

数式処理：代入・仮定その他

Copyright @2006 by Shigeto R. Nishitani

▼ 代入、置換、仮定に関連したコマンド

subs(関係式, exp1):一時的代入

```
> exp1:=x^2-4*x*y+4;  
    subs(x=2,exp1);  
          exp1 := x^2 - 4 x y + 4  
                           8 - 8 y  
> subs({x=a+2,y=sin(x)},exp1);  
          (a + 2)^2 - 4 (a + 2) sin(x) + 4
```

(1.1.1.1)

(1.1.1.2)

assume(関係式):仮定

```
> sqrt(b^2);  
assume(a>0);  
sqrt(a^2);  
          √b²  
a~
```

(1.1.2.1)

exp1 assuming 関係式:一時的仮定

```
> exp1:=x^2-4*x+4;  
    sqrt(exp1);  
          exp1 := x^2 - 4 x + 4  
                           √(-2 + x)²  
> sqrt(exp1) assuming x>2;  
          -2 + x
```

(1.1.3.1)

(1.1.3.2)

additionally:assumeに加えての仮定

assign(exp1):solveで求めた値の確定

```
> x:=x';y:=y';  
s1:=solve({x-y+1=0,x+y-2=0},{x,y});  
assign(s1);  
          sl := {y = 3/2, x = 1/2}  
> x,y;  
          1/2, 3/2
```

(1.1.5.1)

(1.1.5.2)

about:assumeで仮定した内容の確認

```
> about(a);  
Originally a, renamed a~:  
is assumed to be: RealRange(Open(0),infinity)
```

anames('user'):ユーザが定義した変数の確認

```
> anames('user');
```

sl, y, x, a

(1.1.7.1)

▼ restart,a:='a': 値の初期化

▼ 省略操作、 その他のコマンド

||:連結作用素、 前後の変数をくっつけて新たな変数とする。

```
> a||1;  
    a||b;  
          al  
ab  
> for i from 1 to 3 do  
    a||i:=i^2;  
end do;
```

(1.2.1.1)

seq(exp1,i=0..3):for-loopの単純表記

```
> seq(k,k=4..7);  
          4, 5, 6, 7
```

(1.2.2.1)

map(exp1,i=0..3):リスト要素への関数の一括適用

```
> f:=x->exp(-x);  
map(f,[0,1,2,3]);  
          f := x → e⁻x  
[1, e⁻¹, e⁻², e⁻³]  
上記の 3 つを組み合わせると、効率的に式を扱うことができる。
```

(1.2.3.1)

```
> map(sin,[seq(theta||i,i=0..3)]);  
[sin(θ0), sin(θ1), sin(θ2), sin(θ3)]
```

(1.2.3.2)

add,mul:単純な和、 積

```
> add(x^i,i=0..3);  
          1 + x + x² + x³  
> mul(x^i,i=0..3);  
          x⁶
```

(1.2.4.1)

(1.2.4.2)

sum,product:数式に対応した和、 積

```
> add(x^i,i=0..n);  
Error, unable to execute add  
> sum(x^i,i=0..n);  
          xn+1 - 1  
-----  
      x - 1 - x - 1  
> product(x^i,i=0..n);  
          1  
x1/2 (n + 1)² - 1/2 n - 1/2
```

(1.2.5.1)

(1.2.5.2)

limit:極限

```
> limit(exp(-x),x=infinity);  
          0  
> limit(tan(x),x=Pi/2);  
          undefined  
> limit(tan(x),x=Pi/2,left);  
limit(tan(x),x=Pi/2,complex);  
          ∞  
-∞ + ∞ I
```

(1.2.6.1)

(1.2.6.2)

(1.2.6.3)