

Util functions	
Function Signature	Description
show(*args) show(*args,nl=1)	Prints the items in the sequence *args followed with nl blank lines (one blank line by default). Thus show(*args,0) is equivalent to print(*args).
Type(a)	Prints the PyM internal type of a variable. It is not the same as the python function type
log(x) log(x,b)	Computes the logarithm value of x in base b. If b is not included, it computes the natural logarithm
ln(x)	Computes the natural logarithm of x
notnil(x)	Returns True if x is not aproximately zero
nil(x)	Returns True if x is approximately zero
nongeg(x)	Returns True if x is bigger or equal than zero
is_number(x)	Returns True if x is an integer, a rational number, a float, or a complex number
is_real(x)	Returns True if x is an integer, a rational number, or a float
find_zero(f,a,b,tol)	Given a function f, an interval (a,b) of the real numbers and a tolerance, it returns a point x between a and b, that $ f(x) < tol$
lift(a)	If $a = \text{push}(x,A)$, where A is a domain and x an object, lift(a) returns an object x' in the same domain as x such that $a = \text{push}(x',A)$ and which is simplest with this condition. For example, lift(push(19,Zn(11))) returns 8.
push(a,D)	If a is an object belonging to a domain A and there is a natural map $A \rightarrow D$ to a domain D, this function returns the image of a in B. Example. push(19, Zn(11)) yields 8 :: Z11.
is_zero(a)	True if a is 0, False otherwise.
clone(a)	It creates a copy of the object a
islambda(v)	Returns if v is a lambda function
index(s,H)	Given a list or vector H and an element s, it returns the position of s in H.
inclist(p)	Increments the last value of p by 1.
pad(c,r)	If $r \leq \text{len}(c)$, it returns $c[r:]$, otherwise it appends $r - \text{len}(c)$ zeros.
is_invertible(x)	It returns if x is invertible.
types_off() types_on() toggle_types()	By default, PyM does not display the domains of objects (the global variable TYPES is set to 0). The effect of function types_on() is that the domains of objects be displayed (sets TYPES to 1), and toggle_types changes between types_on and types_off.
decimals(n) nodecimals()	decimals(n) represents que rational numbers with n decimals. nodecimals() represent the rational numbers as a fraction.
PS_print_order(n)	It shows the power series until order n.
to_list(f)	If f is a polynomial, it returns the list of coefficients. If f is a vector, it returns f as a list.

Util functions	
Function Signature	Description
invert_table(T)	Given a table T composed by pair of elements $[a_i, b_i]$, it returns the same table but with the elements in the order $[b_i, a_i]$
string(C)	Given a list of characters, it returns the string of all the characters.
take(L,n)	Given a list L and an integer n, it returns $L[:n]$ if $n > 0$ and $L[n:]$ if $n < 0$
simplex(cost,R)	It resolve the linear program $\max\{\text{cost}\}$ subject to $r \geq 0$ for r in R
LP(n,d,R) LP(n,d)	Linear programing bound of an $[n,d]$ code with optional extra relations R.
merge(A,B)	Given two lists of pairs $A = [(a_0, u_0), (a_1, u_1), \dots, (a_n, u_n)]$ and $B = [(b_0, v_0), (b_1, v_1), \dots, (b_m, v_m)]$, let $A_1 = [a_0, a_1, \dots, a_n]$, $A_2 = [u_0, u_1, \dots, u_n]$, $B_1 = [b_0, b_1, \dots, b_m]$ and $B_2 = [v_0, v_1, \dots, v_m]$. Then merge(A,B) creates the list of pairs (c,w), where c runs over $\text{union}(A_1, B_1)$ and w is u_j if $c = a_j$ in $A_1 - B_1$, or v_k if $c = b_k$ in $B_1 - A_1$, or $u_j + v_k$ if $c = a_j = b_k$ in $\text{intersection}(A_1, B_1)$.