Finite dimensional Lie algebras

Anton Nazarov

November 25, 2009

Abstract

We present concise introduction to the representation theory of finite-dimensional Lie algebras and illustrate it with the computational algorithms implemented in Scheme.

1 Introduction

Representation theory of finite-dimensional Lie algebras is central to the study of continous symmetries in physics. This theory is well-understood and there exist standard cources and textbooks on the subject [1], [humphreys]. Nevertheless some problems of the representation theory require extensive computation and no standard textbook on the computational algorithms is known to the author of this notes. There exists a volume [Belinfante], but it was written in 1970-es and have not been updated since, so its contents are limited to the early approaches and implementations on the old hardware wich is unavailable now. Also some progress was made in the computational algorithms of the representation theory since the publication of [Belinfante]. It is important to mention series of papers by Patera et al. (see [2] and references therein) and books [3], [Kass-Moody-Slansky on finite dimensional algebras] which introduce new and optimised algorithms although don't disscuss the implementation. There exist several solid implementation of the core algorithms. We want to mention Maple package Coxeter/Weyl [4] and standalone programs LiE [5] and LambdaTensor [6]. These programs are solid and rather fast but have not seen any updates in last several years. Also they are not always convenient to use since they lack graphical user interface and interoperanility with the popular programming languages and mathematical programs such as Mathematica, Python or Fortran. We want to summarise some basic notions and algorithms of representation theory in order to stimulate the emergence of more modern and universal software or at least give some tools to the scientists who by some reasons can't use the existing software.

Our implementation use programming language Scheme and is presented as the Literate program [Knuth]. The choice of the language is due to high portability of its implementations [1], wide use of Scheme for the teaching [SICP] and personal preferences of the author.

1 ⟨*library.scm* 1⟩≡

[1] There exist Scheme implementations for UNIX, Windows, Linux, Mac OS, Palm OS, Windows CE/Pocket PC/Windows Mobile, Java platform and even microcontrollers, see [some website] for the details

⟨*Semi-standard Scheme procedures .scm* (never defined)⟩
⟨*LispMe Objects .scm* (never defined)⟩

This code is written to file `library.scm`.

References

[1] R. Carter, *Lie algebras of finite and affine type.* Cambridge University Press, 2005.

[2] R. Moody and J. Patera, "Fast recursion formula for weight multiplicities," *AMERICAN MATHEMATICAL SOCIETY* **7** (1982) no. 1, .

[3] S. Kass, R. Moody, J. Patera, and R. Slansky, *Affine Lie algebras, weight multiplicities, and branching rules.* Sl, 1990.

[4] J. Stembridge, "A Maple package for symmetric functions," *Journal of Symbolic Computation* **20** (1995) no. 5-6, 755–758. `http://www.math.lsa.umich.edu/~jrs/maple.html`.

[5] M. Van Leeuwen, "LiE, a software package for Lie group computations," *Euromath Bull* **1** (1994) no. 2, 83–94. `http://www-math.univ-poitiers.fr/~maavl/LiE/`.

[6] T. Fischbacher, "Introducing LambdaTensor1. 0-A package for explicit symbolic and numeric Lie algebra and Lie group calculations," *Arxiv preprint hep-th/0208218* (2002) .