A Comparison of Two Approaches for Hint Generation in Programming Tutors

Barry Peddycord III, Andrew Hicks, Irena Rindos, Christopher Simmons, Dr. Tiffany Barnes
{bwpeddyc, aghicks3, imrindos, csimmon2,tmbarnes}@ncsu.edu
Center for Educational Informatics, Department of Computer Science, North Carolina State University

NC STATE UNIVERSITY

CEI Center for Educational Informatics

Motivation

BOTS is an educational game that teaches programming

We want to incorporate Intelligent Tutoring System (ITS)-like feedback into the game

We explore two techniques to hint generation:
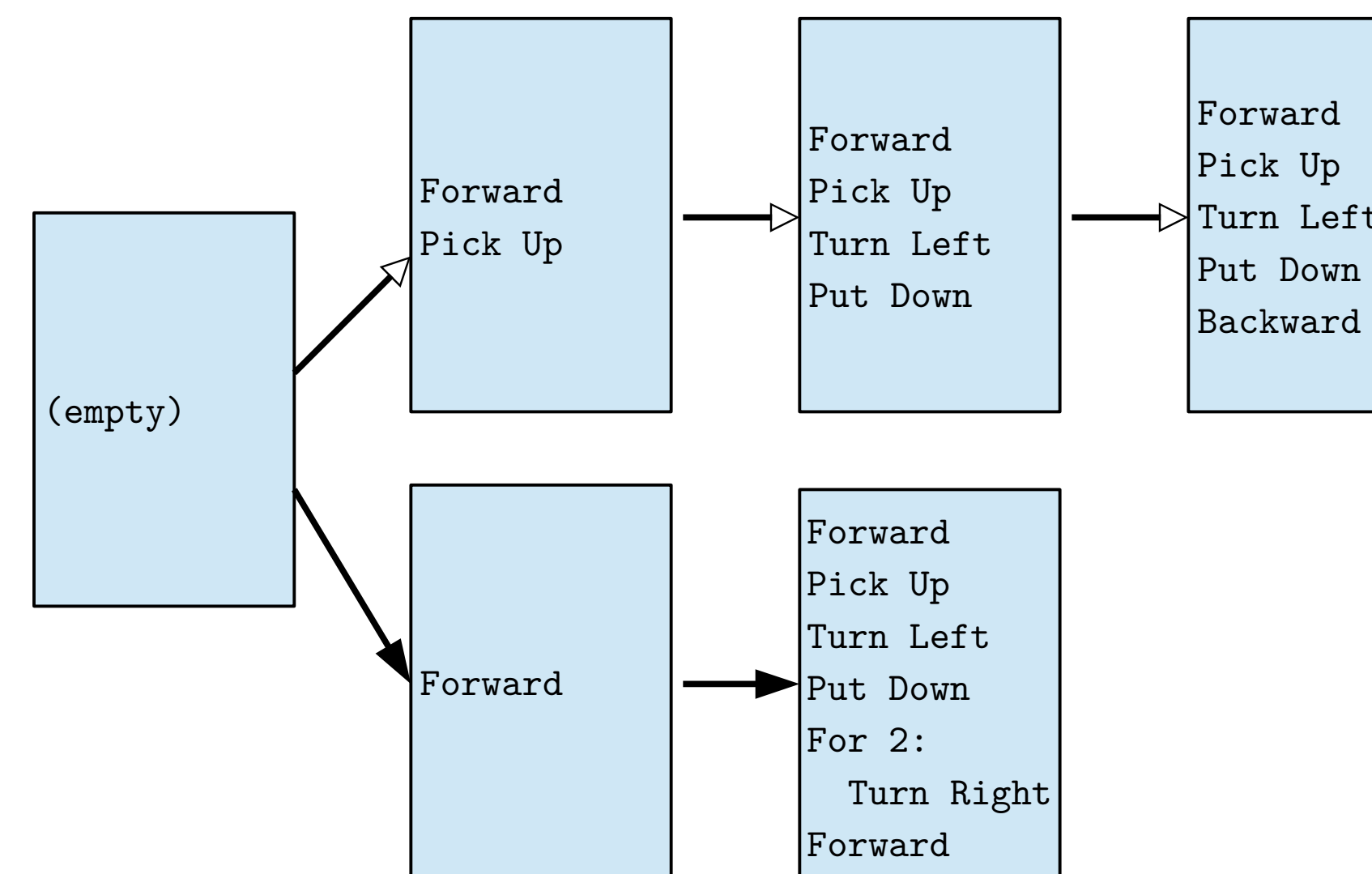- One based on source code (codestate)
- One based on output (worldstate)

Using worldstates, we are able to generate hints with *less data* than codestates alone

Example

A mock-up of how a hint would appear in BOTS. The green hologram represents the next step in a solution.
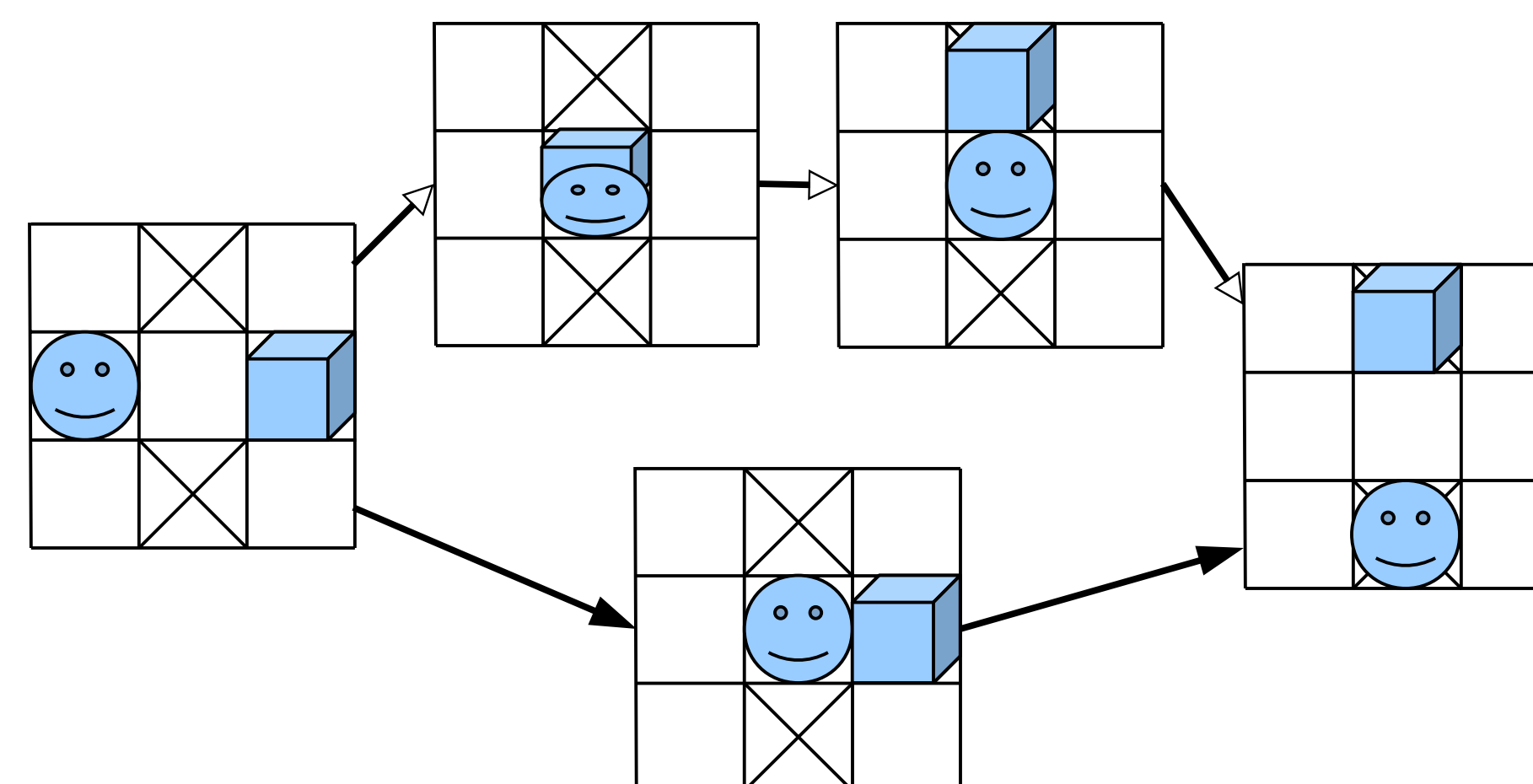
Generating Hints

We use the *Hint Factory* to generate hints in BOTS (Stamper et al, 2008)

We build an interaction network of all student solutions to a puzzle

We provide a hint to help a student in one state move to another state closer to the goal

Many different student programs can have the same output – a *worldstate* is the state of the BOTS world after running a student program
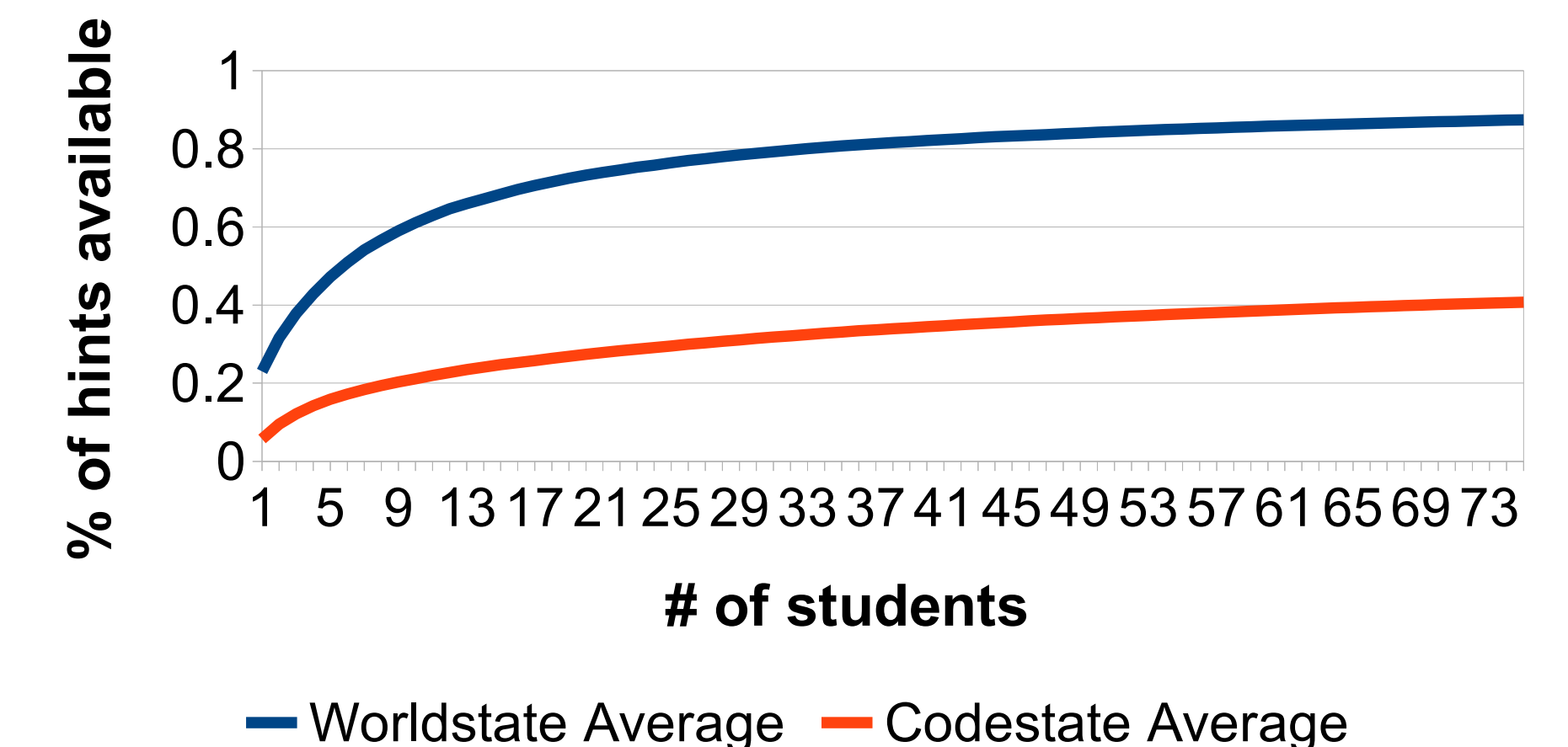

Results

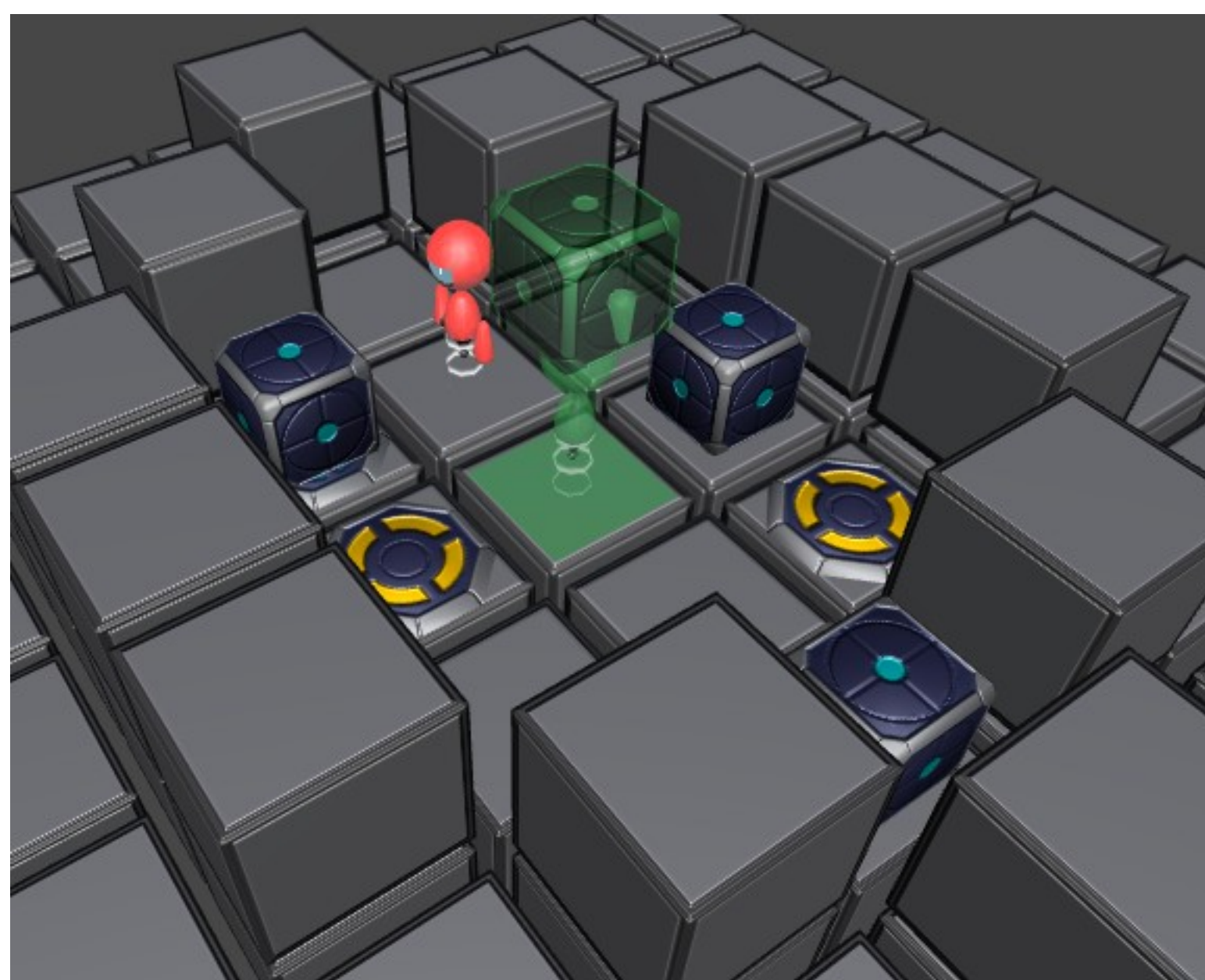Using worldstates substantially reduces the state space of student solutions

Puzzle	# Students	Unique Codestates	Unique Worldstates
Puzzle 1	107	560	191
Puzzle 2	98	431	133
Puzzle 3	89	278	30
Puzzle 4	86	208	45

We simulate students in BOTS asking for hints at every step of their solution

We report how often a hint is available as a function of how many students are in the interaction network

Future Work

We will be testing whether or not hints result in increased learning gains for BOTS players

We will also explore the potential of output-related hints in other programming environments, such as *Scratch*