Wiskunde-bijles - graphs

The INFDEV team

Hogeschool Rotterdam
Rotterdam, Netherlands

Graphs

Denition and notation

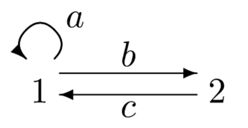
- A graph is made of **nodes** and **arrows**.
- Each arrow has a **source** (domain) and **target** (codomain).
- The notation '$f : a \longrightarrow b$' means that f is an arrow and a and b are its source and target, respectively.
- There may be one or more arrows – or none at all – with given nodes as source and target
- An arrow with the same source and target node will be called an **endoarrow** or **endomorphism** of that node.

Graphs

Denition and notation

- We will denote the collection of nodes of a graph $\mathcal{G}$ by G_0 and the collection of arrows by G_1, and similarly with other letters ($\mathcal{H}$ has nodes H_0, $\mathcal{C}$ has nodes C_0, and so on).
- In literature it is often the case that a graph $\mathcal{G}$ is denoted as a pair (V, E) where V is a finite set and E is a binary relation on V. The set V is called **vertex set** of $\mathcal{G}$ (and its elements are called **vertices**). The set E is called **edge set** of $\mathcal{G}$ (and its elements are called **edges**).

Example

- Let G_0 ={1,2}
- Let G_1 ={a,b,c},
  source(a) = target(a) = source(b) = target(c) = 1 $\wedge$ target(b) = source(c) = 2
- Then we can represent $\mathcal{G}$ as
  

-

Graphs

Small and large graphs

- A graph that has a `set` of nodes and arrows is a **small graph**; otherwise, it is a **large graph**.

Graphs

Discrete graph

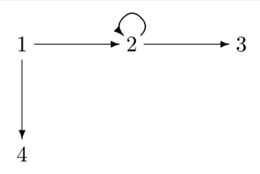
- A graph is called **discrete** if it has no arrows.
- The empty graph, with no nodes and no arrows, is discrete.
- A small discrete graph is essentially a set

Graphs

Finite graph

- A graph is **nite** if the number of nodes and arrows is nite.

Example

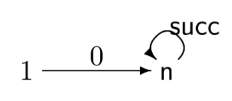
- It is often convenient to picture a relation on a set as a graph.
- Let A $=\{1,2,3\}$, B $=\{2,3,4\}$ and $\alpha = \{$ (1,2),(2,2),(2,3),(1,4) $\}$
- Then α can be pictured as
- 

- Note that the graph of a function, as dened in the previous lecture, is a relation and so corresponds to a graph.

Graph of a function - remark

- Note that the graph of a function, as dened in the previous lecture, is a relation and so corresponds to a graph.
- The resulting picture has an arrow from each element x of the domain to $f(x)$ so it is not the graph of the function in the sense used in calculus.

Example

- Sometimes one can represent a data structure by a graph. The following graph represents the set $\mathbb{N}$ of natural numbers in terms of zero and the successor function (adding 1):
- 

- The name '1' for the left node is the conventional notation to require that the node denote a **singleton set**, that is, a set with exactly one element.

Graphs

Example in C#

- And now some practical example on the computer
- A counter

Example in C#

- And now some practical example on the computer
- Types and casting

In conclusion

- Graphs offer a powerfull tool for abstracing computation details in favor of high-level transformations.
- By defining arrows over nodes, interesting properties can be deduced such as composition.
- Sometimes it is easier to reason in terms of graphs, so in terms of transformations, rather than thinking about actual implementations (see types transformations).
- ...Next class homomorphisms of graphs that is a structure-preserving map between two algebraic structures (in our case two graphs)

This is it!

The best of luck, and thanks for the attention!