```
Rational: trait
  % For use in the trait FloatingPoint.
  includes
    Exponentiation (Q for T, P for N),
    IntegerAndPositive (Int, P),
    MinMax (Q),
    TotalOrder (Q)
  introduces
    __/__: Int, P → Q
    0, 1: → Q
    -__, __^-1, abs: Q → Q
    __+__, __*__, __-__, __/__: Q, Q → Q
  asserts
    Q generated by __/__:Int,P→Q
    ∀ i, i1, i2: Int, p, p1, p2, p3: P, q, q1, q2: Q
      0/p == 0;
      int(p)/p == 1;
      i1/p1 = i2/p2 == i1 * int(p2) = i2 * int(p1);
      -(i/p) == (-i)/p;
      (int(p1)/p2)^-1 == int(p2)/p1;
      (-q)^-1 == -(q^-1);
      abs(i/p) == abs(i)/p;
      (i1/p) + (i2/p) == (i1 + i2)/p;
      (i1/p1) * (i2/p2) == (i1 * i2)/(p1 * p2);
      q1 - q2 == q1 + (-q2);
      q1/q2 == q1 * (q2^-1);
      (i1/p) < (i2/p) == i1 < i2
  implies
    AC (+, Q),
    AC (*, Q),
    Field (Q for T)
    ∀ i, i1, i2: Int, p, p1, p2, p3: P, q: Q
      q + 0 == q;
      -q == 0 - q;
      (i1/p) - (i2/p) == (i1 - i2)/p;
      q * 0 == 0;
      q * 1 == q;
      q^-1 == 1/q;
      (i/p1)/(int(p2)/p3) == (i * int(p3))/(p1 * p2)
  converts
      0:→Q, 1:→Q, -:Q →Q, ^-1, abs:Q →Q,
      +:Q,Q→Q, -:Q,Q→Q, *:Q,Q→Q, /:Q,Q→Q,
      **:Q,P→Q, min:Q,Q→Q, max:Q,Q→Q,
      <:Q,Q→Bool, >:Q,Q→Bool,
      ≤:Q,Q→Bool, ≥:Q,Q→Bool
```

exempting 0^{-1}