HAL
open science

Edix: A Software for Editing Algebraic Expressions.

Denis Bouhineau, Jean-François Nicaud, X. Pavard

To cite this version:

Denis Bouhineau, Jean-François Nicaud, X. Pavard. Edix: A Software for Editing Algebraic Expressions.. Proceedings of ACTMT2001, the Sixth Asian Technology Conference in Mathematics. The University of Melbourne, 2001, Australia. 3 p. hal-00962022

HAL Id: hal-00962022

https://hal.science/hal-00962022v1

Submitted on 25 Mar 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Edix : A Software For Editing Algebraic Expressions

Denis Bouhineau
Bouhineau@irin.univ-nantes.fr
IRIN
University of Nantes
2, rue de la Houssinière
France

Jean-François Nicaud
Nicaud@irin.univ-nantes.fr
IRIN
University of Nantes
2, rue de la Houssinière
France

Xavier Pavard
Pavard@irin.univ-nantes.fr
IRIN
University of Nantes
2, rue de la Houssinière
France

Abstract

The paper presents a computer software, called Edix, devoted to the edition of algebraic expressions in their usual 2D representation. At present, many systems display fine algebraic expressions, but the edition of such expressions is weak. Systems like Word and FrameMaker place sub-expressions in too many boxes so that many editing actions are not simple, while usual CAS (computer algebra systems) just use a 1D representation for the edition. Furthermore, Edix allows the user to represent a reasoning as a sequence or a tree, and evaluates the equivalence indicating at each input whether the current step is equivalent to the previous one.

The edition facilities of Edix are very ergonomic and algebraically oriented. This has been obtained by using an external representation at the interface where parts of expressions sometimes are invisible boxes, sometimes are text, and an internal representation as a tree. A correspondence between these representations has been built, including the insertion point that can be placed with the mouse and moved with arrows. A selection with the mouse of well-formed sub-expressions has been implemented including the selection of several parts. For example, it is possible to select x and 2y in x+3+2y-z. It is possible to apply some operators to a selected sub-expression, for example the go-to-denominator operator. In the previous example, it leads to ?/(x+2y)+3-z (where the question mark stands for something to complete; of course it is in the 2D representation and without parentheses).

Usual edition actions have been implemented on selected sub-expressions: copy, cut, paste, and drag&drop. We have defined an algebraic-paste as a paste with an operator. We have limited the operators to the commutative and associative operators of the right type (+ and * for the algebraic type, 'and' and 'or' for the Boolean type). When a paste occurs, it is performed with one operator. The user may switch to the other by hitting the ctrl-A key. For example, when x+3 is pasted on the right of y in y+z, the result is y+x+3+z, hitting the ctrl-A key provides y(x+3)+z. We have also implemented a text-paste with the Alt key, e.g., x+3 text-pasted on the right of 4 in 4y provides 4x+3y. Drag&drop works like paste for the drop. The paper will present the main principles of the edition actions.

Expressions that are unachieved (with question marks), not well-formed (e.g., with unbalanced parentheses), or undefined (e.g., with 0 in a denominator) may be produced; the sub-expressions at the origin of the problem are displayed in blue or red to indicate the problem to the user.

A reasoning is represented at the interface with boxes linked with arrows, each box containing an expression, the link between two boxes meaning that the expressions are equivalent. A click on a special part of a box containing an expression E allows the user to

produce a copy of this box if E is well-formed and, when E is the successor of U, if E is equivalent to U. If E is not equivalent to U, the link between U and E is destroyed, E becoming the starting point of a new reasoning. When an expression E which is the successor of U is modified, at each elementary modification, the information 'E is not well-formed' is indicated with a bleu cross on the link between U and E, and the information 'E is not equivalent to U' is indicated with a red cross. So, the user is always informed of the status of the edited expressions.

The semantics we use for the equivalence is the semantics of functions. At present, the domain used is the set R of real numbers (in the future, we will allow the user to choose between the real and the complex numbers). Two algebraic expressions A and B with variables x1, x2 … xn. are equivalent if and only if the associated functions from R^n to R are identical. Two logical expressions are equivalent if and only if the associated functions from R^n to {false, true} are identical. At present, we have implemented the calculation of the equivalence for polynomials with n variables, equations with one variable and degree less than 4 and systems of linear equations with n variables and p equations. We will extend a little these calculations and connect Edix to a CAS when more complex equivalences will have to be computed.

We consider several contexts for using the Edix software: (1) as a microworld for education, with parameters to adapt the system to the level of the student (2) as a general micoworld for people working on algebraic expressions, (3) as a component plugged inside systems (e.g., word processing, CAS) for the edition of algebraic expressions,

If the paper is accepted, the presentation will include a demonstration.