

Willoughby Seago

Notes on the

Dynkin Classification

31st March to 4th April 2025

BOCHUM SPRING SCHOOL

Dynkin Classification

Willoughby Seago

31st March to 4th April 2025

Abstract

These are my notes from the spring school “The Dynkin Classification”. This spring school was run from 31st March to 4th April 2025, taking place at Ruhr Universität Bochum in Germany as part of the “Combinatorial Synergies” program. I am grateful for funding recieved from “DFG priority program 2458 Combinatorial Synergies” and “EPSRC AGQ CDT (EP/Y035232/1)” which made this trip possible.

Contents

Contents	ii
0 Dynkin Classification	1
0.1 On the Ubiquity of Dynkin Classifications	1
0.2 Coxeter and Dynkin Diagrams	2

Chapter 0

Dynkin Classification

0.1 On the Ubiquity of Dynkin Classifications

The Dynkin classification arises in many areas of mathematics, from representation theory to algebraic geometry, and from combinatorics to string theory and spin structures, and many more. Of these we will only touch on the first three.

The structure of these notes follows that of the original spring school. Each day was focused on a different topic, and each day refers to a chapter here. Some of these days focused on the classification more than others, but all were about objects for which the Dynkin classification plays an important role.

Exactly which Dynkin diagrams are allowed in a given classification (or which of the related Coxeter diagrams appears for the first topic) depends on the situation at hand, but in all cases we get at least the A, D, and E types, leading to the so-called ADE-classification.

The precise reason why Dynkin diagrams, particularly the ADE-types, appear so frequently is somewhat of a mystery. Often, but not always, at least not in an obvious way, it's because there is some root system involved, which is a set of vectors in \mathbb{R}^n subject to some conditions (\cdot), the structure of which may be used to generate more complicated objects. Each root system has a corresponding bilinear form, and it is the properties of this inner product which usually restrict which root systems appear in a given classification. The Dynkin diagrams then are just a neat way of encapsulating the minimum amount of information required to define a root system. This is done by first picking a special basis for the root system (a set of so-called simple roots) and then encoding the relative positions of these basis vectors in a graph. The vertices of the graph correspond to elements of this basis, and the number of edges is controlled by the value of the bilinear form when evaluated at the corresponding basis vectors. The direction of any directed edges is chosen such that the arrow points towards the shorter vector. One may then argue that the omnipresence of the Dynkin classification is simply because many problems come down to, or can be phrased as, arranging some points in space subject to some relation between the position vectors relative angles and lengths.

0.2 Coxeter and Dynkin Diagrams

Throughout the first day the following Dynkin diagrams were on the board for reference. We start with our main focus, the (finite type) Dynkin diagrams. We'll then list the (finite type) Coxeter diagrams, which are similar but distinct. Finally, we'll list the affine Coxeter diagrams.

Dynkin Diagrams

Here are the **(finite type) Dynkin diagrams**. The subscript in each case is the number of nodes, which is also known as the **rank**. Dotted edges just mean a chain of nodes connected by single edges. When there are multiple edges there is an arrow, which indicates the direction of all of those edges. These finite-type Dynkin diagrams correspond to root systems which generate a positive definite bilinear form, giving a genuine inner product.

$$A_n = \bullet \text{---} \bullet \text{---} \bullet \text{---} \cdots \bullet \text{---} \bullet \text{---} \bullet \quad (0.2.1)$$

$$B_n = \bullet \text{---} \bullet \text{---} \bullet \text{---} \cdots \bullet \xrightarrow{\quad} \bullet \quad (0.2.2)$$

$$C_n = \bullet \text{---} \bullet \text{---} \bullet \text{---} \cdots \bullet \xleftarrow{\quad} \bullet \quad (0.2.3)$$

$$D_n = \bullet \text{---} \bullet \text{---} \bullet \text{---} \cdots \bullet \begin{array}{l} \nearrow \bullet \\ \searrow \bullet \end{array} \quad (0.2.4)$$

$$E_6 = \begin{array}{c} \bullet \\ | \\ \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \end{array} \quad (0.2.5)$$

$$E_7 = \begin{array}{c} \bullet \\ | \\ \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \end{array} \quad (0.2.6)$$

$$E_8 = \begin{array}{c} \bullet \\ | \\ \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \end{array} \quad (0.2.7)$$

$$G_2 = \bullet \xrightarrow{\quad} \bullet \quad (0.2.8)$$

$$F_4 = \bullet \xrightarrow{\quad} \bullet \text{---} \bullet \text{---} \bullet \quad (0.2.9)$$

Note that some of these are the same, we have the following graph isomorphisms:

- $A_1 \cong B_1 \cong C_1 \cong D_1$, all of which are just a single node;
- $B_2 \cong C_2$, both of which are two nodes connected by a double directed edge;
- $A_3 \cong D_3$, both of which are three nodes connected in a row by single edges;

Note that the sensible definition of D_2 is two disconnected nodes, just the ones appearing in the far right of the picture above. This gives us the graph isomorphism $D_2 \cong A_1 \sqcup A_1$.

There are two ways around these isomorphisms, one is to restrict the valid indices as follows:

- for B_n we require $n \geq 2$;
- for C_n we require $n \geq 3$;
- for D_n we require $n \geq 4$.

The other is to just allow these, but to keep track of whether we think of one of these graphs as being part of, say the A or D series. I prefer this second approach, since it will give us interesting isomorphisms between the objects we classify. These so-called exceptional isomorphisms are best known on the level of Lie algebras, where they correspond to

- $\mathfrak{sl}_2 \cong \mathfrak{so}_3 \cong \mathfrak{sp}_2$;
- $\mathfrak{so}_5 \cong \mathfrak{sp}_4$;
- $\mathfrak{sl}_4 \cong \mathfrak{so}_6$;
- $\mathfrak{sl}_2 \oplus \mathfrak{sl}_2 \cong \mathfrak{so}_2$.

Coxeter Diagrams

The **Coxeter diagrams** are similar to Dynkin diagrams, and the two are often conflated. Coxeter diagrams are undirected graphs, with edge labellings. The rule is that any unlabelled edges are labelled 3, and any two nodes not connected by an edge actually have an undrawn edge labelled 2. Every Dynkin diagram has, ignoring ordering, a corresponding Coxeter diagram, in which the directed edges are replaced by labelled edges with labels greater than 3. There are also some Coxeter diagrams which don't have corresponding Dynkin diagrams. We will see why these distinctions are made later, for now we just list the Coxeter diagrams. The Coxeter diagrams are as below, they have the same labelling as the corresponding Dynkin diagrams, further confusing the two.

$$A_n = \quad \blacksquare - \blacksquare - \blacksquare - \cdots - \blacksquare - \blacksquare - \blacksquare \quad (0.2.10)$$

$$B_n = C_n = \quad \blacksquare - \blacksquare - \blacksquare - \cdots - \blacksquare - \overset{4}{\blacksquare} - \blacksquare \quad (0.2.11)$$

$$D_n = \quad \blacksquare - \blacksquare - \blacksquare - \cdots - \blacksquare - \blacksquare \begin{array}{l} \nearrow \blacksquare \\ \searrow \blacksquare \end{array} \quad (0.2.12)$$

$$E_6 = \quad \begin{array}{c} \blacksquare \\ | \\ \blacksquare - \blacksquare - \blacksquare - \blacksquare - \blacksquare \\ | \\ \blacksquare \end{array} \quad (0.2.13)$$

$$E_7 = \quad \begin{array}{c} \blacksquare \\ | \\ \blacksquare - \blacksquare - \blacksquare - \blacksquare - \blacksquare - \blacksquare \\ | \\ \blacksquare \end{array} \quad (0.2.14)$$

$$E_8 = \begin{array}{c} \blacksquare \\ | \\ \blacksquare - \blacksquare - \blacksquare - \blacksquare - \blacksquare - \blacksquare - \blacksquare - \blacksquare \end{array} \quad (0.2.15)$$

$$G_2 = \begin{array}{c} 6 \\ | \\ \blacksquare - \blacksquare \end{array} \quad (0.2.16)$$

$$F_4 = \begin{array}{c} 4 \\ | \\ \blacksquare - \blacksquare - \blacksquare - \blacksquare \end{array} \quad (0.2.17)$$

$$H_2 = \begin{array}{c} 5 \\ | \\ \blacksquare - \blacksquare \end{array} \quad (0.2.18)$$

$$H_3 = \begin{array}{c} 5 \\ | \\ \blacksquare - \blacksquare - \blacksquare \end{array} \quad (0.2.19)$$

$$H_4 = \begin{array}{c} 5 \\ | \\ \blacksquare - \blacksquare - \blacksquare - \blacksquare \end{array} \quad (0.2.20)$$

$$I_2(m) = \begin{array}{c} m \\ | \\ \blacksquare - \blacksquare \end{array} \quad (0.2.21)$$

As with the Dynkin diagrams there are certain graph isomorphisms here for specific values of indices. In addition to the isomorphisms of the Dynkin diagrams we also have the following:

- $G_2 \cong I_2(6)$;
- $H_2 \cong I_2(5)$.

Affine Coxeter Diagrams

The final type of diagram we'll need are the affine Coxeter diagrams. We won't be particularly interested in the structures that these classify, they tend to be infinite versions of the finite versions the finite types generate, but they are needed to prove the classification of Coxeter groups. These affine types are formed from the finite types by adding another node. There are generally many ways to do this, but most end up giving isomorphic graphs. In fact, for every outer automorphism of the original graph there is a unique (up to isomorphism) way to add an extra node. We will give only the simplest here, which are the ones corresponding to the identity automorphism. The other types are called "twisted" affine Coxeter graphs. There are two common notations for the affine Coxeter graph, they are \tilde{A}_n and $A_n^{(1)}$. The advantage of the second is that the superscript can index the corresponding outer automorphism, since we're not interested in the twisted case the tilde will be sufficient for our purposes.

Here are all the untwisted affine Coxeter graphs. The new node is the one in colour. Note that the index is now one fewer than the number of nodes.

$$\tilde{A}_1 = \begin{array}{c} \blacksquare - \textcolor{red}{\blacksquare} \end{array} \quad (0.2.22)$$

$$\tilde{A}_n = \begin{array}{c} \textcolor{red}{\blacksquare} \\ / \quad \backslash \\ \blacksquare - \blacksquare - \blacksquare - \cdots - \blacksquare - \blacksquare \end{array} \quad (0.2.23)$$

(0.2.24)

(0.2.25)

(0.2.26)

(0.2.27)

(0.2.28)

(0.2.29)

(0.2.30)

(0.2.31)

First note that \tilde{A}_1 is “weird”, it’s sort of a degenerate case in which adding the extra node as we would for \tilde{A}_n gives us two nodes with edges between them and this 2-cycle somehow explodes into an infinite loop. Note also that \tilde{B}_n and \tilde{C}_n are different Coxeter diagrams, even though B_n and C_n are the same Coxeter diagram.

Chapter 1

Coxeter Groups