Algebraic Expressions with Uncertain Syntax and Their Applications in Online Learning

Andrew Pownuk[1], Iwona Skalna[2]

1 - Department of Mathematical Sciences, University of Texas at El Paso
2 - Department of Business Informatics & Management
AGH University of Science and Technology, Krakow, Poland

27th Joint NMSU/UTEP Workshop on Mathematics, Computer Science, and Computational Sciences
Las Cruces, New Mexico, April 2, 2022

Outline

1. Motivation
2. Online learning
3. How to create online assignments?
4. Typical problems in mathematical representation of the student's answers
5. Arithmetic expressions with errors
6. Multi-valued logic
7. Learning
8. Conclusions

Motivation

Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions

What to do with the student's answers with errors?
Correct answer:

$$(1+2)$$

Answers with errors:

$$(1+2($$

$$(1+*2$$

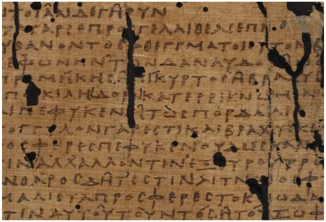
$$(1+(2)$$

$$\dots$$

Online learning

Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions

Early examples of distance learning



Online learning

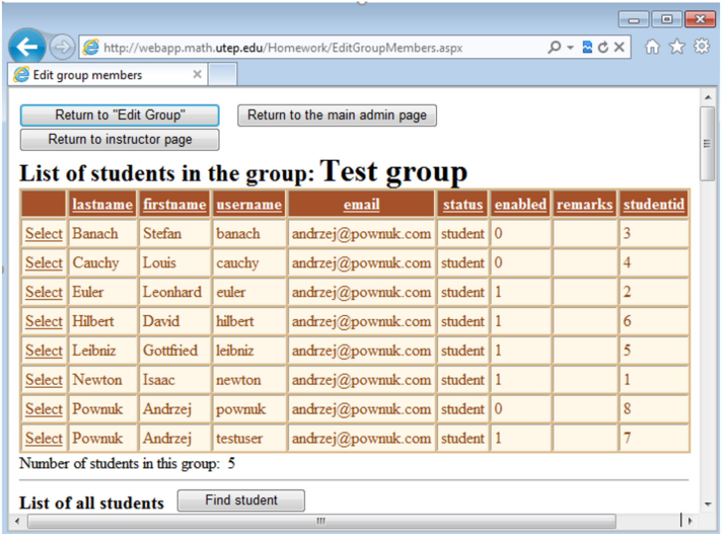
User Main Page

	Logoff
Username:	80080081
First Name:	Pownuk
Last Name:	Andrew
Group:	2020-Fall-MATH-1312-CRN-12219
	Change password
	Change e-mail

Show my grades

Files (notes, syllabus etc.)

List of students



http://webapp.math.utep.edu/Homework/EditGroupMembers.aspx

Edit group members

Return to "Edit Group" | Return to the main admin page | Return to instructor page

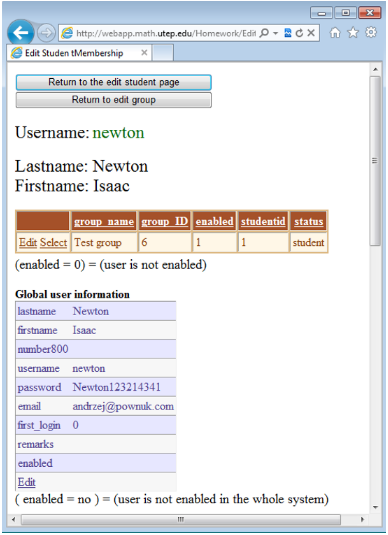
List of students in the group: Test group

	lastname	firstname	username	email	status	enabled	remarks	studentid
Select	Banach	Stefan	banach	andrzej@pownuk.com	student	0		3
Select	Cauchy	Louis	cauchy	andrzej@pownuk.com	student	0		4
Select	Euler	Leonhard	euler	andrzej@pownuk.com	student	1		2
Select	Hilbert	David	hilbert	andrzej@pownuk.com	student	1		6
Select	Leibniz	Gottfried	leibniz	andrzej@pownuk.com	student	1		5
Select	Newton	Isaac	newton	andrzej@pownuk.com	student	1		1
Select	Pownuk	Andrzej	pownuk	andrzej@pownuk.com	student	0		8
Select	Pownuk	Andrzej	testuser	andrzej@pownuk.com	student	1		7

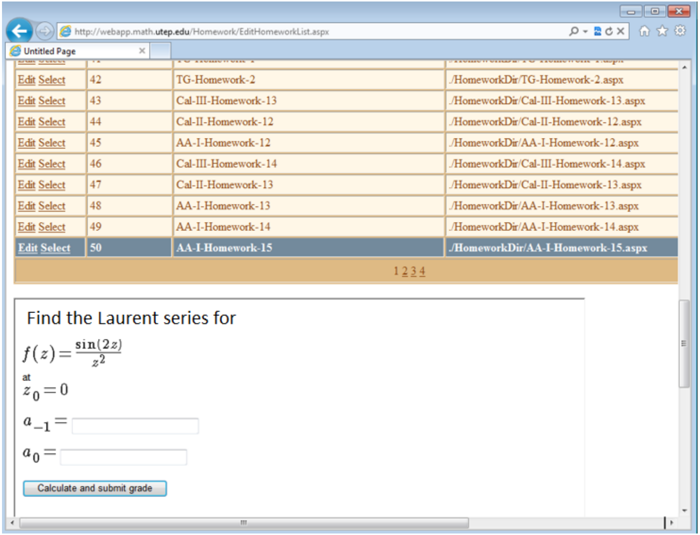
Number of students in this group: 5

List of all students Find student

Information about particular student

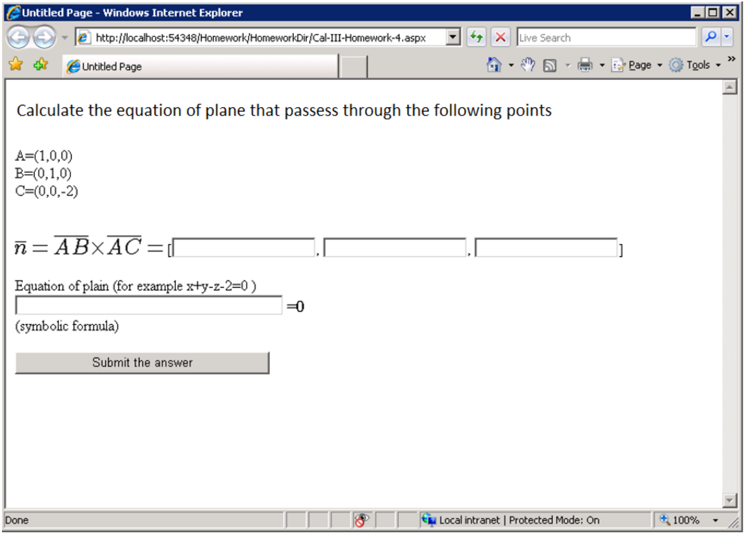


Online homework



Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions

Online homework



Online visualization

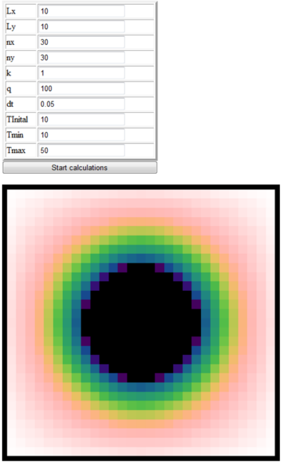


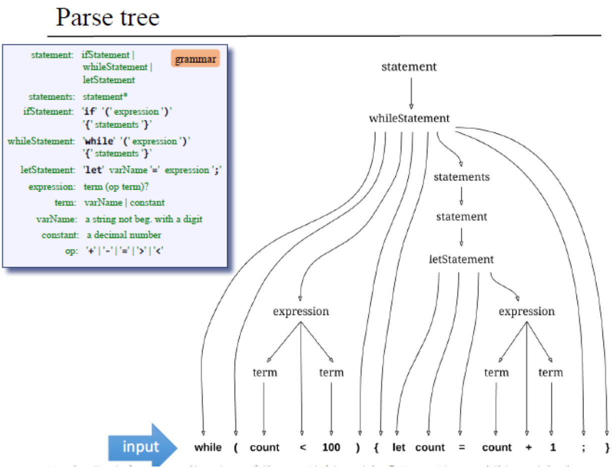
Figure: Solution of the heat transfer equation

Sample Problem

Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions

- Find area of the parallelogram for $\bar{a} = [1, 2, 3]$, $\bar{b} = [3, 2, 1]$. Answer: $A = |\bar{a} \times \bar{b}| = 4\sqrt{6}$.
- How to input $4\sqrt{6}$ into the system in order to provide the answer?
- It is possible to use text description of the expression. For example:
  - $4 * sqrt(6)$
  - $4 * Sqrt[6]$
  - $4sqrt(6)$
  - $4sqrt6$
  - $4 \cdot sq6$
  - etc.

Parse Tree


Parse tree
statement: ifStatement | whileStatement | letStatement
grammar
statements: statement*
ifStatement: 'if' '(' expression ')' '{' statements '}'
whileStatement: 'while' '(' expression ')' '{' statements '}'
letStatement: 'let' varName '=' expression ';'
expression: term (op term)?
term: varName | constant
varName: a string not beg. with a digit
constant: a decimal number
op: '+' | '-' | '=' | '>' | '<'
statement
whileStatement
statements
statement
letStatement
expression
term
term
expression
term
term
input
while (count < 100) { let count = count + 1 ; }

XML Parse Tree

Parse tree

grammar

```
statement:      ifStatement |
                whileStatement |
                letStatement
statements:     statement*
ifStatement:    'if' '(' expression ')'
                '{' statements '}'
whileStatement: 'while' '(' expression ')'
                '{' statements '}'
letStatement:   'let' varName '=' expression ';'
expression:     term (op term)?
term:           varName | constant
varName:        a string not beg. with a digit
constant:       a decimal number
op:             '+' | '-' | '=' | '>' | '<'
```

parser output

```
<whileStatement>
  <keyword> while </keyword>
  <symbol> ( </symbol>
  <expression>
    <term>
      <identifier> count </identifier>
    </term>
    <symbol> < </symbol>
    <term>
      <intConstant> 100 </intConstant>
    </term>
  </expression>
  <symbol> ) </symbol>
  <symbol> { </symbol>
  <statements>
    <letStatement>
      <keyword> let </keyword>
      <identifier> count </identifier>
      <symbol> = </symbol>
      <expression>
        <term> <identifier> count </identifier> </term>
        <symbol> + </symbol>
        <term> <intConstant> 1 </intConstant> </term>
      </expression>
      <symbol> ; </symbol>
    </letStatement>
  </statements>
  <symbol> } </symbol>
</whileStatement>
```

Same parse tree, in XML

Grammar

<table>
<tr><td>Lexical elements:</td><td>The Jack language includes five types of terminal elements (tokens):</td></tr>
<tr><td>keyword:</td><td>'class' | 'constructor' | 'function' | 'method' | 'field' | 'static' | 'var' | 'int' | 'char' | 'boolean' | 'void' | 'true' | 'false' | 'null' | 'this' | 'let' | 'do' | 'if' | 'else' | 'while' | 'return'</td></tr>
<tr><td>symbol:</td><td>'{' | '}' | '(' | ')' | '[' | ']' | '.' | ',' | ';' | '+' | '-' | '*' | '/' | '&' | '|' | '<' | '>' | '=' | '~'</td></tr>
<tr><td>integerConstant:</td><td>A decimal number in the range 0 .. 32767.</td></tr>
<tr><td>StringConstant</td><td>'"' A sequence of Unicode characters not including double quote or newline '"'</td></tr>
<tr><td>identifier:</td><td>A sequence of letters, digits, and underscore ('_') not starting with a digit.</td></tr>
<tr><td>Program structure:</td><td>A Jack program is a collection of classes, each appearing in a separate file. The compilation unit is a class. A class is a sequence of tokens structured according to the following context free syntax:</td></tr>
<tr><td>class:</td><td>'class' className '{' classVarDec* subroutineDec* '}'</td></tr>
<tr><td>classVarDec:</td><td>('static' | 'field') type varName (',' varName)* ';'</td></tr>
<tr><td>type:</td><td>'int' | 'char' | 'boolean' | className</td></tr>
<tr><td>subroutineDec:</td><td>('constructor' | 'function' | 'method') ('void' | type) subroutineName '(' parameterList ')' subroutineBody</td></tr>
<tr><td>parameterList:</td><td>((type varName) (',' type varName)*)?</td></tr>
<tr><td>subroutineBody:</td><td>'{' varDec* statements '}'</td></tr>
<tr><td>varDec:</td><td>'var' type varName (',' varName)* ';'</td></tr>
<tr><td>className:</td><td>identifier</td></tr>
<tr><td>subroutineName:</td><td>identifier</td></tr>
<tr><td>varName:</td><td>identifier</td></tr>
<tr><td>Statements:</td><td></td></tr>
<tr><td>statements:</td><td>statement*</td></tr>
<tr><td>statement:</td><td>letStatement | ifStatement | whileStatement | doStatement | returnStatement</td></tr>
<tr><td>letStatement:</td><td>'let' varName ('[' expression ']')? '=' expression ';'</td></tr>
<tr><td>ifStatement:</td><td>'if' '(' expression ')' '{' statements '}' ('else' '{' statements '}')?</td></tr>
<tr><td>whileStatement:</td><td>'while' '(' expression ')' '{' statements '}'</td></tr>
<tr><td>doStatement:</td><td>'do' subroutineCall ';'</td></tr>
<tr><td>ReturnStatement</td><td>'return' expression? ';'</td></tr>
<tr><td>Expressions:</td><td></td></tr>
<tr><td>expression:</td><td>term (op term)*</td></tr>
<tr><td>term:</td><td>integerConstant | stringConstant | keywordConstant | varName | varName '[' expression ']' | subroutineCall | '(' expression ')' | unaryOp term</td></tr>
<tr><td>subroutineCall:</td><td>subroutineName '(' expressionList ')' | (className | varName) '.' subroutineName '(' expressionList ')'</td></tr>
<tr><td>expressionList:</td><td>(expression (',' expression)*)?</td></tr>
<tr><td>op:</td><td>'+' | '-' | '*' | '/' | '&' | '|' | '<' | '>' | '='</td></tr>
<tr><td>unaryOp:</td><td>'-' | '~'</td></tr>
<tr><td>KeywordConstant:</td><td>'true' | 'false' | 'null' | 'this'</td></tr>
</table>

Infix notation, Prefix notation, Postfix notation

Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions

Different notation for arithmetic expressions.

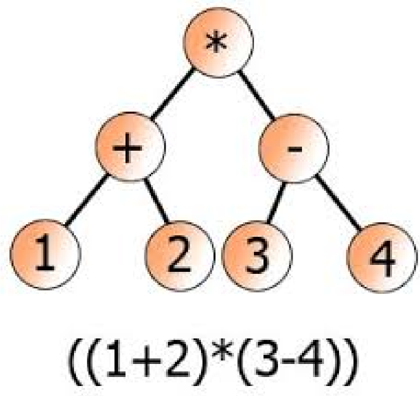
- Infix notation $(5+6)x7$
- Prefix notation $x+567$
- Postfix notation $756+x$

Typical evaluation process of arithmetic expressions.

- InfixToPrefix($(5+6)x7$)=$x+567$
- EvaluatePrefix($x+567$)=77

Expression Tree

Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions



How to evaluate mathematical expression Given as a String?

Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions

Example

- Expression: sqrt(2)*6
- Value: $\sqrt{2} \cdot 6$

Example

- Expression: sqrt2*6
- Value: $\sqrt{2} \cdot 6$

Example

- Expression: Sqrt[2]6
- Value: $\sqrt{2} \cdot 6$

Example

- Expression: SQRT[2]6
- Value: $\sqrt{2} \cdot 6$

WebAssign question modes

Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions

Mode	Description
Algebraic	Students submit a mathematical expression or equation that is evaluated algebraically.
Essay	Students submit an extended textual response. Scored manually.
File-Upload	Students submit a file. Scored manually.
Fill-in-the-Blank	Students submit a brief textual response.
Graphing	Students draw on a Cartesian coordinate plane.
Image Map	Students click a displayed image.
Java	Students use Java, Flash, or other applet to answer the question.
Matching	Students match items from two lists.
Multiple-Choice	Students select one response from a list.
Multiple-Select	Students select one or more responses from a list.
NumberLine	Students place or draw points, lines, segments, or rays.
Numerical	Students submit a numerical response which might include units or be checked for significant figures.
Poll	Used with other modes to create questions that collect information only. All responses are scored correct.
Symbolic	Students submit a mathematical expression (not an equation) that is evaluated by value substitution.

WebAssign questions from code (HTML, Perl etc.)

Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions

The following code in Answer accepts two values between 20 and 100, and then requires your students to multiply them:

```
<EQN $A=userinput(20,100,50)>
<EQN $B=userinput(20,100,50)>
<EQN $SIMPLIFIED=1; $A * $B>
```

To require students to perform the computations instead of stating their response as a mathematical expression, set the $SIMPLIFIED, $FRACTION, or $PROPERFRACTION variables to 1 as you would for any other numerical question.

Turing complete programming languages in teaching

Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions

Any Turing complete programming language (e.g. C,C++,Python, Java etc.) can express wide class of algorithms.
This gives teachers a tool to work interactively online on problems with practically arbitrary complexity.
To use this tool, it is necessary to represent a mathematical problem in a form of computer code and then the students by using webpages can interact with this code to check their progress. In the same way it is possible to work with research problems with arbitrary complexity.

Curry–Howard correspondence

Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions

The Curry–Howard correspondence (also known as the Curry–Howard isomorphism or equivalence, or the proofs-as-programs and propositions- or formulae-as-types interpretation) is the direct relationship between computer programs and mathematical proofs.
Teacher/scientists can use this relation and create computer programs from mathematical formulations and mathematical formulations from computer programs.

Arithmetic expressions with errors

Motivation

Online learning

How to create online assignments?

Typical problems in mathematical representation of the student's answers

Arithmetic expressions with errors

Multi-valued logic

Learning

Conclusions

$1 * +)2)$

$5 + \hat{}2$

$1 * +2)$

$1 + \hat{}2$

$1 * +(2$

$3 + 2$

$1 * +2$

Correct formula

$\mathbf{1 + 2}$

$\frac{1+2}{(\ )}$

$1 + 2 + 0$

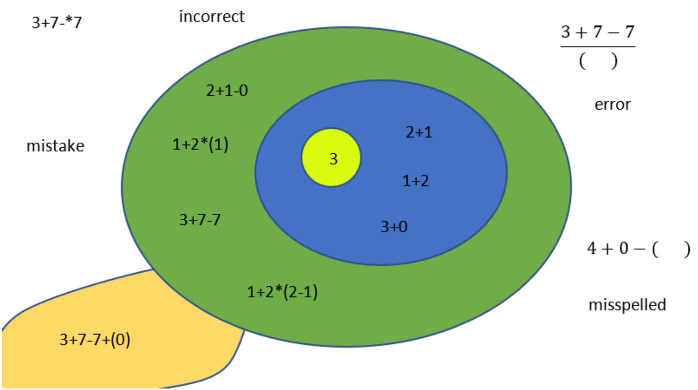
$\frac{1+2}{(1+(\ ))}$

$1 + 2 * 0$

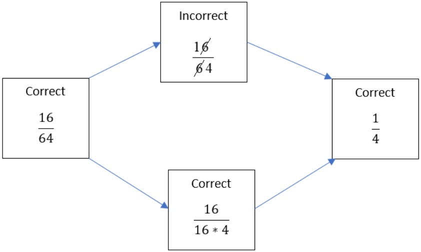
$1 + 2/0$

Correct formulation is surrounded with incorrect statements

Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions

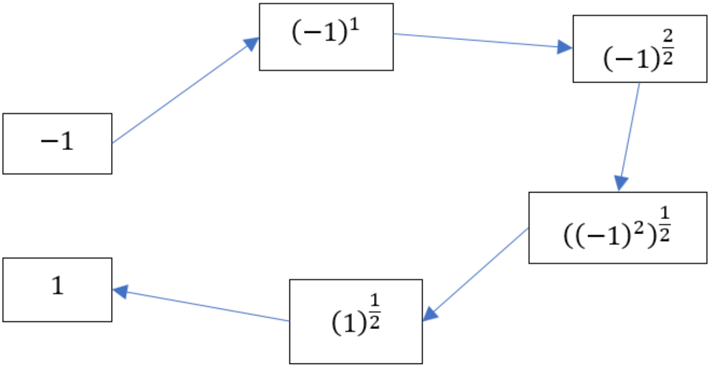


Mathematical fallacy - problems with syntax



$$\frac{d}{dx}\frac{1}{x} = \frac{\cancel{d}}{\cancel{d}x}\frac{1}{x} =$$
$$= \frac{1}{x}\frac{1}{x} = -\frac{1}{x^2}.$$

Mathematical fallacy - problems with algebra



Mathematical fallacy - problems with algebra

$$
\begin{aligned}
-1 &= -1 \\
-1/1 &= -1/1 \\
-1/1 &= 1/-1 \\
\sqrt{-1/1} &= \sqrt{1/-1} \\
i/1 &= 1/i \\
i &= 1/i \\
i^2 &= 1 \\
-1 &= 1
\end{aligned}
$$

Problems with visual learning

Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions

Example

$$\lim_{x \to 8^+} \frac{1}{x-8} = \infty$$

Conclusion

$$\lim_{x \to 5^+} \frac{1}{x-5} =$$

Problems with visual learning

Example

$$\lim_{x \to 8^+} \frac{1}{x-8} = \infty$$

Conclusion

$$\lim_{x \to 5^+} \frac{1}{x-5} = 5$$

Problems with visual learning

Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions

Example

$$a \times b = ab$$

Conclusion

$$2 \times 2 = ?$$

Problems with visual learning

Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions

Example

$$a \times b = ab$$

Conclusion

$$2 \times 2 = 22(incorrect)$$

$$2 \times 2 = 4(correct)$$

Problems with visual learning

Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions

Example

$$a \times b = ab$$

Conclusion

$$2 \times 2 = 22 (incorrect)$$

$$2 \times 2 = 4 (correct)$$

String concatenation

Example

$$`a` + `b` = `ab`$$

Conclusion

$$`2` + `2` = `?`$$

Problems with visual learning

Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions

Example

$$a \times b = ab$$

Conclusion

$$2 \times 2 = 22 (incorrect)$$

$$2 \times 2 = 4 (correct)$$

String concatenation

Example

$$`a` + `b` = `ab`$$

Conclusion

$$`2` + `2` = `4` (incorrect)$$

$$`2` + `2` = `22` (correct)$$

Problems with classification

Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions

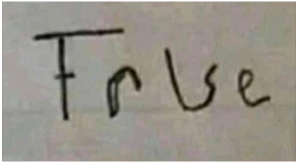
Is this true that this is a cat?

Answer:

Problems with classification

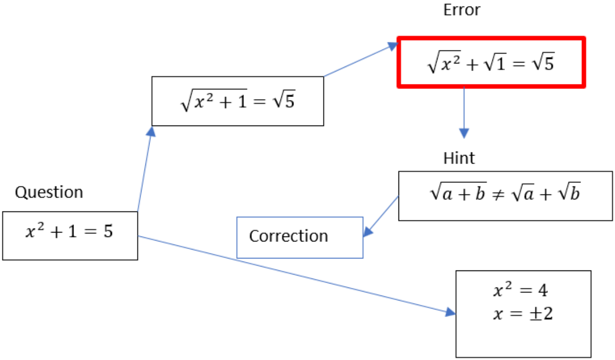
Is this true that this is a cat?

Answer:



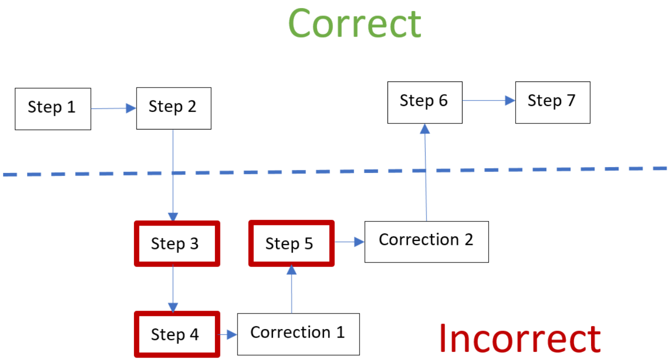
Error correction - example

Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions



Error correction - generalization

Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions



Arithmetic operators with multiple form

Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions

Input expression with a syntax error

$$2 * (3 + */2)$$

Arithmetic operators with multiple form

$$\{+, *, /\}$$

Possible multiple form of the expression

$$2 * (3\{+, *, /\}2) = \{2 * (3 + 2), 2 * (3 * 2), 2 * (3/2)\}$$

Missing parts of the expression

Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions

Input expression with a syntax error

$$2 * (3 + 2$$

Input expression with correction

$$2 * (3 + 2)$$

Missing parts of the expression - values

Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions

Input expression with a syntax error

$$2 * (3+)$$

Input expression with correction

$$2 * (3 + \{x, y, f(x), g(x, y)\})$$

$$\{2 * (3 + x), 2 * (3 + y), 2 * (3 + f(x)), 2 * (3 + g(x, y))\}$$

where x, y are some real/integer/rational values.
$f(x), g(x, y)$ are some functions/expressions.

Applications of multi-valued logic for ordering of errors in arithmetic expressions

Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions

Input expression with a syntax error

$$2 * (3 + */2)$$

Correct expression

$$2 * (3 + 2)$$

Partial ordering relation $\leq$ between "more correct" and "'less correct " expressions

$$2 * (3 + */2) \leq 2 * (3 + *2) \leq 2 * (3 + 2)$$

Some expressions have similar level of "correctness".

$$2 * (3 + *2) \approx 2 * (3 + /2) \approx 2 * (3 * /2)$$

Applications of multi-valued logic for ordering of errors in arithmetic expressions

Correctness of some expressions cannot be compared

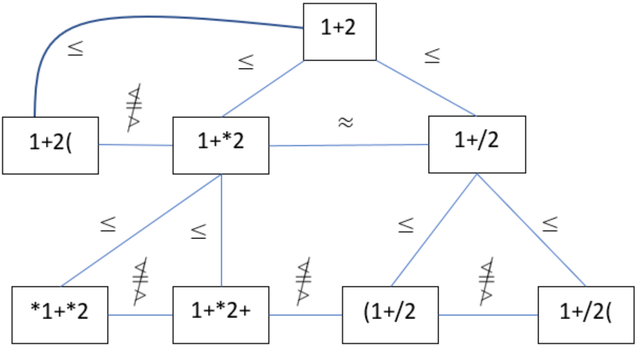
$$2*(3+2 \mathrel{\not\substack{\triangleleft \\ = \\ \triangleright}} 2*(3+*2)$$

In the expression $2*(3+2$ there is a problem with parentheses. In the expression $2*(3+*2)$ there is a problem with mathematical operator.

$$2*)3+*2 \mathrel{\not\substack{\triangleleft \\ = \\ \triangleright}} 1+2*$$

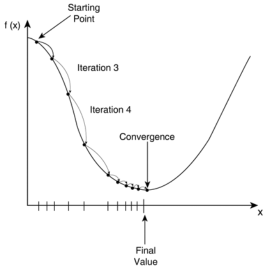
‘Hesse diagram’ for syntax errors

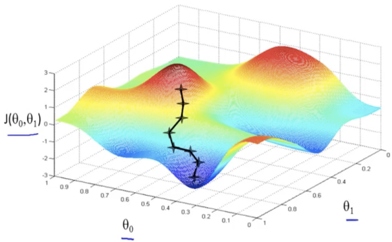
Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student’s answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions



Learning as minimization of functional

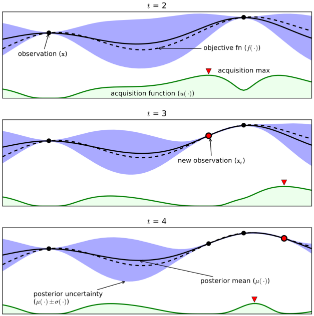
Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions





Learning as sequence of updating

Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions



Recursive Bayesian estimation

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

Learning as sequence of explanations, trials, and corrections

Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions

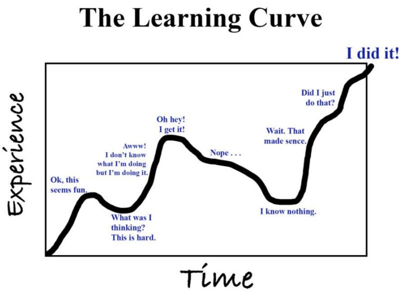


Figure: Learning process is represented as a curve

Learning described in this presentation is modeled as a sequence of explanations, trials, and corrections.

Conclusions

Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions

- Turing complete programming languages give teachers a tool to work interactively online on problems with practically arbitrary complexity.
- Syntax and grammar analysis of the mathematical statements can improve online learning systems.
- Arithmetic expressions with errors can have possibly many different corrections and meanings.
- Some corrections can be represented as set operations.
- Multi-valued logic can be used to study various stages of correctness.
- Presented approach can be use as a mathematical foundation for "partial credits".
- Learning can be viewed as a sequence of explanations, trials, and corrections.

Motivation
Online learning
How to create online assignments?
Typical problems in mathematical representation of the student's answers
Arithmetic expressions with errors
Multi-valued logic
Learning
Conclusions

Thank You