Program traces as things-to-think-with

Andrew Kay

kaya3@aston.ac.uk

School of Engineering and Applied Science
Aston University

13th July 2017

Traces

Code	Instructions	States

`int t=0;`
`for(int i=0; i<n; i++)`
`{`
`  t += a[i];`
`}`
	`int t=0;`
	
	`int i=0;`
	
	`t += a[i];`
	
	`i++`
	
	`t += a[i];`
	
	`i++`
	
	⋮
		
		$t = 0$
		
		$t = 0,\ i = 0$
		
		$t = 7,\ i = 0$
		
		$t = 7,\ i = 1$
		
		$t = 12,\ i = 1$
		
		$t = 12,\ i = 2$
		
		⋮

CESL — a new tool

- "Click, Evaluate, Show" Loop
- Act on program state, not on program code
- Record traces of student action, not computer action
- "Programs late" pedagogy — learn to do it yourself first, then tell the computer how

Prototype demonstration

- Live demonstration
- Video demonstration at `http://andrewkay.name/construit2017`

Thinking with traces

- Reviewing a trace
  - Ad hoc solution, or algorithm?
  - Repetition $\Longrightarrow$ algorithm uses a loop
  - Trace length $\Longrightarrow$ algorithm efficiency
  - Visual program simulation [6]
- Comparing multiple traces
  - Did they follow the same algorithm?
  - How can one piece of code do both?
- Traces as test-cases in debugging
  - Instead of a failure message, see where your code diverges from a correct trace [7]
  - Instruction trace tells you what code to write next
  - Can't use someone else's trace if their solution is different

Learning with traces

- Execution traces have been used successfully in programming education, esp. automated grading and feedback [2, 4, 7].
- Cognitive load theory [3]: using your own traces reduces the cognitive load of trying to understand somebody else's
- Process/object duality [1, 8] — traces are a level of abstraction between algorithm and code
- Papert: *"You can't think about thinking without thinking about thinking about something."* [5]

References

Mark Asiala et al. "A framework for research and curriculum development in undergraduate mathematics education". In: *Research in Collegiate Mathematics Education II, CBMS Issues in Mathematics Education* 6 (1996), pp. 1–32.

Matthew Hertz and Maria Jump. "Trace-Based Teaching in Early Programming Courses". In: *Proceedings of the 44th ACM Technical Symposium on Computer Science Education* (2013), pp. 561–566.

Jeroen J. G. van Merriënboer, Paul A. Kirschner, and Liesbeth Kester. "Taking the Load Off a Learner's Mind: Instructional Design for Complex Learning". In: *Educational Psychologist* 38.1 (2003), pp. 5–13.

Benjamin Paaßen, Joris Jensen, and Barbara Hammer. "Execution Traces as a Powerful Data Representation for Intelligent Tutoring Systems for Programming". In: *Proceedings of the 9th International Conference on Educational Data Mining*. 2016, pp. 183–190.

Seymour Papert. "You Can't Think About Thinking Without Thinking About Thinking About Something". In: *Contemporary Issues in Technology and Teacher Education* 5 (2005), pp. 366 –367.

Juha Sorva. "Visual Program Simulation in Introductory Program Education". PhD thesis. 2012.

Michael Striewe and Michael Goedicke. "Trace alignment for automated tutoring". In: *CAA 2013 International Conference, University of Southampton*. 2013.

David Tall et al. "What Is the Object of the Encapsulation of a Process?". In: *The Journal of Mathematical Behavior* 18.2 (1999), pp. 223–241.