# The plain text trap when copying mathematical formulæ

Paul Libbrecht
PH Weingarten
Weingarten, Germany
paul@hoplahup.net

Matija Lokar
University of Ljubljana
Ljubljana, Slovenia
Matija.Lokar@fmf.uni-lj.si

## Abstract

When an object, of any nature, is displayed and selectable on a computer screen, users expect it to be copy-and-paste-able: one can invoke the copy function and insert (paste) it at other places, within the same programme or beyond. This holds for many different kinds of objects: texts and images, at least. Unfortunately, for mathematical objects, this is rarely so.

Most operating systems offer multiple channels to carry exchanged content but most mathematical systems do not take advantage of it: they transfer the content in plain text, expecting it to have the right syntax or, if necessary, expecting the user to use a different copy function so that the right syntax is exchanged.

While ways to circumvent these issues are available, they are mostly not used by mathematical software. We explore potential justifications and describe for which type of users, these justifications do not apply.

To support this, we report briefly on the experiment students about their expectations and observations on the above mentioned process.

## 1 Intro: Transferring naturally between Competent Softwares

Copy and pasting mathematical formulæ between different systems is a desirable and common action. It is widely known that most mathematical systems have areas where they are very effective and while they only provide an approximate service in other areas. For example, dynamic geometry systems are good at constructive geometric figures and letting them be manipulated but also offer other functions in which they are rather less good: for example, most of them support some part of the TEX language to display formulæ and some offer computer algebra features; each of these extra features are very limited.

Most computer algebra systems (Maple, Mathematica, Macsyma, MuPad, Reduce) themselves have their own user-input parser and use TeX for display purposes mostly [Zha03]

Instead of relying on such limited extra features, users have the possibility to transfer the mathematical object between a system and another so that the receiving system offers its high quality features to solve the translated problem.

To perform the transfer of mathematical objects, the natural procedure of copy and pasting is often expected by users but, in the mathematical world, this procedure is decorated with special methods of many sorts, to

*Copyright © by the paper's authors. Copying permitted for private and academic purposes.*

In: A. Editor, B. Coeditor (eds.): Proceedings of the XYZ Workshop, Location, Country, DD-MMM-YYYY, published at http://ceur-ws.org

work somewhat properly. Switching between different systems, transferring relevant data, worrying about things getting "out of sync",differences in command sets and capabilities between different applications, soon becomes overwhelming[Lie00].

A simple sequence select, copy, switch, insert, paste is mostly expected by users. However, it is not rare a more complex procedure is needed such as the invocation of a special copy or adjusting the pasted content before it is further processed. For example[Kin02], the Stanford Interactive Workspaces'smart clipboard can copy and paste data between incompatible applications on different platforms[Kic00]. The smart clipboard must transparently invoke the machinery whenever the user performs a copy and paste operation. A more sophisticated but less general approach, semantic snarfing, as implemented in Carnegie Mellon's Pebbles project, captures content from a large display onto a small display and attempts to emulate the content's underlying behaviors [Mey01].

In the middle of this process lies the exchanged content. Most operating systems offer multiple channels to carry this exchanged content but most mathematical systems do not take advantage of it: they transfer the content in plain text, expecting it to have the right syntax or, if necessary, expecting the user to use a different copy function so that the right syntax is exchanged.

Why is this a problem? There are ranges of issues which are encountered by users and are all due to this choice of plain text. They range from syntax mismatch to unmasterable expressions, from the failure of apparent syntax compatibility to the somewhat arbitrary text-linearization of the text appearing within the formula.

While ways to circumvent these issues are available since long in a standardized form (clipboard flavours or alternative representations in MathML), they are not used by mathematical software. We propose potential justifications and describe for which type of users, these justifications do not apply.

## 2 Observation Methods for Copy and Paste of Formulæ

To be able to evaluate what are the expectations of users when transferring between systems, the authors employ their usage and teaching experience. These includes courses dedicated to the introduction of various systems (e.g. introduction to LaTeX, to computer algebra systems, or to dynamic geometry) in undergraduate teacher education classes. While this class of users is clearly not representative of the complete population of users of mathematical systems, it represents the important share of moderately technical users and also precludes those that will educate broad masses of citizens.

With this class of users, a practical experiment has been done at the University of Ljubljana for about 30 students in the first cycle professional study program Practical Mathematics: a mathematical task was given, explicitly requiring the exchange of mathematical expressions between various systems, of which comments and reports were expected.

In the first weeks of the subject called *Computer tools in mathematics* they get used to typical examples of mathematical software they are likely to apply in their career. They obtain mostly the basic knowledge, so they know only the most common functions and functionality of the applications. After a few weeks they received the experiment's task as follows (one of the weekly assignments): They should solve a certain mathematical task (and report in detail the process of obtaining the solution). In all the tasks given we foreseen the usage of different tools. Most often the combination of computer algebra system, dynamic geometry system and numerically oriented matrix software was needed. So even when certain tasks could be solved within one software, their their limited familiarity with the software lead them to use multiple softwares. For reporting they mostly used the most common word processing software. We were interested how they will cope with the process of exchanging the mathematical object between programs and how they will report on that where transfer of various mathematical objects has been expet. The subjects were instructed beforehand to report on all difficulties and obstacles.

To be able to evaluate what the mathematical systems are able to import, clipboard inspectors are used: On Windows the ClipSpy utility which shows the bytes allocated for each flavour within the clipboard. The application seems to show most of the information accessible by API as documented on [MSCl]. On MacOSX, ClipboardViewer (an example application of the Apple Developer Tools) has been used. It shows, similarly, an association between the "Uniform Type Identifiers" (the name of the clipboard flavour types defined by Apple, see [ApUTI]) and the byte-values. Both the Uniform Type Identifiers (on MacOSX) and the Windows flavour names are a flexible mechanism to encode the type of content-flavours: the strings define, in a kind of cooperative agreement way, which data-type is put in the clipboard. Platform makers define basic types (e.g. basic images and texts) while applications are free to define new formats. In the case of UTI, a mechansim of inheritance is provided.

## 3 Issues when employing plain text in copy and pasting formulæ

The first family of issues that we have met is that default copy mechanisms are rarely universally applicable and are, very commonly, restricted to only copy and paste internally, for which the system clipboard is not used (as copying from the system to itself is much safer). This same default copy function tends to copy a representation that is close to a source format within the text flavours. While this representation may be readable by users, it is not effective to input in most other mathematical systems as the input syntax is specific to each system. Mismatches of syntaxes then need to be manually fixed by the user who has to understand both syntaxes; something that is cognitively demanding when the mathematics thinking also demands cognitive resources.

Examples of such incompatibilities include an incomplete LaTeX compatibility of a dynamic geometry system: the user is left to explore bits by bits what works and what does not and, contrary to LaTeX, the documentation for the supported features is far smaller.

The second family of issues lies in the apparent compatibility of syntaxes between mathematical systems: While some expressions seem naturally exchangeable, e.g. simple polynomials written with the "^" as exponent, this compatibility breaks very quickly (e.g. these polynomials might be usable in a JavaScript source but not in a Python source because the exponent there is "**"). Much more delicate is the exchange of formulæ between Mathematica and Geogebra as is displayed in the picture below, produced by one of the experiment subjects: it shows that the Sum operator between the two computer algebra systems does not follow a completely similar syntax even if it shares quite some similarity as is the case for elementary formulæ between these systems.

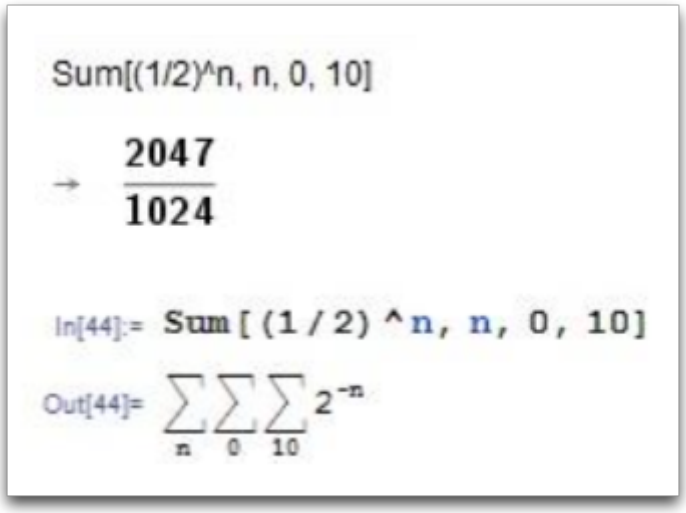


Figure 1: The same expression in Mathematica and GeoGebra...

A third issue lies in the need to use dedicated copy functions. It forces the user to navigate through sequential menu choices to decide which format to copy (at worst she needs to use a dedicated command before). Among the particularly invasive ones lies the frequent function "Copy as MathML" which puts the (generally big) MathML source code as text source and which is intercepted by paste actions ("this looks like MathML"): the difficulty to reach the menu and the rather wild appearance of the XML expression makes this operation very far from natural.

Finally, another issue in using plain-text is on web-browsers: As MathML there is generally represented as part of an HTML text page, the copied content is text, sometimes styled text. The copy function generally copies only an unordered sequence of characters which may be valid only if fractions or roots are not present; moreover, the copy function is supported by a selection mechanism which completely ignores the underlying mathematical structure (even that of MathML).

## 4 User types and their affinity to fiddling with text

While we acknowledge that using plain text to copy and paste often yields readable syntax and, what is probably more important, repairable syntax, it can be accepted differently depending on the users.

Experts we have met generally accept well the syntax differences; they can even be a good hint to the capacity spectrum of the computing system. However, there is a clear need to master well the different syntaxes, which is generally not available in many user types we met.

Such comments appeared in our experiment to illustrate our case:

- *".. that Copy/Paste method does not work with drawing plots in different programs. For drawing plots each one of used programs has its own 'language' which we have to use. "*
- *"... we were very rarely able to use the Copy/Paste method to transfer expressions between programs and even then some corrections were usually needed for programs to work"*
- *"After several tries I gave up. The only way to transfer an expression between programs is to manually type it..."*
- *"To input matrix in X is surprisingly identical to input matrix in Y. But unfortunately here the joy ends."*

In the discussion of the task afterwards students we performed experimentation with all express their deep disappointment about the softwares. When started the task they mostly all expected there will be just some minor problems. As one students said "it is all mathematical software and mathematics is just one, so I expected no problems in using Copy/Paste"

We conclude that the learning of syntaxes is the biggest hurdle for these students and that alternative ways need to be found to make a more productive use of the interchanges between pieces of software.

## 5 Conclusion: Standards that Support Copy and Paste

While copying plain text formulæ has been considered a universal way, we have described a population of users for which this approach fails and for which different exchange mechanisms are needed.

Standards exist to this end. They include the clipboard flavour names of Windows and UTIs on MacOSX as specified in the media-types of MathML: these describe the names of three clipboard flavours that can encode MathML for several applications. Using them, it is possible for a computing system to use its internal mechanisms to export MathML content in the form of a parseable tree and MathML presentation in the form of a rendered LaTeX fragment.

Similarly, embedding MathML within HTML fragments might potentially include the wealth of MathML exchanges as indeed a considerably bigger consideration exists for Web-pages' content. However, many of the efforts towards empowering web-oriented copy-and-paste such as the Clipboard API draft tend to consider [Che15].

Exchanging MathML offers another possibility to combine multiple types: through the `semantics`, `annotation`, and `annotation-xml`, the markup may contain alternative representations of the formulæ in different media-types (which we consider equivalent to clipboard flavour names), see [Car13, §6.4]. It is not clear if this *competing* alternate-representations mechanism may produce different results.

If doing this, the receiving party can choose the format it best understands from the list of clipboard flavour names and from the alternate representations in MathML. Thus a layout software might use a MathML-presentation or Rich Text Format fragment, or even a picture; however another computing system would use the MathML-content fragment or another flavour which is potentially better suited (e.g. to accommodate for bilateral import-export functions). Moreover, receiving multiple representations is possible with clipboard flavours and with MathML alternate representations. This allows the recipient to request all of the data-streams for each of the formats and potentially evaluate further its processability or offer a choice to the user using the internal display mechanisms (as investigated in [Lib09]).

As for systems which do not understand any of the mathematical oriented flavours: a picture, a rich text format or... a fragment of plain text (as a last resort) might be satisfactory.

## References


[ApUTI] Apple Inc. Introduction to Uniform Type Identifiers *Mac Developer Library* Available at `https://developer.apple.com/library/mac/documentation/FileManagement/Conceptual/understanding_utis/understand_utis_intro/understand_utis_intro.html`.

[Car10] D. Carlisle. MathML on the Clipboard, Blog entry. `http://dpcarlisle.blogspot.de/2010/01/mathml-on-clipboard.html`.

[Car13] D. Carlisle and P. Ion and R. Miner. Mathematical Markup Language, Version 3 (2nd edition) W3C. http://www.w3.org/TR/MathML3/.

[Che15] D. Cheng Clipboard API: remove dangerous formats from mandatory data types public-webapps mailinling list post on 2015-06-09. Archive of the thread visible at https://lists.w3.org/Archives/Public/public-webapps/2015AprJun/0819.html.

[Kic00] E. Kiciman and A. Fox. Using Dynamic Mediation to Integrate COTS Entities in a Ubiquitous Computing Environment. in *Handheld and Ubiquitous Computing: Second International Symposium, HUC 2000 Bristol, UK, September 25–27, 2000 Proceedings*,221–226, 2000.

[Kin02] T. Kindberg and A. Fox. System software for ubiquitous computing. *IEEE pervasive computing*, 1(1):70–81, 2002.

[Lie00] H. Lieberman and T. Selker. Out of context: Computer systems that adapt to, and learn from, context. *IBM Systems Journal*, 3.4(39):617–632, 2000.

[Lib09] P. Libbrecht, É. Andrès, and Y. Gu Smart pasting for ActiveMath Authoring *Proceedings of MathUI 09 Workshop*, http://www.activemath.org/workshops/MathUI/09/, 2009.

[MSCl] Microsoft Inc. Clipboard Class, API Documentation available at https://msdn.microsoft.com/en-us/library/system.windows.clipboard.aspx (checked 2016-07-20).

[Mey01] B. A. Myers and C. H. Peck and J. Nichols and D. Kong and R. Miller. Interacting at a Distance Using Semantic Snarfing. in *Ubicomp 2001: Ubiquitous Computing: International Conference Atlanta Georgia, USA, September 30–October 2, 2001 Proceedings*,305–314, 2001.

[Pad04] L. Padovani and R. Solmi. An Investigation on the Dynamics of Direct-Manipulation Editors for Mathematics. in *Mathematical Knowledge Management: Third International Conference, MKM 2004, Białowieża, Poland, September 19-21, 2004. Proceedings*,302–316, 2004.

[Zha03] L. Zhang and R. Fateman. Survey of user input models for mathematical recognition: Keyboards, mice, tablets, voice. *Computer Science Division, University of California*, 2003.