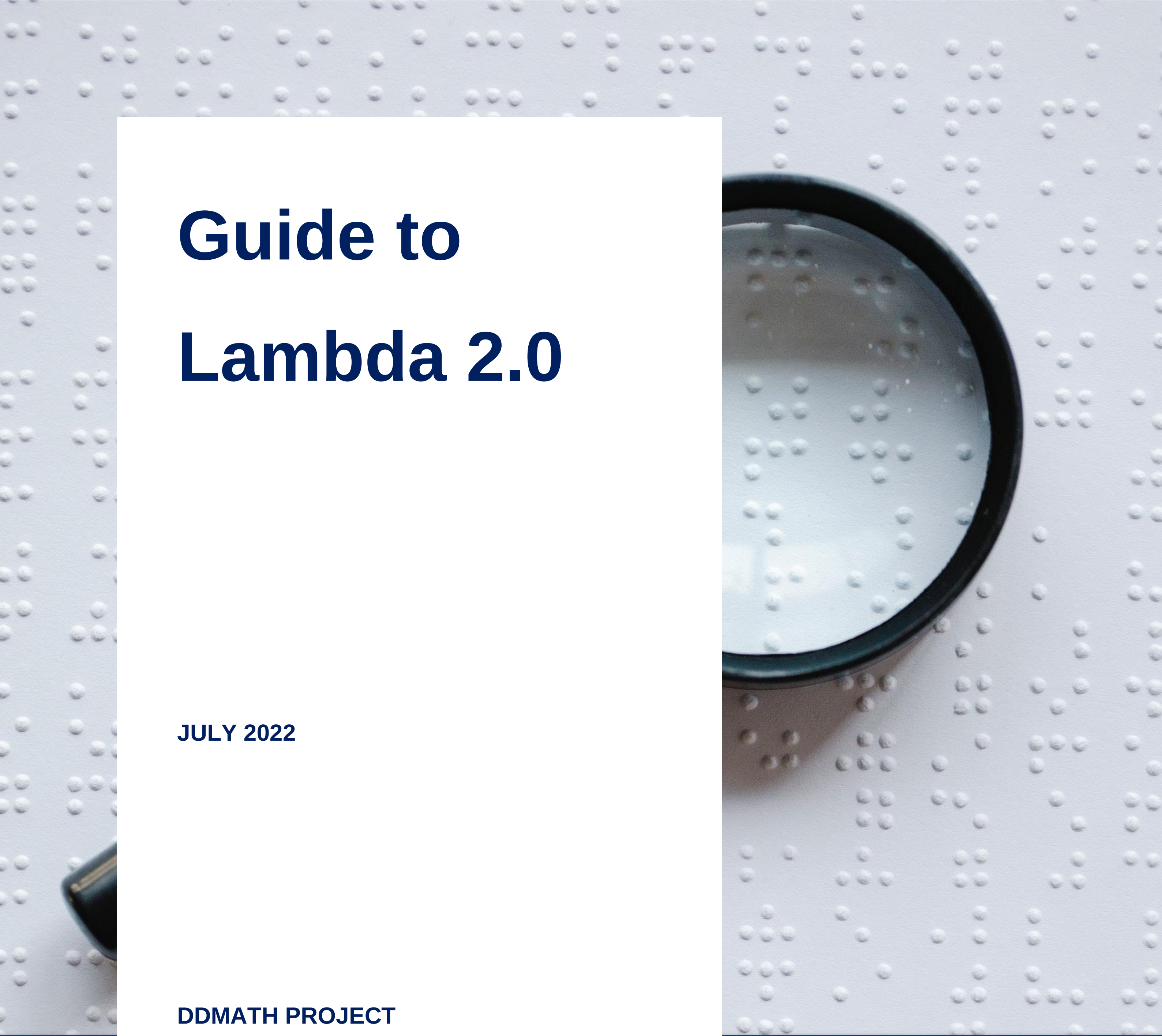


Guide to Lambda 2.0

JULY 2022

DDMATH PROJECT

Digital learning in mathematics for blind students

ERASMUS+ Program

Co-funded by the Erasmus+ Programme of the European Union

DDMATH

ERASMUS+ Program

DDMATH PROJECT

Digital learning in mathematics for blind students

Guide to Lambda 2.0

July 2022

Project within the ERASMUS+ program – School Education Gateway – Action KA2 - Strategic partnerships for digital education KA226 – Convention n. 2020-1-IT02-KA226-SCH-09557

Project number: ERASMUS+ program – School Education Gateway – Action KA2 - Strategic partnerships for digital education KA226 – Convention n. 2020-1-IT02-KA226-SCH-09557
Project title: DDMATH- Digital learning in mathematics for blind students
IO type: Publication
IO number 1.2
Contractual date of the Delivery: July 2022
Actual date of the Delivery: July 2022
Title of IO: Lambda 2.0 Guide
Nature of the Deliverable: Public
Authors: DDMath Consortium
Abstract: Guide/manual del Lambda 2.0
Keyword List: Blind, mathematic, Braille, 8dots, education, computing, LaTeX
The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Summary

1 Introduction

1.1 What is LAMBDA2.0

Lambda Math Code is derived directly from MathML, and is designed to be used with peripheral braille and vocal synthesis. And MathML is automatically converted from Lambda without the possibility of error. In addition, MathML can be converted into all mathematical script formats (LaTeX, MathType, Mathematica...). Both inputs and outputs are supported.

As of version 2.0, the Lambda editor allows users to write and manipulate mathematical expressions linearly. Because Braille represents all formulas in linear mode and is therefore difficult to understand, the Lambda editor has a series of compensatory features that reduce this difficulty. LAMBDA2.0 is intended for students of the first and second grades of secondary schools as well as for university students; it requires a minimum competence in computer use.

Lambda has the following characteristics:

- Eight-dot Braille symbology, concise and easy to remember.
- Extended and abbreviated voice description.
- Graphic representation of Braille to assist teachers and parents in understanding the job that the student is doing.
- Functions for handling mathematical text, which is in linear format, to process even complex expressions.
- From primary school up to upper secondary school, students have the opportunity to experiment with new solution strategies.

The Lambda team:

- Supports the teacher's didactic work every day.
- For students and instructors, it provides training and assistance.
- Transcribes books in Braille Lambda.

1.2 LAMBDA's code and editor

Through the editor, LAMBDA code is presented to users in a compact, easy-to-use way with Braille peripherals. A very important feature of the LAMBDA2.0 editor is that, as with the Latex and MathML visualization programs, it transforms the source code into a linear code, that can be consulted on video and with vocal synthesis. However, unlike viewers for Latex and MathML, LAMBDA2.0 software is an editor, not just a simple browser. It allows you to write and manipulate formulas.

LAMBDA 2.0's source code is hidden from the user. He doesn't need to access it since he can manage it easily and completely through the editor.

In order to create a compact LAMBDA code, mathematical symbols and markers are used with very few characters, often just one; thanks to the management software, which provides alternative reading modes, misunderstandings and difficulties of understanding are avoided. Vocal synthesis can be used to pronounce element names or read formulas in natural language. In addition, the element on which the cursor is positioned always appears entirely on the status bar, which can be consulted with either a Braille display or a voice synthesizer.

1.3 Braille 8 dots

It was decided to create LAMBDA symbols and special markers based on the 8-dot Braille combination that would represent them. The most common operators, markups, and symbols are represented by one character. To facilitate initial training and memorization, this character was chosen to be similar to the 6-dot Braille font in the national code.

Even though the LAMBDA source code is unique (and therefore independent of local linguistic choices), the Braille code applied to it varies from country to country to reflect local customs.

A compound fraction, for example, which includes several elements or expressions in its numerator or denominator, requires three markers in the LAMBDA code: one that indicates the beginning of the fraction, one that indicates its sign, and a third that indicates the end of the fraction.

The program will manage these markers in a specific way, recording them on file with a unique code. However, they will appear on the Braille display differently depending on the country. In addition, as is obvious, the textual descriptions and voices associated with speech synthesis will be translated into various languages.

A compound fraction in the Italian version, for example, includes three Braille symbols; we will provide a text for each symbol to indicate its name and a text to specify the words the synthesis must pronounce in order to ensure that the reading is seamless and natural.

Braille dots(examples in Italian)	Name of marker	Text read by speech synthesis
12467	Open compound fraction	Open fraction (or "fraction" for experts)
47	Intermediate compound fraction	Fraction sign
13458	Close compound fraction	Fraction end

For example, for the Italian language, the formula "a+b over a-b" corresponds to the dots 12467, 1, 235,12, 48,36,12,13458.

It is clear that the pair open/close has a strong analogy with the open and close symbols of 6-dot Italian Braille (it was not possible to use the same symbols at 8 dots since 1246 is assigned to number 6).

There is a strong resemblance between the intermediate, i.e. fraction sign, and the 6-dot slash (34 dots).

Other countries follow different rules to indicate, in 6-dot Braille, the compound fraction, and different combinations of Braille dots will be assigned to the three LAMBDA markers.

In certain cases, it is inevitable to combine several Braille characters (two or more symbols in sequence to define an element) due to the lack of Braille characters available.

The 8-dot Braille code for mathematics uses fewer Braille characters than the theoretical 256. The creation of too many new symbols would have created too many recognition and memory problems. When it is not possible to use analogies with the 6-dot code as seen above, we have tried to exploit logical or mnemonic connections. The use of a prefix, for example, as an analogy in set theory, will allow us to reuse a series of already memorized symbols.

This table shows some symbols defined in LAMBDA2.0 for the Italian language (dots 48

Union 48 235 □+	Prefix set theory and addition
Difference between sets 48 36 □−	Prefix set theory and subtraction
Included strictly 48 12678 □<	Prefix set theory and minor
Included in a broad sense 48 12678 2356 □<=	Prefix set theory and minor-equal

There are also three other prefixes defined:

- negation (dots 3468) reverses the meaning of the following symbol (for example: not equal, does not belong)

To represent Greek letters (45 dots), it must be followed by the corresponding Latin letter in both uppercase and lowercase;

- generic (dots 34568) used in several contexts, in particular geometry and logic.

Symbols and markers are always treated as units, even if they are represented with several characters (they must be inserted, deleted, moved, selected, etc.). In addition, TTS will always read the element name, not the sequence of symbols (for example, "gamma" will be read instead of "Greek-prefix gi", "union" will be read instead of "prefix-set addition", and "much greater" will be read instead of "major major").

1.4 View the mathematical document on video

Although LAMBDA2.0 is intended for blind people, its documents are also accessible to sighted people.

The teacher is fundamental in the didactic field, as he or she must be able to follow the entire instructional process, not just evaluate the final product.

Blind students' method of doing mathematics is characterized primarily by their linear codes, not by their use of Braille. In order to be truly helpful, the teacher must understand the consequences of this type of approach. For example, the need to use markers that are not necessary in graphic notation, the risk of error associated with their use, the difficulty encountered when working with fractional objects (for example, identifying the common denominator to add algebraic fractions), how to manipulate the document using the keyboard rather than pen and paper. It will be the teacher's responsibility to help the pupil in this regard.

The LAMBDA2.0 system displays the mathematical text in a linear mode in full correspondence with the Braille display, using a text graphic font. Symbols without a conventional representation are represented with special characters, which are designed to convey the meaning of the text as clearly and quickly as possible.

1.5 Integrated graphic visualization

Lambda, through the menu or the F4 button, produces a traditional two-dimensional graphic display. In this view, the automatic correction of Lambda syntax is not performed. In graphic format, any writing, even if it is not correct in terms of structure, is accepted. If the mathematical Braille syntax structure is not written correctly, this will only negatively affect the management of some compensatory tools offered by Lambda, (tools that require recognition of the structure) such as, for example, the structure view (F8), or the duplicate expression (CTRL+d). Thanks to the graphic representation the teacher will be able to be even more aware of any errors made by the student, even if they are of a syntactic nature.

2 System LAMBDA 2.0

A linear mathematical code and an editor are the components of LAMBDA 2.0

LAMBDA is a very compact linear mathematical notation system that is especially suitable for management and manipulation of vocal and Braille peripherals through computers.

The focus is on the content of the formulas rather than its visual aspect; this choice is particularly significant in the vocal synthesis, which describes the mathematical elements and structures in a language similar to the one used by the teacher.

For example, the expression:

$$\sum_{i=1}^{n} a_i$$

LAMBDA 2.0 interprets this as "Summation for i from 1 to n of a with index i" and not, as in other graphical-oriented codes, as "Capital sigma with subscript "i=1", superscribing "n", in line a with subscript i".

It is clear that there is a considerable difference from a didactic perspective.

2.1 Markers usage

There is a fundamental role for markers (also called tags) in any linear code, i.e. the codes that delimit mathematical structures.

In this expression, for instance,

$$x + \sqrt{x+1}$$

roots not only denote the operation to be performed, but also delimit, with their shape and extension, the part of the formula on which that operation must be performed, in this case (x+1).

The start and end of the operand can only be indicated with a pair of specific symbols in linear notation.

The linear LAMBDA notation is as follows:

x+√x+1⟩

In some mathematical elements enclosing two objects, an intermediate separator is also required in addition to the start and end markers. The most common case is the fraction x plus an open fraction, with a numerator of x-1 and a denominator of x+1.

$$x + \frac{x-1}{x+1}$$

Linear LAMBDA notation provides a starting marker, an intermediate separator (corresponding to the fraction sign), and an ending marker.
The linear LAMBDA representation of the example is shown below.

x+//x−1⌀x−1\\

With the LAMBDA system, all mathematical structures can be represented linearly by these pairs of open - closed markers with possible intermediates, which can be inserted several times one inside the other.

2.2 LAMBDA's general characteristics

The LAMBDA 2.0 mathematical editor appears to have a similar management environment to a common writing program.
There are the usual commands for opening and saving a file, correcting and deleting it, selecting, copying, and pasting. As in any word processor, all the most common operations are managed as well as shortcut commands.
There are two key differences between a normal writing program and this one:
- when the program is opened, the same workstation from the last session is automatically restored, which means the same document is shown, the cursor is positioned in the same position, and the windows are organized similarly.
Cursor movements are not restricted to the portion of the screen containing text, but can reach any position in the line or lines that have been created with the ENTER key.
With two lines like:

a+1
ab+a+2

Assuming we have the cursor on the "2" of the second line and we press the "up arrow" key, the cursor will go to the end of the previous line, which in our example is after the number 1.

LAMBDA editor, however, positions itself exactly above where it was previously, regardless of how far beyond the existing line it is.

```
a+1      |
ab+a+2
```

With this method of moving the cursor, you can access all points of the screen by moving freely along the vertical and horizontal axes; it is very useful in mathematical environments.

LAMBDA manages the mathematical environment (formulas, symbols...) differently than the textual environment, which requires the presence of open text and close text markers.

Environment for math:

A text environment with an open text marker (red), blue text, and a close text marker(red)

Exercise number 5

Mathematical environment with red operators (for closing analogies) or green operators for unique operators, black numbers

(-3/4)^2*(-8/3)^2

On this point, refer to the chapter Distinguishing between text and mathematics.

3 To start

For those who have never used LAMBDA 2.0 and would like to get started quickly, these pages contain some brief explanations.

3.1 Write the first mathematical formula in Lambda

Let's start with a very simple formula, composed of symbols that are all directly available on our keyboard:

3/4

It can be obtained simply by typing the symbols that make it up, as in any writing program, namely the number 3, the slash character (/), which is usually positioned above the number 7, and the number 4

If you are using a screen reader (such as Jaws, and we are sure to have properly installed the scripts), the formula is read in LAMBDA2.0 as "three divided by four" and not, as in other writing programs, "three slash four".

If the reading is not correct, check the installation again and consult, in particular, the part of this manual that deals with adapting the screen reader.

In addition, we can see our first formula in traditional two-dimensional graphical mode by pressing F4.

The formula will appear in a secondary window as follows:

$$\frac{3}{4}$$

You should know how to activate the function keys on your computer. This is because some models allow you to activate the function keys directly, while others require you to press the Fn + key simultaneously. If you perform a few tests on your computer, you will discover its working mode.

If a Braille display is available, the formula will appear according to the 8-dots code used in Lambda. It is very similar to the 8--dots code used in normal writing programs. However, in special cases, new symbols are required (this is not the case for this very first simple formula), which will generally be easily recognizable because they are similar to the corresponding symbols in the traditional 6--dots mathematical Braille used in your country.

If you have any doubts about remembering a Braille mathematical symbol, simply move the cursor over the symbol and the speech synthesis will pronounce its name; it will also be readable in the status line at the bottom, in the lower left margin of the program window.
All formulas that use symbols that are directly available on the keyboard can be written in a similar way.
Note that the multiplication sign is obtained with the asterisk "*" and the power with the circumflex accent (or hat) "^".
e.g.

```
3^2+2*(2^2-1)-(7-2*3)^2
```

By pressing F4 again, the graphical display will be updated with the new formula:

$$3^2+2(2^2-1)-(7-2\times 3)^2$$

Remember that the graphical display window can be moved and resized as desired in the active window of the Lambda2.0 program. This position will then be retained each time the graphical window is reopened with F4
With this numerical formula, we can also learn another function of the LAMBDA2.0 editor: the calculator. With the cursor in any point of the formula, we type Ctrl +F9 (holding the Ctrl key, type F9). A window opens with the formula and the result. If the function key is necessary for the function keys, the combination will be to press Ctrl +Fn +F9. The LAMBDA calculator offers many other functions, described in the Calculator chapter. To close the calculator window, simply press the Esc key.

3.2 Enter symbols not present on the keyboard

Mathematical symbols are numerous and not all of them are available on our alphanumeric keyboards. How to write, for example, the square root sign? Or the symbols of set theory? To insert symbols, LAMBDA2.0 offers four different modes:

• Through the keyboard: for the most frequently used symbols, it is convenient to use those already present on the keyboard and through a combination of shortcut keys. This last mode requires an initial effort of memory but then the input is very fast;

• Through F5: if you know the name of the symbol to be inserted, it is convenient to use the dynamic menu. Press F5 and write, even partially, the name of the element: the list below quickly reduces to a few elements, among which it is easy to choose the one of interest that will be inserted in the work sheet by pressing enter;

• Through Menu: if you don't know the name, you can search for the element among those in the corresponding group: for example, set theory, logic, analysis... Open the insert menu and then choose the group.

• Through Icons: finally, sighted users such as teachers, can insert symbols using the graphic menu bar, with icons

These four systems for inserting symbols and structure markers are described in detail in the chapter 'Inserting Symbols and Structure Markers.

Here an example.

We have to type the following formula

$$\pm\sqrt{\varphi}$$ *[plus or minus root of fi]*

As you can easily verify, none of the three symbols is present on the keyboard.

In LAMBDA2.0, the "±" symbol is easy to insert because you just need to type the symbols that make it up: +- (plus and minus). On the screen, the two symbols will appear one after the other, but for LAMBDA they are considered a single object: the synthesis voice pronounces them together and it is impossible to separate them or delete only one. If you press the arrow, the cursor moves from the beginning of the first character directly to the end of the second, that is, it considers them a single character and to delete them you just need to press the Del key to delete both. Other symbols are composed in this way, such as

"≤" (less than or equal to) "≪" (much less than)

These symbols are represented in LAMBDA respectively with <= and << (see the section: double symbols) The root symbol can be inserted by typing F5 and writing "root" (in fact, as you can see, just writing "sqr" in the window that opens with F5 will give you a very compact list). See: Searching the list of elements. But the root is usually an element that is used very often and it is convenient to learn the quick selection keys: CTRL + r (while pressing Ctrl, press r). See: list of symbols with the default profile's quick selection keys The symbol φ (fi) can also be inserted with F5, but if you often use Greek letters, it is easy to remember the procedure that works for all of them: you type a prefix (CTRL g) followed by the corresponding Latin letter, in our case the letter "f". Finally, here is how the formula appears on the screen

+−√⊡f

It appears to be composed of 5 characters, but there are actually only 3 LAMBDA elements inserted; to see this, move the cursor step by step with the arrow keys (right or left).
To learn more about the topic, see:
Inserting symbols and markers.

3.3 Structures and markers

For many mathematical objects, it is not enough to insert a sequence of symbols, but it is necessary to define a structure with markers that indicate the beginning and end of a certain property or relationship. For example, take this formula: "numerator: a plus one, denominator: a minus one.

$$\frac{a+1}{a-1}$$

To represent it all on one line, we cannot use the simple fraction bar as in the example in the previous chapter, but we must clearly indicate with a specific marker (a character or symbol) where the fraction line begins and where the fraction ends

//a+1∤a−1\\

If we only used the fraction sign:
a+1/a−1

the resulting formula would be different, that is, a plus one divided by a minus 1

$$a+\frac{1}{a}-1$$

The compound fraction requires three markers:

an initial opening marker

an intermediate marker (the fraction line)

a final closing marker.

There are many mathematical elements that need a structure similar to this. In some cases it will be enough to define an opening marker and a closing marker, in others an intermediate marker will also be needed, as in the fraction. All the pairs of parentheses commonly used in formulas also represent a structure; they will naturally be of the "open/closed" type, without an intermediate marker. Structures are the foundation of linear mathematical notation systems, such as LAMBDA, and all Braille mathematical codes. The LAMBDA2.0 editor offers many tools for effectively managing and manipulating them. It is very convenient to use the same command to insert all closing markers and intermediates. When writing an element with structure, only the opening code must be explicitly chosen by the user; for the others, the same command will always be typed and the system will recognize which structure needs to be closed. In addition to being easier and faster, this system greatly reduces the risk of errors. For example, to insert the fraction seen earlier:

//a+1/a−1\\

First, the symbol of "compound fraction open" is inserted,(better with the shortcut keys CTRL q.)

Then the numerator a+1 is written. To insert the intermediate separator, that is, the fraction sign in this case, the command CTRL i (i stands for intermediate) will be used. Then the denominator a-1 is written and finally, the structure is closed with CTRL k (k stands for closure) CTRL k is the command that closes all structures, including parentheses, while CTRL i is the command that will always be used to insert the intermediate element.

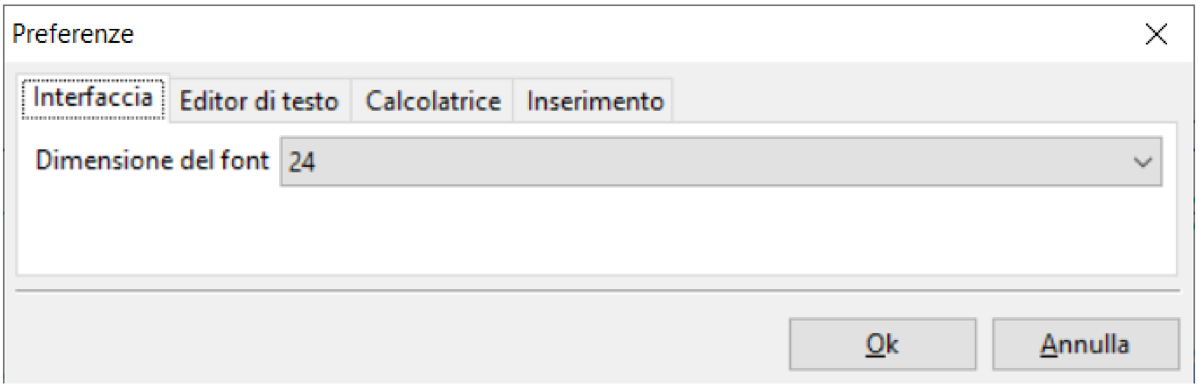
3.4 Colours:

The LAMBDA editor uses different colours to indicate structure elements and operators. The information provided by the colours can be useful but, for obvious accessibility reasons, they are never essential for understanding the text. When a marker requires an intermediate or a closing symbol, it is represented in red, while ifit is unique and does not require a closure, it is represented in green. For more information, see the page on the meaning of colours in LAMBDA.

3.5 Submenu preferences

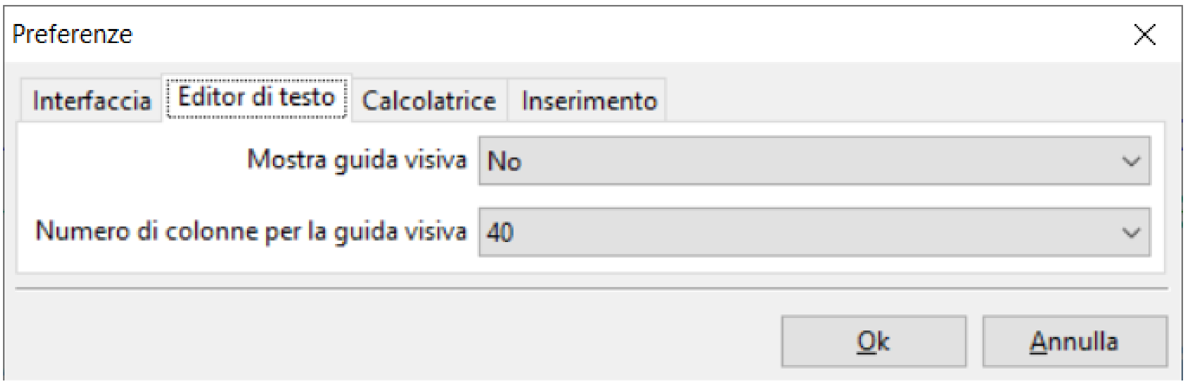
By entering the File menu and selecting the Preferences submenu, a window with 4 tabs opens:

Interface:



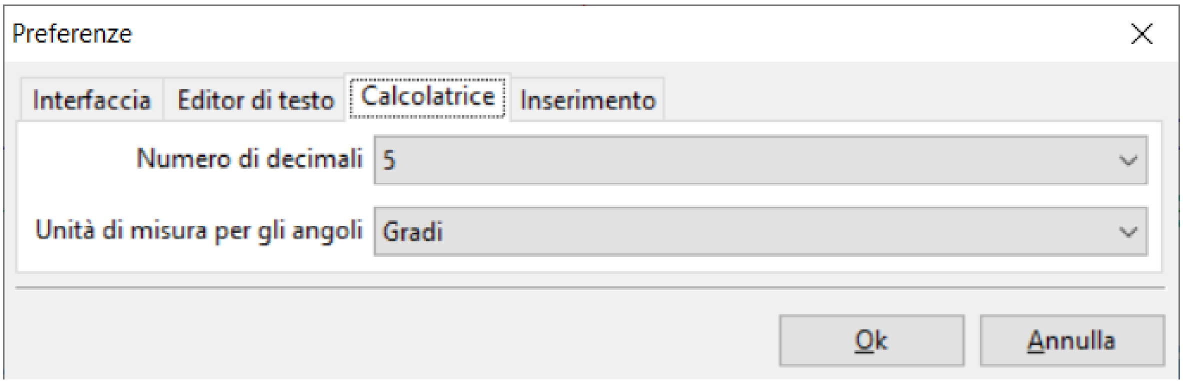
The first tab, "interface," allows you to set the font size on the screen, which is useful for visually impaired children who can use their remaining vision along with the text-to-speech synthesis. The default value is 24, but it can be changed up to 96. After changing the font, you need to restart the Lambda2.0 program. A warning for those who use the Braille line: sometimes with large font sizes, the screen reader drives the Braille line by inserting a series of spaces between the characters.

Text editor



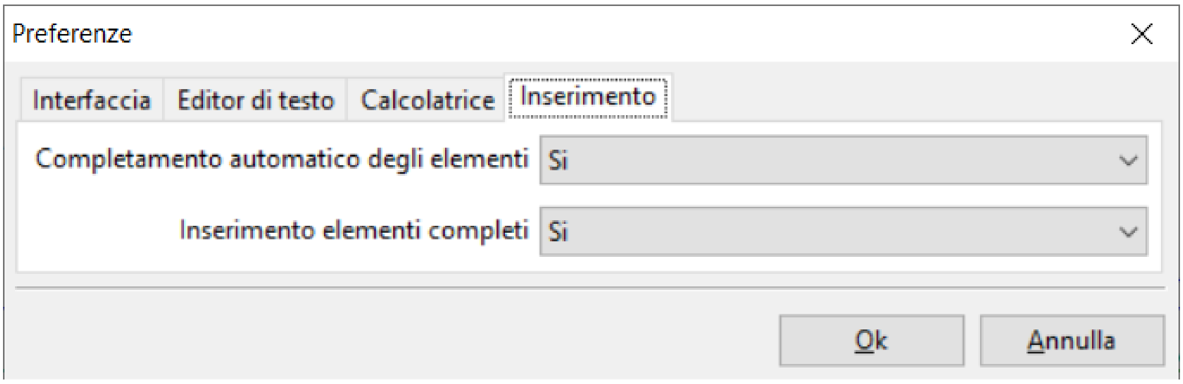
The second tab, "text editor," allows you to display a vertical red line. It is particularly useful for teachers and study assistants transcribing in Braille to set the text according to the type of Braille line used, for example 40 characters. Some students do not like using the continuous horizontal text shifting functions on the Braille line, but they request a fixed position at 40 characters and movement is only possible when moving to the lower or upper line.

Calculator



For the "calculator" tab, there are two choices: Number of decimals for the calculator (value from zero to five) Unit of measure for angles with three choices: degrees, radians, or gradients

Insert



The "insert" tab offers an option that allows for the automatic completion of elements that can be assigned a yes or no. When you enter yes, the program attempts to determine whether you are entering a compound element, such as cos, sen, lim, etc. When the last letter is pressed, it is recognized as cosine, sine, limit. By entering no, this function is deactivated and those elements can only be inserted from the menu, toolbar, or through a shortcut. The second entry on the insert tab offers two options: by selecting "yes," when a compound element is called, it is presented complete with an opening, possible intermediate, and closing. At this point, you just need to complete it with the values. By entering no, when a compound structure is called, only the first opening element of the structure is inserted and you need to complete it with the intermediate (using Ctrl+i) and the closing marker (Ctrl+k).

3.6 Tools for understanding the structure of a formula

We write this expression: "open root, open fraction, b, open parentheses, a + 1, close parentheses, – b, fraction line, a close fraction close root plus a".

√//b(a+1)-b/a\\⟩+a

It should be remembered that the three closing symbols, based on the "complete element insertion" choice, are presented complete with intermediate and closing, or, if "no" is selected, they are all inserted manually with the Ctrl+k command and it will be the program's task to insert the corresponding symbol based on the last open block. The intermediate Æ is inserted, as we have seen, with the Ctrl+i command. Here are a few suggestions: with Ctrl+r (lowercase r), we type the simple root, with Ctrl+R (uppercase R) the compound root. We have already seen how we can display the formula in graphical mode with F4. For complex formulas, with more elements inserted one within the other (as in this case), LAMBDA2.0 offers another convenient display, particularly useful for understanding the hierarchical structure of the formula. We type F8 and a new window appears, immediately displayed on the Braille display, which shows the formula in this way

√ ⟩+a

The highest level block, that is the root in this case, has been emptied and only the two markers and the elements outside the structure appear. In this way, you can quickly and immediately grasp an information: the expression contains something under the root added to a. With the Next Page and Previous Page keys, you can similarly explore the internal blocks, increasing or decreasing the minimum level displayed.

Pressing Next Page multiple times, for example, our formula appears like this:

√⟨   ⟩+a

√⟨//   ⸗/   \\⟩+a

√⟨//b (   ) −b⸗/ a\\⟩+a

As you can see, with this view, program represents hidden parts with empty spaces. This can be useful to have information on the size of the blocks, but if the formula is long, it may be more convenient to remove the spaces to have a more compact representation. To switch to this view, just press F8 again (with this key, you switch cyclically between the expanded and the compact representation). The examples from before will appear like this::

√⟨⟩+a

√⟨//⸗/\\⟩+a

√⟨//b ( ) −b⸗/ a\\⟩+a

The display window is closed with Esc. Both expanded and compact displays are also useful for quickly finding a reference point within the formula: moving the cursor to a point and then keeping it when returning to the editor's work page.

3.7 Solve the first expression in LAMBDA

Let's continue with the formula written earlier: "open root, open fraction, b, open parentheses, a + 1, close parentheses, – b, fraction line, a close fraction close root plus a”

√⟨//b (a+1) −b⸗/ a\\⟩+a

The calculation of this expression is usually done using the successive rewriting system: the expression is copied several times, performing intermediate calculations or transformations at each step. Using a Braille keyboard, for example, this system is not at all easy, but it becomes so when doing mathematics with a computer, especially if there is the possibility of automatically duplicating the expression line and working only on the copy using the correction method. Let's take it step by step:

To duplicate a line, the user can employ the usual procedures. For example, if the cursor is at the beginning of the line: End key with Shift to select the entire line, then Ctrl+c to copy, go to the beginning of the next line with the down arrow and the Home key(↖), paste the selection with Ctrl+v. Alternatively, you can use LAMBDA's Ctrl+d command, which does all this automatically, eliminating any empty spaces and copying the line twice to leave one for any control checks (this function does not work in Lambda2.0). So let's return to our example and duplicate the line (only once):

√⫽b(a+1)−b⁄a⑊⟩+a

√⫽b(a+1)−b⁄a⑊⟩+a

We perform the first possible transformation on the calculation line:

√⫽b(a+1)−b⁄a⑊⟩+a

√⫽ba+b−b⁄a⑊⟩+a

Duplicate the last line:

√⫽b(a+1)−b⁄a⑊⟩+a

√⫽ba+b−b⁄a⑊⟩+a

√⫽ba+b−b⁄a⑊⟩+a

And perform the possible calculations:

√⫽b(a+1)−b⁄a⑊⟩+a

√⫽ba+b−b⁄a⑊⟩+a

√⫽ba⁄a⑊⟩+a

Duplicate again:

√⫽b(a+1)−b⁄a⑊⟩+a

√⫽ba+b−b⁄a⑊⟩+a

√⫽ba⁄a⑊⟩+a

√⫽ba⁄a⑊⟩+a

And calculate

√⫽b(a+1)−b∕a⑊⟩+a

√⫽ba+b−b∕a⑊⟩+a

√⫽ba∕a⑊⟩+a

√⫽b∕1⑊⟩+a

At this point, the equation is practically solved. The structure of the fraction has a denominator equal to 1 and is therefore no longer necessary. The root can be replaced with a simple structure. To delete the markers of a structure, without risking forgetting some, it is advisable to use LAMBDA's Shift+Del command (just move the cursor to any of the markers, initial, intermediate, or final, and they are all deleted)

√⫽b(a+1)−b∕a⑊⟩+a

√⫽ba+b−b∕a⑊⟩+a

√⫽ba∕a⑊⟩+a

√⫽b∕1⑊⟩+a=√b⟩+a (or:√b+a)

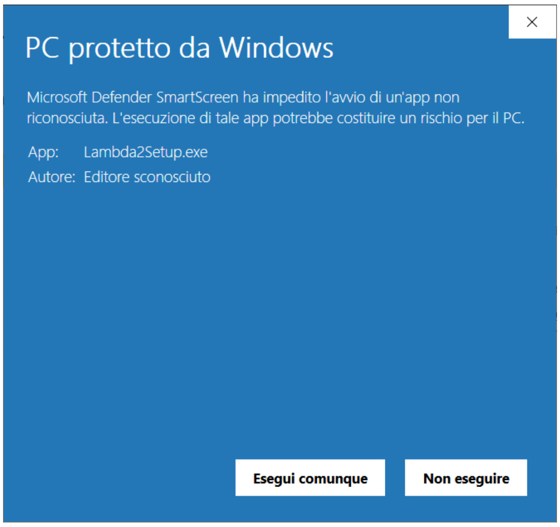
4 Installing LAMBDA2.0

4.1 Installing Lambda 2.0

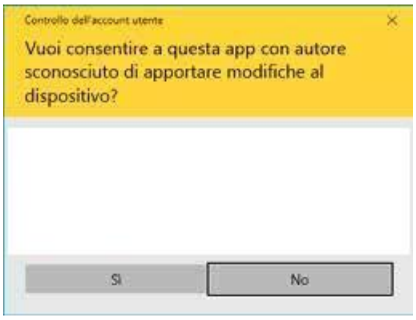
Installing LAMBDA2.0 requires a few simple steps.
Download Lambda 2.0 from this link:

https://ddmath.eu/download/lambda2editor/

Before downloading the latest version of Lambda, you must enter some simple personal data, accept the privacy policy and follow the instructions. Then proceed to install the LAMBDA program by running the Lambda2Setup.exe file. In some cases, and this depends on the level of user account control settings, the system may ask for confirmation to allow the installation of the application from an unknown publisher. Of course, if you want to install Lambda2.0, you must run it anyway and say yes to the warning window.

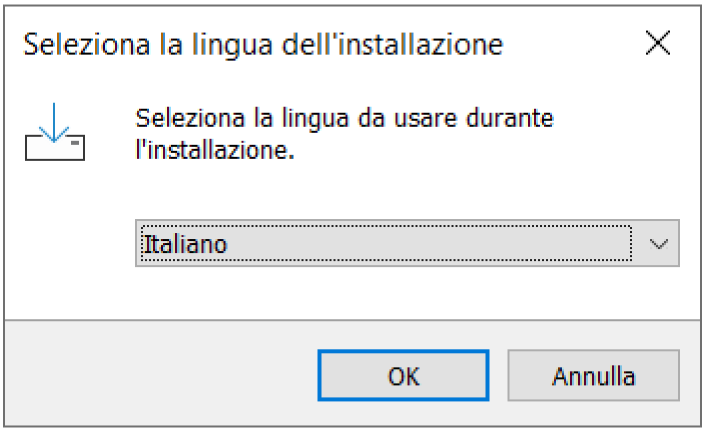


And then also

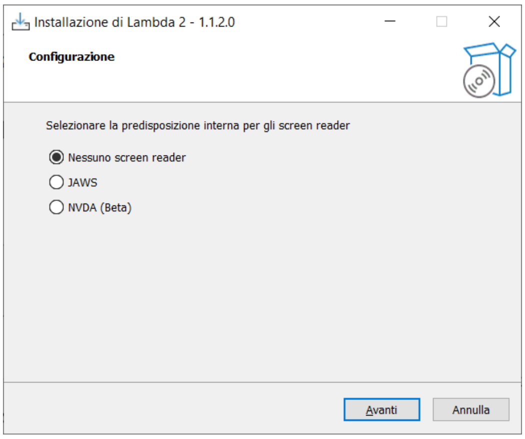


The antivirus program installed on your computer may also block the installation as Lambda2.0 is a new program for it, and in this case, you must press the button that allows you to proceed with the installation. Refer to your antivirus program manual on this point. The installation procedure is very simple, but you can customize the installation by choosing the language, working directory, etc., or leave everything as proposed. Thanks to the Erasmus+ DDMATH project, Lambda2.0 is currently available in the following languages and Lambda code localizations:

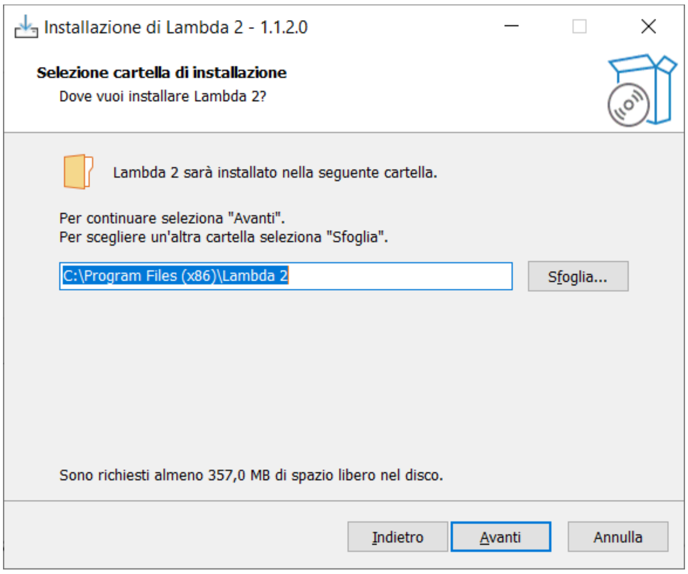
Italian
Polish
German
French
Portuguese
Ukrainian



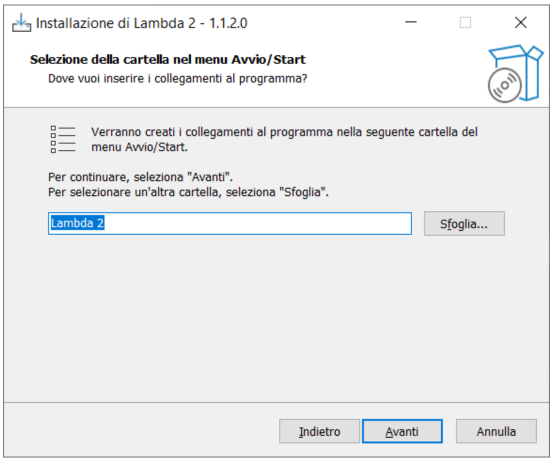
The choice of language also determines the Lambda 8-dots Braille syntax, which is derived from the traditional 6-dots Braille syntax commonly used for that language and country. If there is a previous installation of the program, it will ask you to remove the previous version and then proceed with the new installation. During the installation process, you will be asked which screen reader is installed on your PC, none, Jaws or NVDA. The screen readers that can be used are Jaws up to the latest version 2023 and NVDA 2022-3.



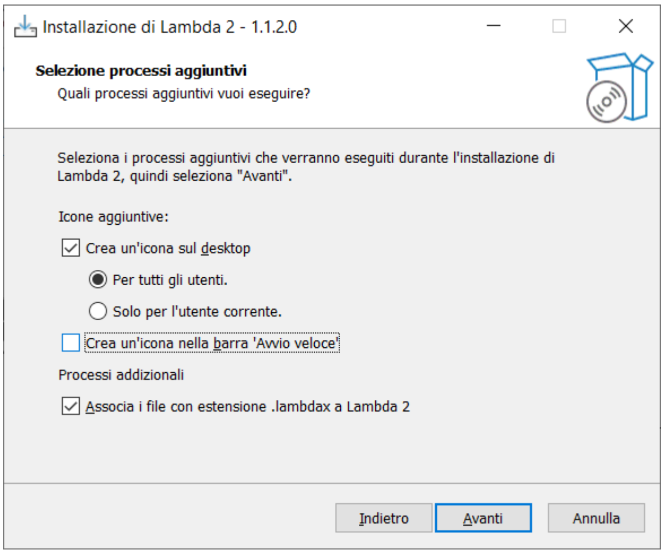
Select the type of screen reader installed on your PC and press next. Let's see in detail all the subsequent steps for the installation. A installation folder is proposed, which can be modified, and a notice that indicates the need to have 357 MB of available space on your hard disk.



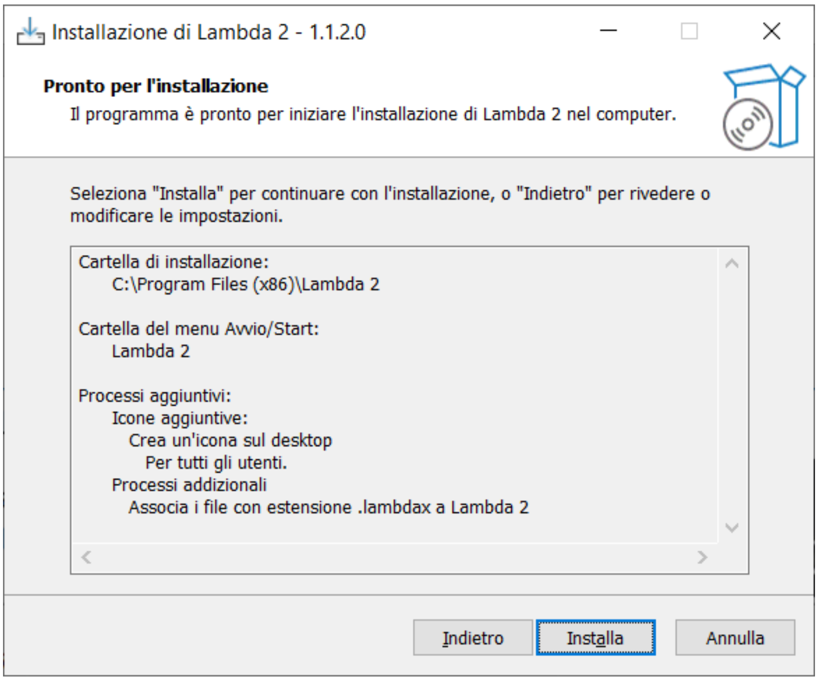
If we press next, a new window will ask you to specify the name of the folder that will be created in the Windows Start/Start menu, and in this case you can simply press next. This will create the Lambda2 folder in the start menu.



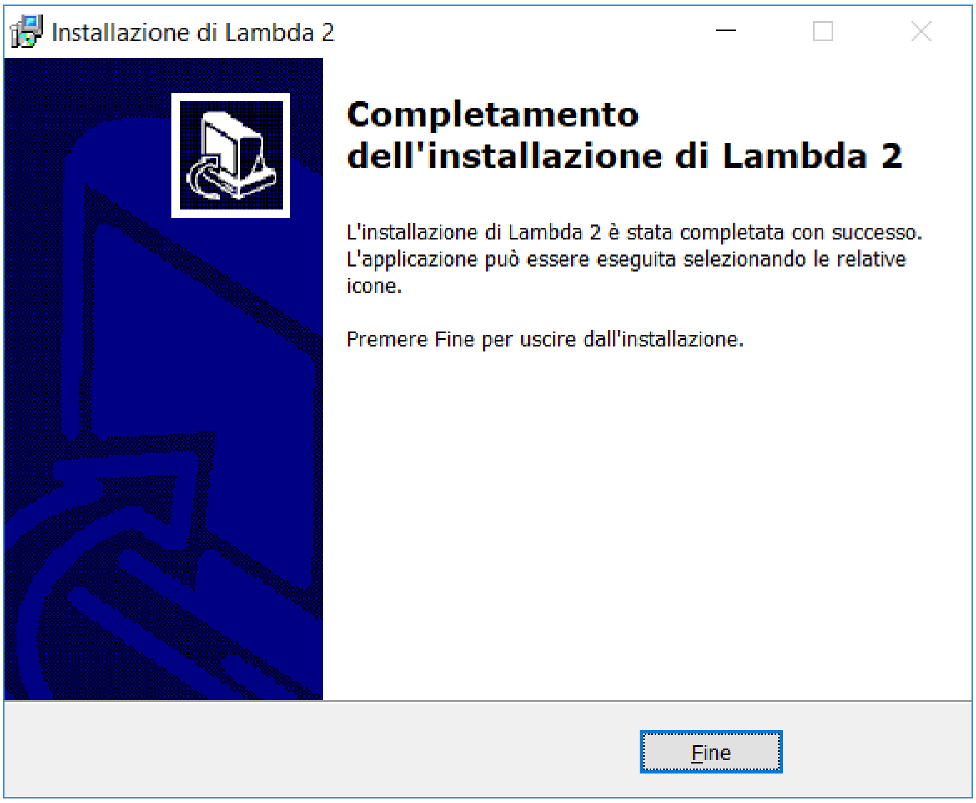
Another window suggests to create a Lambda2.0 icon on the desktop and, in the case of multiple access to the same computer with different users, proposes to install the program for all users or only for the current user. Another selection is whether or not to create an icon in the quick launch bar and finally whether or not to associate the new Lambda2 file extension, which is ".lambdax". Note that documents created with Lambda version 1.3 have a different extension: ".lambda



Finally, everything is ready for installation. After a brief summary of your choices, you can either click install or go back to make changes.

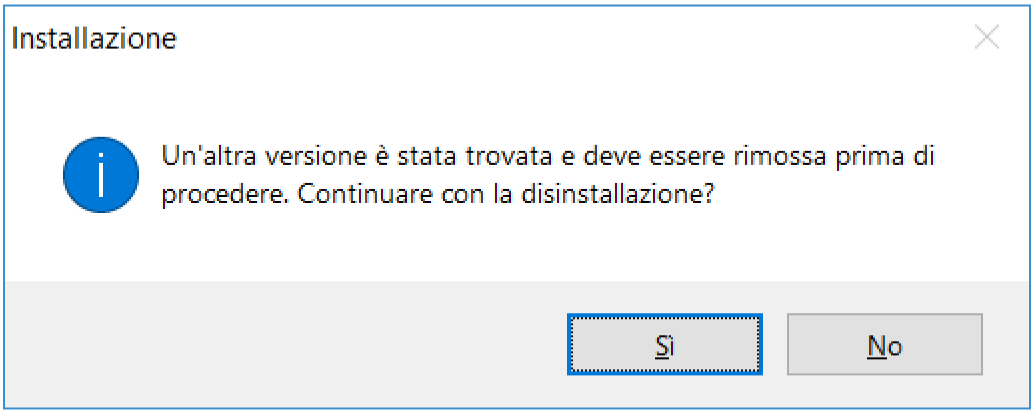


Depending on the speed of the system, the installation process takes about 3 minutes. As a result, in most cases, clicking Next until the installation is done is sufficient. Once the installation of the JAWS scripts is complete, a window appears to complete the installation and you press the Finish button. This is followed by a second window and the FINISH button to close the entire installation process

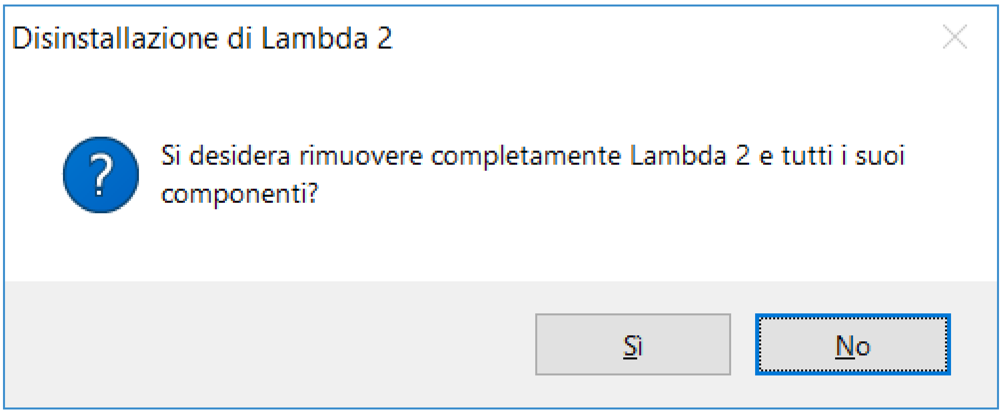


4.2 Reinstalling a new version over the existing one

If you want to install a new version of Lambda2.0 over the existing one, the procedure appears slightly different in the initial part. The first window asks for the installation language. By pressing next, a warning appears asking if you can proceed with the removal of the previous installation. Click Yes if you want to proceed.

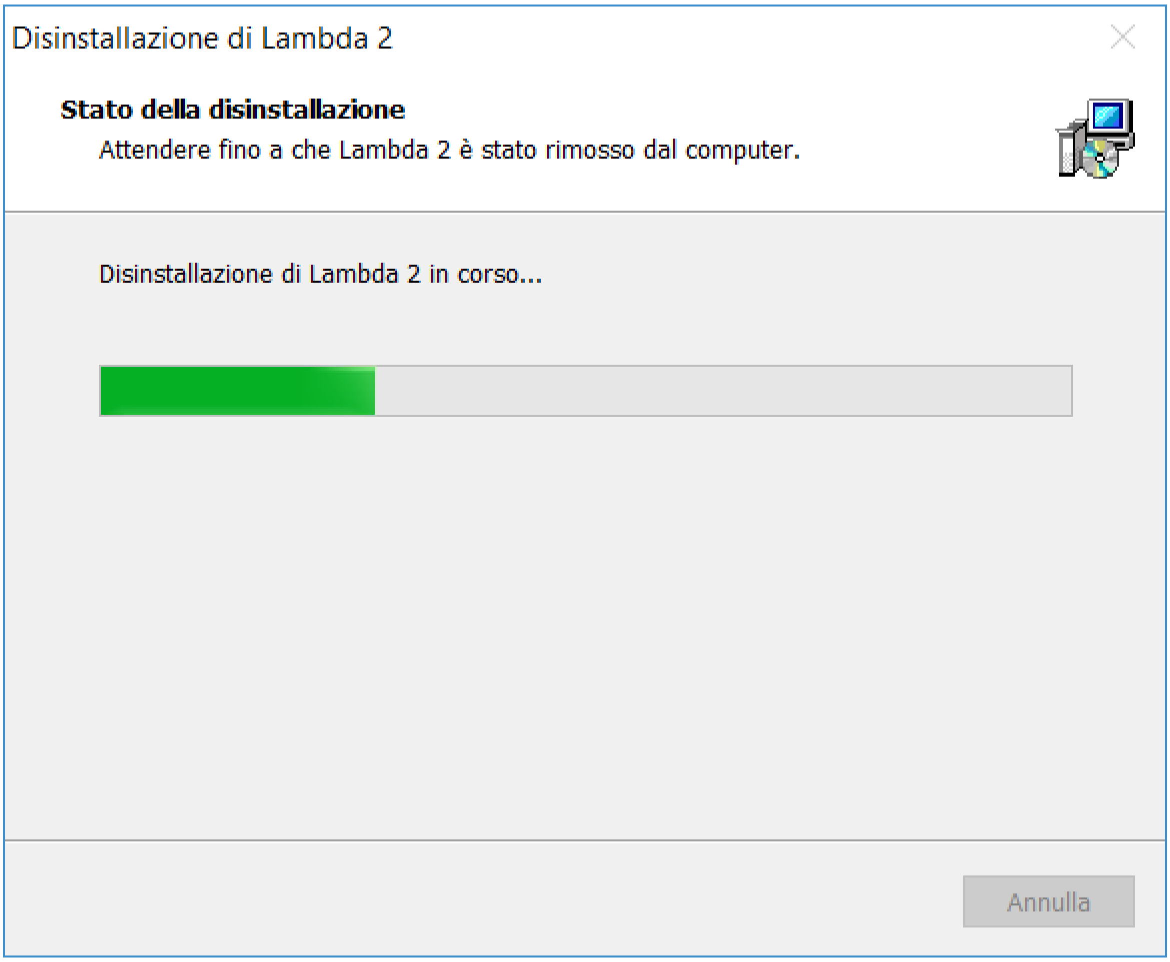


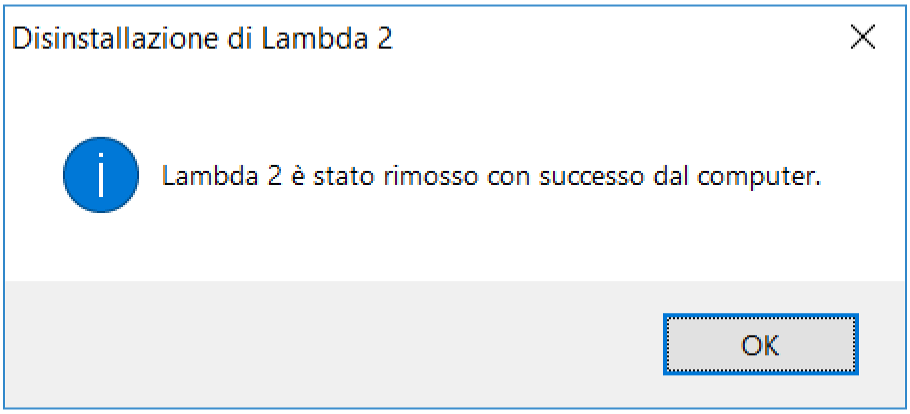
The program requires a second confirmation before proceeding with the removal, and by clicking 'Yes' the removal program finally starts



At the end of the removal process, a window informs of the completion of the removal.

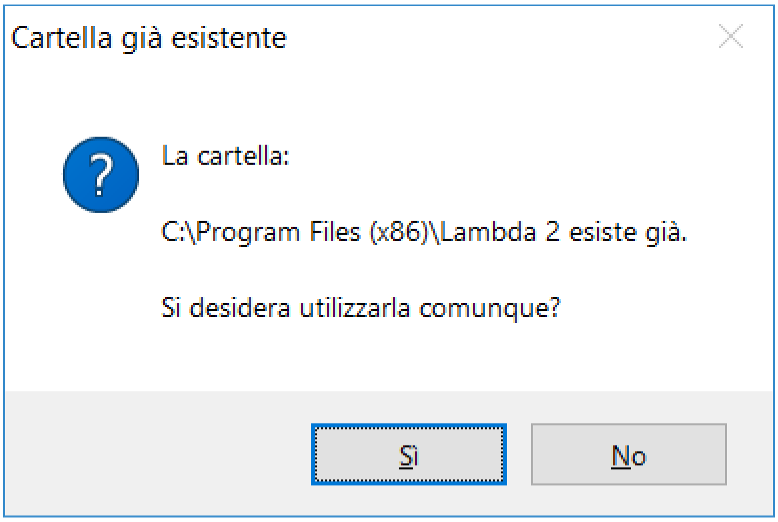
.





To install a new version of Lambda2.0, choose the language and select "Yes" to remove the previous version. If the program is still open, close it and manually remove any remaining components before continuing with the installation.

The removal process will not completely delete the LAMBDA2.0 folders, but will ask if you want to use the same folder. If you do, click Yes and proceed. Otherwise, you can choose a new installation folder."



At this point, the procedure continues in the same way as already described above.

4.3 What are the JAWS and NVDA screen reader scripts

Script is a program that adds information elements to the screen reader to fully adapt to the LAMBDA2.0 window and use appropriate speech synthesis (correct pronunciation of mathematical terms) and Braille display (adaptation of Braille code on the display to Lambda code).

There are scripts for:

JAWS

NVDA

The installation of the scripts is essential if you use a PC with synthesized voice and/or Braille line, but it can be omitted if the program is used by others (teachers, operators...). During the installation, the version of JAWS is identified if it is installed in a standard way; if it is not, or if you have multiple versions of JAWS, it will ask you which version of Jaws is installed on your PC. If you do not know this information, open the Jaws window (Insert J) and activate the "?" button on the menu, then select "Information about Jaws". The installation procedure for the NVDA script is similar to that of JAWS.

5 mathematical notation in LAMBDA

5.1 The Lambda mathematical text

To write a mathematical text, you need to have a much larger number of symbols than a normal literary text. Each of them should be associated with:

a graphical symbol for video representation and ink printing.

one or more 8-dots Braille signs.

a textual expression that is pronounced through the speech synthesis. The LAMBDA 2.0 editor offers several support tools for entering characters not present on the keyboard and facilitates their recognition: their full name appears on the status line at the bottom and can be read by the speech synthesis when the cursor is positioned there.

Remember that in a linear code it is the marking symbols (tags) that act in relation to each other to define a block, that is, a portion of text delimited by an opening and a closing. A block can be enclosed by the usual parentheses (round, square, and curly) but also by symbols that delimit a fraction (numerator and denominator), or a root, exponent, or other. The LAMBDA2.0 system recognizes these relationships and offers various tools to efficiently manage both the interrelated markers and the portion of text they define: it will be possible to automatically switch from one to the other, delete both with a single operation (very useful during simplification), select the entire contents of a block (from the "open" symbol to the corresponding "close") to copy, move, delete, temporarily hide the text contained within parentheses to highlight the general structure of the formula, and more.

The text entered in LAMBDA 2.0 must follow some simple formal rules otherwise its structure cannot be recognized and it will not be processed. For example, for every symbol that opens a block (such as a parenthesis) there must be a corresponding closing symbol, without interweaving or false nesting. It is considered an error, for example, to write [x + 3 (xy + 2x)]. A minimum of formal correctness of the mathematical text is also essential to be able to obtain transformations in MathML and therefore access the tools for conversion into other formats. However, such errors do not affect the display in graphic mode which accurately reports what is written in Braille, in this way the teacher viewing the graphic representation in black can also correct structural errors. The program has various tools designed to facilitate the entry of a formally correct mathematical text. All block closing markers, for example, are inserted with a single command (Ctrl k), leaving the system the task of defining which is the nearest currently open structure level and choosing the corresponding closing marker.

5.2 Display on monitor

Every mathematical symbol is represented on the screen with a graphic character. Many of them are easily recognizable because they are analogous or related to their usual graphic notation. However, some symbols, such as markers indicating a fraction, are specific to linear systems and do not have a corresponding representation in traditional graphic mathematical texts; they will therefore be represented with symbols that we are not used to using in other contexts.

In the video representation, colours are used to highlight the different roles of the symbols used in the LAMBDA code. The function of the colours is only supplementary and is not necessary for understanding the formulas. Consult the page on the meaning of colours in LAMBDA.

5.3 Braille representation

One or more 8-dots Braille characters are associated with each marker and mathematical symbol. We do not have enough Braille combinations to represent all the symbols in a scientific-mathematical text, so some of them require two or more Braille characters to be displayed. Nevertheless, our system will always treat them as a single character and will therefore allow them to be entered, modified, deleted, and selected as a single character. As an example, consider the following symbol:

≠

[not equal]

it is represented in LAMBDA by a pair of characters.

⍁=⢬⠶(3,4,6,8 e 2,3,5,6 on Braille display for Italian code)

In defining these compound symbols, we have tried to make them easier to recognize with logical or mnemonic connections; in the case considered, for example, the symbol "not equal" is formed by a negation prefix that precedes the usual equal character. Other combinations are based on easily recognizable pairs, for example >> for "much greater"; +- for "more or less". See: complete table of Braille symbols. (insert link or page)

5.4 Textual symbols

Some mathematical elements do not have associated symbols but are indicated textually. For example: trigonometric functions (sen, cos, sec...), logarithms (log, ln), the limit (lim). Among the mathematical texts, some have a textual representation using traditional alphabetical text. However, such symbols are handled and recognized by the system as a single element and can only be deleted, selected, and moved globally, not acting on the individual characters that make them up. The insertion can be done using the menus, but the simplest way is by keyboard: they are inserted by typing in sequence the characters of the text that make them up. To insert the limit element, therefore, it will be sufficient to write lim. The system recognizes the limit element after typing the third letter that makes it up.

5.5 Double symbols

To facilitate memorization and recognition, some mathematical elements are represented with a pair of adjacent symbols. I.e.

$\gg$ Much greater**>>**

$\pm$ Plus or minus**+-**

$\geq$ greater or equal **>=**

These symbols will appear as two characters on both the screen and the Braille display, but the synthesizer will only pronounce the overall name of the element. In addition, they are entered and recognized by the system as a single element and can only be deleted, selected, or moved as a whole, not by acting on the individual characters that make it up. The input can be done using menus, but the simplest way is through the keyboard: they are inserted by typing the two symbols that make them up in sequence.

5.6 Meaning of colours in LAMBDA notation

To improve the readability of the text presented on the video, graphic symbols are also associated with colours: there are red markers that delimit a block's structure (open/close, intermediate, and any intermediate);operators and single markers (without closing) are green; numbers, letters, and isolated symbols are black.

For example:

$$x + \frac{x-1}{x+1}$$

x + fraction with (x + 1) over (x - 1)" is represented as

x+//x-1ø x-1\\

following the rule:

- the markers that delimit the fraction (open, intermediate, closed) are red:

// ø \\

- - operators and single markers are green:

\+ -

- - the numbers and letters are black:

x 1

Note: do not confuse the open-close parentheses (double markers) in red with those that do not require closing (in green) as in this example to represent open or closed intervals.

1<x<=3 (1;3]

Non-mathematical text, that is, alphanumeric such as titles, comments, explanations, or other is delimited by the open text marker and close text marker and is blue in colour. Example:

🄰Exercise number 5🄰

6 Mathematical structures

Elements are the main components of the Lambda mathematical notation system. Elements include symbols, operators, functions, attribute markers, and so on. Elements can be single or have an open-close structure (sometimes open-intermediate-close in certain cases). Open-close elements contain, with successive nesting, other elements, and it is possible to represent all mathematical constructions in this way. The structure is a tree: the main element (tree) has its own structure and contains objects (branches) that in turn are structured to contain other objects. At the end of the chain are the elements that do not contain others; they are symbols, characters, digits...

6.1 Unique elements

Single-marker structures obviously have only one marker (known as a tag) that can have different roles depending on how it is connected to other objects. There are four possible structures: (ob stands for object and <one> for single marker).

A1 <one>

A2 <one> [ob1]

A3 [ob1]<one>[ob2]

A4 [ob1] <one>

A1 <one>

This is a typical single symbol or character (analogous to MathML tokens). For example, the number 5, the Greek letter λ (lambda), the symbol ∞ (infinity). They are not connected to any object.

A2 <one> [ob1]

The marker <one> precedes the object. It generally refers to attribute markers or operators that only act on the object that follows. Example: the logical negation ¬p

A3 [ob1]<one>[ob2]

The marker is placed between two objects.

It can be for example a binary operator when both the first and second objects are unambiguously defined, even without the need for additional markers. For example, in a simple power:

a^2 a raised to the power of 2

We have:

[base]<one>[exponent]

Even a common arithmetic operation (such as addition) is of this type.

a+b [a plus b]

A4 [ob1] <one>

The marker follows the object. It is, for example, the case of a factorial..

$5!$ *[5 factorial]*

[number]<one>

6.2 Structure Open-Close

The open/close structure is very common. Note that in Lambda, only the opening marker needs its own command, while the closing is automatically inserted by the program with a single command valid for all structures of this type (quick selection key Ctrl + k), or if selected in preferences, both opening and closing appear simultaneously. <open> stands for opening marker, <close> stands for closing marker The possible combinations are:

<open>[Ob1]<close>
[Ob1]<open>[Ob2]<close>
[Ob1]<open>[Ob2]<close>[Ob3]
<open>[Ob1]<close>[Ob2]

<open>[Ob1]<close>

Structures with parentheses are an example of this type.
(a+b)

[Ob1]<open>[Ob2]<close>

An example of this type is compound power

$$x^{a+1}$$

for example, for power we have [base]<open>[exponent]<close>

6.3 Open-Intermediate-Closed Structure

The structure with three markers has, in addition to an opening and a closing, also an intermediate marker. <sep> stands for the separator or intermediate marker For example, for the fraction:

$$\frac{a+1}{a-3}$$

We will have: <open>[numerator]<sep>[denominator]<close>

In some cases, the intermediate marker is optional: this is the case, for example, of the nth root which becomes a square root in the absence of an intermediate marker (and therefore with a <open><close> structure).
For example:

$$\sqrt[3]{30-3}$$ [cubic root of (30-3)]

It has the separator

```
<open>index<sep>radicand<close>
```

The entering the values:

```
<open>3<sep>30-3<close>
```

A square root could have been indicated with index = 2, but it could also have been indicated more simply by omitting the index and separator.

$$\sqrt{a+b}$$ [root of (a+b)]

The entering the values this the result:

```
<open>2<sep>a+b<close>
```

But it is better:

```
<open>a+b<close>
```

Some structures can have two separators. This is the case for example of the definite integral.

7 Main structures in LAMBDA

The LAMBDA2.0 system includes several block structures, with open and closed markers, as well as an optional intermediate marker. The complete list is available in appendix 1.

(you must have the new LAMBDA 2.0 list with the new symbols. Some of the most frequently used structures are described. Note that the quick selection keys for closing and intermediate are always the same:

Fraction

Root

Exponent

7.1 Fraction

The compound fraction has an open-intermediate-closed structure. The intermediate corresponds to the fraction sign. If the numerator and denominator are simple, uniquely defined objects (e.g. numbers, letters, etc.), then a simple fraction can be used

Structure:

<tag_open> *compound fraction and numerator*

<tag_separator>*open compound fraction and numerator*

<tag_close>*close denominator and entire fraction*

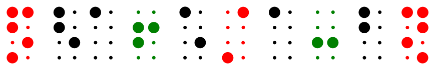
Graphic example:

$$\frac{2a+1}{a-b}$$ [(2a+1) divided by (a-b)]

Example in Lambda linear writing

⫽2a+1ϕa-b⑊

Example in Italian Braille 8-dot syntax:



shortcuts

		or, on the numeric keypad if you are using the extended keyboard profile
Open:	*Ctrl Q*	*Alt /*
Intermediate:	*Ctrl I*	*Alt +*
Close:	*Ctrl K*	*Ctrl +*

See also: Simple fraction

7.2 Root

The nth root has an open-intermediate-closed structure. The intermediate separates the root index from the radicand. The square root can have an open-closed structure; therefore, in the absence of the intermediate, the root is understood to be squared (index = 2). The square root can also be represented in simple form (see Simple Square Root). Higher roots (i.e., with an index other than 2) always require the compound form with an open-intermediate-closed structure.

structure:

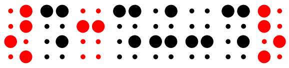
<tag_open>*index*<tag_separator>*root*<tag_close>

Graphic example:

$\sqrt[3]{30-3}$ cube root of (30-3)

Linear structure in LAMBDA:

8 dots examples:



Shortcuts

		or, on the numeric keypad
Open:	*Ctrl uppercase R*	*Alt **
Intermediate:	*Ctrl I*	*Alt +*
Close:	*Ctrl K*	*Ctrl +*

Version without separator:

As mentioned, the index can be omitted (and in this case, the separator is not needed) with square roots:

structure:

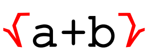
```
<tag_opem>root<tag_close>
```

(implied index=2, i.e. square root)

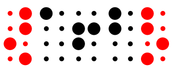
graphic example:

$\sqrt{a+b}$ [root of (a+b)]

Linear LAMBDA representation:



8 dots:



See also: Simple square root.

The compound exponent has an open-closed structure.

7.3 Exponent

If the exponent is a simple object, defined in a unique way (e.g. numbers, letters...) the simple exponent can be used.

structure:

< tag_open> exponent< tag_close>

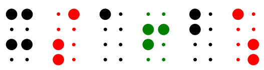
Graphic example:

x^{a+b} [x power (a+b)]

Linear LAMBDA representation:

x ↟ a + b ↟

8 dots:



shortcuts

		Or into numeric keypad
Open	*Ctrl + ^*	*Alt -*
Intermediate:	*Ctrl I*	*Alt +*
Close:	*Ctrl K*	*Ctrl +*

7.4 Simple and compound forms

For simple objects, i.e. those made up of a single well-defined element, it is advisable to also provide a shorter notation without a closing marker. This is an abbreviation strategy adopted in almost all 6-dots Braille math codes, useful for speeding up writing operations and making the math text more compact.

For example, to represent.

$\sqrt{3}$[the square root of 3]

instead of the full structure

√ 3 ⦘

we can write simply

√3

In Lambda, it is possible to write the three most common structures in simple mode:

- simple fraction
- simple square root
- simple exponent

It is important to note that the markers for simple versions are different from those for their compound counterparts, and that the commands to enter them are also different. When writing a fraction, square root, or power, it is necessary to decide immediately whether to use the simple or compound form. If the user so wishes, they can always use the compound form: they will have fewer symbols and commands to learn, but their math text will be more prolix. There is nothing to prevent, for example, a simple fraction like

$\frac{a}{b}$[a divided by b]

being written using the compound structure:

// a ⌿ b \\

instead of

a / b

However, we will have a text of 5 characters instead of 3, as in the simple notation, and this in an expression with many fractions can cause a significant expansion of the text. In the same way, a compound expression can be written in simple form if the elements are uniquely defined with blocks. In this case as well, however, a larger number of characters will have to be used. For example, this fraction

$$\frac{2a+1}{a-b}$$

since it does not have unique terms would require the compound form

//2a+1/a-b\\

but it can also be written with the simple form if the blocks are properly defined.

(2a+1) / (a-b)

7.5 Simple fraction

The simple fraction can only be used if the numerator and denominator are composed of a single element or are unambiguously defined as a single block

structure:

numerator <tag_separator>*denominator*

graphic example:

$\frac{a}{b}$[a divided by b]

LAMBDA representation:

a / b

8 dots :

⠁⣌⠃

hotkey

/ (slash) (also on numeric keypad)

See also: Compound fraction

7.6 The simple square root

The simple square root can only be used if the radicand is composed of a single element or is unambiguously defined as a single block.

structure:

<tag_single_operator>root

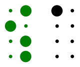
graphic example:

$\sqrt{a}$[root of a]

LAMBDA representation:

√ a

8 dots:



hotkey

Shortcut key Ctrl R (numeric keypad: Ctrl *) Note: the nth root (cubic, fourth, etc...) cannot be abbreviated, even if the radicand consists of only one element.

See also : Compound root

7.7 The simple exponent

The simple exponent can only be used if it consists of a single element or is defined unambiguously as a single block.

structure :

base <tag_single_operator>exponent

graphic example:

x^2[x to the second]

LAMBDA representation:

x ^ 2

8 dots:

⠭⡮⠣

shortcuts:

^ (uppercase + ì) (numeric keypad: Ctrl -)

See also : Compound exponent

8 LAMBDA code repertoire

The appendix contains a complete table with graphical examples and representations in both LAMBDA code and Braille. The 8-dot Braille symbols used for topics are listed in the tables. Additionally, the tables indicate whether there are any closing or intermediate symbols (indicating the math object's structure) as well as double symbols:

Numbers

Latin Characters

Greek Characters

Attributes

Parentheses

Sets

Arithmetic Operators

Relational Operators

Logic

Algebra

Geometry and Vectors

Trigonometry

Analysis

Symbols

Arrows

Logarithmic Functions

Braille Only

8.1 Numbers

Braille characters - Italian 8-dot code.

Element	Symbol
0	346
1	16
2	126
3	146
4	1456
5	156
6	1246
7	12456
8	1256
9	246

Note: the numbers indicate the combination of dots used.

8.2 Latin characters

Braille characters - Italian 8-dot code.

Element	Symbol
a	1
b	12
c	14
d	145
e	15
f	124
g	1245
h	125
i	24
j	245
k	13
l	123
m	134
n	1345
o	135
p	1234
q	12345
r	1235
s	234
t	2345
u	136
v	1236
w	2456
x	1346
y	13456
z	1356
A uppercase	17
B uppercase	127
C uppercase	147
D uppercase	1457

E uppercase	157
F uppercase	1247
G uppercase	12457
H uppercase	1257
I uppercase	247
J uppercase	2457
K uppercase	137
L uppercase	1237
M uppercase	1347
N uppercase	13457
O uppercase	1357
P uppercase	12347
Q uppercase	123457
R uppercase	12357
S uppercase	2347
T uppercase	23457
U uppercase	1367
V uppercase	12367
W uppercase	24567
X uppercase	13467
Y uppercase	134567
Z uppercase	13567

Note: the numbers indicate the combination of dots used.

8.3 Greek characters

Braille characters - Italian 8-dot code.

Element	Symbol
alfa	345 1
beta	345 12
gamma	345 1245
delta	345 145
epsilon	345 15
zeta	345 1356
eta	345 125
theta	345 245
iota	345 24
cappa	345 13
lambda	345 123
mi	345 134
ni	345 1345
csi	345 1346
fi	345 124
omicron	345 135
pi	345 1234
ro	345 1235
sigma	345 234
tau	345 2345
upsilon	345 136
chi	345 12345
psi	345 13456
omega	345 2456
Alfa uppercase	345 17
beta uppercase	345 127
gamma uppercase	345 12457
delta uppercase	345 1457
epsilon uppercase	345 157
zeta uppercase	345 13567
eta uppercase	345 1257

Theta uppercase	345 2457
iota uppercase	345 247
cappa uppercase	345 137
lambda uppercase	345 1237
mi uppercase	345 1347
ni mai	345 13457
csi uppercase	345 13467
omicron uppercase	345 1357
fi uppercase	345 1247
pi uppercase	345 12347
ro uppercase	345 12357
sigma uppercase	345 2347
tau uppercase	345 23457
upsilon uppercase	345 1367
chi uppercase	345 123457
psi uppercase	345 134567
omega uppercase	345 24567

Note: the numbers indicate the combination of dots used.

The first symbol indicates the Greek letter prefix

8.4 Attributes

Braille characters - Italian 8-dot code.

Element	Symbol
Left subscript	57
Subscribed	458
Subscript	28
Left superscript	1278
Superscribed	456
Simple superscript	457
Underlined	23568
Overlined	12458
Tilde	2346
Caret	3467
Asterisk	358
First	7
Second	38540
Third	’’’

Note: the numbers indicate the combination of dots used.

8.5 Brackets

Braille characters - Italian 8-dot code.

Element	Single opening symbol	**Closing**
Round bracket	236	356
Square bracket	2367	3568
Brace	2368	3567
Angular bracket	12378	45678
Generic bracket	1238	4567
Open round bracket	236	
Close round bracket	356	
Open square bracket	2367	
Close square bracket	3568	
Open brace	2368	
Close brace	3567	
Absolute value	1234567	1234568
Double bar	4568 4568	4568
Vertical bar	4568	
Period	3	
Decimal separator	2	

Note: the numbers indicate the combination of dots used.

Any double characters are separated by a space.

8.6 Sets

Braille characters - Italian 8-dot code.

Element	Single opening symbol	Intermediate	Closing
Indexed union	48 235	456	3456
Indexed intersection	48 35	456	3456
Union	48 235		
Intersection	48 35		
Set difference	48 36		
Symmetric difference	48 1457		
Cartesian product	48 378		
Strictly included	48 12678		
Strictly includes	48 34578		
Included in a loose sense	48 12678 2356		
Includes in a loose sense	48 34578 2356		
Not contained	3468 48 12678		
Not contained or equal	3468 48 12678 2356		
Does not contain	3468 48 34578		
Does not contain or equal	3468 48 34578 2356		
Belongs	123567		
Contains	1567		
Does not belong	3468 123567		
Complementary	48 147		
Set of parts	48 P		
Cardinality	card		
Empty set	123458		

NB: the numbers indicate the combination of dots used.

Any double characters are separated by a space.

Some symbols are represented in text mode

8.7 Arithmetic operators

Braille characters - Italian 8-dot code.

Element	Symbol
addition	235
subtraction	36
multiplication	378
product	35
division	256
Plus and minus	235 36
Minus and plus	36 235
factorial	2357
semi factorial	2357 2357
Per cent	123456
Per thousand	235678
Whole part	123478
Generic operator	358
exponential notation	157

Note: the numbers indicate the combination of dots used.

Any double characters are separated by a space.

8.8 Relation Operators

Braille characters - Italian 8-dot code.

Element	Symbol
Less than	12678
Greater than	34578
Less than or equal to	12678 2356
Greater than or equal to	34578 2356
Much less than	12678 12678
Much greater than	34578 34578
Precedes	1234567 12678
Follows	1234567 34578
Equal to	2356
Approximately equal to	12356
Congruent	1234567 2356
Coincides with	2356 2356
Different from	3468 2356
Proportional to	234568
Divisor	4568
Prime divisor	1234567 4568

Note: the numbers indicate the combination of dots used.

Any double characters are separated by a space.

8.9 Logic

Braille characters - Italian 8-dot code.

Element	**Symbol**
or	258
And	1248
Not	3468
For all	2358
Exists	145678
Does not exist	3468 145678
Exists and is unique	145678 2357
Exclusive or	1234567 258
Contradiction	1234567 F
Tautology	1234567 T
Boolean sum	1234567 235
True proposition	T
False proposition	F

Note: the numbers indicate the combination of dots used.

Any double characters are separated by a space.

Some symbols are represented textually.

8.10 Algebra

Braille characters - Italian 8-dot code

Element	**Single opening symbol**	**Intermediate**	**Closing**
simple fractions	3478		
composed fraction	12467	47	13458
Simple power	23467		
power with compound exponent	347		168
simple square root	2468		
compound nth root	3458	25	1267
summation	345 2347	456	3456
Product symbol	345 12347	456	3456
determinant	4568		4568
system of equations	2368 2368	Enter or 3 space	3567 3567

Note: the numbers indicate the combination of dots used.

Any double characters are separated by a space.

8.11 Geometry and vectors

Braille characters - Italian 8-dot code.

Element	Symbol
Parallel	123568
Not parallel	3468 123568
Perpendicular	34567
Not perpendicular	3468 34567
Incident	incid
Arc	1234567 1568
Angle	1568
Vector	1368
Scalar product	1234567 378
Vector product	1234567 35
Tensor product	1368 378
Degrees	267

Note: the numbers indicate the combination of dots used.

Any double characters are separated by a space.

Some symbols are represented textually.

8.12 Trigonometry

Braille characters - Italian 8-dot code.

Element	**Symbol**
Sine	sen
Cosine	cos
Tangent	tg
Cotangent	ctg
Secant	sec
Cosecant	csec
Arcsine	asen
Arccosine	acos
Arctangent	actg
Arccotangent	acctg
Arcsecant	arcsec
Arccosecant	arccosec
Hyperbolic sine	sinh
Hyperbolic cosine	cosh
Hyperbolic tangent	tanh
Hyperbolic cotangent	coth
Hyperbolic secant	sech
Hyperbolic cosecant	cosech
Hyperbolic arcsine	arcsinh
Hyperbolic arccosine	arccosh
Hyperbolic arctangent	arctanh
Hyperbolic arccotangent	arccoth
Hyperbolic arcsecant	arcsech
Hyperbolic arccosecant	arccosech

NB: Symbols are represented in textual mode

8.13 Analysis

Braille characters - Italian 8-dot code.

Element	Unique symbol or opening	Intermediate	2° intermediate	Closure
Indefinite integral	124568			1458
Definite integral	124568	457	3456	1458
Curvilinear integral	124568	3456		1458
Circuit integral	1234567 124568			1458
Derivative	1458			14578
Nth derivative	1458	3456		14578
Partial derivative	14568			14578
Nth partial derivative	14568	3456		14578
Limit	lim	167		3456
Lower limit	liminf			
Upper limit	limsup			
Differential	1458			
Laplacian operator	1234567 1457			
Function composition	6			

Note: the numbers indicate the combination of dots used.

Any double characters are separated by a space.

Some symbols are represented textually.

8.14 Symbols

Braille characters - Italian 8-dot code.

Element	Symbol
Set of natural numbers	1234567 N
Set of integers	1234567 Z
Set of rational numbers	1234567 Q
Set of real numbers	1234567 R
Set of complex numbers	1234567 C
Arithmetic progression	34568
Geometric progression	12368
Infinity	2458
Integral	124568
Nabla	124567
Alef	12567
Euro	1578
Dollar	46

Note: the numbers indicate the combination of dots used.

Any double characters are separated by a space.

Some symbols are represented textually.

8.15 Arrows

Braille characters - Italian 8-dot code.

Element	Symbol
Left arrow	348
Right arrow	167
Down arrow	1268
Up arrow	12468
Implication	1234567 167
Bijective relation	348 167
If and only if	1234567 348 167

Note: the numbers indicate the combination of dots used.

Any double characters are separated by a space.

8.16 Logarithmic functions

Braille characters - Italian 8-dot code.

Element	One symbol or opening	Closing
Natural logarithm	ln	
Decimal logarithm	Log	
Logarithm base	log	3456
Antilogarithm	antilog	

8.17 Braille only

Braille characters - Italian 8-dot code.

Element	Symbol
Negation prefix	3468
Greek letter prefix	45
Set prefix	48
Generic prefix	134568

9 Symbols insertion

To insert symbols that are not present in the keyboard, the LAMBDA2.0 editor offers four possibilities:
a combination of quick selection keys, some also on the numeric keypad
selection from the menu;
dynamic search in the list of elements;
selection through graphic buttons (for teachers and other sighted users)

9.1 Quick Selection Key Combination, the default configuration

The quick selection keys have been set in the full default profile. One or more keys are associated with LAMBDA code elements for quick insertion, to be used in combination with the CTRL key. Profiles and quick selection keys can be modified and customized by users as described in the section "Profiles and Customization of Quick Selection Keys." If you use a computer with an extended keyboard, you can also assign a combination using the numeric keypad, so that you can mainly type with only the right hand and leave the left hand free to instantly check on the Braille display the text that is being entered or modified. The numeric keypad combinations can be associated, in addition to CTRL, with the ALT key. Such combinations must be defined in a custom profile.

For less used symbols, a pair of characters has been defined: the first letter indicates the group, the second the associated key. For example, to insert the Greek letter α (lowercase alpha), type Ctrl + g , a (that is: while holding Ctrl, press the g key, then press a). All letters of the Greek alphabet will be entered in a similar way: Ctrl + g , associated Latin letter, uppercase or lowercase. A particular type of quick selection keys, particularly simple and intuitive, is used for elements represented textually, such as trigonometric and logarithmic functions. In this case, simply type the text normally on the keyboard and the system will recognize the associated element. The sequence of keys required, for example, to insert the element "cos" (cosine) will be "cos" itself.

See: Complete list of hotkeys
Insertion with numeric keypad .

9.2 Insertion with the numeric keypad

Many users who use the Braille display find it convenient to use the numeric keypad for writing, with Num Lock enabled. This is because it can be handled with only the right hand, so that the left hand can be placed firmly on the display and instantly control the text entered. The advantages are much fewer if the user is used to using the numeric keypad, with Num Lock disabled, to control the screen reader. They are obviously null if the user uses a laptop. With the LAMBDA2.0 editor, it is possible to create a custom profile and insert, on the numeric keypad, in addition to numbers and arithmetic operators, other common mathematical elements. The following tables show the layout of the numeric keypad according to a standard profile designed for the extended keyboard with numeric keypad. Lambda2.0, having the function of being able to customize the quick selection commands, does not allow more keyboard and numeric keypad commands to be associated with the same element. The complete list of default quick selection keys, including those associated with the numeric keypad, is also included in the appendix.

Here is a proposal of the default profile

normal

<table>
<tr><td></td><td>Simple fraction separator</td><td>*</td><td>-</td></tr>
<tr><td>7</td><td>8</td><td>9</td><td rowspan="2">+</td></tr>
<tr><td>4</td><td>5</td><td>6</td></tr>
<tr><td>1</td><td>2</td><td>3</td><td>Enter</td></tr>
<tr><td colspan="2">0</td><td>.</td><td></td></tr>
</table>

Pressing the CTRL key

<table>
<tr><td></td><td>division</td><td>simple square root</td><td>simple exponent</td></tr>
<tr><td>x</td><td>display compressed structure</td><td>a</td><td rowspan="2">close block for general use</td></tr>
<tr><td>Duplicate row</td><td>symbol selection window</td><td>b</td></tr>
<tr><td>(</td><td>[</td><td>{</td><td></td></tr>
<tr><td colspan="2"></td><td>=</td><td></td></tr>
</table>

Press ALT key

	Open compound fraction	Open compound fraction	open compound exponent
			general purpose separator

10 Default profile hotkey list

lists of all keyboard shortcuts used in LAMBDA2.0

10.1 Standard Windows commands

	alphanumeric keyboard
Open an existing document	CTRL O
New document	CTRL N
Close document	CTRL F4
copy selected text to clipboard	CTRL C
Cut the selected text to the clipboard	CTRL X

Paste the contents of the clipboard	CTRL V
Stop an operation	ESC
Undo a performed operation	CTRL Z
Redo or redo an operation	CTRL Y
Save	CTRL S
Print	CTRL P
Select all	CTRL A
Close the application (exit Lambda)	ALT F4

10.2 Viewing or editing commands

Alphanumeric keyboard default profile and numeric keypad default profile

	Alphanumeric keyboard	**Numeric keypad**
View the structure in expanded mode (typing F8 again - or CTRL 8 in the numeric keypad - you switch to the compressed structure)	F8	CTRL 8
View the structure in collapsed mode (or with F8 twice - or CTRL in the n.k.- you enter the expanded mode and immediately switch to the other one)	uppercase F8	CTRL 8
Commands active in display mode: Switch to the other view	F8	CTRL 8
Skip to next page	Page Forward	
Skip to previous page	Page Back	

Go back to the normal editor and close any other pages open in the editor	Esc	
Select the block (from an open marker to the corresponding closed one	CTRL B	
Commands active with block selected *Extend the selection*	CTRL B	
Reduce selection	UPPERCASE CTRL B	
It simultaneously erases the other connected marker and any separator. Valid only if the cursor is positioned on a marker.	UPPERCASE CANC	
Duplicate line (copy twice and remove spaces)	CTRL D	
Active commands with persistent blocks: *Cancel selection*	ALT B, E	
Delete selected text	ALT CANC	
Go to the corresponding open/separator/close	CTRL arrow	
Go to next marker	ALT arrow	
View the document in graphic mode by opening the browser window; if already open, it updates the display	F4	
Close the Graphic View window	Shift F4	

10.3 Commands for inserting symbols or markers

	Alphanumeric keyboard	**Numeric keypad**
General		
Opens the search and selection box	F5	CTRL 5
Close block (insert required closing marker)	CTRL K	CTRL +
Intermediate marker (insert intermediate marker)	CTRL I	ALT +
Inserts a text section block	CTRL J	
More frequently used		
Compound fraction (opening marker)	CTRL Q	ALT /
Simple fraction (fraction sign)	/	/
Division (operator)	CTRL 7	CTRL /
Complex exponent (opening marker)	CTRL ^	ALT -
Simple exponent (operator)	^	CTRL -
Compound nth root (opening marker); - if the intermediate is missing, it is a compound square root	CTRL UPPERCASE R	ALT *
Simple square root (operator)	CTRL R	CTRL *
Open round bracket.	(	CTRL 1

Open square bracket.	[	CTRL 2
Open brace	{	CTRL 3
Equal (=)	=	CTRL .
Double characters		
Algebra / Analysis.		
general prefix for Analysis and Algebra (must always be followed by another character as indicated below)	CTRL M	
Natural logarithm	CTRL M L	
Logarithm in a generic base if the implicit base 10 is missing.	CTRL M UPPERCASE L	
Definite integral	CTRL M I	
Double integral	CTRL M I I	
Limit	CTRL M T	
Summation	CTRL M S	
Product	CTRL M P	
Determinant	CTRL M D	
Sets		
General prefix for sets (always followed by another character)	CTRL E	
Empty set	CTRLE 0 (zero)	
Belongs to	CTRLE E	
Intersection	CTRLE I	
Union	CTRLE U	

Logic		
General prefix for logic elements (always followed by another character)	CTRLL	
And	CTRLL A	
Boolean sum	CTRLL B	
Contradiction	CTRLL C	
False	CTRLL F	
For every	CTRLL P	
Not	CTRLL N	
Or	CTRLL O	
Tautology	CTRLL T	
Exist	CTRLL E	
True	CTRLL V	
Geometry and trigonometry		
General prefix for geometry and trigonometry (must always be followed by another character)	CTRL T	
Angle	CTRL T A	
degrees	CTRL T G	
Incidence	CTRL T I	
Parallel	CTRL T P	
Vector	CTRL T V	
Sin	CTRL T S	
Cos	CTRL T C	

Tangent	CTRL T T	
Greek letters		
General prefix for Greek letters	CTRL G	
Greek letters are obtained by following the CTRL G prefix with the associated Latin letter, uppercase or lowercase. Associations are generally easily identifiable; only the less obvious cases are reported here:		
eta →h		
theta →j		
ksi →x		
khi →q		
psi →y		
omega →w		
Examples:		
lowercase delta	CTRL G D	
uppercase delta	CTRL G UPPERCASE D	
Lowercase omega	CTRL G W	
uppercase omega	CTRL G UPPERCASE W	

10.4 Calculator

There are three options for using the calculator in Lambda 2.0:

• opening it for ad-hoc calculations anywhere in the Lambda sheet;

• calculating a previously written numeric expression

• inserting the result when it is difficult to remember.

In the window that opens by typing F9, you can type the operation to be performed, even non-arithmetic, according to the functions listed in the menu. You can also configure the calculator according to the options on the number of decimals you want to display and on the unit of measure of the angles for the trigonometric functions.

The second submenu of tools is "calculate expression" which is activated with Ctrl F9. This command, in the presence of a series of operations, opens a window presenting the result. This result remains in memory and can be inserted into the page with Ctrl Shift F9

Active commands from the editor	**Alphanumeric keyboard**
Open calculator	F9
Calculate selected expression and display the expression with result	CTRL F9
Active commands from the calculator window	
Numbers, the operators + - * / ^ and parentheses are permitted for direct formula entry.	
Calculate expression	enter
Close calculator	ESC *or* Alt F4
Clear the contents of the display	Canc

10.5 Matrices

10.6 and tables

In Lambda2.0, it is possible to have a two-dimensional representation both as a matrix and as a group of elements with their own structure and usable for specific elements. From the Insert menu, you can select one of the following items:

Binomial coefficient

System of equations

Generic table

Table for decomposing prime numbers

Table for inequalities

Table for the Ruffini method

All of these elements (after being inserted) can be viewed in table form by accessing the "table" text with the cursor and pressing F10, which allows for navigation by row and column using the arrow keys, as with the cells of a spreadsheet.

1) Selecting Binomial Coefficient brings up the structure (open, separator, close) - on screen with round brackets - of a table with one column and two rows. Numbers, letters, expressions can be inserted and managed globally with the Duplicate and Display various keys.

2) Choosing Equation System brings up the structure (open close) and it will be possible to insert all equations one under the other. It can be managed globally with the Duplicate and Display various keys.

3) Clicking on Table brings up a window for its configuration with the number of rows and columns indicated (by default 3 but it is possible to increase or decrease this number). The structure navigable with F10 appears (open; row separators and column separators according to the previously chosen number; close) (they appear on the screen with square brackets). It will be possible to add or delete rows and columns and insert any element: letters, numbers or expressions.

4) To decompose into prime factors, just click the entry in Insert, Tables, table of decomposition into prime factors and a predefined structure navigable with F10, a

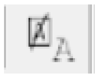
table with 2 columns and 10 rows appears. It will be possible to insert numbers and letters (not expressions) and modify the number of rows with Add, Insert or Delete."

5) The table of signs and the table of common parts for solving inequalities are the modes that are normally taught to students to visualize common solutions in case of products or quotients of inequalities or in systems of inequalities. In both cases, it is a graphical representation that, while certainly helpful for sighted students, may be forced and complex for blind students. We have chosen to create a structure that imitates the procedure used in class - even for those who transcribe textbook - but we remain convinced that it is not the optimal solution. In any case, we have created this structure with compensatory elements that can help as much as possible in the use of this method. By choosing in the Insert Table menu: Table for inequalities, a configuration window appears that requires the insertion of the number of expressions to be compared (number of rows to which a default summary row will be added) and the list of points (numbers in increasing order, separated by space) involved in the inequalities. A sort of table appears where each row shows the numbers previously entered in the dialogue window that are preceded and followed by spaces. In these spaces, the signs + and - will be inserted appropriately chosen in the resolution of the various inequalities and in the final row the sign of the overall inequality can be counted. The advantage of this structure is the facilitated navigation with the F10 key in rows and columns as in the previous tables. In case of systems of inequalities, it will be possible to insert in the preceding and following spaces symbols instead of signs + and -, arbitrary symbols (for example V and F, or * where the inequality is satisfied and no character in other spaces). Also in this case, the final control will be particularly easy on the last row.

6) The Table for the Ruffini method is a construction used to factorize particular polynomials and involves manipulating only the coefficients of the monomials that make up the polynomial previously ordered by decreasing degree of the main variable. In the Insert Table menu, select the Table for the Ruffini rule. A guided configuration window appears, which requires the insertion of the coefficients of the ordered polynomial, separated by spaces. A table with 3 rows and a number of columns corresponding to the number of previously inserted coefficients plus one column preceding them is then created, where the known term of the divisor

polynomial will be inserted. Then, operations are performed according to known rules. The advantage of this structure is the facilitated navigation with the F10 key in rows and columns as in all tables

10.7 Text selection

The shortcut key that inserts a text section when in a math environment, or inserts a math section in the text environment	CTRL J
Insert menu, Change context	
Copying and pasting a text (even from another text program) in a mathematical context.	
Copying and pasting a mathematical fragment (even taken from another text program) in a mathematical context.	
By tapping the icon from the visual menu.	

10.8 Selection from menu

From the menu bar, open the insert menu and select the group that interests you. Then select the symbol to be entered. Inserting by menu selection can be useful in certain special situations, especially when you are not sure about the name of the symbol to be inserted and you prefer to scroll through a predefined list.

When the name of the symbol is known, it is best to use the search on the dynamic list, via F5..

10.9 Searching in the list of elements

By pressing the F5 key, or the corresponding entry in the insert menu, the complete list of all elements in alphabetical order is opened. By starting to type the name of the element to be searched in the box at the top, the list is reduced, showing only the names of elements that contain words that begin with the entered text. Just two or three characters are enough to get a list that is compact enough to be easily consulted with synthesis or the Braille display. This is considered the fastest input system when you do not know or do not remember the quick selection keys.

10.10 Selection by graphic buttons

For sighted users, the possibility of inserting mathematical elements through a graphic menu with icons (toolbar) is offered. The most common symbols, including commands for inserting the intermediate marker and closing, are present in the element toolbar (left group), with direct access. Other elements are available in the mathematical toolbar by first selecting the group and then, in the new menu that opens, the chosen symbol. The meaning of the symbols is quite intuitive. In case of doubt, place the mouse pointer over it; a small explanation window appears. The toolbar is fully active even in the textual section. In this case, the element is accompanied by a pair of text markers (close and open).

10.11 Distinguishing between text and mathematics

LAMBDA2.0's editor has two distinct environments, one for text and the other for mathematics, and it is possible to freely switch between them, even in the same line. The rules and mathematical functions described in this manual only apply in the mathematical environment. In the text environment, the available commands are the basic ones of a common text editor. Every new LAMBDA document opens by default in the mathematical environment. To enter the text environment, type Ctrl + J or select Text section in the Insert menu. When a text section is opened, the closing marker is automatically inserted right away, with the cursor positioned within the two A and A markers (text and end of text).

ACalcola: A

The markers appear identical on both the screen and the Braille display (dots 123467 in the Italian code). Not to overly slow reading, the two symbols are generally ignored by speech synthesis. The name of the text section markers is only pronounced when exploring the line point by point, moving the cursor with the arrow keys. It should be relatively easy for the user to determine whether they are in a text or maths section. In addition to the contextual information (the contents of the two environments are clearly different), the type of reading provided by synthesis is completely different. In the lower status bar, the name of the element on which the cursor is positioned appears if you are in the mathematical environment. However, the cursor letter always appears if you are in the text environment. The text section on the screen is entirely blue, while the mathematical section uses three different colours (black, green, red) depending on the type of element. To exit the text section and go to the maths section, you must move to the right with the cursor arrow when positioned at the end of the text. This will bypass the closing marker.

Example

By typing Crtl + J, both the open and closed text markers are inserted, and the cursor is positioned in between them.

🄰 | 🄰̸

The non-mathematical text is written freely. The screen reader works as in any text editor, and the synthesis reads the words in the usual way.

🄰Calculate expression🄰̸

At the end of the text, the cursor is moved outside the text area with the right arrow key. Now we are in the maths area: all the commands for editing maths offered by LAMBDA are active, and the synthesis pronounces the name of the various mathematical elements. This is one of the rare cases in which the LAMBDA editor inserts both markers at the same time: the blocks of the mathematical environment are in fact first opened and then closed, with two distinct commands.

The distinction between the two environments, text and maths, is rigorous. Their function is completely different, and any situation of ambiguity must be avoided. For this reason, both markers must always be present, and should therefore be inserted or deleted simultaneously. It is therefore not possible to delete the markers to transform a text into maths or vice versa: the markers can only be deleted together with the text contained within them.

Any selection of text copied and pasted into a maths block will automatically be delimited by the two text markers. The same will happen when copying a maths selection into a text one.

Example

Let's transform the previous example by inserting the formula within the text. The mathematical part is selected, and it is cut with Ctrl + X.

Verify that this equation is an identity

2+x/2+1=3+x/2

When it is inserted with Ctrl + V, automatically the two resulting text blocks are correctly marked with new opening and closing symbols.

11 Manipulating Mathematical Text

For a mathematical editor intended for use in schools, it is not enough to be able to write an expression or equation, it is also necessary to be able to process it in order to solve it appropriately. The LAMBDA2.0 editor offers various tools to facilitate these manipulations. The resolution by transformation (copy, paste, modify) is the most suitable system for managing mathematical expressions that require successive calculations and partial transformations to reach the result. The line duplication function is useful in this case. It is recommended to also consult the section on tips for using the Braille display in this regard.

11.1 Resolution by transformation

Copying and pasting a line and making corrections on the copy is the simplest way to work with Braille text in many circumstances.

For a sighted person dealing with an expression or equation to be solved with subsequent transformations, it is normal to perform many steps when transcribing a new line (calculations, simplifications, etc.). This operation is not possible with Braille tools that allow access to one line at a time, so it is not possible to read a line of the expression and simultaneously rewrite it, transform, lower down. Working by correction is much more effective, i.e. first copying the text and then reading and processing it with the Braille display. Copying and pasting a line is an operation that can be done with the normal editing tools available in LAMBDA2.0, according to the usual procedures common to all writing programs. For example:

go to the beginning of the line and type Shift + End to select it all

type Ctrl + C to copy it

go down with the cursor and type Ctrl + V to paste the text in the new position.

The procedure is described in more detail, with an example, on the Automatic line duplication page of the next paragraph

11.2 Automatic line duplication

This method offers the possibility of controlling the steps taken through check lines, which are not modified. Corrections are made in overwrite (without insertion) mode, so the overall structure of the expression remains unchanged. To delete characters, they are replaced with a space so that the overall length of the formula does not change.

Example:

2[x(x-1)-1+x(3-x)]=2(1+6x)+4*check line*

2[x^2-x -1+3x-x^]=2 -12x +4*Working line*

In the second line (working line), the elements that have not been modified have remained exactly below those corresponding to the check line. This way, by moving the cursor of the Braille display upward, they will remain stable while the modified elements will vary, and will therefore be easily identifiable. In the next steps, the copied lines are compressed, eliminating empty spaces. Here is the complete solution of the previous example (in red the check lines):

2[x(x-1)-1+x(3-x)]=2(1+6x)+4*check line*

2[x^2-x -1+3x-x^2]=2 +12x +4*Working line*

2[x^2-x-1+3x-x^2]=2+12x+4*check line*

2[-1+2x]=6+12x*Working line*

2[-1+2x]=6+12x*check line*

-2 +4x =6+12x*Working line*

-2+4x=6+12x*check line*

-8x=8*Working line*

-8x=8*check line*

x=-1*Working line*

The procedure may seem much longer than normal, but many operations are performed automatically and very quickly. Note that if necessary, it is possible to check the entire process in reverse, verifying the various steps. The check lines are always equivalent to the previous line of work, from which they differ only by any additional or fewer spaces. The LAMBDA editor has a command (Ctrl + D shortcut) that automatically performs the duplication of the line according to this method. In particular, Ctrl + D performs these operations in sequence: 1 - select the entire line where the cursor is located (it is not necessary to jump to the beginning) 2 - copy twice below the previous line, deleting any spaces present; 3 - at the end of the operation, the cursor is positioned at the beginning of the lower line (working line).

12 Tips for using the Braille display with LAMBDA

Using the Braille display with the LAMBDA editor makes it easier and faster to explore and manipulate mathematical expressions. Almost all Braille displays provide the user with some buttons that allow to recall specific functions to explore the contents of the window. The most common functions are:

Window start / end. Move the Braille display respectively to the first or last line of the window;

Line start / end. Move the Braille display respectively to the beginning or end of the current line.

Line right / left. Move the Braille display right or left on the line where is positioned. If the end of the line is reached, the Braille display is moved to the next one. If it is at the beginning of the line, it is moved to the previous one.

Previous / next line. Move the Braille display respectively to the previous or next line.

Move to cursor. Move the Braille display to the cursor's position.

Cursor routing. By pressing the cursor routing keys, located above each cell of the display, the cursor is moved to the indicated position.

These functions offer many advantages when using the Braille display with the LAMBDA2.0 editor. Below are described techniques for exploring and manipulating mathematical expressions using only the presented functions. Exploration:

Example 1 - Exploration of the working environment

Example 2 - Exploration of a mathematical expression Manipulation:

Example 3 - Comparison with the previous step

Example 4 - Solution of an inequality

12.1 Example 1- Work environment exploration

Since the LAMBDA editor allows you to load multiple documents and move from one to another, it may be useful to know in which document you are writing. This is immediately possible by pressing the "Start window" key on the Braille display. The Braille display moves to the first line of the window where the name of the current document appears. When you start writing again, the Braille display will automatically move to the cursor position. It is also possible to force a return to the cursor position by pressing the "Move to cursor" key

While writing, it is essential to be able to quickly find out the information contained in the status line (the writing mode, insertion or overwriting, if the "automatic line break" option is active, the cursor position within the document, the name of the symbol at the cursor position, etc.). With the Braille display, you can reach the status line by pressing the "End window" key and then the "right/left line" keys.

12.2 Example 2 - Exploring a mathematical expression

Mathematical expressions can be read on the Braille display and explored by pressing the "right/left line" and "start/end line" keys.
However, when the complexity of the mathematical expression structure is high, the tools offered by LAMBDA, such as the expanded or compressed structure display (F8 key), can be useful. On the Braille display, it is possible to quickly understand which blocks make up a specific part of the mathematical expression. If you need to know the content of one of these blocks, you can press the cursor routing key next to the start marker, the separator, or the end marker, and then press Page up. If you want to continue writing inside a certain block, you can press the cursor routing key to move the cursor to the block of interest and then press Esc to exit the structure display mode and work in the editor window.

12.3 Example 3 - Compare with the previous step

To perform the next step in simplifying a mathematical expression, you can select the current expression and copy it to the next line. If "Automatic line break" is not active, while reading and modifying the copied expression, you can immediately compare it with the expression at the previous step by pressing the "Previous line" button. If "Automatic line break" is active, you will need to press the "Previous line" button multiple times and explore each individual line with the "Right/left line" functions. In either case, to return to the cursor position, you need to press the "Move to cursor" button.

12.4 Example 4 – Solution of an inequality

student must solve the inequality.

`x^3+x-2>0`

1) he writes in the editor:

`x^3+x-2>0`

To factor the polynomial, use Ruffini thus:

A)cancel the polynomial;

B)writes the coefficients of the polynomial (x^2 has coefficient 0).

To write them:

* he presses the cursor routing key corresponding to cell 4 or 5. He starts writing the coefficients a few columns to the right because he will have to enter -1 on the next line;

* writes the coefficients: 1 0 1 -2

(splits the coefficients with two or three spaces to be able to insert the appropriate number under each coefficient);

C) returns with Enter. If auto-indent is enabled, he will have to press home to go to the beginning of the line;

D) he writes -1 and returns with Enter;

E) reads the first coefficient, previously entered, by pressing the "Previous line" key twice. Keeping his finger on the cell of the Braille display where he reads the first coefficient, it moves two lines below. Presses the cursor routing key above the cell and the cursor moves to the same column as the coefficient. This is possible thanks to the particular function of the LAMBDA editor which allows you to write anywhere in the window.

F) he writes the read coefficient;

G) moves up with the Braille display to the previous line by pressing "Previous line". Can read -1 on the left, multiplies the coefficient just entered by -1, remembers the result. Moves up one line with the Braille display. Reads the second coefficient. Moves down one line keeping his finger on the same cell of the Braille display. Presses the cursor routing key corresponding to the cell so the cursor moves exactly under the second coefficient. Writes the result of the multiplication;

H)He holds finger on multiplication result. Moves the Braille display down one line by pressing "Next line". Presses the corresponding cursor routing key. The cursor moves to the second column of coefficients, on the third row, exactly below the result of the previous multiplication. Writes the result of the difference between the second coefficient and the number in the second line. Both can be read easily by going up a line or two on the Braille display;

I) Repeats the above steps (E to H) as appropriate to complete the algorithm

J) Presses Enter to go to the next line. On the last line read the coefficients of the new polynomial. Write:

```
(x-1)(x^2+x+2)>0
```

K) Proceeds to solve the inequalities

```
X-1>0
```

and

```
x^2+x+2>0
```

L) He solves the inequality with the rule of signs:

1- Place the cursor on the Braille display's midline by pressing the corresponding cursor routing key. Write 1. Press Enter;

2- Move left by 1 and write - (sign of the first inequality);

3- Move right by 1 and write +;

4- Press Enter. Thanks to the auto-indentation function, the focus is exactly under the minus sign;

5- Write +;

6- With the Braille display, read the position of the + on the previous line, move to the current line, and press the cursor routing key;

7- Write +;

8- Press Enter and write the results of the rule of signs on the next line.

Note: All operations performed with cursor routing keys can be done with arrow keys, but take more time.

13 Selection of mathematical text

Lambda has very powerful functions for selecting part of the text and copying it to the memory buffer. The basic functions are practically identical to those of a normal writing program. This includes the ability to select text, copy it to the main buffer, and then paste it as desired. Others, more complex, are intended for expert users. We would like to mention in particular:

The commands to select portions of mathematical text by recognizing its structure (blocks selection).
The ability to keep the selection active even if the cursor moves away from the selected part (persistent blocks).
The management of multiple buffers simultaneously to store, and retrieve, multiple different information (multiple buffers).

13.1 Block selection

It is possible to select, with a single command, the entire mathematical block within which the cursor is placed. By "block" we mean the portion of text enclosed between a pair of open/closed markers, such as two parentheses, a compound root, or other. The command can be activated from the menu (Selections/Select block) or with the Ctrl B shortcut
At the beginning, the smallest block containing the cursor is selected; the selection can be extended, by typing Ctrl B again, and gradually include the outermost open/close structures, until the entire line is reached. Similarly, it is possible to subsequently reduce the selection and return to the previous inner blocks, until the starting selection (smallest block containing the cursor) is reached.
shortcut:

Select block	Ctrl B
Extendselection	Ctrl B
Reduceselection	Uppercase, Ctrl, B

See also:

Shortcut Commands - Select text

13.2 Storing in multiple buffers

Experienced users may find it useful to be able to buffer various sections of text so that they can be retrieved flexibly according to their needs. (Copy, Cut, Paste) use the main buffer: it is unique, so each new data recorded overwrites the previous one. Using multiple buffers, it is possible to store multiple data sets simultaneously to be retrieved at a later time (up to nine memories). It is also possible to add data to the stored data, without deleting it.

In the Edit menu under the buffer section, there are three commands available for managing multiple buffers:

Cut to buffer

Copy to buffer

Paste from buffer

The cut, copy, and paste commands operate similarly to the usual commands, but each time the user is asked in which, or from which, of the 6 buffers they want to act. The contents of the buffer can be pasted directly using the paste command and selecting the value 11

See also: Shortcut Commands - Select text

14 Alternative views

In linear representations, structural information is harder to grasp than in normal graphical representations. The problem mainly concerns complex maths objects, with many elements inserted one inside the other, on multiple levels (nesting). LAMBDA's editor has two alternative display modes designed to facilitate the understanding of the structure of formulas and internal relationships, overcoming, as far as possible, the limitations of linear notation. Being display tools, and not writing tools, it is not possible to modify the text inside them. However, you can freely navigate by moving the cursor that, when you return to the normal window, will retain the new position. Compressed structure display, Expanded structure display.

14.1 Compressed structure

The "compressed structure" display shows the formula by emptying the contents of a block, from one marker to the other. In this way, it is clear to which marker each block is associated with and which part of the formula it acts on

For example, the equation

$$1+\sqrt{\frac{x^2-y^2}{x+y}(x-y)}=0$$

[1+square root of (((x^2+y^2)divided by(x+y))*(x-y))=0]

in LAMBDA representation

1+√⫽x^2−y^2∤x+y⑊(x−y)⦘=0

It becomes clear that there is a loss of information, structure, and relationship compared to the graphical representation.

The compressed structure of this formula will be, at the highest level, as follows:

1+√⦘=0

Decreasing the level makes other blocks visible:

1+√⫽∤⑊()⦘=0

Compression level is based on cursor location (innermost block) and can be adjusted with Page Forward/Back buttons.

Associated commands:

To enter collapsed structure view	*F8* *or Ctrl 8 in the num pad for the num pad profile*
To return to the normal window	*Esc*
To switch to expanded tree view	*F8* *or Ctrl 8 on the numeric keypad* *(with F8 you alternately switch from one structure to another)*
To reduce the level of the display	*Back page* (Pag🡅)
To increase the level of the display	Page *forward* (Pag🡇)

See also:

Expanded structure.

14.2 Expanded structure

The "expanded structure" display is similar to the compressed one: the hidden blocks are not eliminated but replaced by spaces. The formula will be less compact but useful information is obtained on the size of the blocks. Taking the same example from before:

1+√⫽x^2−y^2∤x+y⑊(x−y)⟩=0

the expanded structure of the formula will be, at the highest level, the following:

1+√   ⟩=0

Here too, by decreasing the level, the other blocks become visible:

1+√⫽   ∤   ⑊(   )⟩=0

Associated commands:

To enter the expanded structure view	*Shift + F8* *or 2 times F8* *or Ctrl 8 in the numeric keypad using the numeric keypad profile*
To return to the normal window	*Esc*
To switch to collapsed design view	*F8or Ctrl 8 on the numeric keypad(with F8 you alternately switch from one structure to another)*
To reduce the level of the display	*Pageback*(Pag↑)
To increase the level of the display	*Next page*(Pag↓)

See also: Compressed structure

15 Particular structures

The LAMBDA editor pays particular attention to some two-dimensional mathematical objects whose management with a purely linear system, such as that required by Braille or vocal access systems for the visually impaired, is generally considered complex. We are particularly referring to equation systems and matrices.

- **Systems of equations**
- **Matrices**

15.1 Equation systems

In LAMBDA, systems of equations can be displayed both in multiple lines and in a single line.

They must always be delimited by a pair of markers represented on the screen and in Braille by two adjacent curly brackets (considered as a single symbol). Equations on the same line are separated with at least three consecutive spaces (it should be remembered that in LAMBDA, three spaces are usually used to separate formulas that are placed on the same line but should be considered separately).

Example of a system on multiple lines:

```
.{x+y+z=0
{  x=2y
{  3y-z=0}.
```

Example of a system on one lines:

```
.{x-2y=0   x=2y}.
```

To insert a new system of equations

Look for 'System of equations' in the insert menu or in the object list (F5). The system markers will appear, with the cursor inside.

```
.{}.
```

Insert the first equation of the system. To move on to the next one, press Enter if you prefer to keep the system on multiple lines, three spaces if you want to stay on a single line.

At the end, press right arrow to exit the system.

The Ctrl D command to duplicate and carry out the calculation by substitution copies the whole system even if on two staves, or in the case of a two-equation system or three lines for three-equation systems.

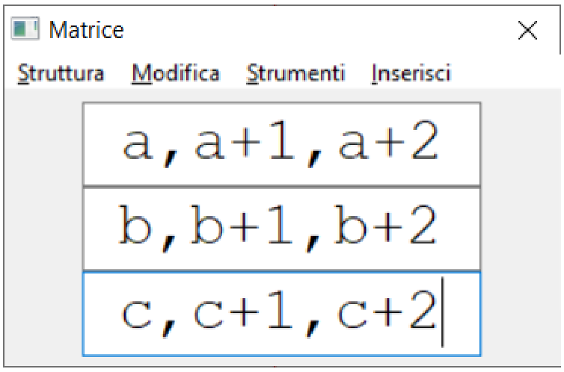
15.2 Matrices

The LAMBDA editor provides the user with special tools for working with matrices, based on the close interaction of the two access systems: linear and two-dimensional (or tabular).

Linear access is convenient for small-sized matrices because it allows to examine the entire object on the Braille display at the same time. Table view is useful for working on more complex elements: reading on the display will necessarily involve only one line at a time, but the user can still understand the vertical relationships by scrolling through the different lines.

The following example shows the linear and tabular representation of the same matrix:

{a,a+1,a+2;b,b+1,b+2;c,c+1,c+2}



15.3 Insert a matrix

Inserting a matrix into a document happens in two steps:

- inserting the structure (number of rows and number of columns);
- inserting the content of the cells

To insert the structure:

Choose the table and matrix option from the insert menu (or via F5). The focus must be positioned in a mathematical section of the document. A window appears where you are asked to enter the number of rows and columns of the matrix;

Type the number of rows and columns that make up the matrix in the Rows and Columns fields. If the fields are not completed, the matrix will have 3 rows and 3 columns. You can later change the number of rows and columns of the matrix; confirm by pressing the Confirm button or by pressing the Enter key.

The matrix structure is inserted into the document in linear form, that is, as a sequence of markers: For example, if a matrix of 2 rows and 3 columns is defined, it appears in the document as follows:

{ , , ; , , }

To insert the content:

After inserting the matrix structure, the contents of the cells can be introduced in two ways: by writing directly in the linear structure inserted in the document or by using the two-dimensional mode.

To write in the linear structure:

Once the linear structure of the matrix has been inserted in the document, the focus is on the start matrix marker. The cell at row 1 and column 1 of the matrix is therefore located to the right of the focus. If you want to insert the content of this cell, just press the right arrow key once. If you prefer to fill the cells of the matrix in a different order, just move with the right arrow or left arrow keys within the linear structure. The exploration is facilitated by the presence of the end of column and end of row markers, which appear on the Braille display and are read by the speech synthesizer;

Type the content of the cell. In typing, you can use all the usual LAMBDA editor commands.

To use the two-dimensional mode:

Once the linear structure of the matrix is inserted, select the two-dimensional element view from the View menu or press the F10 key to enter the two-dimensional mode;

a window appears consisting of: the number of rows and columns of the matrix, the menu bar, the cells of the matrix arranged in two dimensions and separated by the end of column marker. The focus is positioned in the cell in the first row and first column of the matrix. You can reach other cells with the arrow keys or with the Braille display cursor routing keys. Exploration is facilitated by the presence of the end of column marker that can be easily identified on the Braille display and by the vocal reading of the current focus position in the matrix. Move the focus to the cell where you want to write;

type the content of the cell. The insertion of the mathematical expression in the cell can be done using all the insertion commands of the LAMBDA editor (shortcut keys, element list, and Insert menu present in the menu bar). Note that when any symbol is inserted in a cell, the other cells are correctly aligned so that the correct two-dimensional arrangement is never lost. This is particularly important for the Braille display user who can thus explore the two-dimensional structure using the movement keys. This helps to quickly understand the mutual relationships between the cells, thereby facilitating the identification of regularities in the matrix or the execution of particular operations (e.g. calculating the determinant);

if necessary, modify the content of the cells using select, copy, and paste. In two-dimensional mode, it is possible to select groups of adjacent cells using Shift+arrow keys. You can then copy and paste them with Ctrl+C and Ctrl+V. To leave the two-dimensional mode and return to the document, you must press the Esc key. The focus is positioned on the matrix start marker.

15.4 Changing the structure of a matrix

The structure of a matrix can be modified in two ways: redefining the number of rows or columns that make it up, or introducing or deleting rows or columns in specific positions

Redefine the number of rows or columns of the matrix:
position the focus inside the matrix or on the beginning/end matrix markers; Select the two-dimensional mode with F10. From the structure menu it is possible to add a row and the matrix is extended downward, insert a row and the row is added immediately below, delete a row and the row where the cursor is positioned is deleted From the structure menu it is possible to add a column and the matrix is extended to the right, insert a column and the column is added immediately to the right, delete a column and the column where the cursor is positioned is deleted. Note that:
if the number of rows increases, the matrix is extended downward;
if the number of columns increases, the matrix is extended to the right;
if the number of rows or columns decreases, the matrix is reduced from the bottom or from the right. The content of the deleted rows or columns is lost.

Note that from the linear representation of the matrix in the document it is not possible to make changes to the structure.

To delete a matrix, make the selection with Shift+movement keys;
press the Delete key.
Note that it is only possible to delete an entire matrix from the linear editor, not from the view.

15.5 Manage matrix groups

Very often in mathematics it is necessary to operate on more than one matrix. For example, in the sum or product of matrices, at least three matrices are used: two or more operands and the result. For this reason, the LAMBDA editor offers users an operational mode to manage the exploration and processing of multiple matrices.

Select matrices.

The LAMBDA editor automatically selects the matrices to work on. All matrices in the same mathematical expression are selected, i.e up to the end of the line on which the focus is. For example, in the expression:

```
3*{2,1;5,4}+5*{2,6;-7,9}-{5,0;8,-2}
```

the three matrices present are selected;

16 Graphic display

The linear formula written with the LAMBDA editor can be displayed in a graphical mode in a separate window on the screen. The display uses the MathML conversion performed by LAMBDA. The graphical display window opens with the quick selection key F4, or through the menu by selecting View and Graphics. It can be moved and resized as desired; to close it, use the SHIFT F4 key or, from the menu, View and Close graphics, or finally use the usual Window commands to close a window.

It is also possible to have the graphical display in the web browser (such as Internet Explorer) by selecting the View and Graphics menu in the browser.

To print the graphic page, you can use the appropriate voice of the menu that appears by clicking on the display window with the right mouse button, or in the browser from the main menu.

If the formula is not correct, the display is almost always possible even if the formula is not correct, even with errors in the structure (incomplete or incorrect blocks). In this way, the teacher can check through traditional graphic display all the steps of their student. Example:

```
10+(⫽a+5⌀b+5)
```

The system will interpret a possible form of display by eliminating the fraction marker and the parentheses placed in the wrong way. In order to reduce errors of this kind, or forgetfulness of closing the structure, the option of complete element insertion has been introduced in the preferences.

Appearance and content: LAMBDA is a mathematical writing system oriented to the content of the document, not to its graphic appearance. The transformation into graphic display goes first through the conversion into MathML content which is subsequently displayed in graphic mode by the browser. It may happen that the displayed formula, although equivalent in content, has a different appearance than the one initially intended to represent.

17 Calculator

LAMBDA has a scientific calculator designed to be easily used with Braille peripherals and voice synthesis. It can be used in two distinct ways: as a tool connected to the editor or as a standalone environment to be used in a separate window.

17.1 Calculator linked to the editor

The calculations are performed by selecting a portion of text directly in the editor and activating the calculator. You can view the operation in a window or keep the result in memory and paste it later in the editor wherever and whenever you want. The calculator can process any portion of text in a mathematical environment, even with subsequent and nested calculations, as long as the data consists only of numbers or known and defined constants.

Then expressions such as

⫽√⌈3⌉ 64⌉⁄18/3^2⑊+5

5^2*⧉p (⧉p , also know as π , is a known constant)

but not expressions that contain undefined variables, such as

12+a

In this case, an error message like this would appear:

“The element 'a' in the expression is not valid”

To activate the calculator connected to the editor, these commands are used (each of them can be activated, in addition to the shortcut keys indicated here, with the relative voice of the tool menu):

Calculator window	*F9*
View expression (displays in a separate, read-only window, the last calculated expression along with its result;)	*Ctrl F9*
Paste result (inserts the result of the last calculated expression at the current cursor position)."	*Ctrl Shift F9*

17.2 Calculator windows

In this calculator, expressions are entered into a text window and then calculated. The writing possibilities are significantly reduced compared to the LAMBDA editor and this system should therefore be used mainly for simple calculations, primarily entered with normal keyboard commands. In addition to numbers and the 4 operations (+,-,*,/), round brackets and the simple power sign ^ are accepted. Other calculations can be performed through the Operations menu: roots, logarithms, trigonometric functions, and more.

Calculator commands active from editor (calculator closed)

Open calculator;	*F9*
Paste result, inserts the result of the last calculated expression at the current cursor position	*Ctrl Shift F9*

Calculator commands active from the calculator window (calculator open)

Calculate	Enter
Close calculator and return to editor	*ESC* or *Alt F4*
Delete the contents of the cell	DEL o Canc
Calculates the square root of the number in the display (or of the result, if there is a valid expression in the box)	*Ctrl R*
Calculate the cube root of the number in the box (or of the result, if there is a valid expression in the box)	*Ctrl Shift R*
Natural Logarithm	Ctrl L
Decimal Logarithm	*Ctrl Shift L*
Integer	Ctrl I

Exponential	Ctrl E
Change sign	Ctrl -
Sine	Ctrl S
Cosine	Ctrl C
Tangent	Ctrl T
Arcsine	*Ctrl Shift S*
Arcsine	*Ctrl Shift C*
Arctangent	*Ctrl Shift T*

17.3 Changing calculator settings

The settings are defined only in the calculator window but are valid for both modes of use. To change the settings of the calculator connected to the editor, you must open the calculator window. The settings that can be defined are:

The number of decimal places displayed (from 0 to 5);

The angle measurement system, to be chosen between sexagesimal degrees, radians, and gradients.

18 Import - export

The LAMBDA editor offers various import and export tools.

Import

- MathML

Export

- MathML
- XHTML

18.1 Import from MathML

The editor can import MathML (.mml) files, both content and presentation, converting them into LAMBDA2-0 code. They are automatically recognized, converted, and displayed in a new editor window.

To import a MathML file:

Select the File menu, then Import MathML.

This will open the dialog allowing you to select the MathML file (.mml extension) to convert.

18.2 Export to MathML

You can export LAMBDA documents in content or semantic MathML format.

The LAMBDA code is content oriented and therefore favours the transformation towards MathML of content, but it is possible to export according to both MathML notations.

When the file contains mathematical elements not expected in the content MathML it uses mixed code, part appearance part content.

To export a MathML file

The editor exports the currently loaded file in memory; therefore, open the file if it is not already. Select the File menu, then Export, and finally, choose between Presentation MathML or Content MathML, depending on the chosen format. The usual dialog box will open that allows you to name the MathML file (extension .mml) to be created and decide in which folder to save it.

18.3 Export to XHTML

The XHTML (eXtensible HyperText Markup Language) markup language, is an extension of HTML code. Its goal is to preserve both the presentation and structure of information. It is possible to export LAMBDA documents in XHTML in order to view them with an external browser.

To export a file in XHTML

To export a file in XHTML, the editor exports the currently loaded file in memory; open the file if it is not already.

Select the File menu, then Export and finally XHTML.

The usual dialog box will open that allows you to name the file (the extension is .xml) to be created and decide in which folder to save it.

19 User profiles

19.1 Profile Description

One of the most common strategies employed by users is to use the shortcut key combinations that allow them to speed up the insertion of elements or make some procedures immediate (such as CTRL+d to duplicate the expression, or CTRL+i or CTRL+k for intermediate and closing).

Obviously, it is not possible to associate a "hot" key with every mathematical element present in LAMBDA2.0, and therefore some choices have been made, prioritizing the elements that occur most frequently.

With Lambda2.0, it will be possible to add, remove, or change the assignments of these combinations according to the needs (for example, when measuring angles, you can associate keys to degrees, first and second) by creating a new profile. In the Profiles menu, the list of active default profiles that can be selected appears. To create a new one, just click on Customize:

- Add a new profile with +
- Give it a Name and check the items
    - Copy information from the source profile
    - Copy the state
    - Copy the shortcuts (this way all the default assignments are kept)
- In the right window, choose the element to which you want to associate a new hotkey (for example, first look in attributes and select)
- In the appropriate space, enter the chosen key (for example, CTRL + 1)
- If the key is already occupied, a warning is displayed
- Otherwise, by pressing enter the profile is saved and will be active until you want to remove it (again in Profiles, Customize, click on "-", the minus sign)

19.2 The pre-set profiles

The LAMBDA editor can be customized to meet user needs. In particular, it is possible to simplify the menu for entering mathematical elements by hiding those that are not used. You can hide an entire group (e.g. trigonometry) or one or more elements, choosing them from those listed in the group

It may be appropriate to also hide those that are used very often and that are usually inserted with direct keyboard commands. If their presence in the menus appears unnecessary, it is advisable to remove them to make them more compact and faster to consult. Each customization is called a profile and will be saved in a separate file; the name of the active profile is reported on the status bar (last item on the right). In profiles, it is also possible to modify the shortcut keys associated with various mathematical elements.

Profiles can be built for the general needs of a student, that is, for the type of studies followed and the class attended, but they can also vary based on specific or contingent needs. For example, we can design profiles for set theory, logic, trigonometry ... in which to make more immediate access to symbols or operators used very frequently, both through menus (by placing them in the first positions) and shortcut keys (assigning simpler and more compact combinations).

Some user profiles, already set up, are provided along with the program.

Basic: It is a profile suitable for elementary and middle school students; in addition to numbers, letters, and basic operators, it contains the elements necessary for elementary algebra (parentheses, fractions, roots, powers), some attributes for characters and numbers, and the main symbols of numeric sets.

Complete: It is the most complete profile, suitable for students in the last three years of high school and university. All mathematical elements of the LAMBDA system are present.

19.3 How to change a profile

To upload an existing profile:

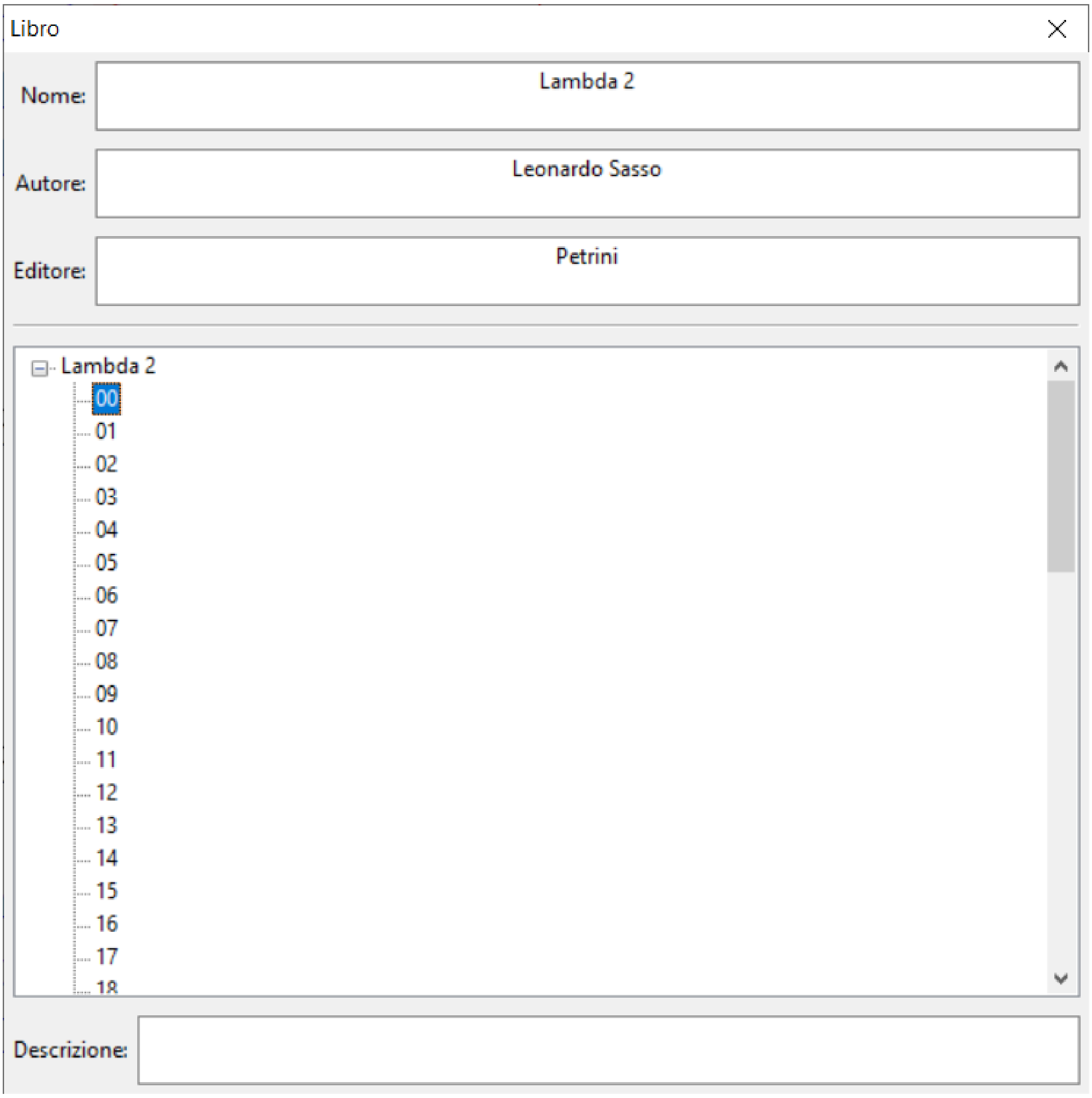
Go to Profile Menu, Customize

A window appears with two working areas, the one on the left shows the default profiles and the customized ones.

Select the profile that interests you and on the right side are all the mathematical elements of Lambda2.0. For each element (or group of elements) it is possible to enable or disable it through the checkbox "Enabled". It is possible to assign a shortcut key combination to each element. It can be entered in the appropriate box by typing it out in the form CTRL + key or CTRL + key1, key2 for double combinations (for example: CTRL + K in the first case, CTRL + G,d in the second). If the shortcut is already in use, a warning appears. At the end of the work, the profile file should be saved, possibly with a different name if you want to create a new profile. In the window there is also a button with the + sign to add a new profile, a minus sign to delete a profile, an up and down button to scroll through the list of existing profiles. When the program opens, it automatically loads the last profile that was used.

20 Menu for Lambda books

The "Books" menu allows you to open books in the lambdabook format (a file containing an entire text book in lambda format). Once the file is opened via "Books / Open", you can navigate the entire structure of the chapters both within the menu and through the dedicated "Books / View" window. By selecting a chapter within the structure, it will automatically be opened as a new Lambda document within the editor.



21 Appendix Lambd2.0 mathematical elements and hotkeys

general	
comma	(Ctrl+,)
point	
exp_sep	
repeating_decimals	(Ctrl+-)
numbers	
1	
2	
3	
4	
5	
6	
7	
8	
9	
0	
Latin_characters	
a	(Ctrl+Num 9)
b	(Ctrl+Num 6)
c	
d	
e	
f	
g	
h	
i	
j	
k	
l	
m	
n	
o	
p	
q	
r	
s	
t	
u	

v	
x	(Ctrl+Num 7)
y	
z	
w	
B	
C	
D	
E	
F	
G	
H	
I	
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	
T	
U	
V	
X	
Y	
Z	
W	
A	
Greek_characters	
gamma	(Ctrl+G, G)
alfa	(Ctrl+G, A)
capital_gamma	(Ctrl+G, Uppercase+G)
beta	(Ctrl+G, B)
delta	(Ctrl+G, D)
epsilon	(Ctrl+G, E)
eta	(Ctrl+G, H)
theta	(Ctrl+G, J)
lambda	(Ctrl+G, L)
mi	(Ctrl+G, M)
ni	(Ctrl+G, N)

pi	(Ctrl+G, P)
ro	(Ctrl+G, R)
sigma	(Ctrl+G, S)
tau	(Ctrl+G, T)
fi	(Ctrl+G, F)
chi	(Ctrl+G, Q)
omega	(Ctrl+G, W)
capital_sigma	(Ctrl+G, Uppercase+S)
capital_delta	(Ctrl+G, Uppercase+D)
capital_theta	(Ctrl+G, Uppercase+J)
capital_fi	(Ctrl+G, Uppercase+F)
capital_lambda	(Ctrl+G, Uppercase+L)
capital_omega	(Ctrl+G, Uppercase+W)
capital_pi	(Ctrl+G, Uppercase+P)
embellishments	
superscript	
upper_line	
subscript	
overscript	
underscript	
tilde	
hat	
up_left_index	
down_left_index	
underlining	
asterisk	
first	
second	
third	
fences	
absolute_value	
round_bracket	(Ctrl+Num 1)
square_bracket	(Ctrl+Num 2)
curly_bracket	(Ctrl+Num 3)
decimals_separator	
thousands_separator	
open_round_bracket	
close_round_bracket	
open_square_bracket	
close_square_bracket	
open_curly_bracket	
close_curly_bracket	

double_bar	
angle_brackets	
generic_brackets	
sets	
empty_set	(Ctrl+E, 0)
union	(Ctrl+E, U)
intersection	(Ctrl+E, I)
cartesian_product	
complement	
element_of	(Ctrl+E, E)
not_element_of	
subset_or_equal	
superset	
superset_or_equal	
power_set	
cardinality	
subset	
contains	
union_underover_limits	
intersection_underover_limits	
symmetric_difference	
not_contained	
not_contained_equal	
not_containing	
not_containing_equal	
arithmetic_operators	
plus	
minus	
times	
inline_divide	(Ctrl+7)
percent	
perthousand	
plus_or_minus	
factorial	
cross_multiplication	
semifactorial	
minus_or_plus	
integer_part	
generic_operator	
exponential	
relational_operators	

less_than	
equals	(Ctrl+Num 0)
greater_than	
less_than_equal	
greater_than_equal	
not_equal	
congruent	
proportional	
much_less_than	
much_greater_than	
equivalent	
precedes	
succeeds	
almost_equal	
prime_divisor	
divisor	
logic	
true	(Ctrl+L, V)
false	(Ctrl+L, F)
not	(Ctrl+L, N)
and	(Ctrl+L, A)
or	(Ctrl+L, O)
boolean_sum	(Ctrl+L, B)
forall	(Ctrl+L, P)
exists	(Ctrl+L, E)
not_exists	
there_exists_exactly_one	
such_that	
tautology	(Ctrl+L, T)
contradiction	(Ctrl+L, C)
excludind_disjunction	
algebra	
power	
compound_power	(Ctrl+Uppercase+Ã¬)
compound_root	(Ctrl+Uppercase+R)
compound_fraction	(Ctrl+Q)
root	(Ctrl+R)
fraction	
summation	(Ctrl+M, S)
product	(Ctrl+M, P)
determinant	(Ctrl+M, D)

geometry_and_vectors	
vector	(Ctrl+T, V)
scalar_product	
vectorial_product	
incident	(Ctrl+T, I)
parallel	(Ctrl+T, P)
not_parallel	
perpendicular	
not_perpendicular	
angle	(Ctrl+T, A)
degrees	(Ctrl+T, G)
tensorial_product	
arc	
goniometric_functions	
sine	(Ctrl+T, S)
cosine	(Ctrl+T, C)
tangent	(Ctrl+T, T)
cotangent	
cosecant	
secant	
arccosine	
arcsine	
arccotangent	
arctangent	
arcsec	
arccosec	
sinh	
arcsinh	
cosh	
arcosh	
tanh	
arctanh	
sech	
arcsech	
cosech	
arcosech	
coth	
arcoth	
calculus	
function_composition	
limit	(Ctrl+M, T)
derivative	

nth_derivative	
partial_derivative	
definite_integral	(Ctrl+M, I)
indefinite_integral	
line_integral	
differential	
nabla	
closed_line_integral	
nth_partial_derivative	
liminf	
limsup	
laplacian	
symbols	
naturals	
integers	
rationals	
reals	
complexes	
alef	
infinity	
integral_symbol	
euro	
dollar	
arrows	
implies	
logically_equivalent	
left_right_arrow	
down_arrow	
left_arrow	
right_arrow	
up_arrow	
logarithmic_functions	
logarithm	(Ctrl+M, Uppercase+L)
natural_logarithm	(Ctrl+M, L)
logarithm_base_a	
antilogarithm	
statistics	
principal	
string	
combination	

amount	
permutation	
random_distribution	
mean_value	
mean_value_operator	
standard_deviation	
chi_squared_distribution	
uniform_distribution	
standard_normal_distribution	
normal_distribution	
fisher_distribution	
gamma_distribution	
mode	
median	
harmonic_mean	
geometric_mean	
central_moment	
asymmetry_coefficient	
kurtosis_coefficient	
mean_deviation	
median_deviation	
poisson_distribution	
variance	
covariance	
bernoulli_distribution	
binomial_distribution	
student_distribution	
interest	
discount	
mortgage_rate	
discounting_back_factor	
accumulation_factor	
annual_interest_rate	
annual_discount_rate	
instant_intensity_of_interest	
present_value_in_the_future_in_perpetuity	
present_value_in_the_future	