

A FLEXIBLE, EXTENSIBLE ONLINE TESTING SYSTEM FOR MATHEMATICS

Tim Passmore Leigh Brookshaw Harry Butler

Department of Mathematics and Computing
University of Southern Queensland
AUSTRALIA

14th December 2007

OVERVIEW

BACKGROUND

FEATURES OF OTS

INFORMAL INPUT SYNTAX

FUTURE DEVELOPMENT

COMPUTER AIDED ASSESSMENT IN MATHEMATICS

ORIGINALLY DEVELOPED FOR ENTRY-SKILLS TESTING IN ALGEBRA & CALCULUS

OTS DESIGN PARAMETERS:

- ▶ General tool for computer-aided assessment & online delivery.
- ▶ Weak areas: algebra, functions, trigonometry & inequalities.
- ▶ Parse student answers rather than use multiple-choice format.
- ▶ Need to use a computer algebra system: Maxima (open source).
- ▶ **But** students do not need to know full Maxima syntax.
- ▶ Keep input syntax as simple as possible
- ▶ e.g. handle equivalents like $2.5 = 2\frac{1}{2} = \frac{5}{2}$ seamlessly.
- ▶ Mark and give immediate feedback to students online.

FEATURES OF OTS

- ▶ Questions downloaded as PDF.
- ▶ Answers uploaded on a web form.
- ▶ Informal input syntax
- ▶ Two-stage: Check and Submit cycle.
- ▶ Use XML for config files.
- ▶ PHP scripts run the system and implement data types.
- ▶ Data types: integers, floating point, inequalities, lists,...
- ▶ Computer algebra system (Maxima) only called when needed.
- ▶ Modular and self-validating against RELAX NG schema.
- ▶ Multistep decision tree: `<and>`, `<or>`, `<not>`

FEATURES OF OTS

- ▶ Questions downloaded as PDF.
- ▶ Answers uploaded on a web form.
- ▶ **Informal input syntax**
- ▶ Two-stage: Check and Submit cycle.
- ▶ Use XML for config files.
- ▶ PHP scripts run the system and implement data types.
- ▶ Data types: integers, floating point, inequalities, lists,...
- ▶ Computer algebra system (Maxima) only called when needed.
- ▶ Modular and self-validating against RELAX NG schema.
- ▶ Multistep decision tree: `<and>`, `<or>`, `<not>`

FEATURES OF OTS

- ▶ Questions downloaded as PDF.
- ▶ Answers uploaded on a web form.
- ▶ Informal input syntax
- ▶ **Two-stage: Check and Submit cycle.**
- ▶ Use XML for config files.
- ▶ PHP scripts run the system and implement data types.
- ▶ Data types: integers, floating point, inequalities, lists,...
- ▶ Computer algebra system (Maxima) only called when needed.
- ▶ Modular and self-validating against RELAX NG schema.
- ▶ Multistep decision tree: `<and>`, `<or>`, `<not>`

FEATURES OF OTS

- ▶ Questions downloaded as PDF.
- ▶ Answers uploaded on a web form.
- ▶ Informal input syntax
- ▶ Two-stage: Check and Submit cycle.
- ▶ Use XML for config files.
- ▶ PHP scripts run the system and implement data types.
- ▶ Data types: integers, floating point, inequalities, lists, . . .
- ▶ Computer algebra system (Maxima) only called when needed.
- ▶ Modular and self-validating against RELAX NG schema.
- ▶ Multistep decision tree: `<and>`, `<or>`, `<not>`

FEATURES OF OTS

- ▶ Questions downloaded as PDF.
- ▶ Answers uploaded on a web form.
- ▶ Informal input syntax
- ▶ Two-stage: Check and Submit cycle.
- ▶ Use XML for config files.
- ▶ PHP scripts run the system and implement data types.
- ▶ Data types: integers, floating point, inequalities, lists,...
- ▶ **Computer algebra system (Maxima) only called when needed.**
- ▶ Modular and self-validating against RELAX NG schema.
- ▶ Multistep decision tree: `<and>`, `<or>`, `<not>`

FEATURES OF OTS

- ▶ Questions downloaded as PDF.
- ▶ Answers uploaded on a web form.
- ▶ Informal input syntax
- ▶ Two-stage: Check and Submit cycle.
- ▶ Use XML for config files.
- ▶ PHP scripts run the system and implement data types.
- ▶ Data types: integers, floating point, inequalities, lists,...
- ▶ Computer algebra system (Maxima) only called when needed.
- ▶ Modular and self-validating against RELAX NG schema.
- ▶ Multistep decision tree: `<and>`, `<or>`, `<not>`

FEATURES OF OTS

- ▶ Questions downloaded as PDF.
- ▶ Answers uploaded on a web form.
- ▶ Informal input syntax
- ▶ Two-stage: Check and Submit cycle.
- ▶ Use XML for config files.
- ▶ PHP scripts run the system and implement data types.
- ▶ Data types: integers, floating point, inequalities, lists, . . .
- ▶ Computer algebra system (Maxima) only called when needed.
- ▶ Modular and self-validating against RELAX NG schema.
- ▶ Multistep decision tree: `<and>`, `<or>`, `<not>`

IMPLEMENTATION

- ▶ Used in 3 semesters in 2006–2007 a total of ~ 450 students.
- ▶ Most accept the system well and seem to like the format.
- ▶ Good wake-up call to revise or gain skills.
- ▶ Takes time to build student confidence in reliability of marking.
- ▶ Some students fear being marked wrong because of a syntax error.
- ▶ Takes time to develop and test good questions.

IMPLEMENTATION

- ▶ Used in 3 semesters in 2006–2007 a total of ~ 450 students.
- ▶ Most accept the system well and seem to like the format.
- ▶ Good wake-up call to revise or gain skills.
- ▶ Takes time to build student confidence in reliability of marking.
- ▶ Some students fear being marked wrong because of a syntax error.
- ▶ Takes time to develop and test good questions.

PRE-FILTER STUDENT INPUT

EXAMPLE

- ▶ Students type: $(2x-1)(x+3)$
- ▶ Maxima needs: $(2*x-1)*(x+3)$
- ▶ Solution: prefilter to “add stars”

PRE-FILTER STUDENT INPUT

EXAMPLE

- ▶ Students type: $(2x-1)(x+3)$
 - ▶ Maxima needs: $(2*x-1)*(x+3)$
 - ▶ Solution: prefilter to “add stars”
-
- ▶ Keep syntax as close to standard conventions as possible.
 - ▶ Give specific syntax hints as necessary
e.g. indicate exponentiation with \wedge symbol.

USE XML TO SPECIFY SOLUTION

EXAMPLE (ANSWER: $16b/a$ OR $b*16/a$)

Express $\frac{16(a^2b^4)^{-\frac{1}{2}}}{b^{-3}}$ as a simple fraction involving no negative powers.

```
<solution>
  <regexp>
    <prefilter action="remove white space"/>
    <prefilter action="lower case"/>
    <prefilter action="remove stars"/>
    <prefilter action="substitute parenthesis"/>
    <prefilter action="remove brackets"/>
    <value>@(16b|b16)/a$</value>
  </regexp>
</solution>
```

USE XML TO SPECIFY SOLUTION

EXAMPLE (ANSWER: $16b/a$ OR $b*16/a$)

Express $\frac{16(a^2b^4)^{-\frac{1}{2}}}{b^{-3}}$ as a simple fraction involving no negative powers.

```
<solution>
  <regex>
    <prefilter action="remove white space"/>
    <prefilter action="lower case"/>
    <prefilter action="remove stars"/>
    <prefilter action="substitute parenthesis"/>
    <prefilter action="remove brackets"/>
    <value>@(16b|b16)/a$</value>
  </regex>
</solution>
```


USE XML TO SPECIFY SOLUTION

EXAMPLE (ANSWER: $16b/a$ OR $b*16/a$)

Express $\frac{16(a^2b^4)^{-\frac{1}{2}}}{b^{-3}}$ as a simple fraction involving no negative powers.

```
<solution>
  <regexp>
    <prefilter action="remove white space"/>
    <prefilter action="lower case"/>
    <prefilter action="remove stars"/>
    <prefilter action="substitute parenthesis"/>
    <prefilter action="remove brackets"/>
    <value>@^(16b|b16)/a$@</value>
  </regexp>
</solution>
```

USE XML TO SPECIFY SOLUTION

EXAMPLE (ANSWER: $16b/a$ OR $b*16/a$)

Express $\frac{16(a^2b^4)^{-\frac{1}{2}}}{b^{-3}}$ as a simple fraction involving no negative powers.

```
<solution>
  <regexp>
    <prefilter action="remove white space"/>
    <prefilter action="lower case"/>
    <prefilter action="remove stars"/>
    <prefilter action="substitute parenthesis"/>
    <prefilter action="remove brackets"/>
    <value>@^(16b|b16)/a$@</value>
  </regexp>
</solution>
```

PASSING CODE TO MAXIMA

EXAMPLE (ALGEBRA QUESTION)

Expand $(x + 1)(-2x + 1)(x - 3)$.

(Exponents or powers must be typed using the caret character \wedge . For example, type x^2 as x^2 .)

```
<solution>
  <and>
    <not>
      <regexp>
        <prefilter action="substitute parenthesis"/>
        <prefilter action="remove white space"/>
        <prefilter action="add stars"/>
        <value>@^\(.+\)\*\(.+\)\*\(.+\)$@</value>
      </regexp>
    </not>
```

PASSING CODE TO MAXIMA

EXAMPLE (ALGEBRA QUESTION)

Expand $(x + 1)(-2x + 1)(x - 3)$.

(Exponents or powers must be typed using the caret character \wedge . For example, type x^2 as x^2 .)

```
<solution>
```

```
<and>
```

```
<not>
```

```
<regexp>
```

```
<prefilter action="substitute parenthesis"/>
```

```
<prefilter action="remove white space"/>
```

```
<prefilter action="add stars"/>
```

```
<value>@^\(.\+\)\*\(.\+\)\*\(.\+\)$@</value>
```

```
</regexp>
```

```
</not>
```

PASSING CODE TO MAXIMA (2)

```
<maxima>
  <prefilter action="substitute parenthesis"/>
  <prefilter action="remove white space"/>
  <prefilter action="lower case"/>
  <prefilter action="add stars"/>
  <script>
    display2d:false$ expand((x+1)*(-2*x+1)*(x-3)-(@));
  </script>
  <value>0</value>
</maxima>
</and>
</solution>
```

PASSING CODE TO MAXIMA (2)

```
<maxima>
  <prefilter action="substitute parenthesis"/>
  <prefilter action="remove white space"/>
  <prefilter action="lower case"/>
  <prefilter action="add stars"/>
  <script>
    display2d:false$ expand((x+1)*(-2*x+1)*(x-3)-(@));
  </script>
  <value>0</value>
</maxima>
</and>
</solution>
```

PASSING CODE TO MAXIMA (2)

```

<maxima>
  <prefilter action="substitute parenthesis"/>
  <prefilter action="remove white space"/>
  <prefilter action="lower case"/>
  <prefilter action="add stars"/>
  <script>
    display2d:false$ expand((x+1)*(-2*x+1)*(x-3)-(@));
  </script>
  <value>0</value>
</maxima>
</and>
</solution>

```

PASSING CODE TO MAXIMA (2)

```
<maxima>
  <prefilter action="substitute parenthesis"/>
  <prefilter action="remove white space"/>
  <prefilter action="lower case"/>
  <prefilter action="add stars"/>
  <script>
    display2d:false$ expand((x+1)*(-2*x+1)*(x-3)-(@));
  </script>
  <value>0</value>
</maxima>
</and>
</solution>
```


FUTURE DEVELOPMENT

- ▶ Front-end question/solution authoring system in \LaTeX or other.
- ▶ Better-targeted feedback to student on areas of weakness.
- ▶ Question banks, randomly generated tests, specifically targeted tests.
- ▶ Back-end admin systems: reports, error analysis etc.
- ▶ Currently data captured as TSV but easily adapted to MySQL etc.