OverCode: Visualizing Variation in Student Solutions to Programming Problems at Scale

Elena Glassman, Jeremy Scott, Rishabh Singh, Philip Guo, Rob Miller
MIT CSAIL, University of Rochester Computer Science
{elg,jks,rishabhs,rcm}@mit.edu, pg@cs.rochester.edu

Problem

• In MOOCs, a single programming exercise may produce **thousands of solutions** from learners.
•There may be **several distinct, correct solutions** to coding assignments.
•**Some solutions may be unknown** to the teaching staff.
•Understanding solution variation is important for providing **appropriate feedback to students at scale**.

OverCode Approach

•**Visualize and explore** thousands of **programming solutions**
•Use both **static and dynamic analysis to cluster** similar *correct* solutions
  •*rename variables* to create clean composite solutions that reflect student naming choices
  •*scale linearly* with solution length and number of solutions, unlike CodeWebs.[1]
•**Let instructors further filter and cluster** solutions based on different criteria.

Evaluation

Dataset of Solutions

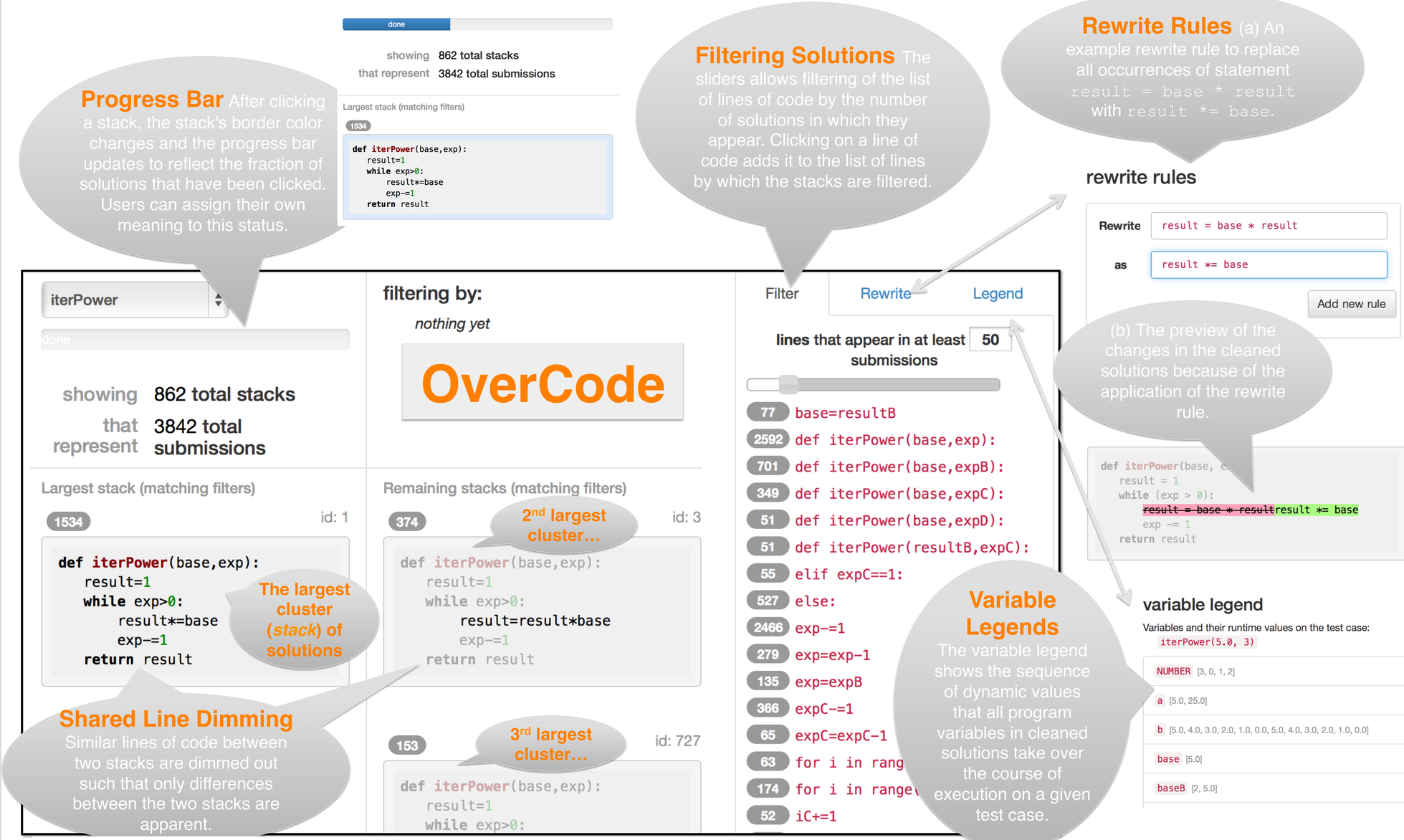
- We use a dataset of 1000's of *correct* solutions from 3 exercise problems in edX's 6.00x, an introductory programming course in Python.

User Studies

- We ran two user studies with **24 current and potential teaching assistants**.
- *Compared to an unclustered baseline*, OverCode allowed instructors to
  - more **quickly develop a high-level view** of students' understanding and misconceptions
  - **provide feedback that is relevant to more students**.

Summary

- **An information visualization approach is necessary** for instructors to explore the variations among solutions at the scale of MOOCs.
- **OverCode is an important step towards that goal.**



The OverCode user interface

- The top left panel shows the number of clusters, called *stacks*, and the total number of solutions visualized.
- The next panel down in the first column shows the largest stack, while the second column shows the remaining stacks.
- The third column shows the lines of code occurring in the cleaned solutions of the stacks together with their frequencies.

Acknowledgements

This material is based, in part, upon work supported by the National Science Foundation Graduate Research Fellowship (grant 1122374), the Microsoft Research Fellowship, the Bose Foundation Fellowship, and by Quanta Computer as part of the Qmulus Project. Any opinions, findings, conclusions, or recommendations in this paper are the authors', and do not necessarily reflect the views of the sponsors.

[1] Andy Nguyen, Chris Piech, Jonathan Huang, Leonidas Guibas. "Codewebs: Scalable Homework Search for Massive Open Online Programming Courses." In *Proceedings of the 23rd International World Wide Web Conference* (WWW 2014). Seoul, Korea, April 2014.