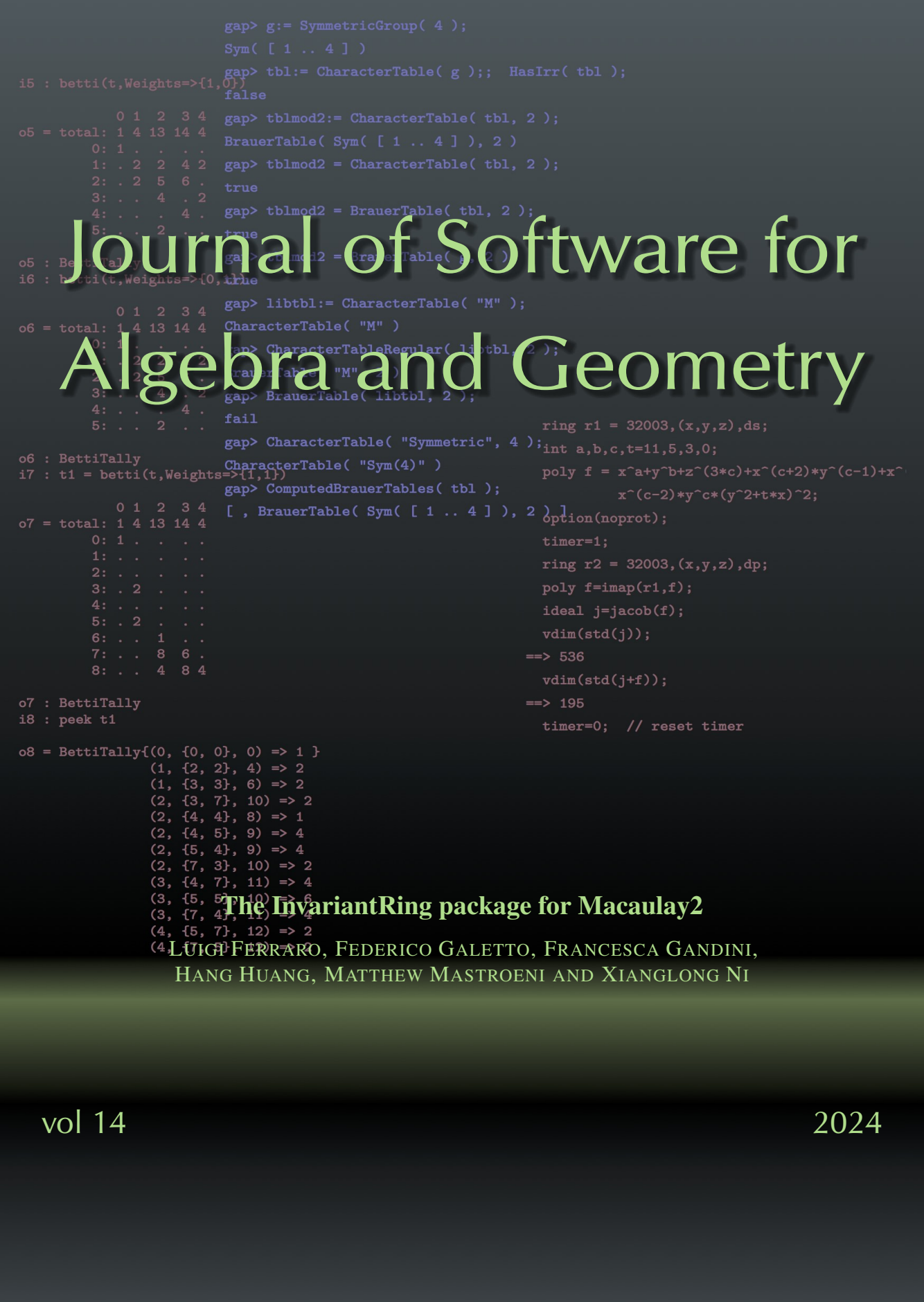

Journal of Software for Algebra and Geometry

The InvariantRing package for Macaulay2

Luigi Ferraro, Federico Galetto, Francesca Gandini, Hang Huang, Matthew Mastroeni and Xianglong Ni

vol 14 2024

JSAG **14** (2024), 5–11
https://doi.org/10.2140/jsag.2024.14.5

The Journal of Software for
Algebra and Geometry

The InvariantRing package for Macaulay2

LUIGI FERRARO, FEDERICO GALETTO, FRANCESCA GANDINI,
HANG HUANG, MATTHEW MASTROENI AND XIANGLONG NI

ABSTRACT: We describe a significant update to the existing `InvariantRing` package for *Macaulay*2. In addition to expanding and improving the methods of the existing package for actions of finite groups, the updated package adds functionality for computing invariants of diagonal actions of tori and finite abelian groups, as well as invariants of arbitrary linearly reductive group actions. The implementation of the package has been completely overhauled with the aim of serving as a unified resource for invariant theory computations in *Macaulay*2.

1. INTRODUCTION. Let G be a group acting linearly on an n-dimensional vector space V over a field $\mathbb{K}$ via $\boldsymbol{v} \mapsto g \cdot \boldsymbol{v}$ for $g \in G$ and $\boldsymbol{v} \in V$. The action of G on V induces an action of G on the polynomial ring $\mathbb{K}[V] = \mathbb{K}[x_1, \ldots, x_n]$ by $(g \cdot f)(\boldsymbol{x}) = f(g^{-1} \cdot \boldsymbol{x})$. By a classical result of Hilbert (see [3, Theorem 2.2.10]), the subring of invariant polynomials

$$\mathbb{K}[V]^G = \{f \in \mathbb{K}[V] \mid g \cdot f = f,\ \forall g \in G\}$$

is always a finitely generated $\mathbb{K}$-algebra provided that G is linearly reductive and V is a rational representation of G.[1] A linearly reductive group is a group that can be identified with a Zariski-closed subgroup of some general linear group $\mathrm{GL}_n = \mathrm{GL}_n(\mathbb{K})$ and has good representation-theoretic properties while still being general enough to encompass all finite groups, all tori $(\mathbb{K}^\times)^r$, and all semisimple Lie groups (at least if $\mathrm{char}(\mathbb{K}) = 0$). Since Hilbert's proof of the finite generation of such invariant rings is famously nonconstructive, determining a minimal set of generating invariants for a specific linearly reductive group action remains a central question in invariant theory. This article describes a significant update to the package `InvariantRing` for *Macaulay*2 [7], originally created by Hawes [5], implementing several algorithms to compute generators for $\mathbb{K}[V]^G$ for various types of group actions.

2. TYPES OF GROUP ACTIONS. The updated version of the `InvariantRing` package introduces the `GroupAction` type and the specialized subtypes `FiniteGroupAction`, `DiagonalAction`, and `LinearlyReductiveAction` as a unified approach to creating group actions in *Macaulay*2. Each type has its own constructor and methods for computing invariants.

MSC2020: 13-04, 13A50, 13P25.
Keywords: Macaulay2, invariants, group actions.
`InvariantRing` version 2.0

[1] However, this is not the case in general by Nagata's celebrated solution to Hilbert's fourteenth problem [8].

2.1. ***Finite group actions.*** Creating a group action of type `FiniteGroupAction` requires a polynomial ring $R = \mathbb{K}[x_1, \ldots, x_n]$ over a field and a list of $n \times n$ matrices generating a finite subgroup of GL_n. The example below illustrates the construction of the natural action of the alternating group A_4 on a polynomial ring in four variables.

```
i2 : R = QQ[x_1..x_4];

i3 : L = {permutationMatrix [2,3,1,4], permutationMatrix [2,1,4,3]};

i4 : A4 = finiteAction(L, R)

o4 = R <- {| 0 0 1 0 |, | 0 1 0 0 |}
          | 1 0 0 0 |  | 1 0 0 0 |
          | 0 1 0 0 |  | 0 0 0 1 |
          | 0 0 0 1 |  | 0 0 1 0 |

o4 : FiniteGroupAction
```

A minimal set of generators for the ring of invariants (as an algebra) can be computed using the method `invariants`.

```
i5 : invariants A4

                                2    2    2    2   3    3    3    3   4
o5 = {x  + x  + x  + x , x  + x  + x  + x , x  + x  + x  + x , x
       1    2    3    4   1    2    3    4   1    2    3    4   1
     ------------------------------------------------------------------
       4    4    4   3 2        3 2    2   3    2 3      3 2
     + x  + x  + x , x x x  + x x x  + x x x  + x x x  + x x x  +
       2    3    4   1 2 3    1 2 3    1 2 3    1 2 4    1 3 4
     ------------------------------------------------------------------
      2 3      3   2    3   2      3 2      2 3    2   3      2 3
     x x x  + x x x  + x x x  + x x x  + x x x  + x x x  + x x x }
      2 3 4    1 2 4    2 3 4    1 3 4    1 2 4    1 3 4    2 3 4

o5 : List
```

By default, the invariants of a finite group action are now computed using an implementation of King's algorithm [3, Algorithm 3.8.2] based on the Reynolds operator. An alternative algorithm based on linear algebra [3, §3.1.1], which can be faster for large groups, is also available through the optional argument `Strategy=>"LinearAlgebra"`.

The methods `molienSeries`, `primaryInvariants`, and `secondaryInvariants` from version 1.1.0 of the package are still available. The method previously known as `invariants`, which computes primary and secondary invariants at once, has been renamed `hironakaDecomposition`. We note that a generating set for the ring of invariants can be obtained from a list of primary and secondary invariants, although this is typically more time consuming. For example, on the same machine, the previous computation of invariants for A_4 took 0.49 seconds using King's algorithm, 0.05 seconds using linear algebra, and 16.49 seconds by computing a Hironaka decomposition.

2.2. ***Diagonal actions.*** A diagonal action over an algebraically closed field $\mathbb{K}$ is an action by a group $G = T \times A$, that is, the product of a torus T and a finite abelian group A. If $T = (\mathbb{K}^\times)^r$ and $A = \mathbb{Z}/d_1\mathbb{Z} \times \cdots \times \mathbb{Z}/d_s\mathbb{Z}$, the action of G on the polynomial ring $R = \mathbb{K}[x_1, \ldots, x_n]$ is *diagonal* if there is an $n \times (r+s)$ matrix of integers $W = (w_{i,j})$, called the *weight matrix*, such that

$$t \cdot x_j = t_1^{w_{1,j}} \cdots t_r^{w_{r,j}} x_j$$

for all j and all $t = (t_1, \ldots, t_r) \in T$, and there is a primitive d_i-th root of unity ζ_i such that the generator u_i of the cyclic abelian factor $\mathbb{Z}/d_i\mathbb{Z}$ acts by

$$u_i \cdot x_j = \zeta_i^{w_{r+i,j}} x_j$$

for all i, j. Because a diagonal action preserves the natural $\mathbb{Z}^n$-grading of R, a polynomial is easily seen to be invariant under a diagonal action if and only if each of its terms is invariants. Thus, we can choose a set of invariant monomials as a minimal set of generating invariants.

Creating a group action of type `DiagonalAction` requires a polynomial ring R, a weight matrix W, and list of positive integers defining the orders of the cyclic factors of the group G. We illustrate this below for a torus action over $\mathbb{F}_9$.

```
i2 : R = (GF 9)[x_1..x_4];

i3 : W = matrix {{5, -3, -1, 4}, {-3, 1, 1, 5}, {0, -4, 2, 6}};

                3       4
o3 : Matrix ZZ  <--- ZZ
i4 : T = diagonalAction(W, {}, R)

                    *  3
o4 = R <- ((GF 9)[x_1 ])  via

     | 5  -3 -1 4 |
     | -3 1  1  5 |
     | 0  -4 2  6 |

o4 : DiagonalAction

i5 : invariants T

             2
o5 = {x x x }
      1 2 3

o5 : List
```

Because a diagonal action involving finite cyclic factors requires the existence of roots of unity for the action to be defined, the invariants computed for a diagonal action should *always* be understood to be the minimal generating invariant monomials over an appropriate infinite extension of the coefficient field. Over such an extension field, the monomial

$$\boldsymbol{x}^{\boldsymbol{a}} = x_1^{a_1} \cdots x_n^{a_n}$$

is invariant if and only if the i-th coordinate of the vector $W\boldsymbol{a}$ is zero for $1 \leq i \leq r$ and the $(r+i)$-th coordinate is zero modulo d_i for $1 \leq i \leq s$. As a result, finding a minimal set of invariant monomials for a diagonal action reduces to a suitable problem in polyhedral geometry. Using the fact that $R^G = (R^A)^T$, we have implemented a recent algorithm of Gandini [4] for first computing invariants of the finite abelian part of a diagonal action; we then use a modified version of [3, Algorithm 4.3.1] to find the torus-invariant monomials in R^A.

As noted above, some form of polyhedral geometry computation is unavoidable when computing diagonal action invariants. The current implementation relies on one of the *Macaulay*2 packages `Polyhedra` [1] or `Normaliz` [6] to handle this part of the computation. The latter package is simply an interface to the stand-alone Normaliz program [2] written in C++. As a program specializing in polyhedral geometry,

Normaliz already has the ability to compute the invariants of a diagonal action. Although Normaliz can have a significant speed advantage on larger examples since it is written in C++ (for example, the above computation on the same machine takes about 25.60 seconds using the Derksen–Gandini algorithm versus 0.01 seconds by calling Normaliz directly), we believe there is value in having an algorithm for computing diagonal invariants implemented in *Macaulay*2 that is independent of any particular external software.

In addition, since *Macaulay*2 has the ability to perform computations over any finite field, it is possible to define diagonal actions (as in the above example) where the action makes sense over the coefficient field specified by the user rather than a suitable infinite extension. In this case, our algorithm includes the ability to compute a minimal set of generating monomials *literally* over the given coefficient field through the option `UseCoefficientRing`, a feature that Normaliz currently lacks. Computing invariants literally over $\mathbb{F}_9$ in the above example, we find the additional invariants:

```
i6 : invariants(T, UseCoefficientRing => true)

           2   8   8   4 4   8   2 6   4 4   4 4   6 2   8
o6 = {x x x , x , x , x x , x , x x , x x , x x , x x , x }
       1 2 3   4   3   2 3   2   1 2   1 3   1 2   1 2   1

o6 : List
```

We note that the above computation took only 0.53 seconds on the same machine as the previous calculation.

2.3. ***Linearly reductive actions.*** In the case of a group action by an arbitrary linearly reductive group G, an action of type `LinearlyReductiveAction` is constructed from the data of an ideal I in an ambient polynomial ring S defining the linearly reductive group G as a Zariski-closed subset of some $\mathbb{K}^m$, a polynomial ring $R = \mathbb{K}[x_1, \ldots, x_n]$ on which the group acts, and an $n \times n$ matrix of polynomials in S defining the action of G on R.

The example below illustrates how to set up the classic action of SL_2 on the coefficients of binary quadrics $ax^2 + bxy + cy^2$ in the ring $\mathbb{K}[x, y]$ where $\mathrm{char}(\mathbb{K}) = 0$. We begin by constructing the defining data for the group SL_2.

```
i2 : S = QQ[z_(1,1)..z_(2,2)];

i3 : I = ideal(z_(1,1)*z_(2,2) - z_(1,2)*z_(2,1) - 1)

o3 = ideal(- z     z     + z     z     - 1)
             1,2   2,1     1,1   2,2

o3 : Ideal of S
```

There is a natural action of SL_2 on $\mathbb{K}[x, y]$ by linear changes of coordinates. We construct the matrix representing the restriction of this action to the space of quadrics $\mathbb{K}[x, y]_2$ with respect to the monomial basis x^2, xy, y^2.

```
i4 : A = S[x,y];

i5 : M = (map(S,A)) last coefficients sub(basis(2,A),
         {x => z_(1,1)*x+z_(1,2)*y, y => z_(2,1)*x+z_(2,2)*y});

                3       3
o5 : Matrix S  <--- S
```

Viewing $R = \mathbb{K}[a,b,c]$ as the dual polynomial ring of coefficients of the quadrics in $\mathbb{K}[x,y]_2$, the transpose of this matrix represents the induced SL_2-action on R.

```
i6 : R = QQ[a,b,c];

i7 : L = linearlyReductiveAction(I, transpose M, R)

o7 = R <- S/ideal(- z   z    + z   z    - 1) via
                    1,2 2,1    1,1 2,2

     | z_(1,1)^2      2z_(1,1)z_(1,2)               z_(1,2)^2      |
     | z_(1,1)z_(2,1) z_(1,2)z_(2,1)+z_(1,1)z_(2,2) z_(1,2)z_(2,2) |
     | z_(2,1)^2      2z_(2,1)z_(2,2)               z_(2,2)^2      |

o7 : LinearlyReductiveAction
```

The *Hilbert ideal* of the action is the ideal of the polynomial ring R generated by all nonconstant homogeneous invariant polynomials. A minimal set of invariant generators for the Hilbert ideal forms a minimal set of algebra generators for the ring of invariants. The package uses an implementation of [3, Algorithm 4.2.8] to compute a set of generators of the Hilbert ideal.

```
i8 : hilbertIdeal L
                  2
o8 = ideal(b  - 4a*c)

o8 : Ideal of R
```

In this case, the generator is none other than the discriminant, and it is already invariant. Typically, this is not the case, and so, the final step is to find *invariant* generators of the Hilbert ideal. As the Reynolds operator is not yet implemented for a general linearly reductive group (see last section), the package uses an alternative approach [3, Algorithm 4.5.1] to find a vector space basis of the invariants degree-by-degree.

3. RINGS OF INVARIANTS. Finally, the package also introduces the `RingOfInvariants` type as a container for all ring-theoretic information about a given group action. Rings of invariants can be computed by calling the method `invariantRing` on any type of group action or by using the natural superscript notation. We illustrate the latter for the finite group action of A_4 on a polynomial ring R defined in Section 2.1.

```
i6 : S=R^A4

o6 =                                2    2    2    2   3    3    3    3
      QQ[x  + x  + x  + x , x  + x  + x  + x , x  + x  + x  + x ,
          1    2    3    4   1    2    3    4   1    2    3    4
     ------------------------------------------------------------
      4    4    4    4   3 2        3 2    2   3    2 3
     x  + x  + x  + x , x x x  + x x x  + x x x  + x x x  +
      1    2    3    4   1 2 3    1 2 3    1 2 3    1 2 4
     ------------------------------------------------------------
      3 2      2 3      3   2    3   2      3 2      2 3
     x x x  + x x x  + x x x  + x x x  + x x x  + x x x  +
      1 3 4    2 3 4    1 2 4    2 3 4    1 3 4    1 2 4
     ------------------------------------------------------------
      2   3      2 3
     x x x  + x x x ]
      1 3 4    2 3 4

o6 : RingOfInvariants
```

The `RingOfInvariants` type has access to methods such as `definingIdeal`, which gives a presentation of the ring of invariants as an affine algebra, and `hilbertSeries`. Unlike the Hilbert series of a typical quotient ring, the Hilbert series of a ring of invariants of a finite group action is presented in a

way that emphasizes the degrees of the primary invariants, which generate a polynomial subring of the ring on invariants over which it is finitely generated as a module.

```
i7 : hilbertSeries(S,Reduce=>true)

                          6
                    1 + T
o7 = --------------------------------
                4       3       2
     (1 - T )(1 - T )(1 - T )(1 - T)

o7 : Expression of class Divide
```

4. Closing remarks.

4.1. ***Performance considerations.*** The methods in this package rely on very different algorithms and so differ in performance. Generally speaking, a diagonal action computation will be much more efficient than one calling on methods for linearly reductive group actions. For reference, the algorithm for invariants of diagonal actions could perform computations with the torus in $\mathrm{GL}_3 \times \mathrm{GL}_3 \times \mathrm{GL}_3$ acting on $\mathbb{C}^3 \otimes \mathbb{C}^3 \otimes \mathbb{C}^3$ (27 variables) in less than three hours, but Macaulay2 ran out of memory when looking at $\mathbb{C}^4 \otimes \mathbb{C}^4 \otimes \mathbb{C}^4$ (64 variables). On the other hand, for the computation of invariants of binary forms of degree n, the computation was nearly instant for n up to 4. However, for $n = 5$, Macaulay2 ran out of memory after a few hours. This is probably because for n up to 4 the invariants are generated in degree up to 3 or 4, but there is an invariant of degree 18 for $n = 5$, so the complexity of the computation increases significantly.

4.2. ***Future directions.*** The major drawback of the function `hilbertIdeal` is that, even though its output generates all invariants over the polynomial ring, the generators themselves do not need to be invariant. A set of algebra generators for the ring of invariants can then be obtained by applying the Reynolds operator of a linearly reductive group to the generators of the Hilbert ideal. Currently, the method `reynoldsOperator` implements the Reynolds operator for finite groups and diagonal actions. However, the Reynolds operator has not yet been implemented for infinite linearly reductive groups because its definition requires building the Lie algebra structure associated to the group. One special exception where it is possible to explicitly construct the Reynolds operator is the case of GL_n acting on a vector space V. In this case, Cayley's Omega Process [3, §4.5.3] constructs the Reynolds operator in terms of the multiplication map on coordinate rings $\mu^* : \mathbb{K}[V] \to \mathbb{K}[V] \otimes_{\mathbb{K}} \mathbb{K}[\mathrm{GL}_n]$ and all partial derivatives on the coordinate ring $\mathbb{K}[\mathrm{GL}_n]$. We plan to implement Cayley's Omega Process in a future update. More generally, implementing Reynolds operators for all linearly reductive groups is a long-term goal for the development of the `InvariantRing` package.

Acknowledgements. Much of the work on the updated version of this package was completed as part of the virtual *Macaulay*2 workshop in May 2020, originally to be held at Cleveland State University, which was partially supported by NSF award DMS-2003883. We thank Harm Derksen and Visu Makam for their helpful suggestions and feedback on the package.

Supplement. The online supplement contains version 2.0 of `InvariantRing`.

REFERENCES.

[1] R. Birkner, "Polyhedra: a package for computations with convex polyhedral objects", *J. Softw. Algebra Geom.* **1** (2009), 11–15. MR Zbl

[2] W. Bruns, B. Ichim, T. Römer, R. Sieg, and C. Söger, "Normaliz: algorithms for rational cones and affine monoids", available at https://www.normaliz.uni-osnabrueck.de. Zbl

[3] H. Derksen and G. Kemper, *Computational invariant theory*, Encyclopaedia of Mathematical Sciences **130**, Springer, Heidelberg, 2015. MR Zbl

[4] F. Gandini, *Ideals of subspace arrangements*, 2019. Zbl

[5] T. Hawes, "Computing the invariant ring of a finite group", *J. Softw. Algebra Geom.* **5** (2013), 15–19. MR Zbl

[6] G. Kaempf and C. Söger, "Normaliz, an interface to Normaliz in *Macaulay2*", available at https://github.com/Macaulay2/M2/tree/master/M2/Macaulay2/packages. Version 2. Zbl

[7] D. R. Grayson and M. E. Stillman, "Macaulay2, a software system for research in algebraic geometry", available at https://faculty.math.illinois.edu/Macaulay2/. Zbl

[8] M. Nagata, "On the fourteenth problem of Hilbert", pp. 459–462 in *Proc. Internat. Congress Math.* 1958, edited by J. A. Todd, Cambridge Univ. Press, New York, 1960. MR Zbl

RECEIVED: 16 Dec 2020 REVISED: 26 Jul 2022 ACCEPTED: 14 Sep 2023

LUIGI FERRARO:

luigi.ferraro@utrgv.edu

School of Mathematical and Statistical Sciences, University of Texas Rio Grande Valley, Edinburg, TX, United States

FEDERICO GALETTO:

f.galetto@csuohio.edu

Department of Mathematics and Statistics, Cleveland State University, Cleveland, OH, United States

FRANCESCA GANDINI:

fra.gandi.phd@gmail.com

Department of Mathematics, Statistics, and Computer Science, St. Olaf College, Northfield, MN, United States

HANG HUANG:

huanghang1109@gmail.com

Department of Mathematics, Texas A&M University, College Station, TX, United States

MATTHEW MASTROENI:

mmastro@iastate.edu

Iowa State University, Ames, IA, United States

XIANGLONG NI:

xlni@berkeley.edu

Department of Mathematics, University of California Berkeley, Berkeley, CA, United States