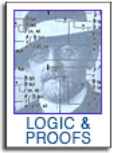
Strategic Proof Tutoring in Logic

Douglas Perkins

Department of Philosophy
Carnegie Mellon University
douglasperkins@cmu.edu

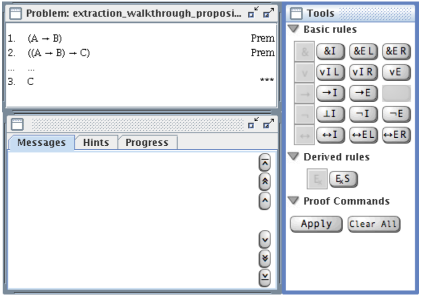
June 29, 2007

The Logic and Proofs course



- Logic and Proofs is a mostly-online introduction to classical propositional and first-order logic.
- The course text is online, and much of the homework can be done online as well.
- There are required homework problems along with optional problems that illustrate concepts.
- Four weeks are spent on proofs in propositonal logic; another four are spent on first-order logic.
- `http://www.cmu.edu/oli/courses/enter_logic.html`

The Carnegie Proof Lab



Problem: extraction_walkthrough_proposi...

1.	(A → B)	Prem
2.	((A → B) → C)	Prem
...	...	
3.	C	***

- The Carnegie Proof Lab is an online proof construction environment.
- Students are given the partial proof and inference rules. They can easily apply the inference rules; error messages are displayed on invalid use.
- Extraneous details are handled automatically.

Student proof construction woes

"It is generally useless to carry out details without having seen the main connection, or having made a sort of plan." – G. Polya

- $$\begin{array}{r|l} \vdots & \vdots \\ 1 & P \lor \neg P \qquad *** \end{array}$$
- At this point in the proof, the student has several choices: Disjunction Introduction Left, Disjunction Introduction Right, or Negation Elimination. The best move may be unclear.
- Other problems of a similar character include Peirce's Law, $((P \to Q) \to P) \to P$, and DeMorgan's Laws.

Two questions

Questions:

1. Given a partial proof, what inference rules should the student use next?
2. If the computer can figure out what inference rules to use next, how can this information be imparted to the student in a sensible way?

Two questions

Questions:

1. Given a partial proof, what inference rules should the student use next?
2. If the computer can figure out what inference rules to use next, how can this information be imparted to the student in a sensible way?

Answers:

1. Strategic proof search.

Two questions

Questions:

1. Given a partial proof, what inference rules should the student use next?
2. If the computer can figure out what inference rules to use next, how can this information be imparted to the student in a sensible way?

Answers:

1. Strategic proof search.
2. Computer-generated tutoring with sound pedagogical underpinnings.

Outline

1. The problem for students
   - The Logic and Proofs course
   - The Carnegie Proof Lab
   - Proof construction woes

2. Formal logic in the course
   - Natural Deduction
   - Strategic proof search

3. Proof tutoring
   - The Explanation Tutor
   - The Walkthrough Tutor
   - The Completion Tutor

4. Putting it together

Natural deduction

-

1	$A\&(B \rightarrow C)$	Premise
⋮	⋮	
2	C	***

- Natural deduction proofs in Fitch notation.
- The premises are at the top, and the goal is at the bottom.
- Inference rules are used to bridge the gap from the premises to the goal.

Inference rules

- Elimination rules work from premises and assumptions forwards: &EL, &ER, $\vee$E, $\rightarrow$E, $\leftrightarrow$EL, $\leftrightarrow$ER.
- Introduction rules work from goals backwards: &I, $\vee$IR, $\vee$IL, $\rightarrow$I, $\leftrightarrow$I, $\neg$I.
- Special cases:
  - $\perp$I is an elimination rule.
  - $\neg$E is an introduction rule.
  - $\vee$E takes a premise and a goal.

Normal proofs

	1	$A \& B$	Premise
	2	C	Premise
●	3	$(A \& B) \& C$	&I, 1, 2
	4	$A \& B$	&EL, 3
	5	A	&EL, 4

- This proof has extra work in it – lines 3 and 4 are unnecessary.
- Normal proofs are ones where no formula occurrence is the conclusion of an introduction rule and the major premise of an elimination rule.
- Non-normal proofs contain detours that do not contribute to the proof and consequently should be avoided.

Strategic proof search

- Strategic proof search is an algorithm for determining what sequence of inference rules to apply when. Given a partial proof, the student is pursuing a particular goal with several premises or assumptions available.
- The strategy is explainable in simple terms.
- The strategy is (mostly) deterministic, for the student.
- The strategy produces normal proofs.
- See "Normal Natural Deduction Proofs in Classical Logic", Sieg & Byrnes, 1998.

Tactic 1: Extraction

- Use a sequence of elimination rules from a premise or assumption to a goal.
- The goal must be strictly positively embedded in the premise or assumption.
-

1	$A \& (B \rightarrow C)$	Premise
⋮	⋮	
2	C	***

$\Longrightarrow$

1	$A \& (B \rightarrow C)$	Premise
2	$B \rightarrow C$	&ER, 1
⋮	⋮	
3	B	***
4	C	→E, 2, 3

Tactic 2: Inversion

- Use an introduction rule on a complex goal to break it into smaller subgoals.
- $$\begin{array}{r|ll} \vdots & \vdots & \\ 1 & A \,\&\, B & *** \end{array} \Longrightarrow \begin{array}{r|ll} \vdots & \vdots & \\ 1 & A & *** \\ \vdots & \vdots & \\ 2 & B & *** \\ 3 & A \,\&\, B & \&\mathrm{I},\ 1,\ 2 \end{array}$$

Tactic 3: Cases

- Use Disjunction Elimination on a disjunction premise and a goal.
- For the rule's premise, consider any disjunction that is a premise, assumption, or strictly positively embedded in a premise or assumption.
-

1	$A \vee B$	Premise
$\vdots$	$\vdots$	
2	C	***

$\Longrightarrow$

1	$A \vee B$	Premise
2	A	Assume
$\vdots$	$\vdots$	
3	C	***
4	B	Assume
$\vdots$	$\vdots$	
5	C	***
6	C	$\vee$E, 1, 3, 5

Tactic 4: Refutation

- Use Negation Elimination (classical negation) on a goal.

-

1	$A \& \neg A$	Premise
$\vdots$	$\vdots$	
2	B	***

$\Longrightarrow$

1	$A \& \neg A$		Premise
2		$\neg B$	Assume
$\vdots$		$\vdots$	
3		$\bot$	***
4	B		¬E, 3

The strategy spelled out

- For each goal, try each of the following.
  1. Extraction.
  2. Inversion.
  3. Cases.
  4. Refutation.
- If all tactics have been exhausted for a particular subgoal, backtrack.
- If all tactics have been exhausted for the starting goal, the problem is not provable.

An example of a strategic search

	1	$(A \& B) \& C$	Premise
1	⋮	⋮	
	2	$B \& C$	***

An example of a strategic search

1.

1	$(A \& B) \& C$	Premise
⋮	⋮	
2	$B \& C$	***

2.

1	$(A \& B) \& C$	Premise
⋮	⋮	
2	B	***
⋮	⋮	
3	C	***
4	$B \& C$	&I, 2, 3

An example of a strategic search

1

1	$(A \& B) \& C$	Premise
⋮	⋮	
2	$B \& C$	***

2

1	$(A \& B) \& C$	Premise
⋮	⋮	
2	B	***
⋮	⋮	
3	C	***
4	$B \& C$	&I, 2, 3

3

1	$(A \& B) \& C$	Premise
2	$(A \& B)$	&EL, 1
3	B	&EL, 2
⋮	⋮	
4	C	***
5	$B \& C$	&I, 2, 3

An example of a strategic search

1

1	$(A \& B) \& C$	Premise
⋮	⋮	
2	$B \& C$	***

2

1	$(A \& B) \& C$	Premise
⋮	⋮	
2	B	***
⋮	⋮	
3	C	***
4	$B \& C$	&I, 2, 3

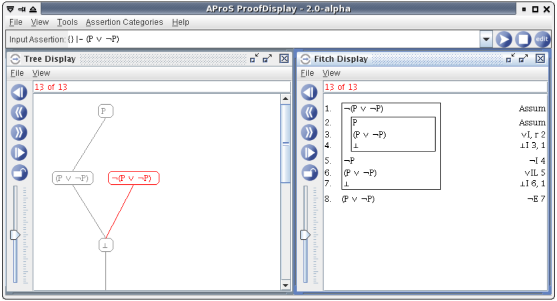
3

1	$(A \& B) \& C$	Premise
2	$(A \& B)$	&EL, 1
3	B	&EL, 2
⋮	⋮	
4	C	***
5	$B \& C$	&I, 2, 3

4

1	$(A \& B) \& C$	Premise
2	$(A \& B)$	&EL, 1
3	B	&ER, 2
4	C	&ER, 1
5	$B \& C$	&I, 2, 3

AProS: Automated Proof Search



Input Assertion: {} |- (P ∨ ¬P)

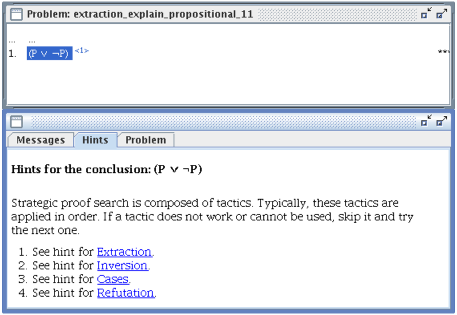
```
1.  ¬(P ∨ ¬P)             Assum
2.     P                  Assum
3.     (P ∨ ¬P)           ∨I, r 2
4.     ⊥                  ⊥I 3, 1
5.  ¬P                    ¬I 4
6.  (P ∨ ¬P)              ∨IL 5
7.  ⊥                     ⊥I 6, 1
8.  (P ∨ ¬P)              ¬E 7
```

- AProS is an automated theorem prover that can operate in classical first-order and propositional logic.
- It has a graphical interface, but it can also operate as a library.
- For propositional logic, AProS uses the strategic proof search algorithm just described. For first-order it uses a suitable extension.

Outline

1. The problem for students
   - The Logic and Proofs course
   - The Carnegie Proof Lab
   - Proof construction woes
2. Formal logic in the course
   - Natural Deduction
   - Strategic proof search
3. Proof tutoring
   - The Explanation Tutor
   - The Walkthrough Tutor
   - The Completion Tutor
4. Putting it together

The Explanation Tutor



1. $(P \vee \neg P)$ <1> ***

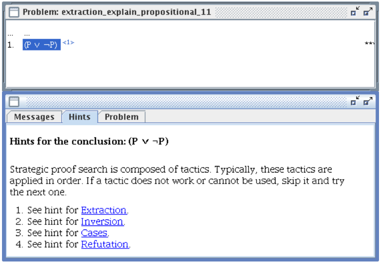
Hints for the conclusion: $(P \vee \neg P)$

Strategic proof search is composed of tactics. Typically, these tactics are applied in order. If a tactic does not work or cannot be used, skip it and try the next one.

1. See hint for Extraction.
2. See hint for Inversion.
3. See hint for Cases.
4. See hint for Refutation.

- For the explanation tutor, the student may select any open subgoal.
- The tactics are listed, and the student may examine each tactic for details on its use in the current setting.
- “The student has to learn to associate [the basic declarative facts in a domain] with problem solving goal structures.” Corbett and Koedinger, 1997.

Explanation Tutor details



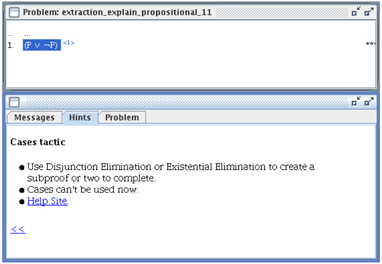
Problem: extraction_explain_propositional_11

... ...
1. $(P \vee \neg P)$ <1>

Hints for the conclusion: $(P \vee \neg P)$

Strategic proof search is composed of tactics. Typically, these tactics are applied in order. If a tactic does not work or cannot be used, skip it and try the next one.

1. See hint for Extraction.
2. See hint for Inversion.
3. See hint for Cases.
4. See hint for Refutation.



Problem: extraction_explain_propositional_11

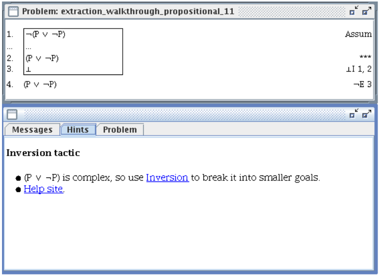
... ...
1. $(P \vee \neg P)$ <1>

Cases tactic

- Use Disjunction Elimination or Existential Elimination to create a subproof or two to complete.
- Cases can't be used now.
- Help Site.

<<

The Walkthrough Tutor



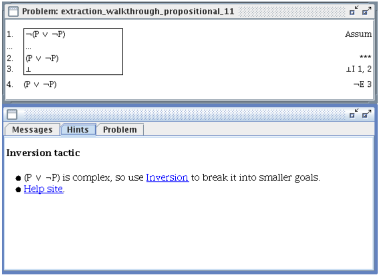
```
1.   ¬(P ∨ ¬P)             Assum
...  ...
2.   (P ∨ ¬P)                ***
3.   ⊥                   ⊥I 1, 2
4.   (P ∨ ¬P)                ¬E 3
```

Inversion tactic

- (P ∨ ¬P) is complex, so use Inversion to break it into smaller goals.
- Help site.

- Students are given hints for each move, initially in a general form and optionally with more detail.
- When students use the recommended rule, the hint advances.
- When students use a different rule, the hints stop until the student returns to the recommended rule.
- See Towne and Munro, 1992, on directed step by step performance, and Chi and Bassok, 1989, on the value of examples.

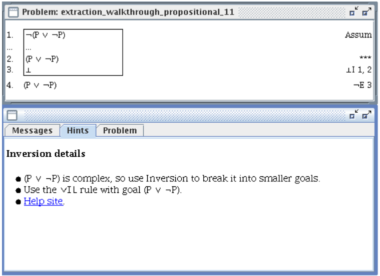
Walkthrough Tutor details



1.	¬(P ∨ ¬P)	Assum
...	...	
2.	(P ∨ ¬P)	***
3.	⊥	⊥I 1, 2
4.	(P ∨ ¬P)	¬E 3

Inversion tactic

- (P ∨ ¬P) is complex, so use Inversion to break it into smaller goals.
- Help site.

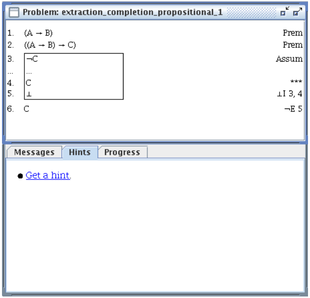


1.	¬(P ∨ ¬P)	Assum
...	...	
2.	(P ∨ ¬P)	***
3.	⊥	⊥I 1, 2
4.	(P ∨ ¬P)	¬E 3

Inversion details

- (P ∨ ¬P) is complex, so use Inversion to break it into smaller goals.
- Use the ∨I L rule with goal (P ∨ ¬P).
- Help site.

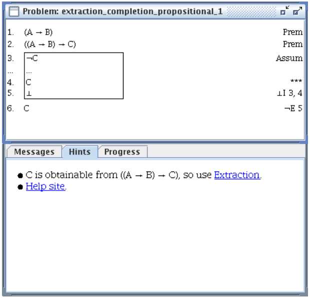
The Completion Tutor



```
1.   (A → B)                 Prem
2.   ((A → B) → C)           Prem
3.   │¬C                     Assum
...  │...
4.   │C                      ***
5.   │⊥                      ⊥I 3, 4
6.   C                       ¬E 5
```

- Sometimes a student has a non-canonical partial proof.
- When reasonable, the student's work should be retained.
- "Facilitate successive approximations to the target skill." Anderson et al., 1995.

Completion Tutor details



1.	$(A \to B)$	Prem
2.	$((A \to B) \to C)$	Prem
3.	$\neg C$	Assum
...	...	
4.	C	***
5.	$\bot$	$\bot$I 3, 4
6.	C	$\neg$E 5

- C is obtainable from $((A \to B) \to C)$, so use Extraction.
- Help site.

- If the student's partial proof is normal, the tutor works from that if reasonable.
- If the partial proof is not completable, some backtracking must be done.
- Currently, backtracking is done to the point of deviation from the canonical proof.

Outline

1. The problem for students
   - The Logic and Proofs course
   - The Carnegie Proof Lab
   - Proof construction woes

2. Formal logic in the course
   - Natural Deduction
   - Strategic proof search

3. Proof tutoring
   - The Explanation Tutor
   - The Walkthrough Tutor
   - The Completion Tutor

4. Putting it together

Situating the Explanation Tutor

- The Explanation Tutor doesn't tell students what tactic ought to be pursued, so enabling it in homework problems is less prone to abuse.
- The tutor is placed using non-technical language in parallel with the Completion Tutor.
- A few weeks into proof construction, the Explanation Tutor uses more precise language to help students learn the tactics with greater precision.
- It has been extended to first-order logic.

Situating the Walkthrough Tutor

- When there are multiple proofs possible, the Walkthrough Tutor produces hints for just one, but the student may want to pursue another.
- Its explanations minimize technical vocabulary and are generally accessible.
- Thus, the Walkthrough Tutor is ideal for use starting when students first learn to construct proofs, particularly in problems chosen ahead of time.

Situating the Completion Tutor

- The Completion Tutor has wording like the Walkthrough Tutor, so it should be available around the same time in the course.
- It is less restrictive, as it provides on-demand hints. As students get better at proving things, they will not need to use it as much.
- Thus, the Completion Tutor should be used in some example problems as well as some assigned exercises.

Putting it all together

- Students have difficulty constructing non-trivial proofs.
- The Walkthrough Tutor gives examples of desired performance.
- The Completion Tutor allows students to complete proofs they've started.
- The Explanation Tutor is used to remind students of the tactics while leaving it to the students to put it together.
- We provide scaffolding, starting with the Walkthrough Tutor and decreasing. The student uses the Completion Tutor and Explanation Tutor to solve problems, and with experience, tutor use should decrease.

Where to go from here

- The tutors described here show a comprehensive way to use an automated theorem prover to dynamically produce hints for students in proof search.
- Just how good is this tutor? While it can provide useful hints to students, what will its effect be on the first few weeks of with proofs? What components of it are the most useful?
- These questions may be addressed this winter, using logging data from the fall term.