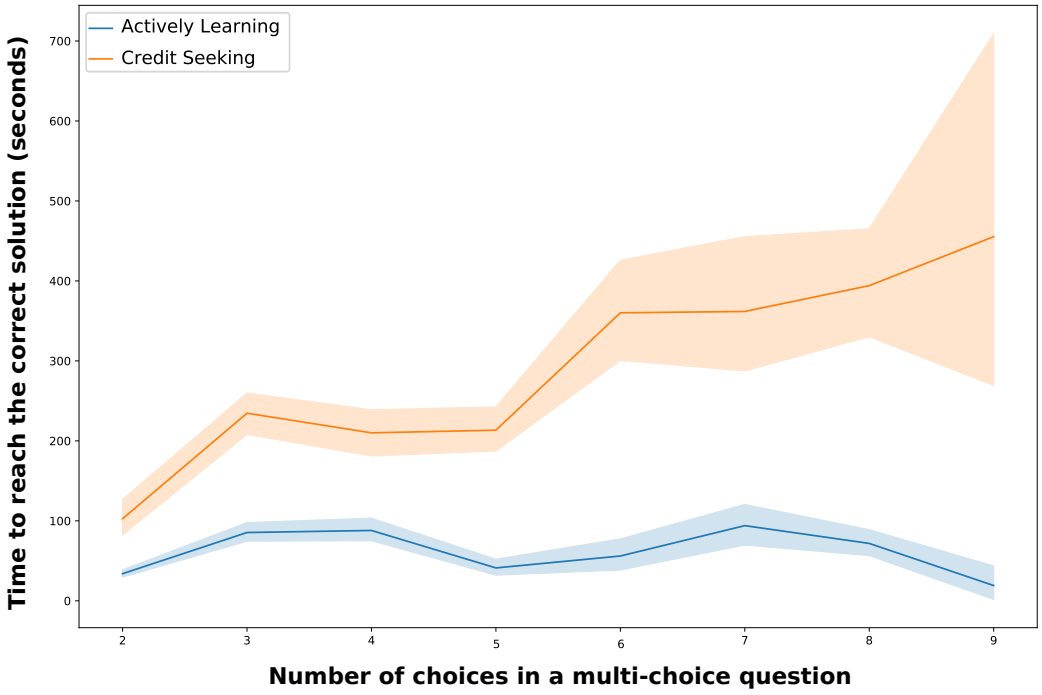


Figure 4. The relationship between the number of choices in a multi-choice question and the time spent on incorrect attempts on that question.

In total, 39.6% of work sessions contained at least one single-choice question attempt. Of these, 66.0% were clustered as active learning sessions, and 34.0% were clustered as credit-seeking sessions. In contrast to multi-choice questions, increasing the number of choices in single-choice questions did not change the students' behavior. This means that if

a student was actively learning in most of their PI frameset sessions, then it was less likely they would convert to credit-seeking behavior due to a single-choice question, while they were more likely to perform so for multi-choice questions. Credit-seeking students did not take longer than actively learning students to correctly answer single-choice questions. However, in credit-seeking sessions, students attempted to answer single-choice questions quickly with an average of 14.70 seconds between any two consecutive attempts, while it took 21.04 seconds for active learners, a statistically significant difference ($p < 0.000001$).

In total, 37.85% of work sessions contained at least one T/F question attempt. Of these, 69.0% were active learning sessions, and 31.0% were credit-seeking sessions. Recognizing credit-seeking behavior in a T/F question can be challenging because students guess the right answer on the first try half the time. We analyzed the percentage of incorrect attempts on T/F questions in credit-seeking vs. active learning sessions. We found that the percentage of incorrect attempts on T/F questions in credit-seeking sessions, 34.47%, was significantly higher than the percentage of incorrect attempts on T/F questions in active learning sessions, 15.86% ($p < 0.000001$). When comparing only sessions with more T/F questions than single-choice and multi-choice questions, we found that in credit-seeking sessions, the percentage of incorrect attempts on T/F questions, 48.00%, was again significantly higher than the percentage of incorrect attempts in active learning sessions, 23.00% ($p < 0.000001$). We found that the average percentage of incorrect T/F attempts a student had for all of their sessions had a correlation of $r = 0.6119$ with their percentage of credit-seeking sessions, a significant correlation ($p < 0.000001$).

6. Discussion

Our results align with research indicating that credit-seeking behavior in educational software negatively impacts student learning outcomes R. Baker et al. (2004); R. S. Baker et al. (2004); Haig et al. (2009); Koh et al. (2018). Specifically, we observed that students who engaged in credit-seeking behavior demonstrated lower overall academic achievement and performed significantly worse on exams compared to their peers who did not exhibit such behavior. This finding is consistent with the work of R. S. Baker et al. (2004), who reported that students who game the system often have lower pre-test understanding and post-test performance.

Moreover, our research extends the findings of Aleven et al. (2006) and Peters et al. (2018) by providing a detailed analysis of how different types of questions influence credit-seeking behavior. We found that multi-choice questions are particularly prone to triggering credit-seeking behavior. This observation supports the notion that students misuse feedback and hints to obtain correct answers without genuine engagement with the material. By identifying the types of questions that are most susceptible to credit-seeking behavior, our study offers valuable insights for the design of educational software aimed at minimizing such behavior and promoting authentic learning.

Our results highlights the importance of designing educational tools that discourage gaming the system and foster meaningful student engagement. For instance, reducing the number of choices in multi-choice questions or incorporating adaptive feedback mechanisms could help mitigate credit-seeking behavior. Additionally, providing real-time interventions when credit-seeking behavior is detected could encourage students to engage more deeply with the material. Our findings contribute to a deeper understanding of student interactions with Programmed Instruction (PI) framesets in digital textbooks. By highlighting the negative correlation between credit-seeking behavior and learning outcomes, our study underscores the importance of designing educational tools that discourage gaming the system and foster meaningful student engagement.

7. Threats to Validity

The most critical limitation of this study is that we analyzed only the interactions that happened within the Programmed Instruction framework. It is possible that credit-seeking behavior with PI framesets is different than that in other activities. Future studies should consider other external resources, such as the amount of time spent on other auto-graded exercises, the frequency of webpage reloads, the tracking of the mouse, etc., to add more diversity to the input features of the clustering model and provide more information about students' behavior across all parts of the course.

This study clusters students into two behaviors only, either credit-seeking or active learning. Better analysis might distinguish between credit-seeking behavior versus when a student is struggling and honestly does not know the correct answer.

8. Conclusions and Future Work

This study explored the credit-seeking behavior of students using Programmed Instruction in a Formal Languages course. We found that the percentage of credit-seeking sessions for a student was negatively correlated with their scores, such as the total exam scores and the overall final score. Comparing populations, we found that the average score in credit-seeking students was significantly lower than the average score in actively learning students.

Looking at individual question types, we found that it took more time for credit-seeking students to solve multi-choice questions than it did for actively learning students and that the number of choices in multi-choice questions correlated with the students' chances of exhibiting credit-seeking behavior in this session. The results of this study contribute to a deeper understanding of student interactions with PI-based digital textbooks and guide the effective use of such eTextbooks.

Our clustering model could be used in real time to provide a personalized and supportive message to offer additional resources and assistance to students exhibiting patterns associated with credit-seeking behavior. This approach aims to foster a more inclusive and supportive learning environment, acknowledging the diverse challenges students may face while still providing the necessary guidance for improved academic outcomes.

Author Contributions: Conceptualization, C.A.S. and Y.E.; methodology, Y.E., C.A.S. and M.F.; software, Y.E. and M.M.; validation, Y.E., M.F. and C.A.S.; formal analysis, Y.E.; investigation, Y.E., M.F. and C.A.S.; resources, C.A.S. and M.M.; data curation, Y.E. and M.M; writing—original draft preparation, Y.E.; writing—review and editing, C.A.S. and M.F.; visualization, Y.E.; supervision, C.A.S. and M.F.; project administration, C.A.S.; funding acquisition, C.A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Science Foundation under CCRI award number 2213790.

Institutional Review Board Statement: This study was approved by the Institutional Review Board of Virginia Tech, IRB number 17-1095, with the most recent revision approved on 12 March 2025.

Informed Consent Statement: Informed consent was obtained from all subjects where required under the terms of the approved IRB protocol.

Data Availability Statement: The original data presented in the study are openly available in GitHub through this link: https://github.com/OpenDSA/Analysis/tree/master/Yusuf/Datasets.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

PI	Programmed Instruction
FLA	Formal Languages and Automata
PCA	Principal Components Analysis
FPC	Fuzzy Partition Coefficient

References

Aleven, V., Mclaren, B., Roll, I., & Koedinger, K. (2006). Toward meta-cognitive tutoring: A model of help seeking with a cognitive tutor. *International Journal of Artificial Intelligence in Education*, *16*, 101–128.

Baker, R., Corbett, A., & Koedinger, K. (2004). Detecting student misuse of intelligent tutoring systems. In *Intelligent tutoring systems: 7th international conference, ITS 2004, Maceió, Alagoas, Brazil, August 30–September 3, 2004. Proceedings 7* (Vol. 3220, pp. 531–540). Springer. [CrossRef]

Baker, R., Mitrovic, A., & Mathews, M. (2010). Detecting gaming the system in constraint-based tutors. In *International conference on user modeling, adaptation, and personalization* (Vol. 6075, pp. 267–278). Springer. [CrossRef]

Baker, R. S., Corbett, A. T., Koedinger, K. R., & Wagner, A. Z. (2004). Off-task behavior in the cognitive tutor classroom: When students "Game the System". In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 383–390). Association for Computing Machinery. [CrossRef]

Beck, J. (2005). Engagement tracing: Using response times to model student disengagement. In *Proceedings of the ITS2004 workshop on social and emotional intelligence in learning environments* (pp. 88–95). IOS Press.

Cheng, R., & Vassileva, J. (2006). Design and evaluation of an adaptive incentive mechanism for sustained educational online communities. *User Modeling and User-Adapted Interaction*, *16*, 321–348. [CrossRef]

Computing Curricula, A., & Society, I. (2013). *Computer science curricula 2013: Curriculum guidelines for undergraduate degree programs in computer science*. Association for Computing Machinery.

Fouh, E., Akbar, M., & Shaffer, C. (2012). The role of visualization in computer science education. *Computers in the Schools*, *29*, 95–117.

Fouh, E., Breakiron, D. A., Hamouda, S., Farghally, M. F., & Shaffer, C. A. (2014). Exploring students learning behavior with an interactive etextbook in computer science courses. *Computers in Human Behavior*, *41*(C), 478–485. [CrossRef]

Fouh, E., Karavirta, V., Breakiron, D., Hamouda, S., Hall, T. S., Naps, T., & Shaffer, C. (2014). Design and architecture of an interactive eTextbook—The OpenDSA system. *Science of Computer Programming*, *88*, 22–40. [CrossRef]

Haig, E., Hershkovitz, A., & Baker, R. (2009, July 8–12). *The impact of off-task and gaming behaviors on learning: Immediate or aggregate?* Proceedings of the 14th International Conference on Artificial Intelligence in Education (Aied) (pp. 507–514), Recife, Brazil. [CrossRef]

Johns, J., & Woolf, B. P. (2006). *A dynamic mixture model to detect student motivation and proficiency*. American Association for Artificial Intelligence.

Koh, K. H., Fouh, E., Farghally, M. F., Shahin, H., & Shaffer, C. A. (2018). Experience: Learner analytics data quality for an etextbook system. *Journal of Data and Information Quality (JDIQ)*, *9*(2), 1–10. [CrossRef]

Lockee, B., Moore, D., & Burton, J. (2004). Foundations of programmed instruction. In *Handbook of research on educational communications and technology* (pp. 545–569). Routledge

Ma, Y., Cain, K., & Ushakova, A. (2024). Application of cluster analysis to identify different reader groups through their engagement with a digital reading supplement. *Computers & Education*, *214*, 105025.

Mohammed, M. (2021). *Teaching formal languages through visualizations, machine simulations, auto-graded exercises, and programmed instruction* [Ph.D. dissertation]. Faculty of the Virginia Polytechnic Institute and State University.

Mohammed, M., Shaffer, C. A., & Rodger, S. H. (2021, March 13–20). Teaching formal languages with visualizations and auto-graded exercises. *52nd ACM Technical Symposium on Computer Science Education* (pp. 569–575), Virtual Event, USA. [CrossRef]

Molenda, M. (2008). The programmed instruction era: When effectiveness mattered. *TechTrends*, *52*, 52–58.

Munshi, A., Biswas, G., Baker, R., Ocumpaugh, J., Hutt, S., & Paquette, L. (2023). Analysing adaptive scaffolds that help students develop self-regulated learning behaviours. *Journal of Computer Assisted Learning*, *39*(2), 351–368.

Murray, R. C., & Vanlehn, K. (2005, July 18–22). *Effects of dissuading unnecessary help requests while providing proactive help*. 12th International Conference on Artificial Intelligence in Education (AIED) (pp. 887–889), Recife, Brazil.

Peters, C., Arroyo, I., Burleson, W., Woolf, B., & Muldner, K. (2018). Predictors and outcomes of gaming in an intelligent tutoring system. In *Intelligent tutoring systems: 14th international conference, ITS 2018, Montreal, QC, Canada, June 11–15. 2018, proceedings 14* (pp. 366–372). Springer. [CrossRef]

Rocha, H. J. B., de Barros Costa, E., & de Azevedo Restelli Tedesco, P. C. (2024). A knowledge engineering-based approach to detect gaming the system in novice programmers. In *Brazilian conference on intelligent systems* (pp. 18–33). Springer.

Ross, T. J. (2012). *Fuzzy logic with engineering applications* (3rd ed.). Wiley.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65. [CrossRef]

Sambasivarao, R. (2020). Impact of programmed instruction in learning mathematics. *Aut Aut*, *XI*, 269.

Skinner, B. (1986). Programmed instruction revisited. *Phi Delta Kappan*, *68*(2), 103–110.

Tait, K., Hartley, J. R., & Anderson, R. C. (1973). Feedback procedures in computer-assisted arithmetic instruction. *British Journal of Educational Psychology*, *43*, 161–171.

Walonoski, J., & Heffernan, N. (2006). Prevention of off-task gaming behavior in intelligent tutoring systems. In *Intelligent tutoring systems: 8th international conference, ITS 2006, Jhongli, Taiwan, June 26–30. 2006. Proceedings 8* (pp. 722–724). Springer Berlin Heidelberg. [CrossRef]

Wangila, M. J., Martin, W., & Ronald, M. O. (2015). Effect of programmed instruction on students' attitude towards structure of the ATOM and the periodic table among Kenyan Secondary Schools. *Science Education International*, *26*, 488–500.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.