

CRIL-Graphen

Jan Helwich -
Graphenbasierte Navigation eines
geometrischen Agenten: Integration von
Perzeption und Instruktion

Vortrag von
Martin Malzahn

Gliederung

- Taxonomien
- CRIL-Graphen
- Ähnlichkeitsmaße
 - von Konzepten
 - von Knoten
 - von Graphen
 - Algorithmus zur Berechnung der Ähnlichkeit

Einführung

- Interne Repräsentation mittels CRIL-Graphen
- matchen von Objekten der Perzeption mit solchen der Instruktion

Taxonomien

- Konzepte werden in Relation gesetzt
 - „Vater-Sohn“ Beziehung → Subsumption
 - Geschwister Beziehung
 - Exklusion
- Informationen werden vererbt
- Je tiefer im Baum, desto spezifischer das Konzept → höherer Informationsgehalt

Taxonomien: Relationen

- Subsumption
 - reflexiv
 - antisymmetrisch
 - transitiv
- Exklusion
 - irreflexiv
 - symmetrisch

Taxonomien: Definitionen

- Spezifischster Vater:

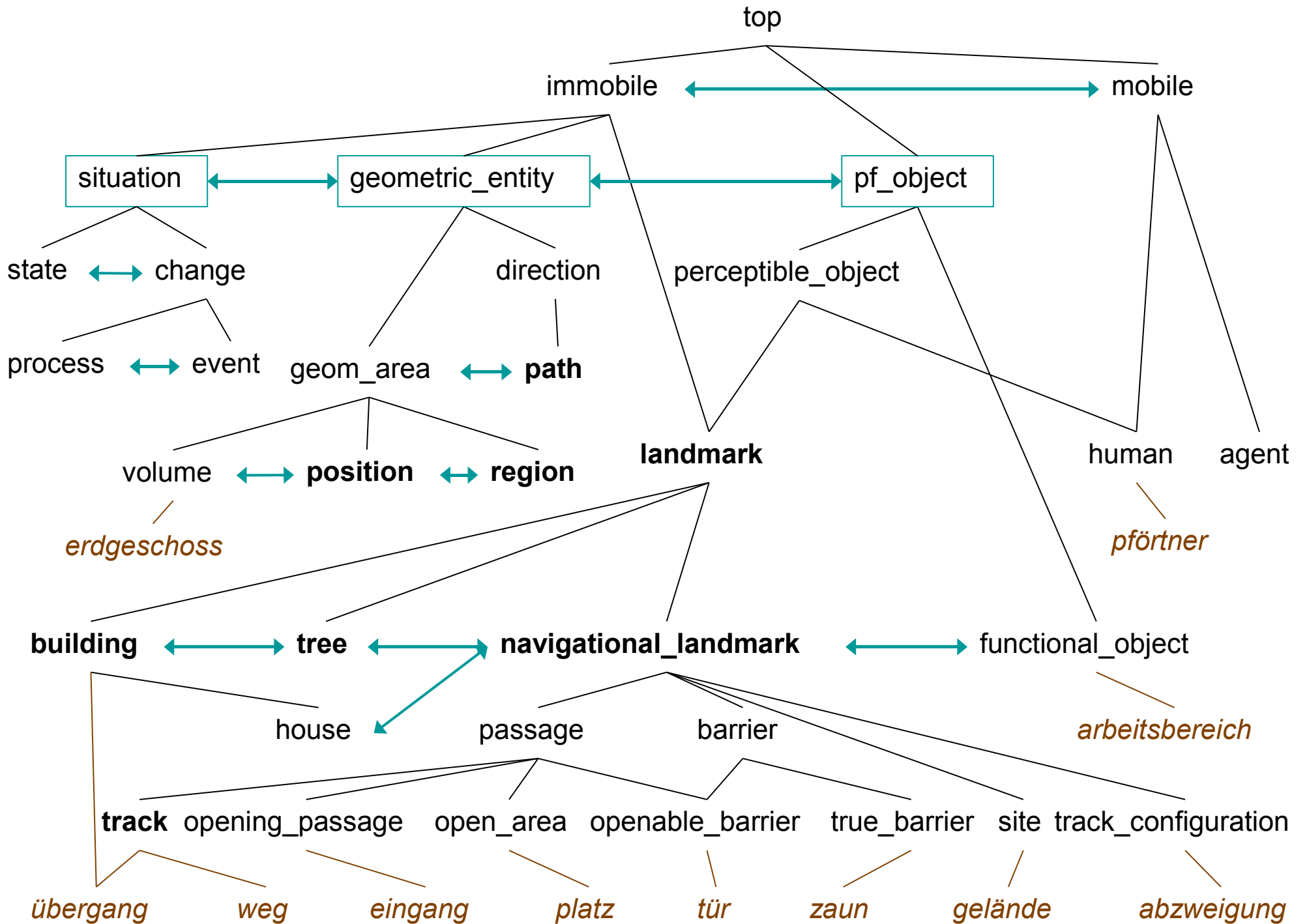
Seien c , c_1 und c_2 Elemente einer Taxonomie

$T = (S, \leq, \leftrightarrow)$. Dann ist c ein spezifischster Vater von c_1 und c_2 , wenn c ein Konzept aus S ist, dass sowohl c_1 als auch c_2 subsumiert, und für alle Konzepte c' aus S , die sowohl c_1 als auch c_2 subsumieren, gilt, dass c von c' subsumiert wird.

$$c = \text{spezVater}_T(c_1, c_2) \Leftrightarrow c_1 \leq c \wedge c_2 \leq c \wedge \forall c' \in S: \\ [c_1 \leq c' \wedge c_2 \leq c' \Rightarrow c \leq c']$$

Taxonomien: Distanz von Konzepten

- Je größer die Distanz zweier Konzepte, desto geringer die Ähnlichkeit
- Ermitteln von Distanz
 - Subsumptionsgraph erstellen
 - kürzesten Pfad zwischen c und $\text{spezVater}(c,c')$
 - kürzesten Pfad zwischen c' und $\text{spezVater}(c,c')$
 - Summe der Länge der Pfade
- Tiefe eines Konzepts ist die Distanz des Konzepts zum **T**-Konzept



CRIL-Graphen

$$(V, E, T_A, T_R, op)$$

- Knoten die Referenzobjekte darstellen
 - $ck = (SOR, EIG, NAM)$
- Gerichtete Kanten für Relationen zwischen Objekten
- Taxonomie der Attribute
- Taxonomie der Relationen
- Operatorfunktion

Perzeptions- und Instruktionsgraphen

- Instruktionsgraph, kurz I-Graph, repräsentiert das Wissen der Wegbeschreibung und enthält sowohl
 - I-Knoten die Objekte der Instruktion beschreiben
 - als auch P-Knoten die perzipierte Objekte darstellen und koreferent zu einem I-Knoten sind
- Perzeptionsgraph, kurz P-Graph, repräsentiert das Wissen aus der Wahrnehmung und besteht ausschliesslich aus P-Knoten
- Taxonomien beider Graphen sind identisch

Ähnlichkeitsmaße

- Um koreferente Knoten in zwei Graphen zu finden wird auf die Ähnlichkeit zurückgegriffen

$$\ddot{a}(A, B) = \frac{2i(gi(A, B))}{i(A) + i(B)}$$

Ähnlichkeit von Konzepten

- Je tiefer der spezifischste Vater, desto ähnlicher die Konzepte
- Je höher die Tiefe zwischen den Konzepten und dem spezifischsten Vater, desto unähnlicher die Konzepte

$$\ddot{a}(c_1, c_2) = \frac{2t_T(sV_T(c_1, c_2))}{t_T(c_1) + t_T(c_2)}$$

Ähnlichkeit von Konzepten (2)

- Falls T Spezifischster Vater, dann geht die Ähnlichkeit gegen 0
- Schliessen sich die Konzepte gegenseitig aus, geht die Ähnlichkeit auch gegen 0

Ähnlichkeit von Knoten in CRIL-Graphen

- Information durch Sorten, Eigenschaften und Namen bestimmt
- Ähnlichkeit wird mittels Vergleich von Sorten und Eigenschaften geprüft
- Zwei Knoten sind kompatibel, wenn der eine keine Sorten beinhaltet, die eine Sorte des anderen ausschliessen
- Zwei Knoten sind identisch, wenn sie den gleichen Namen tragen oder ihre Sorten übereinstimmen

Ähnlichkeit von CRIL-Graphen

- Oberflächliche Ähnlichkeit:
Ähnlichkeit von Objekten und Attributen
- Strukturelle Ähnlichkeit:
Anzahl der Kanten, Knotengrade, Pfadlängen...
sind ähnlich
- Thematische Ähnlichkeit
Relationen zwischen Knoten und Muster von
Konzepten und Relationen sind ähnlich

Ähnlichkeit von CRIL-Graphen (2)

Ganz grob:

- Es werden Knoten und deren Relationen in betracht gezogen und auf Ähnlichkeit geprüft

Algorithmus zur Bestimmung der Ähnlichkeit – Schritt 1

Prozedur:

PRODUKTVERKNÜPFUNG(A , B)

1. `knotenMatrix := BILDEKNOTENMATRIX(A, B)`
2. **for all** `kA ∈ VA` **do**
3. **for all** `kB ∈ VB` **do**
4. `kP := VERKNÜPFE(kA , kB)`
5. `KNOTENEINFÜGEN(P, kP)`
6. `knotenMatrix[kA][kB] := kP`
7. **end**
8. **end**
9. **for all** `rA ∈ EA` **do**
10. **for all** `rB ∈ EB` **do**
11. `rP := VERKNÜPFE(rA , rB)`
12. **if** `rP ≠ nil` **do**
13. **if** `(knotenMatrix[ANFANG(rA)][ANFANG(rB)] ≠ null ∧
 knotenMatrix[ENDE(rA)][ENDE(rB)] ≠ null)` **do**
14. `RELATIONEINFÜGEN(P , rP)`
15. **end**
16. **end**
17. **end**
18. **end**
19. **return** P

Algorithmus zur Bestimmung der Ähnlichkeit – Schritt 2

Prozedur 4.5.3: SETZEKNOTENDURCH(Verknüpfung G, CRIL-Knoten m)

1. **for all** $n \in K_G$ **do**
2. **if** inkonsistent(m , n)
3. LÖSCHE(G , n)
4. **end**

Prozedur:

VERKNÜPFEN(A, B)

1. $C := \text{PRODUKTVERKNÜPFUNG}(A, B)$
2. **while** $\neg \text{kons}(C)$ **do** //vgl. Definition 4.4.4
3. knoten := WÄHLEKNOTEN(C)
4. $C' := \text{KOPIERE}(C)$
5. SETZEKNOTENDURCH(C, knoten)
6. FÜGE_HINZU(offen, C)
7. LÖSCHE(C', knoten)
8. FÜGE_HINZU(offen, C')
9. SORTIEREBZGL(offen, ähnlichkeitsgehalt)
10. $C := \text{KOPF}(\text{offen})$
11. **end**
12. **return** C