An algorithm for computing invariants of linear actions of algebraic groups up to a given degree

Thomas Bayer*

Institut für Informatik (I14), Technische Universität München, 80290 Munich, Germany

Received 31 October 2000; accepted 26 November 2002

Abstract

We propose an algorithm for computing invariant rings of algebraic groups which act linearly on affine space, provided that degree bounds for the generators are known. The groups need not be finite nor reductive, in particular, the algorithm does not use a Reynolds operator. If an invariant ring is not finitely generated the algorithm can be used to compute invariants up to a given degree.

© 2003 Elsevier Science Ltd. All rights reserved.

1. Introduction

There are several efficient algorithms for computing invariant rings of finite matrix groups of linear actions of reductive algebraic groups and (possibly nonlinear) actions of $G_a$ on affine space. For finite groups Kemper provided efficient algorithms, cf. Kemper (1996, 1999), and Kemper and Steel (1999), together with an implementation in Maple and the Magma computer algebra system (cf. Bosma et al., 1997). Other approaches can be found, e.g. in Bayer (1998) and Decker et al. (1998) (implemented in the SINGULAR 2.0 library finvar.lib, cf. Heydtmann, 2001), or Sturmfels (1993).

The computation of invariant rings of compact Lie groups has been investigated in Gatermann (2000) and invariant rings of reductive groups (both finite and infinite) can be computed by Derksen’s algorithm, cf. Derksen (1999) (implemented in the SINGULAR 2.0 library rinvar.lib) and, in positive characteristic, by the algorithm given in Kemper (in press). For invariants of $G_a$-actions we refer, e.g. to Maubach (2000) and to the SINGULAR 2.0 library ainvar.lib (cf. Pfister and Greuel, 2001). To the best of our knowledge there are no specialized algorithms for unipotent groups.

* Tel.: +49-89-289-17700; fax: +49-89-17707.
E-mail address: bayert@in.tum.de (T. Bayer).
URL: http://www14.in.tum.de/personen/bayert.
In general, the approaches for compact Lie groups and reductive groups require the use of the Reynolds operator to obtain algebra generators, which seems to be particularly difficult for infinite groups. For the computation of the Reynolds operator for compact Lie groups we refer to Gatermann (2000) and for semi-simple groups we refer to Section 4.5 in Derksen and Kemper (2002), which also contains a description of Cayley’s $\Omega$ process (which is an alternative method for computing the Reynolds operator for the groups $GL_n$ and $SL_n$).

We propose an algorithm which computes the invariant ring of an arbitrary algebraic group which acts linearly on affine space without using the Reynolds operator, provided that a degree bound for the generators is known and all variables of the coordinate ring have weight $>0$. In particular, the algorithm can handle unipotent groups which play an important role in the construction of moduli spaces for singularities, cf. Greuel and Pfister (1993). For degree bounds for finite, respectively linearly reductive algebraic groups we refer, e.g. to Sections 3.9 and 4.7 and the references in Derksen and Kemper (2002). If the ring is not finitely generated, as might happen if the group is not reductive (cf. Nagata, 1959), the algorithm can be used to compute invariants up to a given degree.

2. Invariant rings

Let $K$ be a field and $G$ be an algebraic group defined by the radical ideal $I_G \subseteq K[s_1, s_2, \ldots, s_m]$. The algebraic group action of $G$ on the affine space $K^n$ is given on the ring level by

$$
\Psi : K[t_1, t_2, \ldots, t_n] \to K[s_1, s_2, \ldots, s_m] / I_G \otimes K[t_1, t_2, \ldots, t_n],
$$

$$
t_i \mapsto \psi_i(s_1, s_2, \ldots, s_m, t_1, t_2, \ldots, t_n)
$$

where $\psi_1, \psi_2, \ldots, \psi_n \in K[s_1, s_2, \ldots, s_m, t_1, t_2, \ldots, t_n]$. For $\sigma \in G$ and $t \in K^n$ the group action is given by $\sigma \cdot f(t) := \Psi(f)(\sigma, t)$. We consider polynomials as functions by allowing them to take values in the algebraic closure of $K$ if the field $K$ is finite. A polynomial $f \in K[t_1, t_2, \ldots, t_n]$ is invariant w.r.t. $G$ if $\sigma \cdot f(t_1, t_2, \ldots, t_n) = f(t_1, t_2, \ldots, t_n)$ for all $\sigma \in G$. The invariant ring $K[t_1, t_2, \ldots, t_n]^G$ of $G$ is the subring of $K[t_1, t_2, \ldots, t_n]$ containing all polynomials invariant under $G$. Note that the invariant ring $K[t_1, t_2, \ldots, t_n]^G$ is isomorphic to

$$
K[t_1, t_2, \ldots, t_n]^G \cong K[\psi_1, \psi_2, \ldots, \psi_n] \cap K[t_1, t_2, \ldots, t_n]
$$

$$
f \mapsto [f],
$$

where the rings on the right-hand side are considered to be subrings of $K[s_1, s_2, \ldots, s_m, t_1, t_2, \ldots, t_n]/I_G$. We obtain generators for the invariant ring of $G$ by computing generators for the intersection on the right-hand side.

Testing if a polynomial is invariant w.r.t. the action of $G$, given by $\psi_1, \psi_2, \ldots, \psi_n$, can be done as follows (cf. Vasconcelos, 1998, Proposition 7.4.3). Note that $I_G$ is a radical ideal.
Lemma 2.1. A polynomial \( f \in K[t_1, t_2, \ldots, t_n] \) is invariant w.r.t. \( G \) iff
\[
f - \Psi(f) \in (I_G) \subset K[s_1, s_2, \ldots, s_m, t_1, t_2, \ldots, t_n].
\]

Proof. For \( f \in K[t_1, t_2, \ldots, t_n]^G \) the polynomial \( f - \Psi(f) \) vanishes on the variety \( G \times K^n \) by assumption. Hence \( f - \Psi(f) \) is contained in the ideal of \( G \times K^n \) which is precisely the ideal \( I_G \). Conversely, \( f - \Psi(f) \in I_G \) implies that \( f - \Psi(f) \) vanishes identically on \( \sigma \times K^n \) for every \( \sigma \in G \), i.e. \( f(t_1, t_2, \ldots, t_n) = \Psi(f)(\sigma, t_1, t_2, \ldots, t_n) = \sigma \cdot f(t_1, t_2, \ldots, t_n) \). Therefore \( f \) is invariant w.r.t. \( G \). \( \square \)

3. The algorithm

In this section let \( I_G \subset K[s_1, s_2, \ldots, s_m] \) be a radical ideal defining an algebraic group \( G \). We make use of homogenization of polynomials and ideals w.r.t. a new variable \( X \), which we denote by \( ^h \) and refer, e.g. to Vasconcelos (1998) for computational properties. We assume that the polynomials \( \psi_1, \psi_2, \ldots, \psi_n \in K[s_1, s_2, \ldots, s_m, t_1, t_2, \ldots, t_n] \) which define a linear action of \( G \) on \( K^n \), are homogeneous of the same degree. This can be achieved by homogenizing \( \psi_1, \psi_2, \ldots, \psi_n \) w.r.t. a new variable \( s \) (not \( X \)) and adding the equation \( s - 1 \in I_G \).

The algorithm is based on the following observation.

Proposition 3.1. Let \( \psi_1, \psi_2, \ldots, \psi_n \in K[s_1, s_2, \ldots, s_m, t_1, t_2, \ldots, t_n] \) be homogeneous polynomials of degree \( \delta \) defining a linear action of the algebraic group \( G \) on \( K^n \) and let
\[
I = \langle \psi_1, \psi_2^e, \ldots, \psi_n^e : [\sigma] = d \rangle \cup I_G \subset K[s_1, s_2, \ldots, s_m, t_1, t_2, \ldots, t_n, X].
\]
If \( GB = \{ f_1, f_2, \ldots, f_k \} \) is a Gröbner basis of the ideal \( J = I \cap K[t_1, t_2, \ldots, t_n, X] \) we have, as \( K \)-vectorspaces,
\[
(f_i(t_1, t_2, \ldots, t_n, 1) : 1 \leq i \leq k, \ \deg(f_i) = d \cdot \delta)_K = K[t_1, t_2, \ldots, t_n]^G.
\]

Proof. If \( \deg(f_i) < d \delta \) then \( f_i \in I_G \) and therefore \( f_i \notin I \cap K[t_1, t_2, \ldots, t_n, X] \), a contradiction. Hence \( \deg(f_i) \geq d \delta \) for \( 1 \leq i \leq k \). Since \( GB \) is a Gröbner basis and \( \deg(f_i) \geq d \delta \) the \( K \)-vectorspace \( (f_i(t_1, t_2, \ldots, t_n, 1) : \deg(f_i) = d \delta)_K \) is the dehomogenization of \( I_G \). Let \( f \in K[t_1, t_2, \ldots, t_n]^G \) be a homogeneous invariant of degree \( d \delta \) and note that \( \Psi(f) \in I \). By Lemma 2.1 \( f - \Psi(f) \in (I_G) \) which implies \( f X^{\delta - (\delta - 1)} - \Psi(f) \in (I_G)^d \), and therefore \( f X^{\delta - (\delta - 1)} \in J \). In particular, \( f X^{\delta - (\delta - 1)} \in J \) is of degree \( d \delta \) as required.

Now assume \( \deg(f_i) = d \delta \) and note that the dehomogenization \( f_i^1 = f_i(t_1, t_2, \ldots, t_n, 1) \) is a homogeneous polynomial of degree \( d \). The condition \( f_i \in I \) implies the existence of \( p \in K[t_1, t_2, \ldots, t_n] \) and \( g \in (I_G) \) s.t. \( p \) is homogeneous of degree \( d \) and \( f_1 = \Psi(p) + g \). Therefore \( f_1 - \Psi(p) \in (I_G) \) and \( f_1(t, 1) = f_1^1(t) = \Psi(p)(\sigma, t) \) for all \( \sigma \in G \) and \( t \in K^n \). In particular,
\[
f_1^1(t) = \Psi(p)(id, t) = id \cdot p(t) = p(t)
\]
so \( f_1^1 = p \) and the claim follows from Lemma 2.1. \( \square \)
In the \( j \)-th iteration the algorithm computes a \( K \)-basis \([f_1], [f_2], \ldots, [f_r] \) of the degree \( d_j \) part of the intersection of \( K[\psi_1, \psi_2, \ldots, \psi_n] \cap K[t_1, t_2, \ldots, t_n] \) as subrings of \( K[s_1, s_2, \ldots, s_m, t_1, t_2, \ldots, t_n] / I_G^j \), where \( f_i \in K[t_1, t_2, \ldots, t_n]^{\delta} \) and \( d_j \) is some degree.

**Algorithm 3.1.** **INVARAINTS** \((I_G, (\psi_1, \psi_2, \ldots, \psi_n), \text{degrees})\)

In: radical ideal \( I_G \subset K[s_1, s_2, \ldots, s_m] \) of an algebraic group \( G \) and polynomials \( \psi_1, \psi_2, \ldots, \psi_n \in K[s_1, s_2, \ldots, s_m, t_1, t_2, \ldots, t_n] \) defining a linear action of \( G \) on \( K^n \), a list of degrees of positive integers, \(<\text{a lex. order s.t.}\ s_\rho > t_\tau > X.\)

Out: \( K \)-vectorspace basis of \([f \in K[t_1, t_2, \ldots, t_n]^G_d : d \in \text{degrees}]\).

begin
\{\psi_1', \psi_2', \ldots, \psi_n':=\text{homogenization of }\{\psi_1, \psi_2, \ldots, \psi_n\}\ \text{w.r.t.}'s\ s.t. \deg(\psi_1) = \deg(\psi_j).\]
\(\delta:=\deg(\psi_1');\)
\(I_G:=I_G \cup \{s - 1\};\)
\(I_G^h:=\text{homogenization of }I_G\ \text{w.r.t. new variable }X.\)
\(B:=[ ];\)
for \(j:=1\) to \{|\text{degrees}|\} do
\(I:=\langle\psi_1'^\alpha_1, \psi_2'^\alpha_2 \ldots, \psi_n'^\alpha_n \mid |\alpha| = \text{degrees }[j] \cup I_G^h\rangle;\)
\(\{f_1, f_2, \ldots, f_k\}:=\text{Gröbner Basis }_<>(I) \cap K[t_1, t_2, \ldots, t_n, X];\)
for \(i:=1\) to \(k\) do
if \(\deg(f_i) = \text{degrees }[j] \cdot \delta\) then \(B:=B \cup \{f_i(t_1, t_2, \ldots, t_n, 1)\};\)
end;
end;
return\(B);\)
end **INVARAINTS**.

**Remark 3.1.**

(a) It suffices to compute the Gröbner basis of \( I \) up to degree \( d\delta \). Suppose that \( f_i \in GB \) is of degree \( \deg(f_i) > d\delta \). Then \( \deg_{[t_1, t_2, \ldots, t_n]}(f_i) > d \) because the action is linear in \( t_1, t_2, \ldots, t_n \) and \( f_i \) does not contribute to the invariants of degree \( d \). Without this restriction, the algorithm seems to be too slow to compete with other algorithms.

(b) As mentioned above, the algorithm computes invariants up to a given degree and depends therefore on good degree bounds. For degree bounds of finite respectively linearly reductive groups we refer to Section 3.9 respectively Section 4.7 of Derksen and Kemper (2002) and the references therein. Note that there are no known degree bounds for reductive groups in positive characteristic (e.g. \( SL_2 \)), although it is known that the invariant ring is finitely generated.

The ideal operations in the algorithm are performed by Gröbner bases computations, cf. Buchberger (1985). For the elimination of variables we refer, e.g. to Vasconcelos (1998).

**Theorem 3.1.** The algorithm **INVARAINTS** is correct.
Fix a lexicographic order where \( s_p > t_r > X \) and let \( j > 0 \). We show that any homogeneous invariant of degree \( d = \text{degrees}[j] \) is contained in the linear span of \( B^{(j)} \), where \( B^{(j)} \) denotes the set \( B \) in the \( j \)th iteration of the for-loop. Let \( GB = \{ f_1, f_2, \ldots, f_r \} \) be the Gröbner basis of \( I \subset K[s_1, s_2, \ldots, s_m, t_1, t_2, \ldots, t_n, X] \) in the \( j \)th iteration s.t. \( GB \cap K[t_1, t_2, \ldots, t_n, X] = \{ f_1, f_2, \ldots, f_k \} =: G B'. \) By elimination theory (cf., e.g. Vasconcelos, 1998, Proposition 2.1.1), \( G B' \) is a Gröbner basis of the ideal \( I \cap K[t_1, t_2, \ldots, t_n, X] \). By Proposition 3.1 we have

\[
\forall f \in K[t_1, t_2, \ldots, t_n, X]^G \quad f = \sum_{i=1}^{k} \lambda_i f_i(t_1, t_2, \ldots, t_n, 1)
\]

for some \( \lambda_1, \lambda_2, \ldots, \lambda_k \in K \). Hence \( f \) is contained in the linear span of \( B^{(j)} \). \( \square \)

In the two examples below we apply the algorithm to finite reductive/non-reductive and infinite reductive/non-reductive groups. The invariant rings of the first example can be computed, e.g. with Kemper’s algorithms, or with the SINGULAR 2.0 library and infinite reductive/non-reductive groups. The ideal \( I G \) of the group \( G \) is given by \( \langle s_1(s_1 - 1)(s_1 - 2) \rangle \subset K[s_1] \) and the action is defined by the two polynomials \( t_1 + s_1 t_2, t_2 \). In the application of the algorithm the action is homogenized w.r.t. \( s \), the equation \( s - 1 \) is added to \( I G \) and \( I G \) is homogenized w.r.t. \( X \). We have \( I G^2_2 = \langle s_1(s_1 - 1)(s_1 - 2), s - X \rangle \) and the new action equals \( s t_1 + s_1 t_2, s t_2 \). The algorithm computes the following invariants (degree bound = 3),

\[
t_2, t_2^2, t_2^3 - t_1 t_2^2
\]

which turn out to be fundamental invariants\(^1\). Note that \( G \) does not admit a Reynolds operator.

---

\(^1\) If no degree bounds are known, this has to proved (e.g. by hand).
We apply the algorithm to an infinite reductive and an infinite non-reductive group.

Example 3.2.

(a) The action of $SL_2(\mathbb{C})$ on $V \oplus V \oplus S^2V$, where $V$ is the usual 2-dimensional representation of $SL_2(\mathbb{C})$ (cf. Example 6.2 of Derksen, 1999). The ideal of $SL_2(\mathbb{C})$ equals $\langle s_1s_4 - s_2s_3 - 1 \rangle \subset \mathbb{C}[s_1, s_2, s_3, s_4]$ and the action is given by the representation

$$
\begin{pmatrix}
  s_1 & s_2 \\
  s_3 & s_4
\end{pmatrix}
\mapsto
\begin{pmatrix}
  s_1 & s_2 \\
  s_3 & s_4
\end{pmatrix}
\oplus
\begin{pmatrix}
  s_1^2 & s_2 \ \\
  s_3 & s_4
\end{pmatrix}
\oplus
\begin{pmatrix}
  2s_1s_2 & s_2s_3 \\
  s_3s_4 & s_4
\end{pmatrix}.
$$

Derksen’s algorithm delivers an ideal basis of the nullcone having degrees 2, 2, 3, 3 and containing the polynomial $t_1t_2t_7 - 2t_2t_5t_6 + t_2t_4t_5$ which is not invariant. By applying \textsc{Invariants} with upper bound 3 we obtain fundamental invariants

$$
\begin{align*}
-t_1t_4 + t_2t_3, & \quad -t_3t_7 + t_6^2, & \quad t_1^3t_7 - 2t_1t_4t_6 + t_2^2t_5, \\
t_1t_3t_7 - t_1t_4t_6 - t_2t_3t_6 + t_2t_4t_5, & \quad t_1^3t_7 - 2t_1t_2t_6 + t_2^2t_5
\end{align*}
$$

having degrees 2, 2, 3, 3.

(b) Consider the linear action of the non-reductive group $G \subset GL_8(\mathbb{C})$,

$$
G = \left\{ \begin{pmatrix}
  B_1 & 0 & 0 & 0 \\
  0 & B_2 & 0 & 0 \\
  0 & 0 & B_3 & 0 \\
  0 & 0 & 0 & B_4
\end{pmatrix} : B_i = \begin{pmatrix}
  1 & b_i \\
  0 & 1
\end{pmatrix}, b_1 + 2b_2 + 3b_3 + 5b_4 = 0 \right\},
$$

on $\mathbb{C}^6$. The ideal of $G$ is $I_G = \langle s_1 + 2s_2 + 3s_3 + 5s_4 \rangle \subset \mathbb{C}[s_1, s_2, s_3]$ and the action is given by $\{\psi_1, \psi_2, \ldots, \psi_8 = \{t_1 + s_1t_2, t_2, t_3 + s_2t_4, t_4, t_5 + s_3t_6, t_6, t_7 + s_4t_8, t_8\}$. A variant of \textsc{Invariants}, where only those elements not contained in $\mathbb{C}[B]$ are added to $B$, yields the invariants $t_2, t_4, t_8, 5t_2t_4t_6t_7 + 3t_2t_4t_5t_6 + 2t_2t_3t_5t_6 - t_1t_4t_6t_8$ of degree $\leq 4$ which are fundamental invariants (cf. footnote 1 on page 5).

4. Performance and limitations

4.1. Performance

We provide running times of the implementation of the algorithm in the computer algebra system \textsc{Singular} 2.0 (cf. Greuel et al., 2001) on a PC (Pentium III 1 GHz, 2 GB) for the groups of the previous section and for the following three additional group actions.

1. The induced action of the octahedron group on $\mathbb{C}^6$ via the Cauchy–Green strain tensor (the order of the group equals 24).
2. $S^4V \oplus S^2V$ where $V$ is the usual representation of $SL_2(\mathbb{C})$.
3. The linear action of $G_a$ on $\mathbb{C}^6$, given by $t_1 + st_2, t_2, t_3 + st_4, t_4, t_5 + st_6, t_6$.

The implementation and the examples can be found at the homepage of the author. For reasons of comparison, we provide additional running times of Derksen’s algorithm.
(cf. Derksen, 1999), the standard linear algebra algorithm\(^2\) and the algorithm \texttt{invariant(...)\_ring} from the SINGULAR 2.0 library \texttt{finvar.lib} (cf. Heydtmann, 2001, for a description we refer to Decker et al., 1998).

Note that the algorithm \texttt{INVAR}\_\texttt{ANTS} and the standard algorithm have been called either with optimal degree bounds or with a list of degrees as mentioned in the examples. By ‘*’ we denote that the Reynolds operator must be applied to the output, by ‘−’ we denote that the algorithm cannot handle the current group, and by ‘x’ we denote the fact that the computation has been aborted due to time/memory constraints (>50h/>2 GB).

We provide running times for \texttt{INVAR}\_\texttt{ANTS} with and without elimination of dependent invariants, denoted by \texttt{INV1}, \texttt{INV2} respectively. By \texttt{DERKSEN}, \texttt{STANDARD} and \texttt{finvar} we denote the running time of Derksen’s algorithm, the standard algorithm and of the algorithm \texttt{invariant(...)\_ring} from \texttt{finvar.lib} respectively. For finite groups (rows 3.1(a), 3.1(b) and 1) we have used the algorithm \texttt{invariant(...)\_basis} from \texttt{finvar.lib} as the standard algorithm. In the column ‘Generators’ is a list of degrees of algebra generators where \(d^k\) means that there are \(k\) generators of degree \(d\). Running times are measured in seconds.

<table>
<thead>
<tr>
<th>Example</th>
<th>Generators</th>
<th>\texttt{INV1}</th>
<th>\texttt{STANDARD}</th>
<th>\texttt{finvar}</th>
<th>\texttt{INV2}</th>
<th>\texttt{DERKSEN}</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1(a)</td>
<td>[8, 12]</td>
<td>0.06</td>
<td>0.06</td>
<td>1.30</td>
<td>0.07</td>
<td>1.20*</td>
</tr>
<tr>
<td>3.1(b)</td>
<td>[1, 3]</td>
<td>0.02</td>
<td>0.02</td>
<td>0.04</td>
<td>0.02</td>
<td>–</td>
</tr>
<tr>
<td>3.2(a)</td>
<td>[2, 3]</td>
<td>0.09</td>
<td>2.00</td>
<td>–</td>
<td>0.09</td>
<td>0.05*</td>
</tr>
<tr>
<td>3.2(b)</td>
<td>[1, 3]</td>
<td>0.59</td>
<td>42.38</td>
<td>–</td>
<td>0.63</td>
<td>–</td>
</tr>
<tr>
<td>1</td>
<td>[1, 2(^2), 3(^3), 4(^2), 5]</td>
<td>12.64</td>
<td>18.39</td>
<td>1360.17</td>
<td>15.84</td>
<td>x</td>
</tr>
<tr>
<td>2</td>
<td>[2(^2), 3(^2), 4, 6]</td>
<td>97.96</td>
<td>x</td>
<td>450.14</td>
<td>41 h 33 m</td>
<td>–</td>
</tr>
<tr>
<td>3</td>
<td>[1, 2]</td>
<td>0.10</td>
<td>1.41</td>
<td>–</td>
<td>0.15</td>
<td>–</td>
</tr>
</tbody>
</table>

**Remark 4.1.**

(a) As already mentioned, the running time of \texttt{INVAR}\_\texttt{ANTS} depends heavily on degree bounds. For groups with no known bound it might be useful to compute invariants of low degree, then ‘guess’ a degree bound and prove it by other methods (as we have done for the groups in 3.1(b), 3.2(b), 2 and 3).

(b) We have not taken into account the algorithms from Gatermann (2000) because they are restricted to representations of compact Lie groups over \(\mathbb{R}\) (and their extension to \(\mathbb{C}\)).

\(^2\) If \(\Psi\) defines the group action \(G \times \mathbb{K}^d \rightarrow \mathbb{K}^d\) on the ring level, \(I_G\) is the defining ideal of \(G\) and \(F\) is the sum of all monomials of degree \(d\) (with parameters as coefficients), the algorithm builds a linear system by comparing coefficients in normalform \((F − \Psi(F), I_G) = 0\). A basis for the solution space can be found by linear algebra methods.
4.2. Limitations

Theoretical Limitations

The algorithm cannot handle nonlinear actions and variables of weight 0. The variables \( t_1, t_2, \ldots, t_n \) must be of weight \( > 0 \) and the polynomials \( \psi_1, \psi_2, \ldots, \psi_n \) defining the group action must be homogeneous of degree 1 in the variables \( t_1, t_2, \ldots, t_n \). E.g. the algorithm cannot handle the nonlinear \( G_a \)-action on \( \mathbb{C}^7 \), given by

\[
(\lambda, (t_1, t_2, \ldots, t_7)) \mapsto (t_1, t_2, t_3, \lambda \cdot t_1 + t_4, \lambda \cdot t_2 + t_5, \lambda \cdot t_3 + t_6, \lambda \cdot t_1^2 t_2^2 t_3^2 + t_7)
\]

where the degree of \( t_1, t_2, t_3 \) equals 0, the degree of \( t_4, t_5, t_6, t_7 \) equals 1. Note that the invariant ring is not finitely generated (cf. A’Campo-Neuen, 1994). To apply the algorithm INVARIANTS, the action must be linearized, which can be done, e.g. by the SINGULAR 2.0 library rinvar.lib (cf. Bayer, 2001).

For \( G_a \)-actions, as in the example above, there are algorithms for computing invariants up to a given degree, cf. Maubach, 2000 or the ainvar.lib library of SINGULAR 2.0 (cf. Pfister and Greuel, 2001).

Practical Limitations

For invariant subrings of \( \mathbb{K}[t_1, t_2, \ldots, t_n] \) where the degree of a (minimal) generator equals \( d \) and \( \binom{n+d-1}{d-1} > 600 \), the computation may become infeasible (e.g. action of \( S_3 \times \mathbb{Z}_3 \times \mathbb{Z}_3 \) on \( \mathbb{C}^4 \), a minimal generator has degree 15, computation aborted after 1 week). The algorithm seems to be better suited for infinite groups.

5. Conclusion

We have provided an algorithm for computing invariants of linear actions of algebraic groups, given equations defining the group and the action, which does not make use of the Reynolds operator (provided it exists). Despite the generality, the algorithm can compete with other algorithms, provided good degree bounds are known. In particular, the algorithm seems to outperform the standard algorithm in the case of infinite groups.

The algorithm is very useful in combination with Derksen’s algorithm, which can be used to compute degree bounds, and then INVARIANTS can be used to compute invariants. Hence one may bypass the (non-trivial) computation of the Reynolds operator for infinite groups.

Acknowledgements

I am very grateful to the anonymous referees for their comments on the algorithm and degree bounds.

References


