

# Algorithmen und Datenstrukturen

Datenstrukturen: Anordnung von Daten, z.B.

- als Liste (d.h. in bestimmter Reihenfolge)

Beispiel: alphabetisch sortiertes Wörterbuch

... *Ei* - *Eibe* - *Eidotter* ...

- als Baum (d.h. aufgrund von Abhängigkeitsstrukturen)

Beispiele: morphologisch sortiertes Wörterbuch

*Ei* - *Eibe*  
*Eidotter* *Eibenhholz*

syntaktische  
Strukturbeschreibung

semantische Hierarchie

*Pflanze* - *Baum* - *Nadelbaum*  
- *Eibe*  
- *Laubbaum* -  
*Birke*

- als Netz (d.h. aufgrund von Zusammenhängen)

Beispiel: nach Bedeutungen sortiertes Wörterbuch

("onomasiologische  
Sortierung")

*Huhn*  
|  
*Ei* - *Frühstück*  
|  
*Ostern*

Algorithmen:

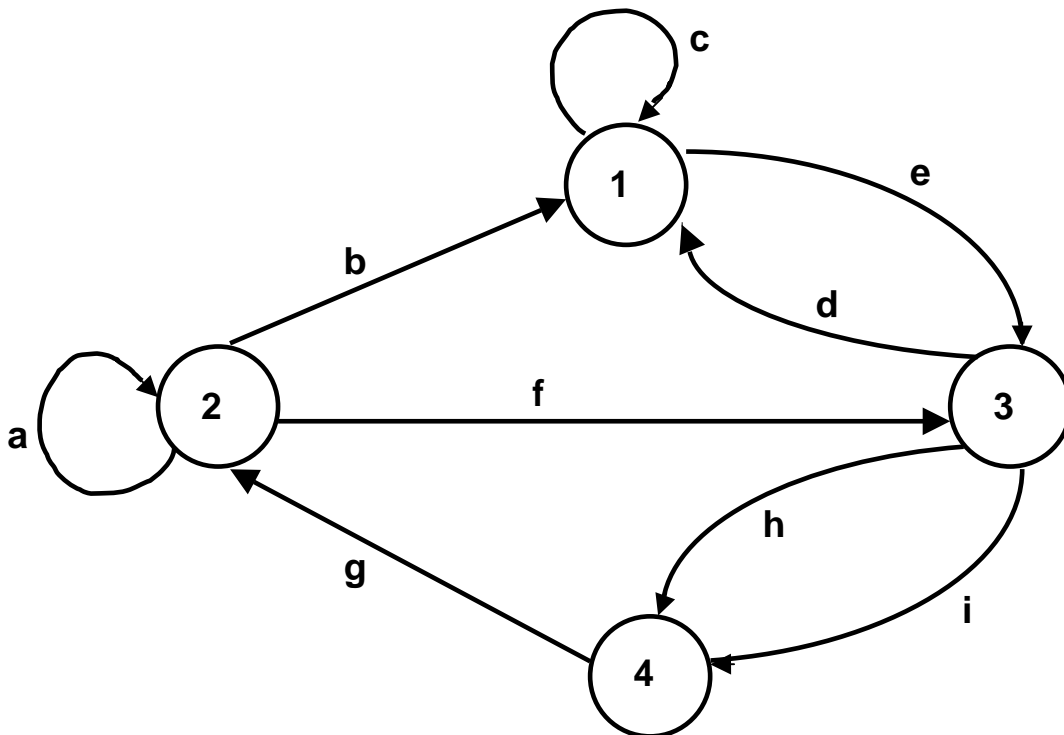
exakt beschriebene Verfahren zur Verarbeitung von Daten mit einer bestimmten Datenstruktur

- Suchen eines bestimmten Elements  
(z.B. in einer alphabetisch sortierten Liste)
- Sortieren der Daten zur Beschleunigung des Suchens
- Löschen alter und Einfügen neuer Daten

## Allgemeine Terminologie: Bäume und Netze als Graphen

- Graphen bestehen aus einer nichtleeren Menge von Knoten  $V$  (engl. Vertex) und einer Menge von Kanten  $E$  (engl. Edge), wobei die Schnittmenge von  $V$  und  $E$  leer ist, sowie einer Abbildung  $\varphi : E \rightarrow V \times V$
- eine Kante wird durch die beiden Knoten festgelegt, die sie verbindet.
- ein Knoten wird durch die Kanten festgelegt, die von ihm ausgehen (und damit auch durch die anderen Knoten, mit denen er direkt verbunden ist)
- im allgemeinen können beliebig viele Kanten von einem Knoten ausgehen, aber jede Kante verbindet nur zwei Knoten
- Kanten können gerichtet oder ungerichtet sein; bei gerichteten Kanten spricht man auch von gerichteten Graphen und von Eingangs- bzw. Ausgangskanten eines Knotens
- Kanten und Knoten können Attribute besitzen. Man spricht dann von bewerteten bzw. gewichteten Kanten und Knoten.
- Spezialfall: Bäume: gerichtete Graphen, bei denen jeder Knoten bis auf einen (die Wurzel) genau eine Eingangskante hat
- Zusammenhängende Graphen, die keine Bäume sind, werden oft als Netze bezeichnet. Sie enthalten in der Regel Zyklen.
- Graphen sind abstrakte Objekte, bei denen die räumliche Anordnung der Knoten und die Länge der Kanten keine Rolle spielt (anders als z.B. bei topographischen Landkarten)

Ein Beispiel für einen gerichteten Graphen:



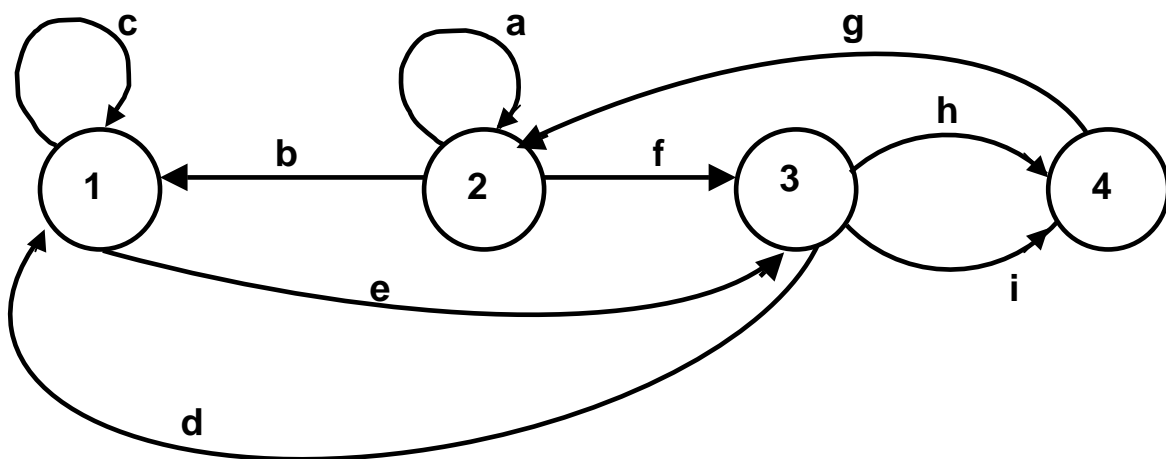
Formale Definition dieses Graphen:

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

$$E = \{ a, b, c, d, e, f, g, h, i, \}$$

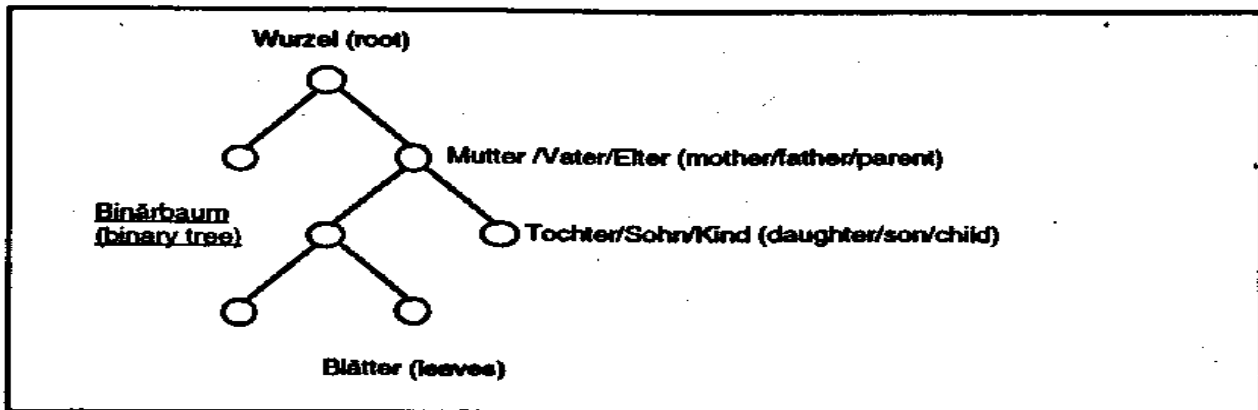
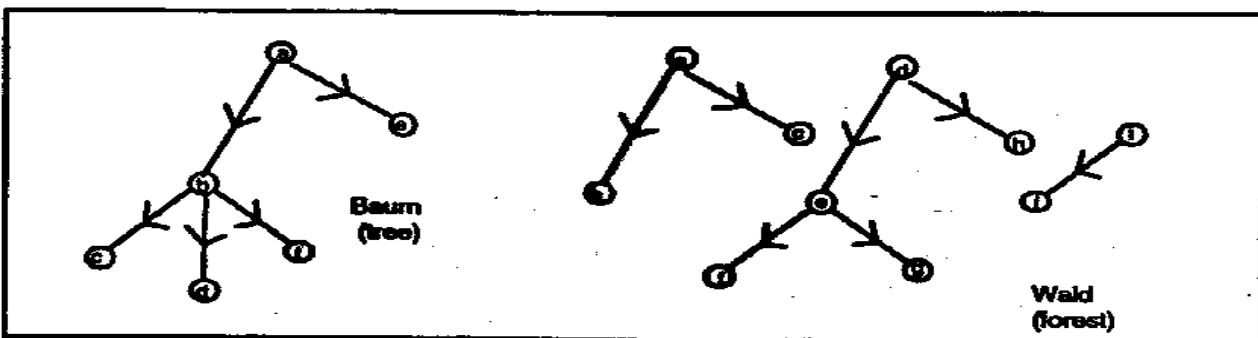
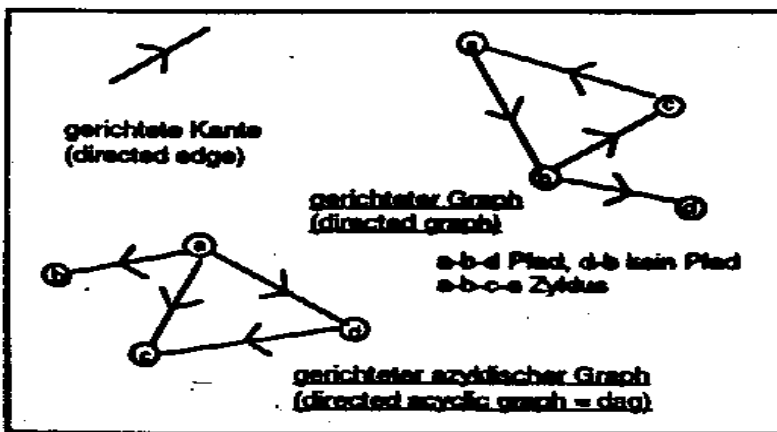
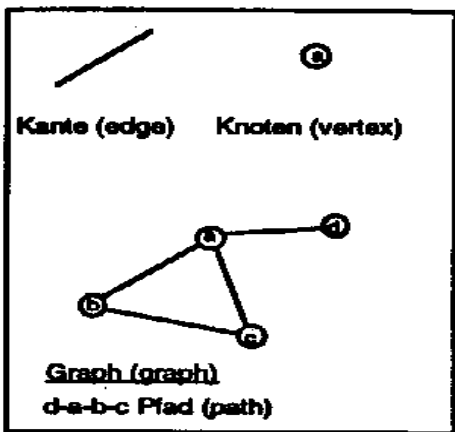
	a	b	c	d	e	f	g	h	i
$\varphi$	(2,2)	(2,1)	(1,1)	(3,1)	(1,3)	(2,3)	(4,2)	(3,4)	(3,4)

Dieser Graph kann auch anders visualisiert werden als oben:



# Graphen und Bäume - grundlegende Begriffe

(vergl. auch Sedgewick Kapitel 4 u. 29-34)



# Tiefensuche und Breitensuche in Bäumen

Problem: Gesucht wird ein bestimmtes Knotenetikett. Begonnen wird bei der Wurzel, aber jeder Knoten eines Baumes hat in der Regel mehrere Nachfolger. Wie soll die Suche weitergehen?

Beispiel: Suche nach bestimmter syntaktischer Struktur in einer Sammlung syntaktisch analysierter Sätze (Treebank)

## Möglichkeiten:

- Vorläufige Beschränkung auf einen Nachfolgerknoten (Tiefensuche)
- Gleichzeitige Verfolgung aller Nachfolgerknoten (Breitensuche)

## Tiefensuche:

einer der Nachfolger eines Knotens wird ausgewählt (z.B. der erste), danach einer von dessen Nachfolgern etc. Steht kein Nachfolger mehr zur Verfügung, geht der Algorithmus zurück zum letzten Knoten, dessen Nachfolger noch nicht alle untersucht wurden (Backtracking).

## Breitensuche:

alle Nachfolger eines Knotens werden untersucht; anschließend wird einer (z.B. der erste) ausgewählt und dessen Nachfolger untersucht. Danach kommen erst die Nachfolger des zweiten usw. dran. Mit der Verarbeitung der nächsten Ebene wird erst begonnen, wenn alle Knoten auf einer Ebene besucht worden sind.

Sowohl Tiefensuche als auch Breitensuche durchlaufen den vollständigen Baum (wenn nicht die Suche vorher erfolgreich abgebrochen wird). Allerdings werden die Knoten in anderer

Reihenfolge besucht: Breitensuche erreicht die Blätter eines Baums erst sehr spät.

## Tiefensuche und Breitensuche beim Parsing

Bei der Syntaxanalyse entsteht immer wieder die Frage, welche von mehreren anwendbaren Regeln ausgewählt werden soll.

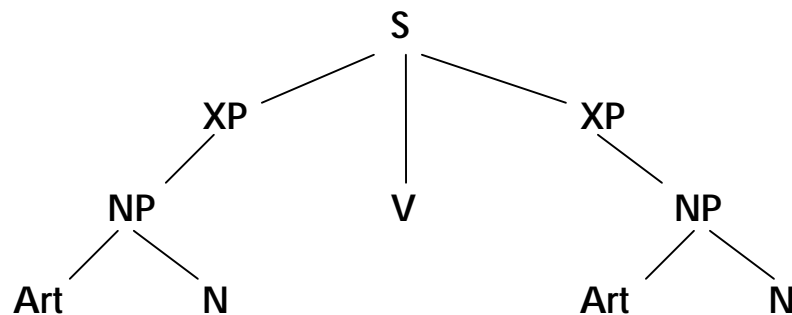
Beispiel:

Top-down-Analyse:  $S \rightarrow XP V XP$  vs.  $S \rightarrow XP V XP XP$

Bottom-up-Analyse:  $NP \rightarrow Art N$  vs.  $NP \rightarrow Art N XP$

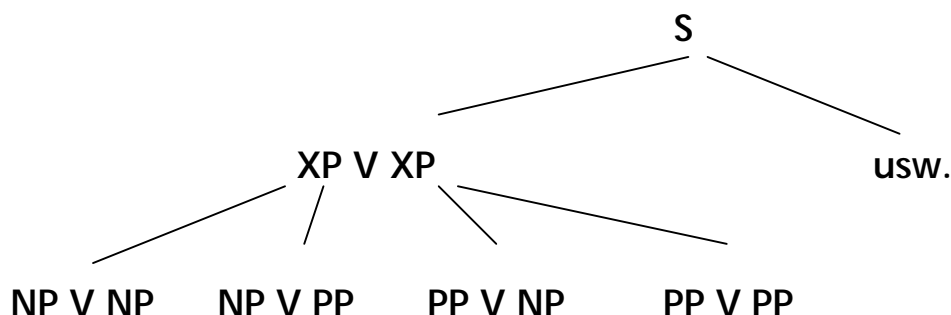
Dieser Entscheidungsprozess lässt sich in Form eines Baums darstellen: Jede Entscheidungssituation bildet einen Knoten, jede mögliche Regelanwendung einen Nachfolger.

Beispiel: *Diese Bäume illustrieren die Suchprozesse.*



Die Top-down-Analyse in der oben dargestellten Form entspricht also einer Tiefensuche in diesem Analysebaum.

Möglich wäre auch eine Top-down-Breitensuche oder eine Mischform, bei der sämtliche Nichtterminalsymbole gleichzeitig expandiert werden:



Allerdings wären solche Verfahren wesentlich schwieriger zu programmieren.