RLFI
A REDUCE LaTeX Formula Interface
Version 1.2.1

Richard Liska, Ladislav Drska
Computational Physics Group
Faculty of Nuclear Sciences and Physical Engineering
Czech Technical University in Prague
Brehova 7, 115 19 Prague 1, Czech Republic
E-mail: liska@siduri.fjfi.cvut.cz

May 23, 1995

High quality typesetting of mathematical formulas is a quite tedious task. One of the most sophisticated typesetting programs for mathematical text TeX [?], together with its widely used macro package LaTeX [?], has a strange syntax of mathematical formulas, especially of the complicated type. This is the main reason which lead us to designing the formula interface between the computer algebra system REDUCE and the document preparation system LaTeX. The other reason is that all available syntaxes of the REDUCE formula output are line oriented and thus not suitable for typesetting in mathematical text. The idea of interfacing a computer algebra system to a typesetting program has already been used, eg. in [?] presenting the TeX output of the MACSYMA computer algebra system.

The formula interface presented here adds to REDUCE the new syntax of formula output, namely LaTeX syntax, and can also be named REDUCE - LaTeX translator. Text generated by REDUCE in this syntax can be directly used in LaTeX source documents. Various mathematical constructions are supported by the interface including subscripts, superscripts, font changing, Greek letters, divide-bars, integral and sum signs, derivatives etc.

The interface can be used in two ways:

- for typesetting of results of REDUCE algebraic calculations.
- for typesetting of users formulas.

The latter can even be used by users unfamiliar with the REDUCE system, because the REDUCE input syntax of formulas is almost the same as the syntax of the majority of programming languages. We aimed at speeding up the process of formula typesetting, because we are convinced, that the writing of correct complicated formulas in the REDUCE syntax is a much more simpler task than writing them in the LaTeX syntax full of keywords and special characters `\, {, ^` etc. It is clear, that not every formula produced by the interface is typeset in the best format from an aesthetic point of view. When a user is not satisfied with the result, he can add some LaTeX commands to the REDUCE output - LaTeX input.

The interface is connected to REDUCE by three new switches and several statements. To activate the LaTeX output mode the switch `latex` must be set `on`. this switch, similar to the switch `fort` producing FORTRAN output, being `on` causes all outputs to be written in the LaTeX syntax of formulas. The switch `VERBATIM` is used for input printing control. If it is `on` input to REDUCE system is typeset in LaTeX verbatim environment after the line containing the string `REDUCE Input:`.

The switch `lasimp` controls the algebraic evaluation of input formulas. If it is `on` every formula is evaluated, simplified and written in the form given by ordinary REDUCE statements and switches such as `factor`, `order`, `rat` etc. In the case when the `lasimp` switch is `off` evaluation, simplification or reordering of formulas is not performed and REDUCE acts only as a formula parser and the form of the formula output is exactly the same as that of the input, the only difference remains in the syntax. The mode `off lasimp` is designed especially for typesetting of formulas for which the user needs preservation of their structure. This switch has no meaning if the switch `Latex` is `off` and thus is working only for LaTeX output.

For every identifier used in the typeset REDUCE formula the following properties can be defined by the statement `defid`:

- its printing symbol (Greek letters can be used).
- the font in which the symbol will be typeset.
- accent which will be typeset above the symbol.

Symbols with indexes are treated in REDUCE as operators. Each index corresponds to an argument of the operator. The meaning of operator arguments (where one wants to typeset them) is declared by the statement `defindex`. This statement causes the arguments to be typeset as subscripts or superscripts (on left or right-hand side of the operator) or as arguments of the operator.

The statement `mathstyle` defines the style of formula typesetting. The variable `laline!*` defines the length of output lines.

The fractions with horizontal divide bars are typeset by using the new REDUCE infix operator \. This operator is not algebraically simplified. During typesetting of powers the checking on the form of the power base and exponent is performed to determine the form of the typeset expression (eg. sqrt symbol, using parentheses).

Some special forms can be typeset by using REDUCE prefix operators. These are as follows:

- `int` - integral of an expression.
- `dint` - definite integral of an expression.
- `df` - derivative of an expression.
- `pdf` - partial derivative of an expression.
- `sum` - sum of expressions.
- `product` - product of expressions.
- `sqrt` - square root of expression.

There are still some problems unsolved in the present version of the interface as follows:

- breaking the formulas which do not fit on one line.
- automatic decision where to use divide bars in fractions.
- distinction of two- or more-character identifiers from the product of one-character symbols.
- typesetting of matrices.

Description of files

rlfi.red - REDUCE source file for this interface.

rlfi.tex - this document.

rlfi.bib - bibliography file for this document.

rlfi.tst - test file for this interface.

rlfi.log - LATEX output of the test session, can be directly used as LATEX input file.

<div align="center">**Remark**</div>

After finishing presented interface, we have found another work [**?**], which solves the same problem. The RLFI package has been described in [**?**] too.

References

1 APPENDIX: Summary and syntax

Warning

The RLFI package can be used only on systems supporting lower case letters with `off raise` statement. The package distinquishes the upper and lower case letters, so be carefull in typing them. In REDUCE3.6 the REDUCEcommands have to be typed in lower-case while the switch `latex` is `on`, in previous versions the commands had to be typed in upper-case.

Switches

`latex` - If `on` output is in LATEX format. It turns `off` the `raise` switch if it is set `on` and `on` the `raise` switch if it is set `off`. By default is `off`.

`lasimp` - If `on` formulas are evaluated (simplified), REDUCE works as usually. If `off` no evaluation is performed and the structure of formulas is preserved. By default is `on`.

`verbatim` - If `on` the REDUCE input, while `latex` switch being `on`, is printed in LATEX verbatim environment. The acutal REDUCE input is printed after the line containing the string `"REDUCE Input:"`. It turns `on` resp. `off` the `echo` switch when turned `on` resp. `off`. by default is `off`.

Operators

infix - \

prefix - `int`,`dint`,`df`,`pdf`,`sum`,`product`,`sqrt` and all REDUCE prefix operators defined in the REDUCE kernel and the SOLVE module.

```
<alg. expression> \ <alg. expression>
int(<function>,<variable>)
dint(<from>,<to>,<function>,<variable>)
df(<function>,<variables>)
<variables> ::= <o-variable>|<o-variable>,<variables>
<o-variable> ::= <variable>|<variable>,<order>
<variable> ::= <kernel>
<order> ::= <integer>
<function> ::= <alg. expression>
<from> ::= <alg. expression>
<to> ::= <alg. expression>
pdf(<function>,<variables>)
sum(<from>,<to>,<function>)
product(<from>,<to>,<function>)
sqrt(<alg. expression>)
```

`<alg.  expression>` is any algebraic expression. Where appropriate, it can include also relational operators (e.g. argument `<from>` of `sum` or `product` operators is usually equation). `<kernel>` is identifier or prefix operator with arguments as described in [**?**]. Interface supports typesetting lists of algebraic expressions.

Statements

```
mathstyle <m-style>;
<m-style> ::= math | displaymath | equation
defid <identifier>,<d-equations>;
<d-equations> ::= <d-equation> | <d-equation>,<d-equations>
<d-equation> ::= <d-print symbol> | <d-font>|<d-accent>
<d-print symbol> ::= name = <print symbol>
<d-font> ::= font = <font>
<d-accent> ::= accent = <accent>
<print symbol> ::= <character> | <special symbol>
<special symbol> ::= alpha|beta|gamma|delta|epsilon|
    varepsilon|zeta|eta|theta|vartheta|iota|kappa|lambda|
    mu|nu|xi|pi|varpi|rho|varrho|sigma|varsigma|tau|
    upsilon|phi|varphi|chi|psi|omega|Gamma|Delta|Theta|
    Lambda|Xi|Pi|Sigma|Upsilon|Phi|Psi|Omega|infty|hbar
```

```
<font> ::= bold|roman
<accent> ::=hat|check|breve|acute|grave|tilde|bar|vec|
    dot|ddot
```

For special symbols and accents see [**?**], p. 43, 45, 51.

```
defindex <d-operators>;
<d-operators> ::= <d-operator> | <d-operator>,<d-operators>
<d-operator> ::= <prefix operator>(<descriptions>)
<prefix operator> ::= <identifier>
<descriptions> ::= <description> | <description>,
    <descriptions>
<description> ::= arg | up | down | leftup | leftdown
```

The meaning of the statements is briefly described in the preceding text.