Seminar aus maschinellem Lernen

Vortrag 14.01.2009

TECHNISCHE UNIVERSITÄT DARMSTADT

Graph-Based Hierarchical Conceptual Clustering

Marco Ghiglieri

Prof. Dr. Johannes Fürnkranz

Agenda

TECHNISCHE UNIVERSITÄT DARMSTADT

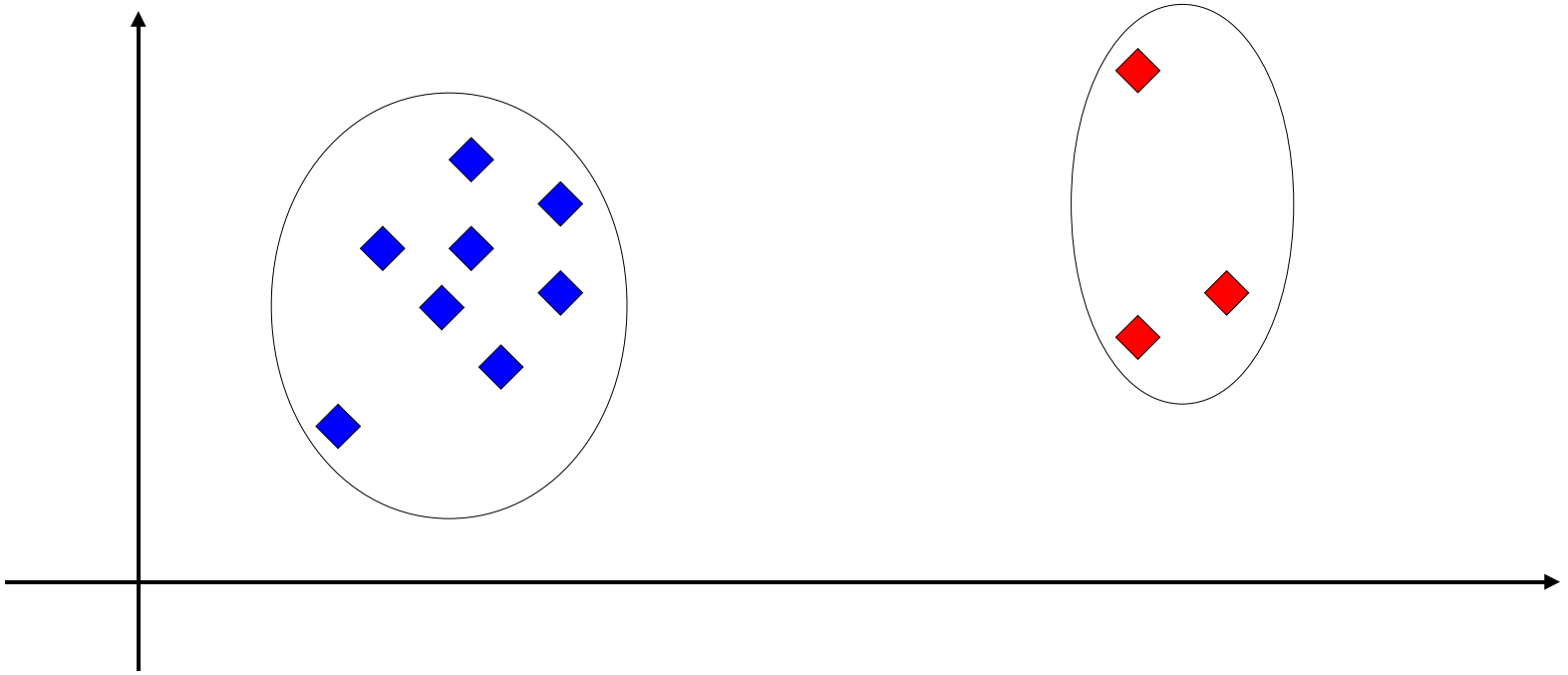
- Einleitung
- Überblick SUBDUE
- Hierarchisches konzeptuelles Clustering in SUBDUE
- Evaluierung

Was ist Clustering ?

TECHNISCHE UNIVERSITÄT DARMSTADT

Gegeben: Menge von ungelabelten Beispielen (=unsupervised)

Ziel: Gruppierung in aussagekräftige Einheiten (=Cluster),
d.h. Elemente in einem Cluster haben hohe Ähnlichkeit zueinander

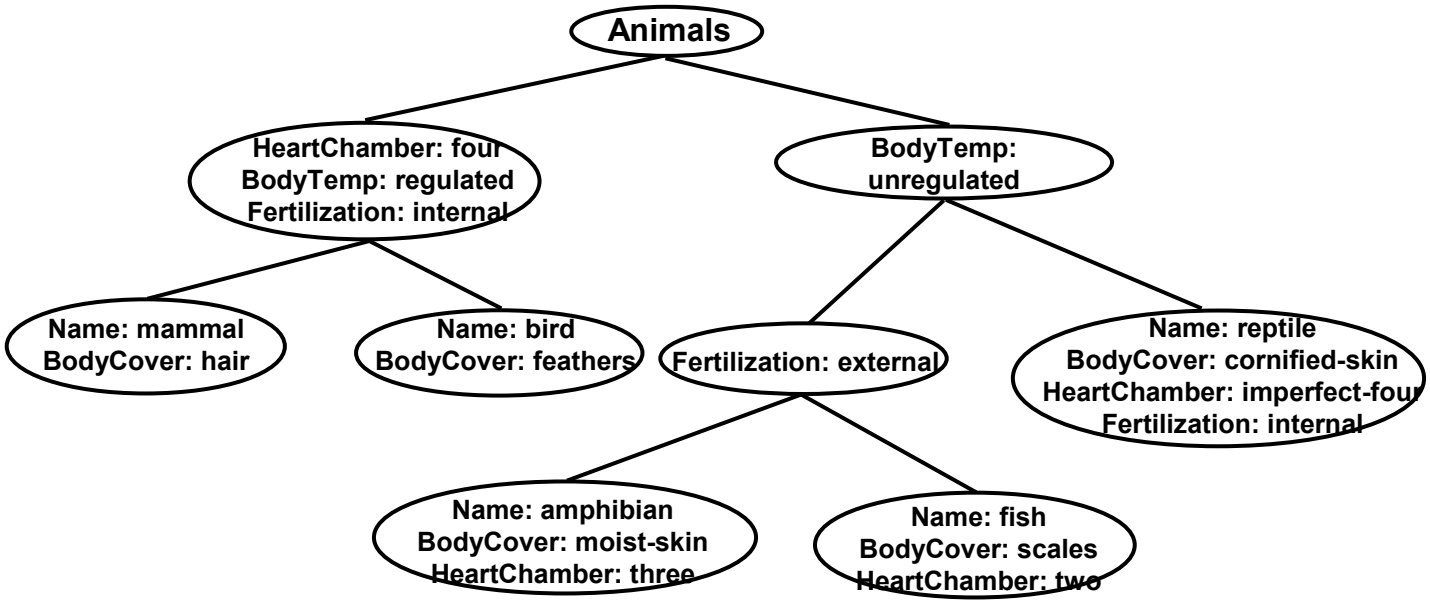


Was ist hierarchisches Clustering ?

TECHNISCHE UNIVERSITÄT DARMSTADT

Gegeben: Menge von ungelabelten Beispielen (=unsupervised)

Ziel:
- Erstellen von Hierarchien, die Daten erklären können
- Vorhersage auf Basis der Hierarchie
- Finden einer Systemmatik

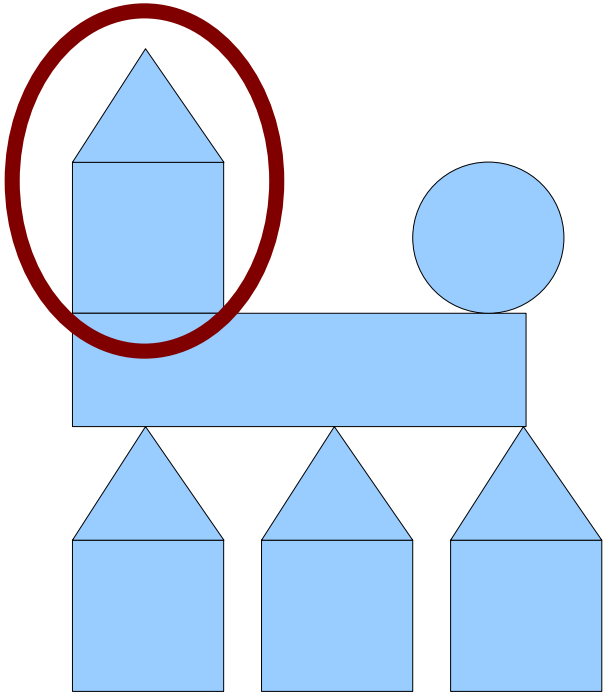


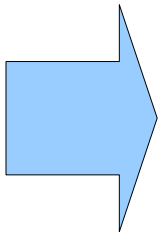
Was ist SUBDUE ?

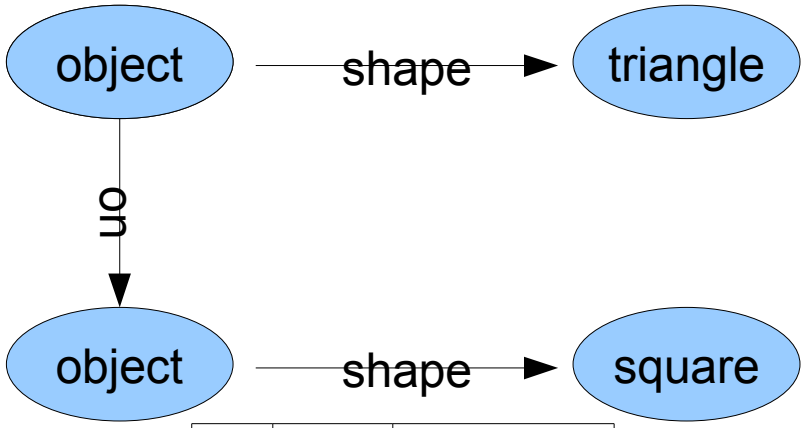
TECHNISCHE UNIVERSITÄT DARMSTADT

- SUBDUE ist ein Verfahren, dass mit Hilfe des Minimum Description Length Principle Teilstrukturen findet
- SUBDUE findet Teilstrukturen, die die Originaldaten komprimieren
- Vereinfachung der Struktur

Eingabe







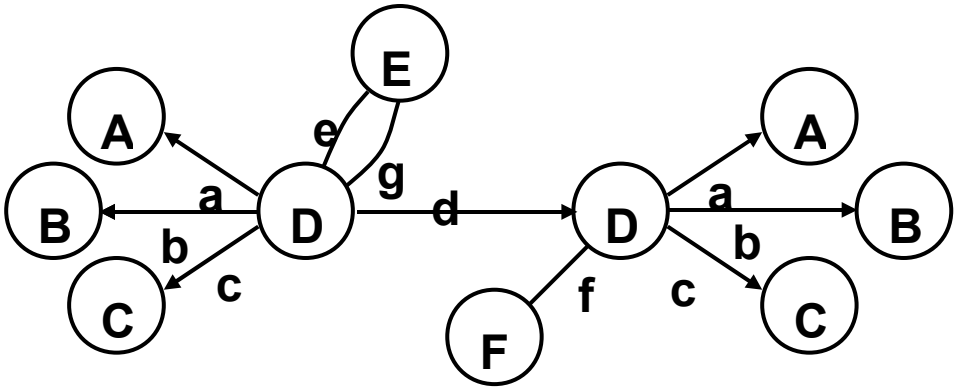
v	1	object
v	2	triangle
v	3	object
v	4	square
d	1	2 shape
d	3	4 shape
d	1	3 on

Suchalgorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

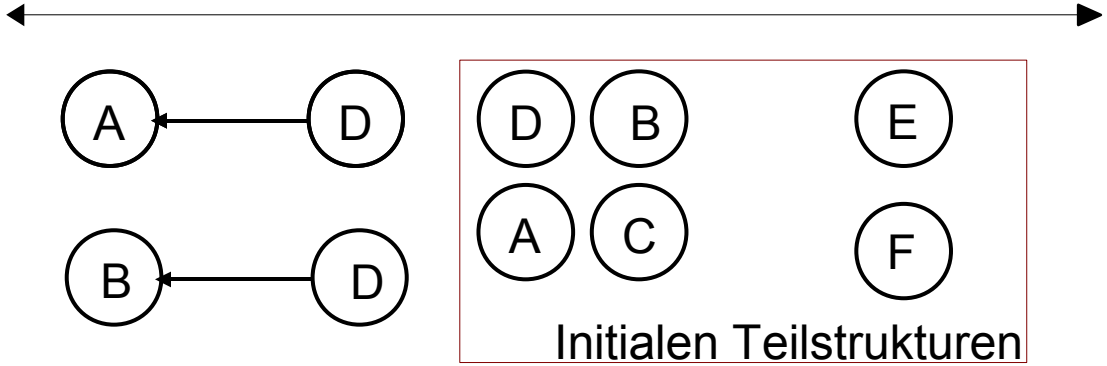
- **Ziel**: Finde die Teilstruktur, die den Originalgraphen am besten komprimiert.
- Eine Teilstruktur besteht aus einer Definition der Teilstruktur und ihrer Häufigkeit.
- Initiale Teilstrukturen sind die Menge an verschiedenen Knoten
- Einziger Suchoperator ist „Extends-Substructure“
- Die maximale Anzahl an Iterationen nicht vorherbestimmbar
  - Graph mit einem Knoten hat keine weiteren Teilstrukturen

Finden einer Teilstruktur



Mögliche Teilstrukturen:

höhere Komprimierung niedrigere Komprimierung



Sortierung der Strukturen nach MDL-Heuristik

Größe der Liste

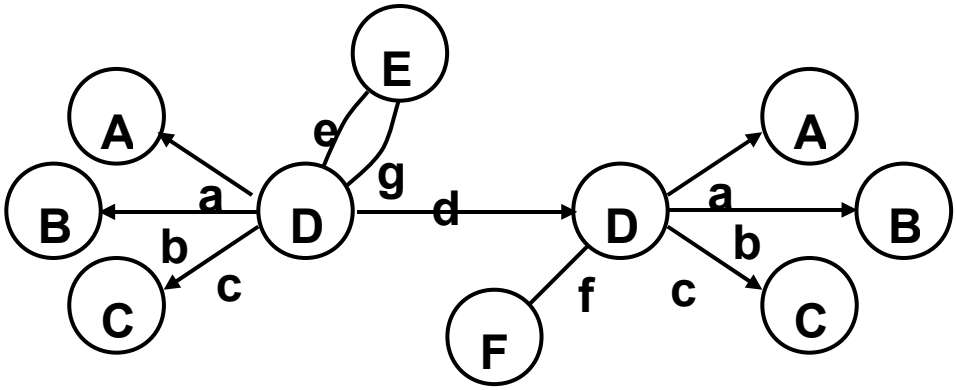
TECHNISCHE UNIVERSITÄT DARMSTADT

- Durch Benutzer angegeben
  - Einträge stellen verschiedene Komprimierungen dar
  - Listengröße verändert sich
- Suchraum vollständig durchsucht

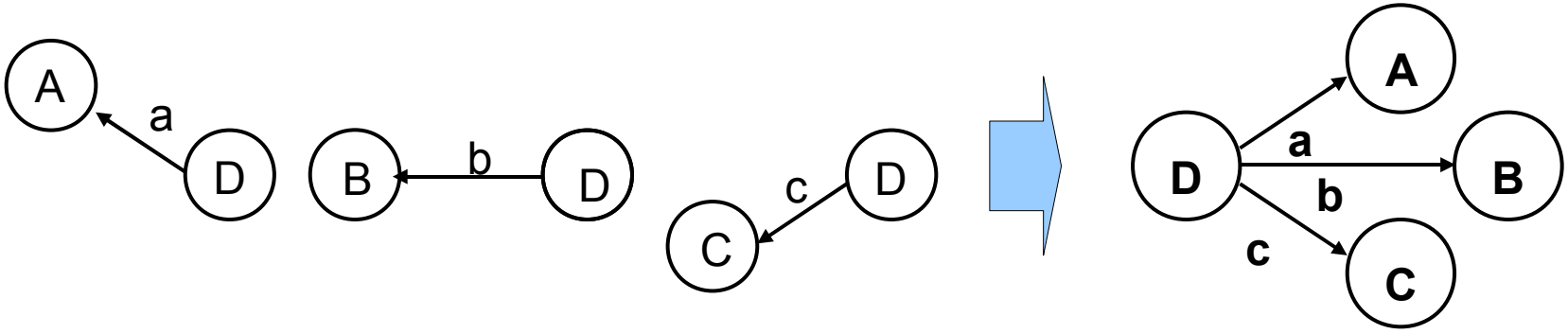
Finden einer Teilstruktur
Verbesserung

TECHNISCHE UNIVERSITÄT DARMSTADT

Voraussetzung:
- Teilstrukturen müssen gleich häufig sein

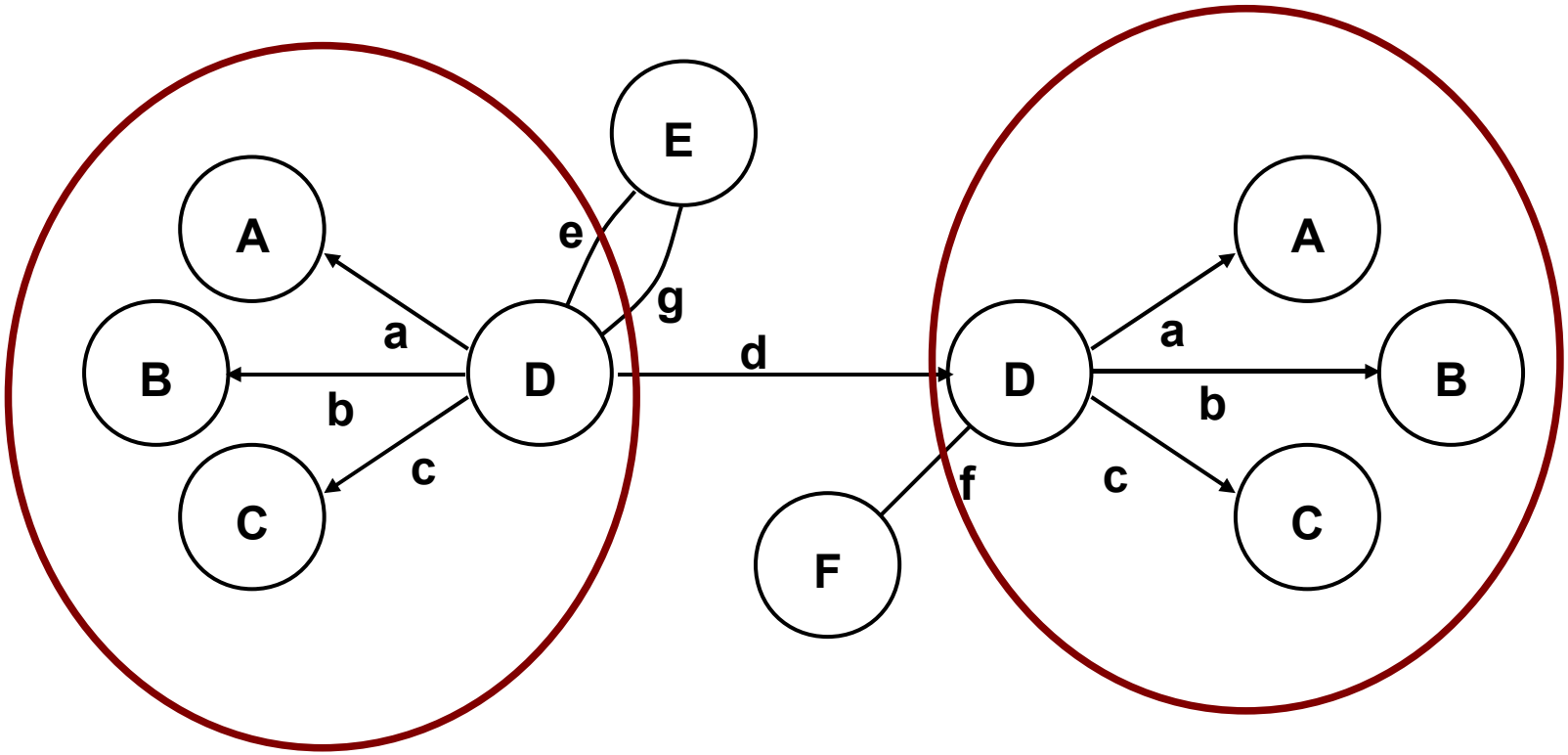


Zusammenfassen von Teilstrukturen:



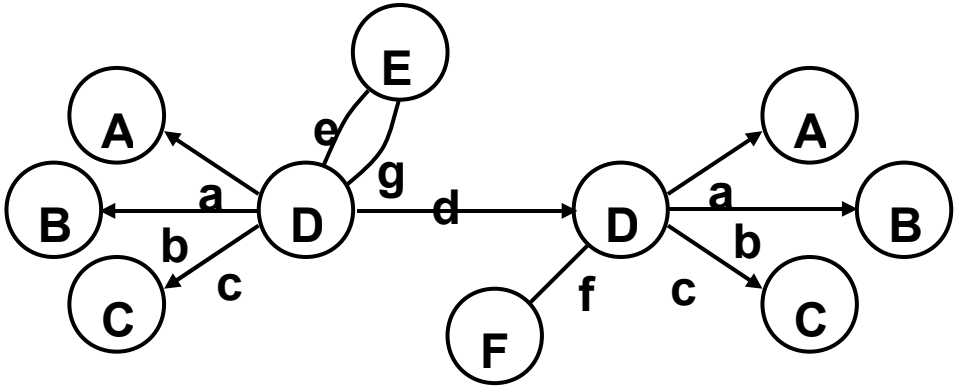
Beispiel: SUBDUE
Finden einer Teilstruktur

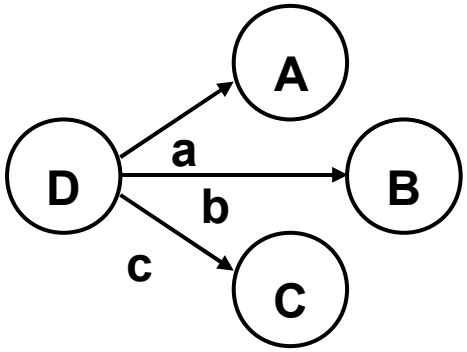
TECHNISCHE UNIVERSITÄT DARMSTADT



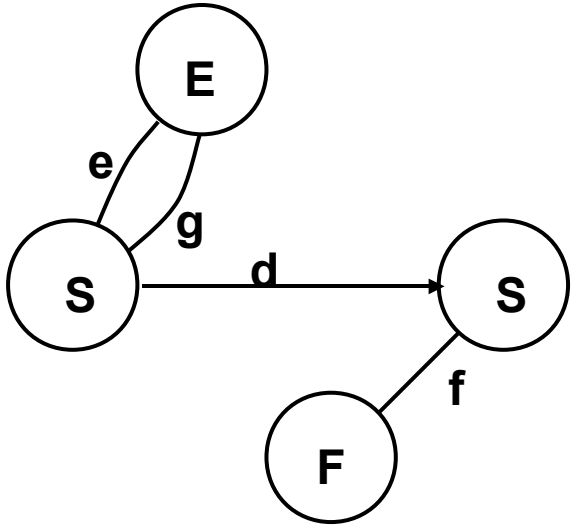
Iteratives Suchen einer besten Teilstruktur mit Hilfe der MDL-Heuristik

SUBDUE: Beispiel Komprimierung





- Komprimierung mit der besten Teilstruktur
- Nicht verlustfrei



Minimum Description Length Heuristik

TECHNISCHE UNIVERSITÄT DARMSTADT

$$Compression = \frac{DL(S) + DL(G|S)}{DL(G)}$$

wobei DL(S) die Größe der Teilstruktur,
DL(G|S) die Größe der Originalstruktur verkleinert um D(S) und DL(G) die Größe der Originalstruktur ist.

Die Größe wird mit der Adjazenzmatrix, die den Graph beschreibt,berechnet.

Inexact Graph Matching

TECHNISCHE UNIVERSITÄT DARMSTADT

- Bestimmte Abweichungen zwischen Strukturen können auftreten
- Absicht über Abweichungen hinweg zu abstrahieren
- Distanz zwischen den Strukturen ist dann die Transformationen um die beiden Strukturen isomorph zu machen
  - Hinzufügen, Löschen von Kanten
  - Hinzufügen, Löschen von Knoten
  - Änderung der Bezeichnung
  - Änderung der Richtung eines Pfeils
- Schwellwert wird vom Nutzer festgelegt

Konzeptuelles Clustering in SUBDUE

TECHNISCHE UNIVERSITÄT DARMSTADT

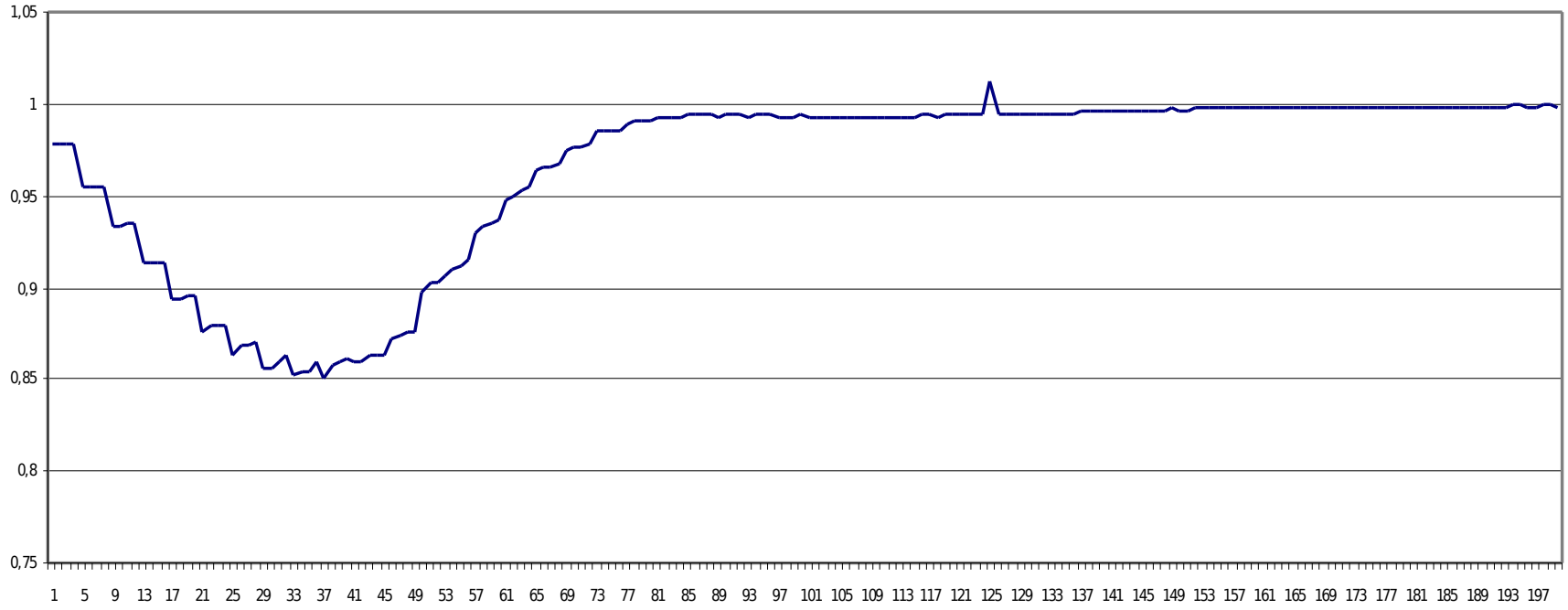
- SUBDUE wird benutzt um Cluster zu identifizieren
  - die beste Teilstruktur einer Iteration definiert ein Cluster

- Wann ist eine Iteration beendet ?
  - SUBDUE kann immer beendet werden
  - Bei Erreichen einer bestimmten Anzahl an Teilstrukturen
  - Analyse des besten Zeitpunktes (prune2)
    - First Minimum Heuristic

First Minimum Heuristic

TECHNISCHE UNIVERSITÄT DARMSTADT

Benutze Teilstruktur beim ersten lokalen Minimum (prune2)



First Minimum Heuristic

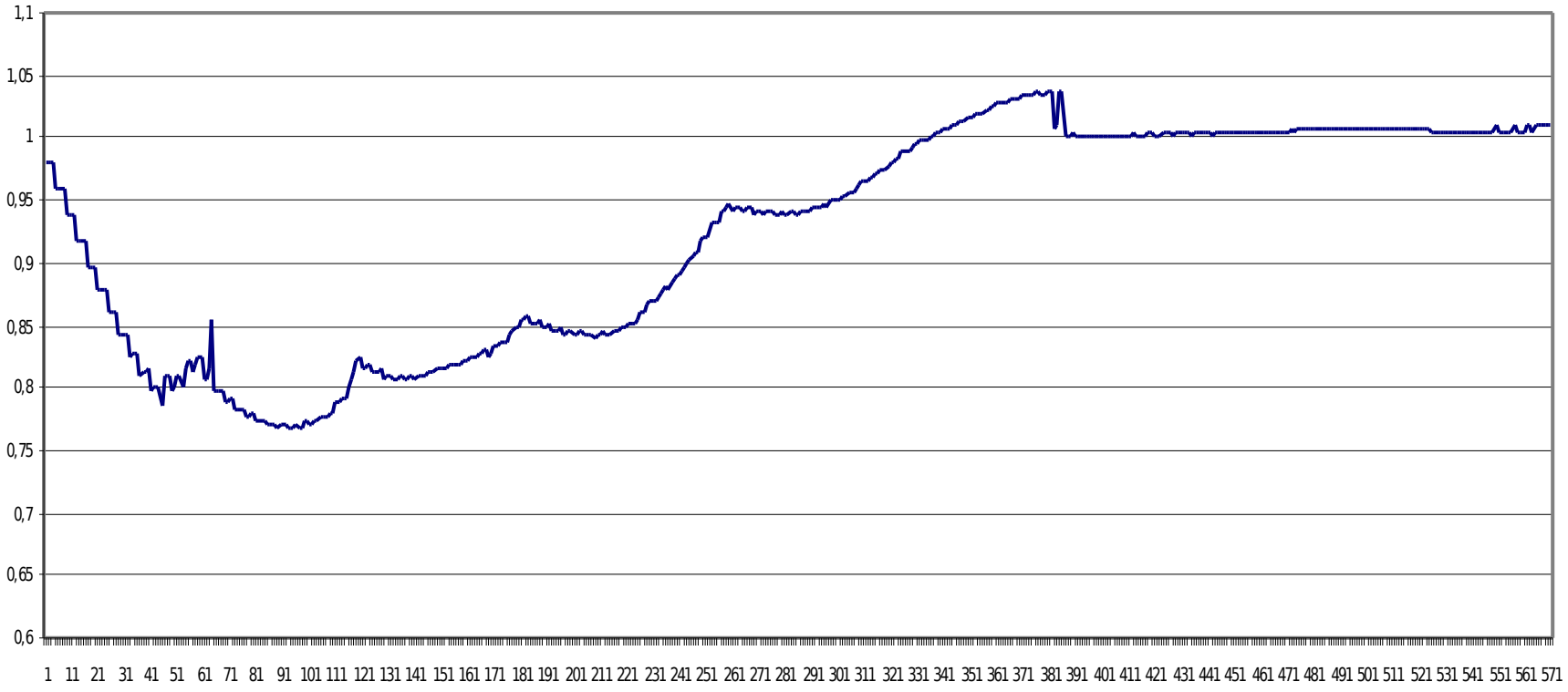
TECHNISCHE UNIVERSITÄT DARMSTADT

- Das erste lokale Minimum ist normalerweise das globale Minimum
- Erstes lokales Minimum
  - kleinere Teilstrukturen
  - häufiger auftretende Teilstrukturen
- Weitere Minima
  - größere Teilstrukturen
  - seltener auftretende Teilstrukturen
- ==> Erste Teilstruktur ist allgemeiner

First Minimum Heuristic

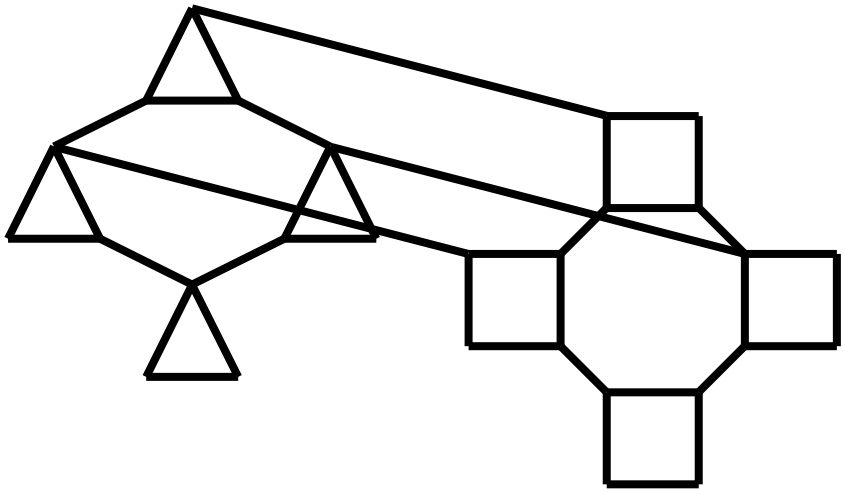
TECHNISCHE UNIVERSITÄT DARMSTADT

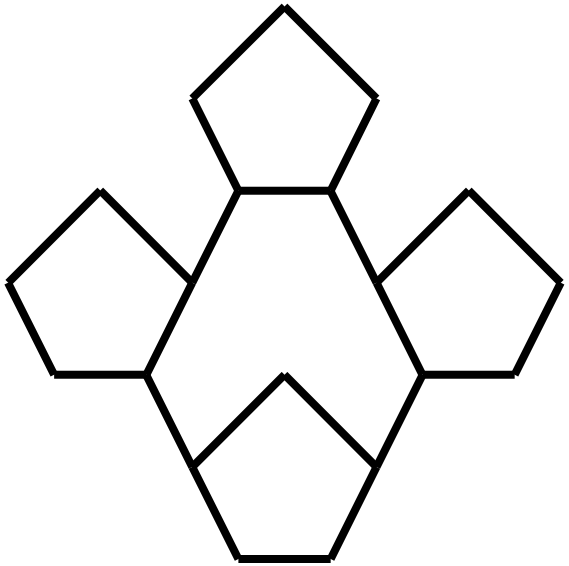
Multi-Minimum Suchraum



Beispiel

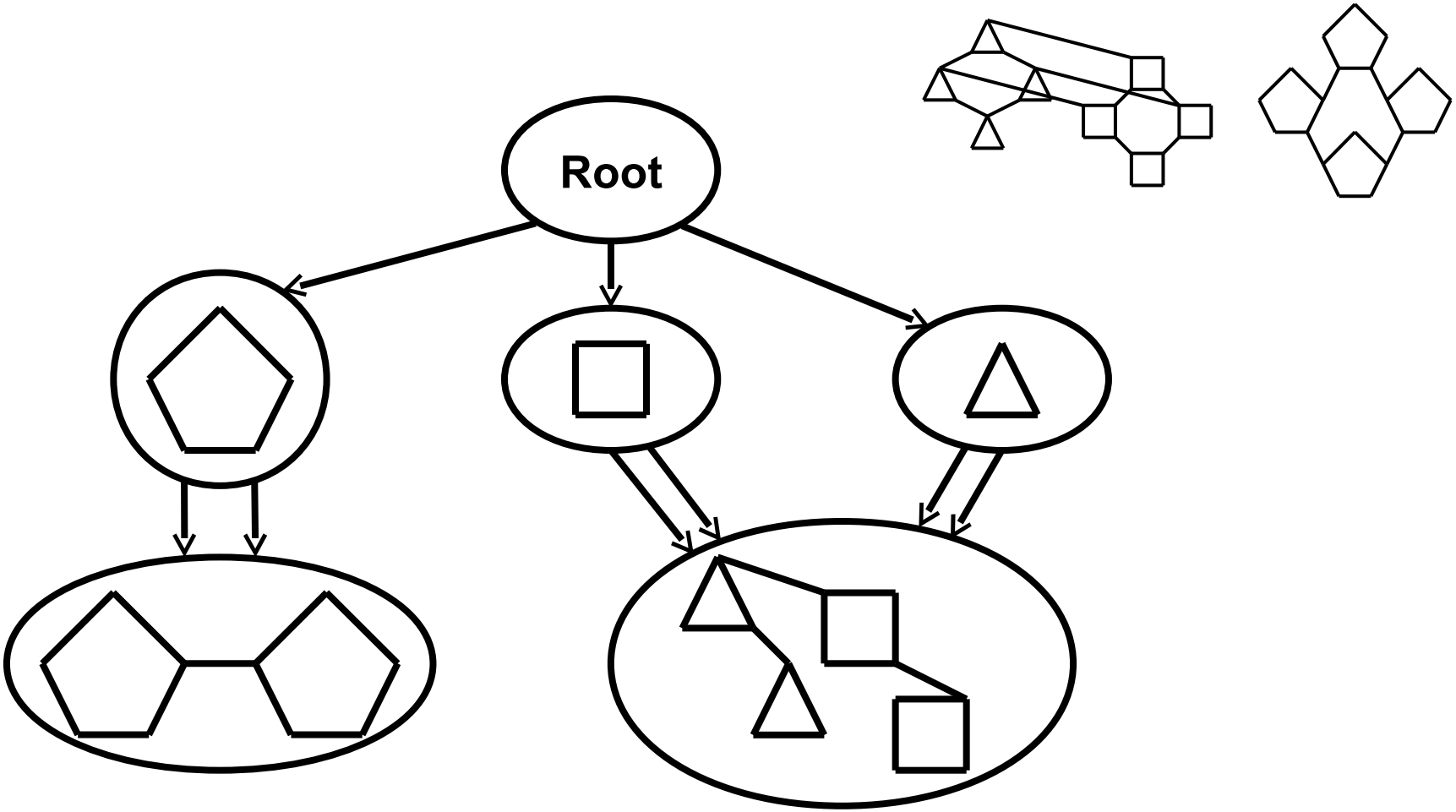
TECHNISCHE UNIVERSITÄT DARMSTADT





Beispiel

TECHNISCHE UNIVERSITÄT DARMSTADT



Evaluierung Clustering

TECHNISCHE UNIVERSITÄT DARMSTADT

$$ClusteringQuality = \frac{InterClusterDistance}{IntraClusterDistance}$$

wobei InterClusterDistance die durchschnittliche Entfernung zwischen Elementen verschiedener Cluster und IntraClusterDistance der Unterschied zwischen zwei Elementen des gleichen Clusters darstellt

- Nicht anwendbar auf hierarchische Probleme
- Keine bekannte hierarchische Evaluierung

Neue Evaluierung hierarchischer Cluster

TECHNISCHE UNIVERSITÄT DARMSTADT

- Merkmale eines guten Clusterings:
  - Kleine Anzahl an Clustern:
  - Große Abdeckung => Gute Generalisierung
  - Große Clusterbeschreibung:
    - Bessere Zuordnung
  - Minimale oder keine Überlappungen zwischen den Clustern:
    - Klarere Cluster => Besser definierte Konzepte

Neue Evaluierung hierarchischer Cluster

TECHNISCHE UNIVERSITÄT DARMSTADT

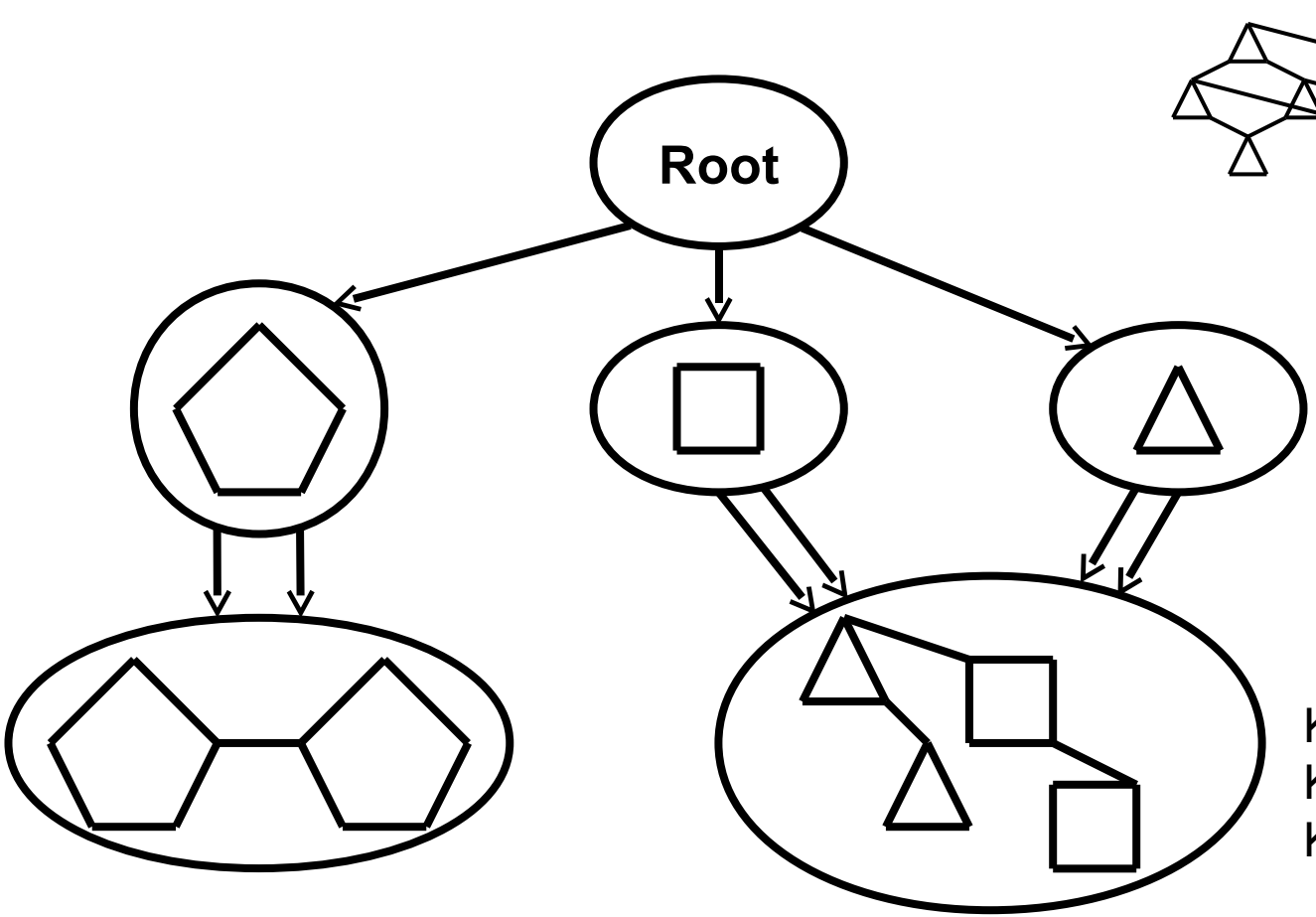
$$Quality(L, G) = \frac{Diversity(root(L))}{Coverage(L, G)}$$

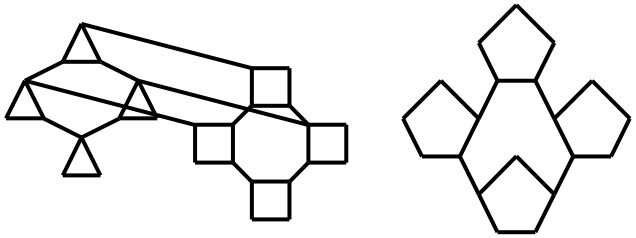
$$Coverage(L, G) = \frac{\| \bigcup_{C \in L} \bigcup_{i=1}^{|C|} C_i \|}{\|G\|}$$

L = Lattice
C = Cluster
G = Originalgraph

Berechnung Coverage

TECHNISCHE UNIVERSITÄT DARMSTADT





Kanten:48+15=63
Knoten:12+16+20=48
Kanten + Knoten = 111

Kanten: 2*(11+17)=56
Knoten: 2*(10+14)=48
Kanten+Knoten = 104

Berechnung Coverage

TECHNISCHE UNIVERSITÄT DARMSTADT

$$Coverage(L,G) = \frac{\left\| \bigcup_{C \in L} \bigcup_{i=1}^{|C|} C_i \right\|}{\|G\|} = \frac{104}{111} = 0.9369$$

Neue Evaluierung hierarchischer Cluster

TECHNISCHE UNIVERSITÄT DARMSTADT

$$Diversity(C) = \frac{\sum_{i=1}^{Degree(C)-1} \sum_{j=i+1}^{Degree(C)} \sum_{k=1}^{|Child_i(C)|} \sum_{l=1}^{|Child_j(C)|} \frac{distance(Child_{i,k}(C), Child_{j,l}(C))}{max(\|Child_{i,k}(C)\|, \|Child_{j,l}(C)\|)}}{\sum_{i=1}^{Degree(C)-1} \sum_{j=i+1}^{Degree(C)-1} max(|Child_i(C)| * |Child_j(C)|)} + \sum_{i=1}^{Numchildren(C)} Diversity(Child_i(C))$$

C ist ein einzelner Cluster

C_i bezieht sich auf die i-te Instanz von C

$|C|$ ist die Anzahl der Instanzen von C und $\|C_j\|$ ist die Größe des Graphen (Anzahl der Kanten plus Anzahl der Knoten)

$Degree(C)$ Anzahl der Kinder von Cluster C

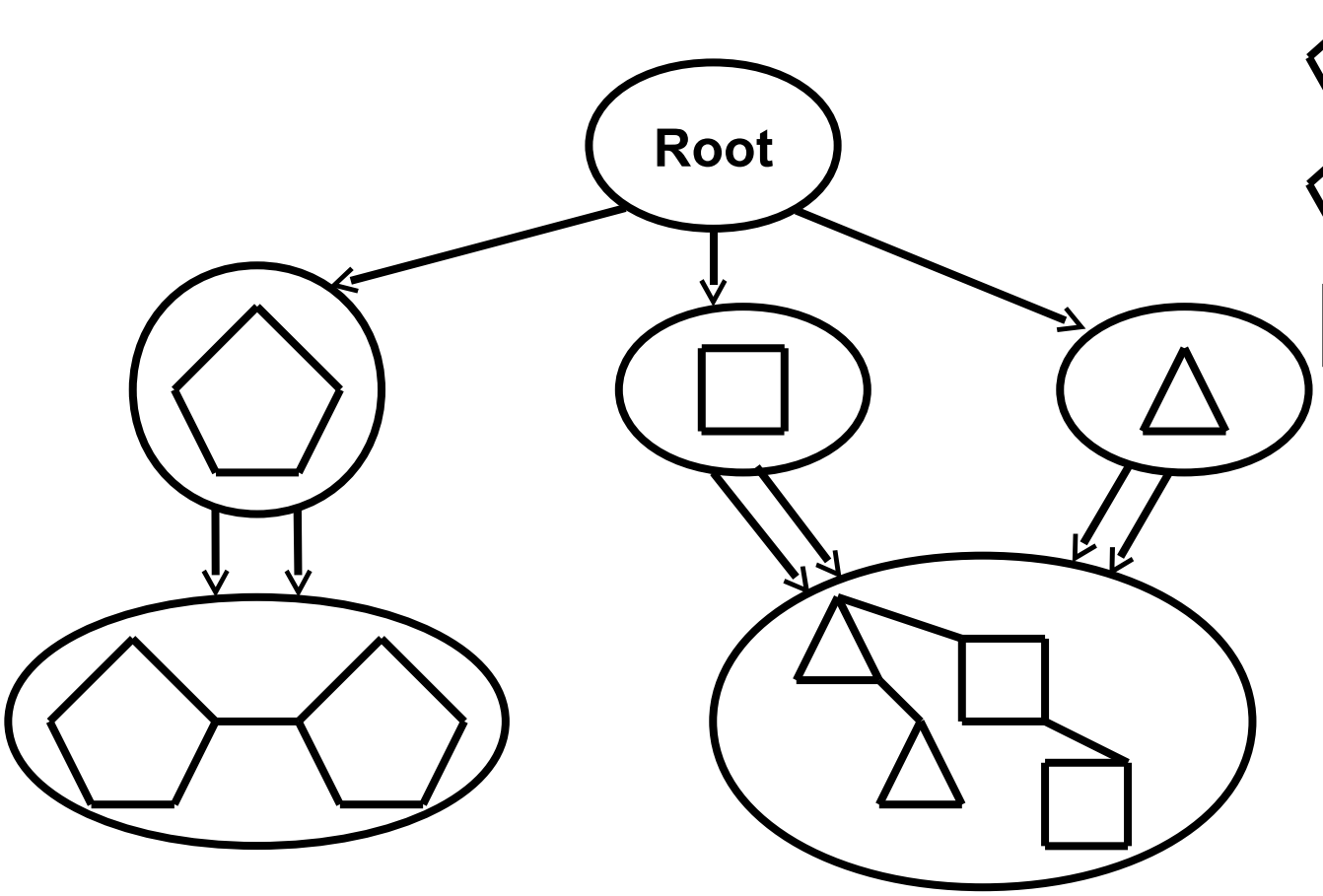
$distance$ ist die Methode wie beim Inexact Matching

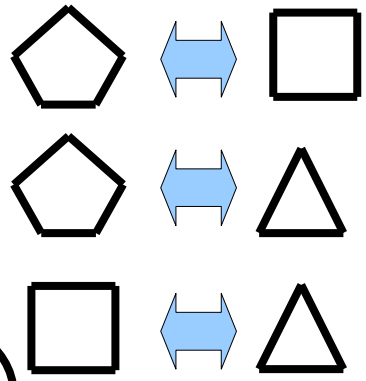
$Child_i(C)$ gibt i-tes Kindes von Cluster C zurück.

$Child_{i,k}(C)$ gibt die k-te Instanz des i-ten Kindes von C wieder

Berechnung Diversity

TECHNISCHE UNIVERSITÄT DARMSTADT





37.6

48 Instanzenpaare

37.6/48 = 0.7833

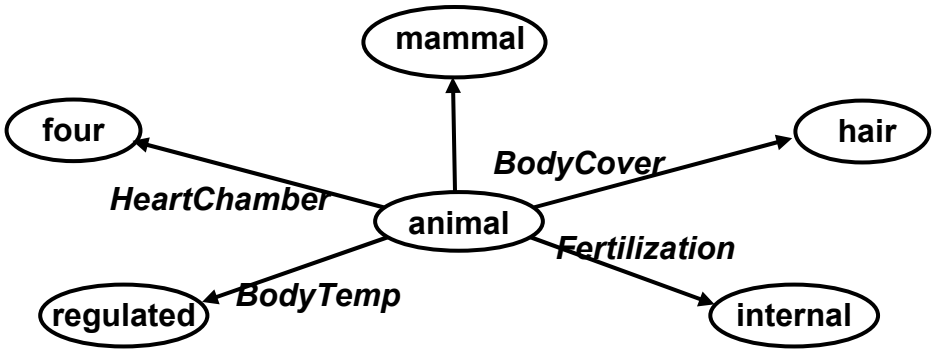
Quality

TECHNISCHE UNIVERSITÄT DARMSTADT

$$Quality(L, G) = \frac{Diversity(root(L))}{Coverage(L, G)} = \frac{0.7833}{0.9369} = 0.8360$$

Evaluierung Tierwelt

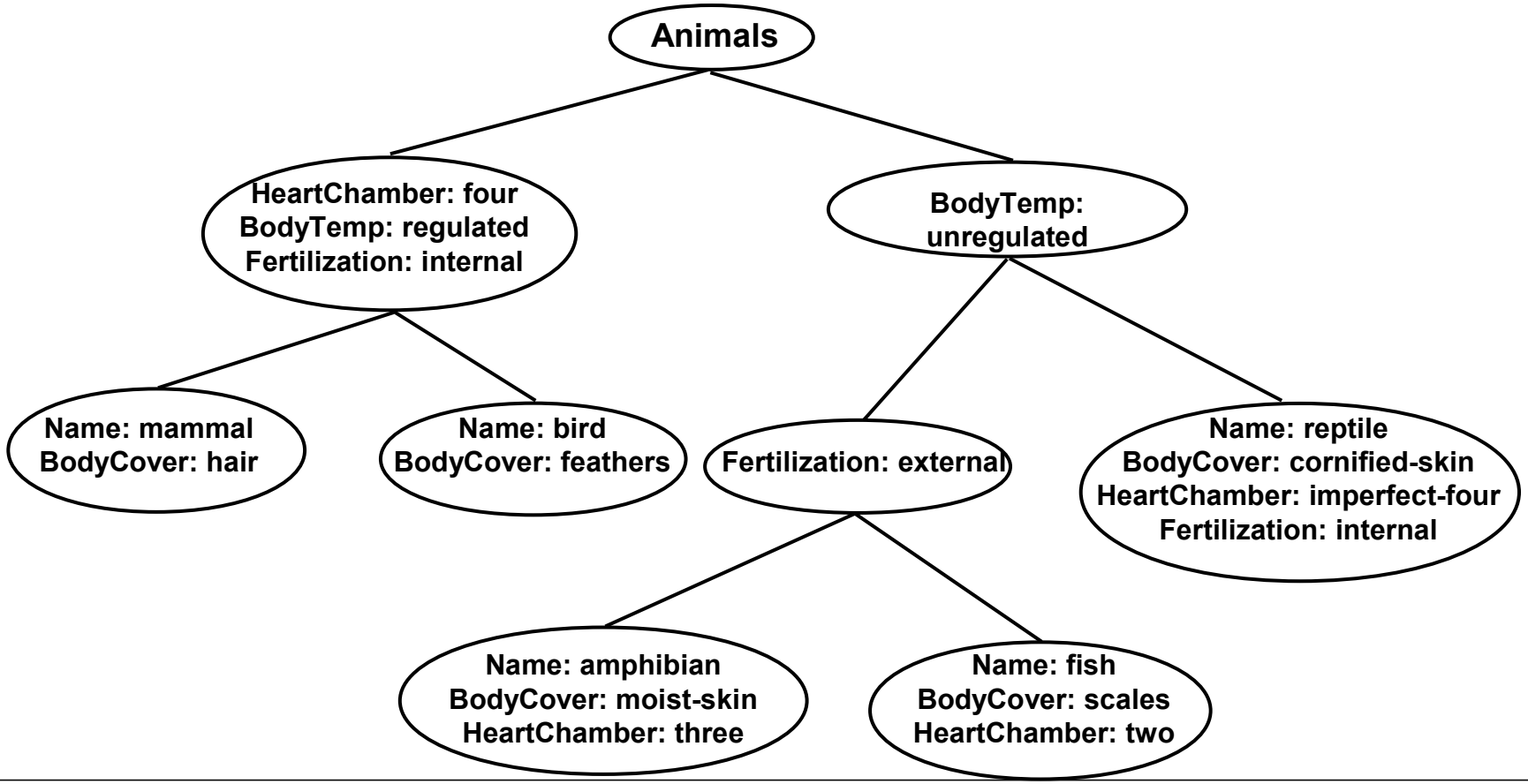
TECHNISCHE UNIVERSITÄT DARMSTADT



Name	*Body Cover*	*Heart Chamber*	*Body Temp.*	*Fertilization*
mammal	hair	four	regulated	internal
bird	feathers	four	regulated	internal
reptile	cornified-skin	imperfect-four	unregulated	internal
amphibian	moist-skin	three	unregulated	external
fish	scales	two	unregulated	external

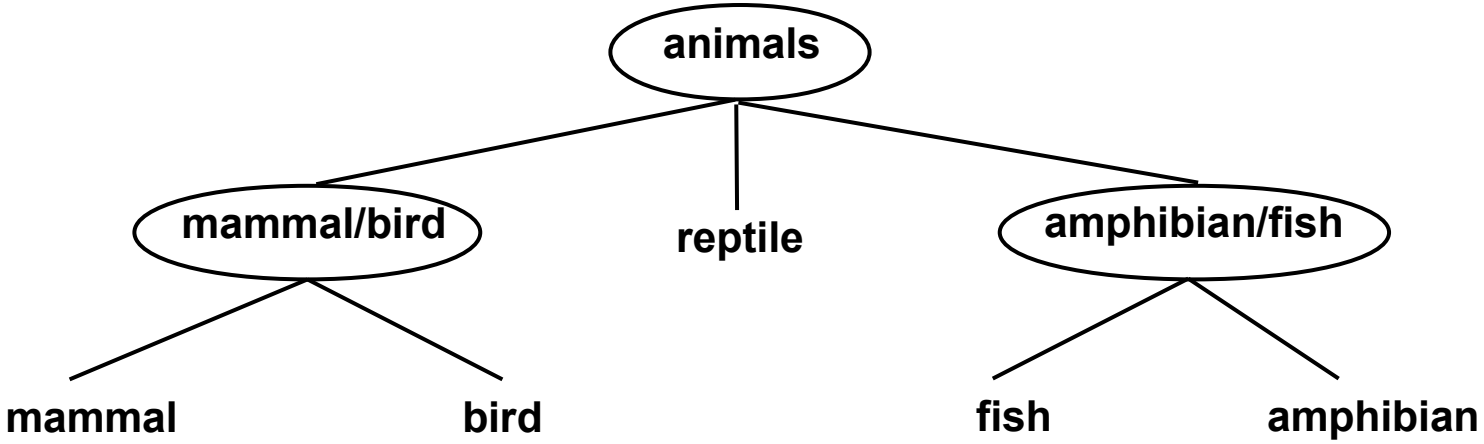
SUBDUE
Hierarchisches Clustering

TECHNISCHE UNIVERSITÄT DARMSTADT



Cobweb
Hierarchisches Clustering

TECHNISCHE UNIVERSITÄT DARMSTADT



Vergleich zwischen SUBDUE und Cobweb

TECHNISCHE UNIVERSITÄT DARMSTADT

- Qualität des SUBDUE lattice: 2.32
- Qualität des Cobweb Baum: 1.48
- SUBDUE ist besser
- Gründe
  - Bessere Generalisierung mit weniger Clustern
  - Keine Überlappung der Cluster zwischen reptile und amphibian/ fish

Evaluierung
Einsatz in der Webwelt

TECHNISCHE UNIVERSITÄT DARMSTADT

- 182 Webseiten mit ähnlichem Aufbau (Universitäten)
  - 32.000 Webseiten besucht
  - Graph hat 41782 Knoten und 168421 Kanten

- SUBDUE hat 136 Teilstrukturen gefunden und benötigt ca. 20 Stunden
  - Speicherprobleme
- Qualität des SUBDUE lattice 10.08

- Cobweb benötigt ca. 40 Stunden
- Qualität des Cobweb Baum 6.23

Fragen ?

TECHNISCHE UNIVERSITÄT DARMSTADT

Vielen Dank für Ihre Aufmerksamkeit !

Anhang

TECHNISCHE UNIVERSITÄT DARMSTADT

- Subdue – Suchalgorithmus
- Größenberechnung
- Inexact Graph Matching
- Cobweb - Algorithmus

SUBDUE - Suchalgorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

```
Subdue ( graph G, int Beam, int Limit)
   queue Q = { v | v has a unique label in G }
   bestSub = first substructure in Q
   repeat
      newQ = {}
      for each S in Q
         newSubs = S extended by an adjacent edge from G in all
                              possible ways
         newQ = newQ U newSubs
         Limit = Limit -1
     evaluate substructures in newQ by compression of G
     Q = first Beam substructures in newQ in decreasing order of
          value
     if best substructure in Q better than bestSub
     then bestSub = first substructure in Q
   until Q is empty or Limit <= 0
   return bestSub
```

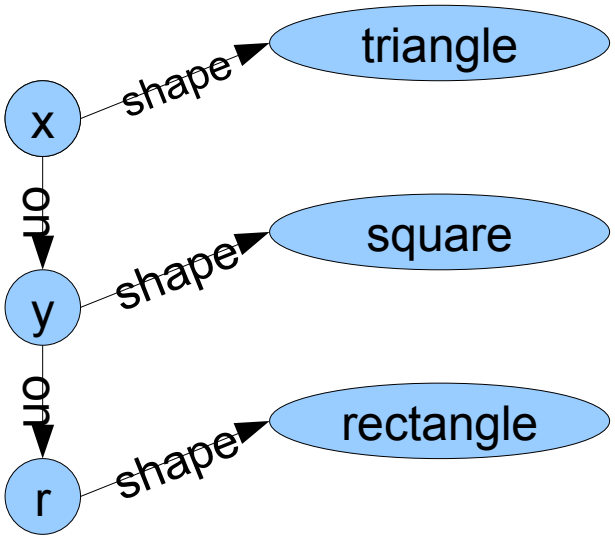
Größenberechnung

TECHNISCHE UNIVERSITÄT DARMSTADT

vbits + rbits + ebits

vbits ist die Anzahl der bits, die benötigt werden um die Knotenlaben zu kodieren
rbits ist die Anzahl der bits, die benötigt werden um die Zeilen der Adjazenzmatrix zu kodieren
ebits ist die Anzahl der bits, die benötigt werden um die Kanten zu kodieren

Beispiel



x	0	1	1	0	0	0
triangle	0	0	0	0	0	0
y	0	0	0	1	1	0
square	0	0	0	0	0	0
r	0	0	0	0	0	1
rectangle	0	0	0	0	0	0

$$vbits = \log_2 v + v \log_2 l_u = \log_2 6 + 6\log_2 8 = 20.58\ bits$$

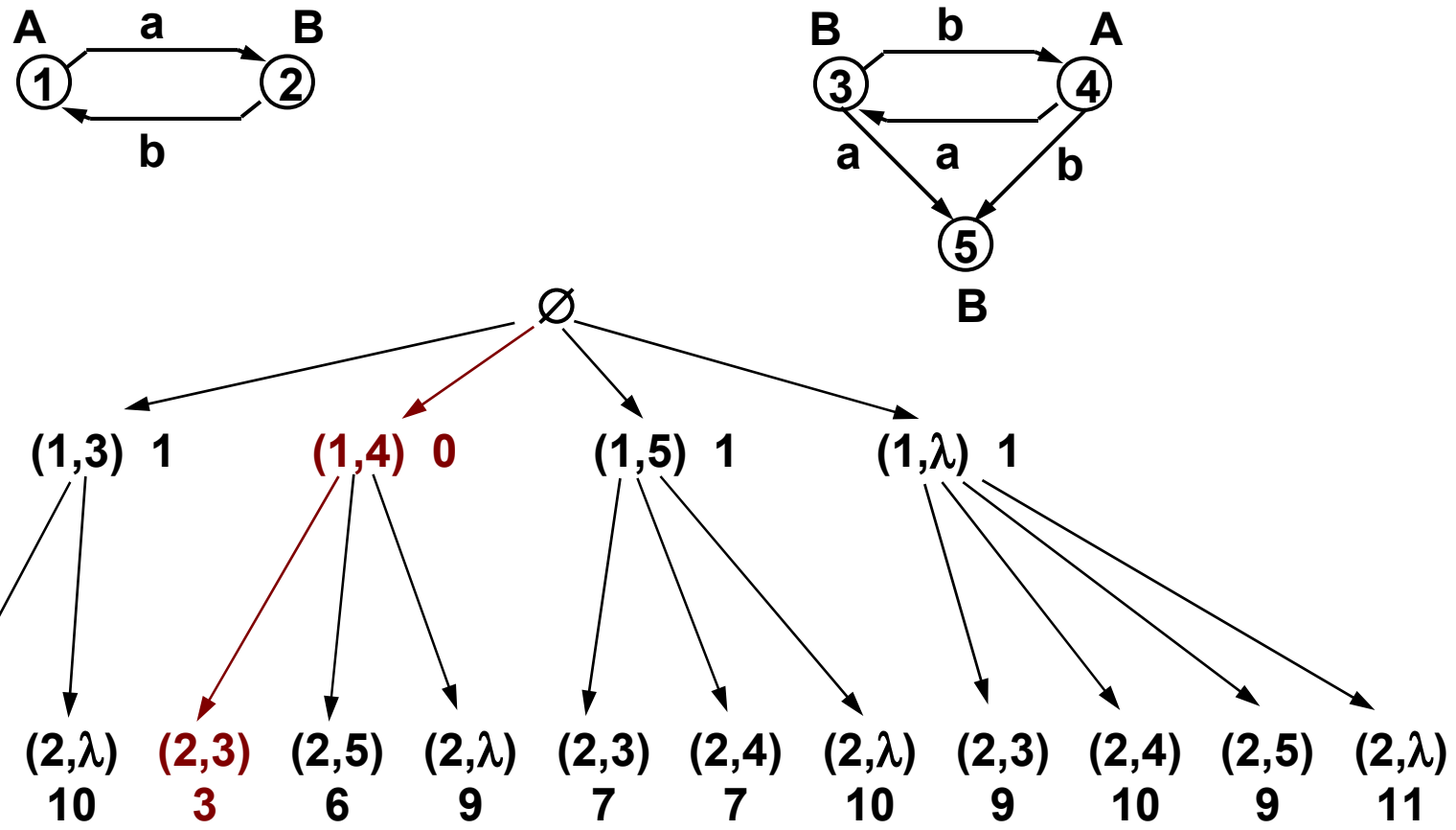
$$rbits = (v+1)\log_2(b+1) \sum \log_2 \binom{v}{k_i} = 7\log_2 3 + 2\log_2 \binom{6}{2} + 3\log_2 \binom{6}{0} + \log_2 \binom{6}{1} = 21.49\ bits$$

$$ebits = e(1+\log_2 l_u) + (K+1)\log_2 m = 5(1+\log_2 8) + 6\log_2 1 = 20$$

$$vbits + rbits + ebits = 62.07\ bits$$

Inexact Graph Matching Beispiel

TECHNISCHE UNIVERSITÄT DARMSTADT



Least-cost match is {(1,4), (2,3)}

Cobweb - Suchalgorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

```
Algorithm COBWEB
  COBWEB(root, record):
  Input: A COBWEB node root, an instance to insert record
  if root has no children then
    children := {copy(root)}
    newcategory(record)
    insert(record, root) \\ update root's statistics
  else
    insert(record, root)
    for child in root's children do
      calculate Category Utility for insert(record, child),
      set best1, best2 children w. best CU.
    end for
    if newcategory(record) yields best CU then
      newcategory(record)
    else if merge(best1, best2) yields best CU then
      merge(best1, best2)
      COBWEB(root, record)
    else if split(best1) yields best CU then
      split(best1)
      COBWEB(root, record)
    else
      COBWEB(best1, record)
    end if
  end
```

Literaturangaben

- I. Jonyer, D. J. Cook, and L. B. Holder – Graph- Based Hierarchical Conceptual Clustering, 2001 aus Journal of Machine Learning Research
- D. J. Cook, L. B. Holder – Substructure Discovery Using Minimum Description Length and Background Knowledge, 1993 aus Journal Artificial Intelligence Research
- SUBDUE – Graph Based Knowledge - http://ailab.wsu.edu/subdue/ - Online zugegriffen am 08.01.2009
- Cobweb (clustering) - http://en.wikipedia.org/wiki/Cobweb_(clustering) - Online zugegriffen am 11.01.2009
- I. Jonyer, L. B. Holder, and D. J. Cook - Hierarchical Conceptual Structural Clustering, 2001 aus International Journal on Artificial Intelligence Tools
- I. Jonyer, L. B. Holder, and D. J. Cook, Graph-Based Hierarchical Conceptual Clustering in Structural Databases',2001
- I. Jonyer, L. B. Holder, and D. J. Cook, Graph-Based Hierarchical Conceptual Clustering, 2001