SL_2 Morphisms

Alexandre Borovik $\&$ Şükrü Yalçınkaya

1 Part 1: Preliminary functions

`IsInCenter(S,g)`

Input:

- "S" is a list which is a generating set of a finite group.

- "g" is an element in the group generated by "S".

Output:

- True if "g" commutes with elements from "S" and therefore with all elements in the group generated by "S".

- False: otherwise

`Invo(S,Eo)`

Input:

- "S" is a list which is a generating set of a finite group.

- "Eo" is the odd part of an exponent of the group generated by "S".

Output:

- An involution produced from a random element in the group generated by "S".

`CentralizerInvo(S,i,Eo)`

Input:

- "S" is a list which is a generating set of a finite group.

- "i" is an involution in the group generated by "S".

- "Eo" is the odd part of an exponent of the group generated by "S".

Output:

- A list of elements from the centralizer of "i" in the group generated by "S". The elements of the list is produced by using only the map ζ_1.

`TorusRandom(S,C)`

Input:

- "S" is a list which is a generating set of a group isomorphic to SL_2.

- "C" is a list which is a generating set for a centralizer of an involution in the group generated by "S".

Output:

- A random element from the torus in the group generated by the list "C" avoiding the possibility to construct the involution in the torus.

CentralizerProtoInvo(S,C,t,Eo)

Input:

- "S" is a list which is a generating set of a group isomorphic to $(\mathrm{P})\mathrm{SL}_2$ or PGL_2.

- "C" is a list which is a generating set of a centralizer of an involution in the group generated by "S".

- "t" is a semisimple group element of odd order in the group generated by "S".

- "Eo" is the odd part of an exponent of the group generated by "S".

Output:

- A list of elements from the centralizer of an involution which inverts the torus in the group generated by "C" and centralizes "t". The involution may not exist in PSL_2 but exists in PGL_2. This function is used to construct PGL_2 from PSL_2. The involution is the corresponding diagonal automosphism.

IsUnipotent(S,i,j,Eo)

Input:

- "S" is a list which is a generating set of a finite group isomorphic to PGL_2 or $(\mathrm{P})\mathrm{SL}_2$.

- "i" and "j" are (non-central) involutions in the group generated by "S".

- "Eo" is the odd part of an exponent of the group generated by "S".

Output:

- The truth value whether the product "i*j" is a unipotent element or not.

BinaryRep(n)

Input:

- "n" is a natural number.

Output:

- A list consisting of decreasing natural numbers $c_1, \ldots, c_k$ such that $2^{c_1} + \ldots + 2^{c_k} = n$.

---

2 Part 2: The semi-direct product PGL_2

`SetUpForPGL2(S,Eo)`

Input:

- "S" is a list which is a generating set of a finite group isomorphic to $(\mathrm{P})\mathrm{SL}_2$.

- "Eo" is the odd part of an exponent of the group generated by "S".

Output:

- A list whose components are as follows:

    1- The list "S".

    2- A list of elements from a centralizer of an involution, "i", inverted by a diagonal automorphism.

    3- A list of semisimple elements generating a torus centralized by the same diagonal automorphism.

    4- The involution "i".

`PseudoRandomPGL2(Snew)`

Input:

- "Snew" is the output of the function "SetUpForPGL2".

Output:

- A pseudo-random element from the semidirect product isomorphic to PGL_2.

`PseudoRandomPSL2(Snew)`

Input:

- "Snew" is the output of the function "SetUpForPGL2".

Output:

- A pseudo-random element from the semidirect product but belongs to the subgroup PSL_2.

`GeneratorsPGL2(Snew)`

Input:

- "Snew" is the output of the function "SetUpForPGL2".

Output:

- A list of random group elements from the semidirect product isomorphic to PGL_2.

`IsIdentitySD(Snew,g)`

Input:

- "Snew" is the output of the function "SetUpForPGL2".

- "g" is an element from the semidirect product isomorphic to PGL_2.

3

Output:

- The truth value whether the element "g" is the identity element in the semidirect product.

ProductSD(g,h)

Input:

- "g" and "h" are elements from the semidirect product isomorphic to PGL_2.

Output:

- The product of "g" and "h" in the semidirect product isomorphic to PGL_2.

PseudoRandomSD(Ss)

Input:

- "Ss" is a generating set for a subgroup in the semidirect product isomorphic to PGL_2.

Output:

- A pseudo-random element from the group generated by "Ss".

PseudoRandomCyclicSD(g)

Input:

- "g" is an element from the semidirect product isomorphic to PGL_2.

Output:

- A pseudo-random element from the cyclic group $\langle g \rangle$.

PowerSD(g,n)

Input:

- "g" is an element from the semidirect product isomorphic to PGL_2.
- "n" is a natural number.

Output:

- The nth power of "g" in the group.

InverseSD(g)

Input:

- "g" is an element from the semidirect product isomorphic to PGL_2.

Output:

- The inverse of "g" in the group.

```
ConjugationSD(g,h)
```

Input:

- "g" and "h" are elements from the semidirect product isomorphic to PGL_2.

Output:

- g^h.

```
EquivalentInSD(Snew,g,h)
```

Input:

- "Snew" is the output of the function "SetUpForPGL2".

- "g" and "h" are elements from the semidirect product isomorphic to PGL_2.

Output:

- The truth value whether the elements "g" and "h" represent the same element in the semidirect product.

```
2HeightSD(Snew,g,Eo)
```

Input:

- "Snew" is the output of the function "SetUpForPGL2".

- "g" is an element from our semidirect product isomorphic to PGL_2.

- "Eo" is the odd part of an exponent of the group.

Output:

- The 2-height of the group element "g".

```
4DividesOrderSD(Snew,g,Eo)
```

Input:

- "Snew" is the output of the function "SetUpForPGL2".

- "g" is an element from the semidirect product isomorphic to PGL_2.

- "Eo" is the odd part of an exponent of the group.

Output:

- The truth value whether the order of "g" is divisible by 4.

```
ElementOrder4SD(Snew,g,Eo)
```

Input:

- "Snew" is the output of the function "SetUpForPGL2".

- "g" is an element from the semidirect product isomorphic to PGL_2.

- "Eo" is the odd part of an exponent of the group.

Output:

- An element of order 4 in the cyclic group generated by "g" — this function is used precisely when the order of "g" is divisible by 4.

InvoSD(Snew,Eo)

Input:

- "Snew" is the output of the function "SetUpForPGL2".

- "Eo" is the odd part of an exponent of the group.

Output:

- An involution from the semidirect product isomorphic to PGL_2.

InvoSubSD(Snew,Ss,Eo)

Input:

- "Snew" is the output of the function "SetUpForPGL2".

- "Ss" is a generating set of a subgroup in the semidirect product isomorphic to PGL_2.

- "Eo" is the odd part of an exponent of the group.

Output:

- An involution produced from a random element in the group generated by "Ss".

InvoEvenSD(Snew,g)

Input:

- "Snew" is the output of the function "SetUpForPGL2".

- "g" is an element from the semidirect product isomorphic to PGL_2.

Output:

- The involution produced from the element "g" of even order — it is known that "g" has even order for this function.

InvoEvenHeightSD(Snew,g)

Input:

- "Snew" is the output of the function "SetUpForPGL2".

- "g" is an element from the semidirect product isomorphic to PGL_2.

Output:

- An involution produced from the element "g" of even order — it is known that "g" has even order for this function.

- The 2-height of the order of the element "g".

`CentralizerInvoSD(Snew,i,Eo)`

Input:

- "Snew" is the output of the function "SetUpForPGL2".

- "i" is an involution in the semidirect product isomorphic to PGL_2.

- "Eo" is the odd part of an exponent of the group.

Output:

- A list of elements from the centralizer of "i" in the semidirect product isomorphic to PGL_2. The elements of the list is produced by using only the map ζ_1.

`CentralizerInvoInSubgroupSD(Snew,Ss,i,Eo)`

Input:

- "Snew" is the output of the function "SetUpForPGL2".

- "Ss" is a generating set for a subgroup in the semidirect product isomorphic to PGL_2.

- "i" is an involution in the semidirect product isomorphic to PGL_2 belonging to the subgroup generated by "Ss".

- "Eo" is the odd part of an exponent of the group.

Output:

- A list of elements from the centralizer of "i" of the subgroup generated by the list "Ss". The elements of the list is produced by using only the map ζ_1.

`TorusRandomSD(Snew,C)`

Input:

- "Snew" is the output of the function "SetUpForPGL2".

- "C" is a generating set of a centralizer of an involution in the semidirect product isomorphic to PGL_2.

Output:

- A random element from the torus in the group generated by "C" avoiding the possibility to construct the involution in the torus.

`SquareRootTorusSD(Snew,C,g,Eo)`

Input:

- "Snew" is the output of the function "SetUpForPGL2".

- "C" is a list which is a generating set of a centralizer of an involution.

- "g" is an element from our semidirect product isomorphic to PGL_2.

- "Eo" is the odd part of an exponent of the group.

Output:

- Either
  - a square root of the group element "g", if exists, inside the torus it lies in, or
  - false.

ConjugatingInvoSD(Snew,Ss,i,j,Eo)

Input:

- "Snew" is the output of the function "SetUpForPGL2".
- "Ss" is a generating set of a subgroup in the semidirect product isomorphic to PGL_2.
- "i" and "j" are involutions from our semidirect product isomorphic to PGL_2.
- "Eo" is the odd part of an exponent of the group.

Output:

- A conjugating involution "k" satisfying $i^k = j$. The involution "k" belongs to the subgroup generated by the list "Ss"

DecompositionIntoInvosSD(Snew,g,Eo)

Input:

- "Snew" is the output of the function "SetUpForPGL2".
- "g" is an element from our semidirect product isomorphic to PGL_2.
- "Eo" is the odd part of an exponent of the group.

Output:

- A list consisting of two involutions whose product is the group element "g".

IsUnipotentSD(Snew,i,j,Eo)

Input:

- "Snew" is the output of the function "SetUpForPGL2".
- "i" and "j" are involutions in the semidirect product isomorphic to PGL_2.
- "Eo" is the odd part of an exponent of the group.

Output:

- The truth value whether the product "i*j" is a unipotent element or not.

PermijkSD(Snew,i,j,Eo)

Input:

- "Snew" is the output of the function "SetUpForPGL2".
- "i" and "j" are commuting conjugate involutions in the semidirect product isomorphic to PGL_2.
- "Eo" is the odd part of an exponent of the group.

Output:

- An element of order 3 in the normaliser of $\langle i, j, i*j \rangle$ — the output conjugates the involutions in the following way: $i \rightarrow j \rightarrow i*j \rightarrow i$.

---

3 Part 3: Projective Geometry

`ReificationPSF(Snew,i,j,Eo)`

Input:

- "Snew" is the output of the function "SetUpForPGL2".

- "i" and "j" are projective points.

- "Eo" is the odd part of an exponent of the group.

Output:

- A projective point which is either an involution (as a projective point) "k" where the projective points "i" and "j" lie on a projective line determined by the involution "k" or a unipotent element (as a projective point) "u" where "i" and "j" lie on the parabolic line determined by "u". In other words, either an involution "k" which commutes with both "i" and "j" (as group elements) or a unipotent element "u".

`IntersectionLinesPSF(Snew,i1,i2,j1,j2,Eo)`

Input:

- "Snew" is the output of the function "SetUpForPGL2".

- "i1", "i2", "j1"and "j2" are projective points.

- "Eo" is the odd part of an exponent of the group.

Output:

- The projective point corresponding to the intersection of the lines determined by the projective points "i1", "i2" and "j1", "j2".

`UnitElementsSD(Snew,i,j,k,ti,elt3,Eo)`

Input:

- "Snew" is the output of the function "SetUpForPGL2".

- "i", "j" and "k" are involutions determining the vertices of the basis triangle for the projective geometry.

- "ti" is an element of order 4 in $C_G(i)$ whose square is "i".

- "elt3" is an element of order 3 permuting "i, "j", "k".

- "Eo" is the odd part of an exponent of the group.

Output:

- A list [P1,P2,P3,Q] where "P1,P2,P3" are the involutions corresponding to the unit elements on the lines "j ∨ k", "i ∨ k" and "i ∨ j", respectively; and "Q" is the involution corresponding to the homogenous coordinate (1,1,1).

`ToolBoxSL2(S,E)`

Input:

- "S" is a list which is a generating set of a finite group isomorphic to SL_2.

9

- "E" is the exponent of the group generated by "S".

Output:

- A list whose components are as follows:

  1- The output of the function "SetUpForPGL2".

  2- A list of elements serving as a generating set for the semidirect product isomorphic to PGL_2.

  3,4,5- Three commuting involutions forming the vertices of the projective plane.

  6- An element of order 3 permuting the three commuting involutions.

  7,8,9- A unity element on the corresponding coordinate axes.

  10- A generating set for the centralizer of a fixed involution (item 3) which is a vertex determining the projective plane — the black box field is constructed on the corresponding axis.

  11- The point which serves as 0 in the black box field.

  12- The projective point corresponding to the homogenous coordinate (1,1,1).

  13- Odd part of the exponent of the group generated by the list "S".

  14- Binary representation of the odd part of the exponent.

  15- An element of order 4 in the item 10 whose square is the corresponding involution.

  16- Identity of the group generated by the list "S".

EquivalentInBBF(Snew,a,b)

Input:

- "Snew" is the output of the function "SetUpForPGL2".

- "a" and "b" are elements from our black box field.

Output:

- The truth value whether the elements "a" and "b" are representing the same element in the black box field.

AdditionBBF(TB,a,b)

Input:

- "TB" is the output of "ToolBoxSL2".

- "a" and "b" are elements from our black box field.

Output:

- The sum of the two black box field elements "a" and "b" in our black box field.

AdditiveInverseBBF(TB,a)

Input:

- "TB" is the output of "ToolBoxSL2".

- "a" is an element from our black box field.

Output:

- The additive inverse of "a" in our black box field.

MultiplicationBBF(TB,a,b)

Input:

- "TB" is the output of "ToolBoxSL2".

- "a" is an element from our black box field.

Output:

- The product of the two black box field elements "a" and "b" in our black box field.

MultiplicativeInverseBBF(TB,a)

Input:

- "TB" is the output of "ToolBoxSL2".

- "a" is an element from our black box field.

Output:

- The multiplicative inverse of "a" in our black box field.

PowerBinaryRepBBF(TB,a,l)

Input:

- "TB" is the output of "ToolBoxSL2".

- "a" is an element from our black box field.

- "l" is a list consisting of natural numbers.

Output:

- The power $a^{2^{l[1]}+2^{l[2]}+\cdots+2^{l[r]}}$ in our black box field.

2HeightBBF(TB,a)

Input:

- "TB" is the output of "ToolBoxSL2".

- "a" is an element from our black box field.

Output:

- The 2-height of the black box field element "a", that is, the maximum power of 2 appears in the multiplicative order of "a".

SquareRootBBF(TB,n)

Input:

- "TB" is the output of "ToolBoxSL2".

- "n" is an element from our black box field.

Output:

- Either

    - a square root of the black box field element "n", if exists; or
    - false, otherwise.

`MatrixMultiplication2x2BBF(TB,A,B)`

Input:

- "TB" is the output of "ToolBoxSL2".

- "A" and "B" are 2x2 matrices with entries (given as lists) from our black box field.

Output:

- The matrix which is the multiplication of the black box matrices "A" and "B".

`MatrixMultiplication3x3BBF(TB,A,B)`

Input:

- "TB" is the output of "ToolBoxSL2".

- "A" and "B" are 3x3 matrices with entries (given as lists) from our black box field.

Output:

- The matrix which is the multiplication of the black box matrices "A" and "B".

`Equivalent3x3MatricesBBF(Snew,A,B)`

Input:

- "Snew" is the output of the function "SetUpForPGL2".

- "A" and "B" are 3x3 matrices with entries (given as lists) from our black box field.

Output:

- The truth value whether the two 3x3 matrices "A" and "B" are equal to each other.

`Determinant3x3BBF(TB,A)`

Input:

- "TB" is the output of "ToolBoxSL2".

- "A" is a 3x3 matrix with entries (given as a list) from our black box field.

Output:

- The black box field element which is the determinant of the black box matrix "A".

```
HomogenousCoordinatesBBF(TB,P)
```

Input:

- "TB" is the output of "ToolBoxSL2".

- "P" is a black box projective point.

Output:

- The homogenous coordinates of the point "P" where its z-coordinate is set as 1. The coordinates are elements from our black box field.

```
ColumnBBF(TB,P)
```

Input:

- "TB" is the output of "ToolBoxSL2".

- "P" is a black box projective point.

Output:

- The corresponding column from the projective point "P" in the construction of SO_3 over our black box field.

```
InvoMatrixGeneralSD(TB,g)
```

Input:

- "TB" is the output of "ToolBoxSL2".

- "g" is an involution in our semidirect product isomorphic to PGL_2.

Output:

- The image of the involution "g" in $SO_3(K)$ where "g" does not commute with coordinate involutions.

```
InvoMatrixSpecialSD(TB,g)
```

Input:

- "TB" is the output of "ToolBoxSL2".

- "g" is an involution in our semidirect product isomorphic to PGL_2.

Output:

- The image of the involution "g" in $SO_3(K)$ where "g" commutes with one of the coordinate involutions and K our black box field.

```
BBGtoSO3InvoSD(TB,g)
```

Input:

- "TB" is the output of "ToolBoxSL2".

- "g" is an involution in our semidirect product isomorphic to PGL_2.

Output:

- The image of the involution "g" in $\mathrm{SO}_3(\mathrm{K})$ where K is our black box field.

RowAsBlackBoxElementBBF(TB,m)

Input:

- "TB" is the output of "ToolBoxSL2".

- "m" is a list corresponding to a row in the group SO3 over our black box field.

Output:

- The projective point whose homogenous coordinates is the row "m".

SO3toBBGInvo(TB,gK)

Input:

- "TB" is the output of "ToolBoxSL2".

- "gK" is an involution from the group $\mathrm{SO}_3(\mathrm{K})$ where K is our black box field.

Output:

- The involution whose image in $\mathrm{SO}_3(\mathrm{K})$ is the black box matrix "gK".

PseudoRandomField(TB)

Input:

- "TB" is the output of "ToolBoxSL2".

Output:

- A random element from our black box field. These elements are never unipotents, that is, we ignore the unipotent subgroups as black box field elements in this output.

SharpVsFlat(TB)

Input:

- "TB" is the output of "ToolBoxSL2".

Output:

- The list whose first component is the change of basis matrix from $\mathrm{SO}_3^{\sharp}$ to $\mathrm{SO}_3^{\flat}$ with entries from our black box field and the second component is the inverse of the first component.

UnipotentAsInvoProductsV1BBF(TB,gK)

Input:

- "TB" is the output of "ToolBoxSL2".

- "gK" is a unipotent element in $\mathrm{SL}_2(\mathrm{K})$ of the form [1,x,0,1] where K is our black box field.

Output:

- A list with two components consisting of involutions whose product gives the element "gK".

UnipotentAsInvoProductsV2BBF(TB,gK)

Input:

- "TB" is the output of "ToolBoxSL2".

- "gK" is a unipotent element in $\mathrm{SL}_2(\mathrm{K})$ of the form [1,0,x,1] where K is our black box field.

Output:

- A list with two components consisting of involutions whose product gives the element "gK".

ElementAsUnipotentProductsBBF(TB,gK)

Input:

- "TB" is the output of "ToolBoxSL2".

- "gK" is an arbitrary element in $\mathrm{SL}_2(\mathrm{K})$ where K is our black box field.

Output:

- A list consisting of unipotent elements whose product gives the element "gK".

PGL2toSO3Flat(TB,gK)

Input:

- "TB" is the output of "ToolBoxSL2".

- "gK" a 2x2 projective matrix with entries from our black box field K.

Output:

- The image of the matrix "gK" in $\mathrm{SO}_3^\flat(\mathrm{K})$ where K is our black box field.

SL2KtoBBGInvo(TB,svf,gK)

Input:

- "TB" is the output of "ToolBoxSL2".

- "svf" is the output of "SharpVsFlatKSD".

- "gK" is an involution in $\mathrm{SL}_2(\mathrm{K})$ represented as a 2x2 matrix where K is our black box field.

Output:

- The image of "gK" in our black box group.

`SL2KtoBBGv01BBF(TB,svf,ggK)`

Input:

- "TB" is the output of "`ToolBoxSL2`".

- "svf" is the output of "`SharpVsFlatKSD`".

- "ggK" is a unipotent element from $\mathrm{SL}_2(\mathsf{K})$ of type [1,0,x,1].

Output:

- The image of the element "ggK" in our black box group.

`SL2KtoBBGv02BBF(TB,svf,ggK)`

Input:

- "TB" is the output of "`ToolBoxSL2`".

- "svf" is the output of "`SharpVsFlatKSD`".

- "ggK" is a unipotent element from $\mathrm{SL}_2(\mathsf{K})$ of type [1,x,0,1].

Output:

- The image of the element "ggK" in our black box group.

`SL2KtoBBGv11BBF(TB,svf,gU)`

Input:

- "TB" is the output of "`ToolBoxSL2`".

- "svf" is the output of "`SharpVsFlatKSD`".

- "gU" is a list of unipotent elements from SL_2: [[1,x,0,1], [1,0,x,1]].

Output:

- The image of the element "gU" in our black box group.

`SL2KtoBBGv12BBF(TB,svf,gU)`

Input:

- "TB" is the output of "`ToolBoxSL2`".

- "svf" is the output of "`SharpVsFlatKSD`".

- "gU" is a list of unipotent elements from SL_2: [[1,0,x,1], [1,x,0,1]].

Output:

- The image of the element "gU" in our black box group.

`SL2KtoBBGv13BBF(TB,svf,gU)`

Input:

- "TB" is the output of "ToolBoxSL2".

- "svf" is the output of "SharpVsFlatKSD".

- "gU" is a list of unipotent elements from SL_2: [[1,x,0,1], [1,0,x,1], [1,x,0,1]].

Output:

- The image of the element "gU" in our black box group.

SL2KtoBBGv23BBF(TB,svf,gU)

Input:

- "TB" is the output of "ToolBoxSL2".

- "svf" is the output of "SharpVsFlatKSD".

- "gU" is a list of unipotent elements from SL_2: [[1,0,x,1], [1,x,0,1], [1,0,x,1]].

Output:

- The image of the element "gU" in our black box group.

SL2KtoBBGv3BBF(TB,svf,gU)

Input:

- "TB" is the output of "ToolBoxSL2".

- "svf" is the output of "SharpVsFlatKSD".

- "gU" is a list of unipotent elements from SL_2: [[1,0,x,1], [1,x,0,1], [1,0,x,1],[1,x,0,1]].

Output:

- The image of the element "gU" in our black box group.

FieldElementBBF(TB,a)

Input:

- "TB" is the output of "ToolBoxSL2".

- "a" is a non-negative integer.

Output:

- The black box field element representing the integer "a".

SL2ptoSL2(TB,g)

Input:

- "TB" is the output of "ToolBoxSL2".

- "g" is an element from $\mathrm{SL}(2,p)$ where p is a prime number. We assume that "g" is given in GAP format.

Output:

- The image of the element "g" in the group $\mathrm{SL}_2(\mathsf{K})$ where K is our black box field.

SL2ptoBBG(TB,svf,g)

Input:

- "TB" is the output of "ToolBoxSL2".

- "svf" is the output of "SharpVsFlatKSD".

- "g" is an element from a group SL_2 over a prime field produced in GAP format.

Output:

- The image of "gK" in our black box group.