

Abstract

The project aims at researching the principles and various techniques for representing large mathematical expressions in summarised form so that their structure becomes apparent by concealing the details to a certain degree.

We focus on efficiency in terms of: (1) computational representation; (2) graphical representation; and (3) preservation of semantic content, which determines the perceiving and understanding of mathematical expressions.

The practical outcome of the project is the tool **OpenMath Browser**. It is based on **OpenMath** and offers a set of functionalities demonstrating the main summarisation principles and can be used as a starting point for testing and further developments.

Introduction

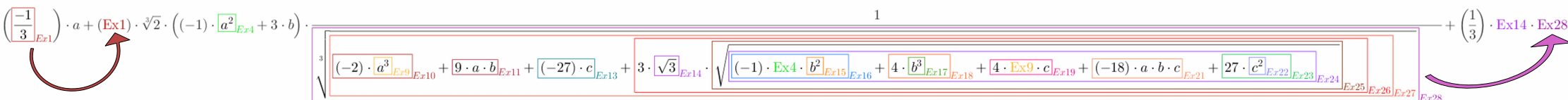
In areas such as education in Mathematics or communicating mathematical expressions to visually impaired it is vital to provide means for representing the abstract structure as well as navigating the expressions in order to improve their understanding.

The problem of representing expressions in suitable form is general and concerns all Computer Algebra Systems (CASs). We consider the problem in its generic dimensions, i.e. CAS independently.

The intended use of the **OpenMath Browser** is to supplement any OpenMath speaking system by facilitating the understanding of the output from mathematical manipulations.

Examples

▷ Labelling repeating subexpressions. **Questions:** What subexpressions need to be labelled and what do not? How to visualise the abstract structure of expressions?

$$\left(\frac{-1}{3}\right)_{Er1} \cdot a + (Ex1)_{Er1} \cdot \sqrt[3]{2} \cdot ((-1) \cdot a^2)_{Er4} + 3 \cdot b \cdot \sqrt[3]{(-2) \cdot a^3}_{Er9} + 9 \cdot a \cdot b_{Er11} + (-27) \cdot c_{Er13} + 3 \cdot \sqrt{3} \cdot \sqrt[3]{(-1) \cdot Ex4 \cdot b^2}_{Er15} + 4 \cdot b^3_{Er17} + 4 \cdot Ex9 \cdot c_{Er19} + (-18) \cdot a \cdot b \cdot c_{Er21} + 27 \cdot c^2_{Er22} + (-1) \cdot Ex14 \cdot Ex28$$


▷ Suppressing and labelling large (sub)expressions. **Questions:** How to define “large” expressions? How to label missing output?

(a) E.g. large polynomials: $1 + x + x^2 + \llong Long:494 \gg + x^{497} + x^{498} + x^{499}$

(b) Large matrices: $Det(\llong Matrix:1000 \times 1000 \gg)$

Prior art

The problem is not well researched and only a few CASs offer some basic system specific summarising functionalities. (Refer to products’ documentation for further details.)

Wolfram Mathematica 7 

- ▷ Middle part of long expressions in linear form (e.g. polynomial) is suppressed and labelled;
- ▷ Large 2D output (e.g. matrices) is suppressed and labelled;
- ▷ Options to vary the degree of summarisation.

MwxMaxima

- ▷ Package **format** for Macsyma/Maxima provides means for manipulating the structure of expressions by targeting subparts matching some pattern and manipulating them.

Maple 13

- ▷ Suppresses and labels large output. Labels contain description of the expression.
- ▷ Procedures **evalindets** and **subsindets** to transform subexpressions matching some description.

MATLAB

- ▷ Matlab’s symbolic toolbox provides function **subexpr** for rewriting a symbolic expression in terms of common subexpressions.

Future work

- Investigating and applying efficient methods for equality testing considering mathematical equality, characteristics of functions, etc.
- Investigating further efficiency in terms of graphical and computational representation, as well as in terms of semantic content and understandability.

Objectives

- ▷ To outline key concepts such as computational, graphical and semantic efficiency of representation, by defining the notions of “large” expressions, abstract structure, equality between expressions;
- ▷ To describe, implement and test various techniques for intelligent summarising and browsing of mathematical expressions: denoting common subexpressions, using descriptive labelling of large expressions in length or in depth with respect to the tree representation, options for varying the degree of summarisation;
- ▷ **Practical outcome:** to develop a system demonstrating the summarisation methods, which is easy to extend and provides an environment for further testing and improvements.

Results

The main practical result of the project is the tool **OpenMath Browser** for summarising and browsing mathematical expressions. It offers the following capabilities:

- Reading and parsing OpenMath 2.0 input in XML format (using the RIACA library*);
- Performing summarisation and expansion on the OpenMath object;
- Translating the OpenMath object into L^AT_EX (using an OpenMath to L^AT_EX phrasebook) and rendering mathematical expressions in L^AT_EX (using J^Lat_exMath package*).

Fig. 1 shows examples for efficiency improvements in representation: Graphical (width of the expression as an image); OpenMath (number of symbols in the file); and Computational (number of distinct objects in the data structure).

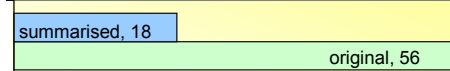
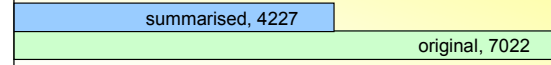
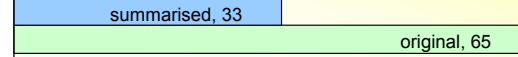

Image Width, cm	summarised, 18	original, 56
		
Number of Symbols	summarised, 4227	original, 7022
		
Number of OMOjects	summarised, 33	original, 65
		

Figure 1. Representation of a solution to $x^3 + a \cdot x^2 + b \cdot x + c = 0$.

Examples 

* Third party packages.

The area is open to research and further development which might include:

- Improving methods, applying and testing new approaches for summarising various types of expressions;