

appVersion(4) = "0.99.7921.69"

appVersion(4) = "0.99.7921.69"

num2expr

Converts an expression with binary operators to its functional representation

```

N2E(Ej) := for k ∈ [1..cols(N2E)]
  [ ans := concat( "Clear(", N2E2k, "(a,b))" ) str2num(ans) ]
  #N2E#(Tj, bj) := str2num(concat("equirep(", Tj, ", a⊗b, ", bj, ")"))
  #N2E#(Xj, aj, bj) := tj := strrep(strrep(num2str(Xj), "{", "(", ")", "}"))
  tj := strrep(strrep(tj, "(", "#N2E#", ")", aj, "⊗"))
  for Nj ∈ [1..EquRepN]
    if num2str(tj) = num2str(tj := #N2E#(num2str(tj), bj))
      break
    else
      continue
  str2num(strrep(num2str(tj), "#N2E#(", " ("))
  ansj := equirep(Ej, a ⊕ b, #Xor#(a, b))
  ansj := equirep(ansj, a ∧ b, #And#(a, b))
  ansj := equirep(ansj, a ∨ b, #Or#(a, b))
  ansj := equirep(ansj, a > b, b < a)
  ansj := equirep(ansj, a ≥ b, b ≤ a)
  ansj := equirep(ansj, a = b, #Equals#(a, b))
  ansj := equirep(ansj, a ≠ b, #NotEquals#(a, b))
  ansj := equirep(ansj, a < b, #LessThan#(a, b))
  ansj := equirep(ansj, a ≤ b, #LessOrEquals#(a, b))
  ansj := str2num(strrep(num2str(ansj), "-", "+#neg#"))
  for kj ∈ [1..3]
    ansj := #N2E#(num2str(ansj), N2E1kj, concat("#", N2E2kj, "#(a,b)"))
  ansj := #N2E#(ansj, "/", "#Times#(a, #Inv#(b))")
  for kj ∈ [1..cols(N2E)]
    str2num(concat("#", N2E2kj, "#(a,b):", N2E2kj, "(a,b)"))
  [ #Inv#(b) := Pow(b, #neg#) #neg# := -1 #N2E#(bj) := bj ]
  ansj := num2str(num2str(ansj))
  str2num(num2str(strrep(num2str(ansj), "#N2E#", "")))

```

Matrix of binary operators

N2E :=	"+"	"⋅"	"⋆"	"&"	" "	"≡"	"≠"	"<"	"≤"	"⊗"
	"Plus"	"Times"	"Pow"	"And"	"Or"	"Equals"	"NotEquals"	"LessThan"	"LessOrEquals"	"Xor"

Replacing deep

EquRepN := 20

$$\text{equrep}\left(\frac{u}{v \cdot w}, a \cdot b, \text{times}(a, b)\right) = \frac{u}{v \cdot w} \qquad N2E\left(\frac{u}{v \cdot w}\right) = \text{Times}\left(u, \text{Pow}\left(\text{Times}(v, w), -1\right)\right)$$

□ — num2expr examples —

Assign binary operators

```
for k ∈ [1..cols(N2E)]
  ans := str2num(concat(N2E2 k, "(a,b):(a", N2E1 k, "b)"))
```

$$E1 := N2E\left(\frac{1}{x + 7 \cdot y}\right) = \text{Times}\left(1, \text{Pow}\left(\text{Plus}(x, \text{Times}(7, y)), -1\right)\right)$$

$$E1 = \frac{1}{x + 7 \cdot y}$$

$$E2 := N2E\left[\left[\begin{array}{l} (a \wedge b) \vee (c \wedge (\neg d)) + \cos(x) \\ ((A \wedge B) \vee (C \wedge (\neg D))) = E \end{array}\right]\right] = \left[\begin{array}{l} \text{Plus}\left(\text{Or}\left(\text{And}(a, b), \text{And}(c, \neg d)\right), \cos(x)\right) \\ \text{Equals}\left(\text{Or}\left(\text{And}(A, B), \text{And}(C, \neg D)\right), E\right) \end{array}\right]$$

$$E2 = \left[\begin{array}{l} (a \wedge b) \vee (c \wedge \neg d) + \cos(x) \\ ((A \wedge B) \vee (C \wedge \neg D)) = E \end{array}\right]$$

$$E3 := N2E\left(\frac{-3 + \cos(-a) + b^{u - (v + w)}}{A \cdot X + B \cdot Y}\right) = \text{Times}\left(\text{Plus}\left(1, \text{Times}\left(\text{Pow}\left(b, \text{Plus}\left(\text{Plus}(v, w), \text{Times}(-1, u)\right)\right), \text{Plus}\left(\text{Times}(3, -1), \cos(a)\right)\right)\right), \text{Plus}\left(\text{Times}(3, -1), \cos(a)\right)\right), \cos(a)\right)$$

$$E3 = \frac{1 + b^{v + w - u} \cdot (-3 + \cos(a))}{b^{v + w - u} \cdot (A \cdot X + B \cdot Y)}$$

Unassign binary operators

```
for k ∈ [1..cols(N2E)]
  [ans := concat("Clear(", N2E2 k, "(a,b)"), str2num(ans)]
```

$$E1 = \text{Times}\left(1, \text{Pow}\left(\text{Plus}(x, \text{Times}(7, y)), -1\right)\right)$$

$$E2 = \left[\begin{array}{l} \text{Plus}\left(\text{Or}\left(\text{And}(a, b), \text{And}(c, \neg d)\right), \cos(x)\right) \\ \text{Equals}\left(\text{Or}\left(\text{And}(A, B), \text{And}(C, \neg D)\right), E\right) \end{array}\right]$$

$$E3 = \text{Times}\left(\text{Plus}\left(1, \text{Times}\left(\text{Pow}\left(b, \text{Plus}\left(\text{Plus}(v, w), \text{Times}(-1, u)\right)\right), \text{Plus}\left(\text{Times}(3, -1), \cos(a)\right)\right)\right), \text{Plus}\left(\text{Times}(3, -1), \cos(a)\right)\right), \cos(a)\right)$$