## Setting the scene for Kant

###### 

1. Introduction

One does not just jump into Kant. He was the product o fthis time

1

The purpose of these notes to capture preliminary notes regarding the nature of graphs. But why graphs? Because a graph is a vitally important structure, wikth a rich and foundational ontological structure and deep implications. Its sheer generality allows for far reaching investigation across multiple fields. Graphs appear everywhere: mathematics, chemistry, physics, both integrating and undermining the deepest truths of these disciplines.

The structure of these notes that follow aim to provide on overview of graphs, but loosely following Normal Wildberger's work on graphs, and placing them in a philisophical framework. These notes are a kind of commentary and preliminary, and some of the philisophical rigour will be put off for a more formal write up.

Many of the foundational ideas to be explored in these notes come from Wildberger's seminal paper: The Mutation Game, Coxeter–Dynkin Graphs, and Generalized Root Systems, Normal Wildberger, March 2020. This is a difficult read, and I would suggest that not to spend too much time on this paper (not yet at least) until I have laid some foundations. It should start to make sense in the end.

I am going to try and be careful here about the way I use language and as I am trying to put it into a philisophical framework. But these are preliminary notes, so sometimes it will not be too formal.

2[¶](http://127.0.0.1:8888/lab/tree/work/MathNotebooks/ES1_1.ipynb#2)

So let's begin with a definition. What is a graph?

A Graph is defined as an unordered list (let's call the list ListA noting it does not matter what we call it) and another list which is not ListA (let's call that one ListB)

List A has the following properties:

It is list of things (such as [thing1, thing2, thing3....][thing1, thing2, thing3....]).

Each thing in ListB is not the same as any other thing ListB

ListB has the following properties:

Each thing in ListB is a list.

Each list holds a thing and some other thing is not the same.

Each thing that is in the lists in List B must also be in List A

Each thing in ListB is not the same as any other thing ListB

Admittedly, this definition feels convoluted. The definitions is difficult arising because I am trying to avoid things thing like the concept of 2 and the like, sequential (like there is this thing and then there is another thing). This turns out to matter later on. As a simpler, guiding information intuiation, it is ok to say that a Graph is two lists, one tracking some things, the other tracking relationships between things, like this:

ListA = [thing1, thing2, thing3...][thing1, thing2, thing3...],

ListB = [[thing1, thing2], [thing1, thing3], [thing2, thing3]...][[thing1, thing2], [thing1, thing3], [thing2, thing3]...].

For an intuition, this is fine, and I would advise to keep the simpler conception in mind. But there are some reasons for choosing this definition which I will return to later.

3[¶](http://127.0.0.1:8888/lab/tree/work/MathNotebooks/ES1_1.ipynb#3)

So this definition (or at least its accompanying intuition) seems simple. So simple that I might be tempted to ask a silly question such as, how big can ListA or ListB? But of course, that does not really matter (for what does big mean?). It may be worth noting that if things keep getting added to ListA or ListB, it might become become impossible to add more things. So you should not think of ListA as infinite (and you have no concept of what that means anyway). If you know a little about computers, a guiding intuition that you might find helpful is, if you keep on trying to put things into List A, you might run out of storage.

Maybe I am not out of silly questions. How about, what is a thing? It does not matter. Maybe a thing is a galaxy, maybe a thing is an electron. It does not matter. A thing is just a delineation that has no content that I can know. This all gets a bit category theory

More interesting to note, is that definiton describes a very general structure. Now of course, it certainly not completely general. It assumes a list, it assumes state change, it does not assume to understand what a thing is but assumes differentiation. Somewhere here there is a nothingness, Satre. But think of all the things that are not here: there is no notion of length, no geometry,distance. A graph really only arises through state and storage (and I have written on this here). I am in the world, I have the capacity state and memory, I am goint to try. Might think of this is a Descartes definition. Decartes regito cos all he needed to say was interesting thing here is that only names things

4

Let's introduce some more names for things. Until I run out of storage, I can rename things to my heart's content. So let's name anything in ListA a Vertice. And let's also name anything ListB an Edge. I will also start using the computer to explore this. This does not add anything, but will help make it explicit.

Let's also create a graph by creating lists as per the definition above.

Why are my lists called 𝐹1�1 and 𝐹2�2? It doesn't matter. I could have named them anything, that's the thing with names. The name of one list cannot be the name of the other list, but it can called anything. And why are the things in the my list '1', '2''1', '2' types of symbols. It does not matter. They could be anything. But all of this seems convenient.

Let's also create some names for other things. let's make it intuitively explicit. I need a way to tap into the mathematical language

We say that eedge is 'undirected'. Intuitively it just means that there is no order between an edge. So (1,2)(1,2) is the same thing as (2,1)(2,1) This turns out not to matter but it comes up in language. Language is full of these conveniences and we need to be careful

There can be only one edge between any two verticies (i.e. only one edge is allowed to connect to a given vertice). note that this fals straight from the definiton above

it has no loops (i.e. a node cannot be connected to itself through an edge). Again, straight from the definition above.

5

Let's return to this definition. Note that there are some rules around ListB. What I did not enforce them?

Each list holds a thing and some other thing is not the same becomes →→ Each list holds a thing and some other thing that can be the same same thing

Each thing in ListB is not the same as any other thing ListB →→ Each thing in ListB can be the same as any other thing ListB

What does this do? It means that there is some kind of repeition that begines to emerge.

6

Language is always problematic. We like to name things endlessly, name those things more things, name a thing and another thing some other thing! Soon it all gets confusing. To get some clarity, let's look at a picture! It is strange that this migth clarify things at all. Because to ask what a graph might look like is meaningless. It is completely relative. We should acknowledge we are using this intution as a crutch. I can feel Kant somewhere getting nervous. But even so, it will help with these slightly different rules

I am going to use F3 above, my very first graph and give an indication of what it might look like:

EXAMPLES

You can see in this example, The ListB allow repetition and here is the outcome. Multiple edges joining the same verticies. Edges between a single node and itself. But the most important thing here is the rules. Note that as soon as we are looking at things, everything loses rigour. Look here, now there! What is here and there? Who knows

7

So let's now capture these definitions of a graph, that have been created using two types of rules. We will call the first one Simple Graph and the second Multigraph

A Simple Graph is defined as an unordered list (let's call the list ListA noting it does not matter what we call it) and another list which is not ListA (let's call that one ListB)

List A has the following properties:

It is list of things (such as [thing1, thing2, thing3....][thing1, thing2, thing3....]).

Each thing in ListB is not the same as any other thing ListB

ListB has the following properties:

Each thing in ListB is a list.

Each list holds a thing and some other thing is not the same.

Each thing that is in the lists in List B must also be in List A

Each thing in ListB is not the same as any other thing ListB

A MultiGraph is defined as an unordered list (let's call the list ListA noting it does not matter what we call it) and another list which is not ListA (let's call that one ListB)

List A has the following properties:

It is list of things (such as [thing1, thing2, thing3....][thing1, thing2, thing3....]).

Each thing in ListB is not the same as any other thing ListB

ListB has the following properties:

Each thing in ListB is a list.

Each thing that is in the lists in List B must also be in List A

Each list in ListB holds a thing and some other thing that can be the same same thing

Each thing in ListB can be the same as any other thing ListB

8[¶](http://127.0.0.1:8888/lab/tree/work/MathNotebooks/ES1_1.ipynb#8)

This definitions means that each graph might not be the same, or might not look the same. Maybe I spend my entire life, each momement I have graph. Maybe I don't? They are different. I will not elaborate here too much, but just to note in the study of graphs, things get categorised in different ways. So Simple Graphs can be categorised into the following:

ADE graphs

ADE~ graphs

All other simple graphs

And Multigraphs can be categorised

BCFG Graphs

BCFG~ graphs

All other directed multigraphs

For the moment, this is not too important, it highlights a tendency to categorise things, that human experience just an endless parade of namning things. We started with a graph, now there is a Simple Graph, now a MultiGraph. And of course it just keeps on going. It keeps going just like a graph!

9

Now what can I do with this structure to pass the time? How about a game? I am going to create a game that uses Simple Graphs. There are other games to be sure, the different rules of a game that is based on MultiGraphs seems interesting. But I will with Simple Graph as its extra rules might keep things simple.

This game should have a name. Let's call it the Mutation Game. Start by creating a graph, called 𝑋�.

Note that I will need to start bringining in more mathematical language, mathematical concepts. But I want to be careful. I can introduce definitions for things, but I don't want to bring in anything that is extra to me original definition. But I need to be very careful, I can make graphs and there are things in graphs, but there is not math lying around and I don't want to have explain maths, to create axioms (whatever that means). I don't want to go beyond. Recall I can make a graph, because I can create a unordered list. So I don't want to suggest anythign that goes byond this. But because I can name things, I want to take advantage of this.