```
let%expect_test "P_1(t) P_2(t) - P_1(t) + P_2(t)" =
  let open Expr.Syntax in
  let x = scalar_var `X in
  let y = scalar_var `Y in
  let coeff1 = -x + y in
  let coeff2 = x * -y in
  let coeff3 = x - y in
  let poly1 = Poly.of_alist_exn [ 2, coeff1; 1, coeff2 ] in
  let poly2 = Poly.of_alist_exn [ 2, coeff3; 1, coeff2; 0, coeff1 ] in
  let poly = Poly.Syntax.((poly1 * poly2) - poly1 + poly2) in
  let poly_numbers =
    Poly.to_map
      poly
      ~values:(function
        | `X -> 24.
        | `Y -> 4.)
      ~scoped_values:never_returns
    |> Polynomial.of_map ~eps:1e-7
  in
  print_s [%sexp (Polynomial.to_string_hum ~var:"t" poly_numbers : string)];
  [%expect {| -400t^4+9_656t^2+1_920t^1-20t^0 |}];
  let roots = Equation.roots poly_numbers ~eps:1e-7 in
  print_s [%sexp (roots : float list)];
  [%expect
    {|
    (-4.810405636875929l -0.2091233965987575б 0.009921606232339078
     5.009607427277204б) |}]
;;
```