The Lurch Project
A word processor that checks your math

Past, Present, and Future

Nathan Carter
Bentley University
• Not computational
  *Maple* does math for you.
  Lurch checks your work.

• Not a homework-grading system
  *WeBWorK* checks answers when you submit.
  Lurch checks steps as you work.

• Not a “proof assistant”
  *Coq* helps automate the proving process.
  Lurch is a learning tool.
Removing the Disadvantages

Lurch is a platform that will make it easy for instructors and students to build and use a variety of mathematical experiences all under one hood.
The Web

- **HTML**: A markup language for describing documents with rich content
- **Javascript**: Scripting language that makes HTML pages interactive

Lurch

- **OpenMath**: A markup language for describing mathematics with unambiguous semantics
- **Javascript**: Scripting language that makes Lurch able to validate OpenMath documents
Nathan's Homework 1

1. Depends on misc-utils by Nathan Carter
2. Depends on simple-doc-structure by Nathan Carter
3. Depends on hierarchical...
4. Depends on nd-parser by Nathan Carter...
5. Depends on nd-prop-utils by Nathan Carter...
6. Depends on replacement...
7. Depends on nd-prop-utils by Nathan Carter...

Section 1, work:

1. \( W \rightarrow rB \)
2. \( A \wedge W \)
3. \( B \vee (J \wedge K) \)
4. \( W \) \& E 2
5. \( rB \) \& E 1, 4
6. \( J \wedge K \) \& E 3.5
7. \( K \) \& E 6

Section 2, work:

3. \( P \wedge Q \)
   \[ P \quad \& E 3 \]
   \[ Q \quad \& E 3 \]

All premises for this rule are available in the proof already.

Lurch in 2008
Student Responses

• “It helped me learn how to do proofs through trial and error.”

• “…I liked using Lurch because I was able to see what was needed for rules to work.”

• “It told me if I was right or wrong”

• “It was also easier to move lines around than erasing everything on paper.”
Student Responses

• I used Lurch for experimentation; I tinkered to learn the results of various actions. 4.6
• The constant feedback Lurch provides about my work is valuable. 4.2
• It was helpful that proofs in Lurch looked just like proofs in our textbook. 4.2

Scale: 1=Strongly Disagree, 5=Strongly Agree
Student Responses

• It is possible to do a proof in Lurch by experimental clicking and typing, without thinking.

Scale: 1=Strongly Disagree, 5=Strongly Agree
Bijectivity and Inverse Functions

In lecture I stated the following theorem without proof. Here is an essay-style proof of that result.

**Theorem:** A function has an inverse function if and only if it is bijective.

**Proof:** Let \( f : A \to B \).

(\(\Rightarrow\)) Assume \( f \) has an inverse function \( g \). Then \( g \circ f = \text{id}_A \) and \( f \circ g = \text{id}_B \) by the definition of inverse function.

Let \( x, y \in A \). Assume \( f(x) = f(y) \). Then

\[
\begin{align*}
x &= \text{id}_A(x) & \text{by the def. of identity function} \\
   &= (g \circ f)(x) & \text{by substitution} \\
   &= g(f(x)) & \text{by def. of } \circ \\
   &= g(f(y)) & \text{by substitution} \\
   &= (g \circ f)(y) & \text{by def. of } \circ \\
   &= \text{id}_A(y) & \text{by substitution}
\end{align*}
\]
Here are some example steps of work in a Lurch document. These steps are steps of algebra.

\[ \int xe^x \, dx \]

\[ = \left( \int x \, dx \right) \left( \int e^x \, dx \right) \text{ invalid} \]
This document shows some example work in using the computer algebra system in Lurch, which can validate algebra and simple calculus.

\[
\frac{d}{dx} (\sin e^x) = (\cos e^x) \frac{d}{dx} e^x \\
= (\cos e^x)e^x \\
= \cos e^{2x}
\]

You can also see that when editing a math expression, both a palette (top right) and tooltip (below the input) are provided: \text{sqrt}(x).

Use standard calculator notation, such as 3*x^2, or e^(x/y).
Progress to 2009

• Lurch came with many built-in math topics, not just the two you’ve seen.

• Most involved interactively editing a document in word-processing style.

• But each was built by a programmer.

• So the disadvantage of uncustomizability remained. Each instructor who might want to use Lurch had to use the exact same notation, rules, etc.
## Groups

A group is a set $G$ with a binary operation $\cdot$, a unary operation $^{-1}$, and a particular element $e$ in $G$ (called the identity element) satisfying the following axioms.

For any elements $x, y, z$ in $G$

<table>
<thead>
<tr>
<th>Equation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(xy)z = x(yz)$</td>
<td>Associativity</td>
</tr>
<tr>
<td>$ex = x$</td>
<td>Left Identity</td>
</tr>
<tr>
<td>$xe = x$</td>
<td>Right Identity</td>
</tr>
<tr>
<td>$xx^{-1} = e$</td>
<td>Right Inverse</td>
</tr>
<tr>
<td>$x^{-1}x = e$</td>
<td>Left Inverse</td>
</tr>
</tbody>
</table>

Lurch in 2010
The identity element is its own inverse

Using these axioms, we can prove the following theorem about any group.

**Theorem 1:** $e^{-1} = e$

**Proof:** Using the axioms above, we see that

\[
    e^{-1} = ee^{-1} \quad \text{Left Identity}\n\]

\[
    = e \quad \text{Right Inverse}\n\]

QED

The identity element is unique
Example - Generators and Relations in the Klein four-group

The Klein four-group has two elements, \( a \) and \( b \) called generators, satisfying

\[
\begin{align*}
    aa &= e & \text{Nilpotence of } a \\
    bb &= e & \text{Nilpotence of } b \\
    ab &= ba & \text{Abelian}
\end{align*}
\]

We can use these properties, and the properties of a group defined above, to prove many other words written in the alphabet

\[
\{a, b, a^{-1}, b^{-1}\}
\]

represent the same element of the group.
Theorem 3: In the Klein four-group, \((ab)a = b\).

Proof: We have

\[(ab)a = (ba)a \quad \text{Abelian} \checkmark\]

\[= b(aa) \quad \text{Associativity} \checkmark\]

\[= be \quad \text{Nilpotence of } a \checkmark\]

\[= b \quad \text{Right Identity} \checkmark\]

as desired.

QED

Exercise: Prove that \(ab\) is its own inverse in the Klein four-group.
Exercise: Prove that $ab$ is its own inverse in the Klein four-group.

Dear Professor, Thank you for these helpful notes. Here is my proposed solution to the exercise. Lurch seems to think I am correct!
-Your Favorite Student

Proof: We wish to show that $ab = (ab)^{-1}$. However,

$$ab = (ab)e \quad \text{Right Identity \checkmark}$$
$$= (ab)((ab)(ab)^{-1}) \quad \text{Right Inverse \checkmark}$$
$$= (((ab)(ab))(ab)^{-1} \quad \text{Associativity \checkmark}$$
$$= (((((ab)a)b)(ab)^{-1} \quad \text{Associativity \checkmark}$$
$$= ((((ba)a)b)(ab)^{-1} \quad \text{Abelian \checkmark}$$

Lurch in 2010
New Language

This defines the language boolean algebra, and marks it as the default math expression type from this point forward. Edit

Atomic expressions include:

- variables, which are lower- or upper-case letters
- a constant 0, typeset as $\bot$
- a constant $T$, typeset as $T$

Compound expressions include:

- negation, of the form $-A$, typeset as $\neg A$
- conjunction, of the form $A \land B$, typeset as $A \land B$
- disjunction, of the form $A \lor B$, typeset as $A \lor B$

Comparisons of expressions include:

- equality, of the form $A=B$, typeset as $A = B$

Grouping symbols allowed are ( ).

Axioms

$x \lor y = y \lor x$   Commutativity of $\lor$
$x \land y = y \land x$   Commutativity of $\land$
$(x \lor y) \lor z = x \lor (y \lor z)$  Associativity of $\lor$
$(x \land y) \land z = x \land (y \land z)$  Associativity of $\land$
$x \lor x = x$  Idempotency of $\lor$
$x \land x = x$  Idempotency of $\land$
Progress to 2010

• New equation-based topics can be created by an instructor without any programming required.

• Even new notations (such as the boolean algebra example) can be created from scratch, with custom typesetting and user input.

• Customizability is not yet ready for non-equation-based topics.
Plans for 2011

• The user interface is not smooth and requires an overhaul; we will dedicate the beginning of 2011 to that project.

• We aim to increase our development and testing team. Are you interested?

• Both authors have sabbaticals for the 2011-2012 academic year, and can visit your institution to bring you on board!
Goal

Bijectivity and Inverse Functions

In lecture I stated the following theorem without proof. Here is an essay-style proof of that result.

**Theorem:** A function has an inverse function if and only if it is bijective.

**Proof:** Let \( f : A \to B \).

\( \Rightarrow \) Assume \( f \) has an inverse function \( g \). Then \( g \circ f = \text{id}_A \) and \( f \circ g = \text{id}_B \) by the definition of inverse function.

Let \( x, y \in A \). Assume \( f(x) = f(y) \). Then

\[
\begin{align*}
x &= \text{id}_A(x) & \text{by the def. of identity function} \\
    &= (g \circ f)(x) & \text{by substitution} \\
    &= g(f(x)) & \text{by def. of } \circ \\
    &= g(f(y)) & \text{by substitution} \\
    &= (g \circ f)(y) & \text{by def. of } \circ \\
    &= \text{id}_A(y) & \text{by substitution}
\end{align*}
\]
Get Involved

- Download *Lurch*, try it, and send feedback.

- Join our email list.

- Give some assignments that use *Lurch*, and send us students’ feedback.

- Do development: Either write and share math topics or help write *Lurch* itself.

- Supervise student developers: We have supervised 9 math and/or CS undergrads so far, 4 of them remotely.
http://lurch.sourceforge.net
(or just Google “lurch math”)

• Download Lurch (free, open source)
• Join the email list
• Contact us by email
• Learn how to become a developer