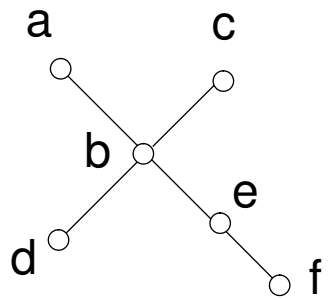
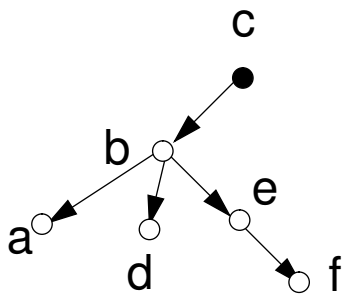
4.4 Modellierung mit Bäumen

In einem **ungerichteten Baum** gibt es
zwischen zwei beliebigen Knoten genau einen Weg.



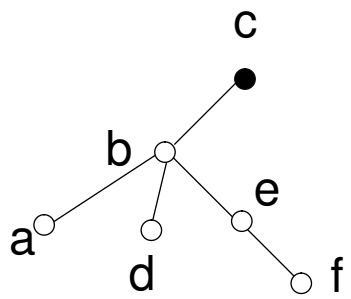
Ein gerichteter, **azyklischer** Graph G ist ein **gerichteter Baum**, wenn alle Knoten einen **Eingangsgrad** $\leq$ **1** haben und es genau einen Knoten mit **Eingangsgrad 0 gibt,**
er ist die Wurzel von G. G ist ein **gewurzelter Baum**.



Wurzel

Man kann aus einem **ungerichteten Baum** in eindeutiger Weise einen gerichteten machen, indem man **einen Knoten zur Wurzel bestimmt**.

Deshalb wird in gewurzelten Bäumen häufig die **Kantenrichtung nicht angegeben**.



In einem gewurzelten Baum ist die **Höhe eines Knotens v** die größte Länge eines Weges von v zu einem Blatt.
Die Höhe der Wurzel heißt **Höhe des Baumes**.

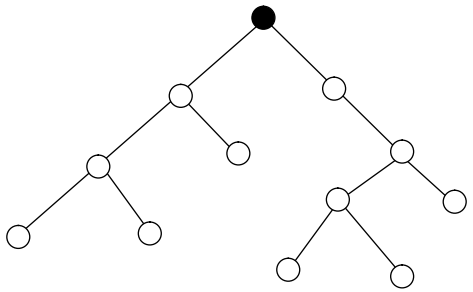
Knoten, die weder Wurzel noch Blatt sind heißen **innere Knoten**.

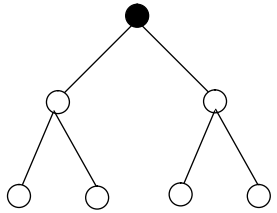
© 2001 bei Prof. Dr. Uwe Kastens

Binärbäume

Ein gewurzelter Baum heißt **Binärbaum**, wenn
seine Knoten einen **Ausgangsgrad von höchstens 2** haben.

Ein **Binärbaum heißt vollständig**, wenn jeder Knoten außer den Blättern den **Ausgangsgrad 2** hat und die **Wege zu allen Blättern gleich lang** sind.





Höhe 2

Knoten: 7

Blätter: 4

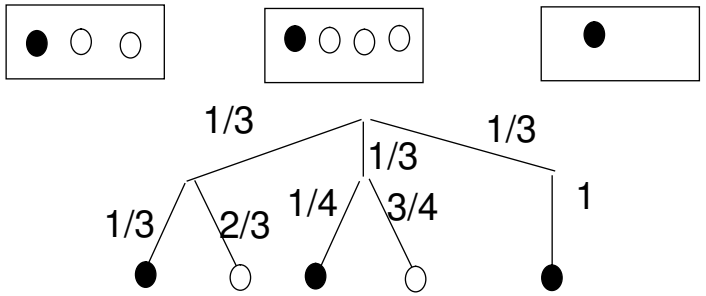
Ein vollständiger **Binärbaum der Höhe h** hat **2^h Blätter** und **2^{h+1}-1 Knoten**

© 2001 bei Prof. Dr. Uwe Kastens

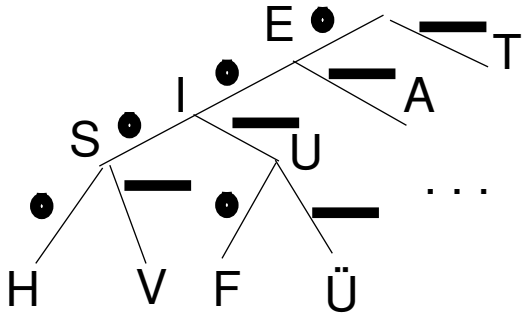
Modellierung von Entscheidungsbäumen

Knoten modelliert **Zwischenstand** einer mehrstufigen Entscheidungsfolge
Kante modelliert eine der wählbaren **Alternativen**

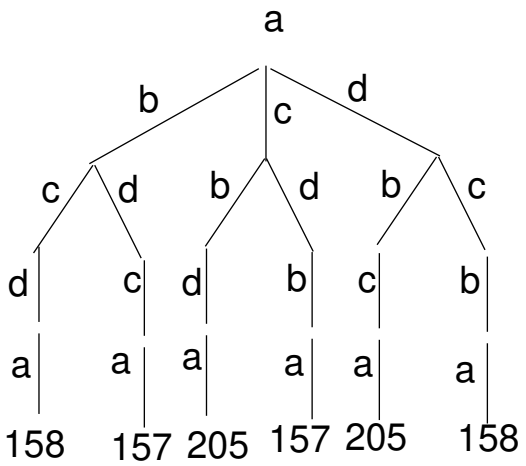
1. **Wahrscheinlichkeiten**,
   z. B. erst Schachtel, dann Kugel ziehen:



2. **Codierungen**,
   Z. B. Morse-Code



3. **Lösungsbaum** für kombinatorische Probleme,
   z. B. Traveling Salesman's Problem (Mod-4.12)
   Blätter repräsentieren einen Rundwege von a aus,
   Kanten sind mit Entscheidungen markiert



4. **Spielzüge**, z. B. Schach (ohne Bild)

Wird **derselbe Zwischenstand** durch verschiedene Entscheidungsfolgen erreicht,
kann man **Knoten identifizieren**.
Es entsteht ein azyklischer oder zyklischer Graph.

© 2001 bei Prof. Dr. Uwe Kastens

Modellierung von Strukturen durch Bäume

Knoten modelliert ein **Objekt**.
Kante modelliert **Beziehung „besteht aus“**, „enthält“, „spezialisiert zu“, ...

Beispiele:

- **Typhierarchie**: Typ - Untertypen
- **Klassenhierarchie**: Oberklasse als Abstraktion ihrer Unterklassen (Mod-4.19a)
  Vererbungshierarchie: Unterklassen erben von ihrer Oberklasse
- **Objektbaum**: Objekt enthält (Referenzen auf) Teilobjekte
- **Kantorowitsch-Baum**: Operator mit seinen Operanden (Mod-4.19b)
- **Strukturbaum**: (Programm-)Struktur definiert durch eine kontextfreie Grammatik (Mod-4.19c)

Identifikation gleicher Teilbäume führt zu azyklischen Graphen (DAGs).

Vorsicht:

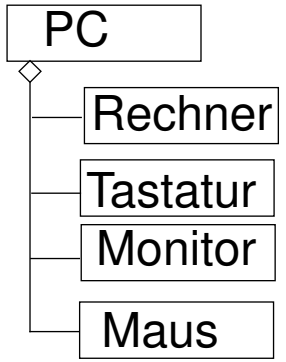
Identifikation muss mit der **Bedeutung der Kanten verträglich** sein;
z. B. Ein Gegenstand kann nicht dasselbe Objekt mehrfach als Teil enthalten, wohl aber mehrere Objekte derselben Art.

© 2001 bei Prof. Dr. Uwe Kastens

Klassen- und Objekthierarchien

Kompositionsbeziehung im Klassendiagramm (UML Notation):

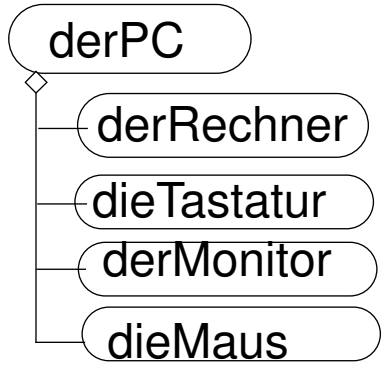
Knoten: **Klassen**
Kanten: definieren, **aus welcher Art von Objekten** ein Objekt besteht
z. B. ein Objekt der Klasse PC besteht aus
einem Rechner-Objekt, einem Tastatur-Objekt, ...



Diese Beziehung zwischen den Klassen könnte
auch ein allgemeiner Graph sein

Objektbaum im Objektdiagramm (fast UML):

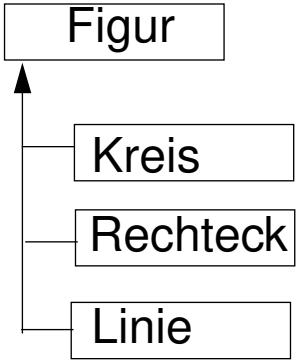
Knoten: **Objekte**
Kanten: definieren, **aus welchen Objekten** ein Objekt besteht
z. B. dieser PC besteht aus, diesem Rechner, ...



Diese Beziehung muss **konzeptionell ein Baum** sein.

Vererbungsbeziehung im Klassendiagramm (UML Notation):

Knoten: **Klassen**
Kanten: Unterklasse **erbt von** -> Oberklasse
Oberklasse **ist Abstraktion** <- ihrer Unterklassen
Kanten sind zur Wurzel hin gerichtet



Baum bei Einfachvererbung (Java)
azyklischer Graph bei Mehrfachvererbung (C++)

© 2000 bei Prof. Dr. Uwe Kastens

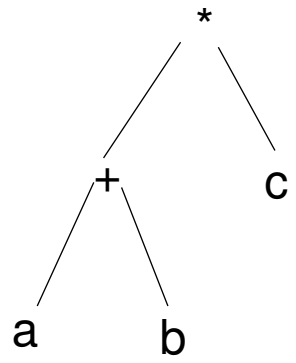
Kantorowitsch-Bäume

Darstellung der Struktur von Termen, Formeln, Ausdrücken
(siehe Mod-2.19)

Knoten: Operator

Kanten: Verbindung zu den Operanden eines Operators
Die Kanten sind geordnet (Kantenmarkierung):
erster, zweiter, ... Operand

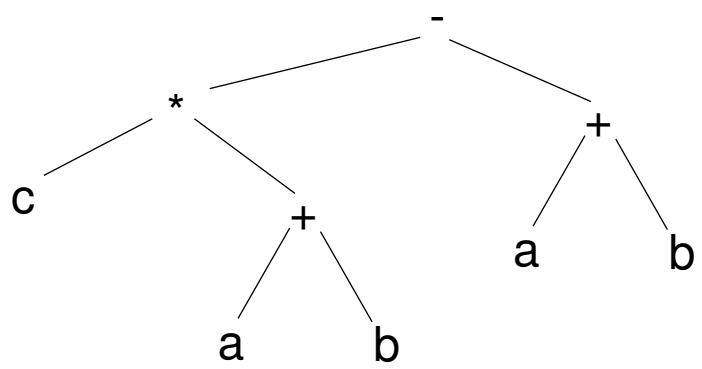
(a + b) * c

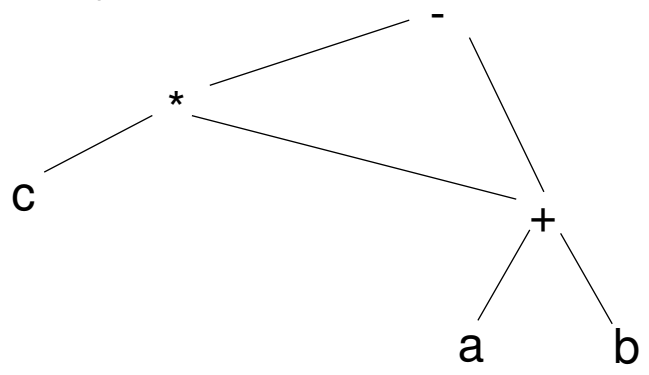


Identifikation gleicher Teilbäume führt zu azyklischen Graphen (DAGs):

Z. B. identifizieren Übersetzer gleiche Teilbäume, um Code zu erzeugen,
der sie nur einmal auswertet:

c * (a + b) - (a + b)





© 2001 bei Prof. Dr. Uwe Kastens

Strukturbäume zu kontextfreien Grammatiken

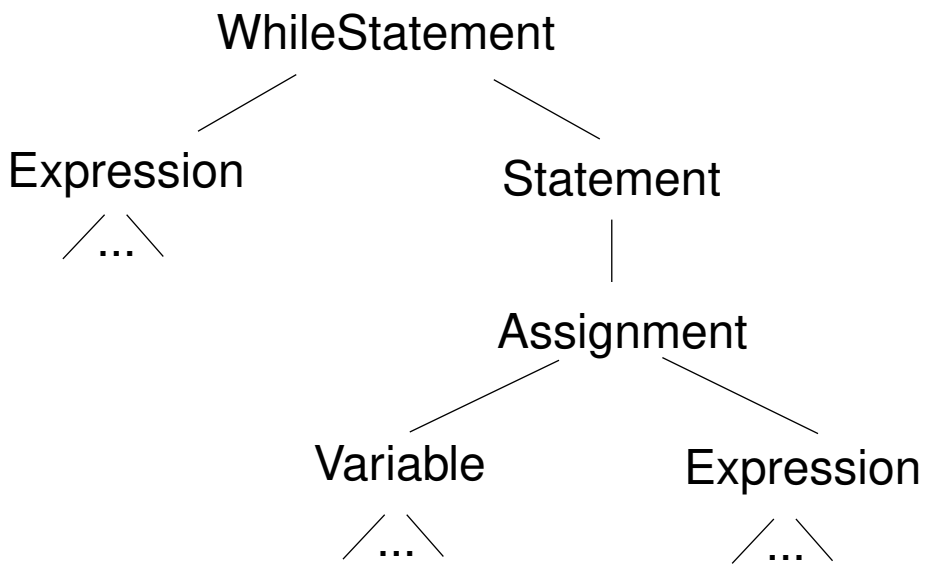
Kontextfreie Grammatiken definieren die Struktur von Programmen, Texten oder Daten.
Ein Programm, Text oder strukturierte Daten werden als Strukturbaum dargestellt.

Knoten: Programmkonstrukt (Nichtterminal der Grammatik)

Kante: Bezug zu Bestandteilen des Programmkonstruktes (Produktion der Grammatik)

Für die Repräsentation von Texten sind die **Kanten geordnet** (Kantenmarkierung)

Strukturbaum:




```
while(i<10)   a[i]    =     2*i    ;
```

Produktionen aus der kontextfreien Grammatik:

Statement ::= Assigment

Statement ::= WhileStatement

...

WhileStatement ::= Expression Statement

Assignment ::= Variable Expression

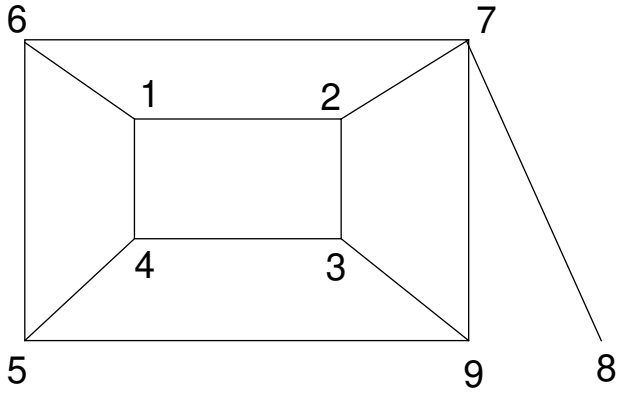
© 2001 bei Prof. Dr. Uwe Kastens

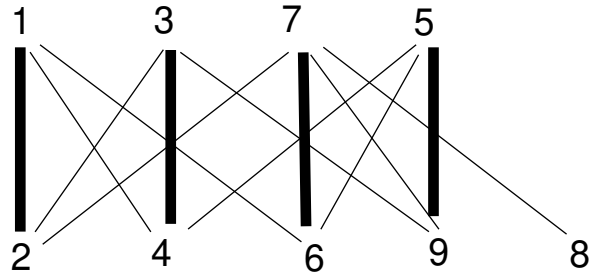
4.5 Zuordnungsprobleme

Aufgabenklasse **paarweise Zuordnung (Matching):**

Im ungerichteten Graphen G = (V, E) modelliert eine Kante {a, b} „a passt zu b“, ggf. mit einer Kantenmarkierung als Abstufung

Gesucht ist eine **maximale Menge unabhängiger Kanten**,
das ist einTeilgraph M mit allen Knoten aus V und möglichst vielen Kanten aus E,
so dass der **Grad der Knoten höchstens 1** ist. M heißt ein **Matching** der Knoten von G.





bipartiter Graph

Matching

Graph G heißt **bipartit**, wenn V in **2 disjunkte Teilmengen $V = V_1 \cup V_2$** zerlegt werden kann, so dass jede Kante zwei Knoten aus verschiedenen Teilmengen verbindet.

Häufig liefert die Aufgabenstellung schon bipartite Graphen, sogenannte **Heiratsprobleme**:

Mann - Frau Aufgabe - Bearbeiter Verbraucher - Produkte

© 2001 bei Prof. Dr. Uwe Kastens

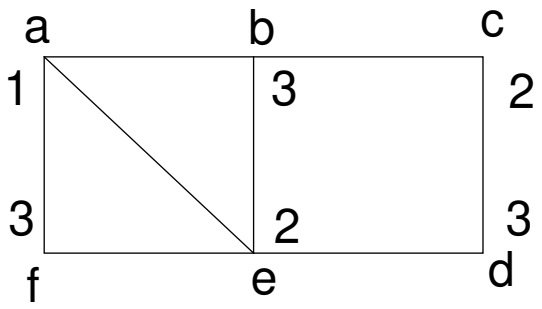
Konfliktfreie Knotenmarkierung (Färbung)

Aufgabenklasse **konfliktfreie Knotenmarkierung (Färbung):**

Im ungerichteten Graphen G = (V, E) modelliert eine Kante {a, b} „a ist unverträglich mit b“,

Gesucht ist eine Knotenmarkierung Färbung: V -> $\mathbb{N}$ („Farben“),
so dass durch eine Kante verbundene Knoten verschiedene Marken haben

Die chromatische Zahl eines Graphen G ist die minimale Zahl verschiedener „Farben“, die nötig ist, um G konfliktfrei zu markieren.



Es gilt: chromatische Zahl ≤ 1 + maximaler Knotengrad

Anwendungen:

Knoten:	**Kante:**	**Farbe / Marke:**
Staat auf Landkarte	gemeinsame Grenze	Farbe
Partygast	unverträglich	Tisch
Kurs	haben gemeinsame Teilnehmer	Termin
Prozess	benötigen gleiche Ressource	Ausführungszeitpunkt
Variable im Programm	gleichzeitig lebendig	Registerspeicher

© 2000 bei Prof. Dr. Uwe Kastens

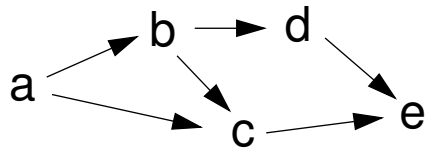
4.6 Abhängigkeitsprobleme

Graphen modellieren **Abhängigkeiten** zwischen Operationen und **Ausführungsreihenfolgen** von Operationen.

Abhängigkeitsgraph: gerichtet, azyklisch, voneinander abhängige Operationen.

Aufgaben dazu: sequentielle oder parallele **Anordnungen finden** (engl. **scheduling**).

Knoten: **Operation**, Ereignis; ggf. mit Dauer markiert
Kante: a -> b a ist **Vorbedingung** für b oder
b **benutzt** Ergebnis von a oder
a liest oder schreibt Ressource
bevor b sie überschreibt

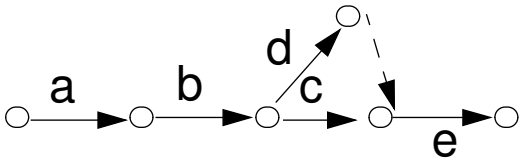


Anwendungen:

- **Projektplanung** mit abhängigen Teilaufgaben (PERT, CPM)
- abhängige **Transaktionen** mit einer Datenbank
- **Anordnung von Code** für die parallele Auswertung von Ausdrücken (Übersetzer)

Kritischer Pfad: längster Weg von einem Anfangsknoten zu einem Endknoten

Duale Modellierung:
Knoten: Ereignis, Anfang und Ende einer Operation
Kante: Operation, ggf. mit Dauer markiert

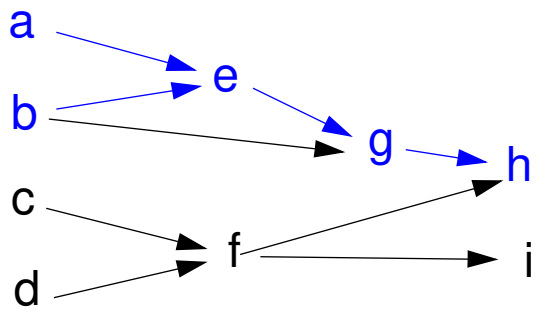


© 2002 bei Prof. Dr. Uwe Kastens

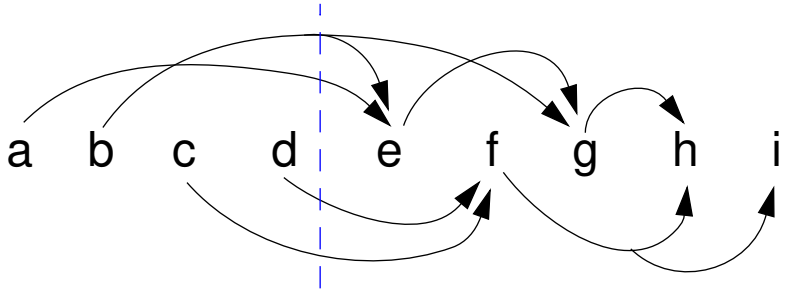
Anordnung von Abhängigkeitsgraphen

Anordnungsaufgaben:
gegebener Abhängigkeitsgraph

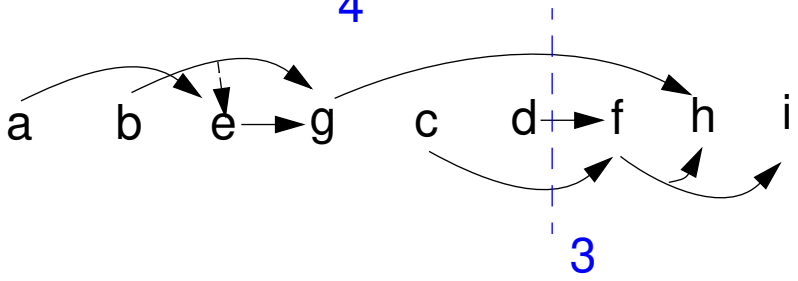
kritischer Pfad



sequentielle Anordnung der Knoten,
so dass **alle Kanten vorwärts** zeigen.

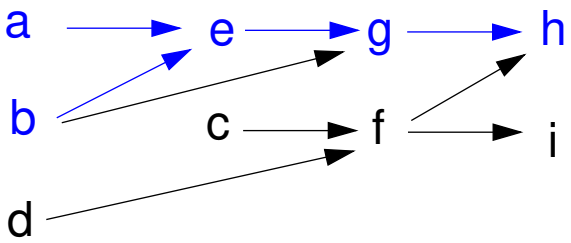


Meist sollen **Randbedingungen** erfüllt werden,
z. B. geringste Anzahl gleichzeitig benötigter Zwischenergebnisse im Speicher



parallele Anordnung mit
beschränkter Parallelität 3

Länge: 4 Schritte (Operationen)



© 2001 bei Prof. Dr. Uwe Kastens

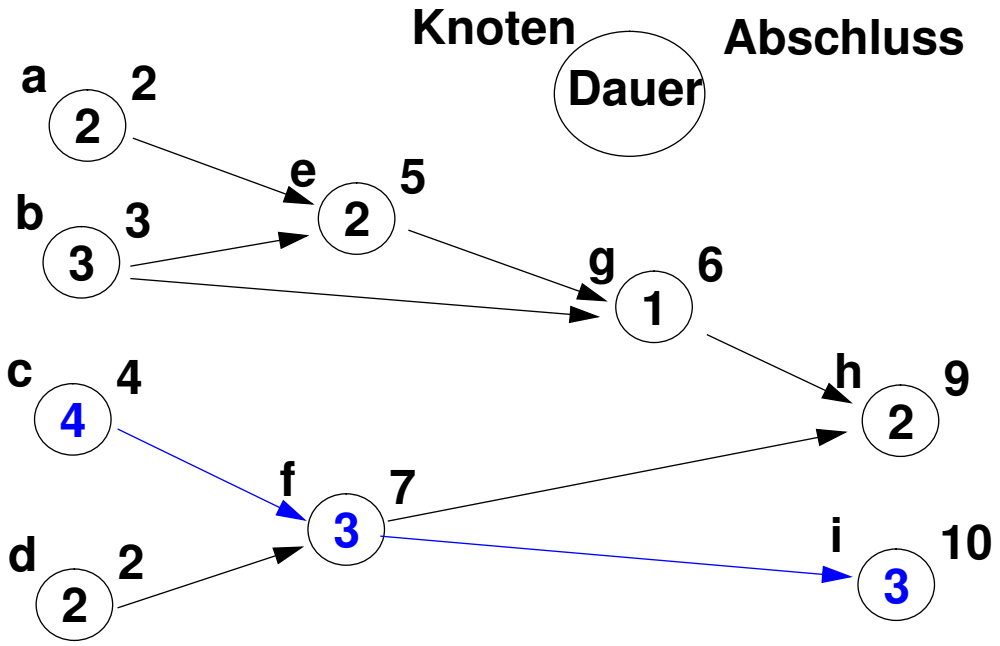
Operationen unterschiedlicher Dauer

Zwei **Knotenmarkierungen**:

Dauer der Operation und

frühester Abschlusstermin
= max. Abschluss der Vorgänger
+ Dauer des Knotens

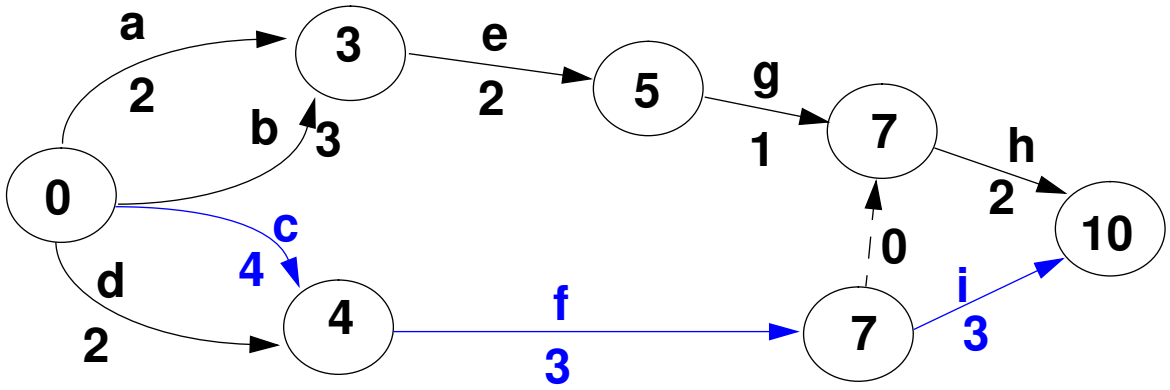
Kritischer Pfad gemäß maximaler Summe der Dauer der Operationen



Duale Modellierung:

Kante: Operation
mit **Dauer** als Marke
Mehrfachkanten, Multigraph

Knoten: Ereignis
„vorangehende Operationen sind abgeschlossen“
mit frühestem **Abschlusstermin** als Marke



© 2002 bei Prof. Dr. Uwe Kastens

Ablaufgraphen

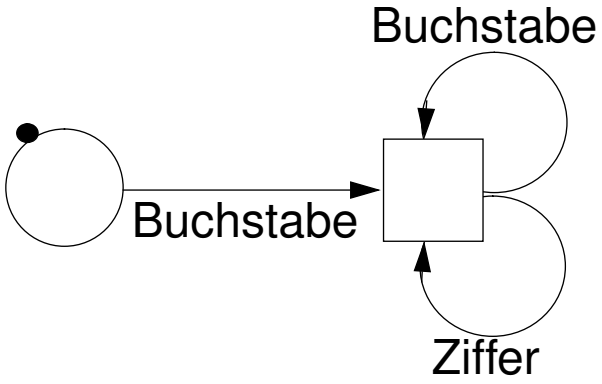
Gerichteter Graph (auch zyklisch) modelliert Abläufe.

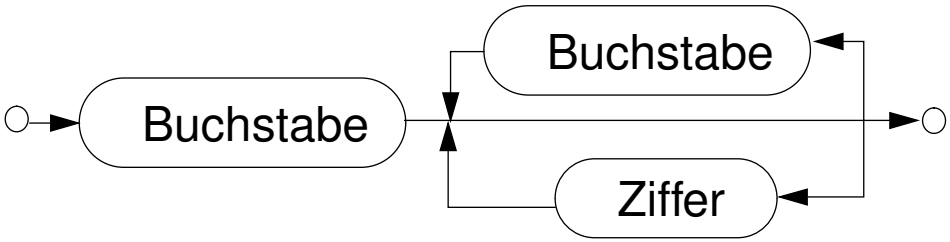
Knoten: Verzweigungsstelle, Zustand
Kanten: Fortsetzungsmöglichkeit

Jeder **Weg durch den Graphen** beschreibt einen **potenziellen Ablauf**
Die **Folge der Markierungen eines Weges** kann einen **Satz einer Sprache** modellieren.

Anwendungen:

- **Endlicher Automat** (siehe Kapitel 6)
  modelliert **Folgen von Zeichen**, Symbolen, ...
  Knoten:Zustand
  Kante: Übergang markiert mit Zeichen
- **Syntaxdiagramm**
  modelliert **Folgen von Zeichen**, Symbolen, ...
  Knoten: markiert mit Zeichen
  Kante: „kann folgen auf“
  dual zum endlichen Automaten
- **Aufrufgraphen** (siehe Mod-4.26)
- **Ablaufgraphen** (siehe Mod-4.27)





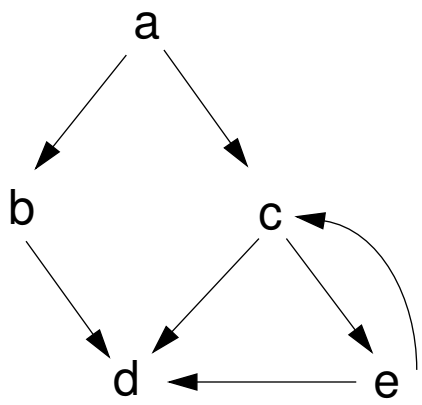
© 2002 bei Prof. Dr. Uwe Kastens

Aufrufgraphen

Gerichteter Aufrufgraph: **Aufrufbeziehung zwischen Funktionen** in einem Programm; wird benutzt in **Übersetzern** und in **Analysewerkzeugen** zur Software-Entwicklung.

Knoten: **Funktion** im Programm
Kante a -> b: Rumpf der Funktion a enthält einen Aufruf der Funktion b; **a könnte b aufrufen**

Zyklus im Aufrufgraph:
Funktionen, die sich **wechselweise rekursiv** aufrufen, z. B. (c, e, c)



Fragestellungen z. B.

- Welche Funktionen sind nicht rekursiv?
- Welche Funktionen sind nicht (mehr) erreichbar?
- Indirekte Wirkung von Aufrufen,
  z. B. nur e verändere eine globale Variable x;
  welche Aufrufe lassen x garantiert unverändert? b, d

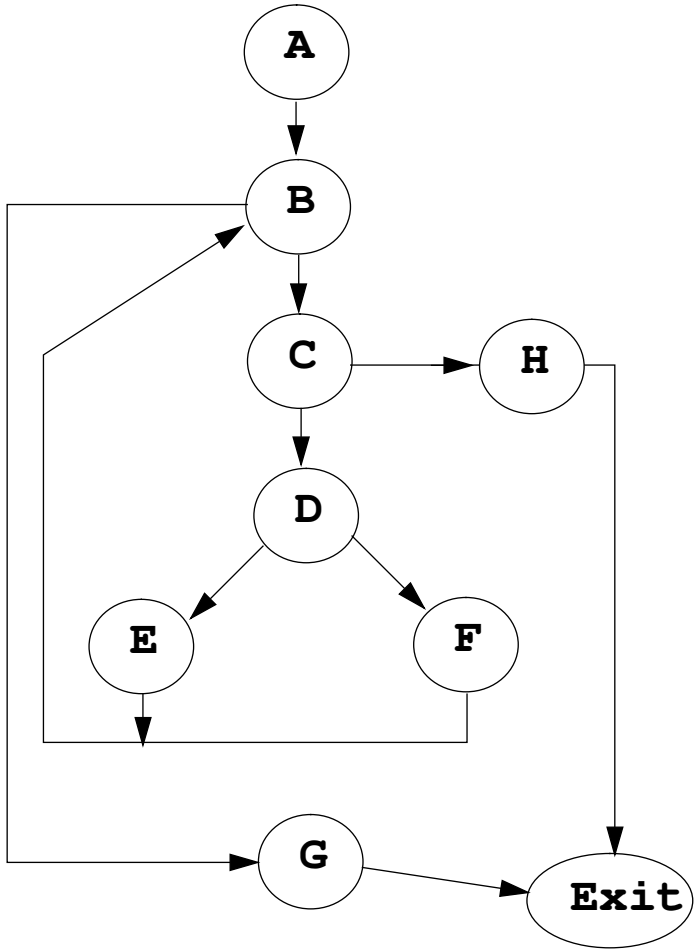
© 2001 bei Prof. Dr. Uwe Kastens

Programmablaufgraphen

Gerichteter Graph, modelliert **Abläufe durch ein verzweigtes Programm** (bzw. Funktion); wird benutzt in **Übersetzern** und in **Analysewerkzeugen** zur Software-Entwicklung.

Knoten: **unverzweigte Anweisungsfolge** (Grundblock), mit Verzweigung (Sprung) am Ende
Kante: **potenzieller Nachfolger im Ablauf**

```
ug = 0;                                  A
og = obereGrenze;
while (ug <= og)                         B
{   mitte = (ug + og) / 2;               C
    if (a[mitte] == x)
           return mitte;                 H
    else if (a[mitte] < x)               D
           ug = mitte + 1;               E
    else   og = mitte - 1;               F
}
return nichtGefunden;                    G
```



Fragestellungen, z. B.

- Menge von Wegen, die den **Graph überdecken**, **Software-Testen**
- Wege mit bestimmten Eigenschaften, **Datenflussanalyse**

© 2002 bei Prof. Dr. Uwe Kastens

Zusammenfassung zu Graphen

Problemklassen:

- Wegeprobleme
- Verbindungsprobleme
- Entscheidungsbäume
- hierarchische Strukturen
- Zuordnungsprobleme
- Abhängigkeitsprobleme
- Anordnungen in Folgen
- verzweigte Abläufe

Kanten- und Knotenbedeutung:

- verbunden, benachbart, ...
- Entscheidung, Alternative, Verzweigung
- Vorbedingung, Abhängigkeit
- (Un-)Verträglichkeit
- allgem. symmetrische Relation
- besteht aus, enthält, ist-ein
- (Halb-)Ordnungsrelation

Kanten-, Knotenmarkierungen:

- Entfernung, Kosten, Gewinn, ... bei Optimierungsproblemen
- „Färbung“, disjunkte Knotenmengen bei Zuordnungsproblemen
- Symbole einer Sprache

© 2001 bei Prof. Dr. Uwe Kastens