Graphentheorie

Anwendungsbeispiel Faust-Makrogenese

Thorsten Vitt

WS 2018/2019

Ausgangssituation

- Genese-Informationen in der Edition von Goethes *Faust*
- Es gibt zahlreiche Handschriften ...
- (ggf. gibt es mehrere Inskriptionen pro Handschrift)
- Handschriften sind nicht datiert!
- Einzelaussagen zu Datierungen aus der Forschung

Datierungen: relativ

- Aussagen aus der Forschung zum Verhältnis einzelner Handschriften

```xml
<relation name="temp-pre">
  <source uri="faust://bibliography/wa_I_15_2">S.153-154</source>
  <item uri="faust://document/wa/2_V_H.3alpha"/>
  <item uri="faust://inscription/wa/2_V_H.4/i_uebrige"/>
  <item uri="faust://document/wa/2_V_H.3"/>
</relation>
```

Datierungen: absolut

- Einschränkung des möglichen Niederschriftsdatums

```
<date notBefore="1825-02-26" notAfter="1825-04-05">
  <comment>Frühjahr 1825</comment>
  <source uri="faust://bibliography/bohnenkamp1994">S.763</source>
  <item uri="faust://inscription/wa/2_V_H.4/i_uebrige"/>
</date>
```

Auswertungsziele

1. partielle Ordnung auf der Basis der relativen Datierungen

Auswertungsziele

1. partielle Ordnung auf der Basis der relativen Datierungen
2. Einbeziehung der absoluten Datierungen

Auswertungsziele

1. partielle Ordnung auf der Basis der relativen Datierungen
2. Einbeziehung der absoluten Datierungen
3. Erkennen und ggf. Entfernen von Widersprüchen in der Forschung

Auswertungsziele

1. partielle Ordnung auf der Basis der relativen Datierungen
2. Einbeziehung der absoluten Datierungen
3. Erkennen und ggf. Entfernen von Widersprüchen in der Forschung
4. Reduktion der Graphkomplexität für Teilvisualisierungen

Auswertungsziele

1. partielle Ordnung auf der Basis der relativen Datierungen
2. Einbeziehung der absoluten Datierungen
3. Erkennen und ggf. Entfernen von Widersprüchen in der Forschung
4. Reduktion der Graphkomplexität für Teilvisualisierungen
5. Chronologische Ordnung, die konsistent mit den Daten ist

1. Modellierung nur relativer Datierungen

- Datenmodell
- ggf. Beschränkung auf einzelne Quellen
- ggf. Beschränkung auf einzelne Dateien

1. Modellierung nur relativer Datierungen

- Datenmodell
- ggf. Beschränkung auf einzelne Quellen
- ggf. Beschränkung auf einzelne Dateien

Ziele:

- Reihenfolge erzeugen

1. Modellierung nur relativer Datierungen

- Datenmodell
- ggf. Beschränkung auf einzelne Quellen
- ggf. Beschränkung auf einzelne Dateien

Ziele:

- Reihenfolge erzeugen
- Aussagen widersprüchlich?

1. Modellierung nur relativer Datierungen

- Datenmodell
- ggf. Beschränkung auf einzelne Quellen
- ggf. Beschränkung auf einzelne Dateien

Ziele:

- Reihenfolge erzeugen
- Aussagen widersprüchlich?
- Reihenfolge von Teilknotenmengen?

1. Modellierung nur relativer Datierungen

- Datenmodell
- ggf. Beschränkung auf einzelne Quellen
- ggf. Beschränkung auf einzelne Dateien

Ziele:

- Reihenfolge erzeugen
- Aussagen widersprüchlich?
- Reihenfolge von Teilknotenmengen?

1. Modellierung nur relativer Datierungen

- Datenmodell
- ggf. Beschränkung auf einzelne Quellen
- ggf. Beschränkung auf einzelne Dateien

Ziele:

- Reihenfolge erzeugen
- Aussagen widersprüchlich?
- Reihenfolge von Teilknotenmengen?

Analyse des Graphen

- nicht verbundene Graphteile

1. Modellierung nur relativer Datierungen

- Datenmodell
- ggf. Beschränkung auf einzelne Quellen
- ggf. Beschränkung auf einzelne Dateien

Ziele:

- Reihenfolge erzeugen
- Aussagen widersprüchlich?
- Reihenfolge von Teilknotenmengen?

Analyse des Graphen

- nicht verbundene Graphteile
- Widersprüche

Zusammenhang (1): ungerichteter Graph

- Ein ungerichteter Graph G heißt **zusammenhängend**, falls es zu jedem beliebigen Knotenpaar u, v aus G einen Pfad von u zu v gibt.
- Zu einem ungerichteten Graphen G heißt ein maximal zusammenhängender Teilgraph $G' \subseteq G$ **Zusammenhangskomponente** von G.
- Ein Graph zerfällt in seine Zusammenhangskomponenten.

Zusammenhang (2): gerichteter Graph

- Ein gerichteter Graph G heißt **stark zusammenhängend**, falls es zu jedem beliebigen Knotenpaar u, v aus G einen Pfad von u nach v gibt (und offensichtlich auch von v nach u).
- Zu einem gerichteten Graphen G heißt ein maximal stark zusammenhängender Teilgraph $G' \subseteq G$ **starke Zusammenhangskomponente** von G. (Triviale SZKs bestehen aus nur je einem Knoten ...)

Zusammenhang (2): gerichteter Graph

- Ein gerichteter Graph G heißt **stark zusammenhängend**, falls es zu jedem beliebigen Knotenpaar u, v aus G einen Pfad von u nach v gibt (und offensichtlich auch von v nach u).
- Zu einem gerichteten Graphen G heißt ein maximal stark zusammenhängender Teilgraph $G' \subseteq G$ **starke Zusammenhangskomponente** von G. (Triviale SZKs bestehen aus nur je einem Knoten ...)

- Ein gerichteter Graph G heißt **schwach zusammenhängend**, wenn der gerichtete Graph H, der entsteht, wenn man die Richtung der Kanten in G ignoriert, zusammenhängend ist.
- Zu jeder Zusammenhangskomponente U aus H ist der durch U induzierte Teilgraph von G, $G[U]$, eine **schwache Zusammenhangskomponente**.

Topologische Sortierung

Eine **topologische Sortierung** s der Knoten eines gerichteten azyklischen Graphen (DAG) $G = (V, E)$ ist eine Sequenz (Reihenfolge) aller Knoten V, die mit den Kanten des Knoten konsistent ist, d.h. $\forall uv \in E : u <_S v$.

- Zu einem DAG kann es mehrere topologische Sortierungen geben.
- Zu einem zyklischen Graphen gibt es keine topologische Sortierung.

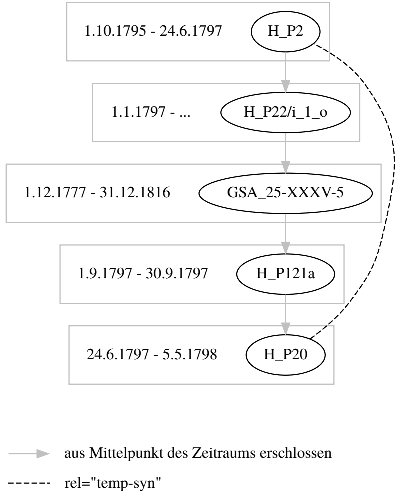
Topologische Sortierung: Algorithmus

$s \leftarrow \emptyset, G = (V, E)$
while $G \neq \emptyset$:
  wähle $u \in V$ mit $d^-(u) = 0$
  $s \leftarrow su$
  $G \leftarrow G - u$

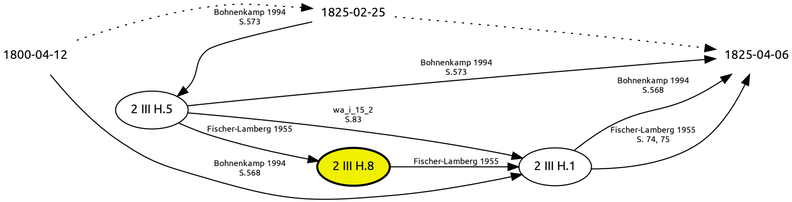
- da G azyklisch ist, muss es mindestens einen Knoten u mit $d^-(u) = 0$ geben.
- $d^-(u) = 0 \Rightarrow \neg\exists vvu \in E$, es kann also bedenkenlos u an die Sequenz gehängt werden, ohne dass gefahr droht, dass noch eine Kante von einem späteren Knoten auf u zeigt

2. Integration der absoluten Datierungen

2. Integration der absoluten Datierungen: Ansatz Wissenbach



2. Integration der absoluten Datierungen: Ansatz Vitt



Minimum Feedback Arc Set

Das *Minimum Feedback Arc Set* ist die kleinste Menge an Kanten eines Graphen, die entfernt werden muss, damit der Graph azyklisch ist.

Minimum Feedback Arc Set

Das *Minimum Feedback Arc Set* ist die kleinste Menge an Kanten eines Graphen, die entfernt werden muss, damit der Graph azyklisch ist.

- NP-vollständig
- problematisch vor allem für Graphen mit vielen einfachen Zyklen
- Heuristiken

Heuristiken

- Heuristik, die einen azyklischen Subgraph mit $\geq |E|/2$ Kanten liefert

Heuristiken

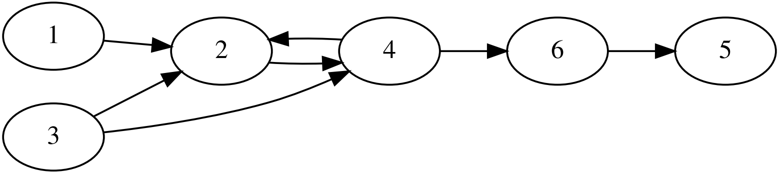
- Heuristik, die einen azyklischen Subgraph mit $\geq |E|/2$ Kanten liefert
- trivial

Heuristiken

- Heuristik, die einen azyklischen Subgraph mit $\geq |E|/2$ Kanten liefert
- trivial

Heuristiken

- Heuristik, die einen azyklischen Subgraph mit $\geq |E|/2$ Kanten liefert
- trivial



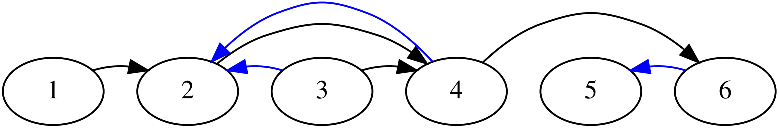
Beispielgraph

Triviale Heuristik

- Idee: Jede lineare Anordnung der Knoten teilt die Kantenmenge in zwei Partitionen

Triviale Heuristik

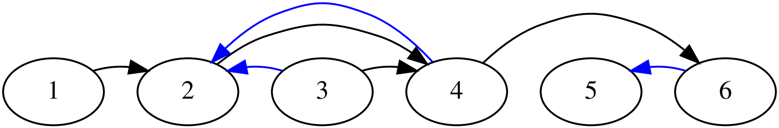
- Idee: Jede lineare Anordnung der Knoten teilt die Kantenmenge in zwei Partitionen



- Knotensequenz s (beliebig gewählt)

Triviale Heuristik

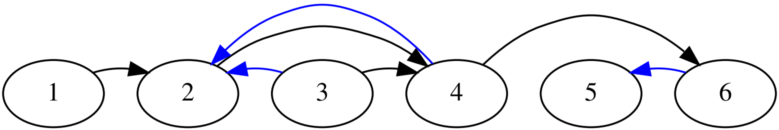
- Idee: Jede lineare Anordnung der Knoten teilt die Kantenmenge in zwei Partitionen



- Knotensequenz s (beliebig gewählt)
- Vorwärtskanten $F_s(G) := \{uv \mid u <_s v\}$

Triviale Heuristik

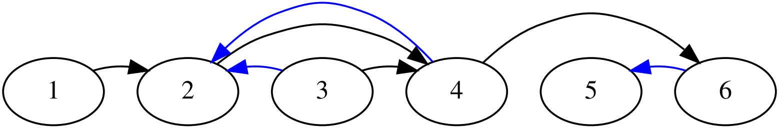
- Idee: Jede lineare Anordnung der Knoten teilt die Kantenmenge in zwei Partitionen



- Knotensequenz s (beliebig gewählt)
- Vorwärtskanten $F_s(G) := \{uv \mid u <_s v\}$
- Rückwärtskanten $R_s(G) := \{uv \mid u >_s v\}$

Triviale Heuristik

- Idee: Jede lineare Anordnung der Knoten teilt die Kantenmenge in zwei Partitionen



- Knotensequenz s (beliebig gewählt)
- Vorwärtskanten $F_s(G) := \{uv \mid u <_s v\}$
- Rückwärtskanten $R_s(G) := \{uv \mid u >_s v\}$
- die kleinere Kantenmenge wird entfernt

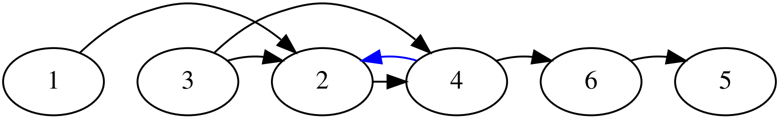
Algorithmus von Eades et al. (1993)

- Sinnvolle, *greedy* Konstruktion der Knotensequenz
- Entfernen der Rückwärtskanten

Algorithmus von Eades et al. (1993)

$s_1 \leftarrow \emptyset, s_2 \leftarrow \emptyset$
while $G \neq \emptyset$:
  while G contains a sink u:
    $s_2 \leftarrow u s_2$
    $G \leftarrow G - u$
  while G contains a source u:
    $s_1 \leftarrow s_1 u$
    $G \leftarrow G - u$
  if $G \neq \emptyset$:
    choose u for which $\delta(u) := d^+(u) - d^-(u)$ is maximum
    $s_1 \leftarrow s_1 u; G \leftarrow G - u$
$s \leftarrow s_1 s_2$
(sink / Senke: Knoten mit $d^+(v) = 0$, source / Quelle: Knoten mit $d^-(v) = 0$)

Algorithmus von Eades et al. (1993)



Graph nach Eades-Heuristik

Algorithmus von Eades et al. (1993)

- Integration von Gewichten

Algorithmus von Eades et al. (1993)

- Integration von Gewichten
- Berechnung von Gewichtsdifferenz statt Graddifferenz

Exakte Berechnung z.B. durch Integer-Programmierung

(Lineare Optimierung mit ganzzahligen Variablen; nach Baharev et al. 2015)

$$\min_y \sum_{j=1}^{m} w_j y_j$$

$$\text{sodass } \sum_{j=1}^{m} a_{ij} y_j \geq 1 \qquad \text{für alle } i = 1, 2, \ldots, \ell$$

- m Kanten, w_j Gewichte,
- $y_j = 1$ gdw. j im Feedback Arc Set, sonst 0
- ℓ Anzahl der einfachen Zyklen
- $a_{ij} = 1$ wenn Kante j in Zyklus i, sonst 0

Exakte Berechnung z.B. durch Integer-Programmierung

(Lineare Optimierung mit ganzzahligen Variablen; nach Baharev et al. 2015)

$$\min_y \sum_{j=1}^{m} w_j y_j$$

$$\text{sodass } \sum_{j=1}^{m} a_{ij} y_j \geq 1 \qquad \text{für alle } i = 1, 2, \ldots, \ell$$

- m Kanten, w_j Gewichte,
- $y_j = 1$ gdw. j im Feedback Arc Set, sonst 0
- ℓ Anzahl der einfachen Zyklen
- $a_{ij} = 1$ wenn Kante j in Zyklus i, sonst 0

- Problem: ℓ kann in $\Omega(2^n)$ sein

Weiterführende Verfahren

- z.B. Heuristiken basierend auf Sortieralgorithmen (Brandenburg 2011)
- Divide-and-Conquer-Heuristik über Multicuts in speziellen Netzwerken (Even et al. 1995, Even et al. 1998)
- Optimierte Exakte Methode nach Baharev et al, 2015.
- Heuristiken in speziellen Graphen (Tournaments, planare Graphen)

Abschlussaufgabe

- Implementierung Makrogenesemodell
- Implementierung einer Heuristik (z.B. Eades)
- Eins von:
  a) Integration abs. Datierung. Feedback-Kantenmengen und erfasste Zeugen mit / ohne abs. Datierungen
  b) Analyse nach Quellen (anz. Kanten, anz. Widersprüche)
  c) kann man irrtümlich entfernte Kanten wieder hinzufügen, ohne dass die Performance einbricht? Wie und wieviele im Experiment?

Literatur

Baharev, A., Schichl, H. and Neumaier, A. An exact method for the minimum feedback arc set problem. : 34.

Brandenburg, F. J. and Hanauer, K. (2011). Sorting Heuristics for the Feedback Arc Set Problem. : 13.

Eades, P., Lin, X. and Smyth, W. F. (1993). A fast and effective heuristic for the feedback arc set problem. *Information Processing Letters*, **47**(6): 319–23 doi:10.1016/0020-0190(93)90079-O. http://www.sciencedirect.com/science/article/pii/002001909390079O (accessed 27 July 2018).

Even, G., Naor, J., Rao, S. and Schieber, B. (1995). Divide-and-Conquer Approximation Algorithms via Spreading Metrics (Extended Abstract). pp. 62–71.

Even, G., Naor, J. (Seffi), Schieber, B. and Sudan, M. (1998). Approximating Minimum Feedback Sets and Multicuts in Directed Graphs. *Algorithmica*, **20**(2): 151–74 doi:10.1007/PL00009191. https://link.springer.com/article/10.1007/PL00009191 (accessed 27 July 2018).