

Web Algorithmen

Dr. Michael Brinkmeier

Technische Universität Ilmenau
Fakultät Informatik und Automatisierung
Institut für Theoretische Informatik
Fachgebiet Automaten und Formale Sprachen

Sommersemester 2006

Einführung

Inhalte der Vorlesung

- **Einführung**
- **Das WWW als Graph**
 - ▶ Graphentheoretische Grundlagen
 - ▶ Grundlegende Eigenschaften
 - ▶ Das Power Law
- **Ranking**
 - ▶ PageRank
 - ▶ HITS
 - ▶ Alternative Ansätze
- **Strukturen, Clustering und kohäsive Gruppen**
 - ▶ Zusammenhang
 - ▶ Clustering
 - ▶ Communities
 - ▶ Alternativen und Erweiterungen

Die Entwicklung des WWW

1696: **ARPANET** (**A**dvanced **R**esearch **P**roject **A**gency)

1974: Entwicklung von **TCP/IP** (Cerf und Kahn)

1984: **NFSNET** (US **N**ational **S**cience **F**oundation)

80er: Das **Internet** (Netz der Netze) entsteht

- exponentielles Wachstum

1990: 200.000 Rechner

1992: 1.000.000 Rechner

- Paxon (94): Das Internet verdoppelt jedes Jahr seine Größe
- Anwendungen: e-Mail, News, Remote Login, FTP

1994: Tim Berners-Lee entwickelt das WWW (neue Anwendung)

Heute: Google hat ca. 8 Milliarden URLs im Index (geschätzt)

Das WWW als Graph

Der WWW-Graph

Das WWW ist eine gerichteter **Multigraph** $D = (V, E)$

- **Knoten:** Dokumente (URL, Inhalt, Titel, Autor, MIME-Type)
- **Kanten:** Links (Anchor-Text, Position im Text, Häufigkeit)

Häufig werden URLs nach bestimmten Regeln gekappt oder geändert.

- Inhalt von Unterverzeichnissen werden identifiziert.
- Navigationslinks werden entfernt (nächste/vorherige Seite, Home etc.).
- Unter Umständen werden Seiten zu verschiedenen Themen identifiziert.

Was bietet der Graph?

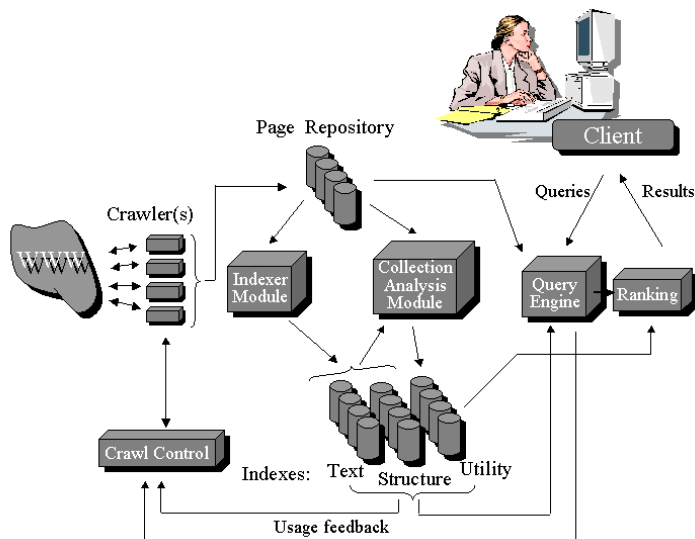
Suchmaschinen: Textuelle Indizierung der Seiten

- ermöglicht Suche nach Stichworten
- **Semantische Probleme**
 - ▶ **Synonyme:** Verschiedene Worte mit gleicher/ähnlicher Bedeutung
 - ▶ **Polynyme:** Verschiedene Bedeutungen des gleichen Wortes
- Inhaltliche Strukturierung?
- Welche Relevanz hat die Seite?

Mögliche Lösungen auf textueller Ebene:

- Kollokationen und Korellationen von Begriffen (gemeinsames Auftreten)
- Strukturierung nach Textähnlichkeit
- Graphenbasierte Strukturierung von Wort-Dokument Netzwerken (Netzwerkanalyse)

Die Architektur einer Suchmaschine



Quelle: Arasu et al., **Searching the Web**, Trans. Int. Tech., 2001

Was bietet der Graph?

Links bieten zusätzliche Informationen:

- Autoren setzen Links \Rightarrow thematischer Bezug
- Trotz unterschiedlicher Begriffswahl/Sprache, werden Beziehungen hergestellt (Strukturierung)
- Links können zur Messung der Relevanz von Seiten benutzt werden (Ranking)
- Links beeinflussen die Erfolgswahrscheinlichkeit einer Clickstream-Suche und von Crawls (Topologie)

Das WWW als Graph:

Das Power Law

Klassische Zufallsgraphen

Traditionelle Annahme: **Klassischer Zufallsgraph**

- Vorgegebene Knotenzahl n
- Kantenerzeugung (2 Alternativen):
 - ▶ Jede Kante wird unabhängig mit Wahrscheinlichkeit p eingefügt
 - ▶ Insgesamt werden m Kanten gleichverteilt eingefügt

⇒ Binomialverteilte Grade

$$P(\deg(v) = k) = \binom{n-1}{k} p^k (1-p)^{n-k}$$

⇒ relativ homogener Grad

Natürliche Netzwerke und Graphen

Seit ca. 10 Jahren werden zunehmend größere und komplexere natürliche Graphen/Netzwerke untersucht.

- Technische Netzwerke
 - ▶ Internet (Server und Router), Stromnetz, Telefonnetz
- Biologische Netzwerke
 - ▶ Nervensysteme, Metabolische Netzwerke, Nahrungsketten
- Soziale Netzwerke
 - ▶ Coacting Graph (www.oracleofbacon.org), Bekanntschaftsgraphen (Epidemiologie)
 - ▶ WWW (hybrid mit technisch)

Diese Netzwerke entsprechen **nicht** den klassischen Zufallsgraphen!

Das Power Law

Viele natürliche Netzwerke haben etwas gemeinsam:

Das Power Law

Verteilung der Eingangs- und/oder Ausgangsgrade der Knoten:

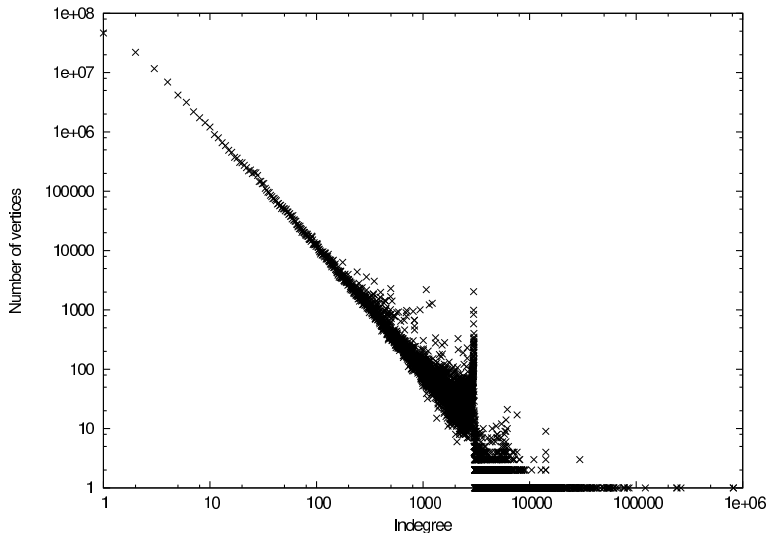
$$P(\deg(v) = k) \sim k^{-\lambda} \text{ für } \lambda > 1 \text{ konstant}$$

Diese Zusammenhang heißt **Power Law**.

Im log-log-Plot ergibt sich eine Gerade

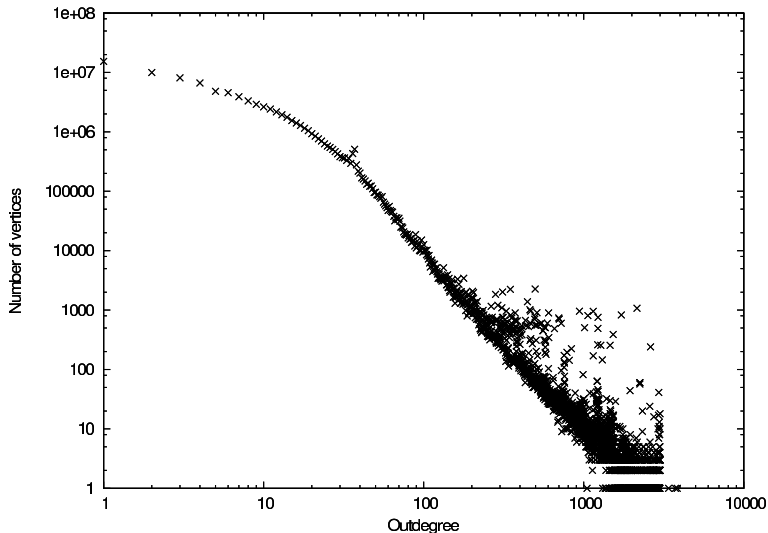
$$\log P(\deg(v) = k) = \log c - \lambda \log k$$

Die Eingangsgrade des WWW



Ermittelt aus dem WebBase Crawl von 2001

Die Ausgangsgrade des WWW



Ermittelt aus dem WebBase Crawl von 2001

Skalenfreiheit

Definition (Skalenfrei)

Ein Graph/Netzwerk $G = (V, E)$ heißt **skalenfrei (scalefree)**, falls ein $\lambda > 1$ existiert mit

$$P_k := P(\deg(v) = k) \sim k^{-\lambda} \text{ für } k \rightarrow \infty.$$

Skalenfrei, weil

$$P(\deg(v) \geq a \mid \deg(v) \geq b) = \frac{\sum_{k \geq a} P_k}{\sum_{k \geq b} P_k} \sim \left(\frac{a}{b}\right)^{-\lambda}$$

nur von a/b und nicht von a und b abhängt.

Häufig beobachtete Eigenschaften

- **Small-World:** Kurze Wege im Vergleich zur Knotenzahl
 - ▶ Milgram (1967): *Six degrees of separation*
 - ▶ Versuch der Beschreibung: $\text{diam}(D) \in O(\log n)$
- **Clustering:** Viele Regionen mit hoher „Linkdichte“
 - ▶ Viele Cliques/cliquenähnliche Teilgraphen (kohäsive Gruppen)
 - ▶ Niedrige Linkdichte zwischen den Regionen
- **Stabilität/Resilience:** Hohe Ausfallsicherheit
 - ▶ Resilience = Elastizität
 - ▶ Entfernen von Kanten/Knoten erhöht die Distanzen nur wenig.
 - ▶ Albert et al. (2000):
 - ★ Internet und WWW wenig anfällig gegen zufällige Ausfälle
 - ★ Sehr anfällig gegen gezieltes Ausschalten von Knoten mit hohem Grad

Ursachen des Power Law

Albert und Barabasi (2002) stellten Modell zur Erzeugung von Power Law Graphen vor. Seine wesentlichen Merkmale sind:

- Wachstum
- **Preferential Attachment** („bevorzugtes Anhängen“)
 - ▶ Die Wahrscheinlichkeit, dass ein neuer Knoten zu einem alten verlinkt, ist proportional zum Eingangsgard des Zieles

Neuere Modelle beinhalten ähnliche Mechanismen.

Das WWW als Graph:

Die Komponenten des WWW

Zusammenhangskomponenten

Sei $D = (V, E)$ ein gerichteter Multigraph. G_D sei der entsprechende ungerichtete Multigraph.

Definition

Eine **(Zusammenhangs-)Komponente** von G_D ist eine maximale Menge C von Knoten, so dass für $u, v \in C$ ein Weg von u nach v in G_D existiert.

Definition

Eine **schwache (Zusammenhangs-)Komponente** von D ist eine Komponente von G_D .

Tiefensuche auf G_D , beginnend bei v ergibt die schwache Komponente von D , die v enthält.

Zusammenhangskomponenten

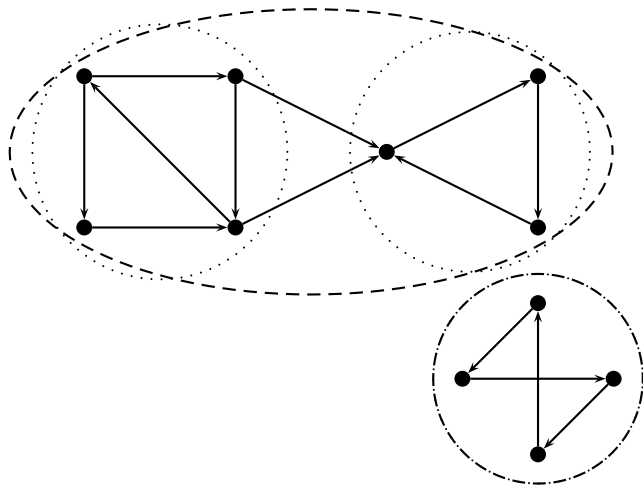
Definition

Eine **starke (Zusammenhangs-)Komponente** von D ist ein maximaler Teilgraph C , so dass jeder Knoten in C von jedem anderen Knoten in C durch einen gerichteten Weg erreichbar ist.

Die eindeutig bestimmte, starke Komponente, die v enthält, bezeichnen wir mit $[v]$.

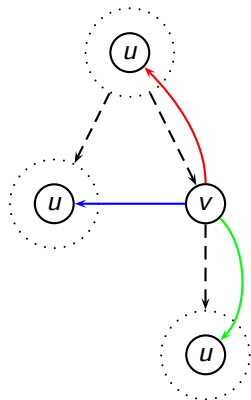
Konstruktion: Algorithmus von Tarjan (1972)

Zusammenhangskomponenten



Der Algorithmus von Tarjan

- $ord(v)$: Position von v bei einer Tiefensuche
- Drei Kantentypen (v, u) :
 - 1 $ord(v) < ord(u)$: **Vorwärtskanten**
 - 2 $ord(v) \geq ord(u)$:
 - 1 Es existiert ein Weg von u nach v : **Rückwärtskanten**
 - 2 Es existiert kein Weg von u nach v : **Seitwärtskanten**



Rückwärtskanten „erzeugen“ starke Komponenten

Der Algorithmus von Tarjan

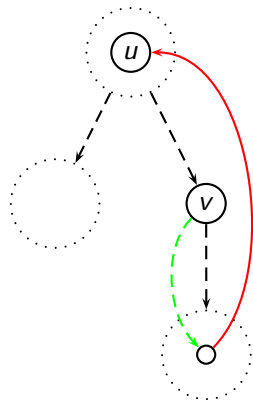
Problem: Wie können **Rückwärtskanten** und **Seitwärtskanten** unterschieden werden?

Idee

Bestimme die kleinste Nummer $l(v)$ eines Knotens, der von v aus durch eine beliebige Folge von **Vorwärtskanten** und höchstens **einer Rückwärtskante** erreicht werden kann.

Das impliziert insbesondere

$$l(v) \leq ord(v)$$

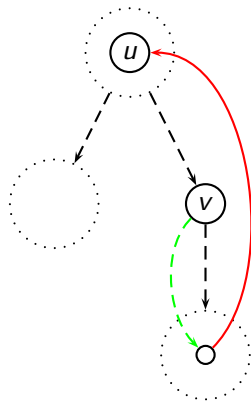


Der Algorithmus von Tarjan

Damit gilt:

$$\forall u, v : ord(u) = l(v) \Rightarrow [v] = [u]$$

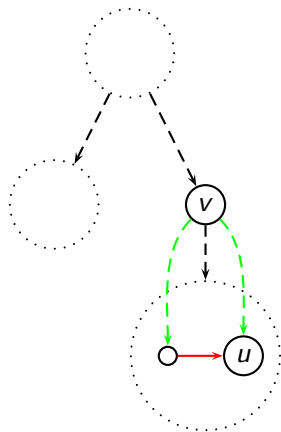
Wir können annehmen, dass die **Rückwärtskante** die letzte Kante auf dem Weg ist, denn die **Vorwärtskanten** erhöhen $ord(u)$.



Der Algorithmus von Tarjan

Betrachtet man einen Weg von v über **Vorwärtskanten** und eine **Rückwärtskante** zu einem Knoten u mit $ord(u) > ord(v)$, so sieht man, dass u auch direkt nur über **Vorwärtskanten** von v aus erreichbar ist (Tiefensuche).

Damit kann jeder aus **Vorwärts-** und **Rückwärtskanten** bestehende Weg von v zu einem Knoten u mit $ord(u) < ord(v)$ so umgestellt werden, dass er nur aus **Vorwärtskanten** und einer **Rückwärtskante** besteht.



Der Algorithmus von Tarjan

Damit erzwingt die Existenz eines Weges aus **Vorwärts-** und **Rückwärtskanten** von v zu einem Knoten u mit $ord(u) < ord(v)$ auch $l(v) < ord(v)$.

Sei nun v gegeben mit $ord(v) = l(v)$.

⇒ Für jeden Knoten u mit $ord(u) < ord(v)$, existieren nur Wege von v nach u , die aus **Vorwärts-** und **Seitwärtskanten** bestehen.

⇒ $[u] \neq [v]$

Konsequenz: Falls $l(v) = ord(v)$ gilt, ist v der erste Knoten seiner starken Komponente, der besucht wurde.

Seitwärtskanten gehen zu Knoten, deren starke Komponente bereits vollständig durchlaufen wurde.

Der Algorithmus von Tarjan

Detektion der starken Komponenten

Wenn $ord(v) = l(v)$ bei der Rückkehr zu v , dann ...

- ... ist v der erste besuchte Knoten von $[v]$.
- ... sind alle Knoten, die nach v besucht wurden und nicht bereits einer starken Komponente zugeordnet wurden, in $[v]$.

⇒ Setze $l(u)$ für alle $u \in [v]$ auf ∞ , um zu markieren, dass ihre starke Komponente bereits gefunden wurde.

Die Aktualisierung von $l(v)$

- Bei der Rückkehr zu einem Knoten v , setze $l(v)$ auf das Minimum der Werte
 - ▶ $l(u)$ für alle **Vorwärtskanten** (v, u) und
 - ▶ $ord(u)$ für alle **Rückwärtskanten** (v, u) .
- **Seitwärtskanten** sind durch $l(u) = \infty$ gekennzeichnet.

Der Algorithmus von Tarjan

Realisierung

- Globale Variable *ord* für die DFS-Nummerierung
- Feld $I[]$ zur Speicherung von $I[v]$
- Stack S zur Speicherung aller Knoten, die bereits besucht wurden und noch nicht einer starken Komponente zugeordnet wurden.
- Ein Knoten wird auf den Stack S gelegt, sobald er von der Tiefensuche erreicht wird.
- Bei Rückkehr zu v mit $ord(v) = I(v)$, stehen alle Knoten in $[v]$ vor v auf dem Stack.

Der Algorithmus von Tarjan

Prozedur tarjan(G)

Eingabe : Ein gerichteter Graph $G = (V, E)$

Beginn

| $ord = 0$

| **solange** *ein unbesuchter Knoten $v \in V$ existiert* **tue**

| | besuche(v, ord, G)

| **Ende**

Ende

Der Algorithmus von Tarjan

Prozedur besuche(v , ord , G)

Eingabe : Ein gerichteter Graph $G = (V, E)$, ein Knoten $v \in V$ und ein Integer o

Daten : Knoten Stack S , Integer Felder $I[]$ und $ord[]$

Beginn

```

Markiere  $v$  als besucht;  $S.push(v)$ 
 $ord[v] = o$ ;  $min = ord$ 
für alle Kanten  $(v, u) \in E$  tue
    |
    | wenn  $u$  unbesucht ist dann
    | |   besuche( $u$ ,  $o + 1$ ,  $G$ )
    | |    $I = I[u]$ 
    | sonst  $I = ord[u]$ 
    | wenn  $I < min$  dann  $min = I$ 
Ende
 $I[v] = min$ 
wenn  $I[v] == ord[v]$  dann
    | wiederhole
    | |    $u = S.pop()$ ;  $I[u] = \infty$ ;  $[v] = [v] \cup u$ 
    | bis  $u == v$ 
Ende
Ende

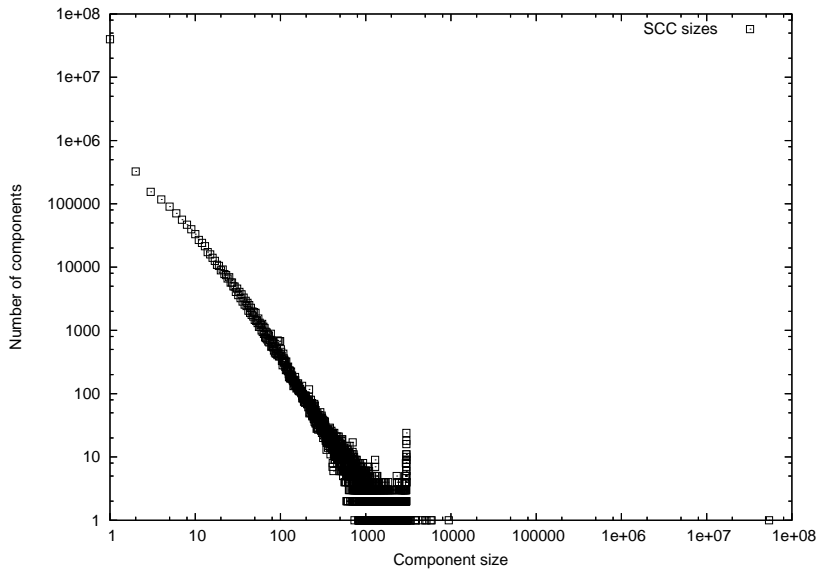
```

Der Algorithmus von Tarjan

- Offensichtlich wird `besuche` für jeden Knoten einmal aufgerufen.
- Für Knoten v benötigt `besuche` $O(1 + out(v))$ Zeit.

⇒ Gesamtzeit $O(|V| + |E|)$.

Die Starken Komponenten des WWW



Einige Zahlen

Anzahl der starken Komponenten	41 126 852
Durchschnittliche Größe	~ 2,8726
Größte starke Komponente:	53 891 939 Knoten
Zweitgrößte starke Komponente	9 428 Knoten
Drittgrößte starke Komponente	5 925 Knoten
Starke Komponenten der Größe 1	39 843 421
Starke Komponenten der Größe 2	323 994
Starke Komponenten der Größe 3	154 786

Daten wurden einem „bereinigten“ Crawl von 2001 entnommen

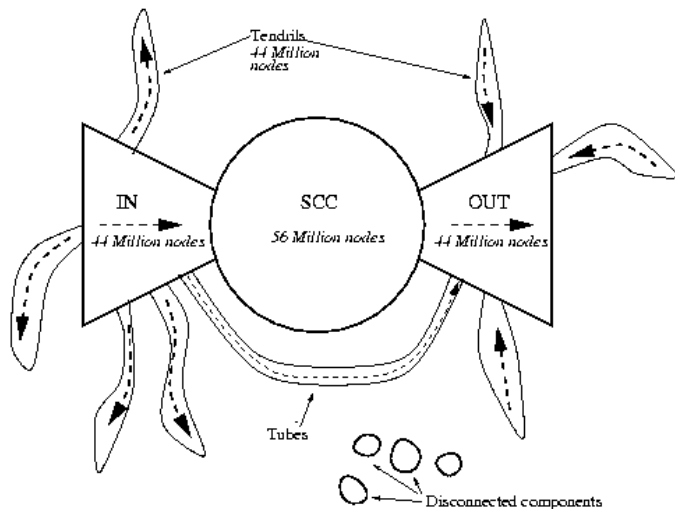
Rohdaten: WebBase Projekt, Stanford University

(<http://www.diglib.stanford.edu/testbed/doc2/WebBase/>)

Testdaten: WebGraph Projekt, Universität von Mailand

(<http://webgraph.dsi.unimi.it/>)

Die Form des WWW



Quelle: Broder et al., **Graph Structure in the Web**, WWW9, 2000