Ähnlichkeit von Strukturierten Daten

Martin Hofmann

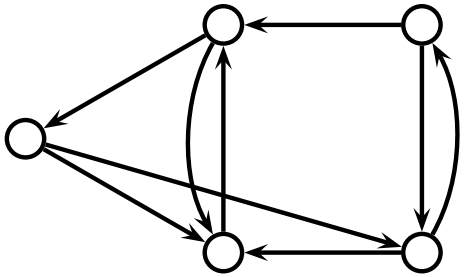
Gruppe Kognitive Systeme
Fakultät für Wirtschafts- und Angewandte Informatik
Universität Bamberg

Reading Club Similarity, SS 2008

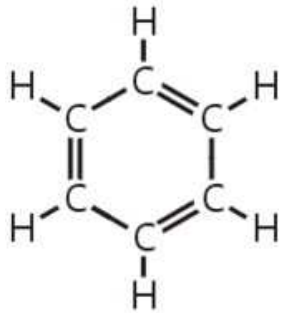
Gliederung

Strukturen

- Graphen



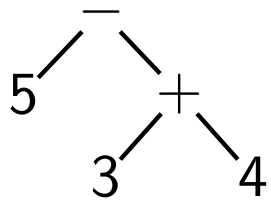
(unbenannt)



(benannt)

- Bäume/Terme

$$5 - (3 + 4)$$

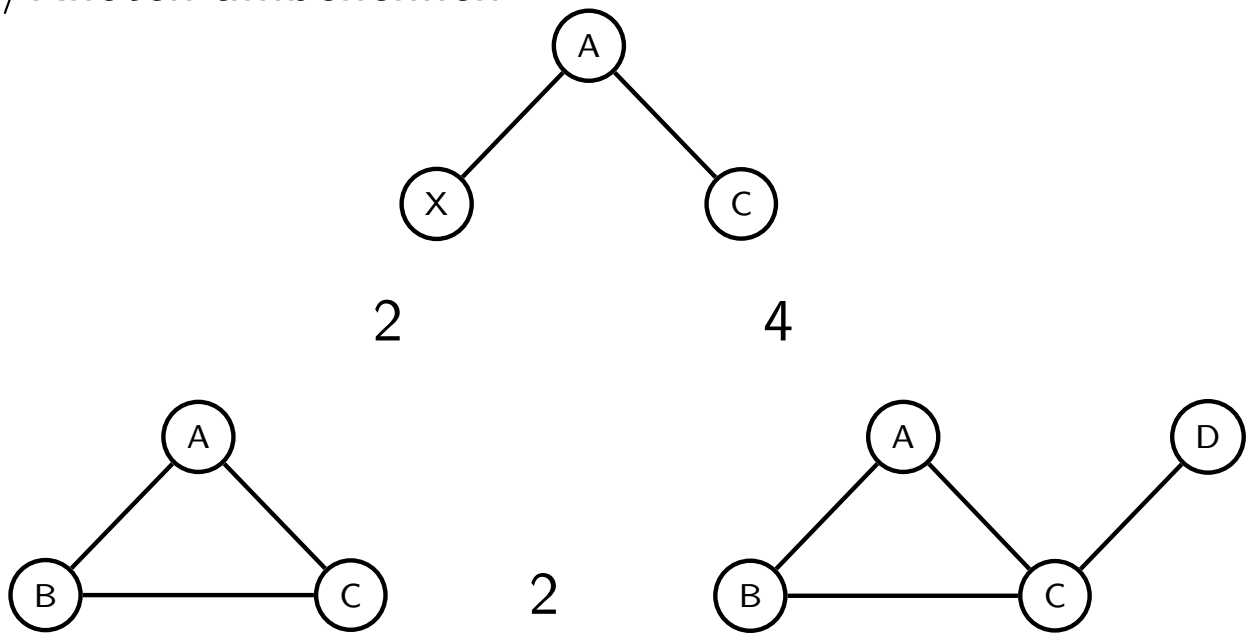


- Strings

ACTCATGTGGTGGATTC

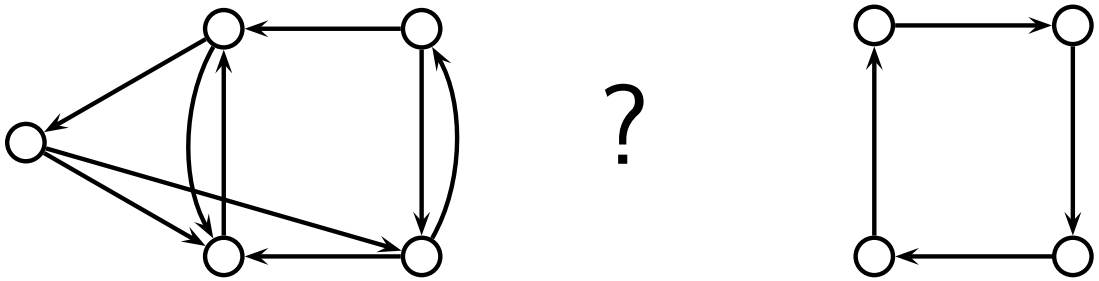
(Un)ähnlichkeit allgemein – Edit-Distanz

- Für Graphen G_1 und G_2 ist Editdistanz $\delta(G_1, G_2)$ die minimale Anzahl der Editoperationen um G_1 in G_2 zu überführen.
- Operatoren
  - Kante/Knoten einfügen
  - Kante/Knoten löschen
  - Kante/Knoten umbenennen

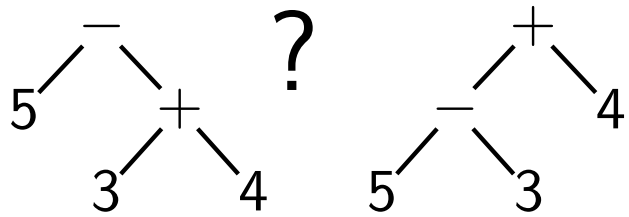


Ähnlichkeit intuitiv — Wo ist das Problem?

- Subgraphisomorphie



- Korrektur Problem



- Längster Gemeinsamer Substring

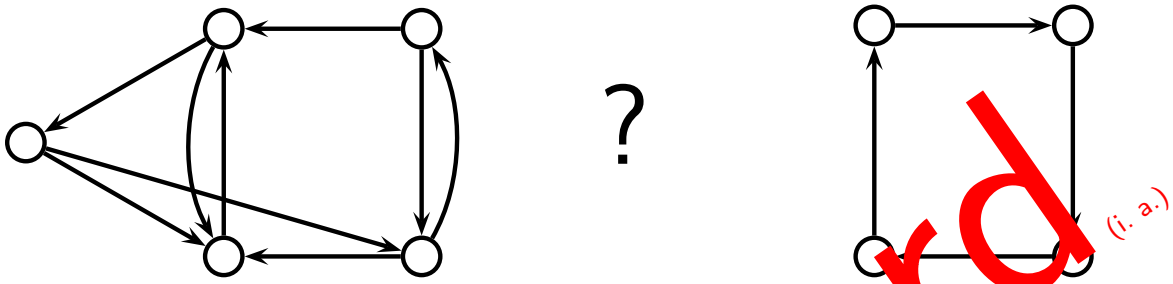
ACTCATGTGGTGGATTC

?

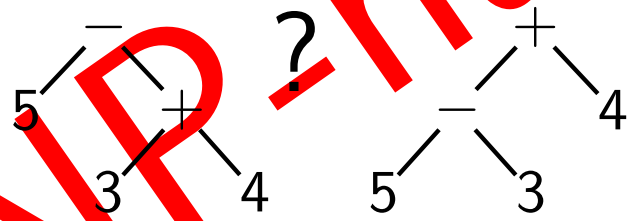
GTGGTGGATTCACTCAT

Ähnlichkeit intuitiv — Wo ist das Problem?

- Subgraphisomorphie



- Korrektur Problem



NP-hard (i. a.)

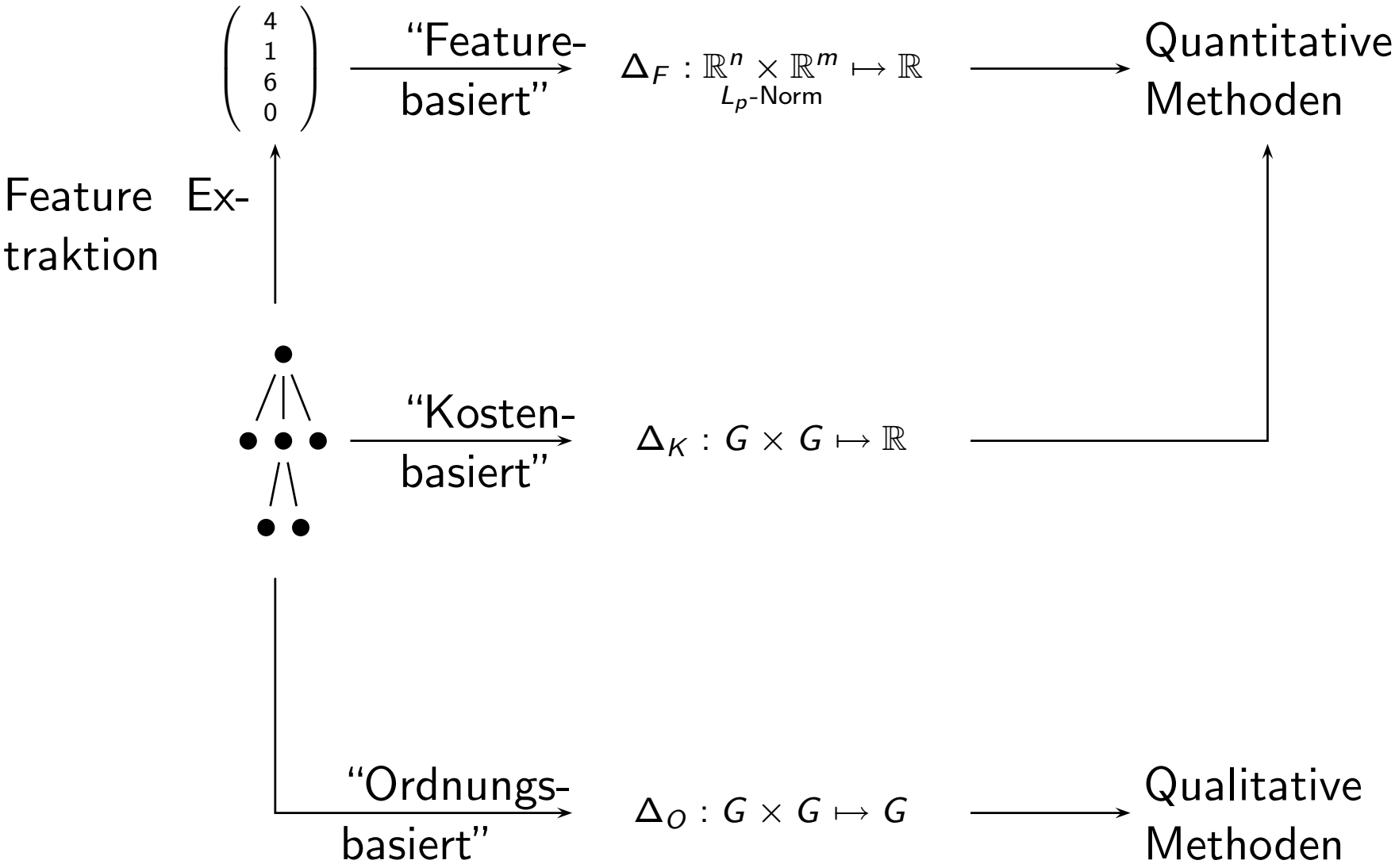
- Längster Gemeinsamer Substring

ACTCATGTGGTGGATTC

?

GTGGTGGATTCACTCAT

Wie messen? – Alternativen



Gliederung

"Feature-basiert" Ansätze

- Vermeidung Edit-Distanz direkt zu berechnen
- Identifikation von Merkmalen $\longrightarrow$ Codierung als Vektor
- L_p-Norm als Distanzmaß $\longrightarrow$ Clusteranaylse, MVM

Probleme

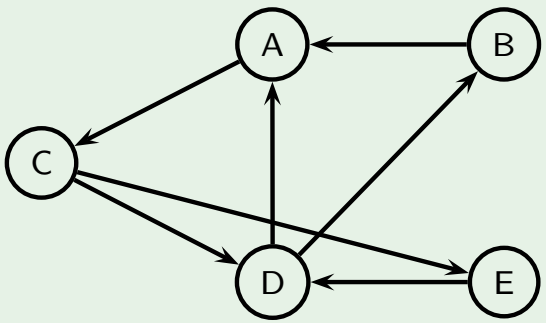
- welche Merkmale
- Berücksichtigung von Knoten-/Kantennamen aufwändig
- Informationsverlust
- *curse of dimensionality* (besonders bei benannten Graphen)

Graph-Histogramm

[Papadopoulos and Manolopoulos, 1999]

- Graph $G(V, E)$, mit $V_G = \{v_1, \ldots, v_n\}$
- *degree* $deg(v)$ ist die Anzahl der verbundenen Kanten
- Gaph-Histogramm $histo(G) = (deg(v_1) + 1, \ldots, deg(v_i) + 1)$
- $histo_s(G) = (x_1, \ldots, x_s, x_t, \ldots, x_n)$ ist sortiert, falls $\forall x_s, x_t . x_s \leq x_t$ mit $1 \leq s \leq t \leq n$

Example



$$\xrightarrow{histo(G)} \begin{matrix} A \\ B \\ C \\ D \\ E \end{matrix} \begin{pmatrix} 4 \\ 3 \\ 4 \\ 5 \\ 3 \end{pmatrix} \xrightarrow{histo_s(G)} \begin{matrix} D \\ A \\ C \\ B \\ E \end{matrix} \begin{pmatrix} 5 \\ 4 \\ 4 \\ 3 \\ 3 \end{pmatrix}$$

Graph-Histogramm

[Papadopoulos and Manolopoulos, 1999]

Probleme (besonders im Datenbankkontext)

- $|V_1| \gg |V_2|$
- $histo(G) \in \mathbb{R}^n$ für großes n

Normierung

$$\left\langle \begin{pmatrix} 4 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 2 \\ 1 \\ 1 \end{pmatrix} \right\rangle \rightarrow \left\langle \begin{pmatrix} 4 \\ 3 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 4 \\ 2 \\ 1 \\ 1 \end{pmatrix} \right\rangle$$

Dimensionsreduktion

- für $SH = (x_1, \ldots, x_n)$ mit $f > 0$ und $n \mod f = 0$ sei

$$SH^* = \left(\sum^{f} x_j, \sum^{2f} x_j, \ldots, \sum^{n} x_j \right)$$

Graph-Histogramm

[Papadopoulos and Manolopoulos, 1999]

Normierung

$$\left\langle \begin{pmatrix} 4 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 2 \\ 1 \\ 1 \end{pmatrix} \right\rangle \rightarrow \left\langle \begin{pmatrix} 4 \\ 3 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 4 \\ 2 \\ 1 \\ 1 \end{pmatrix} \right\rangle$$

Dimensionsreduktion

- für $SH = (x_1, \ldots, x_n)$ mit $f > 0$ und $n \mod f = 0$ sei

$$SH^* = \left(\sum_{j=1}^{f} x_j, \sum_{j=f+1}^{2f} x_j, \ldots, \sum_{j=n-f}^{n} x_j \right)$$

Graph-Histogramm

[Papadopoulos and Manolopoulos, 1999]

Normierung

$$\left\langle \begin{pmatrix} 4 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 2 \\ 1 \\ 1 \end{pmatrix} \right\rangle \rightarrow \left\langle \begin{pmatrix} 4 \\ 3 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 4 \\ 2 \\ 1 \\ 1 \end{pmatrix} \right\rangle$$

Dimensionsreduktion

- für $SH = (x_1, \ldots, x_n)$ mit $f > 0$ und $n \mod f = 0$ sei

$$SH^* = \left(\sum_{j=1}^{f} x_j, \sum_{j=f+1}^{2f} x_j, \ldots, \sum_{j=n-f}^{n} x_j \right)$$

Graph-Histogramm

[Papadopoulos and Manolopoulos, 1999]

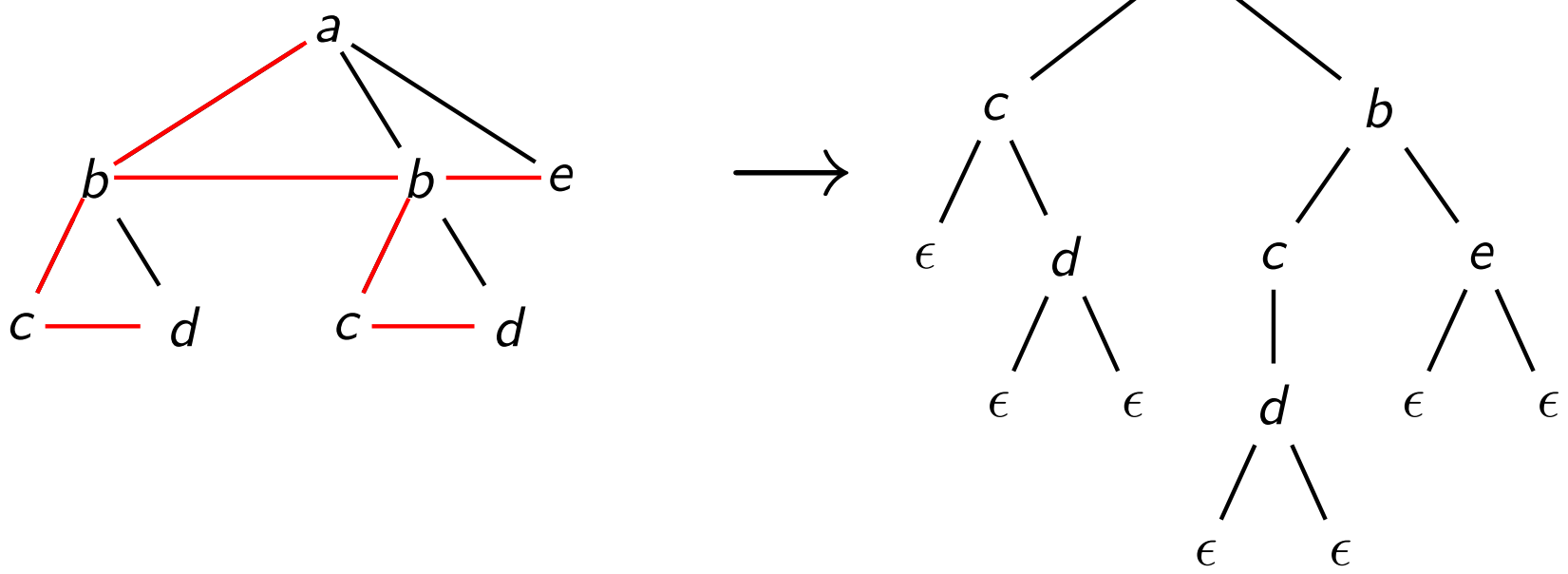
Eigenschaften

$$
\begin{aligned}
L_1(histo(G_1), histo(G_2)) &\geq \delta(G_1, G_2) \\
L_1(histo_s(G_1), histo_s(G_2)) &= \delta(G_1, G_2) \\
L_1(histo_s(G_1), histo_s(G_2)) &\geq L_1(SH_1^*, SH_2^*)
\end{aligned}
$$

Binary Branch Vector

[Yang et al., 2005]

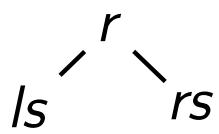
1. Bäume normalisieren (*binary branch tree*)



Binary Branch Vector

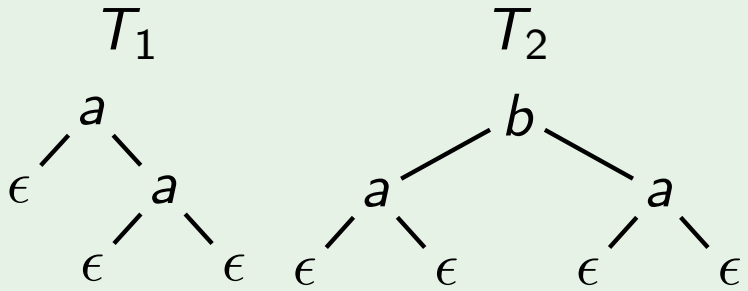
[Yang et al., 2005]

2. Für zwei normalisierte Bäume alle binäre Verzweigungen $b = \langle ls, r, rs \rangle$ bestimmen



3. Binary Branch Vector $BV = (x_1, \ldots, x_n)$ des Baumes T_1 sind die Häufigkeiten x_i der binären Verzweigung b_i in T_1

Example



	BV_1	BV_2
$\langle \epsilon, a, \epsilon \rangle$	1	2
$\langle \epsilon, a, a \rangle$	1	0
$\langle a, b, a \rangle$	0	1

Gliederung

"Edit Distanz"-basierte Ansätze

- Berechnung der Edit-Distanz
- kein Informationverlust
- Berücksichtigung von Labels möglich
- Verwendung der Edit-Distanz für Clusteranalyse etc.

Probleme

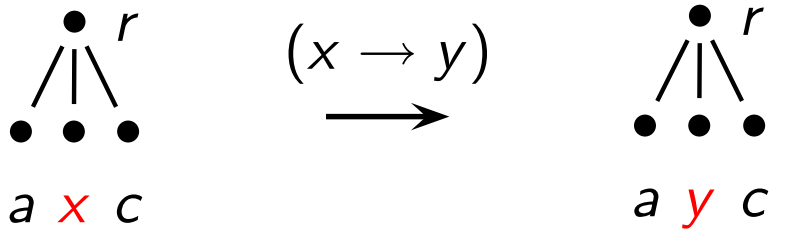
- Edit-Distanz i. a. NP-hard
- Notwendigkeit von Einschränkungen
- u. U. Vereinfachung nötig

Notation

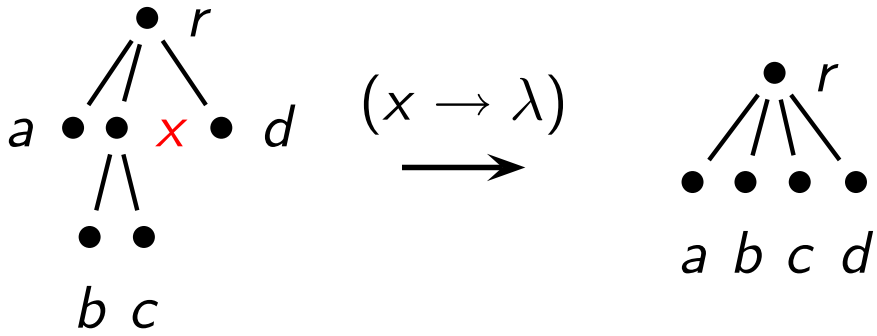
- Baum T mit Wurzel $root(T)$, leerer Baum θ
- $T(v)$ ist Subtree von T mit Wurzel $v \in V(G)$
- Nachfolger, Vorfahr, Elter, Kind, Geschwister (wie üblich)
- T ist *geordnet* falls für alle Kinder eine l-r Ordnung existiert
- T ist *benannt* mit $\beta : V(T) \mapsto \Sigma$ für endliches Alphabet Σ
- Wald F ist eine Menge von Bäumen $T_1, \ldots, T_n$
- F ist geordnet falls eine l-r Ordnung zwischen geordneten $T_i s$ besteht
- für $v \in V(T)$ mit Kindern $v_1, \ldots, v_i$ ist $F(v) = T(v_s), \ldots, T(v_t)$ mit $1 \leq s \leq t \leq i$

Edit Operationen

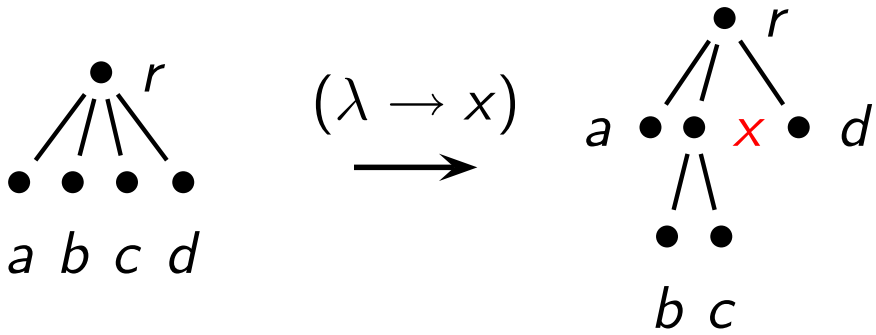
- Relabel



- Delete



- Insert



Edit Operationen

Kostenfunktion

- Knotenlabel l aus endlichem Alphabet Σ
- Sonderzeichen $\lambda \notin \Sigma$ und $\Sigma_\lambda = \Sigma \cup \lambda$
- Kostenfunktion $\gamma : (\Sigma_\lambda \times \Sigma_\lambda) \backslash (\lambda, \lambda) \mapsto \mathbb{R}$
- γ sein eine Metrik die die Eigenschaften der Nichtnegativität, Symmetrie und der Dreiecksungleichung erfüllt

Edit Operation

- $(l_1 \rightarrow l_2)$, wobei $(l_1, l_2) \in (\Sigma_\lambda \times \Sigma_\lambda) \backslash (\lambda, \lambda)$

  Relabel $l_1 \neq \lambda \wedge l_2 \neq \lambda$

  Insert $l_1 = \lambda$

  Delete $l_2 = \lambda$
- $\gamma((l_1 \rightarrow l_2)) = \gamma(l_1, l_2)$

Tree Edit Distance

[Lu, 1979, Tai, 1979]

Edit Script

Edit Script $S = s_1, \ldots, s_n$ eine Folge von Edit Operationen

$$\begin{aligned} \gamma(S) &= \sum_{i=1}^{n} \gamma(s_i) \\ \delta(T_1, T_2) &= \min\{\gamma(S) | S \text{ transformiert } T_1 \text{ nach } T_2\} \end{aligned}$$

Edit Mapping

Tree Edit Distance

[Lu, 1979, Tai, 1979]

Edit Script

Edit Mapping

Mapping (M, T_1, T_2) mit $M \subseteq V(T_1) \times V(T_2)$ und $(v_1, w_1), (v_2, w_2) \in M$:

1. $v_1 = v_2$ gdw. $w_1 = w_2$
2. v_1 ist Vorfahr von v_2 gdw. w_1 Vorfahr von w_2
3. v_1 ist links von v_2 gdw. w_1 links von w_2

$$\gamma(M) = \sum_{(v,w)\in M} \gamma(v \to w) + \sum_{v \in N_1} \gamma(v \to \lambda) + \sum_{w \in N_2} \gamma(\lambda \to w)$$

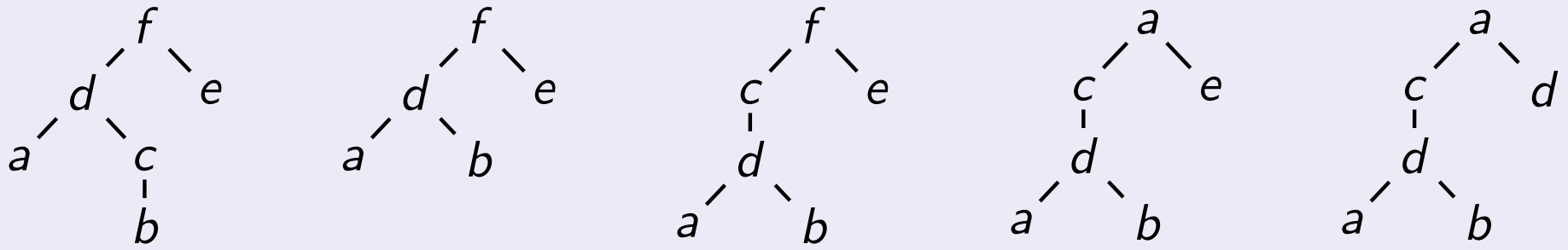
$$\delta(T_1, T_2) = \min\{\gamma(M) | (M, T_1, T_2)\}$$

$$N_1 = \{v | \neg\exists x.(v, x) \in M\}, N_2 = \{w | \neg\exists y.(y, w) \in M\}$$

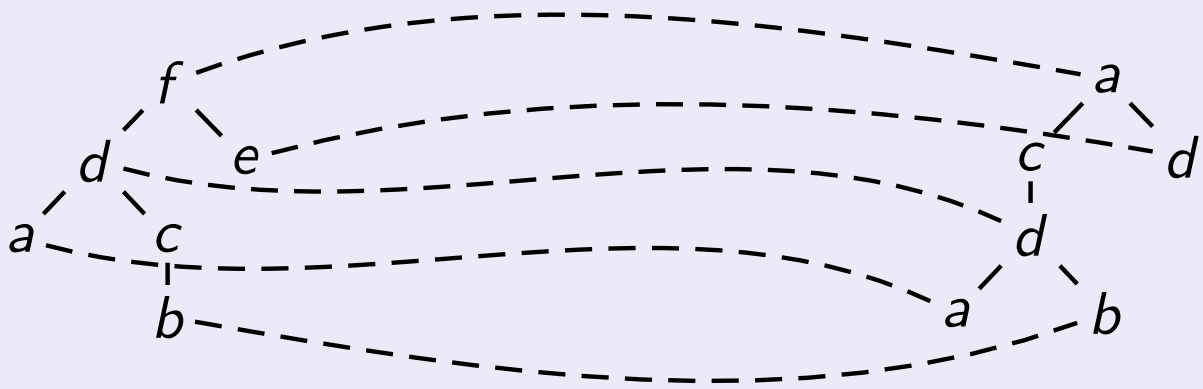
Tree Edit Distance

[Lu, 1979, Tai, 1979]

Edit Script



Edit Mapping



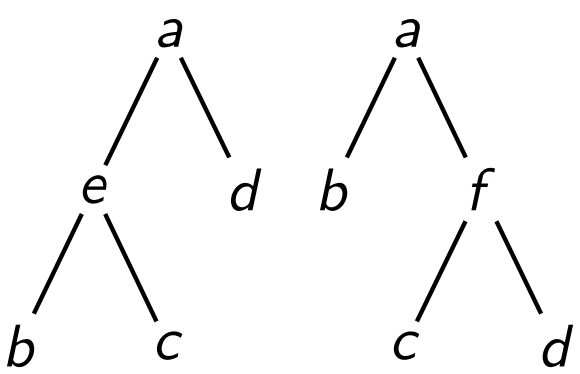
Tree Edit Algorithmus

Seien F_1 und F_2 geordnete Wälder sowie v und w die rechtsäußersten Wurzeln der Bäume in F_1 bzw. F_2

$$
\begin{aligned}
\delta(\theta,\theta) &= 0 \\
\delta(F_1,\theta) &= \delta(F_1 - v,\theta) + \gamma(v \to \lambda) \\
\delta(\theta,F_2) &= \delta(\theta,F_2 - w,\theta) + \gamma(\lambda \to w) \\
\delta(F_1,F_2) &= \min \begin{cases} \delta(F_1 - v,F_2) + \gamma(v \to \lambda) \\ \delta(F_1,F_2 - w,\theta) + \gamma(\lambda \to w) \\ \delta(F_1(v),F_2(w)) + \delta(F_1 - T_1(v),F_2 - T_2(w)) \\ \qquad\qquad +\gamma(v \to w) \end{cases}
\end{aligned}
$$

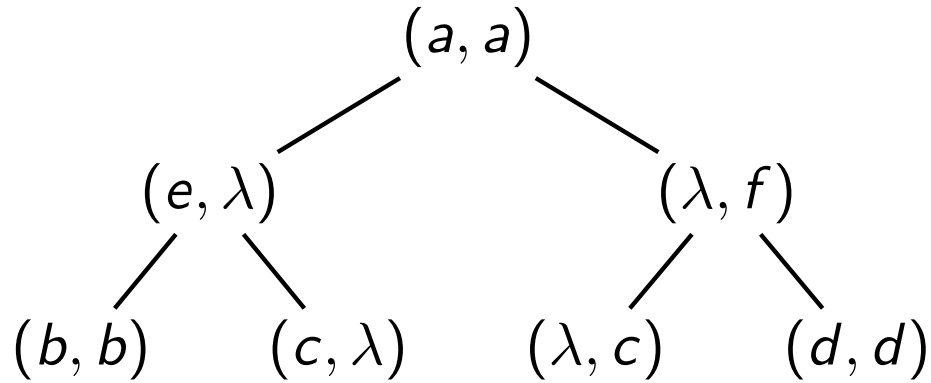
Tree Alignment (Idee)

- λ-Knoten einfügen $\longrightarrow$ Isomorphie
- Bäume T_1 und T_2 überlagern zu A
- $\alpha(T_1, T_2) = \sum_{v \in V(A)} \gamma(v)$
- Spezialfall von Editdistanz $\longrightarrow$ Einfügen vor Löschen



T_1

T_2

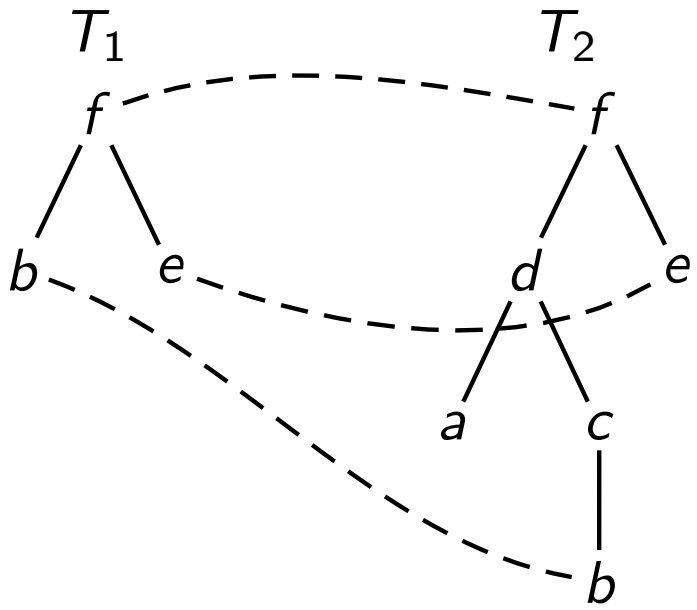


A

$$\delta(T_1, T_2) = 2$$
$$\alpha(T_1, T_2) = 4$$

Tree Inclusion (Idee)

- Knoten in T_2 löschen, bis isomorph zu T_1
- Spezialfall von Editdistanz $\gamma(x \to y) = \begin{cases} \text{wenn } y = \lambda \text{ dann } 0 \\ \text{sonst} > 0 \end{cases}$
- $\delta(T_1, T_2) = 0$



Gliederung

"Ordnungs"-basierte Ansätze

- Definition einer Generalisierungshierarchie über Strukturen
- Berechung der *kleinsten oberen Schranke* zweier Strukturen
- Ähnlichkeit als kleinstes Supremum bezüglich der Halbordnung

Problem

- selten eindeutig

Feature Terme (1)

[Plaza, 1995]

Formalisierung

Feature Term $F = \langle Q, q_r, T, D \rangle$ über *Type* und *Feat*

- Typ Symbole *Type* mit Typhierarchie $\langle Type, \leq \rangle$
- Menge von Attributen *Feat*
- endlicher Knotenmenge Q mit Wurzel $q_r \in Q$
- totale Typisierungsfunktion $T : Q \mapsto Type$ (Knotenlabel)
- partielle *feature value* Funktion $D : Feat \times Q \mapsto Q$ (Kantenlabel)

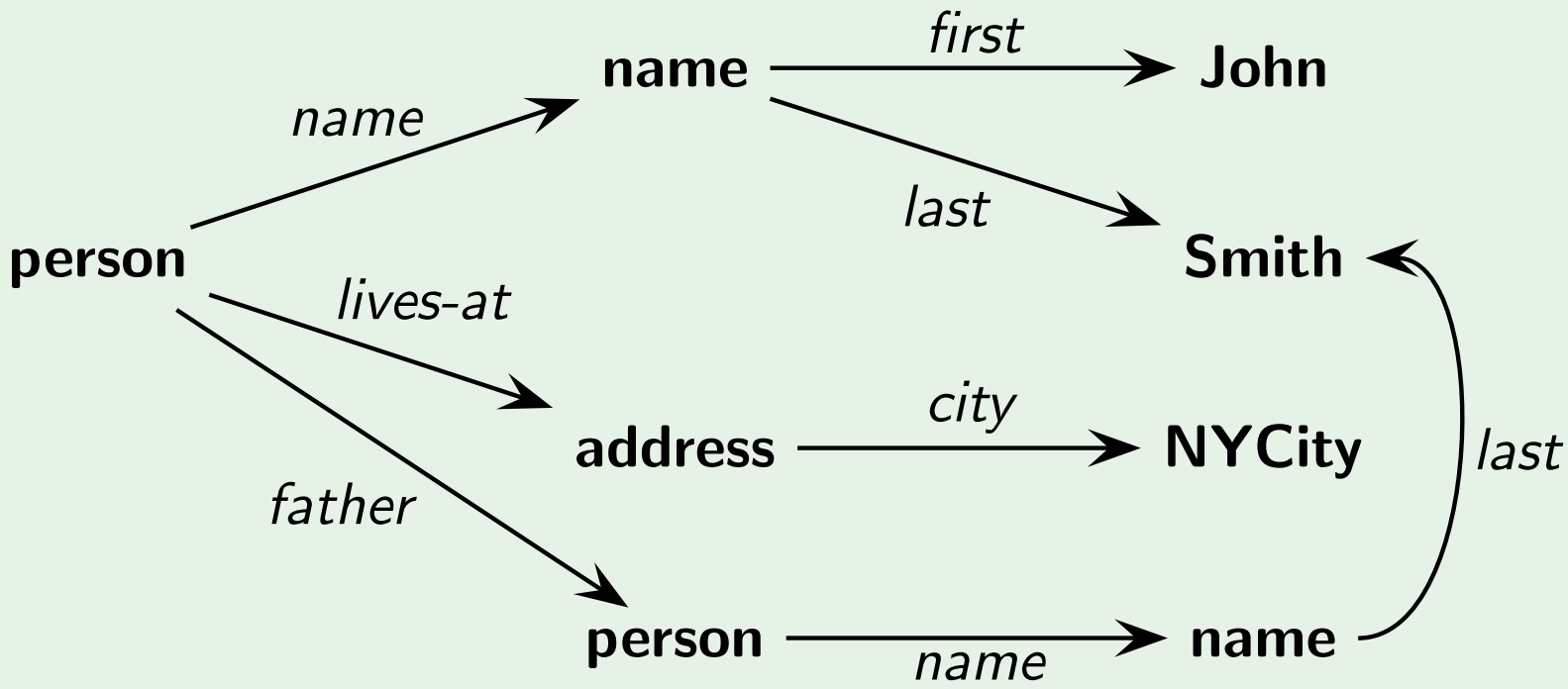
Feature Terme (1)

[Plaza, 1995]

Formalisierung

Feature Term $F = \langle Q, q_r, T, D \rangle$ über *Type* und *Feat*

Example



Feature Terme (2)

Subsumption

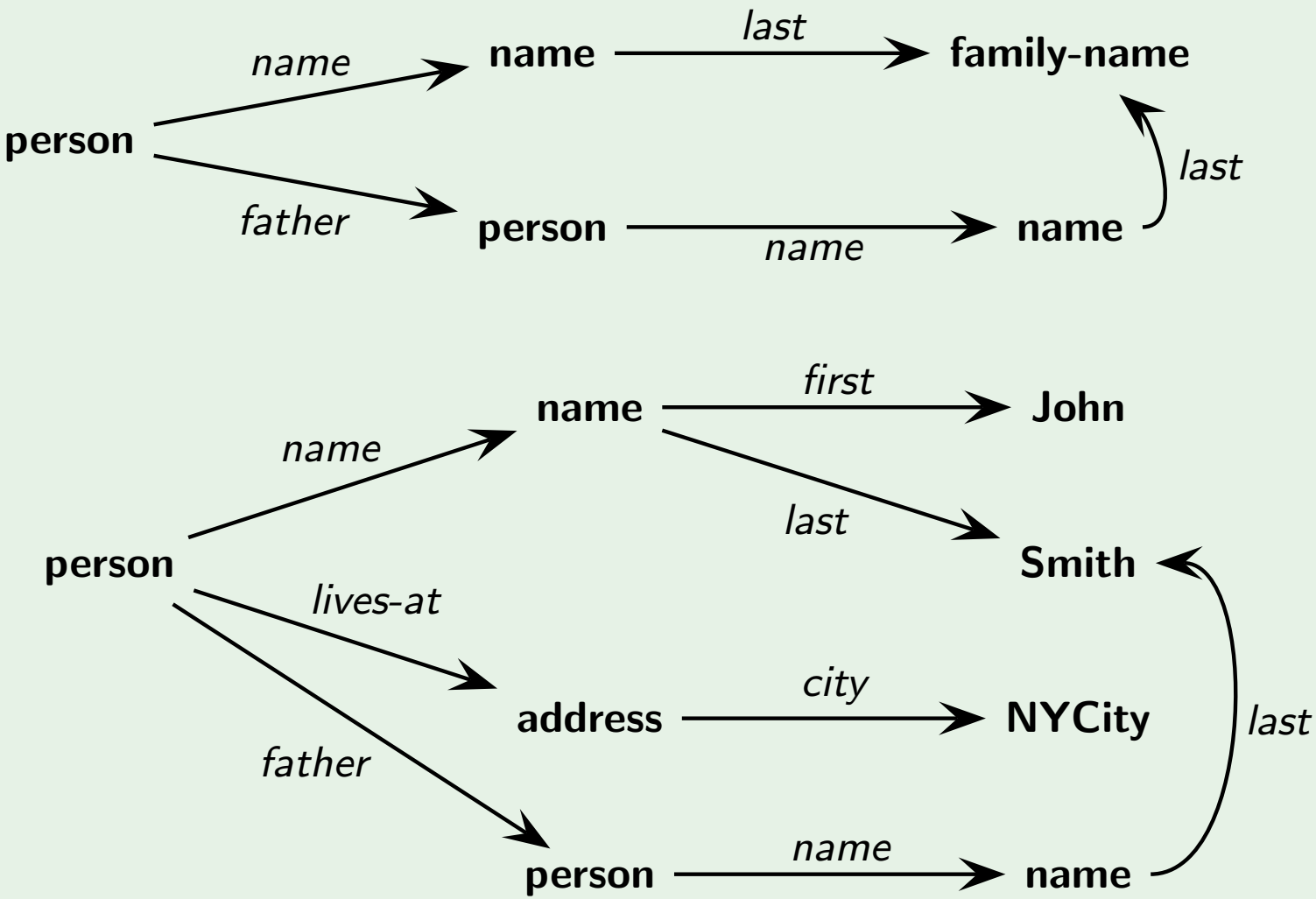
$F = \langle Q, q_r, T, D \rangle$ subsumiert $F' = \langle Q', q'_r, T', D' \rangle$ ($F \preceq F'$) wenn ein Morphismus $h : Q \mapsto Q'$ existiert, so dass:

- $h(q_r) = q'_r$
- $T(q) \preceq T'(h(q))$ für alle $q \in Q$
- $h(D(f, q)) = D'(f, h(q))$ für alle $q \in Q$ und $f \in Feat$ so dass $D(f, q)$ defieniert ist

$\rightarrow$ Subsumption induziert ein Halbordnung auf Feature Termen

Feature Terme (2)

Example



Feature Terme (3)

Antiunifikation

Für $F, F' \in \mathcal{F}$ mit F=$\langle Q, q_r, T, D \rangle$, $F' = \langle Q', q'_r, T', D' \rangle$ unf $Q \cap Q' =$ sei $R = Q \times Q$ so dass:

- $R \subset Q \times Q'$
- $R(q_r, q'_r)$
- $R(D(f, q), D'(f, q')) \rightarrow R(q, q')$ und beide definiert

Die Antiinstanz von F und F' sei definiert als

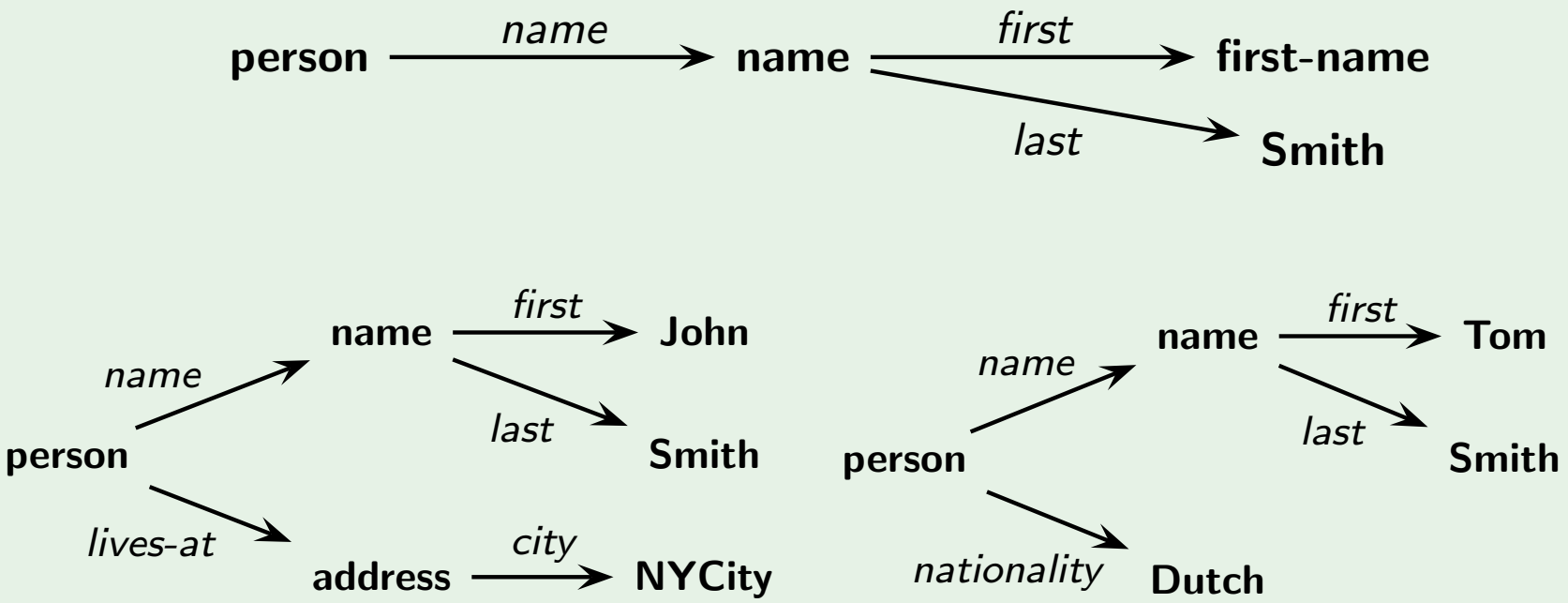
$$F \sqcap F' = \left\langle Q^R, \{q_r, q'_r, T^R, D^R\} \right\rangle$$

mit

$$Q^R = \{\{q, q'\} | R(q, q')\}$$
$$T^R(\{q, q'\}) = T(q) \sqcap T(q')$$
$$D^E(f, \{q, q'\}) = \begin{cases} \{D(f, q), D'(f, q')\} & \text{wenn beide definiert} \\ & \text{undefiniert sonst} \end{cases}$$

Feature Terme (3)

Example



Feature Terme (4)

Ähnlichkeit

- sei $\mathcal{F} = \{f_1, \ldots, f_n\}$ eine Menge von Feature Termen
- sei $f_{i \sqcap j} = f_i \sqcap f_j$
- sei t_0 ein Anfrageterm und $\mathcal{A} = \{f_{0 \sqcap i} | f_i \in \mathcal{F}\}$
- sei $\min_{\preceq}(\mathcal{A}) = f_{0 \sqcap i}$
- dann ist f_i der zu f_0 am ähnlichste Term aus $\mathcal{F}$

Was sonst noch interessant wäre

- String-to-String Edit Distanz (Levenshtein Distanz)
- Similarity Measures on Graphs [Bunke and Messmer, 1994]
- Tree Edit, Tree Alignment, Tree Inclusion für ungeordnete Bäume [Bille, 2005]
- Tree Pattern Matching, Maximum Agreement Tree, Largest/Smallest Common Subtree [Bille, 2005]
- Substrukturen mit Hilfe von Domänenwissen finden [Djoko et al., 1997]
- weitere Feature Extraktionsmethoden (Edge-Matching Distanz [Kriegel and Schönauer, 2003])
- weitere "Ordnungs-basierte" Methoden

Herzlichen Dank!


Bille, P. (2005).
A survey on tree edit distance and related problems.
Theor. Comput. Sci., 337(1-3):217–239.


Bunke, H. and Messmer, B. T. (1994).
Similarity measures for structured representations.
In *EWCBR '93: Selected papers from the First European Workshop on Topics in Case-Based Reasoning*, pages 106–118, London, UK. Springer-Verlag.


Djoko, S., Cook, D. J., and Holder, L. B. (1997).
An empirical study of domain knowledge and its benefits to substructure discovery.
IEEE Trans. on Knowl. and Data Eng., 9(4):575–586.


Kriegel, H.-P. and Schönauer, S. (2003).
Similarity search in structured data.
In *DaWaK*, pages 309–319.


Lu, S.-Y. (1979).
A tree-to-tree distance and its application to cluster analysis.

IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-1:219–224.


Papadopoulos, A. N. and Manolopoulos, Y. (1999).
Structure-based similarity search with graph histograms.
In *DEXA '99: Proceedings of the 10th International Workshop on Database & Expert Systems Applications*, page 174, Washington, DC, USA. IEEE Computer Society.


Plaza, E. (1995).
Cases as terms: A feature term approach to the structured representation of cases.
In *ICCBR '95: Proceedings of the First International Conference on Case-Based Reasoning Research and Development*, pages 265–276, London, UK. Springer-Verlag.


Tai, K.-C. (1979).
The tree-to-tree correction problem.
J. ACM, 26(3):422–433.


Yang, R., Kalnis, P., and Tung, A. K. H. (2005).

Similarity evaluation on tree-structured data.
In *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 754–765, New York, NY, USA. ACM.