

Towards Intelligent Summarising and Browsing of Mathematical Expressions

Ivelina Stoyanova

I.Stoyanova@alumni.bath.ac.uk

Department of Computer Science
University of Bath, Bath BA2 7AY
United Kingdom

Abstract. Most computer algebra systems, by default, output the result of the symbolic computations in expanded form with all the details which usually makes the result difficult to read. Many systems have some techniques to alleviate this difficulty, but the techniques are usually system-specific, and often not programmable.

This paper describes the application `OpenMath Browser`. Its primary purpose is to serve as a tool for demonstrating and testing of summarisation and browsing approaches. Being based on OpenMath as its input, it is not system-specific, and can serve as a basis for experiments into the correct way(s) of displaying, and interacting with, mathematical expressions independently of their origin.

A demo version of the application, described in section 4 and [5], can be seen at:

<http://staff.bath.ac.uk/masjhd/OMBrowser/OpenMathBrowser.html>.

1 Introduction

The main focus of the work presented here is to investigate possible approaches and techniques for efficient graphical representation of large mathematical expressions aiming at improvement of their understanding. This involves summarisation of the expression so that its abstract structure is revealed, and further providing means for browsing the structure and expanding the components of the expression to a varying degree.

The demonstrator reads expressions encoded in OpenMath [1] to ensure system independence. A production version would doubtless use an OpenMath domain object model interface to the algebra system rather than text representations, but for prototyping purposes a text interface is sufficient.

We see several applications for a tool such as ours.

- Maths education. Summarising long mathematical expressions can demonstrate their building components and thus facilitate perception of ideas and understanding of mathematical manipulations.
- Research in Mathematics. Navigating complicated expressions can help give an overview of the result rather than forcing attention to the details.

- Development of tools for presenting mathematical expressions to the visually impaired. In recent years several tools for navigating mathematical expressions were developed for visually impaired people. The specific task of these tools is to represent expressions with various degree of detail corresponding to the natural, intuitive way in which the user perceives mathematical expressions.

2 Prior art

Some computer algebra and other systems offer a limited number of functionalities related to summarisation and navigation of large mathematical expressions. The results of a detailed overview of such functionalities and systems is presented briefly below.

In **Mathematica** very large expressions are displayed in a nested interface allowing for refining the level of detail of the output. It also provides function **Short** which can be used directly for finer control over the display of expressions. For example, it can be used to shorten output which is not large enough for the default suppression to take place.

Mathematica also allows sparse storing of **SparseArray** which can be used for arrays, matrices and vectors. For displaying large matrices or when some of the entries are large the same principles as outlined above are applied.

Maple outputs matrices with a dimension greater than 10 for worksheet and 25 for command-line version, in a summarised form, providing a brief description of the matrix and offering the option to view the matrix in a separate window.

In a sense, the construct **RootOf** is also a form of summarisation, as it represents any element of a set of solutions, which may be very large. Maple also provides procedures **evalindets** and **subindets** which allow to transform all subexpressions of a given type or matching some description.

The package **format** for the computer algebra system **Macsyma** provides means for user-directed hierarchical structuring of expressions with navigation options. It also allows directing certain simplifications and manipulations over selected subexpressions matching a template.[4]

The symbolic toolbox of **Matlab** offers the command **subexpr** which allows rewriting some symbolic expression in terms of common subexpressions. Further, the command **subs** can be used to perform symbolic substitution in the expression.

In recent years a great deal of research was invested in developing tools for representing mathematical expressions in a form suitable for visually impaired people. The significance of “visual syntax” was pointed out, that is the spatial location of symbols and groups of symbols on the page which facilitates parsing the expression first and then helps building a strategy for solving. It is also important for the user to be able to access the structure of the expression and to browse its parts in detail. [3]

3 Technical details

3.1 Consideration of efficiency

Efficiency with respect to the representation of large mathematical objects can be described in terms of the following:

- Efficiency of the graphical representation. This may be measured by the size of the displayed expression, more specifically, the area of the box it is displayed in.
- Efficiency in terms of time, in particular the time it takes to display the expression.
- Efficiency in terms of storage required for the mathematical object and any additional resources necessary for its processing.
- Efficiency in terms of semantics. It is important to find the balance between the richness of encoding and the economy of graphical output. This involves considerations of the particular task the analysis of the expression is involved in.

E.g. if we need to know how many different solution the equation

$$(x - 1)^2(x - \sqrt{y^5 + y^4 + y^3 + y^2 + y + 1})(x - \sqrt[3]{y^5 + y^4 + y^3 + y^2 + y + 1})$$

has, an output of the form A, A, B, C is sufficient. However, if we need to see what the form of the actual solutions is, we need more details, for example $1^2, \sqrt{D}, \sqrt[3]{D}$ (where 1^2 means “1 with multiplicity 2”).

Intelligent operability is closely related to efficiency, especially in terms of semantics. We aim at finding representation of mathematical expressions which facilitates human perception and understanding of mathematical content.

3.2 Equality between mathematical expressions

The problem of equality is one of the fundamental problems of the project and in computer algebra in general (see [2]). We distinguish between the following types of equality:

- Data structure equality: two mathematical objects are considered equal if they are represented by identical data structures.
- Equality by reference: this is the equality between an object and a reference to it, or equality between two references to the same object.
- Mathematical equality: equality between two mathematical objects which can be proved by mathematical means (manipulations, application of axioms, theorems, etc.).

The following proposition appears in [2] and it points to a possible approach in handling equality.

Proposition 1. *If the representation is canonical, then mathematical equality (in O) is the same as data structure equality (in R).*

Defining a canonical representation of OpenMath objects also depends on their mathematical characteristics and it is not easy to do and even not always possible. E.g. for polynomials ordering on monomials can be introduced, but it is far more complicated to define canonical form of some elementary functions (see examples in [2]).

At present we only consider data structure and referential equality.

4 OpenMath Browser

4.1 Purpose of OpenMath Browser

The main purpose of an application for intelligent summarising and browsing of mathematical expressions is to provide means for representing mathematical content in a form facilitating its understanding and to allow the user to adjust the representation to their needs. With regard to this the suggested use of a summarising and browsing tool is as a supplement to any computer algebra system or any other software for mathematical manipulations.

However, at present the main role of **OpenMath Browser** is to demonstrate, test and evaluate the performance of various techniques for summarisation, navigation and display need.

OpenMath Browser is fully developed in Java using a set of external libraries: the RIACA library for parsing OpenMath input in XML format; a phrasebook for translating the OpenMath object into L^AT_EX; the library J_LatexMath for rendering L^AT_EXcode developed by Scilab.

4.2 Options for summarisation and characteristics of labels

An extensive set of options was constructed to allow adjustment of the summarisation and display of expressions in order to enable observations and evaluation of different approaches.

The options for summarisation offered by **OpenMath Browser** are the following.

- Maximum height to display: all expressions of a greater height are suppressed and labeled.
- Maximum number of arguments: expressions with larger number of arguments are suppressed and labeled.

The options for display offered by **OpenMath Browser** are the following.

- Using colours for labels: which would facilitate distinguishing labels visually.
- Option to suppress first occurrence of repeated expressions. In the case when it is not suppressed first occurrence is placed in a box with the label subscripted.
- Option to use the name of the symbol (i.e. `name` attribute of the `OMSymbol`) of the expression. When this option is selected the label contains the name of the symbol and the index of the expression in the hash table.

- There are different labels for repeating expressions, those with large height and those with large width.

One of the main problems of the summarisation is the use of suitable and informative labels. On one hand, labels replace some expression and the requirement for effectiveness of graphical representation implies that they should be at least as short as the expression they stand for. On the other hand, it is desirable that they provide some relevant information or a description of the expression.

Labels we use in the application satisfy the following conditions:

- different expressions are replaced by different labels;
- equivalent expressions are replaced by the same labels (mathematical equality is excluded);
- labels contain the unique ID of the replaced expression which is its index in the hash table;
- labels may contain information about the mathematical operation of the expression;
- labels may contain information about the position of the node representing this expression in the tree, i.e. the height of the node;
- labels may contain the size of the omitted elements of the expression.

4.3 Summarising and browsing functionalities

The expression can be summarised fully by labeling all repeating subexpressions. Expressions (subexpression) for which the maximum width or height values set in the options are exceeded, are automatically summarised to comply with the set options and then they can be seen gradually expanded.

The following operations can be performed on the expression:

- Full summarisation.
- Customised summarisation - by choosing only specific expressions to summarise.
- Customised expansion - same as the above but choosing which expressions to expand.
- Full expansion.

A demo for **OpenMath Browser** in the form of a Java applet can be accessed at the following address:

<http://staff.bath.ac.uk/masjhd/OMBrowser/OpenMathBrowser.html>. A fuller description is in [5]. The demo offers access to a set of examples which demonstrate the main principles implemented for summarisation and browsing of mathematical expressions.

4.4 User feedback

Some informal user feedback was received with respect the functionalities and appearance of **OpenMath Browser**. The application is still in prototype phase and the detailed user evaluation is a future task.

However, the feedback was used to determine the set of default options:

- Colours are considered helpful for noticing repeated expressions.
- Some users find it better to first see the expression in full expanded form and then decide which options to choose. Thus default values for maximum width and height are set big.
- Suppressing the first occurrence is preferred rather than displaying it in a box with the label as a subscript.
- The option of displaying information about the mathematical operation in the labels is not found relevant.

Some additional notes from users were also taken into account although not addressed at present: labels are considered long and odd; better error handling is needed; better navigation from one part of the expression to others or to additional definitions.

5 Examples

The first example is also the first example in the on-line demonstration version of the tool.

Example 1. The solution to the general cubic equation

$$x^3 + ax^2 + bx + c = 0. \quad (1)$$

This can be presented as:

$$\frac{1}{6} \sqrt[3]{36ba - 108c - 8a^3 + 12\sqrt{12b^3 - 3b^2a^2 - 54bac + 81c^2 + 12ca^3}} - \frac{2b - \frac{2}{3}a^2}{\sqrt[3]{36ba - 108c - 8a^3 + 12\sqrt{12b^3 - 3b^2a^2 - 54bac + 81c^2 + 12ca^3}}} - \frac{1}{3}a$$

and using Tschirnhaus transformation and substituting x by $x - \frac{a}{3}$, we obtain the following cubic equation:

$$x^3 + b'x + c',$$

where $b' := b - \frac{1}{3}a^2$ and $c' := \frac{2}{27}a^3 - \frac{1}{3}ba + c$, and the solution can be represented as:

$$\frac{1}{6}T - \frac{2b'}{T},$$

where

$$S := \sqrt{12b'^3 + 81c'^2},$$

$$T := \sqrt[3]{108c' + 12S}.$$

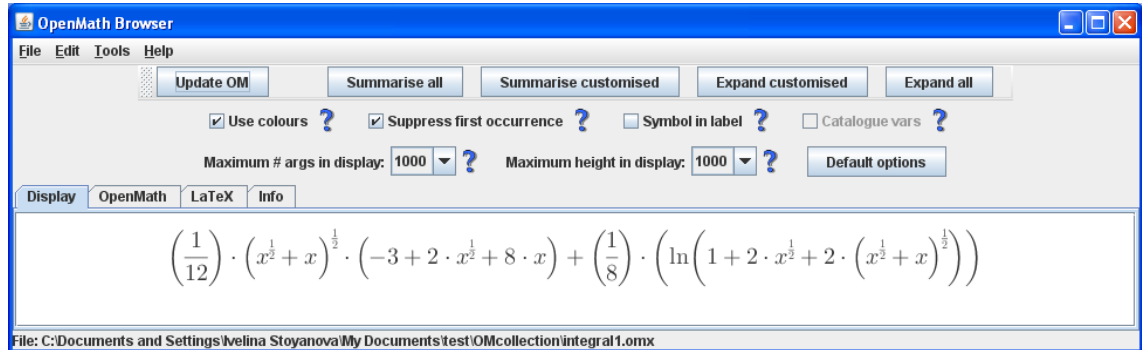


Fig. 1. Simple example (2): original expression.

Example 2. To demonstrate the tool we have chosen a rather simpler formula.

$$\frac{1}{12} (x^{\frac{1}{2}} + x)^{\frac{1}{2}} (-3 + 2 \cdot x^{\frac{1}{2}} + 8 \cdot x) + \frac{1}{8} \ln \left(1 + 2 \cdot x^{\frac{1}{2}} + 2 (x^{\frac{1}{2}} + x)^{\frac{1}{2}} \right) \quad (2)$$

Figure 1 presents this as our tool displays it, without any summarisation, while Figure 2 presents the default fully-summarised behaviour. It is consistent with the traditional mathematical “expression in α and β where $\alpha = \dots$ and $\beta = \dots$ ”. Figure 3 shows an alternative summarised representation where each expression is displayed in full the first time it is used. The choice of the first as the default is based on user preference (see 4.4).

The reader is liable to think, with justification, that these are overkill, so the next few figures (4-6) present variants in which (manually, but see the conclusions) we have suppressed the summarisation of some of the smaller components.

Example 3. Figure 7 presents a long polynomial, without shared sub-expressions. Here our strategy is to admit defeat and just print the first and last few terms so that the new expression satisfies the maximum number of arguments requirement, deferring the rest (the middle terms) until the next line, and so on recursively, as shown in Figure 8. This behaviour is similar to the way long polynomials are presented in **Mathematica**.

Example 4. Figures 9 and 10 present the original and the summarised form of the Sylvester Matrix of the following polynomials in x :

$$p(x) = (y^5 + y^4 + y^3 + y^2 + y + 1) x^3 + (y^4 + y^3 + y^2 + y + 1) x^2 + (y^3 + y^2 + y + 1) x + (y^2 + y + 1)$$

and

$$q(x) = \left(\frac{1}{y^4} + \frac{1}{y^3} + \frac{1}{y^2} + \frac{1}{y} + 1 \right) x^2 + \left(\frac{1}{y^3} + \frac{1}{y^2} + \frac{1}{y} + 1 \right) x + \frac{1}{y^2} + \frac{1}{y} + 1.$$

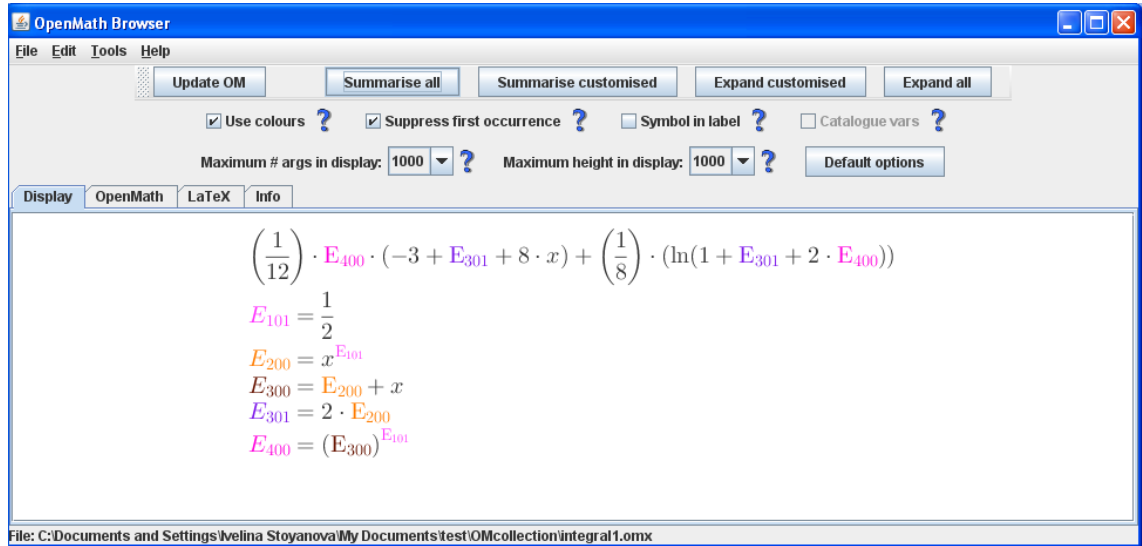


Fig. 2. Simple example (2): fully summarised expression with first occurrence of repeated subexpressions suppressed.

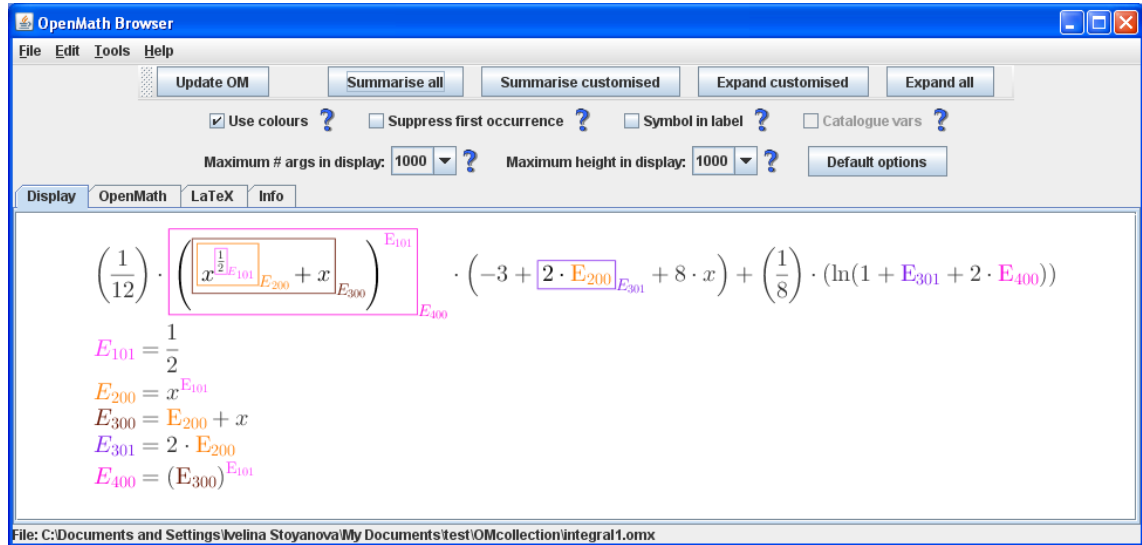


Fig. 3. Simple example (2): fully summarised expression with first occurrence of repeated subexpressions displayed.

Example 5. Figure 11 presents a large matrix of dimension 50×50 too large to display on the screen. In the case when both number of rows and columns exceed

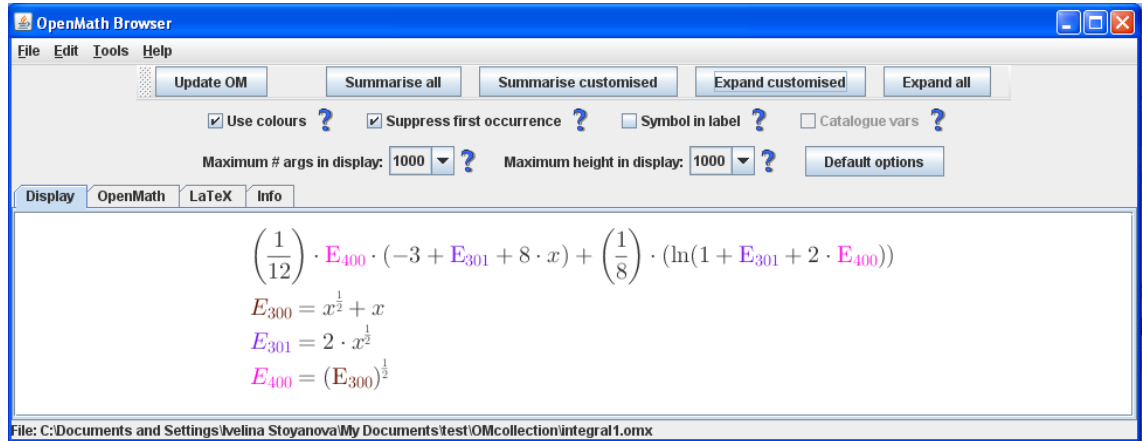


Fig. 4. Simple example (2): partially expanded expression.

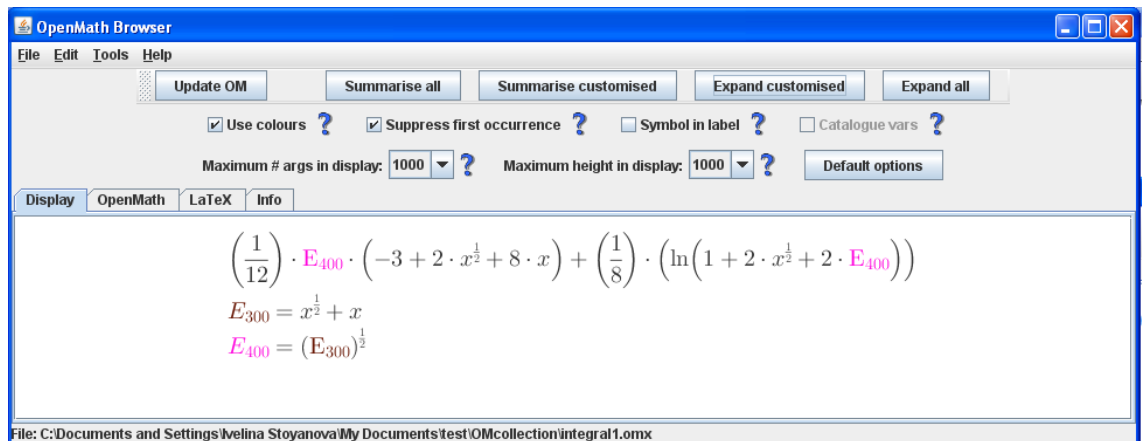


Fig. 5. Simple example (2): partially expanded expression.

the maximum number of arguments to be displayed¹ the matrix is summarised as shown on Figure 12.

However, when only the number of rows exceeds the limit, the default summarising technique for long expressions is applied and the matrix is represented as on Figure 13.

Example 6. The final example presents the combined approach to summarisation of large mathematical expressions. The default summarising by labeling repeated subexpressions is shown on Figure 14. This approach does not provide sufficiently

¹ In OpenMath the matrix is represented as an application object of type matrix containing 50 arguments (rowmatrix) each of which has 50 arguments as well.

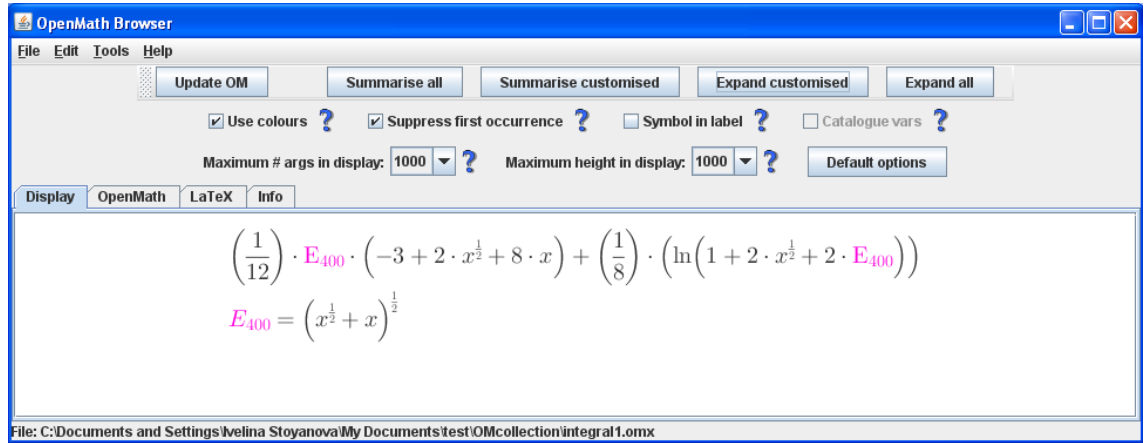


Fig. 6. Simple example (2): partially expanded expression.

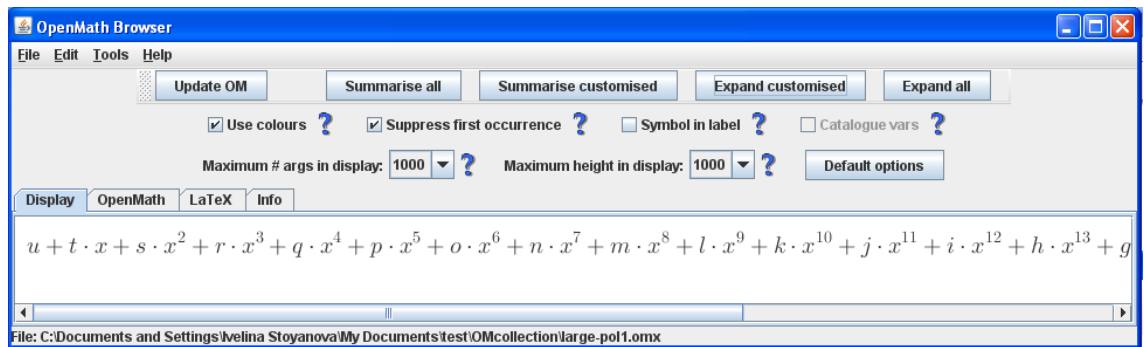


Fig. 7. Long polynomial: original expression.

efficient form of displaying the expression so that its structure is visible. We can vary the maximum number of arguments allowed and obtain the result on Figure 15 and further on Figure 16 where the expression fits within the window.

Alternatively, we can try and vary the maximum height² as well in which case we obtain the representation on Figure 17.

6 Conclusion and future work

We have presented a highly customisable tool for displaying OpenMath expressions with varying degrees of summarisation. The tool has a variety of options — probably too many for the naïve user. We have not attempted in *this* project to discover the “best” summarisation, which would require the “intelligence” mentioned in the title. Some points are relatively obvious, others less so.

² That is the height of the tree representation of the (sub)expressions.

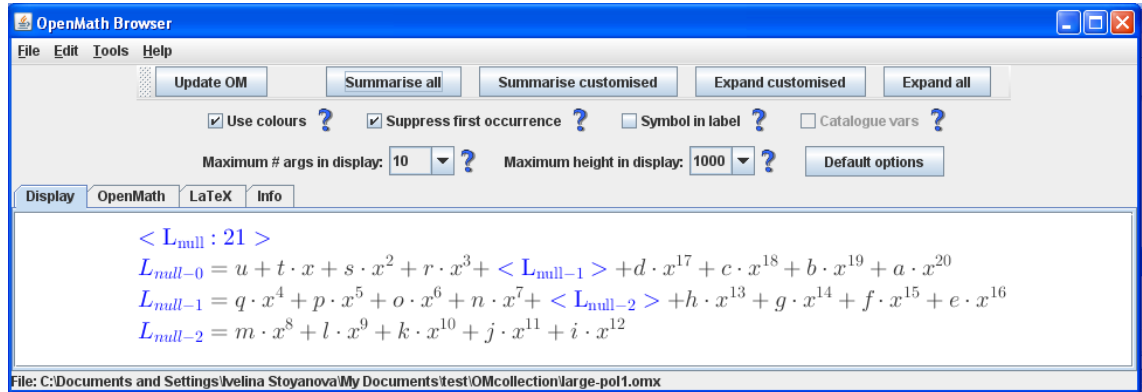


Fig. 8. Long polynomial: summarised expression.

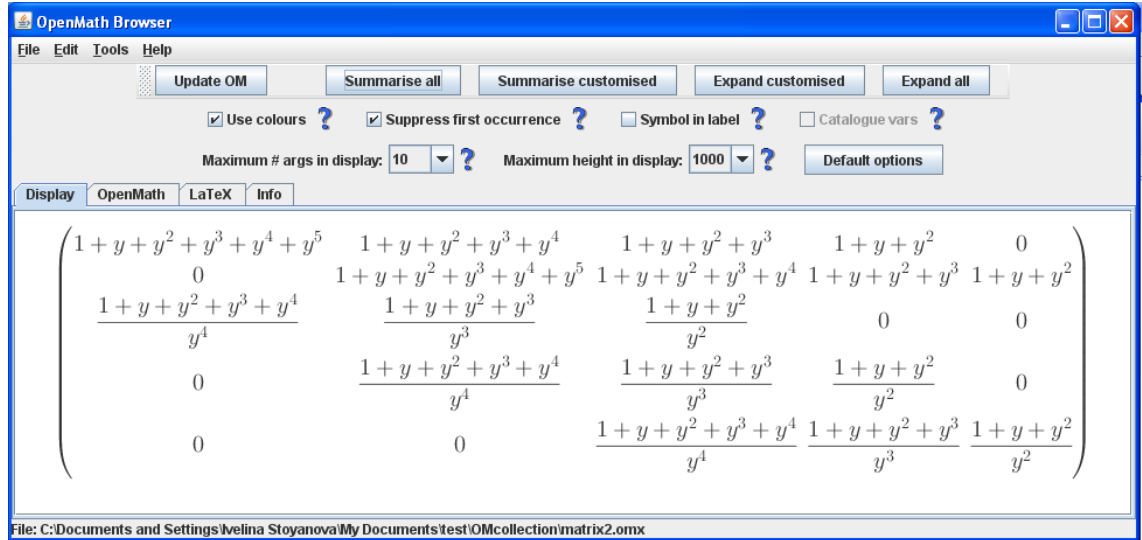


Fig. 9. Matrix: original.

- ✓ Do not, by default, replace sub-expressions by *longer* labels, e.g. the $\frac{1}{2}$ represented as E_{101} in figure 3.
- ✓ Do not, by default, use ‘common’ sub-expressions which are no longer common when the full DAG has been formed, e.g. E_{300} in figure 2.
- * We say “by default” in the above two because the user *might* be interested in all common structure.
- ✓ Not making explicit all the multiplication signs in the expression — the challenge lies, as always, in deciding *which* can be elided.
- ? Adjust the number of terms printed at either end of a “long” expression — see Example 3 (Figures 7 and 8). Here again the key question is “how many

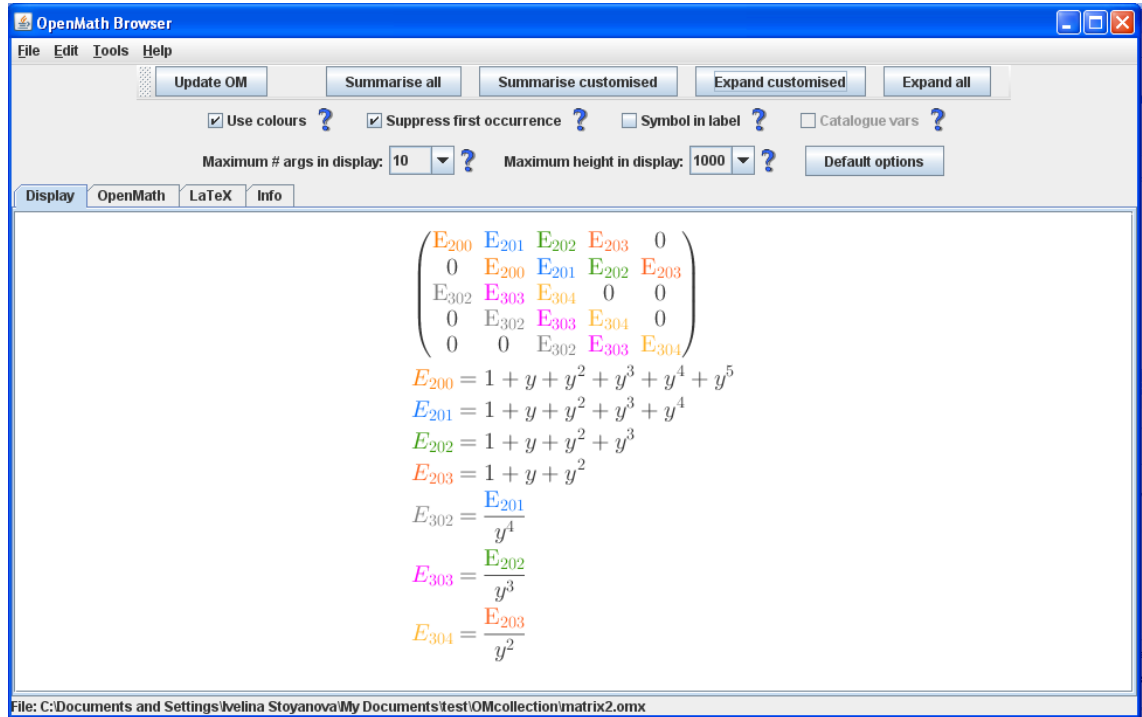


Fig. 10. Matrix: summarised form.

terms". In some cases efficiency of representation is also a question of whether to start from inside out or vice versa (e.g. start expanding $E_{\text{null}-2}$ rather than $E_{\text{null}-0}$ in Figure 8).

- ? Better default behaviour for matrix displays — see Figures 9–13.
- ? Better user interface — accessible display area and interactive access to subexpressions (e.g. enable hyperlinks). Currently the library `JLatexMath` is used for rendering \LaTeX which outputs an icon and activating the display area is a future task.
- ? Allow more flexibility for summarisation and navigation — sometimes it may be required to treat differently particular occurrences of expressions and navigation via controls (e.g. next, previous, up, etc.) may be more efficient.
- ? Consider mathematical equality.

Acknowledgements: **OpenMath Browser** is developed as a part of the author's B.Sc. dissertation ([5]) under the supervision and with the support of Prof. James Davenport.

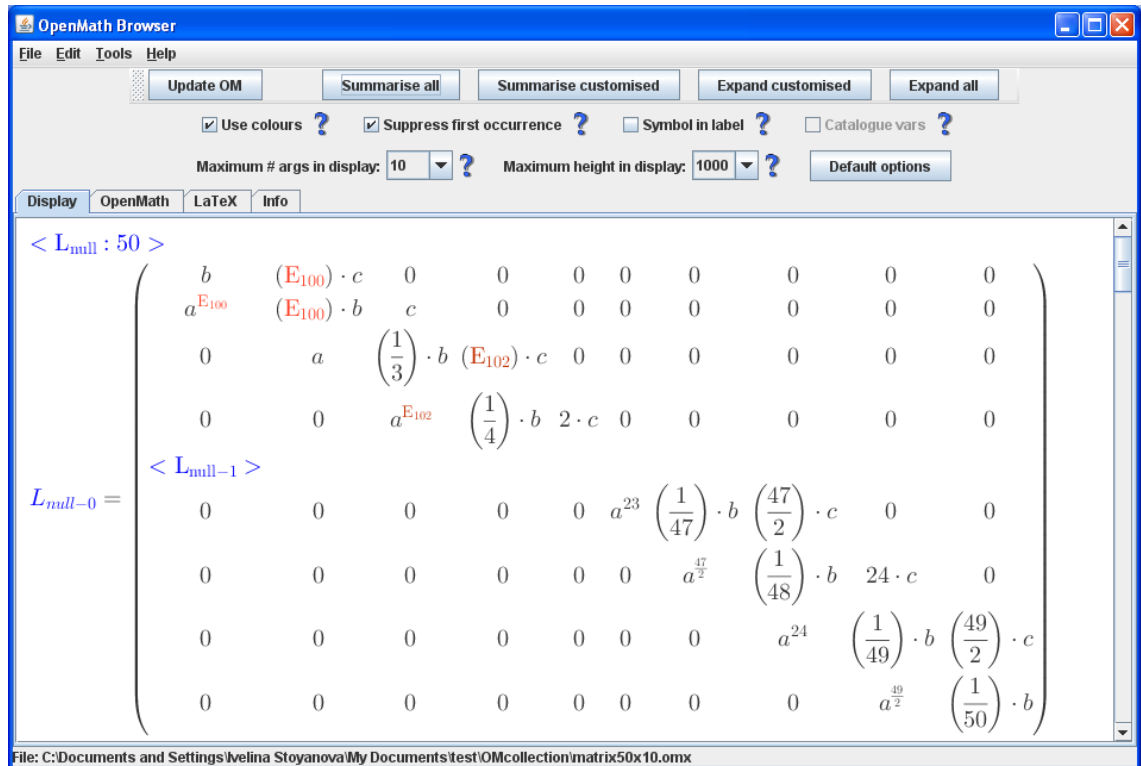


Fig. 13. Matrix: summarised form.

3. A. D. N. Edwards and R. D. Stevens. Mathematical representations: Graphs, curves and formulas. In *Proceedings of the INSERM Colloquium, Non-Visual Human Computer Interactions*, pages 181–193, 1993. http://reference.kfupm.edu.sa/content/m/a/mathematical_representations__graphs__cu_2321607.pdf.
4. Bruce R. Miller. An Expression Formatter for Macsyma, 1995. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.50.9838>.
5. Ivelina Stoyanova. Intelligent Summarising and Browsing of Mathematical Expressions. B.Sc. Dissertation, Department of Computer Science, University of Bath, 2010.

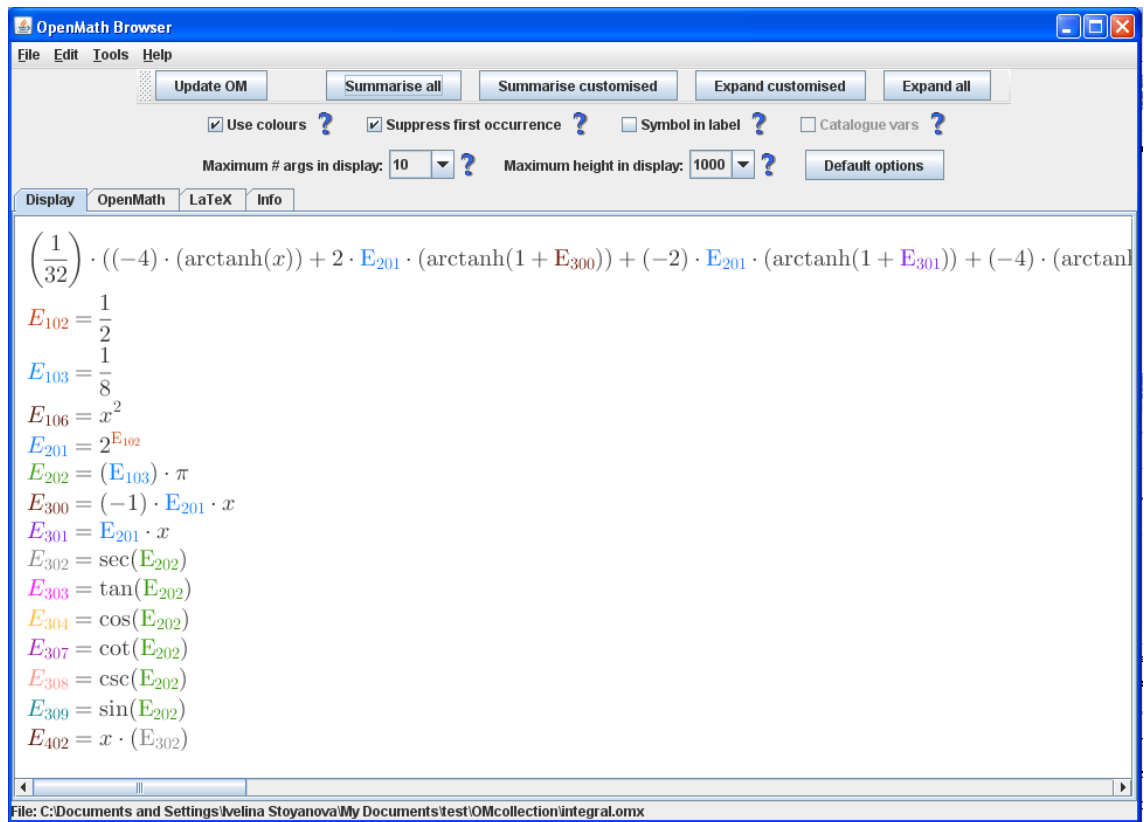


Fig. 14. Long expression: default summarised form.

The screenshot shows the OpenMath Browser window with the following content:

File Edit Tools Help

Update OM Summarise all Summarise customised Expand customised Expand all

Use colours ? Suppress first occurrence ? Symbol in label ? Catalogue vars ?

Maximum # args in display: 5 ? Maximum height in display: 1000 ? Default options

Display OpenMath LaTeX Info

$$\left(\frac{1}{32}\right) \cdot (< L_{800} : 15 >)$$

$$L_{800-0} = (-4) \cdot (\operatorname{arctanh}(x)) + 2 \cdot 2^{\frac{1}{2}} \cdot (\operatorname{arctanh}(1 + E_{300})) + < L_{800-1} > + 2 \cdot (\ln(1 + x^2 + (-2) \cdot x \cdot (E_{309}))) \cdot (E_{309})$$

$$L_{800-1} = (-2) \cdot 2^{\frac{1}{2}} \cdot (\operatorname{arctanh}(1 + 2^{\frac{1}{2}} \cdot x)) + (-4) \cdot (\operatorname{arctanh}(E_{402} + (-1) \cdot (E_{303}))) \cdot (E_{304}) + < L_{800-2} > + 4 \cdot (\operatorname{arctanh}(E_{402} + (-1) \cdot (E_{303}))) \cdot (E_{304}) + 2 \cdot (\ln(-1 + x)) + < L_{800-3} > + 2 \cdot (E_{304}) \cdot (\ln(1 + x^2 + (-2) \cdot x \cdot (E_{309})))$$

$$L_{800-2} = (-4) \cdot (\operatorname{arctanh}(E_{402} + (-1) \cdot (E_{303}))) \cdot (E_{304}) + 2 \cdot (\ln(-1 + x)) + < L_{800-3} > + 2 \cdot (E_{304}) \cdot (\ln(1 + x^2 + (-2) \cdot x \cdot (E_{309})))$$

$$L_{800-3} = (-2) \cdot (\ln(1 + x)) + 2^{\frac{1}{2}} \cdot (\ln(1 + E_{300} + x^2)) + (-1) \cdot 2^{\frac{1}{2}} \cdot (\ln(1 + 2^{\frac{1}{2}} \cdot x + x^2))$$

$$E_{202} = \left(\frac{1}{8}\right) \cdot \pi$$

$$E_{300} = (-1) \cdot 2^{\frac{1}{2}} \cdot x$$

$$E_{302} = \sec(E_{202})$$

$$E_{303} = \tan(E_{202})$$

$$E_{304} = \cos(E_{202})$$

$$E_{307} = \cot(E_{202})$$

$$E_{308} = \csc(E_{202})$$

$$E_{309} = \sin(E_{202})$$

$$E_{402} = x \cdot (E_{302})$$

File: C:\Documents and Settings\velina Stoyanova\My Documents\test\OMcollection\integral.omx

Fig. 15. Long expression: default summarised form with reduced number of terms displayed.

OpenMath Browser

File Edit Tools Help

Update OM Summarise all Summarise customised Expand customised Expand all

Use colours ? Suppress first occurrence ? Symbol in label ? Catalogue vars ?

Maximum # args in display: 4 ? Maximum height in display: 1000 ? Default options

Display OpenMath LaTeX Info

$$\left(\frac{1}{32}\right) \cdot \langle L_{800} : 15 \rangle$$

$$L_{800-0} = (-4) \cdot (\operatorname{arctanh}(x)) + \langle L_{800-1} \rangle + (-2) \cdot (\ln(1 + x^2 + 2 \cdot x \cdot (E_{309}))) \cdot (E_{309})$$

$$L_{800-1} = 2 \cdot 2^{\frac{1}{2}} \cdot (\operatorname{arctanh}(1 + E_{300})) + \langle L_{800-2} \rangle + 2 \cdot (\ln(1 + x^2 + (-2) \cdot x \cdot (E_{309}))) \cdot (E_{309})$$

$$L_{800-2} = (-2) \cdot 2^{\frac{1}{2}} \cdot (\operatorname{arctanh}(1 + 2^{\frac{1}{2}} \cdot x)) + \langle L_{800-3} \rangle + (-4) \cdot (\operatorname{arctanh}((E_{307}) + x \cdot (E_{308}))) \cdot (E_{309})$$

$$L_{800-3} = (-4) \cdot (\operatorname{arctanh}(E_{402} + (-1) \cdot (E_{303}))) \cdot (E_{304}) + \langle L_{800-4} \rangle + 4 \cdot (\operatorname{arctanh}(E_{307} + (-1) \cdot x \cdot (E_{308}))) \cdot (E_{309})$$

$$L_{800-4} = (-4) \cdot (\operatorname{arctanh}(E_{402} + (E_{303}))) \cdot (E_{304}) + \langle L_{800-5} \rangle + (-2) \cdot (E_{304}) \cdot (\ln(1 + x^2 + 2 \cdot x \cdot (E_{304})))$$

$$L_{800-5} = 2 \cdot (\ln(-1 + x)) + \langle L_{800-6} \rangle + 2 \cdot (E_{304}) \cdot (\ln(1 + x^2 + (-2) \cdot x \cdot (E_{304})))$$

$$L_{800-6} = (-2) \cdot (\ln(1 + x)) + 2^{\frac{1}{2}} \cdot (\ln(1 + E_{300} + x^2)) + (-1) \cdot 2^{\frac{1}{2}} \cdot (\ln(1 + 2^{\frac{1}{2}} \cdot x + x^2))$$

$$E_{202} = \left(\frac{1}{8}\right) \cdot \pi$$

$$E_{300} = (-1) \cdot 2^{\frac{1}{2}} \cdot x$$

$$E_{302} = \sec(E_{202})$$

$$E_{303} = \tan(E_{202})$$

$$E_{304} = \cos(E_{202})$$

$$E_{307} = \cot(E_{202})$$

$$E_{308} = \csc(E_{202})$$

$$E_{309} = \sin(E_{202})$$

$$E_{402} = x \cdot (E_{302})$$

File: C:\Documents and Settings\velina Stoyanova\My Documents\test\OMcollection\integral.omx

Fig. 16. Long expression: default summarised form with reduced number of terms displayed (1).

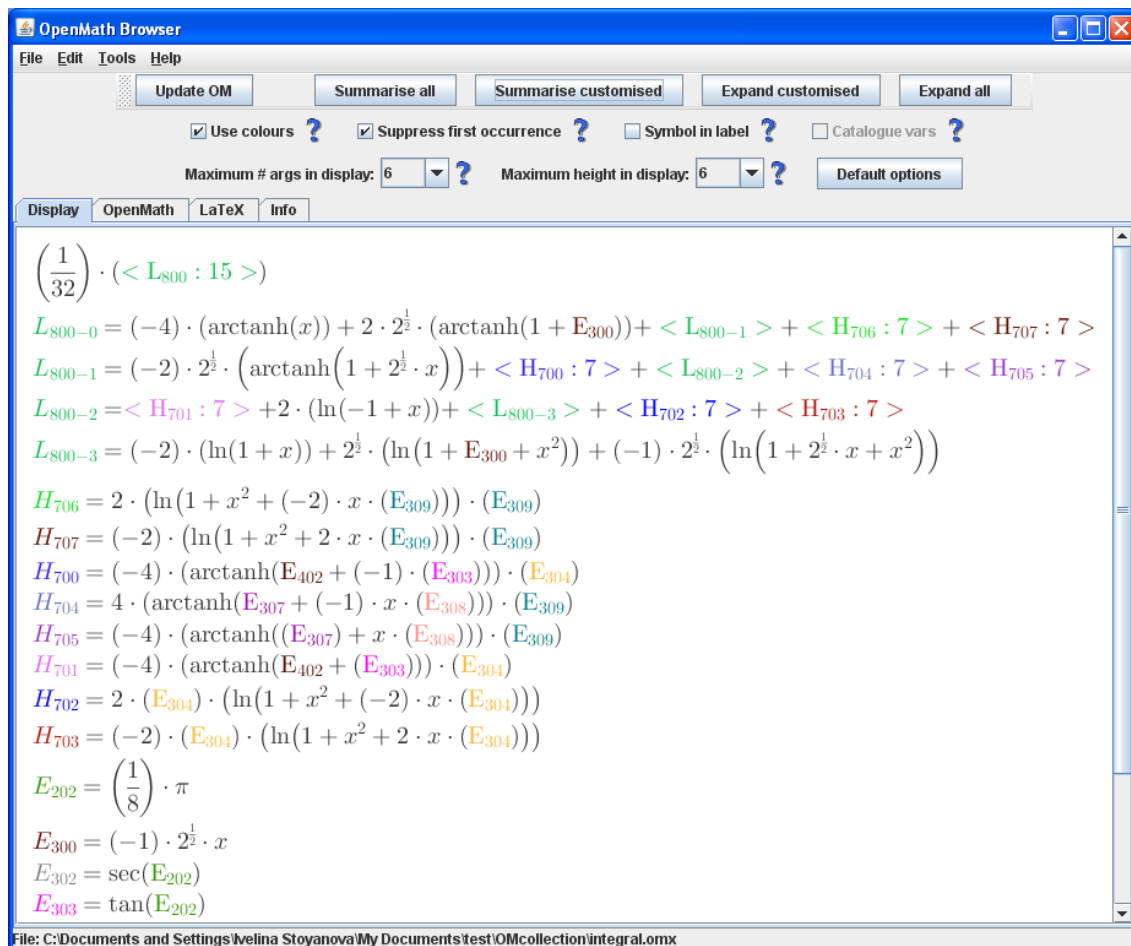


Fig. 17. Long expression: Combined summarisation.