

```

-- primitive operators for regular expressions:
P*   ==: P*
      -- Kleene star-operator
P∪Q  ==: P∪Q
      -- union
P◦Q  ==: P◦Q
      -- concatenation
0    ==: 0
      -- aliases and derived operators:
∅    ==: 0
ι    ==: ∅*
P+   ==: P◦P*
P⊆Q  ↔: is_∪(Q, P)
∪([]) ==: ∅
∪([P|Q]) ==: P∪∪(Q)
semiGroup(P)  Θ: [
  P(P(Q, R), S)=P(Q, P(R, S))]
  -- : associative law
leftMonoid(P, Q)  Θ: [
  semiGroup(P),
  P(Q, R)=R]
  -- : left monoid
rightMonoid(P, Q)  Θ: [
  semiGroup(P),
  P(R, Q)=R]
  -- : right monoid
monoid(P, Q)  Θ: [
  leftMonoid(P, Q),
  P(R, Q)=R]
  -- : bilateral monoid
commMonoid(P, Q)  Θ: [
  leftMonoid(P, Q),
  P(R, S)=P(S, R)]
  -- : commutative monoid
leftDistributes(P, Q)  Θ: [
  P(R, Q(S, T))=Q(P(R, S), P(R, T))]
  -- : left distributive law
rightDistributes(P, Q)  Θ: [
  P(Q(R, S), T)=Q(P(R, T), P(S, T))]
  -- : right distributive law

```