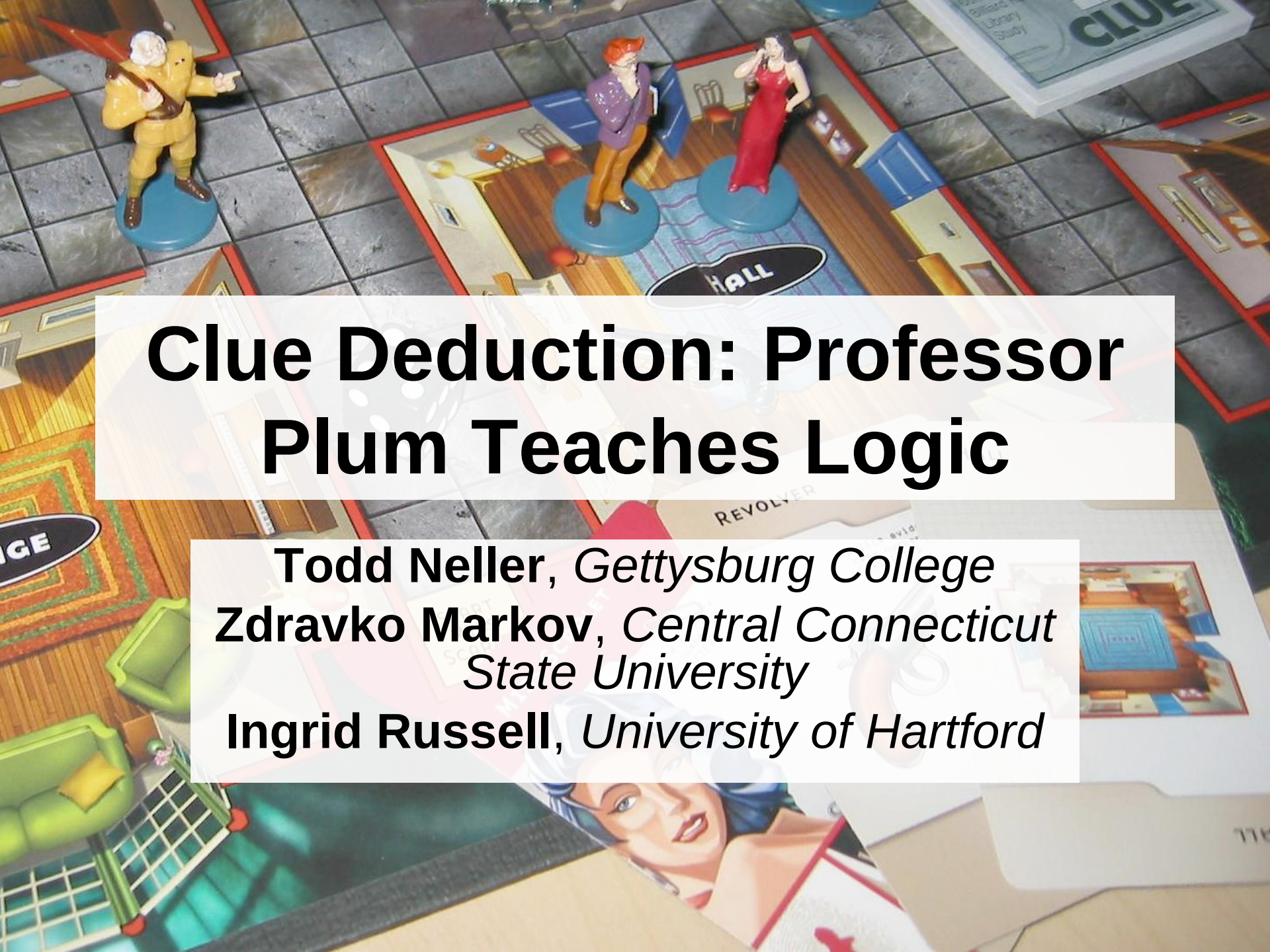

Clue Deduction: Professor Plum Teaches Logic

Todd Neller, *Gettysburg College*
Zdravko Markov, *Central Connecticut State University*
Ingrid Russell, *University of Hartford*

So Many Topics, So Little Time

- Artificial Intelligence
  - the really interesting miscellaneous pile of CS
- Common goals, many diverse techniques
- You can't teach all of Russell & Norvig in a semester.
- Pick and choose, minimalist approach
- *What's a good, minimalist, fun, memorable way to teach concepts of logic?*

KISS Principle + Pop Culture Fun

- KISS – “Keep It Simple, Stupid”
  - With a limited curricular footprint, minimize to the simplest material with maximal benefit.
  - Propositional logic
- Choose a fun reasoning domain.
  - Clue®, a deductive murder mystery game
  - Top-selling board game for almost 60 years
  - Fun, familiar, nostalgic, and *underestimated*


HALL
START
MISS
SCARL
REVOLVER
HALL
MISS SCARLET

The Game of Clue

- 21 cards: 6 suspects, 6 weapons, 9 rooms
- Case file has unknown, random suspect, weapon, and room (SWR)
- Remaining cards dealt to players
- Player *suggests* SWR, first player clockwise that can refute, must show card
- Each player can make 1 SWR *accusation*
- Correct → win; incorrect → lose (& refute)

More Than Child's Play

- Children mark off each dealt/shown card
- Small notepad + instruction = trivial play
- However, consider this:
  - You know player **A** has 3 cards, two of which are **t** and **u**.
  - Player **A** refuted player **B**'s and **C**'s suggestions of (**v**,**w**,**x**) and (**x**,**y**,**z**), respectively, by showing a card.
  - Therefore, … **A**'s 3rd card must be **x**.

Constraint Satisfaction

- Clue reasoning is constraint satisfaction.
- One formulation: Boolean variables c_p denoting “Card *c* is in place *p*.”
- Given CNF representation of Boolean constraints, reason with SAT solver refutations

Clue Deduction Project

- “Clue Deduction: an introduction to satisfiability reasoning”
  - Website, 28 page project guide, starter code
- Guide outline:
  - Introduction to propositional logic
  - Use of SAT solvers
  - Implementation of expert Clue reasoner

Basic Logic Concepts

- Concepts: sentences, operators, literals, truth assignments, (un)satisfiability, models, validity, tautologies, entailment, logical equivalence, derivation, soundness, completeness, …
- That's covering a *lot* without FOL.
- Simple, minimalist, high-utility approach
  - Con: No predicates, unification, FOL generalizations.
  - Pro: Time-efficient, experiential learning.

Applying Logic Concepts

- Collection of favorite propositional logic problems (e.g. “Amy says, ‘Bob is a liar.’ Bob says…”)
- Representation
- Conversion to CNF
- Resolution TP
- Automatic TP through use of a …

(1)	$\{\neg A, \neg B\}$		Knowledge base
(2)	$\{B, A\}$		
(3)	$\{\neg B, \neg C\}$		
(4)	$\{C, B\}$		
(5)	$\{\neg C, \neg A\}$		
(6)	$\{\neg C, \neg B\}$		
(7)	$\{A, B, C\}$		
(8)	$\{C\}$		Assumed negation
(9)	$\{\neg A\}$	(5),(8)	Derived clauses
(10)	$\{B\}$	(2),(9)	
(11)	$\{\neg C\}$	(3),(10)	
(12)	$\{\}$	(8),(11)	*Contradiction!*

SAT Solver Black Box

- Simple SATSolver interface to underlying black box (e.g. zchaff, SAT4J, etc.)
- Alternatively, students may write their own solver (e.g. simple DPLL, WalkSAT)
- Reductio ad absurdum (proof by contradiction)

ClueReasoner

- Some helper functions provided, e.g.
  - Conversion of (card, place) pair to Gödel number for atomic sentence c_p
- Students convert Clue game facts to CNF and add them to SATSolver KB, e.g.
  - A card cannot be in two places.
  - At least one suspect card is in the case file.
  - Player **A** refuted suggestion (**x**,**y**,**z**).

ClueReasoner

```
ClueReasoner cr = new ClueReasoner();
String[] myCards = {"wh", "li", "st"};
cr.hand("sc", myCards);
cr.suggest("sc", "sc", "ro", "lo", "mu", "sc");
cr.suggest("mu", "pe", "pi", "di", "pe", null);
cr.suggest("wh", "mu", "re", "ba", "pe", null);
cr.suggest("gr", "wh", "kn", "ba", "pl", null);
cr.suggest("pe", "gr", "ca", "di", "wh", null);
cr.suggest("pl", "wh", "wr", "st", "sc", "wh");
cr.suggest("sc", "pl", "ro", "co", "mu", "pl");
cr.suggest("mu", "pe", "ro", "ba", "wh", null);
cr.suggest("wh", "mu", "ca", "st", "gr", null);
cr.printNotepad();
```

	sc	mu	wh	gr	pe	pl	cf
mu	n	-	-	n	-	-	-
pl	n	Y	n	n	n	n	n
gr	n	n	Y	n	n	n	n
pe	n	-	n	n	-	-	-
sc	n	Y	n	n	n	n	n
wh	Y	n	n	n	n	n	n
kn	n	-	-	-	n	-	-
ca	n	n	n	Y	n	n	n
re	n	-	-	n	-	-	-
ro	n	-	-	-	-	-	-
pi	n	-	n	n	-	-	-
wr	n	-	-	-	-	-	-
ha	n	-	-	-	-	-	-
lo	n	-	-	-	-	-	-
di	n	n	n	n	-	n	-
ki	n	-	-	-	-	-	-
ba	n	-	-	n	n	-	-
co	n	-	-	-	-	-	-
bi	n	-	-	-	-	-	-
li	Y	n	n	n	n	n	n
st	Y	n	n	n	n	n	n

Exploring Further

- One constraint omitted:
  - Player *p* has exactly *n* cards.
- Pseudo-Boolean constraints
- DIY SAT Solver
  - DPLL
  - WalkSAT, Novelty, etc.
- Many starting points for deeper study of Constraint Satisfaction, KR&R topics

MLExAI

http://uhaweb.hartford.edu - Machine Learning Laboratory Experiences for Introducing Stud...

File Edit View Favorites Tools Help Links

Machine Learning Laboratory Experiences for Introducing Undergraduates to Artificial Intelligence

NSF DUE CCLI-A&I Award Number 0409497

University of Hartford · CCSU · Gettysburg College

Links

- Home
- Overview
- Faculty PIs
- Advisory Board
- Sample Projects
- Affiliate Faculty
- Students
- Machine Learning
- ML Resources
- Publications
- Press
- Grants
- Related Projects

Home

Project Goal

The project goal is to develop a framework for teaching core AI topics using a unifying theme of machine learning. A suite of adaptable, hands-on laboratory projects will be developed that can be closely integrated into a one-semester AI course. The project will enhance the student learning experience in the introductory Artificial Intelligence course by:

1. Introducing machine learning elements into the AI course,
2. Implementing a set of unifying machine learning laboratory projects to tie together the core AI topics, and
3. Developing, applying, and testing an adaptable framework for the presentation of core AI topics which emphasizes the important relationship between AI and computer science in general, and software development in

What does this have to do with Machine Learning?

- Tom Mitchell, *Machine Learning*:
  - "A computer is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P*, if its performance at tasks in *T*, as measured by *P*, improves with experience *E*."
- Deductive learning
  - T = reasoning, P = computation time, E = adding deduced knowledge to KB

Resources

- Clue Deduction: http://cs.gettysburg.edu/~tneller/nsf/clue/
- MLExAI: http://uhaweb.hartford.edu/compsci/ccli/
- This work was sponsored by NSF DUE CCLI-A&I Award Number 0409497.

The Cal-Clue-ator