Vorlesung 2:

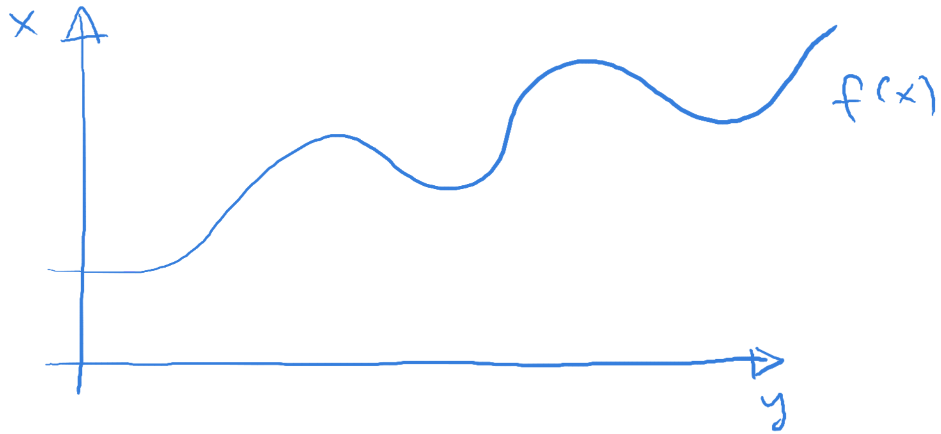
Graphentheorie

Markus Püschel
David Steurer
Peter Widmayer

Algorithmen und Datenstrukturen, Herbstsemester 2017, ETH Zürich

Funktionsgraph

bekannt aus der Schule

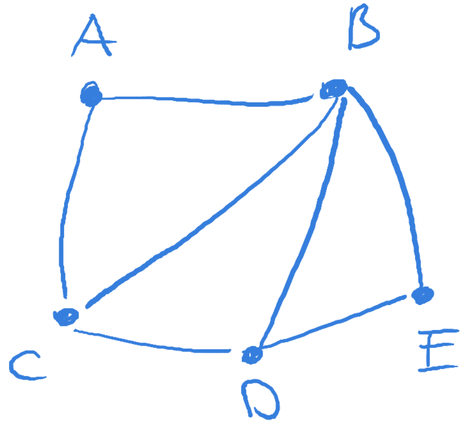


hat aber leider nichts mit Graphentheorie zu tun!

Kartenfärbung

- male jedes Land in einer Farbe
- benachbarte Länder müssen verschiedene Farben haben
- verwende möglichst wenig Farben

Beobachtung: es kommt nur darauf an welche Paare von Länder benachbart sind



Prüfungszeitplan

Eingabe: n geplante Prüfungen und k mögliche Termine

Ziel: weise jeder Prüfung einen Termin zu

Bedingung: zwei Prüfungen brauchen verschiedene Termine wenn $\geqslant 1$ Student beide Prüfungen schreibt

Beobachtung: nur wichtig für welche Paare von Prüfungen ein Konflikt besteht (d.h. $\geqslant 1$ Student schreibt beide)

Prüfungen P_1, P_2, P_3, P_4

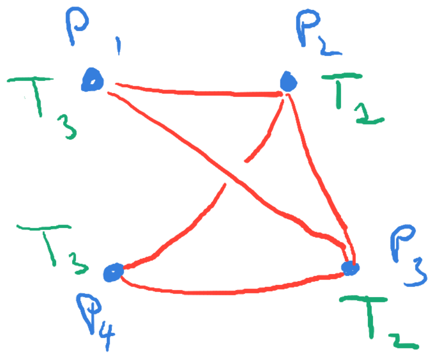
Termine T_1, T_2, T_3

Konflikt paare:

P_1 & P_2 $\quad$ P_2 & P_4

P_1 & P_3 $\quad$ P_3 & P_4

P_2 & P_3



Einsicht: «Kartenfärbung» und «Prüfungszeitplan» sind im Kern dasselbe Problem

Graphentheorie liefert die Abstraktion um diese gemeinsame Struktur hervorheben

ermöglicht: diesselben Lösungsideen können auf viele verschiedene Problem angewandt werden

«die Macht der Abstraktion»

Graphentheorie

Definition: ein *Graph* $G = (V, E)$ besteht aus:

- Menge von *Knoten* $V = \{v_1, \ldots, v_n\}$ (englisch: vertices)
- Menge von *Kanten* $E = \{e_1, \ldots, e_m\}$ (englisch: edges)

dabei ist jede Kante e_k ein ungeordnetes Paar zweier Knoten v_i, v_j (also: $e_k = \{v_i, v_j\}$)

Definition: ein *Graph* $G = (V, E)$ besteht aus:

- Menge von *Knoten* $V = \{v_1, \ldots, v_n\}$ (englisch: vertices)
- Menge von *Kanten* $E = \{e_1, \ldots, e_m\}$ (englisch: edges)

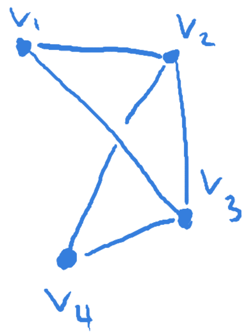
dabei ist jede Kante e_k ein ungeordnetes Paar zweier Knoten v_i, v_j (also: $e_k = \{v_i, v_j\}$)

Begriffe: Knoten v_i, v_j sind *adjazent / benachbart* in G falls $\{v_i, v_j\} \in E$

Knoten v_i und Kante e_k sind *inzident* falls $v_i \in e_k$

Nachbarschaft $N_G(v_i)$ ist die Menge der benachbarten Knoten von v_i

Grad $\deg_G(v_i) := |N_G(v_i)|$



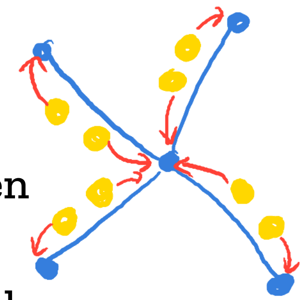
$$E = \left\{ \begin{array}{l} \{v_1, v_2\} \\ \{v_2, v_3\} \\ \{v_1, v_3\} \\ \{v_2, v_4\} \\ \{v_3, v_4\} \end{array} \right\}$$

$$V = \{v_1, v_2, v_3, v_4\}$$

Behauptung: Für jeden Graphen G mit Knoten $V = \{v_1, \dots, v_n\}$ und Kanten $E = \{e_1, \dots, e_m\}$ gilt

$$\deg_G(v_1) + \cdots + \deg_G(v_n) = 2m$$

Beweis: wir verteilen $2m$ Geldstücke gerecht an alle Kanten



jede Kante e_k bekommt zwei Geldstücke

jetzt verteilt jede Kante ihre Geldstücke an die beiden Knoten zu denen sie inzident ist

jeder Knoten v_i bekommt insgesamt also $\deg_G(v_i)$ Geldstücke

wenn wir jetzt alle Geldstücke wieder einsammeln, haben wir insgesamt $\deg_G(v_1) + \cdots + \deg_G(v_n)$ Geldstücke

da während der Prozedur die Zahl der Geldstücke konstant blieb, muss gelten $\deg_G(v_1) + \cdots + \deg_G(v_n) = 2m$

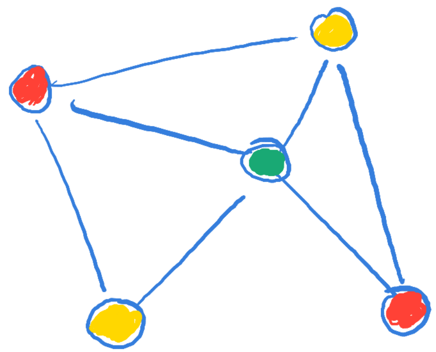
Graphenfärbung

Eingabe: Graph $G = (V, E)$ und Anzahl k von Farben

Ziel: Färbe die Knoten von G wenn möglich mit k Farben, so dass jede Kante $e = \{u, v\}$ in G erfüllt, dass u und v unterschiedliche Farben haben

es gibt exponentiell viele mögliche Färbungen
(k^n Färbungen von n Knoten mit k Farben)

daher können nur für kleine Graphen alle Färbungen durchprobiert werden



Graphenfärbung

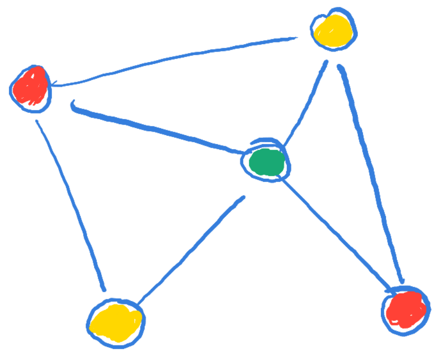
Eingabe: Graph $G = (V, E)$ und Anzahl k von Farben

Ziel: Färbe die Knoten von G wenn möglich mit k Farben, so dass jede Kante $e = \{u, v\}$ in G erfüllt, dass u und v unterschiedliche Farben haben

daher können nur für kleine Graphen alle Färbungen durchprobiert werden

Ausblick: die P$\neq$NP Vermutung besagt dass kein Algorithmus im Allgemeinen wesentlich besser sein kann

Aber: Für $k = 2$ Farben gibt es sehr effiziente Algorithmen



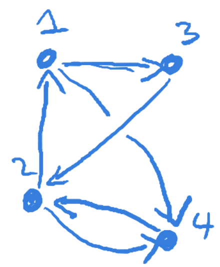
Definition: ein *gerichteter Graph* $G = (V, E)$ besteht aus einer Knotenmenge V und Kantenmenge E, so dass jede Kante $e \in E$ ein (geordnetes) Paar zweier Knoten $u, v \in E$ ist (also: $e = (u, v)$)

Begriffe: Eine Kante $e = (u, v)$ geht von ihrem *Fuss* u zu ihrem *Kopf* v

Falls $(u, v) \in E$, ist u ein *Vorgänger* von v und v ein *Nachfolger* von u

Vorgängerschaft $N_G^-(v)$ ist Menge der Vorgänger von v und *Nachfolgerschaft* $N_G^+(v)$ ist Menge der Nachfolger von v

Eingangsgrad $\deg_G^-(v) := |N_G^-(v)|$
Ausgangsgrad $\deg_G^+(v) := |N_G^+(v)|$



$V = \{1, 2, 3, 4, 5\}$

$$E = \left\{ \begin{array}{c} (1,3) \\ (2,1) \\ (3,2) \\ (2,4) \\ (1,4) \\ (4,2) \end{array} \right\}$$

Beispiele gerichteter Graphen

- *Graph der Hyperlinks im World Wide Web:* spielt zentrale Rolle für Suchmaschinen (PageRank Algorithmus benannt nach Google-Mitgründer Larry Page)
- *Strassennetz:* wichtig für Navigationssysteme
- *soziale Netzwerke*, z.B. «follower»-Netzwerk bei Twitter

Graphen und Datenstrukturen

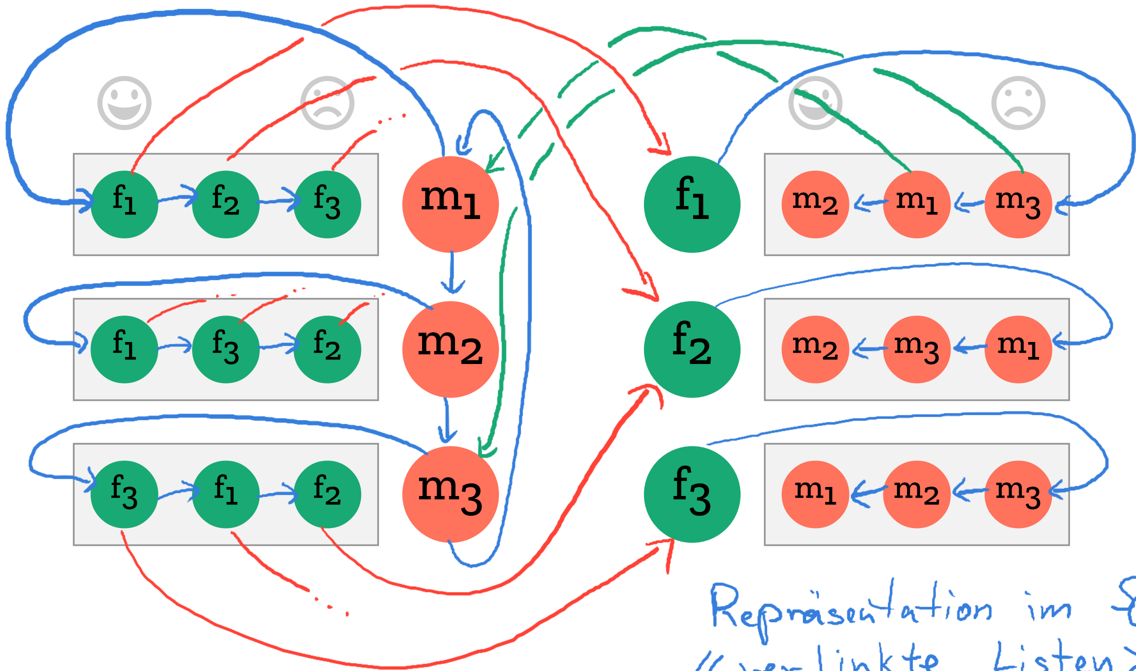
Computerprogramme bearbeiten während ihrer Ausführung den Computerspeicher

Algorithmen abstrahieren Computerprogramme

Datenstrukturen abstrahieren den Computerspeicher

Datenstrukt. können oft durch Graphen beschrieben werden

Beispiel: Gale–Shapley Algorithmus



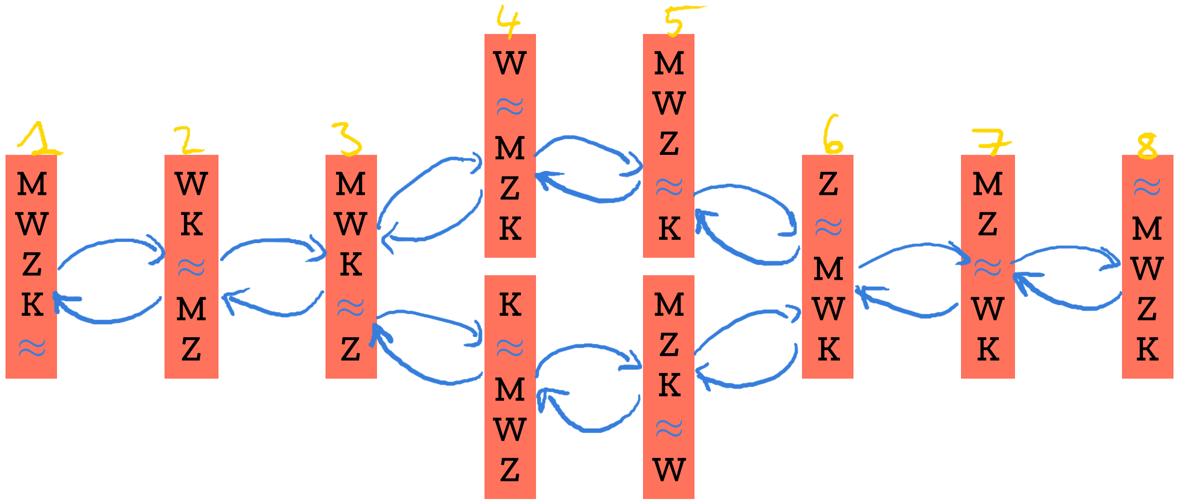
Repräsentation im Speicher: «verlinkte Listen», d.h. Graph

Speicherzugriff von Gale-Shapley: Weg in diesem Graph

Flussüberquerung

- ein Mann möchte zusammen mit einem Wolf, einer Ziege und einem Kohlkopf einen Fluss überqueren
- das Boot fasst ausser ihm nur einen weiteren Passagier
- er kann weder den Wolf mit der Ziege noch die Ziege mit dem Kohl unbeaufsichtigt an einem Ufer lassen

betrachte alle möglichen Zustände und Zustandsübergänge



Lösung:
8 Schritte

Wege und Pfade

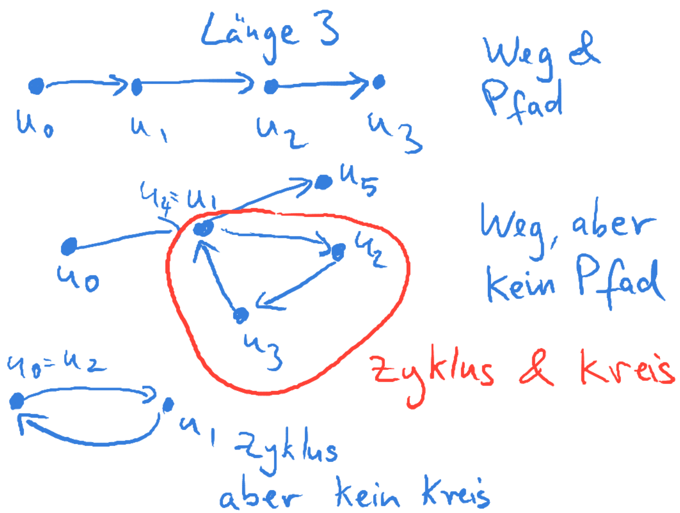
sei $G = (V, E)$ ein gerichteter Graph

Begriffe: ein *Weg* mit *Startknoten* $u_0 \in V$ und *Endknoten* $u_\ell \in V$ der *Länge* ℓ ist eine Folge von Knoten $u_0, \ldots, u_\ell \in V$, so dass $(u_0, u_1), (u_1, u_2), \ldots, (u_{\ell-1}, u_\ell) \in E$

ein *Zyklus* ist ein Weg mit demselben Start- und Endknoten

ein *Pfad* ist ein Weg bei dem jeder Knoten höchstens einmal vorkommt

ein *Kreis* ist ein Zyklus mit $\ell \geqslant 3$, so dass $u_1, \ldots, u_\ell$ paarweise verschieden sind



Wege und Pfade

sei $G = (V, E)$ ein gerichteter Graph

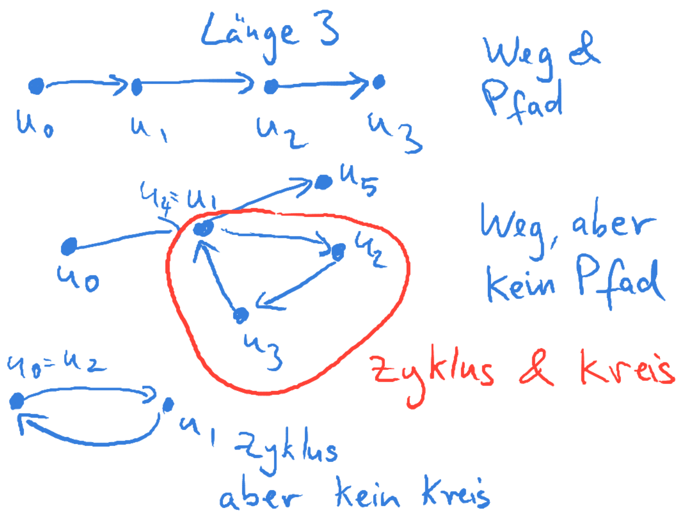
Begriffe: ein *Weg* mit *Startknoten* $u_0 \in V$ und *Endknoten* $u_\ell \in V$ der *Länge* ℓ ist eine Folge von Knoten $u_0, \ldots, u_\ell \in V$, so dass $(u_0, u_1), (u_1, u_2), \ldots, (u_{\ell-1}, u_\ell) \in E$

ein *Zyklus* ist ein Weg mit demselben Start- und Endknoten

ein *Pfad* ist ein Weg bei dem jeder Knoten höchstens einmal vorkommt

ein *Kreis* ist ein Zyklus mit $\ell \geqslant 3$, so dass $u_1, \ldots, u_\ell$ paarweise verschieden sind

Bemerkung: diesselben Begriffe gelten für ungerichtete Graphen; wir stellen uns dabei vor dass jede ungerichtete Kante $\{u, v\}$ für beide gerichtete Kanten $(u, v), (v, u)$ steht

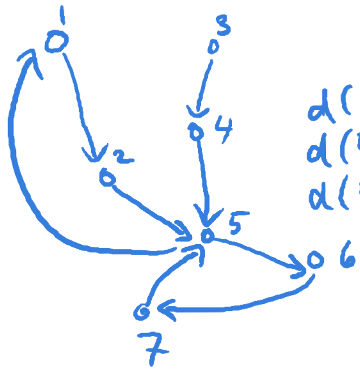


Erreichbarkeit und Distanz

sei $G = (V, E)$ ein Graph (gerichtet oder ungerichtet)

Begriffe: ein Knoten $v \in V$ ist von einem Knoten $u \in V$ *erreichbar*, wenn es einen Weg von u nach v in G gibt

die *Distanz* $d(u, v)$ ist die Länge des kürzesten Wegs von u nach v

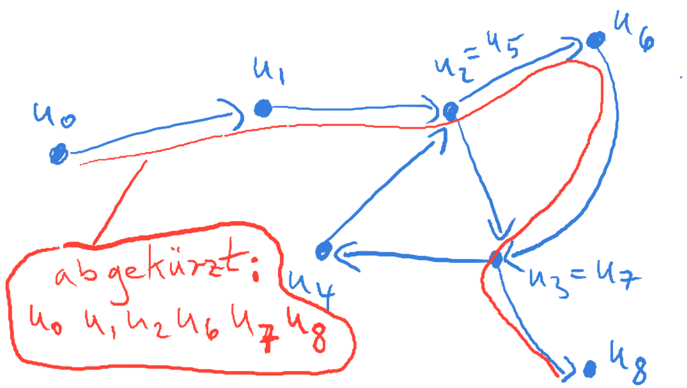


$d(1,5)=2$
$d(5,1)=1$
$d(5,3)=\infty$ (3 nicht von 5 erreichbar)

Ausblick: sowohl für Erreichbarkeit als auch für Distanzen gibt es effiziente Algorithmen

Behauptung: Wenn es einen Weg der Länge ℓ von u nach v gibt und $u \neq v$, dann gibt es einen Pfad von u nach v mit Länge $\leqslant \ell$

Idee: solange Weg abkürzen bis ein Pfad übrig bleibt



hört der Abkürzprozess irgendwann auf?

wie können wir das formalisieren?

Einschub: Induktion

Ziel: zeige $\forall n \in \mathbb{N}.\ A(n)$ für eine Aussage $A(n)$, z.B.

$A(n)$: die Summe $1 + 2 + \cdots + n$ ist gleich $\frac{1}{2}n \cdot (n+1)$

$A(n)$: wenn es einen Weg von u nach v mit Länge $\leqslant n$ gibt,
dann gibt es einen Pfad von u nach v mit Länge $\leqslant n$

Theorem: um zu zeigen dass eine Aussage $A(n)$ für alle natürlichen Zahlen $n \geqslant 1$ gilt, reicht folgendes aus:

- *Induktionsanfang:* zeige, dass $A(1)$ gilt
- *Induktionsschritt:* zeige $A(n-1) \rightarrow A(n)$ für alle $n \geqslant 2$

im Induktionsschritt müssen wir $A(n)$ zeigen, aber dürfen dabei annehmen dass $A(n-1)$ gilt (*Induktionshypothese*)

$A(n)$: die Summe $1 + 2 + \cdots + n$ ist gleich $\frac{1}{2} \cdot n \cdot (n+1)$

Induktionsanfang: zeige $A(1)$

in der Tat gilt $1 = \frac{1}{2} \cdot 1 \cdot (1+1)$

Induktionsschritt: zeige $A(n-1) \to A(n)$ für all $n \geqslant 2$

nach *Induktionshypothese* gilt:

$$1 + 2 + \cdots + n - 1 + n = \frac{1}{2} \cdot (n-1) \cdot n + n$$

nach etwas Algebra sehen wir:

$$\frac{1}{2} \cdot (n-1) \cdot n + n = \frac{1}{2} \cdot n \cdot (n+1)$$

seien u und v zwei verschiedene Knoten

$A(n)$: wenn es einen Weg von u nach v mit Länge $\leqslant n$ gibt,
dann gibt es einen Pfad von u nach v mit Länge $\leqslant n$

Induktionsanfang: $A(1)$

jeder Weg von u nach v der Länge $\leqslant 1$ ist ein Pfad seien u und v zwei verschiedene Knoten

$A(n)$: wenn es einen Weg von u nach v mit Länge $\leqslant n$ gibt,
dann gibt es einen Pfad von u nach v mit Länge $\leqslant n$

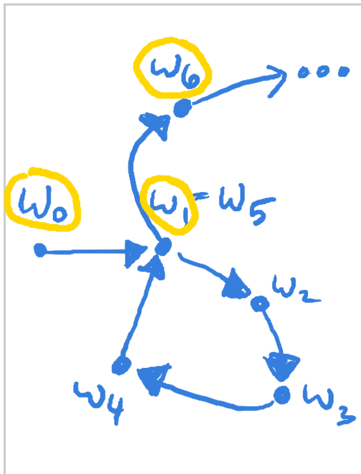
Induktionsschritt: $A(n-1) \to A(n)$ für alle $n \geqslant 2$

sei $w_0 := u$; betrachte beliebigen Weg $w_0, w_1, \ldots, w_{n-1}, v$ der Länge n

wir können annehmen dass dieser Weg kein Pfad ist (ansonsten ist nichts zu tun)

daher $\exists$ Knoten $w_i = w_j$ mit $0 \leqslant i < j$, der mehr als einmal auf dem Weg vorkommt

dann ist $w_0, \ldots, w_i, w_{j+1}, \ldots, v$ ein Weg von u nach v der Länge $\leqslant n-1$



nach *Induktionshypothese* gibt es daher einen Pfad von u nach v der Länge $\leqslant n-1$; q.e.d.

Tücken bei Induktionsbeweisen

sowohl *Induktionsanfang* als auch *Induktionsschritt* müssen sorgfältig geführt werden

insbesondere müssen alle Werte von n betrachtet werden
(entweder im Induktionsanfang oder im Induktionsschritt)

ansonsten können leicht falsche Aussagen «bewiesen» werden

Beispiel für fehlerhaften Induktionsbeweis

$A(n)$: in jeder Herde von n Kühen haben
alle Kühe dieselbe Farbe

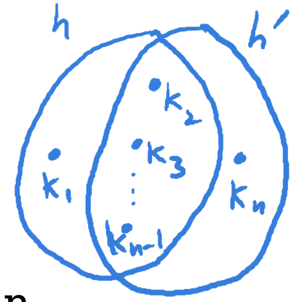
Induktionsanfang: die Aussage $A(1)$ gilt offensichtlich

Induktionsschritt: zu zeigen $A(n-1) \rightarrow A(n)$

für $n \geqslant 2$, sei $\{k_1, \ldots, k_n\}$ eine beliebige Herde von n Kühen

$A(n)$: in jeder Herde von n Kühen haben
alle Kühe diesselbe Farbe

Induktionsschritt: zu zeigen $A(n-1) \to A(n)$

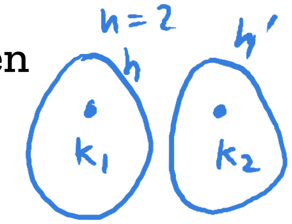


für $n \geqslant 2$, sei $\{k_1, \ldots, k_n\}$ eine beliebige Herde von n Kühen

betrachte Herden $h = \{k_1, ..., k_{n-1}\}$ und $h' = \{k_2, ..., k_n\}$

beide Herden haben $n-1$ Kühe

nach *Induktionshypothese* $A(n-1)$ haben sowohl die Kühe in Herde h also auch die Kühe in Herde h' jeweils diesselbe Farbe



~~da Kuh~~ $\cancel{k_2}$ ~~in beiden Herden~~ $\cancel{h}$ ~~und~~ $\cancel{h'}$ ~~ist,~~ haben beide Herden diesselbe Farbe

es folgt dass alle Kühe $h_1, \ldots, h_n$ diesselbe Farbe haben

Inductionsschritt schlägt fehl für $n = 2$***,***
aber ist korrekt für alle $n \geqslant 3$