Mathematatische Grundlagen der Computerlinguistik

Graphentheorie

Dozentin: Wiebke Petersen

11. Foliensatz

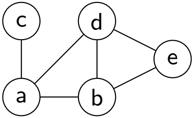
Graph

Ein Graph ist ein geordnetes Paar (V, E) bestehend aus einer Mengen V und einer Menge E von zweielementigen Teilmengen von V. Es gilt $V \cap E = \emptyset$.

Die Elememnte von V heißen Ecken (vertices) und die von E Kanten (edges).

- Statt von den Ecken spricht man auch oft von den Knoten eines Graphen
- Die hier definierten Graphen werden häufig auch einfache Graphen genannt.
- Graphen mit endlicher Eckenmenge heißen endliche Graphen. Sie lassen sich durch Diagramme visualisieren:

$$V = \{a, b, c, d, e\}, E = \{\{a, b\}, \{a, c\}, \{a, d\}, \{b, e\}, \{b, d\}, \{d, e\}\}$$



Terminologie (Graphen)

- Die Ordnung eines Graphen (V, E) ist $|V|$. Graphen mit Ordnungen kleiner 2 heißen trivial.
- Eine Ecke v heißt mit einer Kante e inzident, wenn $v \in e$.
- Der Grad einer Ecke v ist die Zahl der mit v inzidenten Kanten.
- Zwei Ecken $v_1, v_2 \in V$ heißen adjazent oder benachbart, wenn $\{v_1, v_2\} \in E$.
- Ein Graph, in dem je zwei Ecken benachbart sind, heißt vollständig.
- Zwei Graphen (V, E) und (V', E') heißen isomorph, wenn es eine Bijektion $\phi : V \to V'$ gibt mit:
  $\{v, w\} \in E \Leftrightarrow \{\phi(v), \phi(w)\} \in E'$.
- (V', E') ist ein Teilgraph von (V, E), wenn $V' \subseteq V$ und $E' \subseteq E$.

Graphen und symmetrische Relationen

- Jeder Graph (V, E), definiert eine binäre, symmetrische, irreflexive Relation E auf V.
- Jede binäre, symmetrische, irreflexive Relation R auf einer Menge M definiert einen Graphen (M, R).

Graphen und symmetrische Relationen

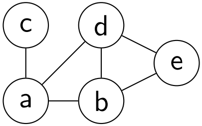
- Jeder Graph (V, E), definiert eine binäre, symmetrische, irreflexive Relation E auf V.
- Jede binäre, symmetrische, irreflexive Relation R auf einer Menge M definiert einen Graphen (M, R).

Frage: Warum muss die Relation symmetrisch und irreflexiv sein, um einen Graphen zu definieren?

Wege in Graphen

Sei (V, E) ein Graph.

- Eine Folge von Knoten $(v_1, v_2, \ldots, v_n)$ bildet einen Weg in (V, E), genau dann wenn für alle $1 \geq i < n$ gilt: $\{v_i, v_{i+1}\} \in E$ und wenn $v_i \neq v_j$ für alle $i \neq j$.
- Ein Folge von Knoten $(v_1, v_2, \ldots, v_n)$ in (V, E) heißt Kreis, wenn $v_1 = v_n$, $\{v_{n-1}, v_n\} \in E$ und $(v_1, v_2, \ldots, v_{n-1})$ einen Weg in (V, E) bildet.



Terminologie (Wege)

- Wenn $(v_1, v_2, \ldots, v_n)$ ein Weg in (V, E) ist, dann heißen v_1 und v_n die Endknoten des Weges.
- Knoten eines Wegs, die keine Endknoten sind, heißen innere Knoten.
- Wege in Graphen können auch als Teilgraphen des Graphen aufgefasst werden.
- Ein Knoten u heißt von einem Knoten v erreichbar in einem Graphen, wenn es einen Weg gibt, der v und u als Endknoten hat. Sonst heißt u unerreichbar von v.
- Ein Graph in für je zwei beliebige Knoten gilt, dass sie untereinander erreichbar sind, heißt zusammenhängend.
- Zwei Wege heißen kreuzungsfrei oder (knoten)disjunkt, wenn sie keine inneren Knoten gemeinsam haben.
- Die Länge eines Weges $(v_1, v_2, \ldots, v_n)$ ist die Zahl seiner Kanten, sprich $v - 1$.

Terminologie (Kreise)

- Ein Kreis (v_1, v_2, v_3, v_1) aus drei Kanten heißt Dreieck
- Ein Graph heißt zyklisch, wenn er mindestens einen Kreis enthält, sonst heißt er azyklisch

Bäume und Wälder

Ein kreisfreier Graph heißt Wald.

Ein zusammenhängender Wald heißt Baum.

Die Ecken vom Grad 1 eines Baums heißen Blätter.

Satz:
Jeder zusammenhängende Graph wird von einem Baum aufgespannt.
Ein Baum (V', E') spannt einen Graphen (V, E) auf, wenn $V' = V$ und $E' \subseteq E$. Dieser Baum wird auch Spannbaum des Graphen genannt.

Planare Graphen

Ein Graph heißt planar oder plättbar, wenn er auf einer Ebene mit Punkten für die Knoten und Linien für die Kanten dargestellt werden kann, sodass sich keine Kanten schneiden.

Hinweis: Jeder Baum ist plättbar. Warum?

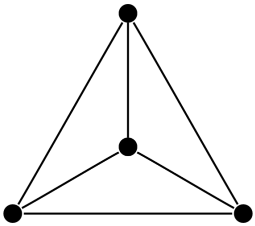


Abbildung: Planare Zeichnung des K_4

Quelle: `https://de.wikipedia.org/wiki/Planarer_Graph`

Berühmte Probleme der Graphentheorie

Die Graphentheorie ist ein extrem breites Gebiet mit zahlreichen Anwendungen, wichtigen Algorithmen und bekannten Problemen. Hier ein kleiner Appetizer.

- Das Königsberger Brückenproblem (`http://www.matheprisma.uni-wuppertal.de/Module/Koenigsb/index.htm`)
- Das Vierfarbenproblem (`http://www.matheprisma.uni-wuppertal.de/Module/4FP/index.htm`)
- Suche kürzester Wege (`http://www.matheprisma.uni-wuppertal.de/Module/Graphen/index.htm`)

Im folgenden werden einige häufig verwendete Erweiterungen einfacher Graphen eingeführt.

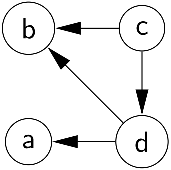
Gerichtete Graphen

Ein gerichteter Graph (auch Digraph, 'directed graph') besteht aus einer Menge V von Knoten und einer Menge geordneter Knotenpaare $E \subseteq V \times V$ von Kanten.

- Die Kanten $(v, w) \in E$ eines gerichteten Graphen sind gerichtete Kanten. Die Darstellung erfolgt meistens als Pfeil. Dieser gibt die zu durchlaufende Richtung an.
- Digraphen werden zum Beispiel zur Darstellung von endlichen Automaten verwendet.
- Eine gerichtete Kante $e = (x, y)$ geht von x nach y. Wobei x der Startknoten und y der Endknoten von e ist. Außerdem gilt y als der direkte Nachfogler von x und x als direkter Vorgänger von y.

Beispiel gerichteter Graph

$$V = \{a, b, c, d\},\ E = \{(d, b), (d, a), (c, d), (c, b)\}$$



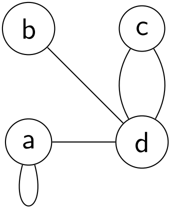
Multigraphen

Ein Multigraph besteht aus einer Menge V von Knoten, einer Menge E von Kanten und einer Funktion $f : E \to \{\{x, y\} : x, y \in V\}$, welche jeder Kante einen oder zwei Endecken zuweist.

- In einem Multigraph können zwei Knoten durch mehrere, unterschiedene Kanten miteinander verbunden sein. Hierbei spricht man auch von Mehrfachkanten.
- Kanten, die von einer Ecke zu dieser zurück laufen (also eine Ecke mit sich selbst verbinden), werden Schlingen genannt.

Beispiel Multigraph

$$V = \{a, b, c, d\},\ E = \{e_1, e_2, e_3, e_4, e_5\}$$
$$e_1 \mapsto \{d, c\},\ e_2 \mapsto \{d, c\},\ e_3 \mapsto \{d, a\},\ e_4 \mapsto \{d, b\},\ e_5 \mapsto \{a\}$$

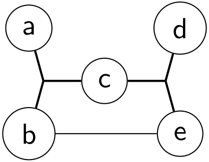


Hypergraph

Ein Hypergraph ist ein Tupel (V,E), wobei V $= \{v_1, v_2, \ldots, v_n\}$ die Eckenmenge und $E \subseteq \mathcal{P}(V)$ mit $\emptyset \notin E$ die Menge der Hyperkanten bezeichnet.

- Eine Kante (bzw. Hyperkante) verbindet hier also nicht nur zwei, sondern mehrere Knoten gleichzeitig.

$$V = \{a, b, c, d, e\},\ E = \{\{a, b, c\}, \{c, d, e\}, \{b, e\}\}$$



Gewichtete Graphen

Ein Kantengewicht wird durch die Kantengewichtsfunktion f: E $\rightarrow \mathbb{R}$ gegeben. Diese Funktion ordnet jeder Kante eine reelle Zahl als Gewicht zu. Hierbei wird das Kantengewicht einer Kante $e \in E$ mit f(e) oder f_e bezeichnet.

Genauso kann jedem Knoten ein Knotengewicht gegeben werden (sprich: Jedem Knoten wird eine reele Zahl als Gewicht zugeordnet).

Gewichtete Graphen

Ein Kantengewicht wird durch die Kantengewichtsfunktion f: E $\rightarrow \mathbb{R}$ gegeben. Diese Funktion ordnet jeder Kante eine reelle Zahl als Gewicht zu. Hierbei wird das Kantengewicht einer Kante $e \in E$ mit f(e) oder f_e bezeichnet.

Genauso kann jedem Knoten ein Knotengewicht gegeben werden (sprich: Jedem Knoten wird eine reele Zahl als Gewicht zugeordnet).

- Zu einem knoten-/kantengewichteten Graphen gehört also neben der Angabe der Knoten- und Kantenmenge auch die Angabe einer Funktion, die von den Knoten/Kanten in die Menge der reellen Zahlen abbildet.
- Mit Kantengewichten kann man z.B. die Stärke der Bindung zwischen zwei Knoten modellieren.
- Knotengewichte können die Wichtigkeit eines Knotens modellieren.

Repräsentation eines Graphen als Adjazenzmatrix

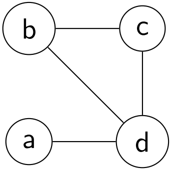
Eine Adjazenzmatrix eines Graphen ist eine Matrix, die speichert, welche Knoten des Graphen durch eine Kante verbunden sind. Sie besitzt für jeden Knoten eine Zeile und eine Spalte, woraus sich für n Knoten eine $n \times n$-Matrix ergibt. Ein Eintrag in der i-ten Zeile und j-ten Spalte gibt hierbei an, ob der i-te und der j-te Knoten adjazent sind. Steht an dieser Stelle eine 0, ist keine Kante vorhanden – eine 1 gibt an, dass eine Kante existiert.

Mit Adjazenzmatrizen lassen sich einfache Graphen, gerichtete Graphen und kantengewichtete Graphen repräsentieren.

Adjazenzmatrix: einfache Graphen

Sei $G = (V, E)$ ein einfacher Graph. Die Adjazenzmatrix $A_G = [a_{ij}]$ des Graphen G ist durch seine Einträge definiert als:

$$a_{ij} = \begin{cases} 1, & \text{falls } \{i, j\} \in E, \\ 0, & \text{sonst} \end{cases}$$

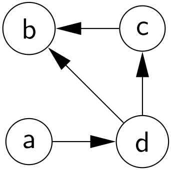
Beispiel: 

$$A_G = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & 0 & 0 & 1 \\ b & 0 & 0 & 1 & 1 \\ c & 0 & 1 & 0 & 1 \\ d & 1 & 1 & 1 & 0 \end{array}$$

Adjazenzmatrix: gerichtete Graphen

Sei $G = (V, E)$ ein gerichteter Graph. Die Adjazenzmatrix $A_G = [a_{ij}]$ des Graphen G ist durch seine Einträge definiert als:

$$a_{ij} = \begin{cases} 1, & \text{falls } (i, j) \in E, \\ 0, & \text{sonst} \end{cases}$$

Beispiel: 

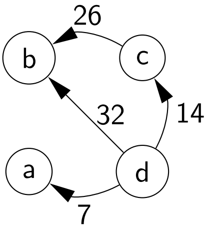
$$A_G = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & 0 & 0 & 1 \\ b & 0 & 0 & 0 & 0 \\ c & 0 & 1 & 0 & 0 \\ d & 0 & 1 & 1 & 0 \end{array}$$

Adjazenzmatrix: kantengewichtete Graphen

Sei $G = (V, E)$ ein gerichteter Graph **mit** Kantengewichten. Die Adjazenzmatrix $A_G = [a_{ij}]$ des kantengewichteten Graphen G = (V,E,c) mit Kantengewicht c, ist durch seine Einträge definiert als

$$a_{ij} = \begin{cases} c_{ij}, & \text{falls } (i,j) \in E, \\ 0, & \text{sonst} \end{cases}$$

Beispiel:



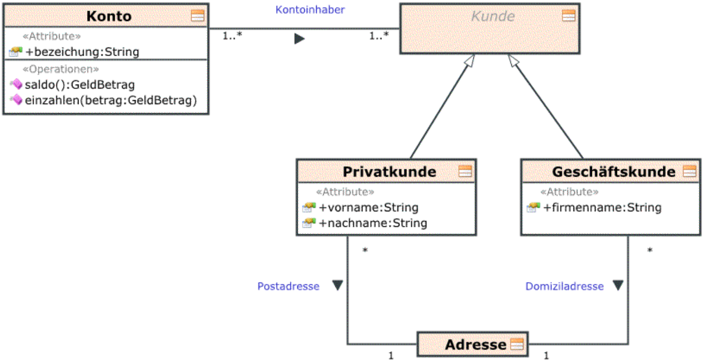
$$A_G = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 26 & 0 & 0 \\ 7 & 32 & 14 & 0 \end{pmatrix}$$

Mögliche Anwendungen als Computerlinguist*in

- Klassendiagramme im objektorientierten Programmentwurf (OOP)
- Projektplanung
- Bäume (spezielle Unterklasse von Graphen)
  - Binärbäume (z.B. für die Datenhaltung in einer Bibliothek oder einem Lexikon)
  - Binäre Suchbäume (möglichst schnelle/effiziente Suche in Binärbäumen ermöglichen)

UML Klassendiagramm (Beispiel)

Die Unified Modeling Language (vereinheitlichte Modellierungssprache) ist eine grafische Modellierungssprache zur Spezifikation, Konstruktion und Dokumentation von Software-Teilen und anderen Systemen. Im Sinne einer Sprache definiert UML dabei Bezeichner für die meisten bei einer Modellierung wichtigen Begriffe und legt mögliche Beziehungen zwischen diesen Begriffen fest.



Binärer Suchbaum (Beispiel)

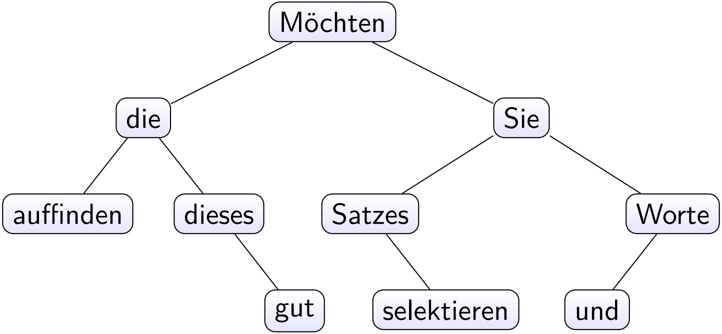
Beispielsatz: "Möchten Sie die Worte dieses Satzes selektieren und gut auffinden?"
Der Schlüssel sei das Wort selber, in alphabetischer Sortierung. Die Erzeugung des Baumes erfolgt wie folgt:

- Sei w_0 der Wurzelschlüssel.
- Sonst:
  - Falls $s < w_0$: Trage Schlüssel s im linken Teilbaum der Wurzel ein
  - Falls $s > w_0$: Trage Schlüssel s im rechten Teilbaum der Wurzel ein

Die Suche eines Eintrags im Baum erfolgt rekursiv (Suche s beginnend bei x):

- Ist s gleich dem Schlüssel x, so liefere x zurück.
- Falls $s < x$: Suche s beginnend beim linken Teilbaum von x
- Falls $s > x$: Suche s beginnend beim rechten Teilbaum von x

Binärer Suchbaum (Beispiel)



Für die Suche nach dem Wort „Sie“ werden also 2 Vergleiche benötigt.

Binärer Suchbaum (Beispiel)

Sei B ein Wurzelbaum. Der Maximalwert unter den Abständen $a(x, x_0)$ der Knoten $x \in B$ zur Wurzel x_0 heißt Länge L des Wurzelbaumes (Abstand = Anzahl Kanten). Die Anzahl der Knoten des längsten Weges zur Wurzel x_0 heißt Höhe H des Wurzelbaumes. Es gilt somit: $H = L + 1$.

Sei B ein Binärbaum mit Höhe H. Dann enthält B maximal $N = 2^H - 1$ Knoten.

Der Baum aus der vorherigen Folie hat demnach Höhe 4 und Länge 3.