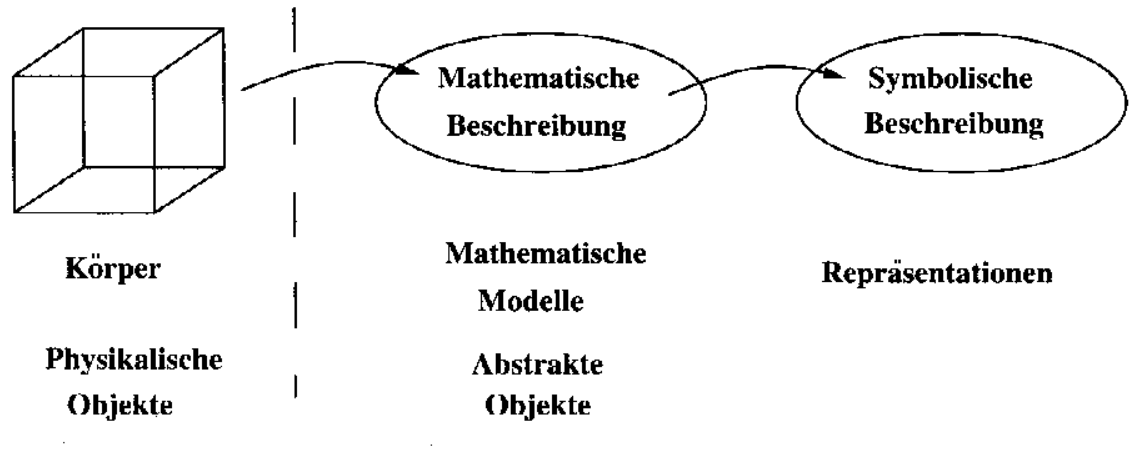
8. Modelle für feste Körper

Modell: Abbild der Realität, welches bestimmte Aspekte der Realität repräsentiert (und andere ausblendet)

- mathematische Modelle
- symbolische Modelle
- Datenmodelle
- Experimentalmodelle

Solid Modelling (Festkörpermodellierung):
mathematische / datenstruktur-orientierte Darstellung geschlossener 3D-Körper

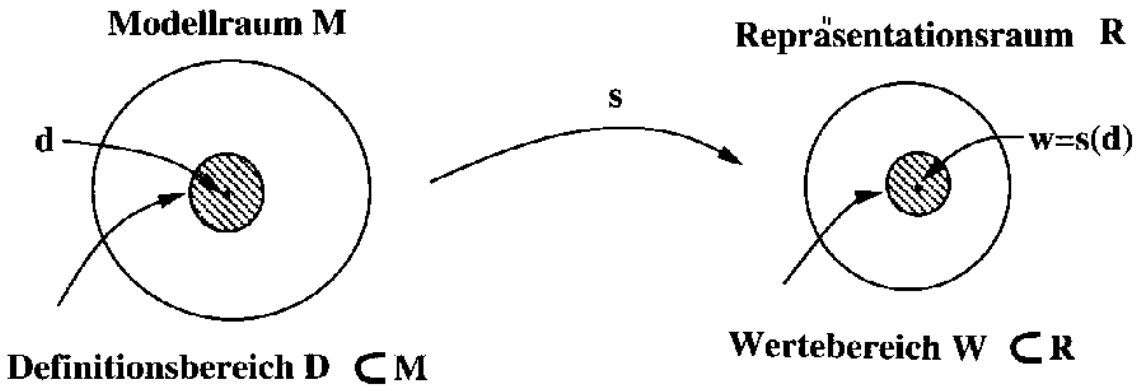
2-stufiger Prozess:



(aus Encarnação et al. 1997)

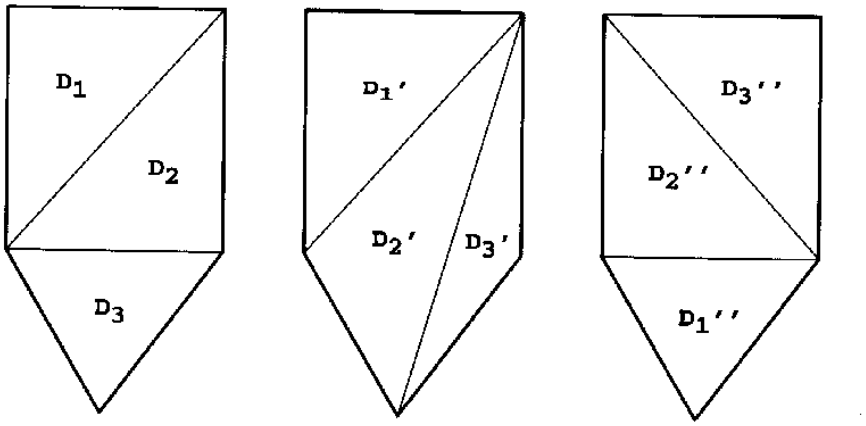
mathematischer Modellraum *M*
symbolischer Repräsentationsraum *R*

Repräsentations-Relation *s*: $M \rightarrow R$
("Repräsentations-Schema")



s eindeutig: jedes Modell aus *D* besitzt genau eine Darstellung in *W* (d.h. *s* ist Funktion).

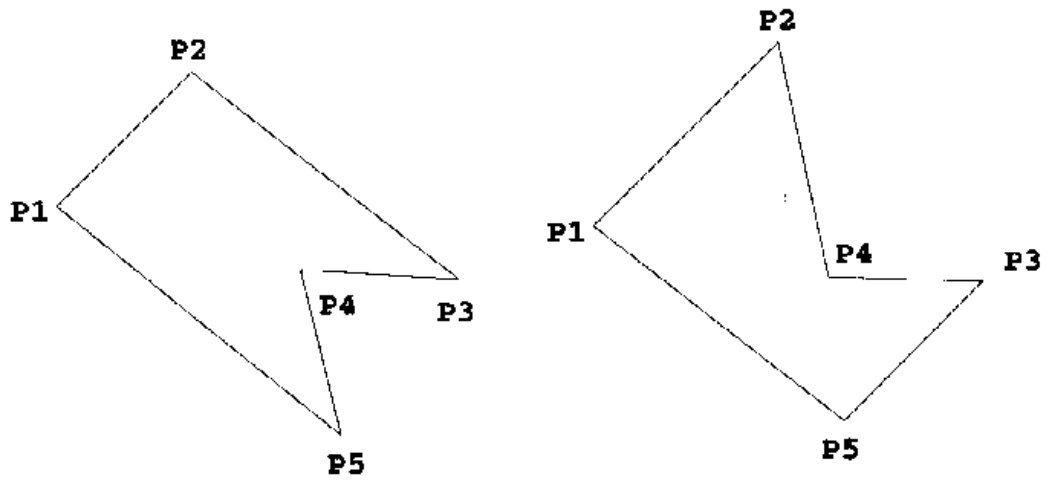
Beispiel einer nicht eindeutigen Repr.:
Darstellung von Polygonen als endliche Vereinigung von Dreiecken



s vollständig: die Relation ist injektiv, d.h. durch jede gültige Repräsentation wird genau ein Modell aus *D* dargestellt.

(wenn das nicht gilt, tritt beim Übergang zur symbolischen Darstellung Informationsverlust ein.)

Beispiel einer nicht vollständigen Repräsentation:
Darstellung eines nicht-konvexen Polygons durch die Menge seiner Eckpunkte

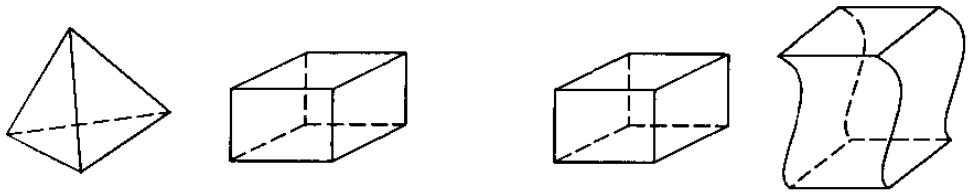


Kriterien zur Beurteilung von Geometrie-Repräsentations-Schemata:

- Mächtigkeit (wie groß ist der Definitionsbereich? Wie genau können komplizierte Objekte modelliert werden?)
- Eindeutigkeit
- Vollständigkeit
- stimmt der Repräsentationsraum *R* mit dem Wertebereich *W* von s überein (oder gibt es "ungültige" symbolische Darstellungen)?
- Genauigkeit (Approximation von Objekten)
- Effizienz (Zeitaufwand für die Darstellung, Speicherplatz)
- Abgeschlossenheit (gegenüber Transformationen, Mengenoperationen)
- wie allgemein sind Operationen, die zur Manipulation von Repräsentationen verwendet werden müssen?
- formale Eleganz der Darstellung (kommt man mit wenigen Operatoren aus, gehorchen diese einfachen Gesetzen?)
- wie kompliziert sind Algorithmen zur Erzeugung und Manipulation von Objekten?

Mathematische Grundlagen:

- Geometrie (Längen, Winkel; *Kongruenzabbildungen*)
- Topologie (Umgebungen, Inzidenzbeziehungen: welches Element hängt mit welchem zusammen? Abbildungen: *Homöomorphismen*)
- Graphentheorie
- Kombinatorik
- algebraische Strukturen
- Maßtheorie



verschiedene Topologie
verschiedene Geometrie

gleiche Topologie
verschiedene Geometrie

Topologische Grundbegriffe:

Sei *A* eine Menge von Punkten, also eine Teilmenge von R^3.

A heißt *beschränkt*, wenn *A* ganz in einer Kugel mit endlichem Radius enthalten ist.

A heißt *Umgebung* eines Punktes *x*, wenn es eine kleine Kugel mit Mittelpunkt *x* gibt, die ganz in *A* enthalten ist. (Die Kugeln bilden eine *Umgebungsbasis* der Standard-Topologie des R^3.)

$x \in A$ heißt *innerer Punkt* von *A* genau dann, wenn es eine Umgebung von *x* gibt, die ganz in *A* liegt (gleichwertig: wenn es eine Kugel mit Mittelpunkt *x* gibt, die ganz in *A* liegt).

x heißt *Randpunkt* von *A*, wenn jede Umgebung von *x* Punkte von *A* und des Komplements von *A* enthält. (Dabei ist auch der Fall $x \notin A$ möglich.)

Der *Rand* von *A*, bezeichnet als δA, ist die Menge aller Randpunkte von *A*.

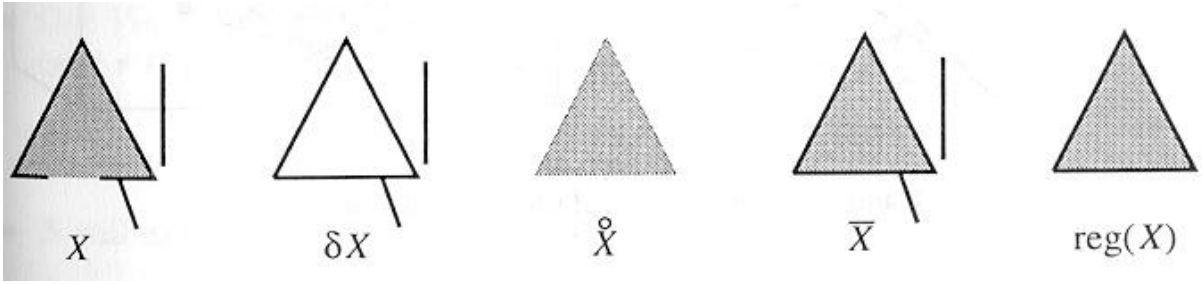
Das *Innere* von *A* ist def. als $\mathring{A} = A - \delta A$ (Mengendifferenz).

A heißt *offen*, wenn $\mathring{A} = A$.

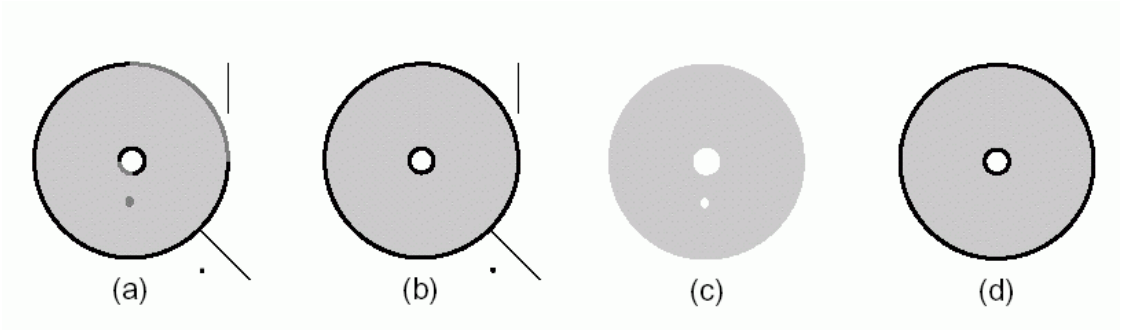
Der *Abschluss* von *A* (auch "abgeschlossene Hülle von *A*") ist def. als $\hat{A} = A \cup \delta A$ (auch: hull(*A*), Hülle(*A*)).

A heißt *abgeschlossen*, wenn $\hat{A} = A$.
Die *Regularisierung* von A ist die Menge reg(*A*) = Hülle($\mathring{A}$).

A heißt *regulär*, wenn reg(*A*) = *A*.



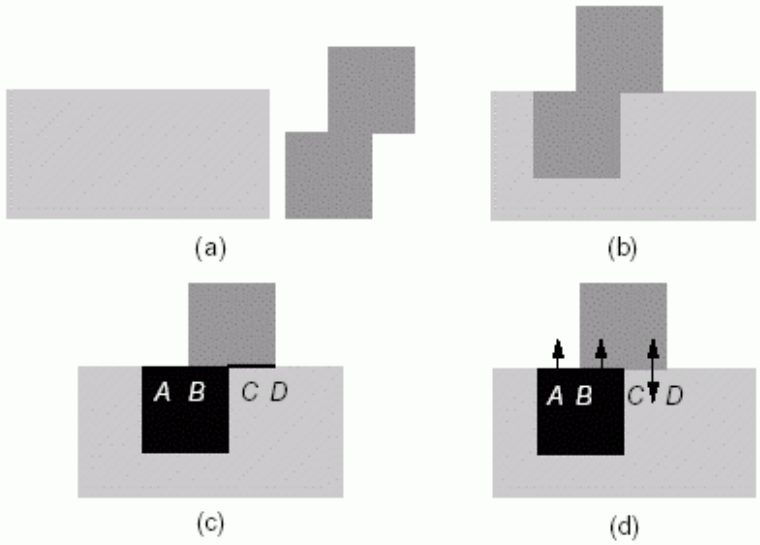
Effekt der Regularisierung: das Objekt wird abgeschlossen, "Löcher" und "hängende Teile" niedriger Dimension werden entfernt:



(a) Ausgangsmenge *A*
(b) Abschluss von *A* (ohne Regularisierung, störende Teile bleiben erhalten)
(c) Inneres von *A*
(d) Regularisierung von *A*.

Regularisierte Mengenoperationen:

Definition: $A\ op^*\ B = \text{Hülle}(\text{Inneres}(A\ op\ B))$



Mögliche Operationen: $\bigcup^*, \bigcap^*, -^*$

Welche 3D-Körper sollen überhaupt modelliert werden?

Vorschlag:
beschränkte, reguläre Teilmengen des R^3.

Probleme:

- endliche Beschreibbarkeit nicht gewährleistet
  ($\rightarrow$ theoretische Informatik)
- Innen und Außen nicht in allen Fällen klar

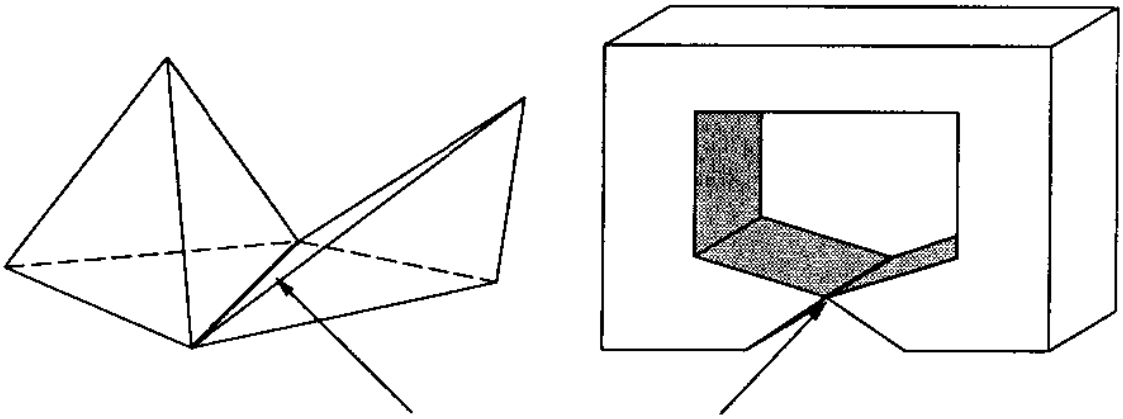
Deshalb def. man zusätzlich:

A heißt *analytisch*, wenn $A = \{ x \mid f(x) \geq 0 \}$ mit einer analytischen (d.h. in jedem Punkt in eine Potenzreihe entwickelbaren) Funktion *f* (Darstellung als "verallg. Halbraum").

A heißt *semianalytisch*, falls *A* als endliche regularisierte Boolesche Kombination (Vereinigung, Schnitt und Mengendifferenz) von analytischen Mengen dargestellt werden kann.

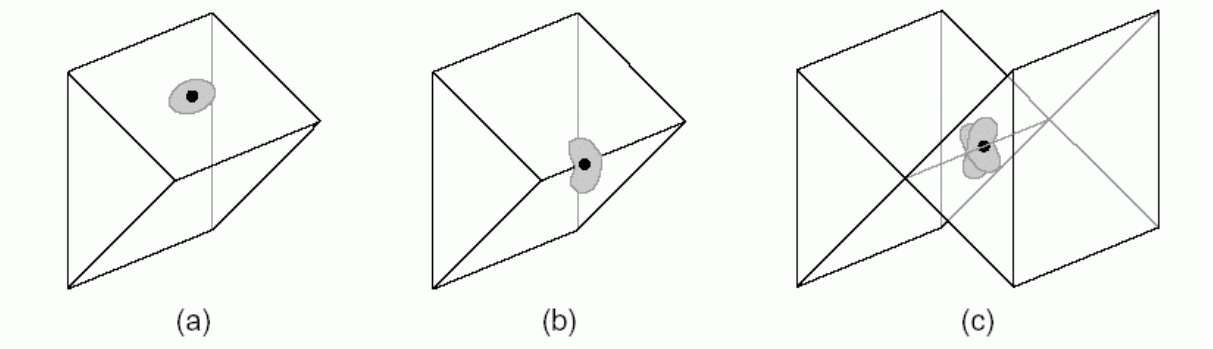
"Starre Körper" werden also math. dargestellt durch beschränkte, reguläre und semi-analytische Teilmengen des R^3.

Immer noch nicht ausgeschlossen:
Kanten mit "nichtmannigfaltiger Umgebung"



Kanten mit nichtmannigfaltiger Umgebung

oft wird daher zusätzlich gefordert: es soll eine *2-Mannigfaltigkeit* vorliegen, d.h. jeder Punkt auf der Oberfläche (dem Rand des Körpers) hat eine Umgebung, die zu einer Kreisscheibe topologisch äquivalent ist.



in (c) liegt keine 2-Mannigfaltigkeit vor, da die gezeigte Umgebung nicht homöomorph auf eine Kreisscheibe abgebildet werden kann (aus Schlechtweg 2001).

Häufig gebrauchte Grundelemente für die Modellierung:

Polyeder

Definition?

dazu folgende Grundlagen:

Graphen (*V*, *E*) (**v**ertices = Knoten oder Ecken, **e**dges = Kanten): sind nicht an Geometrie gebunden, reine Inzidenzstrukturen
(*E* Menge ungeordneter Paare von Knoten)

Ein *geometrischer Graph* ist ein Paar (*V*, *E*), wobei *V* nichtleere, endliche Teilmenge von R^3 und *E* eine Menge von Verbindungsstrecken von Punkten aus *V* ist.
Die Inzidenz von Ecken und Kanten im Sinne der Graphentheorie ist geometrisch erklärt durch "*v* ist Endpunkt der Kante *e*" (im geometrischen Sinne).

Ein *Polygon* ist ein geometrischer Graph (*V*, *E*) mit
$V = \{v_0, ..., v_{n-1}\}$ und $E = \{(v_0v_1), (v_1v_2), ..., (v_{n-2}v_{n-1})\}$.

Ein Polygon heißt

- *eben*, falls alle Kanten in einer Ebene liegen
- *geschlossen*, falls $v_0 = v_{n-1}$
- *einfach*, falls gilt: Der Schnitt jeweils zweier Kanten ist entweder leer oder eine Ecke, und jede Ecke gehört zu höchstens zwei Kanten (d.h.: keine Selbstüberschneidungen des Polygons)

Es gilt der grundlegende
Jordansche Kurvensatz für ebene, geschlossene Polygone:
Jedes geschlossene, einfache Polygon in der Ebene unterteilt die Ebene in zwei disjunkte *Polygongebiete*, ein inneres und ein äußeres Polygongebiet.

Punkte im Inneren können dadurch charakterisiert werden, dass die Anzahl der Schnittpunkte zwischen den Kanten des Polygons und einem Strahl, der von dem Innenpunkt ausgeht, ungerade ist (bereits beim Scangeraden-Algorithmus zum Füllen von Polygonen verwendet).

Ein *Polygonnetz* ist eine Menge *M* von endlich vielen geschlossenen, ebenen und einfachen Polygonen mit folgenden Eigenschaften:

- die inneren Polygongebiete von je 2 Polygonen aus *M* haben keine gemeinsamen Punkte
- je 2 Polygone aus *M* haben entweder keinen Punkt oder eine Ecke oder eine ganze Kante gemeinsam
- jede Kante eines Polygons aus *M* gehört zu höchstens 2 Polygonen
- die Menge aller Kanten, die nur zu einem Polygon aus *M* gehören, ist entweder leer (*M* heißt dann "geschlossen") oder bildet selbst ein einziges, geschlossenes, einfaches Polygon.

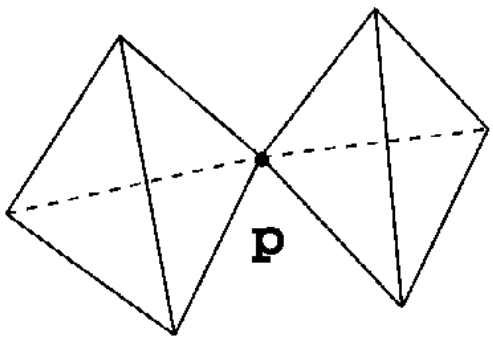
Polygonnetze spielen eine wichtige Rolle in der Modellierung (Beisp. FEM, Kontrollpunkt-Netze von Bézier- und B-Spline-Flächen, *elevation grids*)

Def. *Polyeder*:

Ein Polygonnetz *M* heißt Polyeder, wenn gilt:

- *M* ist geschlossen (d.h. jede Kante gehört zu genau 2 Polygonen)
- *M* ist zusammenhängend
- jede Ecke gehört zu einer endlichen, zyklisch geordneten Menge von Polygonen, in der aufeinanderfolgende Polygone jeweils eine zur Ecke gehörende Kante gemeinsam haben

Beispiel, wo die dritte Bedingung verletzt ist:



Die inneren Polygongebiete von *M* heißen auch *Facetten* oder Seitenflächen des Polyeders.
Der Abschluss der Vereinigung aller Facetten heißt die *Oberfläche* (surface) des Polyeders.

Die Polyeder sind die 3D-Verallgemeinerungen der ebenen, einfachen, geschlossenen Polygone.
Insbes. gilt das 3D-Analogon des Jordanschen Kurvensatzes:

Jedes Polyeder teilt den Raum in zwei disjunkte Bereiche, das Innere und das Äußere.

Das Innere kann wieder dadurch charakterisiert werden, dass die Anzahl der Schnitte eines Strahls, der von einem Innenpunkt ausgeht, mit der Oberfläche ungerade ist.

Das Innere eines Polyeders ist also vollständig durch seine Oberfläche definiert
$\rightarrow$ Grundlage einer wichtigen Repräsentationsform:
boundary representation

Höherdimensionale Analoga zu Polyedern: *Polytope*

topologisch "einfachste" nichttriviale Polytope: *Simplices*

0-Simplex: Punkt
1-Simplex: Strecke
2-Simplex: Dreieck
3-Simplex: Tetraeder
...

math. Def.: *m*-Simplex = konvexe Kombination von *m*+1 affin unabhängigen Punkten im R^m.

simpliziale Zerlegung (a.: Triangulierung) einer Punktmenge *P*: endliche Menge von Simplices mit den Eigenschaften

- jeder Punkt gehört zu mindestens einem der Simplices
- der Schnitt zweier Simplices ist entweder leer oder eine gemeinsame *k*-dim. Facette (ein Simplex niedrigerer Dimension)

- die Menge aller Facetten auf dem Rand der Triangulierung bildet ein konvexes Polytop (Polygon im R^2, Polyeder im R^3).

Es gelten die folgenden Formeln:

Euler-Formel für 2D-Triangulierungen:

Es sei *T* eine Triangulierung im R^2 und *n* die Gesamtzahl der der Ecken von *T*, *b* die Anzahl der Ecken auf der konvexen Hülle von *T*, *e* die Anzahl der Kanten von *T* und *t* die Anzahl der Dreiecke von *T*. Dann gilt:

$$e = 3(n–1) – b,$$
$$t = 2(n–1) – b.$$

Insbesondere folgt: Jede Triangulierung einer 2D-Punktmenge besitzt dieselbe Anzahl von Kanten und dieselbe Anzahl von Dreiecken.

Euler-Formel für 3D-Triangulierungen:

Es sei *T* eine Triangulierung im R^3 und *n* die Gesamtzahl der Eckenpunkte von *T*, *f* die Anzahl der Dreiecke, *e* die Anzahl der Kanten und *t* die Anzahl der Tetraeder von *T*. Dann gilt:

$$n – e + f – t = 1.$$

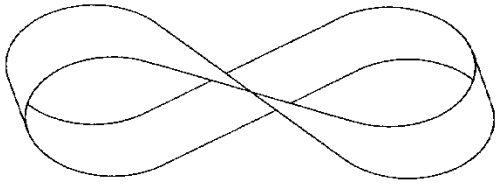
Beachte:
Im Dreidimensionalen ist die Anzahl der Tetraeder einer Triangulierung eines gegebenen Polyeders *nicht* eindeutig bestimmt, auch wenn man innere Punkte nicht zulässt. Beispielsweise lässt sich ein Würfel in 5 oder in 6 Tetraeder zerlegen. (Übungsaufgabe: Wie sehen diese Zerlegungen aus?)

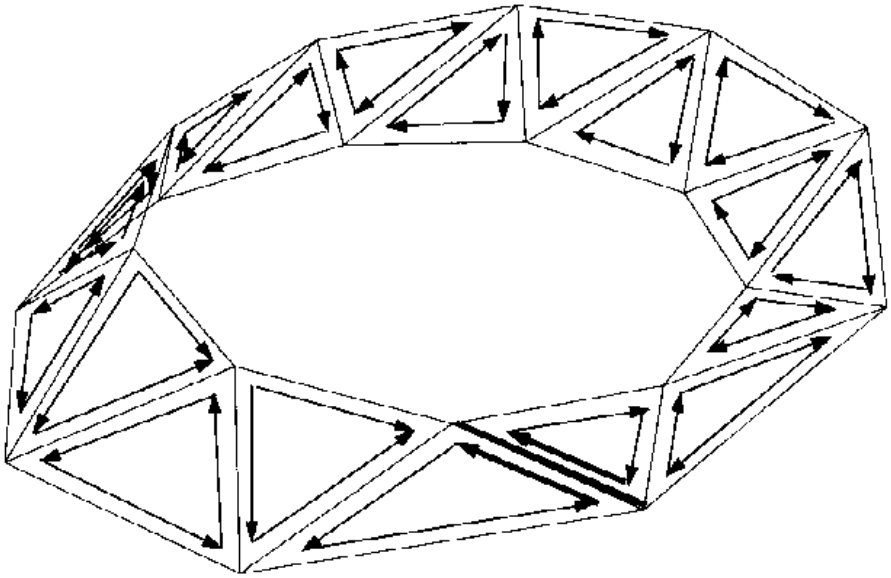
wichtige topologische Eigenschaft:
Orientierbarkeit

Def. eines Umlaufsinns auf jeder Facette eines Polygonnetzes möglich (sukzessive Kanten desselben Polygons in gleiche Richtung orientiert) –
zwei Facetten heißen *gleich orientiert*, wenn die durch die beiden Orientierungen auf der gemeinsamen Kante induzierten Pfeile entgegengesetzte Richtungen haben.

Ein Polygonnetz heißt orientierbar, wenn die Facetten (Polygone) so mit Orientierungen versehen werden können, dass je 2 längs einer Kante benachbarte Facetten gleich orientiert sind.

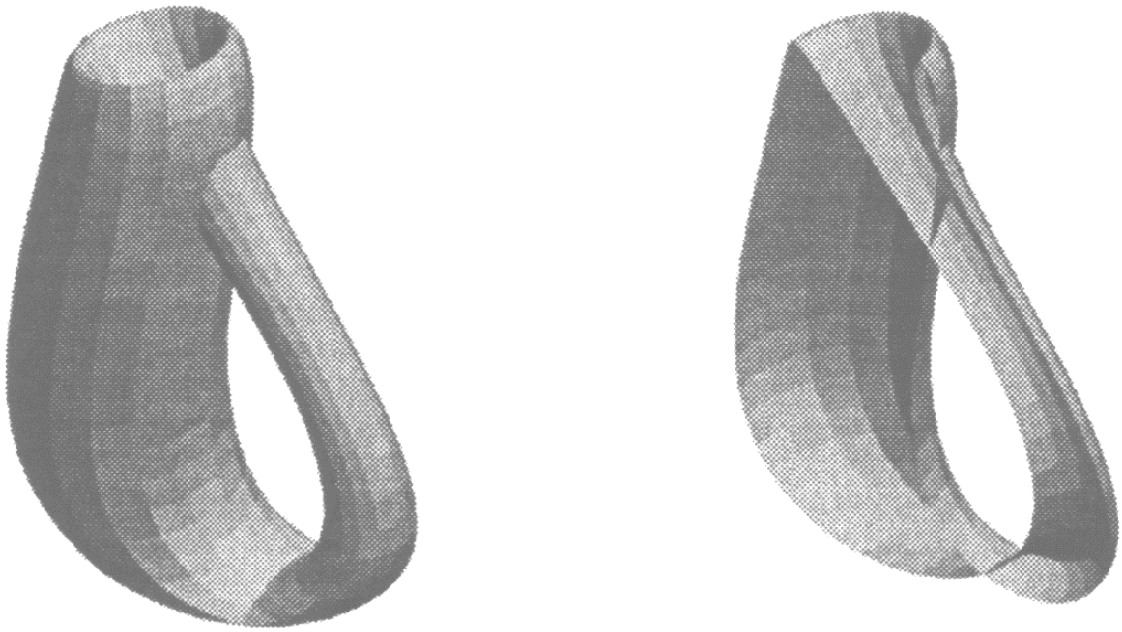
Beispiel einer nichtorientierbaren Fläche:
das *Möbiusband*





nichtorientierbare *geschlossene* Flächen sind im R^3 nur mit Selbstdurchdringung realisierbar.

Beispiel *Kleinsche Flasche* (auch: Kleinscher Schlauch)
(benannt nach Felix Klein):

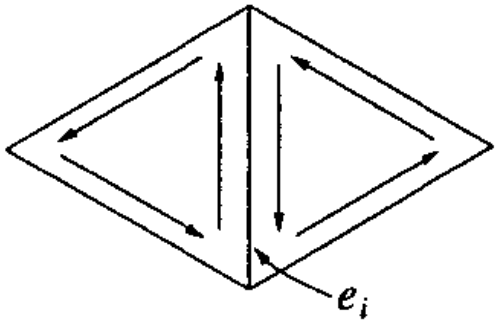


links: Außenansicht eines Papiermodells der Kleinschen Flasche, rechts: Längsschnitt durch das Modell
(aus Bungartz et al. 1996)

Orientierbarkeit sieht man einer symbolischen Repräsentation eines Polygonnetzes nicht auf den ersten Blick an.

<u>Möbius-Algorithmus zum Test auf Orientierbarkeit:</u>

- Initialisiere *E* mit der Menge aller Kanten des Polygonnetzes.
- Orientiere für jedes Polygon die Polygonkanten gegen den Uhrzeigersinn.
- Wenn für eine Kante $e \in E$ gilt: die beiden Kanten-orientierungen von *e* sind gegenläufig, dann wird *e* aus *E* entfernt. Wiederhole diesen Schritt, solange es möglich ist.
- Ist *E* am Schluss leer, dann ist das gegebene Polygonnetz orientierbar, sonst nicht.



Kantenorientierung beim Möbius-Verfahren:
Die mittlere Kante wird aus der aktuellen Kantenmenge entfernt.

weitere top. Invariante von Flächen (2-Mannigfaltigkeiten) neben der Orientierbarkeit:
die *Eulersche Charakteristik* χ .
Sei *G* ein beliebiger, in eine 2-Mannigfaltigkeit *M* eingebetteter Graph, und seien *v*, *e*, *f* die Anzahlen der Knoten, Kanten bzw. Facetten, die *G* auf *M* induziert. Dann gilt:
Die Summe $\chi = v - e + f$ ist konstant und unabhängig von der Wahl von *G*.

Spezialfälle:

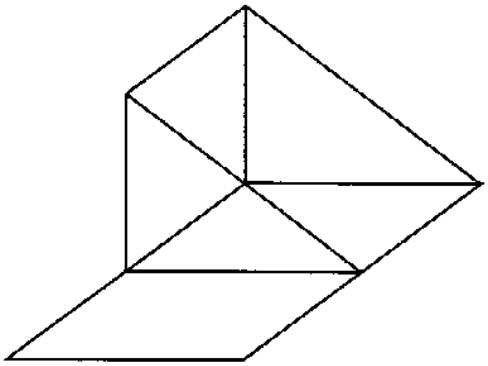
M homöomorph zur Kugel $\Rightarrow \chi = 2$
das gilt insbesondere für alle Polyeder ohne Löcher (Durchbohrungen); man erhält den *Eulerschen Polyedersatz*:

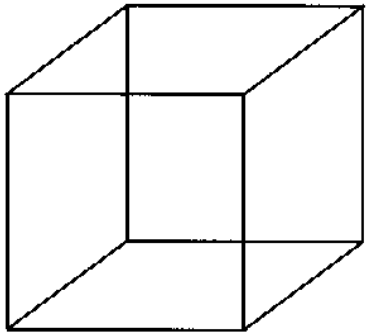
$$v - e + f = 2$$

M homöomorph zum Torus $\Rightarrow \chi = 0$
das gilt insbesondere auch für Zellzerlegungen der Ebene (in Polygone oder speziell Dreiecke), die man aus Endlichkeitsgründen periodisch in beiden Richtungen schließt.
$v - e + f = 0$

M homöomorph zu einer Kugel mit *k* Henkeln $\Rightarrow \chi = 2 - 2k$;
M homöomorph zur Kleinschen Flasche $\Rightarrow \chi = 0$.

Beachte: Nicht jede Mannigfaltigkeit, für die der Eulersche Polyedersatz gilt, ist ein Polyeder (oder auch nur ein Polygonnetz gemäß unserer Def.):





in beiden Fällen gilt $v = 8$, $e = 12$, $f = 6$, aber das linke Objekt ist kein zulässiges Polygonnetz (eine Kante hat mehr als 2 inzidente Polygone).

Verallgemeinerung der Eulerschen Charakteristik auf Polyeder, die auch Hohlräume und verallgemeinerte Polygone mit Löchern enthalten dürfen:

Euler-Poincaré-Formel
Sei r die Anzahl der Löcher in den Facetten, g die Anzahl der Löcher des verallg. Polyeders, s die Anzahl der Zusammenhangskomponenten der Oberfläche (einschließlich der Hohlräume). Dann gilt:

$$v - e + f - r - 2s + 2g = 0.$$

Datenstrukturen für Polygonnetze

1. Polygonorientierte Datenstruktur

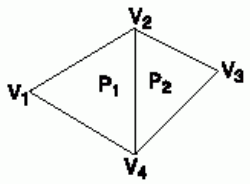
- Jedes Polygon wird durch eine Liste von Koordinaten der Knoten (Eckpunkte, Vertices) beschrieben:

$$P = ((x_1, y_1, z_1), (x_2, y_2, z_2), \ldots, (x_n, y_n, z_n))$$

- Kanten werden zwischen aufeinanderfolgenden Knoten erzeugt
- Knoten werden mehrfach gespeichert
- keine explizite Kennzeichnung geteilter Ecken und Kanten
- Probleme beim Rendering (Kanten werden doppelt gezeichnet)

2. Knotenorientierte Datenstruktur

Beseitigung von Redundanz: Knoten werden nur jeweils einmal gespeichert und dann den einzelnen Facetten durch Zeiger zugeordnet. Auflistung der Knoten pro Facette gewöhnlich mit fester Orientierung (z.B. im Uhrzeigersinn) – nützlich für viele Algorithmen

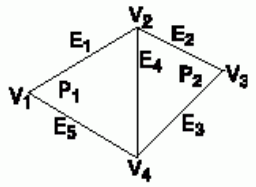


$$\begin{aligned} V &= (V_1, V_2, V_3, V_4) = ((x_1, y_1, z_1), \ldots, (x_4, y_4, z_4)) \\ P_1 &= (1, 2, 4) \\ P_2 &= (4, 2, 3) \end{aligned}$$

- speicherplatzsparend, da Knoten nur einmal gespeichert
- einfache Transformationen
- geteilte Kanten doppelt gespeichert

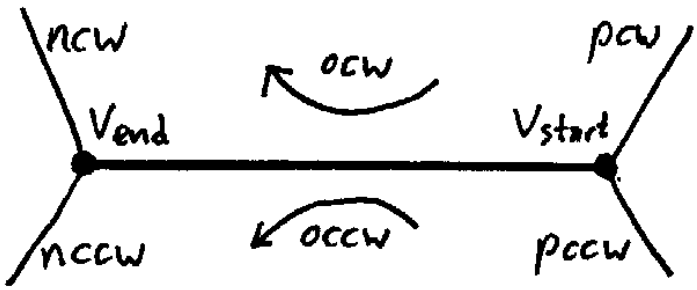
3. Kantenorientierte Datenstruktur

Facetten als Zyklen von Kanten
Kanten als Listen von Knoten und 1 oder 2 angrenzenden Facetten; für jede Kante ist Orientierung durch Reihenfolge der Endpunkte festgelegt



$$
\begin{aligned}
V &= (V_1, V_2, V_3, V_4) = ((x_1, y_1, z_1), \ldots, (x_4, y_4, z_4)) \\
E_1 &= (V_1, V_1, P_1, \lambda) \\
E_2 &= (V_2, V_3, P_2, \lambda) \\
E_3 &= (V_3, V_4, P_2, \lambda) \\
E_4 &= (V_4, V_2, P_1, P_2) \\
E_5 &= (V_4, V_1, P_1, \lambda) \\
P_1 &= (E_1, E_4, E_5) \\
P_2 &= (E_2, E_3, E_4)
\end{aligned}
$$

Erweiterung:
es werden zusätzlich Nachbarschaftsbeziehungen unter den Kanten gespeichert
(4 Nachbarkanten: ncw = next clockwise, pcw = previous clockwise, nccw = next counterclockwise, pccw = previous counterclockwise)
für die Facetten braucht man dann nur einen Zeiger auf eine Startkante und ein Bit für die Orientierung dieser Startkante.



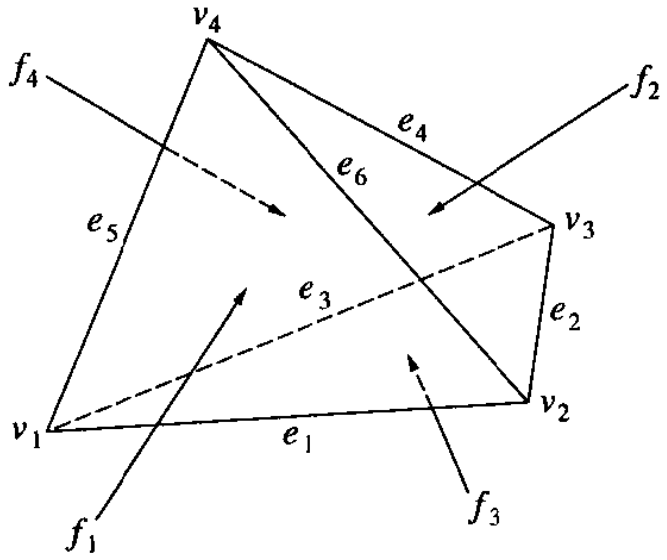
(ocw = orientation clockwise, occw = orientation counterclockwise)

→

Winged-Edge-Repräsentation

- Datenstruktur zur Speicherung von b-reps
- Kante zeigt auf:
  - angrenzende Knoten (V1, V2)
  - angrenzende Flächen (F1, F2)
  - angrenzende Kanten (E2, E3, E4, E5)
- nur Objekte ohne Löcher werden beschrieben
- effizienter Zugriff auf Nachbarschaft (Flächen, Kanten, Knoten)

Beispiel: Tetraeder



Relationen und Listen der zugehörigen Winged-Edge-Struktur:

v:

e_1	v_1	v_2
e_2	v_2	v_3
e_3	v_1	v_3
e_4	v_3	v_4
e_5	v_1	v_4
e_6	v_2	v_4

ef:

e_1	f_1	f_3
e_2	f_2	f_3
e_3	f_3	f_4
e_4	f_2	f_4
e_5	f_1	f_4
e_6	f_1	f_2

ee':

e_1	e_2	e_3	e_5	e_6
e_2	e_1	e_3	e_4	e_6
e_3	e_1	e_2	e_4	e_5
e_4	e_2	e_3	e_5	e_6
e_5	e_1	e_3	e_4	e_6
e_6	e_1	e_2	e_4	e_5

v-Liste:

v_1	e_1
v_2	e_2
v_3	e_3
v_4	e_4

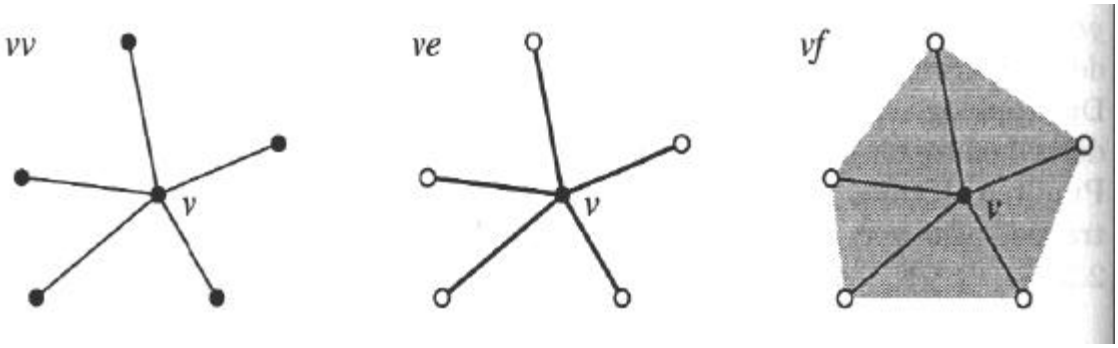
f-Liste:

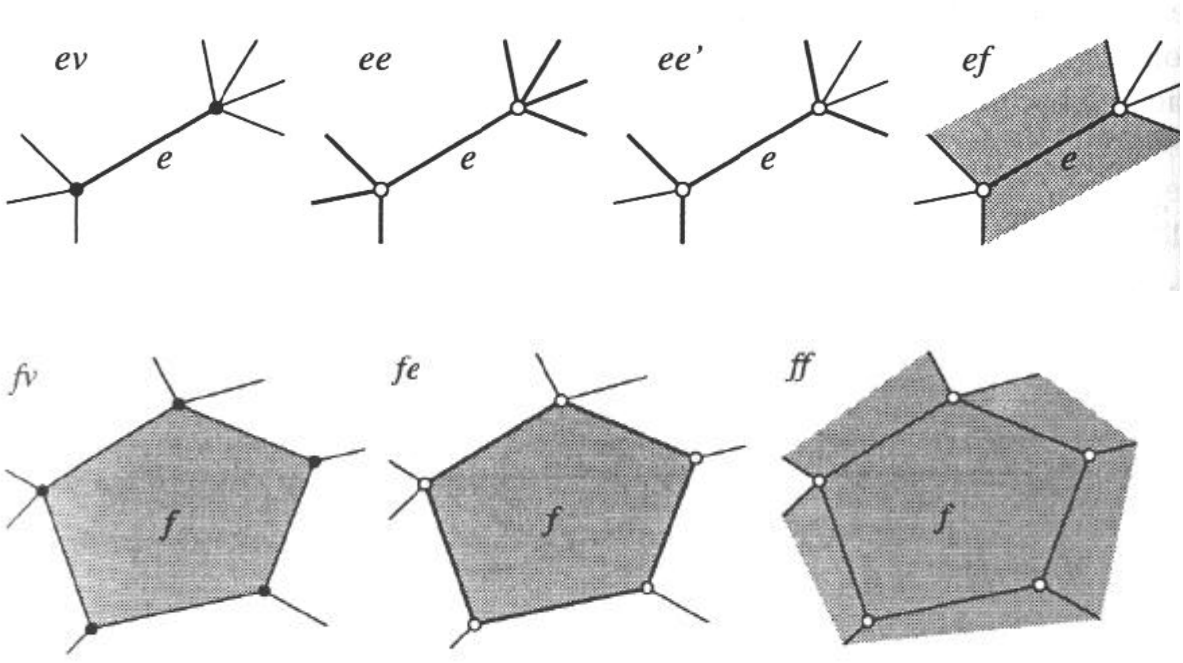
f_1	e_5
f_2	e_2
f_3	e_2
f_4	e_4

Allgemeiner Ansatz: der *vef*-Graph

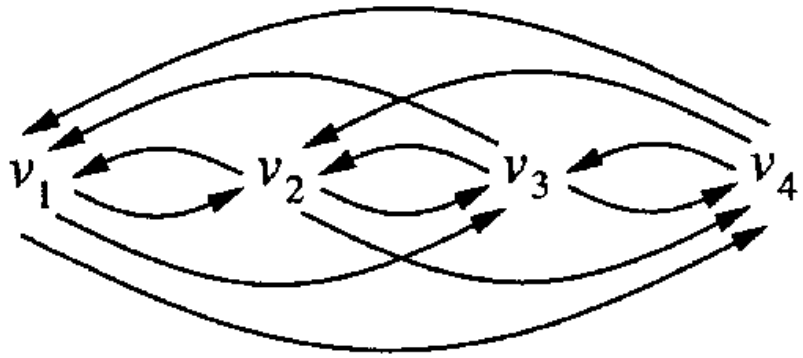
Knoten dieses Graphen:
Mengen der Ecken, Kanten und Facetten
Kanten des Graphen: geeigente Adjazenzrelation
10 Relationen kommen in Frage, davon wird eine im jeweiligen Datenmodell eine Auswahl getroffen:

z.B. *vv*: Punkte sind benachbart (haben gemeinsame Kante)
ve: Punkt begrenzt Kante
vf: Punkt ist Eckpunkt von Facette
usw. bis *ff*: Flächen sind benachbart (haben gemeinsame Kante)
neben ee betrachtet man noch *ee'*: Kanten sind benachbart und begrenzen dieselbe Facette (Teilmenge von *ee*)

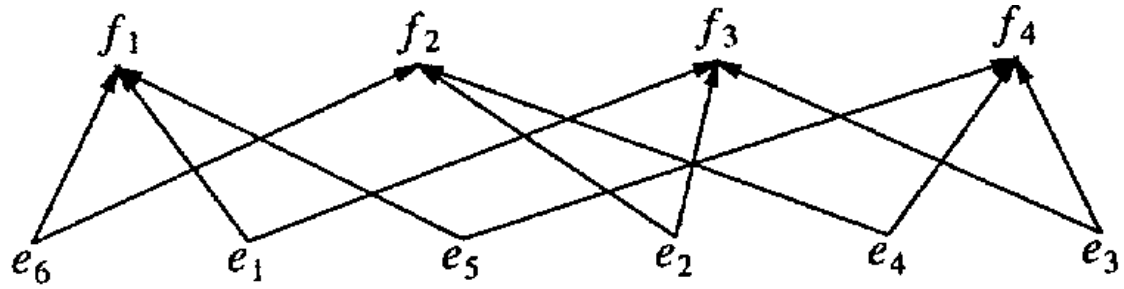




Beispiel: *vef*-Graph für die *vv*-Relation im obigen Tetraeder:



vef-Graph für die *ef*-Relation:



- der Speicheraufwand für die einzelnen Relationen ist unterschiedlich
- je nach Anwendungszweck kann es sinnvoll sein, die eine oder die andere abzuspeichern
- man braucht nicht alle explizit zu speichern: Beziehungen zwischen den Relationen nutzen!

$vf = ve \times ef$
$fv = fe \times ev$
$vv = ve \times ev - id$
$ff = fe \times ef - id$
$ee' = ev \times ve \cap ef \times fe - id$

darin ist $\times$ die von den relationalen Datenbanken bekannte *join*-Operation und *id* die identische Relation auf der entsprechenden Menge.

Fragen:
- welche Kriterien muss ein *vef*-Graph erfüllen, um die top. Struktur eines starren Körpers zu repräsentieren?
  – notwendige Voraussetzung: Euler-Poincaré-Formel
- wie können solche Graphen künstlich konstruiert werden?

Erzeugendensystem von Operatoren für verallgemeinerte Polygonnetze:

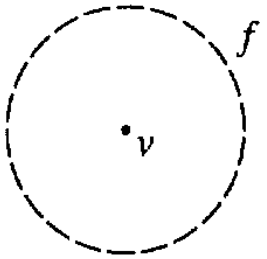
Euler-Operatoren

- erzeugen verallg. Polyeder in konsistenter Weise
- fügen Knoten, Kanten und Facetten hinzu oder löschen sie
- können auch die Anzahl der Löcher eines Polyeders verändern, indem Henkel angefügt werden

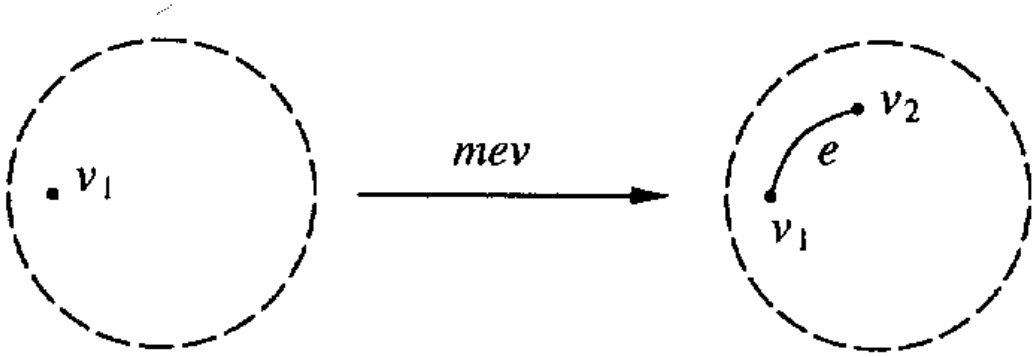
Bezeichnungsweise: Zeichenfolge aus den Zeichen *m*X(*k*|*s*)Y, wobei X und Y für *v*, *e*, *f*, *l* (loop = Polygonrand), *h* (hole = Loch), *r* (Ring), *s* (shell = Oberflächenkomponente) stehen und *m* = make, *k* = kill, *s* = split bedeuten.
Beispiel: *kfmrh* = kill face make ring hole

- die Euler-Poincaré-Formel bleibt bei jeder Operatoranwendung erfüllt
- alle Polyeder können durch eine Folge von Operator-anwendungen erzeugt werden

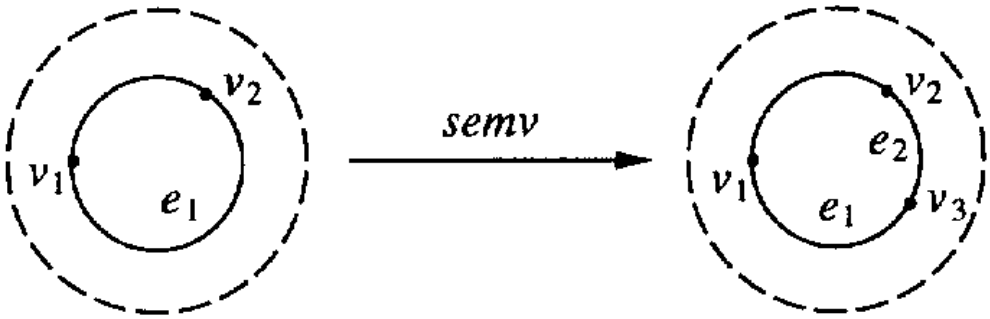
Beispiele:
Startoperator *mvsf* = make vertex shell face
erzeugt einen initialen "Gummiball" mit einer Ecke und einer Facette:



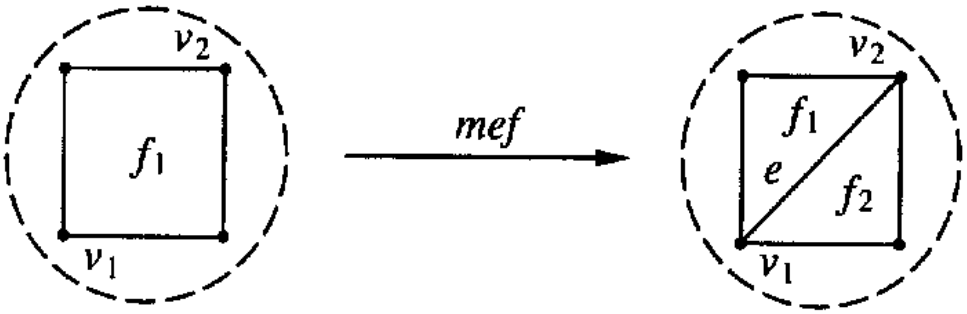
mev = make edge and vertex
fügt eine neue Kante ein, die einen schon existierenden Eckpunkt mit einem neuen Eckpunkt verbindet:



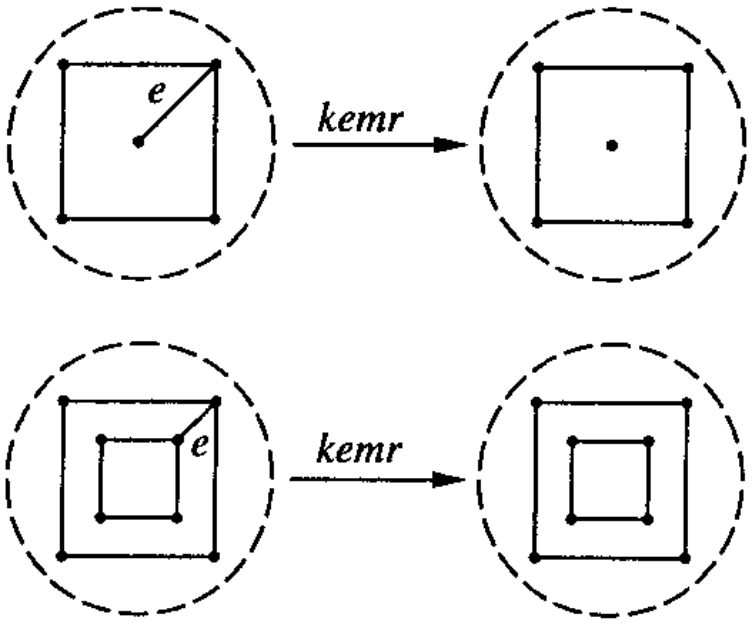
semv = split edge make vertex
teilt eine existierende Kante durch einen neuen Eckpunkt



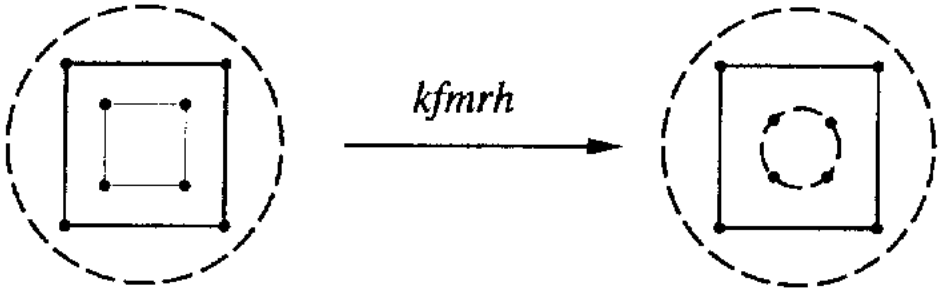
mef = make edge and face



kemr = kill edge make ring



krmrh = kill face make ring and hole
es wird eine Durchbohrung erzeugt



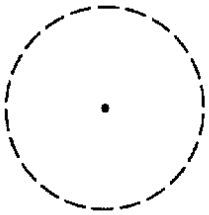
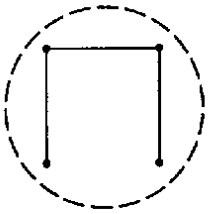
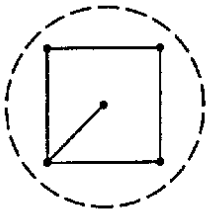
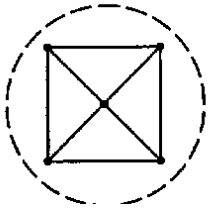
in der Literatur verschiedene Systeme von Euler-Operatoren
man braucht mindestens 5

Beispielsystem (mit Wirkung auf die Parameter *v*, *e*, *f* usw.):

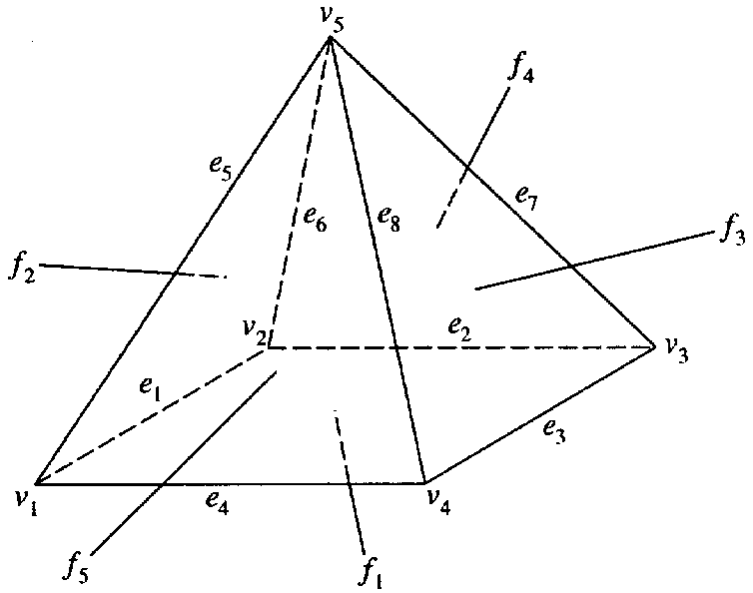
Operator	Basisvektor					
	v	e	f	g	$r = l - f$	s
mev	1	1	0	0	0	0
mef	0	1	1	0	0	0
mvfs	1	0	1	0	0	1
kemr	0	-1	0	0	1	0
kfmrh	0	0	-1	1	1	0
kev	-1	-1	0	0	0	0
kef	0	-1	-1	0	0	0
kvfs	-1	0	-1	0	0	-1
mekr	0	1	0	0	-1	0
mfkrh	0	0	1	-1	-1	0

m	=	make	=	erzeugen
k	=	kill	=	löschen
v	=	vertex	=	Knoten
e	=	edge	=	Kante
f	=	face	=	Facette
s	=	shell	=	Schale
l	=	loop	=	Schleife
g ≡ h	=	genus, hole	=	Geschlecht
r	=	Anzahl innerer Schleifen		

Beispiel: Konstruktion einer Pyramide mit Euler-Operatoren

$mvsf\,(f_1, v_1)$		$n_V = 1$ $n_E = 0$ $n_F = 1$ $n_S = 1$ $n_H = n_R = 0$
$mev\,(v_1, v_2, e_1)$ $mev\,(v_2, v_3, e_2)$ $mev\,(v_3, v_4, e_3)$		$n_V = 4$ $n_E = 3$ $n_F = 1$ $n_S = 1$ $n_H = n_R = 0$
$mef\,(v_1, v_4, f_1, f_2, e_4)$ $mev\,(v_1, v_5, e_5)$		$n_V = 5$ $n_E = 5$ $n_F = 2$ $n_S = 1$ $n_H = n_R = 0$
$mef\,(v_2, v_5, f_2, f_3, e_6)$ $mef\,(v_3, v_5, f_2, f_4, e_7)$ $mef\,(v_4, v_5, f_2, f_5, e_8)$		$n_V = 5$ $n_E = 8$ $n_F = 5$ $n_S = 1$ $n_H = n_R = 0$

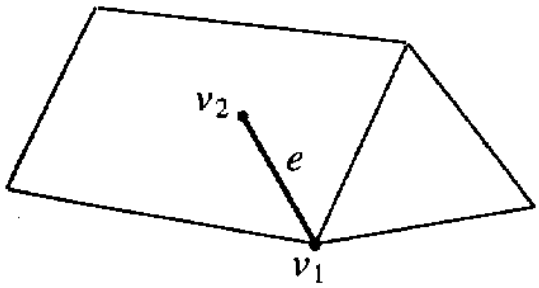
Bezeichnungsweisen der Ecken, Flächen und Kanten:

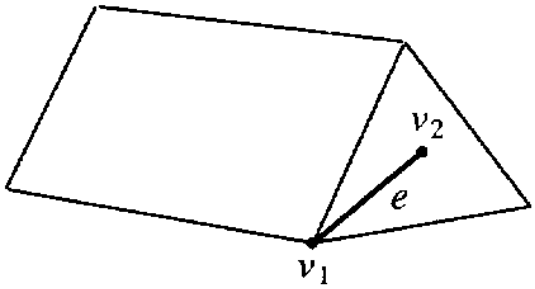


Nachteile:

- es treten auch "nicht sinnvolle Objekte" als Zwischenstufen bei der Konstruktion auf

- die Semantik der Operatoren ist ohne Zusatzinformation mehrdeutig (z.B. kann *mev* eine Kante an unterschiedliche Ecken anfügen, mit top. nichtäquivalentem Ergebnis:)





- fehlende geometrische Informationen

– werden bei Realisierung von Euler-Operatoren meist aber mit aufgenommen und in entspr. Datenstrukturen übertragen

Vorteile:

- sehr allgemeiner Ansatz, konstruktiv (alle Polyeder können erzeugt werden)
- durch eine endliche Folge von Euler-Operatoren können keine ungültigen 2D-Mannigfaltigkeitsmodelle erzeugt werden

bisher nur betrachtet:
"Boundary representation"

Übersicht über die wichtigsten Repräsentationsschemata für 3D-Modelle: