Operating Computer Algebra Systems by Handprinted Input

Reiner Marzinkewitsch

Fachbereich 14
Universität des Saarlandes
D-6600 Saarbrücken
email: marzin@cs.uni-sb.de

Abstract

A prototype of a workstation is presented for calculation with mathematical formulas by hand *and* support by computer algebra systems.
Keywords: Character recognition, neural network, computer algebra system, context free grammar, parsing of two dimensional structures

1 Introduction

Nearly twenty years have passed since the first computer algebra systems (CAS) came up in the beginning of the seventies. Since then CAS have gained a lot of computational power. In contrast to this fact CAS have not experienced the deserved widespread use by potential users.

The main reason for this discrepancy is the unnatural operation of CAS by artificial linearized notations, which tend to give little comprehensive survey of the problem under work. Calculation with pencil and paper not only offers many efficient techniques but also appeals to the user's ease. Especially occasional users need a familiar i.e. paperlike interface to CAS.

In this paper an integrated system will be presented, which offers the demanded facilities: Calculating by hand in a traditional, 'two dimensional' fashion with the computational support of a CAS.

Attempts to design a natural interface to CAS are as old as CAS themselves.

The 'Sophisticated Scratchpad' of [1] is the best known attempt to solve the problem. However this system was never really used because its operation lacked flexebility. Therefore the need for natural interfaces to CAS is unchanged until today.

2 General Features

Our system is operated by an I/O-device that allows for a natural operation of the system. We chose a combination of a liquid crystal display mounted on top of a digitizing tablett. As symbols are inputted by the user, they are displayed on the LCD. Thus the user gets nearly the impression of writing on a real piece of paper.

In addition to the utilization of a natural I/O-device, the user needs all 'modes of operation' which are possible with paper and pencil:

- writing handprinted symbols of arbitrary size at any position of the I/O-device
- inputting the symbols of a formula in any order
- insertion and correction by erasing or overwriting symbols
- incremental completion of formulas, reinputting intermediate results
- user interaction for the rearrangement of formulas (cut&paste, makros)
- results of former operations can be looked up in a journal where they are stored

All activities – esp. recognition of characters and formulas – should take place in real time. On the one hand this is a point of ergonomy on the other hand interpretation of characters and formulas is bound to errors.

Therefore the user should be enabled to control the system as it is interpreting his input. Real time execution was a major design goal during the development of the system.

3 Software Components

The software is structured in five main modules which are deviced to solve the subtasks of the complete problem: Pattern recognition, formula analysis and translation, management of user interaction, calculation of results, output formatting.

As the user writes down a formula, the position of the pen is monitored by the digitizing tablett. A stream of coordinates is sent to the *pattern recognition* module. There the coordinate stream is segmented in disjoint patterns which are then normalized in size. The actual pattern recognition is done by a neural network. A three-layered Backpropagation Network [5] – combined with a linear, space variant optimal filter – classifies the patterns into one of (at the moment) 52 character classes. Classification with an accuracy of up to 98% is achieved in real time (0.1 - 0.2 *sec/char*). Each recognized character is immediately displayed on the I/O-device. A latency of two segments is due to the segmentation process.

At the stage of *formula analysis*, the set of characters which is produced by the pattern recognition process, is parsed. Thus the meaning of the 'two dimensional' formula inputted by the user is obtained. Parsing is done by a graph reduction process. The characters are incorporated in a graph, which represents their spatial relations. Guided by a context free grammar for two dimensional constructs the graph is reduced in successive steps. The right hand sides of the context free productions are localized in the graph and reduced to a single vertice which corresponds to a nonterminal symbol. Ambiguities are resolved with help of methods of compiler construction: Precedences, attributes etc. Moreover the temporal information of the input process is taken into consideration.

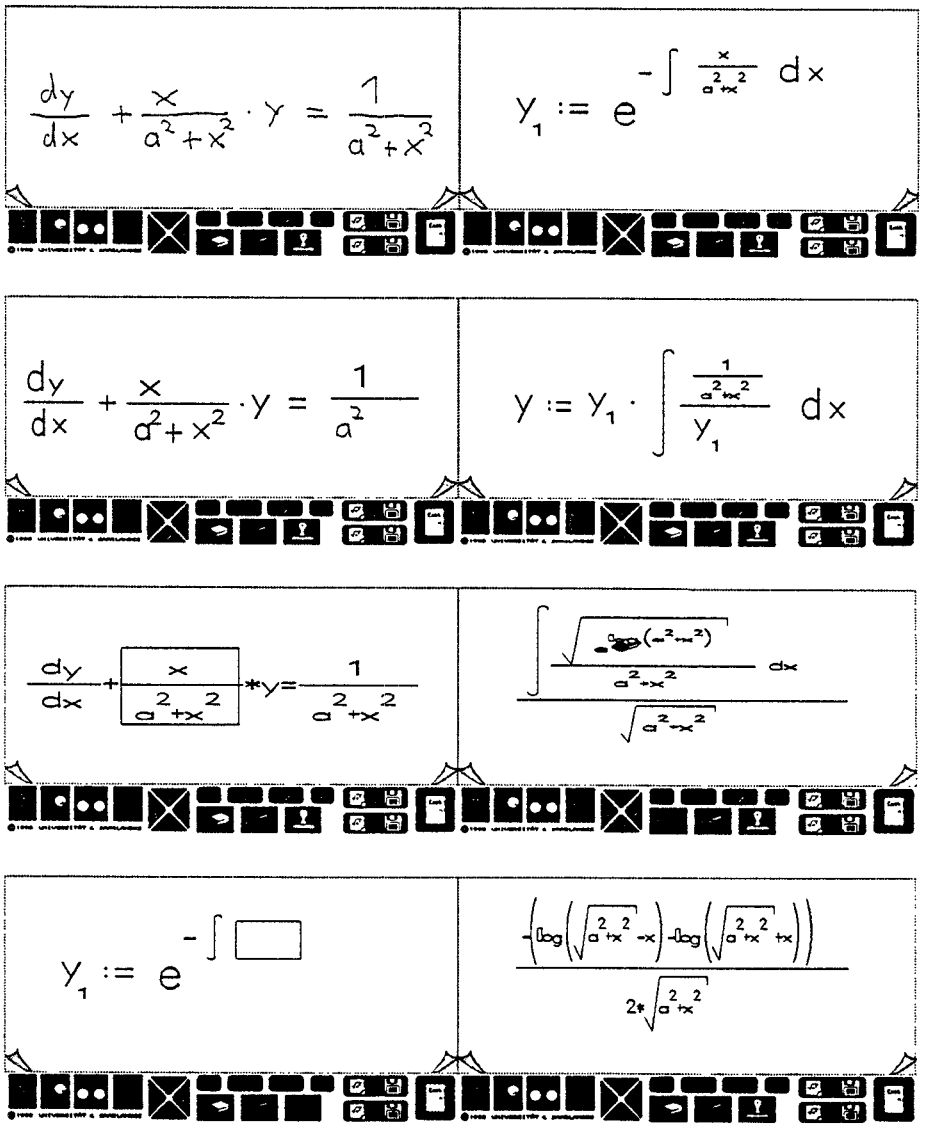
When the formula is parsed it is translated into a format that is understood by a CAS. At the moment two CAS are supported by the system (Reduce [3] and Maple [2]) but the adaption to other CAS is not very difficult. The input into the CAS as well as the result of the CAS's computation is translated into a two dimensional notation by the *output formatter* and then displayed on the I/O-device. Thus the user is not bothered with involved artificial notations any more.

The cooperation of all modules is controlled by the *management module*, especially when several formulas are developped simultaneously. This module implements all services which are advertized to the user and records all user actions in a journal. In order to ensure a consistent system behaviour, the management module was implemented in accordance with the principle of direct manipulation.

All software modules are built as separate processes which are connected by means of process communication. The software is executed by a Sun 4 SLC UNIX-workstation. The I/O-device mentioned above is not connected to the workstation directly but driven by an extra PC. This is due to hardware problems which will be overcome by the design of an interface for direct output from the workstation to the I/O-device.

4 Example

The following example demonstrates the operation of the system. A differential equation is solved 'manually' by variation of the constant. The pictures shown are hardcopies from the display of the I/O-device. Not all steps of the calculation are shown. (The pictures are ordered in a columnwise manner.) The first picture shows the differential equation, which shall be integrated. The pattern recognition was turned off to take this picture. Thus the whole formula is shown as it is written by the user. In normal operation of the system (pattern recognition turned on) the patterns are replaced by the recognized characters with a latency of 2 segments (picture 2). A part is cut out of the formula (picture 3) and pasted into the integral for the solution of the homogeneous equation (pictures 4, 5). The complete solution is found by 'variation of the constant' (picture 6). Reduce 3.3 does not automatically 'reduce' the identity $exp(log(x)) = x$ in this context (though it was tought this identity explicitly). We can apply this identity by hand by rubbing out the respective symbols (picture 7). Picture 8 shows the wanted solution of the problem.



References

[1] R. Anderson,
An Online Mathematics System Using Twodimensional handprinted Notation
Rand Corp., Santa Monica 1970

[2] B. Char et al.
First Leaves: A Tutorial Introduction to Maple
University of Waterloo, Canada 1988

[3] A. C. Hearn
Reduce User's Manual
Rand Corp., Santa Monica 1987

[4] R. Marzinkewitsch,
Ein Arbeitsplatz zum computerunterstützten, handschriftlichen Rechnen mit mathematischen Formeln
Dissertation,
Universität des Saarlandes 1990

[5] D. Rumelhart, J. Mc. Clelland,
Parallel Distributed Processing
MIT Press 1988