```
In[1]:= s[x_] := Switch[{x},
    {u[1]},
    u[2],
    {u[2]},
    -u[1],
    {-u[1]},
    -u[2],
    {-u[2]},
    u[1]
   ]

In[2]:= t[x_] := Switch[{x},
    {u[1]},
    u[2],
    {u[2]},
    u[1],
    {-u[1]},
    -u[2],
    {-u[2]},
    -u[1]
   ]

In[3]:= id[x_] := x

In[4]:= EvaluateFunction[f_, a_, b_] := Switch[{a == 0, b == 0},
    {True, True},
    1,
    {True, False},
    f[u[2]] * EvaluateFunction[f, 0, b - 1],
    {False, True},
    f[u[1]] * EvaluateFunction[f, a - 1, b],
    {False, False},
    f[u[1]] * f[u[2]] * EvaluateFunction[f, a - 1, b - 1]
   ]

In[5]:= FunctionFromList[list_] := Module[{function, n, output, i, f},
    function[x_] := Module[{},
      n = Length[list];
      output = x;
      For[i = n, i ≥ 1, i--,
       f = list[[i]];
       output = f[output];
      ];
      output
     ];
    function
   ]

In[6]:= ss = FunctionFromList[{s, s}]; sss = FunctionFromList[{s, s, s}];
st = FunctionFromList[{s, t}]; sst = FunctionFromList[{s, s, t}];
ssst = FunctionFromList[{s, s, s, t}];
```

```
In[7]:= BasisMonomials = Flatten[Table[Table[u[1]^i u[2]^j, {i, 0, 3}], {j, 0, 1}]]
```

Out[7]= $\{1, u[1], u[1]^2, u[1]^3, u[2], u[1]\,u[2], u[1]^2\,u[2], u[1]^3\,u[2]\}$

```
In[8]:= U[1] = 2^(1/4); U[2] = 2^(1/4) I;

In[9]:= NumericalBasisMonomials =
         Simplify[Flatten[Table[Table[U[1]^i U[2]^j, {i, 0, 3}], {j, 0, 1}]]]
```

Out[9]= $\{1, 2^{1/4}, \sqrt{2}, 2^{3/4}, i\,2^{1/4}, i\sqrt{2}, i\,2^{3/4}, 2\,i\}$

```
In[10]:= PolynomialFromCoefficients[coefficients_] :=
          Module[{polynomial, i, coefficient, monomial},
           polynomial = 0;
           For[i = 1, i ≤ Length[coefficients], i++,
            coefficient = coefficients[[i]];
            monomial = BasisMonomials[[i]];
            polynomial += coefficient * monomial;
           ];
           polynomial
          ]


         TransformationRuleII[polynomial_] := Module[
           {coefficientList, dimensions, newPolynomial, i, j, coefficient, a, b, monomial},
           coefficientList = CoefficientList[polynomial, {u[1], u[2]}];
           dimensions = Dimensions[coefficientList];
           newPolynomial = 0;
           For[i = 1, i ≤ dimensions[[1]], i++,
            For[j = 1, j ≤ dimensions[[2]], j++,
              coefficient = coefficientList[[i]][[j]];
              a = i - 1;
              b = j - 1;
              Switch[{b ≤ 1},
               {True},
               monomial = u[1]^a * u[2]^b,
               {False},
               monomial = u[1]^a * u[2]^(b - 2) * (-u[1]^2)
              ];
              newPolynomial = newPolynomial + coefficient * monomial;
             ];
           ];
           Expand[newPolynomial]
          ]
         TransformationRuleI[polynomial_] := Module[
           {coefficientList, dimensions, newPolynomial, i, j, coefficient, a, b, monomial},
           coefficientList = CoefficientList[polynomial, {u[1], u[2]}];
           dimensions = Dimensions[coefficientList];
           newPolynomial = 0;
           For[i = 1, i ≤ dimensions[[1]], i++,
            For[j = 1, j ≤ dimensions[[2]], j++,
```

```
      coefficient = coefficientList[[i]][[j]];
      a = i - 1;
      b = j - 1;
      Switch[{a ≤ 3},
       {True},
       monomial = u[1]^a * u[2]^b,
       {False},
       monomial = u[1]^(a - 4) * (2) * u[2]^b
      ];
      newPolynomial = newPolynomial + coefficient * monomial;
     ];
   ];
   Expand[newPolynomial]
  ]
ApplyRuleIIRepeatedly[polynomial_] :=
 Module[{revisedPolynomial, continue, revisedPolynomialPrime},
  revisedPolynomial = polynomial;
  continue = True;
  While[continue,
   revisedPolynomialPrime = TransformationRuleII[revisedPolynomial];
   If[revisedPolynomialPrime == revisedPolynomial, continue = False];
   revisedPolynomial = revisedPolynomialPrime;
  ];
  revisedPolynomial
 ]
ApplyRuleIRepeatedly[polynomial_] :=
 Module[{revisedPolynomial, continue, revisedPolynomialPrime},
  revisedPolynomial = polynomial;
  continue = True;
  While[continue,
   revisedPolynomialPrime = TransformationRuleI[revisedPolynomial];
   If[revisedPolynomialPrime == revisedPolynomial, continue = False];
   revisedPolynomial = revisedPolynomialPrime;
  ];
  revisedPolynomial
 ]
ReducePolynomial[polynomial_] :=
 ApplyRuleIRepeatedly[ApplyRuleIIRepeatedly[polynomial]]
```

In[16]:=
```
EvaluateFunctionAtPolynomial[f_, polynomialCoefficients_] :=
 Module[{newPolynomial, i, coefficient, a, b, newCoefficients,
   inputPolynomial, reducedOutputPolynomialCoefficients},
  inputPolynomial = PolynomialFromCoefficients[polynomialCoefficients];
  Print["input polynomial = ", inputPolynomial];
  newPolynomial = 0;
  For[i = 0, i ≤ 3, i++,
   coefficient = polynomialCoefficients[[i + 1]];
   newPolynomial = newPolynomial + coefficient * EvaluateFunction[f, i, 0];
  ];
  For[i = 0, i ≤ 3, i++,
   coefficient = polynomialCoefficients[[i + 5]];
   newPolynomial = newPolynomial + coefficient * EvaluateFunction[f, i, 1];
  ];
  Print["output polynomial = ", newPolynomial];
  newPolynomial = ReducePolynomial[newPolynomial];
  Print["reduced output polynomial = ", newPolynomial];
  reducedOutputPolynomialCoefficients =
   Flatten[Transpose[CoefficientList[newPolynomial, {u[1], u[2]}]]];
  reducedOutputPolynomialCoefficients
 ]
```

In[17]:=
```
EvaluateFunctionAtPolynomialSilent[f_, polynomialCoefficients_] :=
 Module[{newPolynomial, i, coefficient, a, b, newCoefficients,
   inputPolynomial, reducedOutputPolynomialCoefficients},
  inputPolynomial = PolynomialFromCoefficients[polynomialCoefficients];
  newPolynomial = 0;
  For[i = 0, i ≤ 3, i++,
   coefficient = polynomialCoefficients[[i + 1]];
   newPolynomial = newPolynomial + coefficient * EvaluateFunction[f, i, 0];
  ];
  For[i = 0, i ≤ 3, i++,
   coefficient = polynomialCoefficients[[i + 5]];
   newPolynomial = newPolynomial + coefficient * EvaluateFunction[f, i, 1];
  ];
  newPolynomial = ReducePolynomial[newPolynomial];
  reducedOutputPolynomialCoefficients =
   Flatten[Transpose[CoefficientList[newPolynomial, {u[1], u[2]}]]];
  reducedOutputPolynomialCoefficients
 ]
```

```
In[18]:= FixedFieldBasis[F_] :=
  Module[{inputCoefficients, outputCoefficients, identificationCoefficients,
     identificationMatrix, i, expression, row, nullspace, basis,
     numericalBasis, coefficients, vector, numericalVector,
     j, coefficient, monomial, numericalMonomial},
   inputCoefficients = {a, b, c, d, e, f, g, h};
   outputCoefficients = EvaluateFunctionAtPolynomial[F, inputCoefficients];
   Print["implied equations:  ", inputCoefficients, " = ", outputCoefficients];
   identificationCoefficients = inputCoefficients - outputCoefficients;
   identificationMatrix = {};
   For[i = 1, i ≤ Length[identificationCoefficients], i++,
    expression = identificationCoefficients[[i]];
    row = Table[Coefficient[expression, inputCoefficients[[i]]],
       {i, 1, Length[inputCoefficients]}];
    identificationMatrix = Append[identificationMatrix, row];
   ];
   Print["coefficient matrix of implied equations:  ",
    MatrixForm[identificationMatrix]];
   nullspace = NullSpace[identificationMatrix];
   Print["basis for nullspace:  ", MatrixForm[Transpose[nullspace]]];

   basis = {};
   numericalBasis = {};
   For[i = 1, i ≤ Length[nullspace], i++,
    coefficients = nullspace[[i]];
    vector = 0;
    numericalVector = 0;
    For[j = 1, j ≤ Length[coefficients], j++,
     coefficient = coefficients[[j]];
     monomial = BasisMonomials[[j]];
     numericalMonomial = NumericalBasisMonomials[[j]];
     vector += coefficient * monomial;
     numericalVector += coefficient * numericalMonomial;
    ];
    basis = Append[basis, vector];
    numericalBasis = Append[numericalBasis, Simplify[numericalVector]];
   ];
   Print["basis for fixed field:  ", basis];
   Print["numerical basis for fixed field:  ", numericalBasis];

  ]
```

```
In[19]:= FixedFieldBasisSilent[F_, name_] :=
  Module[{inputCoefficients, outputCoefficients, identificationCoefficients,
    identificationMatrix, i, expression, row, nullspace, basis,
    numericalBasis, coefficients, vector, numericalVector,
    j, coefficient, monomial, numericalMonomial},
   inputCoefficients = {a, b, c, d, e, f, g, h};
   outputCoefficients = EvaluateFunctionAtPolynomialSilent[F, inputCoefficients];
   identificationCoefficients = inputCoefficients - outputCoefficients;
   identificationMatrix = {};
   For[i = 1, i ≤ Length[identificationCoefficients], i++,
    expression = identificationCoefficients[[i]];
    row = Table[Coefficient[expression, inputCoefficients[[i]]],
      {i, 1, Length[inputCoefficients]}];
    identificationMatrix = Append[identificationMatrix, row];
   ];
   nullspace = NullSpace[identificationMatrix];
   basis = {};
   numericalBasis = {};
   For[i = 1, i ≤ Length[nullspace], i++,
    coefficients = nullspace[[i]];
    vector = 0;
    numericalVector = 0;
    For[j = 1, j ≤ Length[coefficients], j++,
     coefficient = coefficients[[j]];
     monomial = BasisMonomials[[j]];
     numericalMonomial = NumericalBasisMonomials[[j]];
     vector += coefficient * monomial;
     numericalVector += coefficient * numericalMonomial;
    ];
    basis = Append[basis, vector];
    numericalBasis = Append[numericalBasis, Expand[Simplify[numericalVector]]];
   ];
   Print["basis for field fixed by ", name, " = ", basis, " = ", numericalBasis];
   basis

  ]

In[20]:= FixedFieldBasis[t]
```

```
input polynomial = a + b u[1] + c u[1]^2 + d u[1]^3 + e u[2] + f u[1] u[2] + g u[1]^2 u[2] + h u[1]^3 u[2]

output polynomial = a + e u[1] + b u[2] + f u[1] u[2] + c u[2]^2 + g u[1] u[2]^2 + d u[2]^3 + h u[1] u[2]^3

reduced output polynomial =
 a + e u[1] - c u[1]^2 - g u[1]^3 + b u[2] + f u[1] u[2] - d u[1]^2 u[2] - h u[1]^3 u[2]

implied equations:  {a, b, c, d, e, f, g, h} = {a, e, -c, -g, b, f, -d, -h}
```

coefficient matrix of implied equations: $\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix}$

basis for nullspace: $\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$

basis for fixed field: $\{-u[1]^3 + u[1]^2 u[2], u[1] u[2], u[1] + u[2], 1\}$

numerical basis for fixed field: $\{(-1 + i)\, 2^{3/4}, i \sqrt{2}, (1 + i)\, 2^{1/4}, 1\}$

In[21]:=
```
FixedFieldBasisSilent[s, "s"];
```

basis for field fixed by s = $\{u[1]^3 u[2], 1\}$ = $\{2 i, 1\}$

In[22]:=
```
Module[{automorphisms, i, F, name, names, basis},
 automorphisms = {id, s, ss, sss, t, st, sst, ssst};
 names = {"id", "s", "ss", "sss", "t", "st", "sst", "ssst"};
 For[i = 1, i ≤ Length[automorphisms], i++,
  Print[
   "---------------------------------------------------------------------------
     ------------------------------------------------------------"];
  F = automorphisms[[i]];
  name = names[[i]];
  basis = FixedFieldBasisSilent[F, name];
  Print["dimension = ", Length[basis]];
 ];
]
```

----

basis for field fixed by id = $\{u[1]^3 u[2], u[1]^2 u[2], u[1] u[2], u[2], u[1]^3, u[1]^2, u[1], 1\}$ = $\{2 i, i\, 2^{3/4}, i \sqrt{2}, i\, 2^{1/4}, 2^{3/4}, \sqrt{2}, 2^{1/4}, 1\}$

dimension = 8

----

basis for field fixed by s = $\{u[1]^3 u[2], 1\}$ = $\{2 i, 1\}$

dimension = 2

----

basis for field fixed by ss = $\{u[1]^3 u[2], u[1] u[2], u[1]^2, 1\}$ = $\{2 i, i \sqrt{2}, \sqrt{2}, 1\}$

dimension = 4

----

basis for field fixed by sss = $\{u[1]^3 u[2], 1\}$ = $\{2 i, 1\}$

dimension = 2

----

basis for field fixed by t = $\{-u[1]^3 + u[1]^2 u[2], u[1] u[2], u[1] + u[2], 1\}$ = $\{(-1 + i)\, 2^{3/4}, i \sqrt{2}, (1 + i)\, 2^{1/4}, 1\}$

dimension = 4

----

basis for field fixed by st = $\{u[1]^2 u[2], u[2], u[1]^2, 1\}$ = $\{i\, 2^{3/4}, i\, 2^{1/4}, \sqrt{2}, 1\}$

dimension = 4

----

basis for field fixed by sst = $\{u[1]^3 + u[1]^2 u[2], u[1] u[2], -u[1] + u[2], 1\}$ = $\{(1 + i)\, 2^{3/4}, i \sqrt{2}, (-1 + i)\, 2^{1/4}, 1\}$

dimension = 4

----

basis for field fixed by ssst = $\{u[1]^3, u[1]^2, u[1], 1\}$ = $\{2^{3/4}, \sqrt{2}, 2^{1/4}, 1\}$

dimension = 4