Oliver: an OnLine Inference and VERification system

- Dr. Christelle Scharff, Pace University, NY, USA
- Dr. Andrew Wildenberg, SUNY Stony Brook, NY, USA
Outline

- SUNY Stony Brook Curriculum presentation
- Discrete Structures I
- Propositional logic
- Pedagogical goals
- OLIVER
- WeBWorK for delivering Online homeworks for Discrete Structures
- Students Feedbacks
- Future directions
SUNY SB CS curriculum

- ACM standards for CS education and Discrete Structures

- Sequence of introductory courses of CS at SUNY Stony Brook

  Discrete Structures I (around 1200 students each year until Spring 2002)
  ↓
  Programming with JAVA
  ↓
  Discrete Structures II
  ⇧
  Data Structures with JAVA
Discrete Structures I

- Propositional Logic
- Number Theory
- Set Theory
- Functions
- Recursion
- Definition of functions in SML
- Induction
- Correctness of functions – illustrated through the use of SML and induction
- Automata
- Trees
- Graphs
Propositional Logic

- **Propositional Logic:**
  - **Formulas** are **Propositions**.
    - Propositional variables ($p, q, r...$), conjunctions ($\land$), disjunctions ($\lor$), conditionals ($\rightarrow$), biconditionals ($\leftrightarrow$) and negations ($\neg$) of propositions.
    - **Examples:** $p \land q$, $(p \lor q) \rightarrow q$
  - We consider **interactive direct proofs** based on an **inference system**.

- Cognitive psychologists estimate that only 1 to 4% of the population are able without explicit training to correctly apply the principles of formal logic.

- Visualization in assisting the reasoning process through the use of software and web-based tools can help students learn and enjoy logic [Barwise, Etchemendy, 1993, Bornat, Sufrin, 1999, Abraham and all, 2001].
Pedagogical goals


- Convince students that logic is relevant to their career.
  - Students have no programming experience (60%).

- Convince students that logic is accessible to them and can be enjoyable.

- Encourage experimentation on formal concepts.

- Encourage study groups.

- Instant feedback.

- Fast grading of proofs.
Oliver

OnLine Inference and VERification system

- Software composed of 2 pieces:
  - a propositional logic engine for checking inferences, and
  - a servlet-based interface allowing to access the system online.

- Oliver is **flexible**.
  Previous interactive systems (as Jape or Logic Tutor) use strict inference systems
  But Oliver accepts any valid direct proof.

- Oliver proofs are **“more natural”**.

- Oliver offers a way to **visualize** each step of the non-deterministic and discovery of the considered proof.

- Oliver provides **instant feedback**.

- Oliver encourages **experimentation**.
• Oliver generates **random problems** to reduce cheating and plagiarism.

• Oliver **grades** the proofs (success or failure.)

• Oliver records all proofs in a **database** for further analysis.

• Oliver has been integrated **WeBWorK** [Pizer, Gage].

• Try Oliver:
  http://www.csis.pace.edu/~scharff/SOFTWARE/OLIVER/oliver.html
The statement \( \neg u \land \neg q \) was added to the proof.

Your goal is to deduce \( p \)

<table>
<thead>
<tr>
<th>line</th>
<th>statement</th>
<th>justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( w \rightarrow s \land \neg q )</td>
<td>premise</td>
</tr>
<tr>
<td>2</td>
<td>( \neg u \land \neg q \rightarrow p )</td>
<td>premise</td>
</tr>
<tr>
<td>3</td>
<td>( t )</td>
<td>premise</td>
</tr>
<tr>
<td>4</td>
<td>( \neg w \rightarrow t )</td>
<td>premise</td>
</tr>
<tr>
<td>5</td>
<td>( \neg u \lor \neg w )</td>
<td>premise</td>
</tr>
<tr>
<td>6</td>
<td>( w )</td>
<td>3,4</td>
</tr>
<tr>
<td>7</td>
<td>( s \land \neg q )</td>
<td>1,6</td>
</tr>
<tr>
<td>8</td>
<td>( \neg u )</td>
<td>5,6</td>
</tr>
<tr>
<td>9</td>
<td>( \neg q )</td>
<td>7</td>
</tr>
<tr>
<td>10</td>
<td>( \neg u \lor \neg q )</td>
<td>8,9</td>
</tr>
<tr>
<td>11</td>
<td>( p )</td>
<td>2,10</td>
</tr>
</tbody>
</table>
**Oliver and WeBWorK**

- **WeBWorK**
  - Developed at the University of Rochester [Pizer, Gage]
  - Web-based system for delivering and evaluating homework problems
  - Features: user authentication, encourage experimentation, instant feedback, instant grading, individualized problem sets, evaluation of free-form symbolic answers, distribution of solutions, development of problems using the PG language...

- **Advantages of the integration of Oliver in WeBWorK:**
  - Experimentation
  - Grading
  - Distribution of solutions
  - Control of plagiarism and cheating.
User feedback and analysis

- More than 2000 students used Oliver.
- 71% said that Oliver was the most useful web-based homework and 52% said logic was their favorite topic.
- Students have a different attitude toward Oliver homework w.r.t standard paper homeworks.
  - Students are driven to get every proof right.
- Proof quality and test scores have improved dramatically with the use of Oliver.
- WeBWorK is very popular among students (22% of students preferred paper-based homeworks.)
Conclusion and future directions

• Use of Oliver in Discrete Structures I.

• Develop new Oliver problems and let students develop their own problems.

• Study of our database of proofs.

• Oliver with inference rule names.

• Better integration of Oliver in WeBWorK:
  – One window.
  – Record incomplete proofs to make it more concordant with WeBWorK.

• Oliver for indirect proofs.

• Use of WeBWorK: an active project.