# MOP – ALGORITHMIC MODALITY ANALYSIS FOR PARABOLIC GROUP ACTIONS

ULF JÜRGENS AND GERHARD RÖHRLE

Abstract. Let $G$ be a simple algebraic group and $P$ a parabolic subgroup of $G$. The group $P$ acts on the Lie algebra $\mathfrak{p}_u$ of its unipotent radical $P_u$ via the adjoint action. The modality of this action, $\operatorname{mod}(P : \mathfrak{p}_u)$, is the maximal number of parameters upon which a family of $P$-orbits on $\mathfrak{p}_u$ depends. More generally, we also consider the modality of the action of $P$ on an invariant subspace $\mathfrak{n}$ of $\mathfrak{p}_u$, that is $\operatorname{mod}(P : \mathfrak{n})$. In this note we describe an algorithmic procedure, called MOP, which allows one to determine upper bounds for $\operatorname{mod}(P : \mathfrak{n})$.

The classification of the parabolic subgroups $P$ of exceptional groups with a finite number of orbits on $\mathfrak{p}_u$ was recently achieved with the aid of MOP, cf. [12]. In view of the results from [10], this completes the classification of parabolic subgroups of all reductive algebraic groups with this finiteness property.

Besides this result we present other applications of MOP, and illustrate an example.

## 1. Introduction

Throughout, $G$ is a simple algebraic group over an algebraically closed field $K$ and $P$ is a parabolic subgroup of $G$. The group $P$ acts on its unipotent radical $P_u$ via conjugation and on $\mathfrak{p}_u$, the Lie algebra of $P_u$, via the adjoint action. The *modality of the action of $P$ on $\mathfrak{p}_u$*, denoted by $\operatorname{mod}(P : \mathfrak{p}_u)$, is the maximal number of parameters upon which a family of $P$-orbits on $\mathfrak{p}_u$ depends. More generally, we also want to study the modality of the action of $P$ on an invariant linear subspace $\mathfrak{n}$ of $\mathfrak{p}_u$, that is $\operatorname{mod}(P : \mathfrak{n})$. The *modality of $P$* is defined as $\operatorname{mod} P := \operatorname{mod}(P : \mathfrak{p}_u)$; see Section 2 for a precise definition and [16] for some additional references concerning this notion. Observe that $\operatorname{mod}(P : \mathfrak{n})$ is zero precisely when $P$ operates on $\mathfrak{n}$ with a finite number of orbits.

In this paper we describe the modality algorithm MOP (Modality Of Parabolics) which is designed to compute upper bounds for $\operatorname{mod}(P : \mathfrak{n})$. In [16] the general problem was posed to determine each parabolic subgroup $P$ of $G$ with $\operatorname{mod} P = 0$. After the cases for Borel and semisimple rank one parabolic subgroups were classified in [14] and [16], respectively, all modality zero parabolic subgroups of classical groups were classified in [10]. Extending these results, all such parabolic subgroups of the exceptional groups were determined recently with the aid of MOP, cf. [12].

Apart from these results we indicate other applications of MOP. For instance, we determine $\operatorname{mod} P$ for some parabolics in $E_6$ by combining lower bounds for $\operatorname{mod} P$ from [21] with upper bounds obtained by MOP. The algorithm is implemented as a share package in the computer algebra system GAP [8]. MOP generalizes the algorithm outlined in [6] which was designed to analyze the orbit structure of a Borel subgroup $B$ for the adjoint and coadjoint actions on $\mathfrak{b}_u$ and on $\mathfrak{b}_u^*$. MOP only applies in the case $G$ is simply laced. For details on usage and technical aspects the reader should consult the MOP manual [13].

---

The second author gratefully acknowledges partial support of a DFG grant.

2000 *Mathematics Subject Classification.* 20G15, 17B45.

The basic machinery for investigating the modality of parabolic subgroups of reductive groups was introduced in [16]. There are several recent articles related to this subject, such as [5], [10], [11], [12], [15], and [22].

Our general reference for algebraic groups is Borel's book [3] and for information on root systems we refer the reader to Bourbaki [4]. The simple roots in a base of a root system of $G$ are indexed in accordance with [4, Planches I - IX].

## 2. Notation and Preliminaries

Suppose that the connected algebraic group $R$ acts morphically on the algebraic variety $X$. For $x$ in $X$ the $R$-orbit in $X$ through $x$ is denoted by $R \cdot x$. The *modality of the action of $R$ on $X$* is defined as

$$\operatorname{mod}(R : X) := \max_{Z} \min_{z \in Z} \operatorname{codim}_Z R \cdot z,$$

where $Z$ runs through all irreducible $R$-invariant subvarieties of $X$. In case $X$ is an irreducible variety let $K(X)^R$ denote the field of $R$-invariant rational functions on $X$. By a result of Rosenlicht $\min_{x \in X} \operatorname{codim}_X R \cdot x = \operatorname{trdeg} K(X)^R$, for instance, see [17, 2.3]. Therefore, $\operatorname{mod}(R : X)$ measures the maximal number of parameters upon which a family of $R$-orbits on $X$ depends. The modality of the action of $R$ on $X$ is zero precisely when $R$ admits only a finite number of orbits on $X$, see also [17, 5.2].

We denote the Lie algebra of $G$ by $\operatorname{Lie} G$ or by $\mathfrak{g}$; likewise for subgroups. The Lie algebra of $P_u$ is denoted by $\mathfrak{p}_u$. Let $T$ be a maximal torus in $G$ and $\Psi$ the set of roots of $G$ with respect to $T$. Fix a Borel subgroup $B$ of $G$ containing $T$ and let $\Pi$ be the set of simple roots of $\Psi$ defined by $B$, then $\Psi^+ = \Psi(B)$ is the set of positive roots of $G$. We may assume that every parabolic subgroup of $G$ under consideration contains $B$, i.e. is standard. For a subset $J$ of $\Pi$ we denote by $P_J$ the standard parabolic subgroup corresponding to $J$ such that $P_\varnothing = B$. Further, $\ell(P_u)$ denotes the length of the descending central series of $P_u$, that is its class of nilpotency. We denote the Weyl group of some Levi subgroup of $P$ by $W_P$. By saying that $P$ is of a particular type, we mean the Dynkin type of a Levi subgroup of $P$.

A prime dividing one of the structure constants of the Chevalley commutator relations for $G$ is called a *very bad* prime for $G$.

Let $\beta \in \Psi^+$. Write $\beta = \sum_{\alpha \in \Pi} c_\alpha(\beta)\alpha$, with $c_\alpha(\beta) \in \mathbb{Z}_0^+$ for each $\alpha \in \Pi$. For $J$ a subset of $\Pi$, we call $\sum_{\alpha \in \Pi \setminus J} c_\alpha(\beta)$ the *$J$-height* of $\beta$; cf. [1]. For $J = \varnothing$ this is the usual height function.

If $\operatorname{char} K$ is not a very bad prime for $G$ and $P = P_J$, then $\ell(P_u)$ is just the $J$-height of the highest root in $\Psi$, [1, Lem. 4].

Let $H$ be a closed connected subgroup of $G$ normalized by $T$ (that is $H$ is a *regular* subgroup of $G$, cf. [7]); likewise for subalgebras of $\mathfrak{g}$. In that case the root spaces of $\mathfrak{h}$ relative to $T$ are also root spaces of $\mathfrak{g}$ relative to $T$, and the set of roots of $H$ with respect to $T$, $\Psi(H)$, is a subset of $\Psi$. If $\operatorname{char} K$ is not a very bad prime for $G$, then $\Psi(H)$ is closed under addition in $\Psi$. Furthermore, if $H$ is reductive and regular, then $\Psi(H)$ is a semisimple subsystem of $\Psi$. For a root $\alpha$ of $G$ we denote by $U_\alpha$ the corresponding one-parameter unipotent subgroup of $G$. For every root $\alpha$ we choose a generator $x_\alpha$ of the corresponding root space $\operatorname{Lie} U_\alpha = \mathfrak{g}_\alpha$ of $\mathfrak{g}$.

The *support* of a subset $S$ of $\mathfrak{b}_u$, denoted by $\operatorname{supp} S$, is the set of all roots $\alpha$ such that the restriction to $S$ of the projection from $\mathfrak{b}_u$ onto $\mathfrak{g}_\alpha$ is non-trivial.

By a Levi subgroup of a reductive group $G$ we simply mean a Levi subgroup of some parabolic subgroup of $G$.

We require some basic facts concerning the modality of parabolic groups; the first one is elementary (cf. [16, Lem. 4.3], or [20, Lem. 2.8]):

**Lemma 2.1.** *Let $Q \subseteq P$ be parabolic subgroups of $G$. Then* $\operatorname{mod} P \leq \operatorname{mod} Q$.

*Proof.* Since $Q \subseteq P$, we have $\mathfrak{p}_u \subseteq \mathfrak{q}_u$. Any irreducible $P$-invariant subvariety $Z$ of $\mathfrak{p}_u$ is also $Q$-invariant and $\operatorname{codim}_Z P \cdot z \leq \operatorname{codim}_Z Q \cdot z$ for any $z$ in $Z$. Consequently, we get $\operatorname{mod} P \leq \operatorname{mod} Q$, by the definition of modality above. □

We require Theorem 1.2 from [22], see also [15, Thm. 4], [16, Rem. 2.14]:

**Lemma 2.2.** *Suppose that* $\operatorname{char} K$ *is zero or a good prime for $G$. Let $H$ be a closed reductive subgroup of $G$ normalized by $T$. Set $Q := P \cap H$. Then* $\operatorname{mod} Q \leq \operatorname{mod} P$.

*Remark* 2.3. In the special case of Lemma 2.2 when $H$ is a Levi subgroup of $G$ normalized by $T$ or the derived subgroup thereof the statement of Lemma 2.2 is valid without any characteristic restrictions [22, Cor. 3.10].

For $l \in \mathbb{N}_0$ let $\mathfrak{p}_u^{(l)}$ denote the $l$-th term of the lower central series of $\mathfrak{p}_u$.

*Remark* 2.4. If $\operatorname{char} K$ is not a very bad prime for $G$, then $\mathfrak{p}_u^{(l-1)}/\mathfrak{p}_u^{(l)} \cong \oplus \mathfrak{g}_\alpha$, where the sum is taken over all those roots $\alpha$ in $\Psi$ of $J$-height $l \in \mathbb{N}$, see [1, Lem. 4].

Observe that in the context of our algorithm this hypothesis on $\operatorname{char} K$ is always fulfilled, since MOP only applies when $G$ is simply laced and then the structure constants of the commutator relations are $\pm 1$.

We need a special case of a general theorem due to R.W. Richardson [19, Thm. E].

**Lemma 2.5.** *Let $P = LP_u \subset G$ be a parabolic subgroup of $G$. Then $L$ operates on the quotient $\mathfrak{p}_u^{(l)}/\mathfrak{p}_u^{(l+1)}$ with a finite number of orbits for each $l \geq 0$.*

*Remark* 2.6. If $\operatorname{char} K$ is zero or a good prime for $G$, then $\operatorname{mod}(P : P_u) = \operatorname{mod}(P : \mathfrak{p}_u)$ thanks to [22, Thm. 1.3]. Thus we obtain similar results for the action of $P$ on $P_u$.

The statements in Lemmas 2.1 and 2.2 (and the one in Remark 2.6) also apply if we replace $\mathfrak{p}_u$ by some $P$-invariant linear subspace $\mathfrak{n}$ of $\mathfrak{p}_u$, see [22, Rem. 3.13].

## 3. Some Applications of MOP

The original motivation for developing MOP was to determine modality zero parabolics in exceptional groups, i.e., ones with a finite number of orbits on the Lie algebra of the unipotent radical. Recently a complete description of all these instances was achieved with the aid of MOP, cf. [12]:

**Theorem 3.1.** *Suppose $G$ is of exceptional type and that* $\operatorname{char} K$ *is either zero or a good prime for $G$. Let $P \subseteq G$ be parabolic. Then* $\operatorname{mod} P = 0$ *if and only if one of*

(i) *$\ell(P_u) \leq 4$;*
(ii) *$G$ is of type $E_6$, $\ell(P_u) = 5$, and $P$ is of type $A_1^2 A_2$ or $A_3$;*
(iii) *$G$ is of type $E_7$, $\ell(P_u) = 5$, and $P$ is of type $A_1 A_4$.*

Some of the cases of Theorem 3.1 are also valid for certain bad primes for $G$; see 4.9 below.

The algorithm, however, can be applied more widely to determine parabolic subgroups of higher modality. For instance, for Borel subgroups $B$ of simple groups of small rank $\operatorname{mod} B$ can be determined by combining lower bounds for $\operatorname{mod} B$ from [21] with upper bounds calculated by MOP, extending results from [11]; the latter are based on a precursor of MOP.

More specifically, as an application we calculate explicit upper bounds for the modality of all parabolics in case $G$ is of type $E_6$:

| Type of $P$ | Borel | $A_1$ | $A_1^2$ | $A_1^3$ | $A_2$ | $A_1A_2$ | $A_2^2$ |
|---|---|---|---|---|---|---|---|
| $\operatorname{mod} P \leq$ | 5 | 4 | 3 | 2 | 2 | 1 | 1 |

TABLE 1. Upper bounds for $\operatorname{mod} P$ in $E_6$

In all other instances we have $\operatorname{mod} P = 0$. For $P = P_J$ in Table 1 it follows from a construction in [21] that the given values are in fact also lower bounds for $\operatorname{mod} P$ whenever one of the following holds: $P$ is of type $A_1A_2$ or $A_2^2$, or $J$ does not contain the triality simple root. That is, in these instances $\operatorname{mod} P$ equals the value shown.

Besides computing upper bounds for $\operatorname{mod}(P : \mathfrak{p}_u)$, we can specify an arbitrary invariant linear subspace $\mathfrak{n}$ of $\mathfrak{p}_u$ and calculate an upper bound for $\operatorname{mod}(P : \mathfrak{n})$. Using a particular feature of MOP we can specify such a subspace $\mathfrak{n}$ and can analyze the $P$-orbits in $\mathfrak{n}$. Remark 3.3 below is such an instance. (In fact $\mathfrak{n}$ does not have to be $P$-invariant, we can simply specify any subspace $\mathfrak{n}$ and then we can analyze the modality of the action of $P$ on the $P$-saturation of $\mathfrak{n}$, that is the $P$-variety $P \cdot \mathfrak{n}$.) For details concerning this feature, see [13].

We discuss a consequence of the classification results from [10] and [12].

**Corollary 3.2.** *Suppose $P$ is a non-maximal parabolic subgroup of $G$ and* $\operatorname{mod} P > 0$. *Then there exists a* proper *$P$-invariant subspace $\mathfrak{n}$ of $\mathfrak{p}_u$ such that* $\operatorname{mod}(P : \mathfrak{n}) > 0$.

*Proof.* This follows from an analysis of the inductive construction of all the instances when $\operatorname{mod} P > 0$. More precisely, the statement follows from the classification results in [10] and [12], the argument of the proof of [20, Thm. 6.3], together with [10, Lem. 3.2] and [11, Lem. 3.13]. □

The following $E_8$ example illustrates that the statement of Corollary 3.2 is false if the non-maximality condition on $P$ is relaxed.

*Remark* 3.3. Suppose $G$ is of type $E_8$ and $P$ is conjugate to $P_J$, where $J = \Pi \setminus \{\sigma_5\}$. It was shown in [20] that $\operatorname{mod} P > 0$. Since $P$ is a maximal parabolic subgroup of $G$, the various members of the descending central series of $\mathfrak{p}_u$ are the only $P$-invariant linear subspaces of $\mathfrak{p}_u$. Using MOP one can show that $\operatorname{mod}(P : \mathfrak{p}_u') = 0$. In particular, $\operatorname{mod}(P : \mathfrak{n}) = 0$ for every proper $P$-invariant subspace $\mathfrak{n}$ of $\mathfrak{p}_u$.

As a consequence, $P$ admits a dense orbit on $\mathfrak{p}_u^{(i)}$ for each $i \geq 1$. By Richardson's Dense Orbit Theorem [18] $P$ also has a dense orbit on $\mathfrak{p}_u$ itself. Consequently, every $P$-invariant linear subspace of $\mathfrak{p}_u$ is a *prehomogeneous vector space* for $P$, but nevertheless, $\operatorname{mod} P > 0$.

## 4. The MOP Program

In this chapter we give a description of the modality algorithm MOP and discuss some of its features. For further details on usage we refer to the MOP manual [13]. Throughout this chapter, the notation of the previous sections is in force. Let $P = P_J$ be a standard parabolic subgroup of $G$ for some $J \subset \Pi$.

4.1. MOP. As mentioned above, our algorithm is implemented in the computer algebra system GAP. The advantage of using GAP for our purposes is that firstly it comes with a useful package called CHEVIE for calculations involving Weyl groups and root systems [9], and secondly it provides convenient data structures, so-called *records*, to handle objects which depend on a variety of different parameters specified in *record fields*. For further information concerning commands in GAP we refer the reader to the GAP and CHEVIE manuals [8], [9]. The source file containing the program and data is built up in a similar fashion as the files in GAP. Loading the MOP package is achieved by the following command:

```
gap> RequirePackage("mop");
```

The fact that CHEVIE is not available as of yet for GAP4, limits MOP to GAP Version 3.4.4.

4.2. **Strings.** Instead of working with individual orbits of $P$ on $\mathfrak{p}_u$, we simultaneously consider all $P$-orbits passing through particular kinds of affine subvarieties $S$ of $\mathfrak{p}_u$ of the form $S = \sum K x_\beta + \sum M x_\gamma$, where $M = K \setminus \{0\}$, and the two sums are taken over disjoint subsets of $\Psi(\mathfrak{p}_u)$, of which one may possibly be empty (that is $S \cong M^a K^b$, where $a, b \in \mathbb{N}_0$). In particular, $S$ is a locally closed affine subvariety of $\mathfrak{p}_u$. Such a subvariety of $\mathfrak{p}_u$ is called a *string* or a *string of* $\mathfrak{p}_u$. The support of $S$ is the set of roots in $\Psi(\mathfrak{p}_u)$ whose coefficient is either $M$ or $K$.

With respect to a total ordering of $\Psi^+$, such a subvariety $S$ can be represented symbolically by a sequence of symbols "0", "$M$", and "$K$", where a "0" indicates that the corresponding root $\beta$ say, is not in the support of $S$, while $x_\beta$ has a nonzero coefficient in case of the label "$M$" and an arbitrary one if "$K$" occurs, cf. [6], [16, §6], and [11]. This explains the origin of this terminology. For example, for $A_3$ the usual ordering of positive roots is $\alpha_4 = \alpha_1+\alpha_2, \alpha_5 = \alpha_2 + \alpha_3$, and $\alpha_6 = \alpha_1 + \alpha_2 + \alpha_3$. Therefore, the subvariety $Mx_{\alpha_1} + Mx_{\alpha_1+\alpha_2} + Kx_{\alpha_2+\alpha_3}$ of $\mathfrak{b}_u$ is represented by the string "$M00MK0$".

In MOP a string $S$ is a record with two record fields `m` and `k`, which are boolean lists for the positions of $S$ labeled with $M$ or $K$, respectively. The print function for the record (string) $S$ returns the root numbers of the support of $S$.

The concept behind the program is straightforward. Suppose we aim to show that $\operatorname{mod} P \leq m$, where $m \in \mathbb{N}$. The basic objective is to construct a finite set of strings $S$ of $\mathfrak{p}_u$ with the property that firstly, every $P$-orbit in $\mathfrak{p}_u$ has a representative in some $S$ in this list, and secondly, $\operatorname{mod}(P : P \cdot S) \leq m$ for each such $S$. The desired inequality $\operatorname{mod} P \leq m$ then follows, since $\operatorname{mod} P = \max_S \operatorname{mod}(P : P \cdot S)$, where the maximum is taken over all strings $S$ in our finite collection. MOP proceeds to construct such a finite set of strings by an intricate iteration process of various *splitting* and *elimination* operations discussed below.

4.3. **The Stack.** A finite collection $\mathcal{S}$ of such subvarieties $S$ of $\mathfrak{p}_u$ satisfying the first property that every $P$-orbit on $\mathfrak{p}_u$ is represented by some string $S$, is called a *stack* of $P$ or a *stack of strings* of $P$. In our analysis, we initially start out with the stack $\mathcal{S}$ consisting only of the string $\mathfrak{p}_u$ itself. By setting an optional parameter when calling the MOP function `Modality` it is possible to initialize the stack with another string $S$ in order to compute $\mathrm{mod}(P : P \cdot S)$, see [13].

4.4. **Operations on Strings.** We perform two operations on strings in the stack $\mathcal{S}$ which preserve the property that each $P$-orbit in $\mathfrak{p}_u$ passes through some $S$ in $\mathcal{S}$. These operations are aimed at reducing the support of the strings in the stack $\mathcal{S}$, while preserving this property of $\mathcal{S}$ at the same time. The term "stack" reflects the fact that MOP keeps the strings in the well-known data structure of the same name.

4.4.1. *Splitting Operation.* The first one is simply a *splitting* or *branching* procedure. Let $S$ be a given subvariety in $\mathcal{S}$ with $K$ at position $\beta$. Then $S$ is the union of the two subvarieties $S'$ and $S''$, where in position $\beta$ the $K$ is replaced by 0 in $S'$ and by $M$ in $S''$. Thus $S'$ is of smaller support than $S$, and in $S''$ we have a new position labeled with $M$ which allows for new applications of the elimination technique described next. In this situation we replace $S$ by $S'$ and $S''$ on the stack.

4.4.2. *Elimination Operation.* This we refer to as an *elimination* or *reduction* operation. In its most elementary form it works as follows. Let $S$ be a string from the stack $\mathcal{S}$ with coefficient $M$ at $\beta$ and $K$ at $\alpha + \beta$, with $\alpha \in \Psi(P)$. Suppose that $S$ is invariant under the adjoint action of the root subgroup $U_\alpha$, that is $U_\alpha \cdot S \subseteq S$. Now let $x$ be an arbitrary element in the variety $S$. By definition, $x$ has a non-zero coefficient at $x_\beta$. By acting on $x$ with a suitable element from the root subgroup $U_\alpha$ of $P$ we can remove $\alpha + \beta$ from its support. Consequently, every $P$-orbit passing through $S$ also has a representative in the variety $S'$ say, which (as a string) is obtained from $S$ by replacing the coefficient $K$ at $\alpha + \beta$ by 0. By our assumption that $S$ is $U_\alpha$-invariant, no other roots are introduced into the support of $S'$ in this process, and thus its support is smaller than that of $S$. Finally, we replace $S$ by $S'$ on the stack $\mathcal{S}$. And we may repeat this elimination process with this new collection of strings.

MOP's elimination procedure is in fact considerably more intricate. Suppose now that in the example $S$ above a suitable operation with the root subgroup $U_\alpha$ removes the entry $K$ at $\alpha+\beta$, however, if there is a $K$, say at position $\gamma$ and a 0 at position $\alpha+\gamma$, then the action of $U_\alpha$ also produces a new entry in that coordinate. Nevertheless, suppose that this new entry can be removed again using a different root operation which in turn may or may not reintroduce new roots in the support of the resulting string. We can continue this procedure until no further new roots are being introduced. Then we have obtained a closed system of equations only involving the various positions and operators that are affected in this process. If the corresponding coefficient matrix is invertible, then the desired elimination can actually be performed. MOP checks whether such a system is solvable. The prime divisors which occur in such a process (in the Gaussian elimination) are written into a separate list in the record field `factorlist` of the output. Consequently, the result of the modality calculation is valid assuming that char $K$ is not in this `factorlist`. See the example of an output below in 4.11.

In such a more involved elimination process new positions labeled with an "$M$" or "$K$" are introduced and a subsequent operator also acts on these new positions. Therefore, the order in which a set of operators is applied matters. In the course of analyzing a given string MOP simultaneously builds up all possible systems of equations aimed at eliminating a given targeted position in the support of the string respecting the different orders of operators.

Once a consistent system of operators is obtained, MOP tries to eliminate the targeted position using this system. This guarantees that always a minimal set of operators is used.

There are certain size and time restrictions built into MOP in order to limit the elimination process. If a critical time limit is reached MOP aborts its attempts to eliminate a certain element from the support of a string and proceeds to split at that targeted position.

MOP eliminates a targeted position whenever possible. A splitting only takes place if all attempts to eliminate that position failed or if the attempt is timed out. As long as there are positions labeled with a "$K$", MOP proceeds to eliminate the one with the smallest root number, i.e., one of minimal height. Once all roots in $\operatorname{supp} S$ with coefficient "$K$" have been removed, MOP proceeds to eliminate the positions labeled with "$M$".

The algorithm proceeds by an iteration and combination of these splitting and elimination operations. We have proven that $\operatorname{mod} P$ is bounded above by $m$, once we have arrived at a stack $\mathbb{S}$ satisfying $\operatorname{mod}(P : P \cdot S) \leq m$ for each $S$ in $\mathbb{S}$.

Observe that these two operations do in fact always yield a new stack which is again labeled $\mathbb{S}$ for simplicity, that is another finite collection of strings with the desired property that every $P$-orbit through $\mathfrak{p}_u$ passes through some string in $\mathbb{S}$.

4.5. **Induction.** MOP works inductively in the following sense. Let $H$ be a proper semisimple regular subgroup of $G$ and let $Q = P \cap H$. Inductively, $\operatorname{mod} Q$ is known and in particular, we may assume that $\operatorname{mod} Q$ is at most $m$, as otherwise $\operatorname{mod} P > m$ by Lemma 2.2. It follows from the proof of Lemma 2.2 in [22, Prop. 2.2] that $\operatorname{mod}(P : P \cdot \mathfrak{q}_u) = \operatorname{mod} Q$. Therefore, we only need to consider the $P$-orbits in $\mathfrak{p}_u \setminus P \cdot \mathfrak{q}_u$. This applies to any such $Q$. Here it obviously suffices to only take those $H$ which are maximal among such subgroups leading to maximal candidates for $Q$. Hence, we only take maximal rank subgroups or Levi subgroups $H$ of corank 1 in $G$. We form the list of all conjugates of subsystems $\Psi(H)^+$, where $H$ runs through this fixed set of regular semisimple subgroups of $G$ of large rank, such that $\Psi(H)^+ \subset \Psi^+$. We refer to this list of subsystems $\Psi(H)^+$ as the `InductionList` of $G$.

If $G$ is of type $A_r$ or $D_r$, then we use Levi subsystems of type $A_{r-1}$ and $D_{r-1}$ with their usual embeddings. In that case MOP computes the `InductionList` at each run anew. For $E_6$, $E_7$, and $E_8$ we also use maximal rank subsystems, e.g., for $E_7$ we use standard subsystems of type $E_6$, $A_7$, $A_1D_6$ and $A_2A_5$. Since the computation of the induction lists in the exceptional cases takes some time, they are provided as external files. Instead of being computed, these can be read from the external files, see [13]. The symmetric subsystems corresponding to such semisimple subgroups $H$ of $G$ are determined by means of the algorithm of Borel-de Siebenthal, cf. [4, Exc. Ch. VI §4.4]. In [7] all conjugacy classes of such subsystems of $\Psi$ under the action of the Weyl group of $G$ are classified; see also [2].

For our purpose we need to examine the various parabolic subgroups $Q = P \cap H$ that actually occur in any given instance, where $\Psi(H^+)$ runs through the `InductionList` of $G$ as defined above. MOP has a device to calculate each of these and writes the information into the output record with the record fields `SubDiagrams` and `SubDiagramsDetail`. In the first one MOP writes all occurring types of $Q$ and in the second lists explicitly all embeddings of $H$ into $G$ affording such $Q$'s, see [13].

It is mandatory that we examine the list of sub-configurations in `SubDiagrams`, in order to ensure that no case is added to the induction list by mistake which does not satisfy $\operatorname{mod} Q \leq m$, otherwise the outcome of the algorithm is meaningless.

Furthermore, this feature may help to find new cases of higher modality. For instance, the fact that $\operatorname{mod} P > 0$ in case $P = P_J$ in $E_7$ for $J = \{\alpha_2, \alpha_3, \alpha_4, \alpha_5\}$ was discovered by examining the information in this list of sub-configurations; here it turns out that one of the cases occurring is the Borel subgroup of a simple subgroup of type $A_5$, cf. [11, Lem. 3.13].

Because of the inductive nature of our method, and because of the fact that we simultaneously study a collection of orbits passing through strings, we do not get any information on the number of orbits in case we have shown that $\operatorname{mod} P = 0$.

4.6. **Managing the stack.** If a sequence of splitting and elimination operations yields a string $S$ satisfying $\operatorname{mod}(P : P \cdot S) \leq m$, then, instead of keeping it, we may simply delete $S$ from the stack $\mathbb{S}$. Thus, we will ultimately have reached our goal of showing that $\operatorname{mod} P$ is at most $m$, precisely when all the strings which were generated in the course of this process have again been eliminated, that is when the resulting stack is empty. In our next section we discuss the various possibilities when a string $S$ can be removed from the stack, that is when $S$ satisfies $\operatorname{mod}(P : P \cdot S) \leq m$.

The strings which we considered in the last section on elimination and splitting operations form a tree; we define $\mathfrak{p}_u$ to be the root of the tree, and for the elimination operation, $S$ is the parent node of $S'$, and for the splitting operation, $S$ is the parent node of $S'$ and $S''$. The leaves of this tree form a stack of the operation of $P$ on $\mathfrak{p}_u$. MOP starts with $\mathfrak{p}_u$ and builds up the tree until every leaf has at least one of the following two properties, either no splitting or elimination operation can be applied to it, or one of the criteria discussed below can be used to prove that its modality is at most $m$. In order to save memory and time, MOP does not keep the whole tree in the memory. Instead it enumerates the nodes of the tree in a depth-first-search order. Whenever it finds a string for which none of the criteria applies and none of the operations can be performed, MOP writes it into the `CannotAnalyzeList`. If this list is empty after a run of the program, then $\operatorname{mod} P \leq m$, otherwise MOP computes an upper bound for the modality of $P$ from the strings in this list. The nodes of the tree are stored in a stack, which is the usual first-in-last-out data structure. The explicit use of a stack has the advantage that we can save it to a disc file and recover the data in case of a system crash. First MOP initializes the stack with a string which is just the string $\mathfrak{p}_u$ in the default setting. As long as there are strings left on the stack, MOP removes the one from the top. If one of the deletion criteria defined below applies to it, we have reached a leaf node and just continue. Otherwise we try to apply a splitting or elimination operation, put the resulting string(s) back onto the stack and continue. If these attempts fail, we have reached a leaf node again and apply the `ExtendedOperation` (see below). If this operation does not prove that the string fulfills the modality condition, we add the string to the `CannotAnalyzeList`. In any case we continue with the next string on the stack.

4.7. **Deletion Criteria.** We now describe the four different *deletion criteria* which enable us to remove a string $S$ from the stack $\mathbb{S}$, that is criteria ensuring $\operatorname{mod}(P : P \cdot S) \leq m$.

4.7.1. *Redundancy Criterion.* Instead of removing the string from the top of the stack immediately, we leave it there and mark it as "done". Whenever we find a string $S$ on the stack marked "done", we know that this string has been treated already. In this case MOP puts it into what is called the `RedundancyList`. Apart from merely writing $S$ into the `RedundancyList`, we also write every conjugate of $S$ under the simple reflections of the Weyl group $W_P$ of $P$ (i.e. those corresponding to $J$) into this list, as each such conjugate also can be considered as analyzed. In principle one could add the entire $W_P$-orbit of $S$ in $\mathfrak{p}_u$ to the `RedundancyList`. However, calculating the orbit is too time consuming. Initially, the `RedundancyList` is empty and in the course of the program run MOP adds strings to it which have already been analyzed. Thus, compared to the induction list, this is a dynamic list which is updated continually. Whenever we find a string $S$ which is a subvariety of a string in the `RedundancyList`, we may consider $S$ also as analyzed and in this case we simply drop $S$.

Before a string $S$ is added to the `RedundancyList` we first compare $S$ with any string $S'$ already in the `RedundancyList`. If $S$ is a subvariety of some $S'$ on this list, then we do not add $S$. On the other hand, any such $S'$ which is itself a subvariety of $S$ is removed from the `RedundancyList` and $S$ is added instead. This comparison feature guarantees that we maintain an optimal `RedundancyList` at any time.

4.7.2. *Induction criterion.* Suppose that in the course of our analysis we encounter a string $S$ which satisfies $\operatorname{supp} S \subseteq \Psi(H)^+$, where $\Psi(H)^+$ is a member of the `InductionList` of $G$, that is $S \subseteq \mathfrak{p}_u \cap \mathfrak{h} = \mathfrak{q}_u$, where $Q = P \cap H$. Then, by induction (cf. Section 4.5), we infer that $\operatorname{mod}(P : P \cdot S) = \operatorname{mod}(Q : Q \cdot S) \leq \operatorname{mod} Q \leq m$, and thus we can remove $S$ from $\mathbb{S}$. The equality part of this statement follows from the proof of Lemma 2.2 in [22]. We refer to this as the *induction criterion.*

4.7.3. *Rank Criterion.* Another situation when we can delete a string $S$ from the stack, arises in the following way. Suppose that the support of $S$ consists of at most $\dim T + m$ roots. Let $d$ be the number of linearly independent roots in $\operatorname{supp} S$. Note that $d$ is at most $\dim T$. Then for an arbitrary element $x$ in $S$ we can apply suitable elements from $T$ to scale as many as $d$ coefficients of $x$ to equal 1 with at most $|\operatorname{supp} S| - d$ coefficients of $x$ remaining free. Now, if $|\operatorname{supp} S| - d \leq m$, then the resulting set of $P$-orbits passing through all the elements which are obtained by varying the entries in the remaining free coefficients depends on at most $m$ parameters, and thus, $\operatorname{mod}(P : P \cdot S)$ is bounded above by $m$, as desired. Thus, we can remove $S$ from the stack $\mathbb{S}$. If $|\operatorname{supp} S| \leq m + d$, then we say that $S$ satisfies the *rank criterion.*

4.7.4. *J-height Criterion.* Finally, we have one further possibility to eliminate strings from the stack. Suppose that for $S$ in $\mathbb{S}$ each root in the support of $S$ has a fixed $J$-height $l \in \mathbb{N}$. Since MOP only applies when $G$ is simply laced, the hypothesis of Remark 2.4 on $\operatorname{char} K$ is fulfilled. Thus we have $S \subseteq \oplus \mathfrak{g}_\beta (\cong \mathfrak{p}_u^{(l-1)}/\mathfrak{p}_u^{(l)})$, where $\beta$ runs through all roots in $\Psi$ of $J$-height $l$. Now by Lemma 2.5 there are only finitely many orbits of the standard Levi subgroup of $P$ on this space; whence there are only finitely many $P$ orbits passing through $S$ and thus we can remove $S$ from the stack; then we say that $S$ satisfies the *J-height criterion.*

4.7.5. *Effectiveness of the Criteria.* In the course of a run of MOP, each of these criteria may occur many times. However, the induction and the redundancy criteria are generally the more effective ones of the four. Their advantage is twofold over the others. Firstly, they only involve a subset check of the support of the string at hand and the members of the induction list of $P$, or of the redundancy list. In terms of computing time this is not too costly provided both of these lists are short. Secondly and more importantly, these criteria allow us to remove strings from the stack which may have large support. Else these might take a long time to be analyzed after being broken down into smaller strings using the elimination and branching operations. The fact that we add all conjugates of any analyzed string under the generators of the relative Weyl group $W_P$ to the `RedundancyList` (cf. 4.7.1) makes this a very effective criterion. On the other hand, the rank criterion can only be applied when $\operatorname{supp} S \leq \dim T + m$ which is usually small compared to $\dim \mathfrak{p}_u$. A further disadvantage of it is that it requires calculating the rank of matrices which is less favorable in terms of computing time than a simple subset check as involved in the other three criteria.

4.8. **ExtendedOperation.** In the course of applying splitting and elimination operations to strings from the stack $\mathcal{S}$ it rarely happens that in the end all the strings that are produced in this fashion satisfy one of the four deletion criteria from above. Ultimately, it may happen that strings $S$ occur in $\mathcal{S}$ which have the property that the coefficient of any root in $\operatorname{supp} S$ is $M$ (that is no further branching operations are possible), no further elimination is possible (or aborted due to time limitations), and $S$ does not satisfy any of the four deletion criteria above, so it cannot be removed from the stack. Then MOP enters a process to which we refer to as *ExtendedOperation*. We choose a total ordering of all possible operators, that is of $\Psi(P)$, starting with the negative roots, ordering by height. We apply these operators consecutively to $S$ and produce entirely new strings with new entries in that fashion. The resulting strings are then analyzed further with the aforementioned splitting and elimination techniques. That is we return to the usual procedure with these new strings created in the `ExtendedOperation`. This is a systematic way to obtain a large number of new admissible strings.

4.9. **Prime Restrictions.** In terms of restrictions on $\operatorname{char} K$, MOP's results have to be interpreted as follows. If MOP is run using the `InductionList` of $G$ and $G$ is of exceptional type, then the results obtained are only valid provided $\operatorname{char} K$ is not a bad prime for $G$, cf. Lemma 2.2. If $G$ is of classical type, then using this list does not imply any characteristic restrictions, as here only Levi subsystems are involved in the construction, cf. Remark 2.3. MOP allows a user to disable this inductive feature, e.g., see the example in 4.11 below. Results obtained without using the `InductionList` are valid subject only to characteristic restrictions stemming from the `factorlist` of the output record, see 4.4.2. Often a certain bad prime $p$ does not occur in that list, and thus, the modality statement computed is also valid in case $\operatorname{char} K = p$. For instance, it turns out that each of the finite cases of Theorem 3.1 is also valid when the use of the induction list is suppressed and one observes that the prime 5 does not occur in the `factorlist` for any of the $E_8$ cases; one example is illustrated in 4.11 below. This shows that Theorem 3.1 is also valid in certain bad characteristics as well.

4.10. **Counters.** MOP keeps track of a number of parameters. It counts the number of strings that are analyzed during a run of the algorithm. Each time a string is taken off the stack we raise a counter by one. Another counter keeps track of the number of splitting operations that are performed. Of particular interest is the success of the various deletion criteria; there is a counter for each of the four deletion criteria. Apart from these MOP also has a counter for the number of calls of the internal function `ExtendedOperation`, cf. 4.8; see [13]. If there is no call of `ExtendedOperation`, then the final values of the counters for the various deletion criteria and the splitting operation add up to the value of the counter for the strings. Viewing the strings created by MOP in form of a tree as indicated in 4.6, this amounts to counting all leaf nodes (deletion criteria) together with all internal branching nodes (splittings). If there are calls of `ExtendedOperation`, then the sum of the final values of these counters can exceed the counter for the strings, as here new branchings may happen before MOP returns to the stack. This, for instance, happens in our example 4.11 below.

4.11. **An Example.** We illustrate a call of MOP in the $E_8$ instance when $J = \Pi \setminus \{\alpha_2\}$ with the use of the induction list disabled.

```
gap> RequirePackage("mop");
gap> r:=Modality("E", 8, [1,3,4,5,6,7,8], rec(UseInductionList:=false));
```

The first line reads the GAP package MOP initializing a record `MOP` and defining a function `Modality`. This function returns a record with the results of the computation written in its record fields. All the other functions are internal and are therefore located in the record `MOP`. The first three parameters of `Modality` are mandatory; they define the type of $G$, rank $G$, and the subset $J$ of simple roots defining $P$. The fourth parameter is optional; it allows to overwrite the default setting for a number of global parameters. In our example we set `UseInductionList:=false` in order to suppress the use of the induction list. For a detailed list of all optional parameters, we refer to the MOP manual [13].

After a run of MOP the results can be displayed by the print command in GAP for records:

```
gap> Print(r);
Modality analysis for type E8, J = [ 1, 3, 4, 5, 6, 7, 8 ]
E8          2
            |
    1 - 3 - 4 - 5 - 6 - 7 - 8
--------------------------------------------
19371 strings analyzed
9685 splittings
--------------------------------------------
0 induction list matches
1451 already done
2500 occurrences of rank condition
6160 J-Height criterion invoked
--------------------------------------------
62 extended operations
--------------------------------------------
0 unresolved strings
characteristic restrictions: [ 2, 3 ]
The modality of P is 0.
--------------------------------------------
```

The display is more or less self-evident. Apart from the case studied, MOP prints the result of the counters, the characteristic restrictions stemming from solving various systems of equations, as well as the modality calculation. Observe that the prime 5 does not occur in the `factorlist` and consequently, this finiteness result is also valid in characteristic 5, cf. 4.9.

4.12. **Safety Feature.** In order to recover intermediate results already obtained in case of a system crash, we periodically save all this information and the status of the analysis to a file with the suffix `.save` depending on $J$; e.g., `E8_1_2_3_4_5_7_8.save.` If such a file exists and the function `Modality` is called again with the same parameters, then MOP reads the information and status from this external file and proceeds the analysis. The default setting for the time period after which this data is recorded anew is 20 minutes.

4.13. **The `Verbose` feature.** By setting an optional parameter when calling MOP in the function `Modality` the algorithm prints out the entire analysis. In principle, this allows a user to check every detail of MOP's calculation by inspection. For details of this feature and an example in the verbose mode, see the manual [13].

**Acknowledgments** We are grateful to G. Hiss and F. Lübeck for helpful suggestions concerning programming in GAP.

## References


[1] H. Azad, M. Barry, and G. Seitz, *On the structure of parabolic subgroups*, Com. in Algebra **18**, (1990), 551–562.

[2] P. Bala and R.W. Carter, *Classes of unipotent elements in simple algebraic groups. II*, Math. Proc. Cambridge Phil. Soc., **80** (1976), 1–18.

[3] A. Borel, *Linear Algebraic Groups*, Graduate Textbooks in Mathematics, **126**, Springer-Verlag, New York-Berlin, (1991).

[4] N. Bourbaki, *Groupes et algèbres de Lie, Chapitres 4,5 et 6*, Hermann, Paris, 1975.

[5] T. Brüstle and L. Hille, *Finite, Tame and Wild Actions of Parabolic Subgroups in $GL(V)$ on Certain Unipotent Subgroups*, J. Algebra, **226**, (2000), 347–360.

[6] H. Bürgstein and W.H. Hesselink, *Algorithmic orbit classification for some Borel group actions*, Comp. Math. **61** (1987), 3–41.

[7] E.B. Dynkin, *Semisimple subalgebras of semisimple Lie algebras*, Amer. Math. Soc. Transl. Ser. 2 **6** (1957), 111–244.

[8] The GAP Group, Lehrstuhl D für Mathematik, RWTH Aachen, Germany and School of Mathematical and Computational Sciences, U. St. Andrews, Scotland. *GAP – Groups, Algorithms, and Programming, Version 3.4.4*, 1997.

[9] M. Geck, G. Hiss, F. Lübeck, G. Malle and G. Pfeiffer, CHEVIE – A system for computing and processing generic character tables for finite groups of Lie type, Weyl groups and Hecke algebras, AAECC, **7**, (1996) 175–210.

[10] L. Hille and G. Röhrle, *A classification of parabolic subgroups of classical groups with a finite number of orbits on the unipotent radical*, Transformation Groups, **4**(1), (1999) 35–52.

[11] U. Jürgens and G. Röhrle, *Algorithmic Modality Analysis for Parabolic Groups* Geom. Dedicata, **73**, (1998), 317–337.

[12] ———, *The parabolic subgroups of exceptional algebraic groups with a finite number of orbits on the unipotent radical*, Preprint 00–124, SFB 343, Bielefeld (2000); to appear.

[13] ———, *The MOP Manual*, E00–006, SFB 343, Bielefeld (2000).

[14] V.V. Kashin, *Orbits of adjoint and coadjoint actions of Borel subgroups of semisimple algebraic groups* (Russian), Problems in Group Theory and Homological algebra, Yaroslavl', (1990), 141–159.

[15] V.L. Popov, *A finiteness theorem for parabolic subgroups of fixed modality*, Indag. Math. N. S. **8 (1)**, (1997), 125–132.

[16] V.L. Popov and G. Röhrle, *On the number of orbits of a parabolic subgroup on its unipotent radical*, Algebraic Groups and Lie Groups, Australian Mathematical Society Lecture Series **9**, Editor G. I. Lehrer, Cambridge University Press, (1997), 297–320.

[17] V.L. Popov and E.B. Vinberg, *Invariant Theory*, Encyclopaedia of Math. Sci.: Algebraic Geometry IV. Springer-Verlag, **55**, (1994), 123–284. (Translated from Russian Series: Itogi Nauki i Tekhniki, Sovr. Probl. Mathem., Fund. Napravl. **55** (1989), 137–314).

[18] R.W. Richardson, *Conjugacy classes in parabolic subgroups of semisimple algebraic groups*, Bull. London Math. Soc. **6** (1974), 21–24.

[19] ———, *Finiteness Theorems for Orbits of Algebraic Groups*, Indag. Math. **88**, (1985), 337–344.

[20] G. Röhrle, *Parabolic subgroups of positive modality*, Geom. Dedicata **60**, (1996), 163–186.
[21] ——, *A note on the modality of parabolic subgroups*, Indag. Math. N.S. **8 (4)**, (1997), 549–559.
[22] ——, *On the modality of parabolic subgroups of linear algebraic groups*, Manuscripta Math., **98**, (1999), 9–20.

Fakultät für Mathematik, Universität Bielefeld, Postfach 100131, 33501 Bielefeld, Germany.
*E-mail address*: ulf@mathematik.uni-bielefeld.de
*E-mail address*: roehrle@mathematik.uni-bielefeld.de