Automated development of the theory of equations with uncertain parameters using self-adaptive and self-learning autonomous systems with the applications for estimation of reliability of engineering structures

A. Pownuk[1], I. Skalna[2], M.V. Rama Rao[3]

1 - University of Texas at El Paso, El Paso, Texas, USA
2 - AGH University of Science and Technology, Krakow, Poland
3 - Vasavi College of Engineering, Hyderabad, India

Outline

Automated Text Generation

Write speed of some of today's SSD drives is approximately 7500 MB/s. One standard page is comprised of 1800 characters with spaces. This allows to write more than 4 millions pages of text per second.

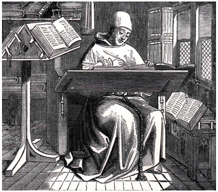

Figure: Writing text.

According to Google in the year 2010 there were approximately 130 millions books published since Gutenberg.

Lexical Decision Tasks

Moscoso del Pradon uses his method to determine how much information the brain can process during lexical decision tasks. The answer? Conscious mind can process no more than about 60 bits per second (Emerging Technology, New Measure of Human Brain Processing Speed, MIT Technology Review, August 25, 2009).

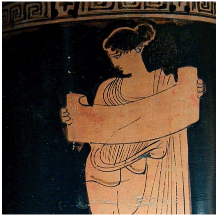

Figure: Muse reading a scroll, attic red-figure, c. 430 BC.

High Performance Computing

High Performance Computing (HPC) refers to the practice of aggregating computing power in a way that delivers much higher performance in order to solve large problems in science, engineering, or business.

Figure: IBM Blue Gene/P supercomputer.

Floating Point Operations per Second (FLOPS)

The Frontier supercomputer, built at the Department of Energy's Oak Ridge National Laboratory in Tennessee, has a processor speed of 1.1 exaFLOPS (1.1 quintillion $= 10^{18}$ floating point operations per second, or FLOPS).

Figure: Frontier (Oak Ridge National Laboratory).

Studies have shown that the average programmer in a production environment puts out about 10-100 lines of code per day (depending on an experience).

Scientific Method

The scientific method is an empirical method of acquiring knowledge.

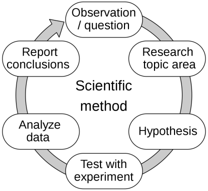

Figure: The scientific method is often represented as an ongoing process.

Aristotle (384–322 BCE) is recognized as the inventor of scientific method.

Tools That Support Scientist and Engineers

Figure: NACA High Speed Flight Station "Computer Room" (1949).

Figure: International Space Station control room.

Tools That Support Scientist and Engineers

Figure: Manual drawing, source: The Mind Circle, 15/05/2019.

Figure: A computer lab.

Benefits of Autonomous Computational Methods

Main benefits:

- scalability,
- improved formal correctness of the results.

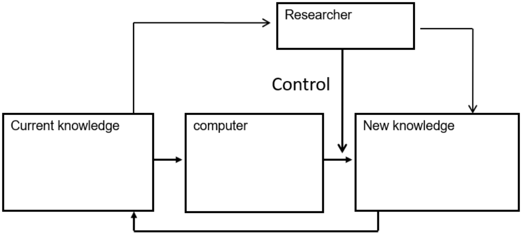

Figure: Autonomous Computational Methods.

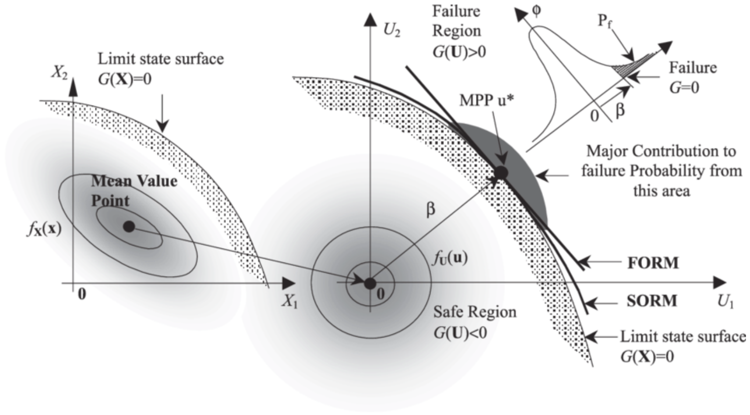

Figure: First-and second-order reliability methods.

Worst Case Analysis

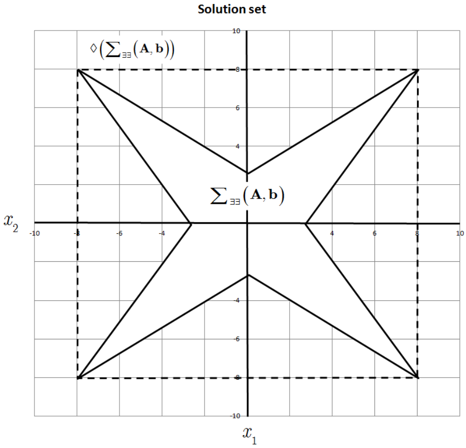

Figure: Solution set of the system of linear interval equations.

Modeling of Engineering Structures

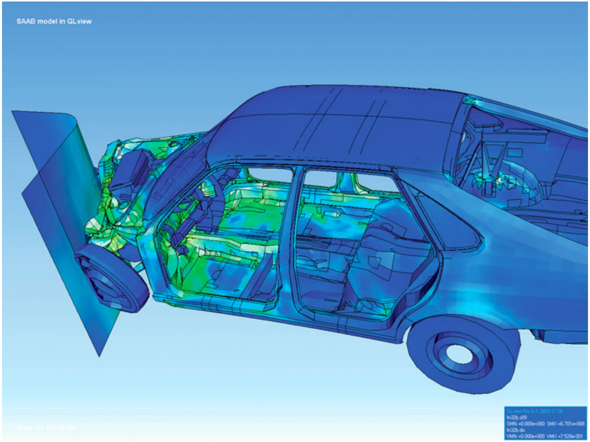

Figure: FEM model of a car.

Linear elasticity

Equation of motion:

$\boldsymbol{\nabla} \cdot \boldsymbol{\sigma} + \mathbf{F} = \rho \ddot{\mathbf{u}}$

Constitutive equations:

$\boldsymbol{\sigma} = \mathsf{C} : \boldsymbol{\varepsilon}$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor, $\boldsymbol{\varepsilon}$ is the infinitesimal strain tensor, $\mathbf{u}$ is the displacement (vector), C is the fourth-order stiffness tensor, $\mathbf{F}$ is the body force per unit volume, ρ is the mass density.

For isotropic media

$C_{ijkl} = K\,\delta_{ij}\,\delta_{kl} + \mu\,(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk} - \tfrac{2}{3}\,\delta_{ij}\,\delta_{kl})$

$\sigma_{ij} = K\delta_{ij}\varepsilon_{kk} + 2\mu\left(\varepsilon_{ij} - \tfrac{1}{3}\delta_{ij}\varepsilon_{kk}\right)$

$\varepsilon_{ij} = \frac{1}{2\mu}\sigma_{ij} - \frac{\nu}{E}\delta_{ij}\sigma_{kk} = \frac{1}{E}[(1+\nu)\sigma_{ij} - \nu\delta_{ij}\sigma_{kk}]$

A beam is a structural element

The Euler–Bernoulli equation $\frac{\mathrm{d}^2}{\mathrm{d}x^2}\left(EI\frac{\mathrm{d}^2 w}{\mathrm{d}x^2}\right) = q$

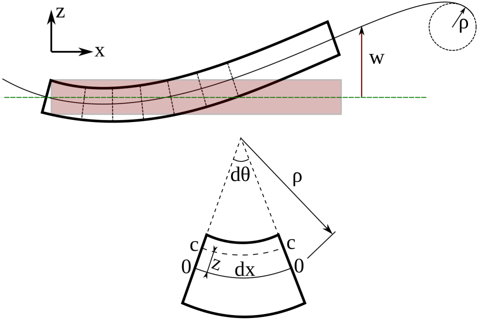

Figure: Bending of an Euler–Bernoulli beam

Numerical Methods

The finite element method.

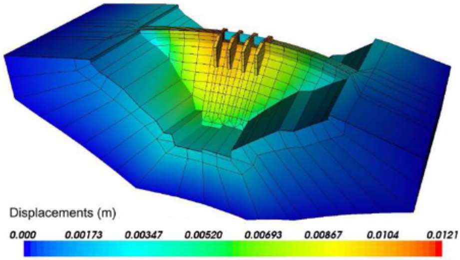

Fig. 13
Finite element model displacements of Ribeiradio dam (LNEC, 2015)
Déplacements du barrage de Ribeiradio (LNEC, 2015)

Figure: Source:A. Morgado et al., Ribeiradio dam foundation treatment – design, effectiveness control and performance, 26 Congress th 86 Annual Meeting 1-7 July, Vienna 2018

Adaptive FEM

The method adaptively changes FEM elements in order to minimize the approximation error. For example, for the following problem

$$\begin{cases} -u'' = f \text{ in } [a, b] \\ u(a) = u(b) = 0 \end{cases}$$

the error can be approximated in the following way

$$\|u - u_h\| \leq Ch\|u''\|_{L^2(a,b)}.$$

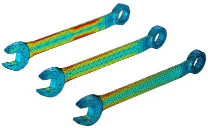

Figure: Source: www.comsol.com

Control Theory

By using the control theory it is possible adaptively control behavior of the system.

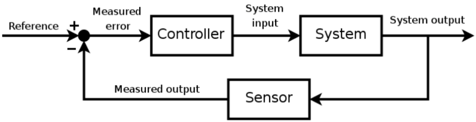

Figure: A block diagram of a feedback control system.

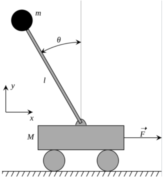

Figure: Inverted pendulum on a cart.

COntinuous CONstraints - Updating the Technology - COCONUT

The goal of this project was to integrate the currently available techniques from mathematical programming, constraint programming, and interval analysis into a single discipline, to get algorithms for global optimization and continuous constraint satisfaction problems that outperform the current generation of algorithms based on using only techniques from one or two of the traditions.

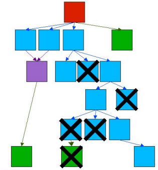

Figure: Graph representation of a search space.

Reinforcement Learning

Reinforcement learning (RL) is an area of machine learning concerned with how intelligent agents ought to take actions in an environment in order to maximize the notion of cumulative reward. Reinforcement learning is one of three basic machine learning paradigms, alongside supervised learning and unsupervised learning.

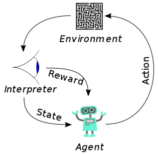

Figure: Reinforcement Learning.

Finding Algorithm

Monte Carlo tree search.

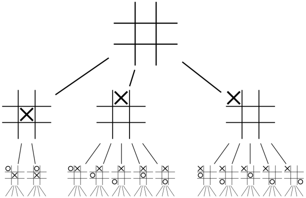

Figure: Graph representing the game tic-tac-toe.

Due to high complexity of the graph which represents many games in general it is not possible to find an optimal strategy/algorithm in an explicit way.

Serendipity

Serendipity is an unplanned fortunate discovery. Serendipity is a common occurrence throughout the history of product invention and scientific discovery.

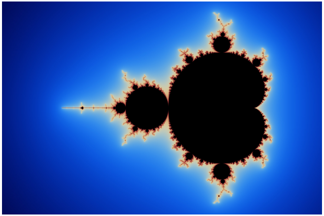

Figure: Mandelbrot set.

In general, many phenomenons in science are highly unpredictable and it is hard to use straightforward extrapolation methods in all situations.

Self-adaptive Computational Methods

In general it is possible to consider a class of problems and adaptability improve all aspects the problem (mathematical formulation and computer implementation).

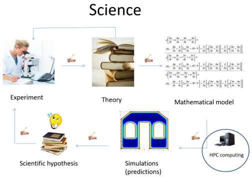

Figure: Science.

In general, it is possible to consider very wide class of possible changes of the code and mathematical formulations which allows to explore large class of algorithms and possibly find better solution of existing scientific problems.

Von Neumann Universal Constructor

John von Neumann's universal constructor is a self-replicating machine in a cellular automaton (CA) environment. It was designed in the 1940s, without the use of a computer.

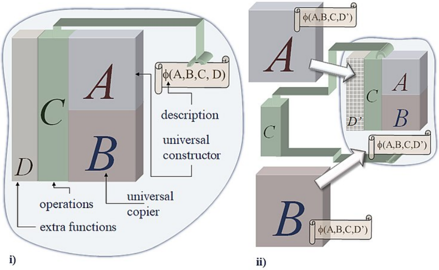

Figure: Von Neumann's System of Self-Replication Automata with the ability to evolve.

Conway's Game of Life

The Game of Life, also known simply as Life, is a cellular automaton devised by the British mathematician John Horton Conway in 1970. It is a zero-player game, meaning that its evolution is determined by its initial state, requiring no further input.

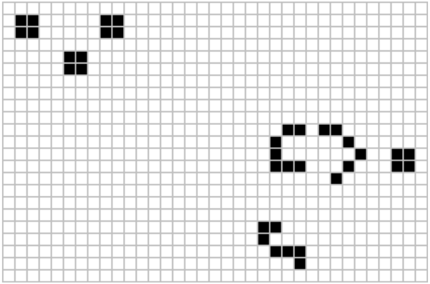

Figure: Conway's Game of Life.

Gödel Machine

A Gödel machine is a hypothetical self-improving computer program that solves problems in an optimal way. It uses a recursive self-improvement protocol in which it rewrites its own code when it can prove the new code provides a better strategy. The machine was invented by Jürgen Schmidhuber (first proposed in 2003), but is named after Kurt Gödel who inspired the mathematical theories (1931).

Automated Theorem Proving

Existing system for theorem proving allows us to find a solution of large class of mathematical problems.

- Metamath (http://us.metamath.org/)
- GPT-f - Open AI (miniF2F benchmark, automatically solving multiple challenging problems drawn from high school olympiads).
- Mizar system (formal language for writing mathematical definitions and proofs).
- Step by step solutions in Mathematica (WolframAlpha).

By using computational methods it is possible to prove large class of mathematical theorems.

Automated Code Generation Based on Mathematical Formulation

SAGA - Scientific Computing with Algebraic and Generative Abstractions. Algebraic software methodologies are a result of the last 20-30 years investigation into abstract data types and algebraic development techniques. The algebraic concepts also abstract modern program structuring mechanisms like classes and generic (or template) modules of object-oriented programming languages such as C++, Generic Java and Fortran-2000.

Curry–Howard correspondence - the direct relationship between computer programs and mathematical proofs.

Optimization of Machine Learning Algorithms

Formulation of machine learning problems can be optimized by using various available software. By using adaptive code optimization techniques it is possible to find optimal formulation of machine learning problems. Presented approach was successfully tested on various machine learning methods applied to sample classification problems.

- neural networks,
- random forest,
- k-nearest neighbors,
- decision tree

Optimization of Finite Element Formulation for Sample Engineering Problems

Self-adaptive computational methods can be applied for automated search of optimal shape function in finite element formulation of various engineering problems. In this approach it is possible to use various symbolic integration and differentiation techniques in order to automatically formulate finite element problems. Sample software for automated generation of FEM equations can be found online on the author's webpage.

Authors of this presentation are experts in the theory of equations with uncertain parameters, interval analysis, and related applications in engineering. Above described algorithm for the optimization of mathematical code and automated code generation can be applied for these class of problems. In this approach it is possible automatically compare results of various computational algorithms and find optimal algorithm for the solution of given problems. In general, it is possible to test various new mathematical formulations and related mathematical implementations. This will be a topic of future research.

Differentiation (Latex source)

```
\begin{equation}\cos\left(x\right)+x+2+\sin\left(2\right)\end{equation}
\begin{equation}\cos\left(x\right)+x+2+\sin\left(x\right)\end{equation}
\begin{equation}\cos\left(x\right)+x+2+\sin\left(\sin\left(x\right)\right)\end{equation}
\begin{equation}\cos\left(x\right)+x+2+\sin\left(\cos\left(x\right)\right)\end{equation}
\begin{equation}\cos\left(x\right)+x+2+\cos\left(2\right)\end{equation}
\begin{equation}\cos\left(x\right)+x+2+\cos\left(x\right)\end{equation}
\begin{equation}\cos\left(x\right)+x+2+\cos\left(\sin\left(x\right)\right)\end{equation}
\begin{equation}\cos\left(x\right)+x+2+\cos\left(\cos\left(x\right)\right)\end{equation}
\begin{equation}\cos\left(x\right)+x+2+2+2\end{equation}
\begin{equation}\cos\left(x\right)+x+2+2+x\end{equation}
\begin{equation}\cos\left(x\right)+x+2+2+\sin\left(x\right)\end{equation}
\begin{equation}\cos\left(x\right)+x+2+2+\cos\left(x\right)\end{equation}
\begin{equation}\cos\left(x\right)+x+2+x+x\end{equation}
\begin{equation}\cos\left(x\right)+x+2+x+\sin\left(x\right)\end{equation}
\begin{equation}\cos\left(x\right)+x+2+x+\cos\left(x\right)\end{equation}
\begin{equation}\cos\left(x\right)+x+2+\sin\left(x\right)+\sin\left(x\right)\end{equation}
\begin{equation}\cos\left(x\right)+x+2+\sin\left(x\right)+\cos\left(x\right)\end{equation}
\begin{equation}\cos\left(x\right)+x+2+\cos\left(x\right)+\cos\left(x\right)\end{equation}
\begin{equation}\cos\left(x\right)+x+x+x\end{equation}
```

Differentiation (step 1)

$$\cos(x) + (\cos(x) + \cos(2)) \tag{765}$$
$$\cos(x) + (\cos(x) + \cos(x)) \tag{766}$$
$$\cos(x) + (\cos(x) + \cos(\sin(x))) \tag{767}$$
$$\cos(x) + (\cos(x) + \cos(\cos(x))) \tag{768}$$
$$\cos(x) + (\cos(x) + (2 + 2)) \tag{769}$$
$$\cos(x) + (\cos(x) + (2 + x)) \tag{770}$$
$$\cos(x) + (\cos(x) + (2 + \sin(x))) \tag{771}$$
$$\cos(x) + (\cos(x) + (2 + \cos(x))) \tag{772}$$
$$\cos(x) + (\cos(x) + (x + x)) \tag{773}$$

Differentiation (step 2)

$$\frac{d}{dx}\cos(x) + \frac{d}{dx}(\cos(x) + \cos(2)) \tag{765}$$

$$\frac{d}{dx}\cos(x) + \frac{d}{dx}(\cos(x) + \cos(x)) \tag{766}$$

$$\frac{d}{dx}\cos(x) + \frac{d}{dx}(\cos(x) + \cos(\sin(x))) \tag{767}$$

$$\frac{d}{dx}\cos(x) + \frac{d}{dx}(\cos(x) + \cos(\cos(x))) \tag{768}$$

$$\frac{d}{dx}\cos(x) + \frac{d}{dx}(\cos(x) + (2 + 2)) \tag{769}$$

$$\frac{d}{dx}\cos(x) + \frac{d}{dx}(\cos(x) + (2 + x)) \tag{770}$$

$$\frac{d}{dx}\cos(x) + \frac{d}{dx}(\cos(x) + (2 + \sin(x))) \tag{771}$$

Differentiation (step 3)

$$(-1) \cdot \sin(x) + \left(\frac{d}{dx}\cos(x) + \frac{d}{dx}\sin(x)\right) \tag{762}$$

$$(-1) \cdot \sin(x) + \left(\frac{d}{dx}\cos(x) + \frac{d}{dx}\sin(\sin(x))\right) \tag{763}$$

$$(-1) \cdot \sin(x) + \left(\frac{d}{dx}\cos(x) + \frac{d}{dx}\sin(\cos(x))\right) \tag{764}$$

$$(-1) \cdot \sin(x) + \left(\frac{d}{dx}\cos(x) + \frac{d}{dx}\cos(2)\right) \tag{765}$$

$$(-1) \cdot \sin(x) + \left(\frac{d}{dx}\cos(x) + \frac{d}{dx}\cos(x)\right) \tag{766}$$

$$(-1) \cdot \sin(x) + \left(\frac{d}{dx}\cos(x) + \frac{d}{dx}\cos(\sin(x))\right) \tag{767}$$

Differentiation (step 4)

$$(-1)\cdot\sin(x) + \left((-1)\cdot\sin(x) + \left(\frac{d}{du}\cos(u)\right)\Big|_{u=\sin(x)}\cdot\frac{d}{dx}\sin(x)\right) \tag{767}$$

$$(-1)\cdot\sin(x) + \left((-1)\cdot\sin(x) + \left(\frac{d}{du}\cos(u)\right)\Big|_{u=\cos(x)}\cdot\frac{d}{dx}\cos(x)\right) \tag{768}$$

$$(-1)\cdot\sin(x) + \left((-1)\cdot\sin(x) + \left(\frac{d}{dx}2 + \frac{d}{dx}2\right)\right) \tag{769}$$

$$(-1)\cdot\sin(x) + \left((-1)\cdot\sin(x) + \left(\frac{d}{dx}2 + \frac{d}{dx}x\right)\right) \tag{770}$$

$$(-1)\cdot\sin(x) + \left((-1)\cdot\sin(x) + \left(\frac{d}{dx}2 + \frac{d}{dx}\sin(x)\right)\right) \tag{771}$$

$$(-1)\cdot\sin(x) + \left((-1)\cdot\sin(x) + \left(\frac{d}{dx}2 + \frac{d}{dx}\cos(x)\right)\right) \tag{772}$$

$$(-1)\cdot\sin(x) + \left((-1)\cdot\sin(x) + \left(\frac{d}{dx}x + \frac{d}{dx}x\right)\right) \tag{773}$$

Differentiation (step 5)

$$(-1)\cdot\sin(x)+((-1)\cdot\sin(x)+\cos(\sin(x))\cdot\cos(x)) \tag{763}$$
$$(-1)\cdot\sin(x)+((-1)\cdot\sin(x)+\cos(\cos(x))\cdot(-1)\cdot\sin(x)) \tag{764}$$
$$(-1)\cdot\sin(x)+((-1)\cdot\sin(x)+0\cdot 0) \tag{765}$$
$$(-1)\cdot\sin(x)+((-1)\cdot\sin(x)+(-1)\cdot\sin(x)) \tag{766}$$
$$(-1)\cdot\sin(x)+((-1)\cdot\sin(x)+(-1)\cdot\sin(\sin(x))\cdot\cos(x)) \tag{767}$$
$$(-1)\cdot\sin(x)+((-1)\cdot\sin(x)+(-1)\cdot\sin(\cos(x))\cdot(-1)\cdot\sin(x)) \tag{768}$$
$$(-1)\cdot\sin(x)+((-1)\cdot\sin(x)+(0+0)) \tag{769}$$
$$(-1)\cdot\sin(x)+((-1)\cdot\sin(x)+(0+1)) \tag{770}$$
$$(-1)\cdot\sin(x)+((-1)\cdot\sin(x)+(0+\cos(x))) \tag{771}$$
$$(-1)\cdot\sin(x)+((-1)\cdot\sin(x)+(0+(-1)\cdot\sin(x))) \tag{772}$$
$$(-1)\cdot\sin(x)+((-1)\cdot\sin(x)+(1+1)) \tag{773}$$

Expression Evaluation

step-1
$(((1+2)+((1+2)-(3+4)))+(5+(6+7)))=((18-1)+0)$
step-2
$(((1+2)+((1+2)-(3+4)))+(5+(6+7)))=(17+0)$
step-3
$((3+((1+2)-(3+4)))+(5+(6+7)))=((18-1)+0)$
step-4
$(((1+2)+((1+2)-(3+4)))+(5+13))=((18-1)+0)$
step-5
$(((1+2)+(3-(3+4)))+(5+(6+7)))=((18-1)+0)$
step-6
$(((1+2)+((1+2)-7))+(5+(6+7)))=((18-1)+0)$
step-7
$(((1+2)+((1+2)-(3+4)))+(5+(6+7)))=17$
step-8
$((3+((1+2)-(3+4)))+(5+(6+7)))=(17+0)$
step-9
$(((1+2)+((1+2)-(3+4)))+(5+13))=(17+0)$
step-10
$(((1+2)+(3-(3+4)))+(5+(6+7)))=(17+0)$

Expression Evaluation

step-93
$((3+(-4))+18) = (17+0)$
step-94
$((-1)+(5+13)) = (17+0)$
step-95
$((-1)+18) = ((18-1)+0)$
step-96
$((3+(-4))+18) = 17$
step-97
$((-1)+(5+13)) = 17$
step-98
$((-1)+18) = (17+0)$
step-99
$17 = ((18-1)+0)$
step-100
$((-1)+18) = 17$
step-101
$17 = (17+0)$
step-102
$17 = 17$

Expression evaluation is possible without specifying any explicit method for expression evaluation.

Automatically Generated Latex Report


```
step-1\\
$x=(x+(2+x))$
\newline
step-3\\
$x=(x+2+x)$
\newline
step-4\\
$x=(x+2+x)$
\newline
step-5\\
$x=x+2+x$
\newline
step-6\\
$x=x+2+x$
\newline
step-7\\
$x=2+2*x$
\newline
step-9\\
$x+(-1)*x+(-1)*x=2$
```

Automatically Generated PDF Reports

Hundred/thousand/unlimited number of examples:

51810	data-node-126.pdf
52690	data-node-127.pdf
52354	data-node-128.pdf
52785	data-node-130.pdf
52603	data-node-131.pdf
52512	data-node-132.pdf
52392	data-node-134.pdf
52565	data-node-135.pdf
53310	data-node-136.pdf
50225	data-node-14.pdf
52728	data-node-142.pdf
51848	data-node-143.pdf
52303	data-node-144.pdf
52666	data-node-146.pdf
52696	data-node-147.pdf
52484	data-node-148.pdf
50286	data-node-15.pdf
52539	data-node-150.pdf

http://andrew.pownuk.com/research/AlgebraicEquations/

Simplification of the Solution

step-1
$(4+6) = (x+x)$
step-3
$4+6 = (x+x)$
step-7
$4+6 = x+x$
step-11
$4+6 = 2*x$
step-15
$4+6+(-2)*x = 0$
step-21
$(-2)*x = 0+(-1)*4+(-1)*6$
step-24
$(-2)*x = (-10)$
step-25
$x = ((-10)/(-2))$
step-26
$x = 5$

Conclusions

- Mathematical/scientific knowledge can be treated as independent units that can interact with each-other and create new, possibly useful knowledge.
- In some cases, generation of new knowledge can be fully autonomous. Presented approach can be applied for automated development of large class of existing algorithms (most of the algorithms known from textbooks).
- Development of new knowledge is possible in many different fields (e.g. engineering, computer science, mathematics etc.).

Conclusions (continued)

- By using presented methodology it is possible to perform automated research in mathematics, engineering, and computer science.
- Adaptive computational methods can be used for finding optimal solution of various engineering problems with uncertain parameters.
- In the most general case it is possible to apply fully self-adaptive computational methods and automatically perform research in various field of mathematics and engineering applications. In particular, in the theory of equations with uncertain parameters. Presented methodology is not limited to any particular computational algorithm.

Conclusions (continued)

- Once the information is available in the system it will NEVER be forgotten and can be used for generation of new mathematical theorems. From that perspective the system can be viewed as self-organizing archive of information.
- Development of this and similar systems should speed up cooperation among scientists around the world (theoretical possibility).

Conclusions (continued)

- Calculations can be done in distributed way (this option is experimental at this moment). Large number of computers can process the information simultaneously in order to get the results faster. Calculations do not require existence of any centralized system.
- Parallel computing can significantly speed up the calculations (future work).
- Autonomous interaction with external sources of information extends internal database of information and should increase productivity of the system (future work).

Thank You