

Lernmodul 2 Geometr.-topol. Datenstrukturen



Geometr.-topol. Datenstrukturen **Übersicht**

- Überblick/Motivation
- Fokus
- Geometrisch-topologische Datenstrukturen
 - Spaghetti
 - Spaghetti mit Punktobjekten
 - Knoten-Maschen-Struktur
 - Geflügelte Kanten
- Zusammenfassung



Geometr.-topol. Datenstr. **Überblick/Motivation**

Wir haben bisher kennengelernt:

- **Objektorientierte** Modellierung der Realität und Darstellung in UML (Abschnitt 2)
- Abbildung dieser Objekte auf **Tabellen** (Abschnitt 3)
- Grundlegende **Konzepte** der Modellierung des Raumes (Abschnitt 4)
- Mathematische Grundlagen der **Topologie** (Kapitel 5)
- **Landkarten**, insbesondere deren Topologie (Kapitel 6)

Wir wollen diese Bereiche jetzt zusammenführen: **Landkarten objektorientiert** als **Vektor-Struktur** modellieren, in Form von **Tabellen** darstellen, sowie die **Beziehungen** der Objekte betrachten



Geometr.-topol. Datenstrukturen **Fokus**

- **Wir haben:**
 - eine Landkarte, eingebettet in die euklidische Ebene
 - die Geometrie, gegeben durch Angabe der Koordinaten von Punkten/Knoten
- **Uns interessiert besonders die geometrisch-topologische Konsistenz:**
 - Beobachtung: Die Topologie kann im Prinzip aus der Geometrie hergeleitet werden.
 - Problem: "Wieviel" Topologie wird **explizit** repräsentiert?
- **Wir betrachten mehrere Optionen, die keine bis viel Topologie repräsentieren.**



Spaghetti Topologie-freie Datenstruktur(en)



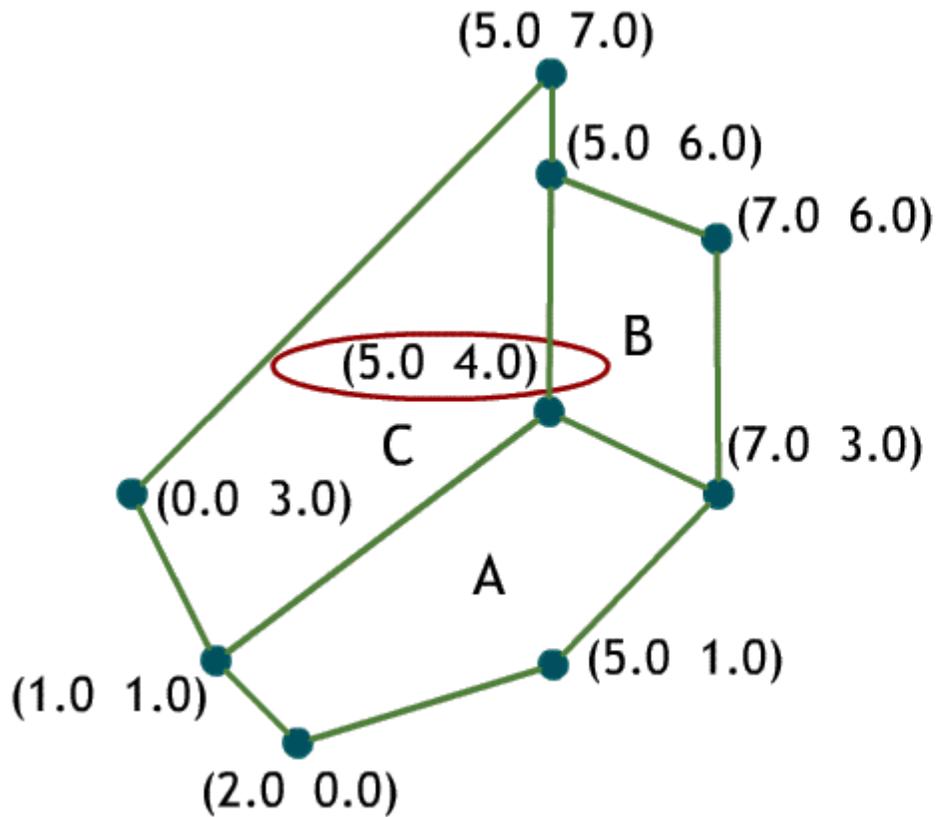
Spaghetti **Allgemeines**

- jede Masche wird durch eine Liste von Koordinatenpaaren ihrer Eckpunkte repräsentiert (>>**Beispiel**)
- zwangsläufig redundante Speicherung von Koordinatenpaaren
- reine Geometrie
- keine explizite Topologie

Übung: Wie läßt sich die Nachbarschaft zweier Maschen A und B feststellen?



Spaghetti Beispiel

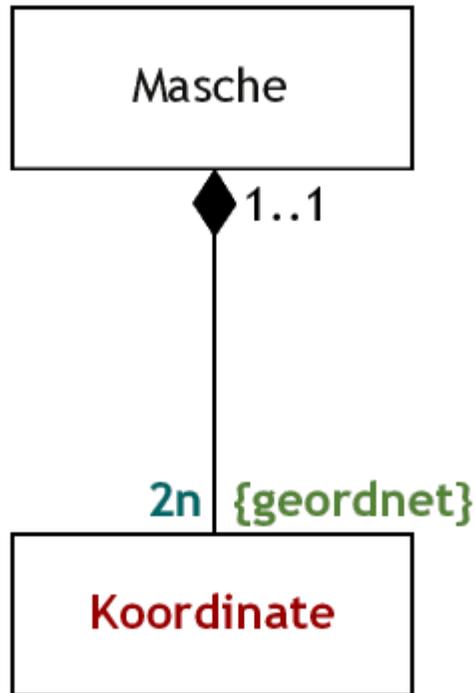


Fläche: x - Koord: y - Koord:

A:	2.0	0.0
	5.0	1.0
	7.0	3.0
	5.0	4.0
	1.0	1.0
B:	5.0	4.0
	7.0	3.0
	7.0	6.0
	5.0	6.0
C:	5.0	4.0
	5.0	6.0
	5.0	7.0
	0.0	3.0
	1.0	1.0



Spaghetti UML-Diagramm für Koordinatenpaare



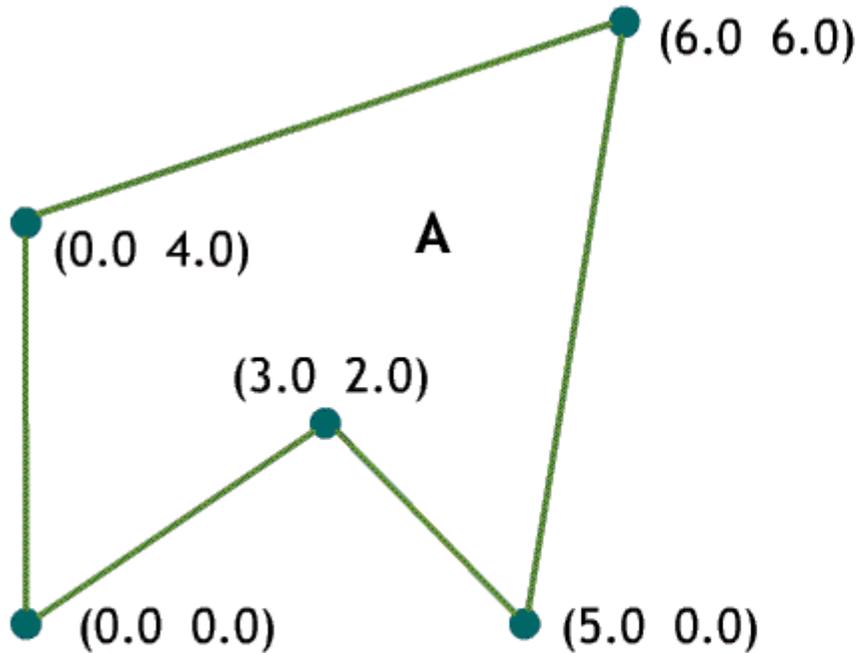
- jede Masche besteht aus Paaren von **Koordinaten**: $2n$ bei n Eckpunkten
- **geordnete** Folge von Koordinaten: Gestalt der Masche ist durch die Folge festgelegt:
 $\gg [0.0,0.0, 3.0,2.0, 5.0,0.0, 6.0,6.0, 0.0,4.0]$ ist ungleich
 $[0.0,0.0, 5.0,0.0, 3.0,2.0, 6.0,6.0, 0.0,4.0]$

Zur Erinnerung:

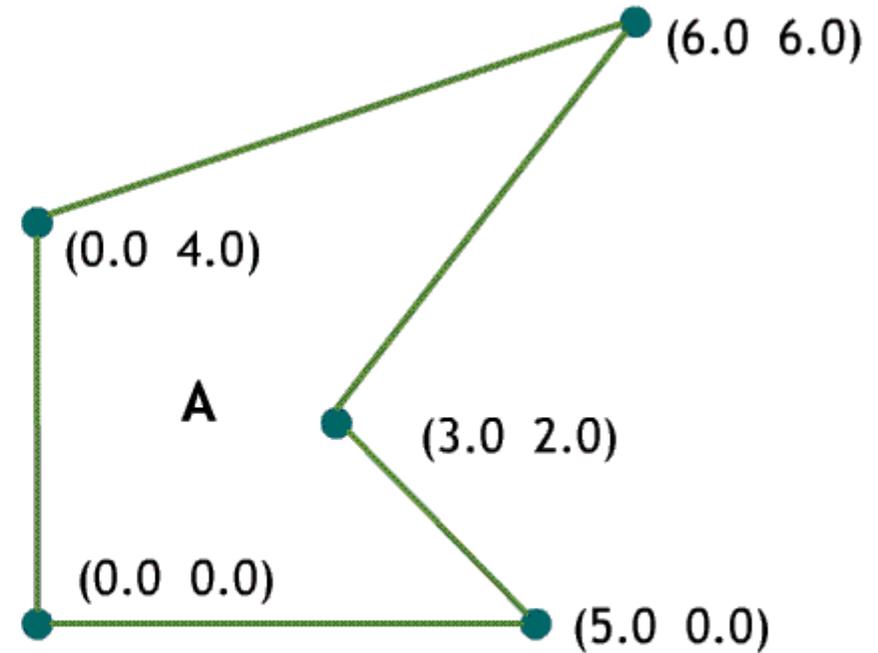
◆ $\hat{=}$ Komposition



Spaghetti **Beispiel zur geordneten Folge**



A = [0.0,0.0, 3.0,2.0, 5.0,0.0,
6.0,6.0, 0.0,4.0]



A = [0.0,0.0, 5.0,0.0, 3.0,2.0,
6.0,6.0, 0.0,4.0]

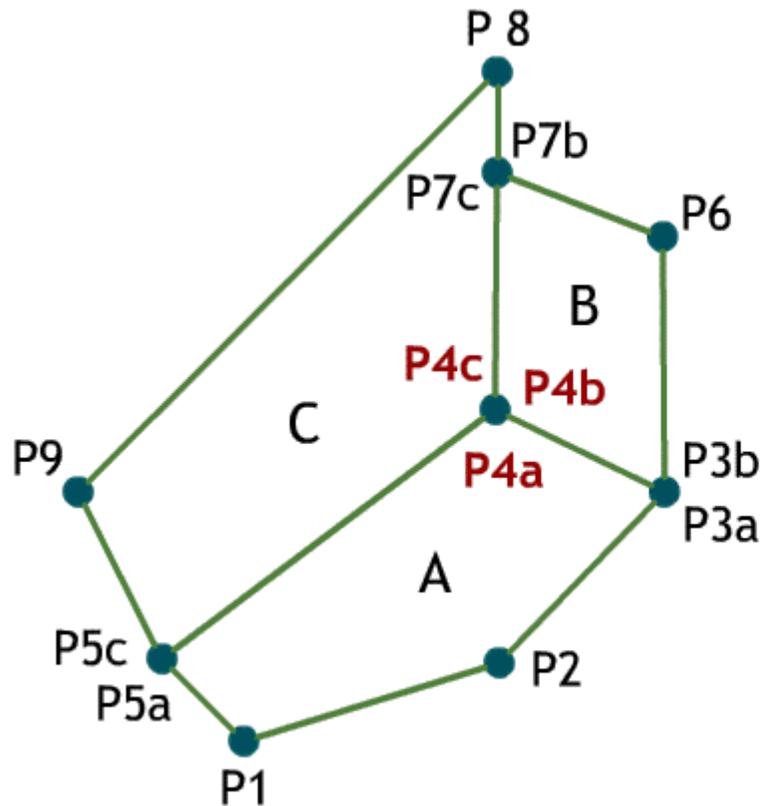


Spaghetti **Komposition von Punktobjekten**

- "Verbesserung" der einfachen Spaghettistruktur
- Paare von Koordinaten werden zu **Punktobjekten**
- jede Masche wird durch eine Liste von Punktobjekten repräsentiert (>>**Beispiel**)
- jede Masche besitzt ihre **eigenen** Punktobjekte (Komposition)



Spaghetti Beispiel zur Komposition von Punktobj.

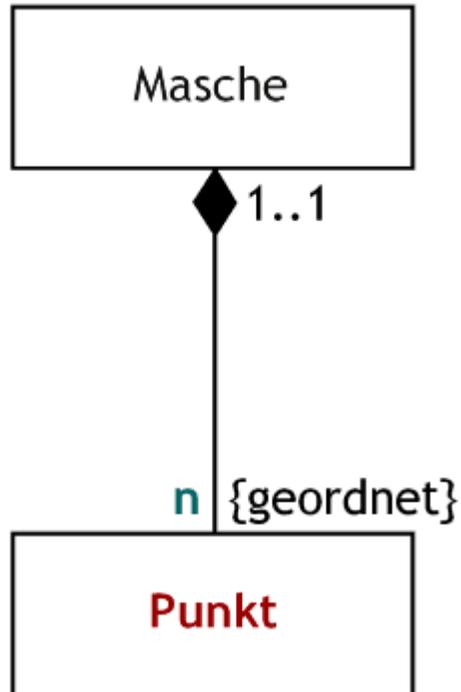


A:	P1	P2	P3a
	P4a	P5a	
B:	P4b	P3b	P6
	P7b		
C:	P4c	P7c	P8
	P9	P5c	

Punkte:	x - Koord:	y - Koord:
P1	2.0	0.0
P2	5.0	1.0
P3a	7.0	3.0
P3b	7.0	3.0
P4a	5.0	4.0
P4b	5.0	4.0
P4c	5.0	4.0
...		



Spaghetti UML-Diagramm für Punktobjekte



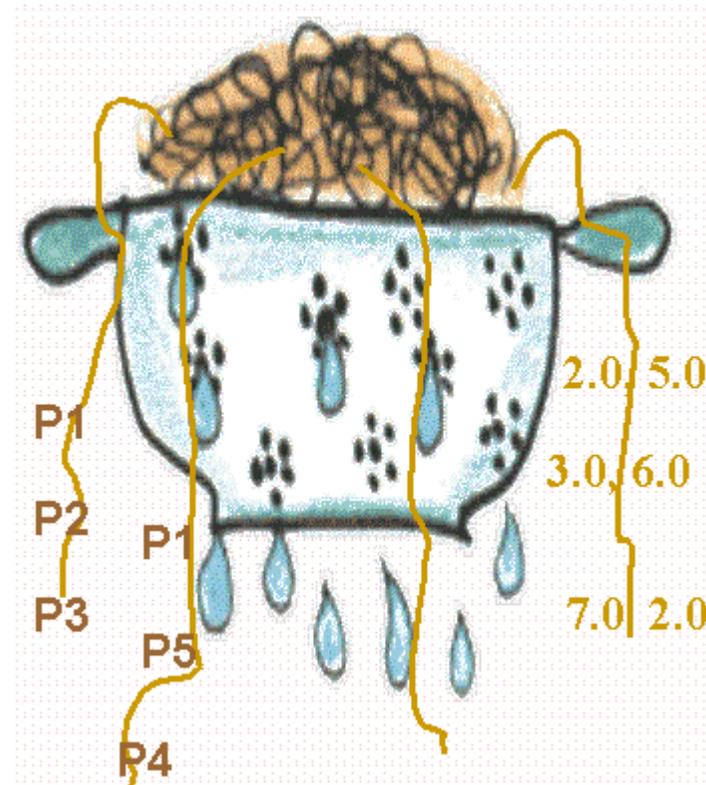
- jede Masche besteht aus **n Punktobjekten**
- Gestalt der Masche ist wieder durch die Folge der Punkte festgelegt:
[P1, **P2,P3**, P4, P5] ist ungleich [P1, **P3,P2**, P4, P5]



Spaghetti Vor- und Nachteile

Für beide bisher betrachteten Datenstrukturen gilt:

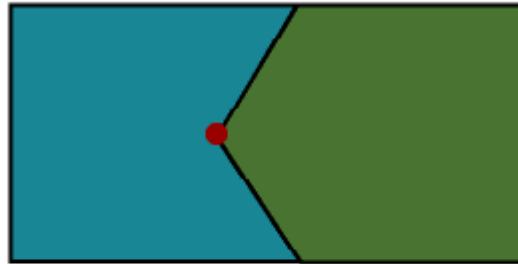
- Vorteile
 - bequem für Flächenberechnung
 - gut für Graphikprogramme
 - Zeichnen von Polygonen
- Nachteile
 - Topologie nur implizit
 - fehleranfällig
 - redundante Daten, wenig änderungsfreundlich
 - >> **Beispiel:** Korrektur von Punktkoordinaten



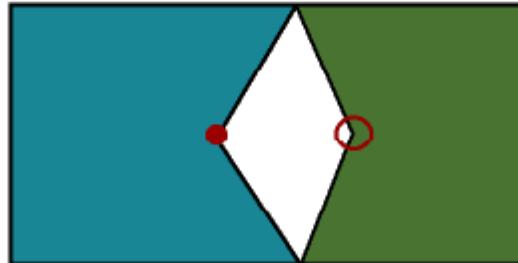
Spaghetti **Beispiel typischer Fehlerfall**

Änderung der Koordinaten eines gemeinsamen Punktes

vorher:



nachher:

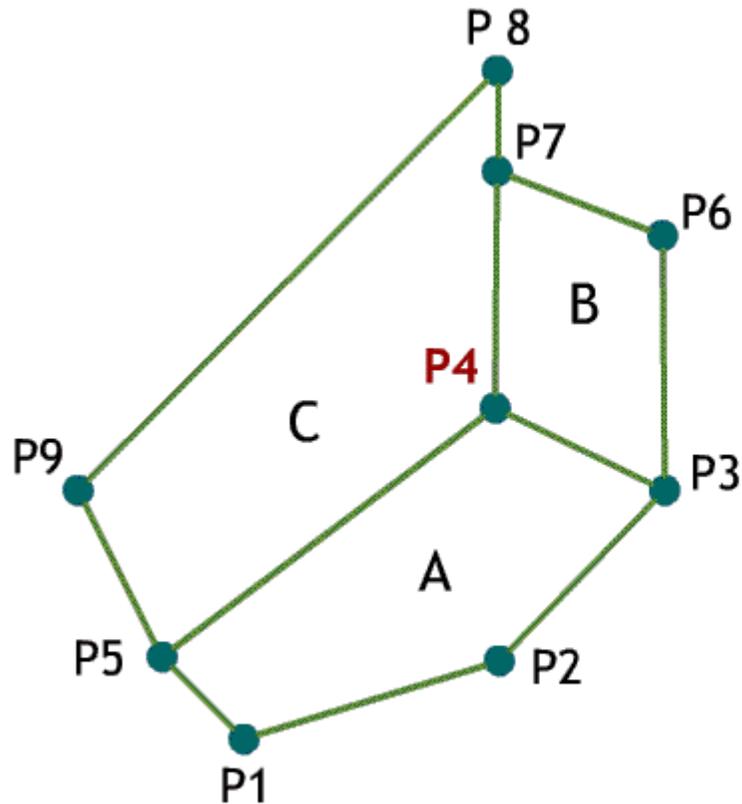


Spaghetti **Punktobjekte ohne Redundanz**

- "Verbesserung" der Komposition von Punktobjekten
- redundante Punktobjekte eines Punktes werden zu **einem** Punktobjekt
- jede Masche wird durch eine Liste von Punktobjekten repräsentiert (>>**Beispiel**)
- Maschen teilen sich die Punktobjekte (Aggregation)



Spaghetti Beispiel für Punktobj. ohne Redundanz



Flächen:

A: P1 P2 P3 **P4** P5

B: **P4** P3 P6 P7

C: **P4** P7 P8 P9 P5

Punkte: x - Koord: y - Koord:

P1 2.0 0.0

P2 5.0 1.0

P3 7.0 3.0

P4 5.0 4.0

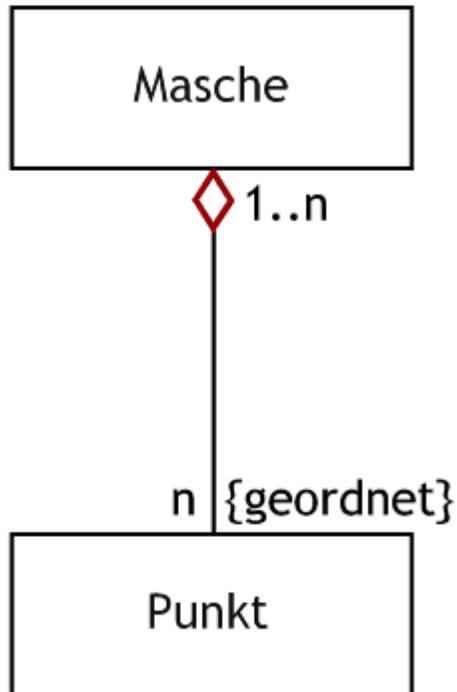
P5 1.0 1.0

P6 7.0 6.0

...



Spaghetti UML für Punktobj. ohne Redundanz



- jede Masche besteht aus n Punktobjekten
- Gestalt der Masche ist durch die Folge festgelegt
- Integritätsbedingung: Verschiedene Punkte können nicht die gleichen Koordinaten haben

Beachte:

Redundanzfreiheit kann durch dieses UML-Diagramm **nicht** erzwungen werden.

Zur Erinnerung:

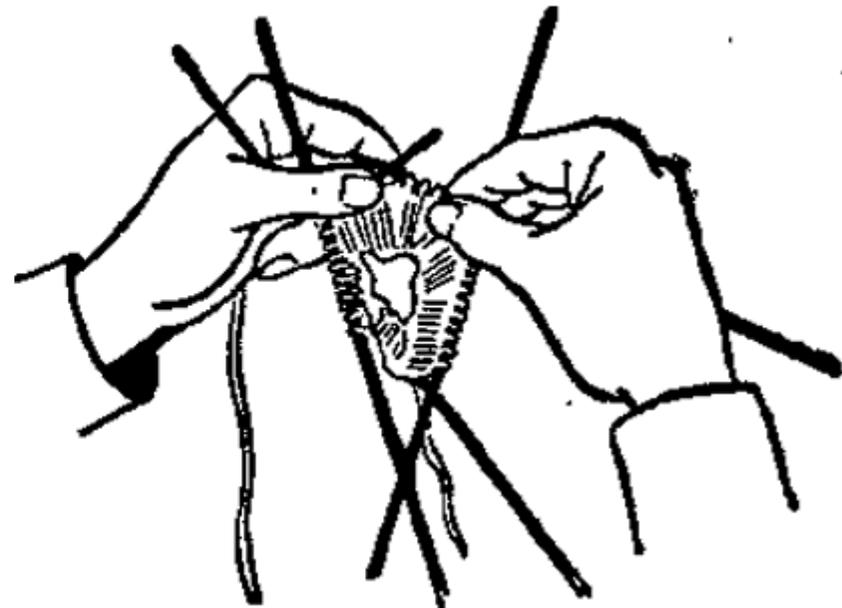
 $\hat{=}$ **Aggregation**



Knoten-Maschen-Struktur **Topologische Struktur I**



Quelle: <http://www.uni-wuerzburg.de>



Quelle: <http://www.puk.de>

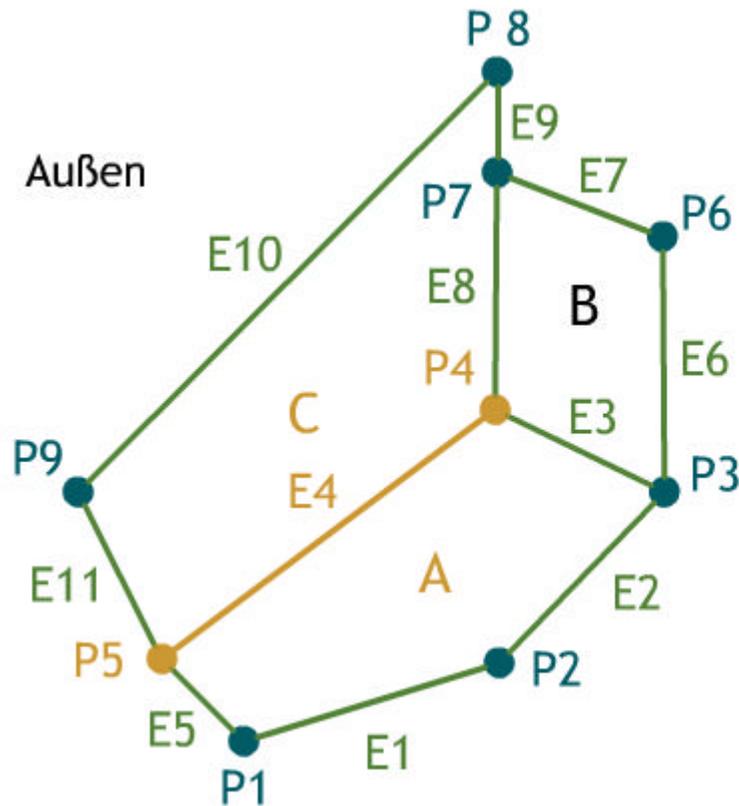


Knoten-Maschen-Struktur **Allgemeines**

- zu jeder Kante werden Anfangs- bzw. Endknoten sowie linke und rechte Masche gespeichert(>>**Beispiel**)
- Geometrie wird redundanzfrei repräsentiert: Knoten stellen eigene Objekte dar
- Adjazenz von Maschen wird explizit repräsentiert



Knoten-Maschen-Struktur **Beispiel**



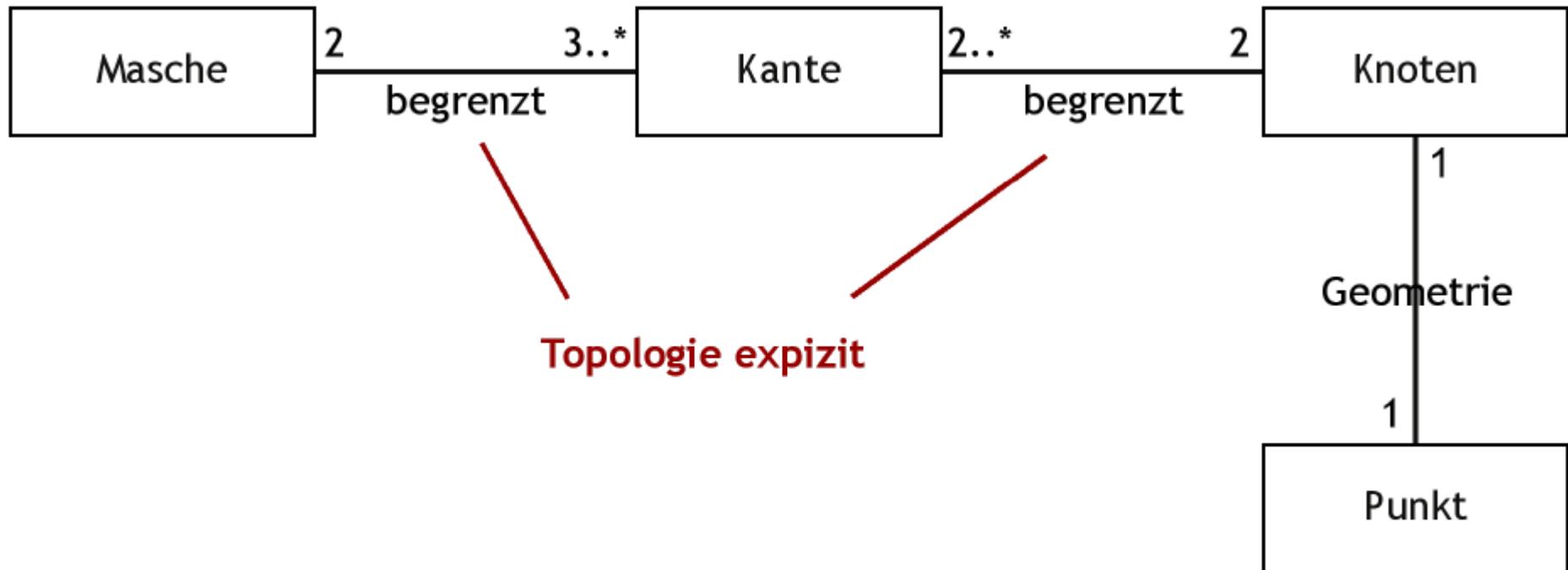
Kante	Anfangsknoten	Endknoten	linke Masche	rechte Masche
E1	P1	P2	A	Außen
E2	P2	P3	A	Außen
E3	P3	P4	A	B
E4	P4	P5	A	C
E5	P5	P1	A	Außen
E6	P3	P6	B	Außen
...				

Knoten
P1	2.0	0.0
P2	5.0	1.0
...		

A 11x



Knoten-Maschen-Struktur UML-Diagramm



Knoten-Maschen-Struktur **Vor- und Nachteile**

- Vorteile:
 - Geometrie ist **redundanzfrei**
 - **Topologie** ist explizit
 - bei Änderungen können Fehler leichter vermieden werden
- Nachteil:
 - der Kantenumring ist nicht direkt gegeben, sondern muß berechnet werden
- **Lösung: Kanten mit Flügeln**



Geflügelte Kanten **Topologische Struktur II**

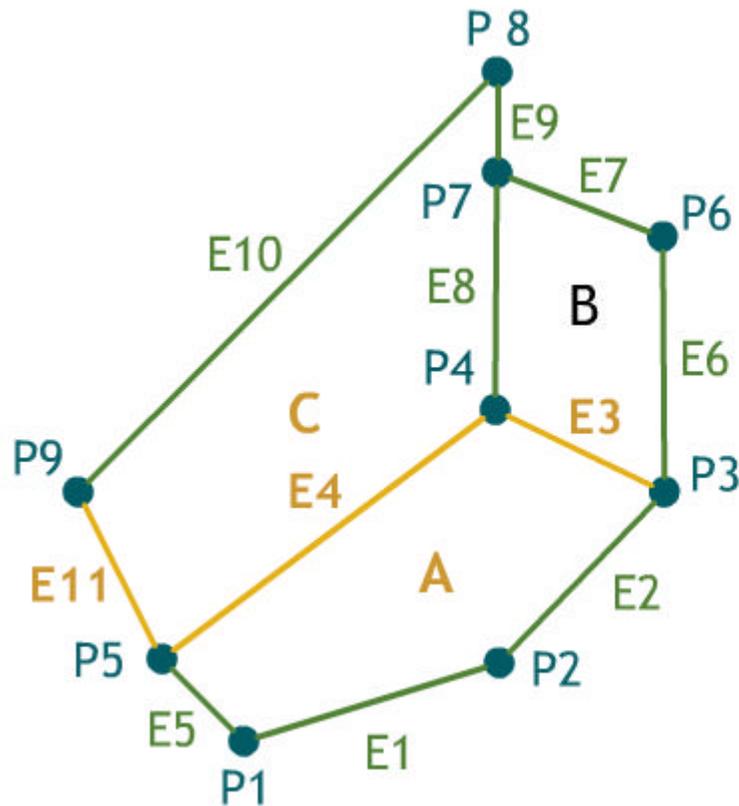


Geflügelte Kanten **Allgemeines**

- Erweiterung der Knoten-Maschen-Struktur um zwei Elemente (>>**Beispiel**):
 - **Vorgängerkante** im Umring der linken Masche
 - **Nachfolgerkante** im Umring der rechten Masche
- Geometrie wird redundanzfrei repräsentiert
- Topologie wird explizit repräsentiert
- Maschenumring läßt sich leicht berechnen



Geflügelte Kanten **Beispiel**



Nachfolger im Umring
der rechten Masche

Vorgänger im Umring
der linken Masche

Kanten:

E1	P1	P2	A	Außen	E5	E2
E2	P2	P3	A	Außen	E1	E6
E3	P3	P4	A	B	E2	E8
E4	P4	P5	A	C	E3	E11
E5	P5	P1	A	Außen	E4	E1
E6	P3	P6	B	Außen	E3	E7
...						

A 11x



Geometr.-topol. Datenstr. **Zusammenfassung**

- Spaghetti mit **Koordinatenpaaren**: redundante Geometrie, keine Topologie
- Spaghetti mit **Komposition von Punktobjekten**: redundante Geometrie, keine Topologie
- Spaghetti mit **Aggregation von Punktobjekten**: **redundanzfreie Geometrie**, keine Topologie
- **Knoten-Kanten-Struktur**: redundanzfreie Geometrie, **explizite Topologie**, Maschenumring muß berechnet werden
- **geflügelte Kanten**: redundanzfreie Geometrie, explizite Topologie, **Maschenumring leicht zu berechnen**



Geometrisch-topol. Datenstrukturen **Literatur**

Worboys, Michael F.: GIS: A Computing Perspective. Taylor & Francis Inc., London
1995

