

	GNU T _E X _{MACS}
Installed version	1.99.4
Supported systems	Most GNU/LINUX systems
Copyright	© 1998–2002 by Joris van der Hoeven
License	GNU General Public License
Web sites	http://www.texmacs.org http://www.gnu.org/software/texmacs)
(descriptive-table	
Contact	contact@texmacs.org
Regular mail	Dr. Joris van der Hoeven Dépt. de Mathématiques (Bât. 425) Université Paris-Sud 91405 Orsay Cedex France

Table 1. Summary of the principal information about GNU T_EX_{MACS}.

Checking for foreign routines

```
AXIOM="/usr/local/lib/fricas/target/x86_64-linux-gnu"
spad-lib="/usr/local/lib/fricas/target/x86_64-linux-gnu/lib/libspad.so"
foreign routines found
openServer result -2
```

```

          FriCAS Computer Algebra System
          Version: FriCAS 1.3.3
          Timestamp: Don Apr 12 19:26:08 CEST 2018
-----
```

```
Issue )copyright to view copyright notices.
Issue )summary for a summary of useful system commands.
Issue )quit to leave FriCAS and return to shell.
-----
```

```
Function declaration sixel : TexFormat -> Void has been added to
workspace.
```

Type: Void

```
Value = #<INTERPRETED-FUNCTION NIL {100258211B}>
```

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

$x + 1$

Type: Polynomial(Integer)

```
(4) -> D(sin(x)*x,x,6)
```

$-x \sin(x) + 6 \cos(x)$

Type: Expression(Integer)

```
(5) -> integrate(x*sin(x),x)
```

$\sin(x) - x \cos(x)$

Type: Union(Expression(Integer),...)

```
(6) -> )d op zerosOf
```

There are 5 exposed functions called zerosOf :

- [1] (SparseUnivariatePolynomial(D),Symbol) -> List(D) from D if D has ACF
- [2] SparseUnivariatePolynomial(D) -> List(D) from D if D has ACF
- [3] Polynomial(D) -> List(D) from D if D has ACF
- [4] (D,Symbol) -> List(D) from D if D3 has Join(COMPAR,INTDOM) and D has ACFS(D3)
- [5] D -> List(D) from D if D2 has Join(COMPAR,INTDOM) and D has ACFS(D2)

```
(6) -> factor(1234567)
```

1279721

Type: Factored(Integer)

```
(7) ->
```

```
kfp@nilqed:~$ ls /usr/local/share/TeXmacs/plugins/  
asymptote  coq      giac      lush      matlab  octave  r        texgraph  
axiom      dratex   gnuplot   macaulay2  maxima  pari    reduce  xypic  
caas      eukleides  graphviz  maple     mmi     pure    sage    yacas  
cadabra   feynmf    gtybalt   mathmagix  mupad   python  scilab  
cocoa5    fricas    lisp      mathematica mycas   qcl     shell
```

```
kfp@nilqed:~$ ls ~/.TeXmacs/plugins/fricas/progs/  
fricas-input.bck  fricas-kbd.scm  init-fricas.scm  
fricas-input.scm  fricas-menus.scm  init-fricas.scm~
```

```
kfp@nilqed:~/.TeXmacs/plugins/fricas/progs$ cat init-fricas.scm
```

```
;;;;;  
;;  
;; MODULE      : init-fricas.scm  
;; DESCRIPTION : Initialize fricas plugin  
;; COPYRIGHT   : (C) 1999, 2012 Joris van der Hoeven and Andrey Grozin  
;;  
;; This software falls under the GNU general public license and comes WITHOUT  
;; ANY WARRANTY WHATSOEVER. See the file $TEXMACS_PATH/LICENSE for details.  
;; If you don't have this file, write to the Free Software Foundation, Inc.,  
;; 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.  
;;  
;;;;;
```

```
(define (fricas-initialize)  
  (import-from (utils plugins plugin-convert))  
  (import-from (utils plugins plugin-cmd))  
  (import-from (dynamic session-menu))  
  (import-from (fricas-kbd))  
  (import-from (fricas-menus)))  
  ;(lazy-input-converter (fricas-input) fricas))
```

```

;;;+kp
;;; tmm(s:String):String == s ;
;;; added to .axiom.input
;;;

(define (fricas-verbatim-serialize lan t)
  (import-from (utils plugins plugin-cmd))
  (with u (pre-serialize lan t)
    (string-append "tmm(\"" (escape-verbatim (texmacs->code u)) "\\")\n")))

(define (fricas-serialize lan t)
  (import-from (utils plugins plugin-cmd))
  (import-from (utils plugins plugin-convert))
  (with s (if (in-math?)
    (fricas-verbatim-serialize lan t)
    (verbatim-serialize lan t))
    s))

;;;-kp

(plugin-configure fricas
  (:require #t)
  (:initialize (fricas-initialize))
  (:launch "C:/NonPortable/TeXmacs/fricasTM.cmd")
  (:launch "fricas -texmacs")
  (:launch "C:/FriCAS/fricas-texmacs.cmd")
  (:session "Fricas")
  (:serializer ,fricas-serialize)
  (:commander ,fricas-command)
  (:tab-completion #t)
  (:scripts "Fricas"))

(define (fricas-input-rsub args)
  (import-from (utils plugins plugin-convert))
  (display " @subB ")
  (plugin-input (car args))
  (display " @subE "))

(define (fricas-input-rsup args)
  (import-from (utils plugins plugin-convert))
  (display " @supB ")
  (plugin-input (car args))
  (display " @supE "))

(define (fricas-input-label args)
  (import-from (utils plugins plugin-convert))
  (display " @labelB ")
  (plugin-input (car args))
  (display " @labelE "))

(define (fricas-input-reference args)
  (import-from (utils plugins plugin-convert))
  (display " @referenceB ")
  (plugin-input (car args)))

```

```

      (display " @referenceE ")

(define (fricas-input-proof args)
  (import-from (utils plugins plugin-convert))
  (display " @proofB ")
  (plugin-input (car args))
  (display " @proofE "))

(define (fricas-input-definition args)
  (import-from (utils plugins plugin-convert))
  (display " @definitionB ")
  (plugin-input (car args))
  (display " @definitionE "))

(define (fricas-input-theorem args)
  (import-from (utils plugins plugin-convert))
  (display " @theoremB ")
  (plugin-input (car args))
  (display " @theoremE "))

(define (fricas-input-lemma args)
  (import-from (utils plugins plugin-convert))
  (display " @lemmaB ")
  (plugin-input (car args))
  (display " @lemmaE "))

(define (fricas-input-proposition args)
  (import-from (utils plugins plugin-convert))
  (display " @propositionB ")
  (plugin-input (car args))
  (display " @propositionE "))

;(plugin-input-converters fricas
; (rsub fricas-input-rsub)
; (rsup fricas-input-rsup)
; (label fricas-input-label)
; (reference fricas-input-reference)
; (definition fricas-input-definition)
; (theorem fricas-input-theorem)
; (lemma fricas-input-lemma)
; (proposition fricas-input-proposition)
; (proof fricas-input-proof))
; ("(" " @lpar ")
; (")" " @rpar ")
; ("[" " @lbrack ")
; ("]" " @rbrack ")
; ("{" " @lbrace ")
; ("}" " @rbrace ")
; (">" " @gt ")
; ("<" " @lt ")
; ("=" " @eq ")
; ("*" " @mult ")

```

```

; ("+" " @plus ")
; ("- " " @minus ")
; ("/" " @slash ")
; ("," " @comma ")
; (". " " @dot ")
; (":" " @colon ")
; (":=" " @assign ")
; ("" " @langle ")
; ("" " @rangle ")
; ("" " @ldots ")
; ("" " @cdots ")
; ("" " @in ")
; ("" " @cap ")
; ("" " @cup ")
; ("" " @subset ")
; ("" " @mapsto ")
; ("" " @leftarrow ")
; ("" " @rightarrow ")
; ("" " @vee ")
; ("" " @neg ")
; ("" " @equiv ")
; ("" " @circ ")
; ("" " @star ")
; ("" " @neq ")
; ("" " @leq ")
; ("" " @geq ")
; ("" " @leq ")
; ("" " @geq ")
; ("" " @times ")
; ("" " @wedge ")
; ("" " @forall ")
; ("" " @exists ")
; ("" " @partial ")
; ("" " @emptyset ")
;
; ("" "@AA")
; ("" "@BB")
; ("" "@CC")
; ("" "@DD")
; ("" "@EE")
; ("" "@FF")
; ("" "@GG")
; ("" "@HH")
; ("" "@II")
; ("" "@JJ")
; ("" "@KK")
; ("" "@LL")
; ("" "@MM")
; ("" "@NN")
; ("" "@OO")
; ("" "@PP")
; ("" "@QQ")
; ("" "@RR")
; ("" "@SS")
; ("" "@TT")
; ("" "@UU")

```

```

; (<bbb-V> " @VV")
; (<bbb-W> " @WW")
; (<bbb-X> " @XX")
; (<bbb-Y> " @YY")
; (<bbb-Z> " @ZZ")
;
;

```

(plugin-input-converters fricas

```

(<infinity> " %infinity")
(<emptyset> " []")
(<mathd> " 1,")
(<mathi> " %i")
(<mathe> " %e")
(<pi> " %pi")
(<ldots> " .. ")
(<cdots> " .. "))
; (" " @lpar ")
; (")" @rpar ")
; ("[" @lbrack ")
; ("]" @rbrack ")
; ("{" @lbrace ")
; ("}" @rbrace ")
; (">" @gt ")
; ("<" @lt ")
; ("=" @eq ")
; ("*" @mult ")
; ("+" @plus ")
; ("- " @minus ")
; ("/" @slash ")
; ("," @comma ")
; (". " @dot ")
; (":" @colon ")
; (":=" @assign ")
; (<langl>" @langl ")
; (<rangl>" @rangl ")
; (<ldots>" @ldots ")
; (<cdots>" @cdots ")
; (<in>" @in ")
; (<cap>" @cap ")
; (<cup>" @cup ")
; (<subset>" @subset ")
; (<mapsto>" @mapsto ")
; (<leftarrow>" @leftarrow ")
; (<rightarrow>" @rightarrow ")
; (<vee>" @vee ")
; (<neg>" @neg ")
; (<equiv>" @equiv ")
; (<circ>" @circ ")
; (<star>" @star ")
; (<neq>" @neq ")
; (<leq>" @leq ")
; (<geq>" @geq ")
; (<leqslant>" @leq ")
; (<geqslant>" @geq ")
; (<times>" @times ")

```

```

; (<wedge>"      " @wedge ")
; (<forall>"     " @forall ")
; (<exists>"     " @exists ")
; (<partial>"    " @partial ")
; (<emptyset>"  " @emptyset ")
;
; (<bbb-A>" @AA")
; (<bbb-B>" @BB")
; (<bbb-C>" @CC")
; (<bbb-D>" @DD")
; (<bbb-E>" @EE")
; (<bbb-F>" @FF")
; (<bbb-G>" @GG")
; (<bbb-H>" @HH")
; (<bbb-I>" @II")
; (<bbb-J>" @JJ")
; (<bbb-K>" @KK")
; (<bbb-L>" @LL")
; (<bbb-M>" @MM")
; (<bbb-N>" @NN")
; (<bbb-O>" @OO")
; (<bbb-P>" @PP")
; (<bbb-Q>" @QQ")
; (<bbb-R>" @RR")
; (<bbb-S>" @SS")
; (<bbb-T>" @TT")
; (<bbb-U>" @UU")
; (<bbb-V>" @VV")
; (<bbb-W>" @WW")
; (<bbb-X>" @XX")
; (<bbb-Y>" @YY")
; (<bbb-Z>" " @ZZ ")

```

kfp@nilqed:~/TeXmacs/plugins/fricas/progs\$ cat fricas-menus.scm

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;; MODULE      : fricas-menus.scm
;; DESCRIPTION : Fricas menus
;; COPYRIGHT   : (C) 1999, 2012 Joris van der Hoeven and Andrey Grozin
;;
;; This software falls under the GNU general public license version 3 or
later.
;; It comes WITHOUT ANY WARRANTY WHATSOEVER. For details, see the file LICENSE
;; in the root directory or <http://www.gnu.org/licenses/gpl-3.0.html>.
;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(texmacs-module (fricas-menus)
  (:use (texmacs texmacs tm-files)
        ;(doc help-funcs)
        (dynamic scripts-edit)))

```

```

(define fricas-apply script-apply)

(menu-bind fricas-menu
  (if (not-in-session?)
      (link scripts-eval-menu)
      ---)
  (-> "Calculus"
      ("Differentiate" (fricas-apply "D" 2))
      ("Integrate"     (fricas-apply "integrate" 2))
      ("Limit"        (fricas-apply "limit" 3))
      ---
      ("Sum"          (fricas-apply "sum" 2))
      ("Product"      (fricas-apply "product" 2))
      ---
      ("Solve an equation" (fricas-apply "solve" 2)))
  (if (not-in-session?)
      ---
      (link scripts-eval-toggle-menu)))

;;;;;
;;;
;;;;;

(menu-extend texmacs-extra-menu
  (=> "Fricas"

      ("Last result(s)... % or %%(+-n)" (insert "%" ))
      ("Assignment ... x:=1"           (insert ":@" ))
      ("Type declaration ... y:Integer" (insert ":" ))
      ("Type conversion ... y::Float"   (insert "::" ))
      ("Line continuation ... abc_"     (insert "_" ))
      ("Comments ... -- double dash"    (insert "--" ))
      ("Function ... f(x) == x^2"        (insert "==" ))
      ("Macro ... macro df == D"         (insert "macro ? ==" )))

      ---
  (-> "Predefined macros"
      ("The square root of -1 ... %i"      (insert "%i" ))
      ("Base of natural log ... %e"        (insert "%e" ))
      ("Number pi ... %pi"                 (insert "%pi" ))
      ("Infinity ... %infinity"           (insert "%infinity" ))
      ("Plus infinity ... %plusInfinity"   (insert "%plusInfinity" ))
      ("Minus infinity ... %minusInfinity" (insert "%minusInfinity" )))

      (-> "Numbers"
          ("Factor an integer n"           (fricas-apply "factor" 1))
          ("Number n in base x"            (fricas-apply "radix" 2))
          ("Integer n to Roman"            (fricas-apply "roman" 1))
          ("Represent x as T ... x@T"      (insert "?@?" ))
          ("Precision #digits"             (fricas-apply "digits" 1))
          ("Complex number ... 1+%i*2"     (insert "?+%i*?" ))
          ("Complex symbolic ... a+%i*b"   (fricas-apply "complex" 2))
          ("Complex conjugate"             (fricas-apply "conjugate" 1))
          ("Rational -> Decimal"           (fricas-apply "decimal" 1))

```

```

("Continued fraction"      (fricas-apply "continuedFraction" 1))
("Partial fraction"       (fricas-apply "partialFraction" 2))
("Padic fraction"        (fricas-apply "padicFraction" 1))
("Root of polynomial"    (fricas-apply "rootOf" 2))

(-> "Calculus"

  ("Differentiate [D]"    (fricas-apply "D" 2))
  ("Integrate"           (fricas-apply "integrate" 2))
  ("Limit"               (fricas-apply "limit" 3))
  ---
  ("Sum"                 (fricas-apply "sum" 2))
  ("Product"             (fricas-apply "product" 2))
  ---
  ("Solve an equation"   (fricas-apply "solve" 2)))

(-> "Algebra"
  ("Matrices" () )
  ("Tensors" () ))
)
)

```

kfp@nilqed:~/TeXmacs/plugins/fricas/progs\$ cat fricas-kbd.scm

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;; MODULE      : fricas-kbd.scm
;; DESCRIPTION : Keyboard shortcuts for Fricas
;; COPYRIGHT   : (C) 2005 Joris van der Hoeven
;;
;; This software falls under the GNU general public license version 3 or
later.
;; It comes WITHOUT ANY WARRANTY WHATSOEVER. For details, see the file LICENSE
;; in the root directory or <http://www.gnu.org/licenses/gpl-3.0.html>.
;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(texmacs-module (fricas-kbd)
  (:use (dynamic scripts-kbd)))

(texmacs-modes
  (fricas-scripts-math% #t fricas-scripts% in-math%))

(kbd-map
  (:mode fricas-scripts-math?)
  (: " = " (insert '(script-assign))))

```

kfp@nilqed:~/TeXmacs/plugins/fricas/progs\$ cat fricas-input.scm

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;; MODULE      : fricas-input.scm
;; DESCRIPTION : Fricas input converters
;; COPYRIGHT   : (C) 1999, 2012 Joris van der Hoeven and Andrey Grozin
;;
;; This software falls under the GNU general public license version 3 or
later.
;; It comes WITHOUT ANY WARRANTY WHATSOEVER. For details, see the file LICENSE
;; in the root directory or <http://www.gnu.org/licenses/gpl-3.0.html>.
;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(texmacs-module (fricas-input)
  (:use (utils plugins plugin-convert)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Specific conversion routines
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(define (fricas-input-var-row r)
  (if (null? r)
      (begin
        (display ", ")
        (plugin-input (car r))
        (fricas-input-var-row (cdr r))))))

(define (fricas-input-row r)
  (display "["))
  (plugin-input (car r))
  (fricas-input-var-row (cdr r))
  (display "]""))

(define (fricas-input-var-rows t)
  (if (null? t)
      (begin
        (display ", ")
        (fricas-input-row (car t))
        (fricas-input-var-rows (cdr t))))))

(define (fricas-input-rows t)
  (display "matrix(["))
  (fricas-input-row (car t))
  (fricas-input-var-rows (cdr t))
  (display "]""))

(define (fricas-input-descend-last args)
  (if (null? (cdr args))
      (plugin-input (car args))
      (fricas-input-descend-last (cdr args))))

(define (fricas-input-det args)
  (display "determinant(")
  (fricas-input-descend-last args)
  (display ")"))

```

```

; (define (fricas-input-big-around args)
;   (let* ((b '(big-around ,@args))
;         (op (big-name b))
;         (sub (big-subscript b))
;         (sup (big-supscript b))
;         (body (big-body b)))
;     (cond
;       ((= op "int")
;        (begin (display "integrate(") (plugin-input body)
;              (if (and sub sup)
;                  (begin (display "=") (plugin-input sub)
;                        (display "..") (plugin-input sup))))))
;       ((= op "sum")
;        (begin (display "sum(") (plugin-input body)
;              (cond
;                ((and sub sup)
;                 (begin (display ",")
;                       (plugin-input (texmacs->code (tm->tree sub)))
;                       (display "..") (plugin-input sup)))
;                (sub (begin (display ",") (plugin-input sub)))))))
;       ((= op "prod")
;        (begin (display "product(") (plugin-input body)
;              (cond
;                ((and sub sup)
;                 (begin (display ",")
;                       (plugin-input (texmacs->code (tm->tree sub)))
;                       (display "..") (plugin-input sup)))
;                (sub (begin (display ",") (plugin-input sub)))))))
;       (else (display op) (display "(") (plugin-input body)))
;     (display ")"))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Initialization
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

```

(plugin-input-converters fricas
  (rows fricas-input-rows)
  (det fricas-input-det)
  ;(big-around fricas-input-big-around)
  ("" "%infinity")
  ("" "[]")
  ("" "1,")
  ("" "%i")
  ("" "%e")
  ("" "%pi")
  ("" " .. ")
  ("" " .. "))
kfp@nilqed:~/TeXmacs/plugins/fricas/progs$

```

