Can Automated Feedback Turn Students into Happy Prologians?

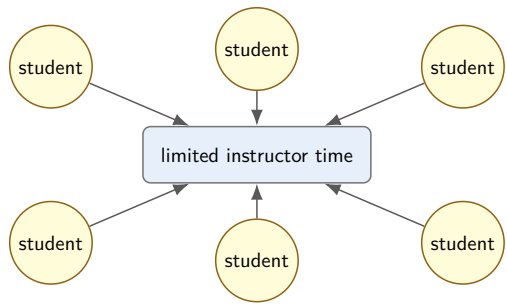
Ricardo Brancas [1] Pedro Orvalho [2] Carolina Carreira [1]
Vasco Manquinho [1] Ruben Martins [3]

[1] INESC-ID / Instituto Superior Técnico, Universidade de Lisboa, Portugal
[2] Institut d'Investigació en Intelligència Artificial (IIIA-CSIC), Barcelona, Spain
[3] Carnegie Mellon University, USA

The 42nd International Conference on Logic Programming, **ICLP @ FLoC 2026**
Lisbon, 21 July, 2026

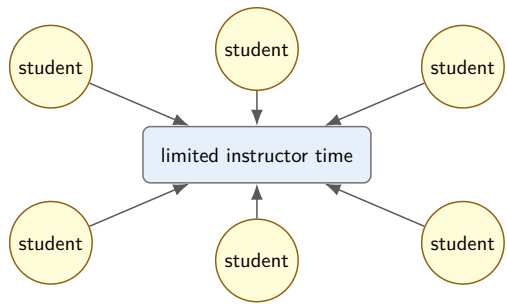
The problem: feedback does not scale

- Timely, personalized feedback matters for learning and computer-aided education.



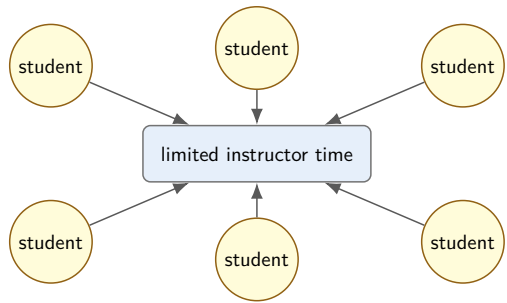
The problem: feedback does not scale

- Timely, personalized feedback matters for learning and computer-aided education.
- In large programming courses, instructors cannot inspect every attempt.



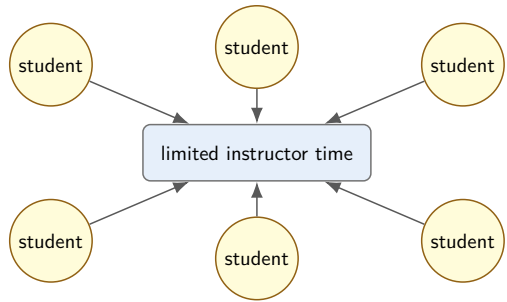
The problem: feedback does not scale

- Timely, personalized feedback matters for learning and computer-aided education.
- In large programming courses, instructors cannot inspect every attempt.
- Logic programming adds a twist: novices struggle with declarative execution, non-determinism, and non-termination.



The problem: feedback does not scale

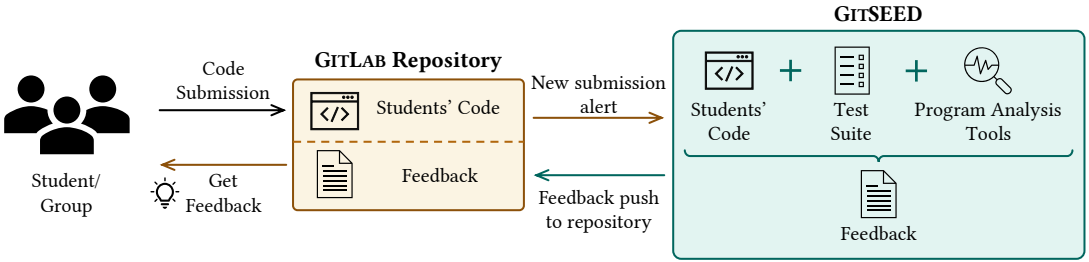
- Timely, personalized feedback matters for learning and computer-aided education.
- In large programming courses, instructors cannot inspect every attempt.
- Logic programming adds a twist: novices struggle with declarative execution, non-determinism, and non-termination.



Question

Can automated feedback help students debug PROLOG, and which feedback do students actually value?

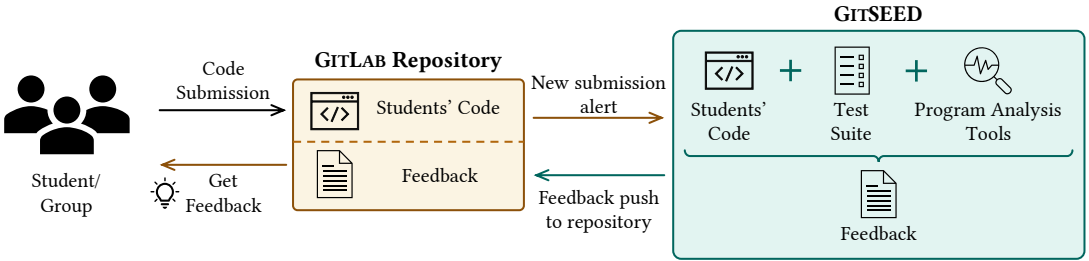
ProHelp: feedback inside the students' Git workflow



Design goal

No new assessment interface: students receive immediate feedback where they already submit and version their code.

ProHelp: feedback inside the students' Git workflow

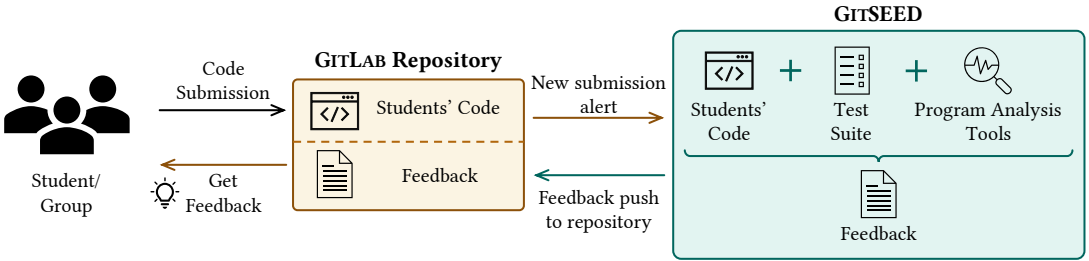


Design goal

No new assessment interface: students receive immediate feedback where they already submit and version their code.

- A git push triggers the pipeline.

ProHelp: feedback inside the students' Git workflow

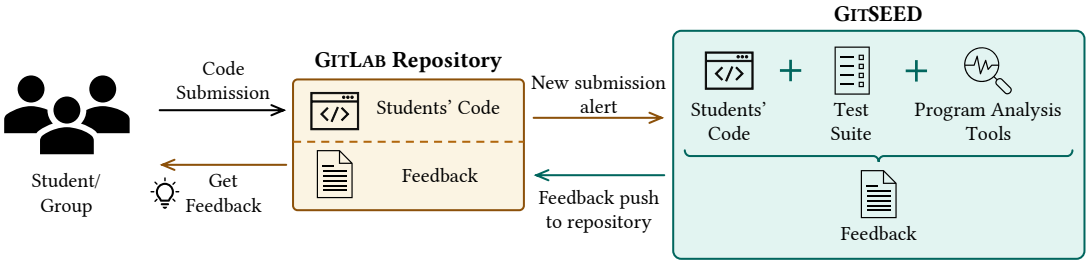


Design goal

No new assessment interface: students receive immediate feedback where they already submit and version their code.

- A git push triggers the pipeline.
- GitSEED runs tests and analyses.

ProHelp: feedback inside the students' Git workflow

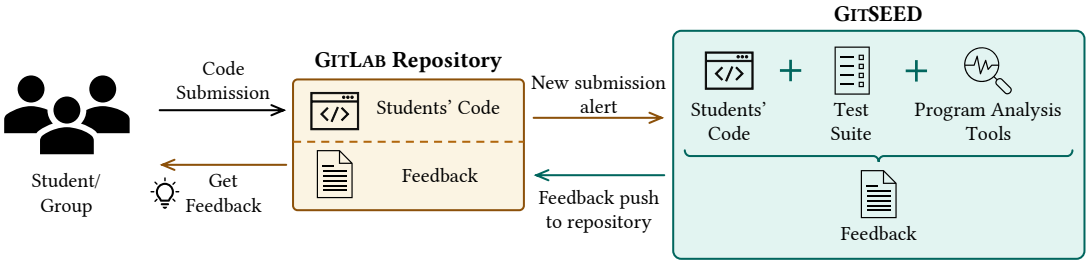


Design goal

No new assessment interface: students receive immediate feedback where they already submit and version their code.

- A git push triggers the pipeline.
- GitSEED runs tests and analyses.
- ProHelp commits a report back to the repository.

ProHelp: feedback inside the students' Git workflow



Design goal

No new assessment interface: students receive immediate feedback where they already submit and version their code.

- A git push triggers the pipeline.
- GitSEED runs tests and analyses.
- ProHelp commits a report back to the repository.
- Students iterate from the feedback.

Seven feedback mechanisms

1. **Automatic testing**

Seven feedback mechanisms

1. **Automatic testing**
2. **Predicate scoring**

Test 1: Successful

Test 2: Successful

Test 3: Unsuccessful

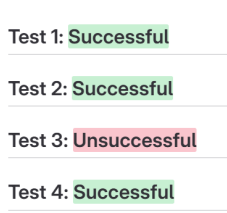
Test 4: Successful

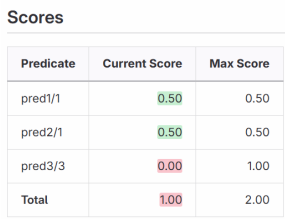
Scores

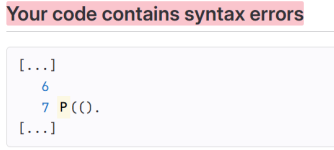
Predicate	Current Score	Max Score
pred1/1	0.50	0.50
pred2/1	0.50	0.50
pred3/3	0.00	1.00
Total	1.00	2.00

Seven feedback mechanisms

1. **Automatic testing**
2. **Predicate scoring**
3. **Syntax highlighting**

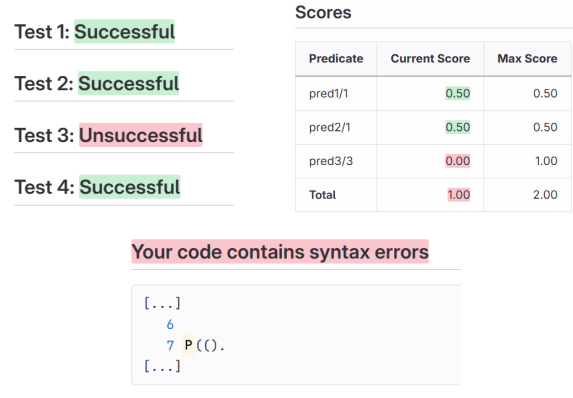






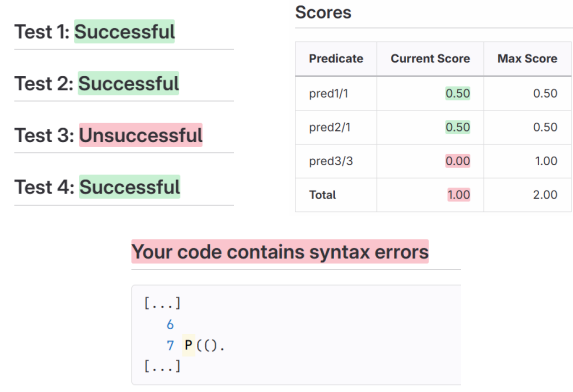
Seven feedback mechanisms

1. **Automatic testing**
2. **Predicate scoring**
3. **Syntax highlighting**
4. **Open choice-point warnings**



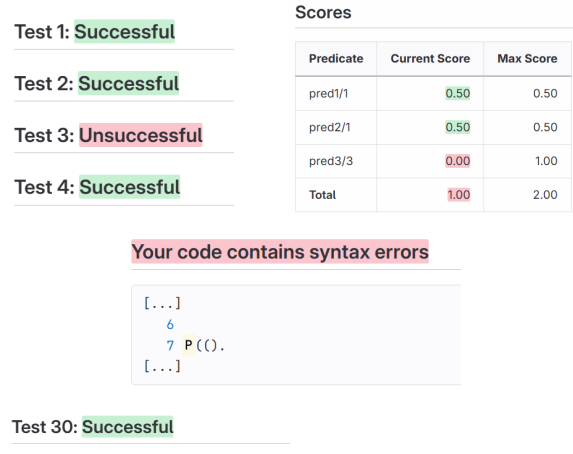
Seven feedback mechanisms

1. **Automatic testing**
2. **Predicate scoring**
3. **Syntax highlighting**
4. **Open choice-point warnings**
5. **Score rankings**



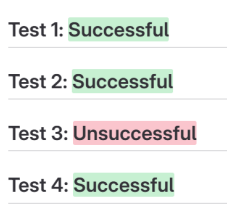
Seven feedback mechanisms

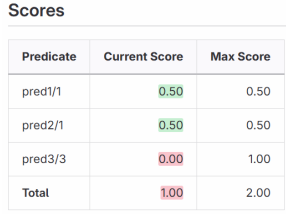
1. **Automatic testing**
2. **Predicate scoring**
3. **Syntax highlighting**
4. **Open choice-point warnings**
5. **Score rankings**
6. **Solution-type validation**

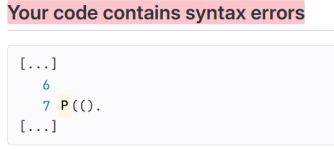


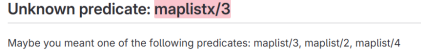
Seven feedback mechanisms

1. **Automatic testing**
2. **Predicate scoring**
3. **Syntax highlighting**
4. **Open choice-point warnings**
5. **Score rankings**
6. **Solution-type validation**
7. **Unknown-predicate suggestions**







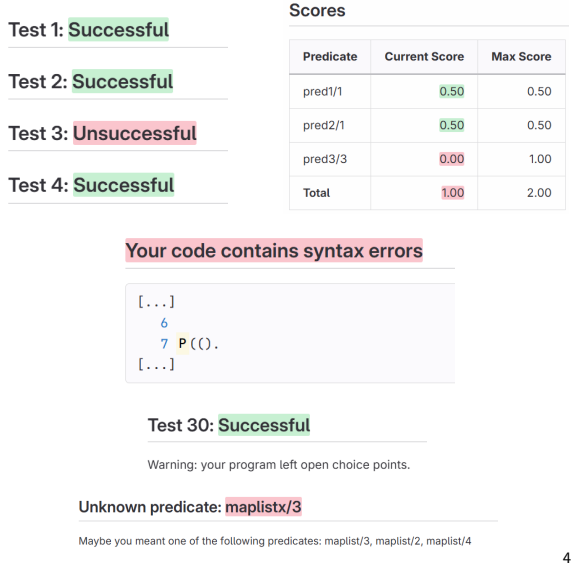


Seven feedback mechanisms

1. **Automatic testing**
2. **Predicate scoring**
3. **Syntax highlighting**
4. **Open choice-point warnings**
5. **Score rankings**
6. **Solution-type validation**
7. **Unknown-predicate suggestions**

Range

Correctness, localization, Prolog-specific behavior, and implementation constraints.



Study at a glance

Deployment

- BSc logic for programming course

Study at a glance

Deployment

- BSc logic for programming course
- 365 enrolled students

Study at a glance

Deployment

- BSc logic for programming course
- 365 enrolled students
- introductory exercises, recitations, and a course project

Study at a glance

Deployment

- BSc logic for programming course
- 365 enrolled students
- introductory exercises, recitations, and a course project
- 312 students interacted with ProHelp

Study at a glance

Deployment

- BSc logic for programming course
- 365 enrolled students
- introductory exercises, recitations, and a course project
- 312 students interacted with ProHelp

Study at a glance

Deployment

- BSc logic for programming course
- 365 enrolled students
- introductory exercises, recitations, and a course project
- 312 students interacted with ProHelp

Survey

- 148 responses

Study at a glance

Deployment

- BSc logic for programming course
- 365 enrolled students
- introductory exercises, recitations, and a course project
- 312 students interacted with ProHelp

Survey

- 148 responses
- 144 valid after attention checks

Study at a glance

Deployment

- BSc logic for programming course
- 365 enrolled students
- introductory exercises, recitations, and a course project
- 312 students interacted with ProHelp

Survey

- 148 responses
- 144 valid after attention checks
- post-experience Likert questions

Study at a glance

Deployment

- BSc logic for programming course
- 365 enrolled students
- introductory exercises, recitations, and a course project
- 312 students interacted with ProHelp

Survey

- 148 responses
- 144 valid after attention checks
- post-experience Likert questions
- usefulness, student factors, and future features

Study at a glance

Deployment

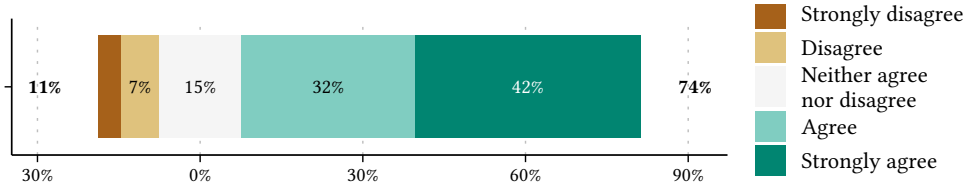
- BSc logic for programming course
- 365 enrolled students
- introductory exercises, recitations, and a course project
- 312 students interacted with ProHelp

Survey

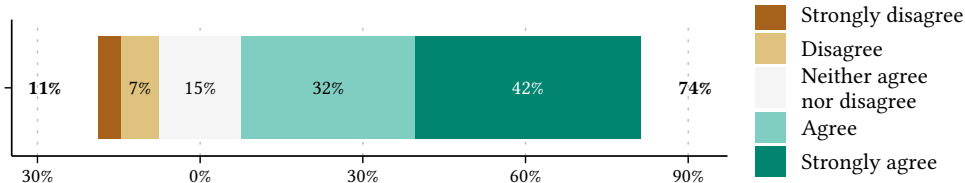
- 148 responses
- 144 valid after attention checks
- post-experience Likert questions
- usefulness, student factors, and future features

365 enrolled $\rightarrow$ **312** users $\rightarrow$ **144** valid survey responses

Did students think feedback helped?



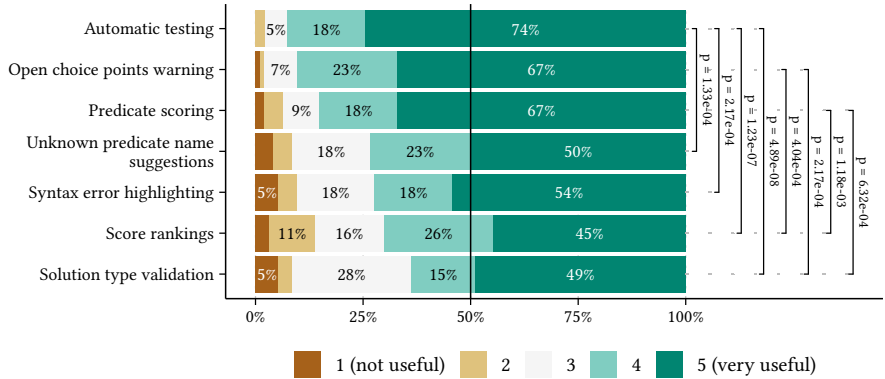
Did students think feedback helped?



Helpful?

Most respondents agreed that the automated feedback helped improve their course grade.

R1: Students found all types of feedback helpful



Clear winner

Automatic testing was rated most useful: 74% of valid respondents selected the highest usefulness level.

Result 2: Who found ProHelp helpful?

RQ2

Do students' prior interest and behaviour affect how helpful they find ProHelp overall?

Interest in Prolog

No evidence of an effect

Result 2: Who found ProHelp helpful?

RQ2

Do students' prior interest and behaviour affect how helpful they find ProHelp overall?

Interest in Prolog	**Use of optional exercises**
No evidence of an effect	No evidence of an effect

Result 2: Who found ProHelp helpful?

RQ2

Do students' prior interest and behaviour affect how helpful they find ProHelp overall?

Interest in Prolog	**Use of optional exercises**	**Use of LLMs**
No evidence of an effect	No evidence of an effect	No evidence of an effect

Result 2: Who found ProHelp helpful?

RQ2

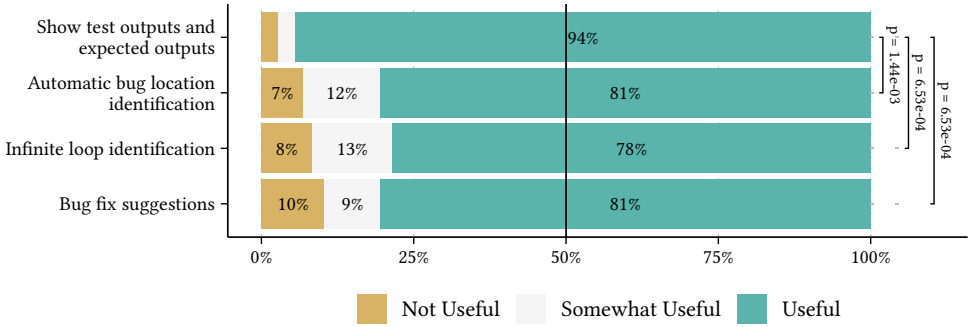
Do students' prior interest and behaviour affect how helpful they find ProHelp overall?

Interest in Prolog	**Use of optional exercises**	**Use of LLMs**
No evidence of an effect	No evidence of an effect	No evidence of an effect

Answer

There is no evidence that interest level, use of the platform for optional exercises, or use of LLMs affects how helpful students found ProHelp overall.

What students want in the future?



Future directions

Students want stronger diagnosis and repair support, but the system should guide without simply revealing the solution.

Takeaways

1. **Automated feedback can scale support** for large logic-programming classes.

Takeaways

1. **Automated feedback can scale support** for large logic-programming classes.
2. Students considered all seven feedback mechanisms useful, with **automatic testing ranked first**.

Takeaways

1. **Automated feedback can scale support** for large logic-programming classes.
2. Students considered all seven feedback mechanisms useful, with **automatic testing ranked first**.
3. Prolog-specific diagnostics add value beyond pass/fail testing.

Takeaways

1. **Automated feedback can scale support** for large logic-programming classes.
2. Students considered all seven feedback mechanisms useful, with **automatic testing ranked first**.
3. Prolog-specific diagnostics add value beyond pass/fail testing.

Can Automated Feedback Turn Students into Happy Prologians?

Yes, automated feedback can make happier Prologians, especially when it is immediate, concrete, and helps them debug rather than merely grades them.

Discussion

Questions?

This work supported by Portuguese national funds through FCT, under project 2023.14280.PEX (DOI: 10.54499/2023.14280.PEX). This work was also supported by grant PID2022-139835NB-C21 funded by MCIN/AEI/10.13039/501100011033 and by ERDF, EU; and by HORIZON-MSCA-2025-PF research and innovation programme under the Marie Skłodowska-Curie (Sherlock4Py, GA No 101269051).. This work was additionally supported by the Carnegie Mellon University Portugal Program and FCT under grants PRT/BD/152086/2021 (DOI: 10.54499/PRT/BD/152086/2021) and PRT/BD/153739/2021 (DOI: 10.54499/PRT/BD/153739/2021). This work was also partially supported by the National Science Foundation (NSF) under Award CCF2427581 and DARPA under Agreement FA8750-24-9-1000.