Determining Problem Type Using Deep Reinforcement Learning in a Data-Driven Intelligent Tutor

Nazia Alam

Kimia Fazeli

Xiaoyi Tian
Presenter

Min Chi

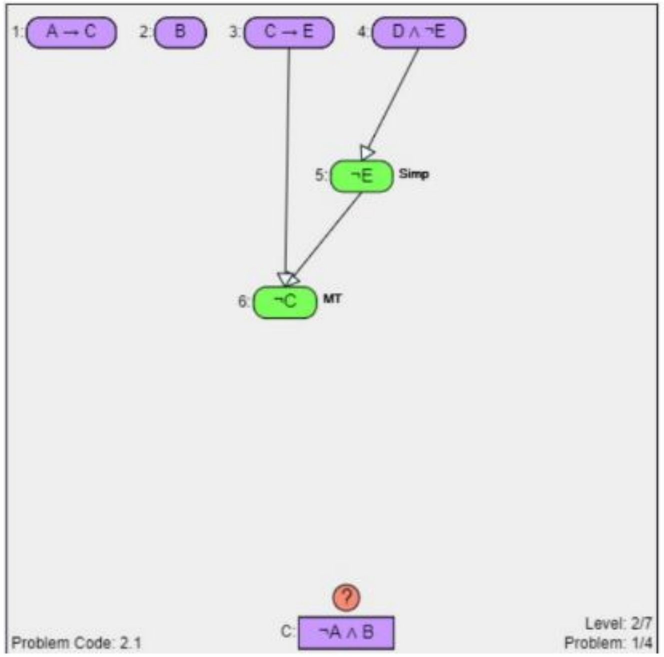
Tiffany Barnes

North Carolina State University, Raleigh, NC, USA

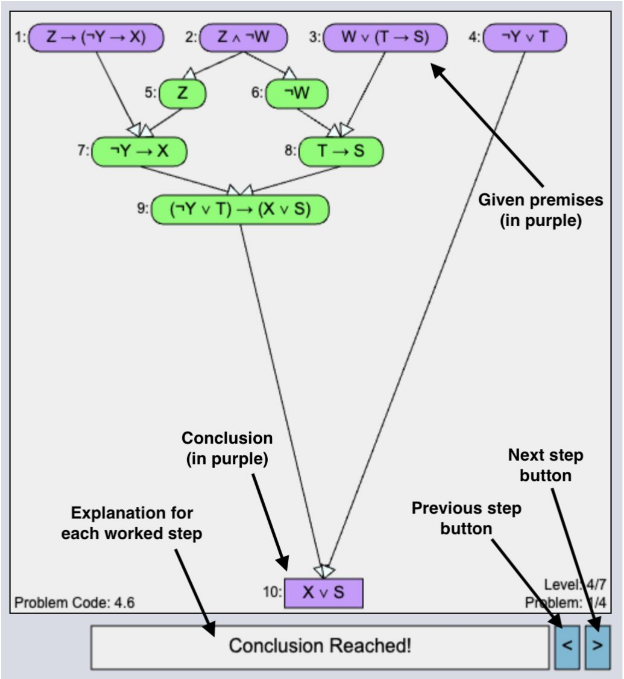
Background

- Different types of problems provided by Intelligent Tutoring Systems:
  - Problem solving (PS) where students solve the problems themselves
  - Worked examples (WE) where the problem is step-by-step solved for them
  - Parsons problems (PP) where the solution to a problem is presented as jumbled-up statements and students need to figure out the missing connections

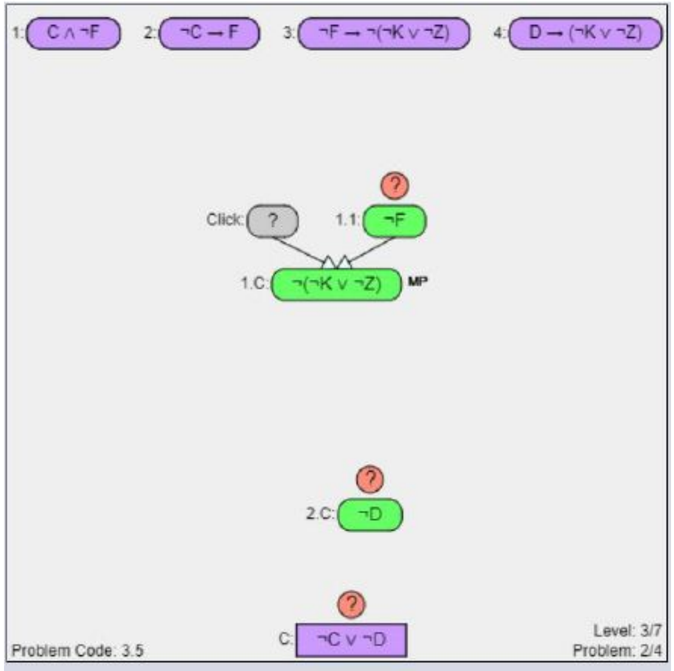
Problem Types in Logic Tutor



Problem Solving (PS)



Worked Example (WE)



Parsons Problem (PP)

Motivation

Cognitive Load theory suggests:

- Learning is most effective when the cognitive load is optimized [sweller et al. 2010]
- Cognitive load can be optimized by presenting problem types that align with students' proficiency and learning needs

Comparison between Problem Types

Problem Type	Benefits	Drawbacks
Problem-Solving (PS)	Encourages deep learning through full problem-solving	High cognitive load, time-consuming [Sweller et al., 1988]
Worked Examples (WE)	Reduces cognitive load, faster to complete [Shabrina et al., 2023]	Risk of passive learning, students may skip steps [Alam et al., 2024]
Parsons Problem (PP)	Balances the cognitive load and time demands [Ericson et al., 2018]	Effectiveness depends on chunking and student engagement

Motivation cont'd

- Prior work in programming education suggests **PPs can enhance learning efficiency** by lowering cognitive load and reducing time demands [Ericson et al., 2018].
- PPs have the potential to **balance the benefits** of Problem-Solving and Worked Examples (low cognitive load), potentially offering a more effective middle ground.
  - learning more effectively than WE, but less cognitive load than PS
- There has been research on PS and WE problems to investigate what type of problem to provide
  - **There has not been much research that included PP problems for logic domain**

Hypothesis

Parsons problem:

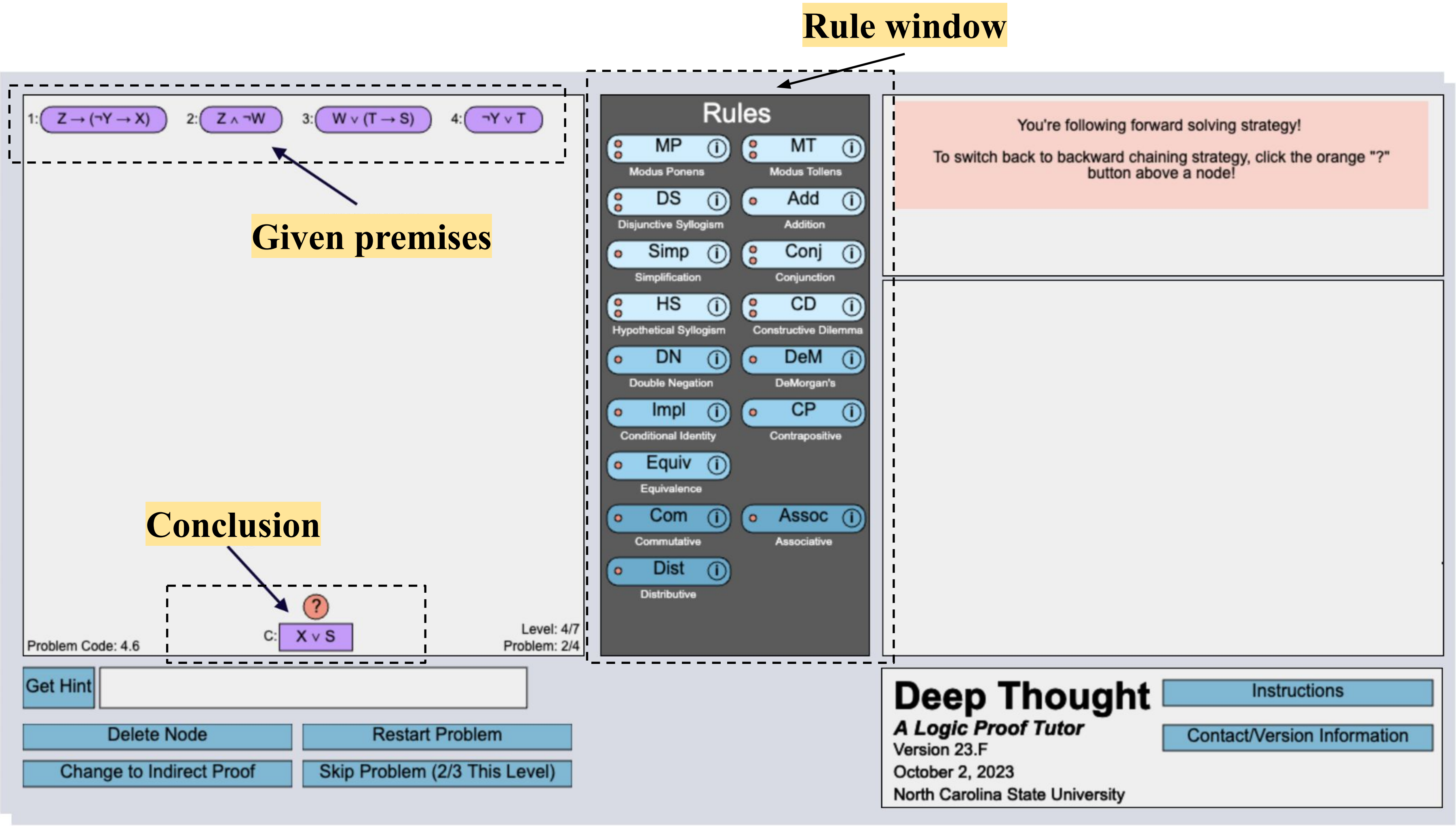
- Parsons problems have shown excellent results for programming [Denny et al. 2008, Zhi et al. 2019, Weinman et al. 2021]
- But have not been shown to significantly improve learning and student performance for logic proofs when provided randomly [Shabrina et al. 2023]

We hypothesize that providing Parsons problem adaptively along with Problem solving, and Worked examples could improve student learning

Problem Statement

Develop a policy to adaptively **determine when to give students what type of problems** in a logic tutor from problem solving (PS), worked example (WE), and Parsons problem (PP)

Context: Deep Thought, The Intelligent Logic Tutor

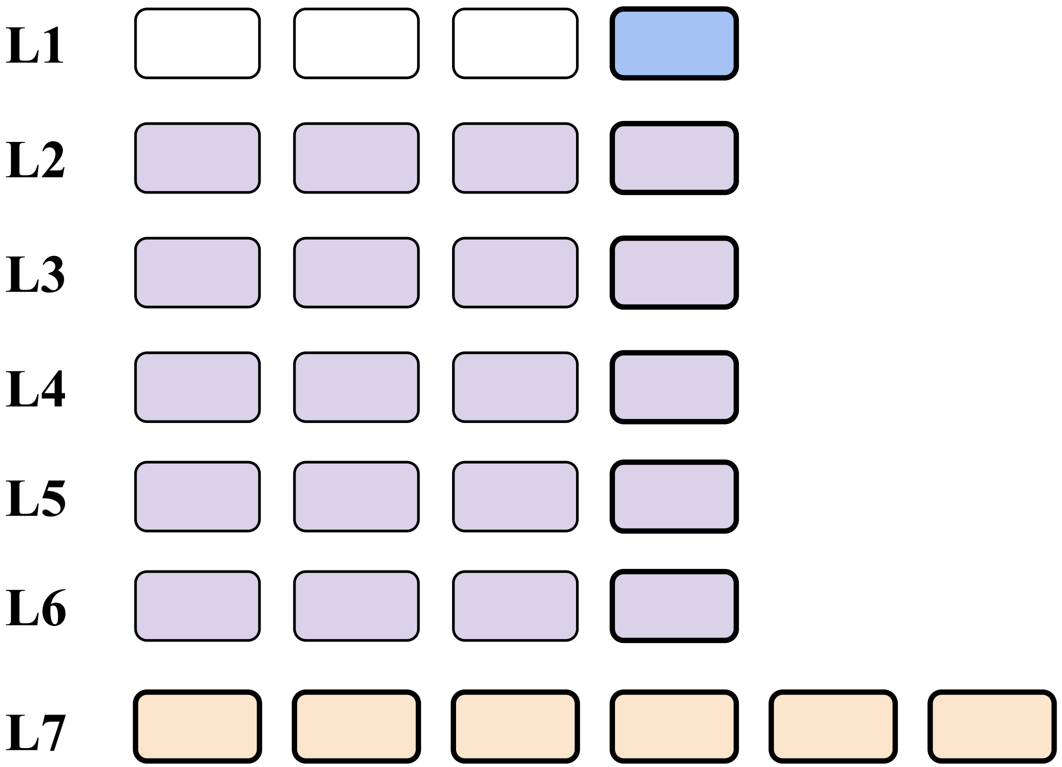


Full Interface of Deep Thought with Student Workspace (left), Rules (middle), Instructions (top-right)

Problem Organization

7 levels

- Level 1
  - 3 Intro problems Pretest
- Level 2-6
  - 3 training problems
  - One level-end test problem
- Level 7
  - 6 posttest problems



Problems in Different Levels

Method – DRL Pedagogical Model

- A Deep Reinforcement learning (DRL) based policy to determine when to provide what type of problem to students
  - Off policy and offline Double Deep Q-Networks (DDQN) model
    - State: 75 student log features that describe students' interaction with the tutor
    - Action: At a training problem of the tutor, there are three possible actions: 1) provide a PS problem, 2) provide WE, 3) provide PP
    - Reward: $posttestScore * (1 - problemTime)$
  - Input: problem level student data
  - Output: whether next training problem should be PS, WE, or PP

Research Questions

- RQ1: How effective is the adaptive DRL policy compared to a non-adaptive expert policy that also selects among problem solving, worked examples, and Parsons problems?
- RQ2: How efficient is the adaptive DRL policy compared to an all-problem solving policy for training problem type selection?
- RQ3: How do providing Parsons Problems impact student performance and learning in logic?

Experimental Design

- **Intelligent Tutor:** Deep Thought logic tutor
- **Deployment:** Fall 2024 Discrete Mathematics course

	Condition	# of students
1	Proposed adaptive DRL policy that provides PS, WE, or PP	63
2	Non-adaptive expert policy that provides PS, WE, or PP	24
3	Control condition: All PS	29

Adaptive vs non-Adaptive policy: Similar time and performance

- RQ1: How **effective** is the adaptive DRL policy **compared to a non-adaptive expert policy** that also selects among problem solving, worked examples, and Parsons problems?

Time

Section	DRL (n=62)	Expert (n=24)	*p*
Training	69.9 (37.2)	68.8 (34.7)	0.98
Level-end post-test	99.1 (83.9)	111.7 (57.5)	0.13
Final post-test	69.9 (71.2)	81.3 (51.9)	0.14
Total tutor	295.4 (154.3)	310.6 (119.8)	0.26

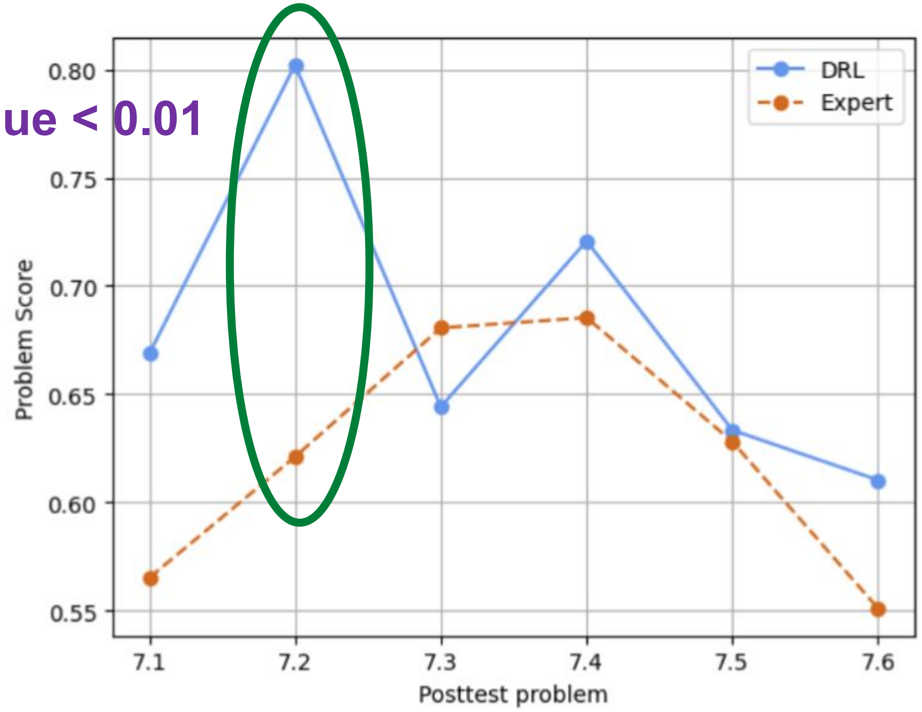
Performance

Metric	DRL	Expert	*p*
Pretest score	0.684	0.678	0.686
post-test score	0.679	0.622	0.137
NLG	-0.068	-0.166	0.351
LE	0.176	0.152	0.166

Performance on each posttest problem

- For problem 7.2, Adaptive DRL group outperformed the non-Adaptive group

P-value < 0.01



Less posttest time for lower proficiency learners

- RQ1: How **effective** is the adaptive DRL policy compared to a non-adaptive expert policy that also selects among problem solving, worked examples, and Parsons problems?

Time Comparison

	High Pretest			Low Pretest		
Section	DRL	Expert	*p*	DRL	Expert	*p*
Training	69.4 (38.7)	60.9 (21.7)	0.71	70.5 (35.5)	76.6 (42.6)	0.74
Level-end post-test	70.5 (52.2)	99.86 (57.8)	0.13	127.7 (98.5)	123.6 (54.7)	0.71
Final post-test	58.2 (53.5)	62.2 (50.7)	0.99	81.5 (83.7)	100.3 (45.8)	**0.08**
Total tutor	230.3 (102.1)	255.9 (118.9)	0.41	360.6 (169.5)	365.0 (92.6)	0.67

Doesn’t impact performance metrics

- RQ1: How **effective** is the adaptive DRL policy compared to a non-adaptive expert policy that also selects among problem solving, worked examples, and Parsons problems?

Performance Comparison

	High Pretest			Low Pretest		
Metric	DRL	Expert	*p*	DRL	Expert	*p*
Pretest score	0.819	0.823	0.96	0.548	0.532	0.24
post-test score	0.709	0.678	0.489	0.650	0.565	0.15
NLG	-0.277	-0.378	0.727	0.134	0.028	0.17
LE	0.181	0.168	0.402	0.171	0.137	0.60

Adaptive vs all-PS – No difference in training time

- RQ2: How **efficient** is the adaptive DRL policy **compared to an all-problem solving policy** for training problem type selection?

Time Comparison

Section	DRL	all-PS	p
Training	69.9 (37.2)	77.0 (37.9)	0.32
Level-end post-test	99.1 (83.9)	96.0 (58.9)	0.49
Final post-test	69.9 (71.2)	66.5 (68.7)	0.99
Total tutor	295.4 (154.3)	297.7 (130.4)	0.49

Adaptive vs all-PS – No difference in training time

- Why no significant difference in training time?
  - The Parsons problems instruction and explanation required considerable reading
  - This reading time may have made the problems more time consuming for students, resulting in longer tutor time

Within Adaptive condition: Better NLG for Learners who got more PPs

- RQ3: How do providing Parsons Problems impact student performance and learning in logic?

Performance comparison in high and low Parsons Problem (PP) group in the adaptive condition

Metric	DRL (high PP)	DRL (low PP)	*p*
Pretest score	0.659	0.715	0.22
post-test score	0.698	0.657	0.28
NLG	0.018	-0.17586	**0.05**
LE	0.195	0.153	0.16

High vs low-PP – no difference in time

- RQ3: How do providing Parsons Problems impact student performance and learning in logic?

Time comparison in high and low PP group in the adaptive condition

Section	DRL (high PP)	DRL (low PP)	p
Training	69.7 (30.0)	70.3 (44.7)	0.460
Level-end post-test	92.1 (91.4)	108.2 (71.9)	0.228
Final post-test	62.9 (48.5)	78.9 (91.9)	0.820
Total tutor	287.0 (159.5)	306.4 (146.7)	0.452

Summary of findings

- The adaptive DRL policy performed **significantly better in one of the posttest problems** compared to the expert policy
- The adaptive DRL policy led to **marginally better posttest time** for the low prior proficiency group
- Overall, the findings show that an adaptive DRL-based policy can be used to adaptively integrate Parsons Problems (PP) with problem solving (PS) and worked examples (WE) without increasing tutor time or decreasing learning

Contact about the paper:

- Nazia Alam (nalam2@ncsu.edu)
- Department of Computer Science
- North Carolina State University

Acknowledgement:
This research was supported by NSF Grants #1726550 and #2013502.

Connect with the presenter:

Xiaoyi Tian
Research Scientist
NC State University

Website: txiaoyi.com
Linkedin: linkedin.com/in/xiaoyi-tian/

Additional Slides

Problem Type: Problem-solving (PS)

➔ Clicking one or two existing statements or nodes, a rule button, and **entering the new derived statement**

Once a step is verified by the tutor, the new node appears.

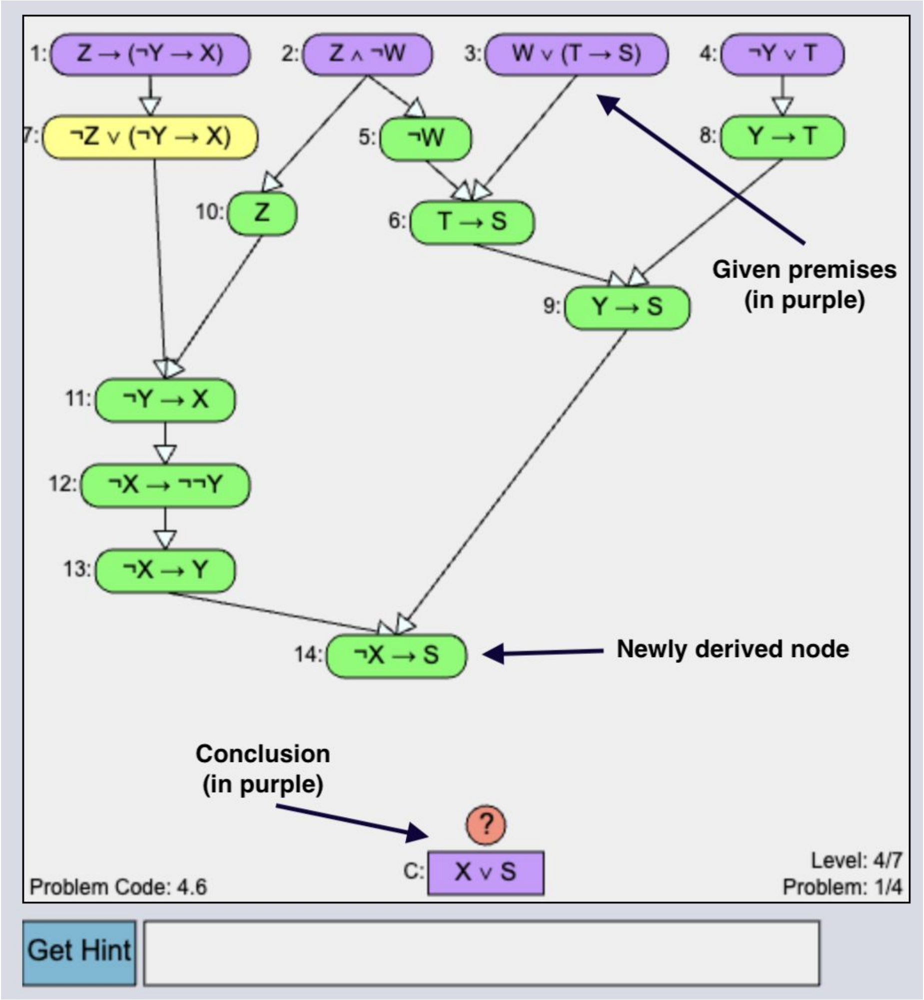


Figure 3: PS Interface

Problem Type: Worked Example (WE)

➔ The tutor shows one step at a time, consisting of adding a new node to the screen with its justification

Students press Next/Previous to progress between steps.

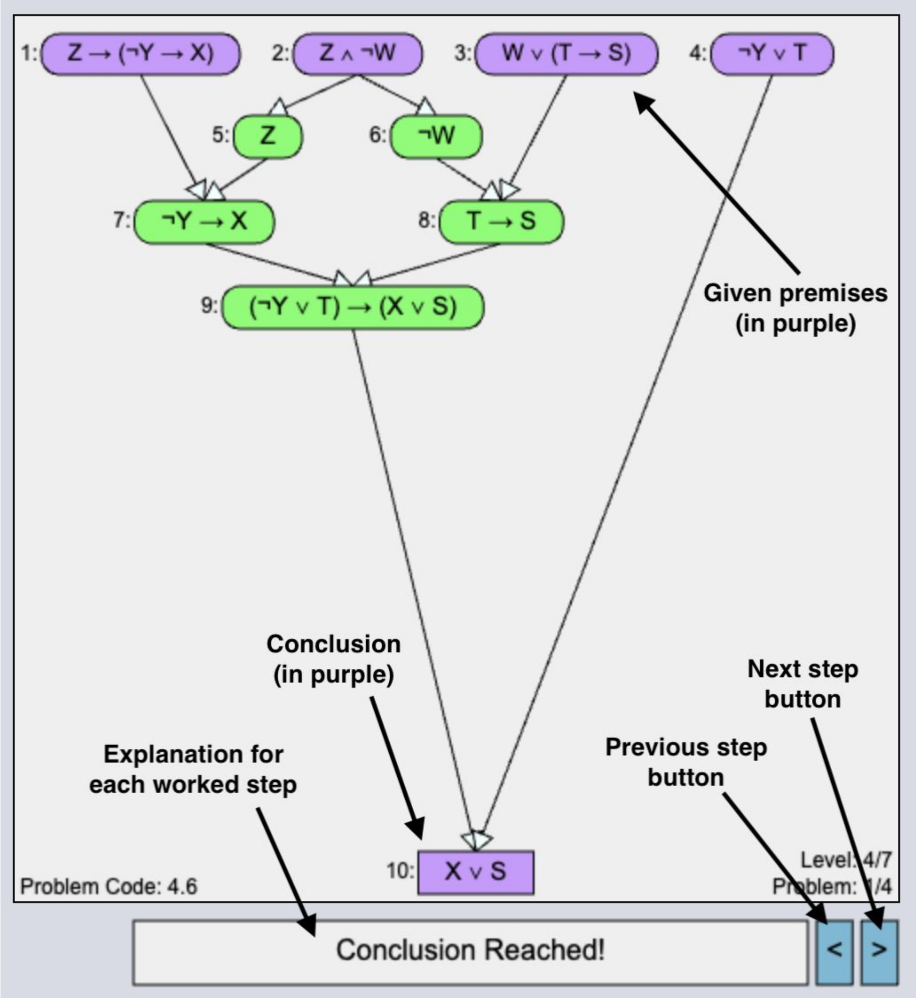


Figure 4: WE Interface

Problem Type: Guided Parsons Problem (GPP)

- Each GPP provides students with **all the statement nodes needed** to complete a proof.

  Students must **add a few justifications to connect all the nodes** to one another with missing edges for rules.

  GPPs **guide students** to justify each unjustified node by **specifying the rule** used to derive it.