Title: How to Teach Recursion: A Formula to Transform an Iterative-Based to a Recursive-Based Method

Authors: Christian Servin

Published: Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE 2021), March 2021

Abstract:

The paper proposes an algorithmic technique to help students translate iterative programming concepts into recursive implementations. It has been used successfully in community college computer science courses.

Introduction:

- Recursion is an essential concept in computer science courses but also very challenging to teach effectively.

- Prior approaches help students identify base and recursive cases when writing recursive functions. However, no systematic process has been proposed to translate iterative to recursive code.

- The paper presents a 5-step teaching technique to transform loop-based code into recursive implementations.

- It has been used for 4 years in a community college CS2 course and shown improved student performance on recursion assignments and quizzes.

Overview:

- Recursion is introduced in CS1 after methods and more extensively in CS2 before data structures like linked lists and trees.

- The courses use Java as the programming language.

- The translation technique consists of:

1. Translating while loops

2. Identifying control variables

3. Identifying return data type

4. Exchanging boolean conditions

5. Exchanging parameters

- It handles 3 recursion types: linear, tail, and binary.

Evaluation:

- Two CS2 course sections were compared - one using the technique and one using regular materials.

- An assignment and quiz measured efficacy of the translation technique.

- Students using the technique implemented correct base cases and recursion more effectively.

- Students without technique had solutions lacking base cases and more syntax errors.

Contributions:

- The paper contributes a formal algorithm to translate iterative programming into recursive implementations.

- It shows the technique helps students handle recursion despite memorization of steps.

- The technique was implemented successfully in a community college CS2 course.

Future Work:

- Evaluate recursive programming skills of students who learned using this technique in subsequent CS courses.

- Expand the approach to additional types of recursion problems.

- Develop additional instruments to measure recursive programming capability.

Conclusion:

- The paper presents a novel 5-step algorithmic technique to translate iterative programming concepts into recursive implementations.

- It has been used successfully for 4 years in a community college CS2 course, leading to improved student performance on recursion.

- The technique helps students methodically transform loop-based code into correct recursive functions.