



**Formalismus
zur Repräsentation von
Geometrie- und
Technologieinformationen
als Teil eines
Wissensbasierten Produktmodells**

**Ansgar Bernardi, Christoph Klauck,
Ralf Legleitner**

Dezember 1990

**Deutsches Forschungszentrum für Künstliche Intelligenz
GmbH**

Postfach 20 80
D-6750 Kaiserslautern, FRG
Tel.: (+49 631) 205-3211/13
Fax: (+49 631) 205-3210

Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11, FRG
Tel.: (+49 681) 302-5252
Fax: (+49 681) 302-5341

Deutsches Forschungszentrum für Künstliche Intelligenz

The German Research Center for Artificial Intelligence (Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI) with sites in Kaiserslautern und Saarbrücken is a non-profit organization which was founded in 1988 by the shareholder companies ADV/Orga, AEG, IBM, Insiders, Fraunhofer Gesellschaft, GMD, Krupp-Atlas, Mannesmann-Kienzle, Nixdorf, Philips and Siemens. Research projects conducted at the DFKI are funded by the German Ministry for Research and Technology, by the shareholder companies, or by other industrial contracts.

The DFKI conducts application-oriented basic research in the field of artificial intelligence and other related subfields of computer science. The overall goal is to construct *systems with technical knowledge and common sense* which - by using AI methods - implement a problem solution for a selected application area. Currently, there are the following research areas at the DFKI:

- Intelligent Engineering Systems
- Intelligent User Interfaces
- Intelligent Communication Networks
- Intelligent Cooperative Systems.

The DFKI strives at making its research results available to the scientific community. There exist many contacts to domestic and foreign research institutions, both in academy and industry. The DFKI hosts technology transfer workshops for shareholders and other interested groups in order to inform about the current state of research.

From its beginning, the DFKI has provided an attractive working environment for AI researchers from Germany and from all over the world. The goal is to have a staff of about 100 researchers at the end of the building-up phase.

Prof. Dr. Gerhard Barth
Director

Formalismus zur Repräsentation von Geometrie- und Technologieinformationen als Teil eines Wissensbasierten Produktmodells

Ansgar Bernardi, Christoph Klauck, Ralf Legleitner

DFKI-D-90-05

© Deutsches Forschungszentrum für Künstliche Intelligenz 1990

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Deutsches Forschungszentrum für Künstliche Intelligenz, Kaiserslautern, Federal Republic of Germany; an acknowledgement of the authors and individual contributors to the work; all applicable portions of this copyright notice. Copying, reproducing, or republishing for any other purpose shall require a licence with payment of fee to Deutsches Forschungszentrum für Künstliche Intelligenz.

**Formalismus zur Repräsentation
von Geometrie- und Technologieinformationen
als Teil eines Wissensbasierten Produktmodells**

**Ansgar Bernardi
Christoph Klauck
Ralf Legleitner**

Dezember 1990

**E-Mail:
bernardi@dfki.uni-kl.de
klauck@dfki.uni-kl.de
legleitner@dfki.uni-kl.de**

Inhalt:

1. Einführung	8
1.1. Anforderungen an die Repräsentation.....	8
2. Die gewählte Repräsentation	11
2.1. Prinzipien.....	11
2.2. Betrachtete Alternativen	12
2.3. Einschränkungen	13
3. Beschreibung der Primitive.....	14
3.1. Syntax.....	14
3.2. Allgemeine Werkstückinformationen	15
3.2.1. Globale Werkstückinformationen	15
3.2.2. Geometrie- und Technologiemarkros	16
3.3. Rotationsflächen oder Teile von Rotationsflächen	17
3.3.1. Zylindermantelfläche	17
3.3.2. Zylindermantelabschnittfläche	18
3.3.3. Kegelmantelfläche	19
3.3.4. Torusabschnittfläche	20
3.4. Ebene Flächen	22
3.4.1. Dreiecksfläche	22
3.4.2. Parallelogrammfläche.....	23
3.4.3. Rechteckfläche.....	24
3.4.4. Kreisfläche	25
3.4.5. Halbkreisfläche.....	26
3.4.6. Kreisabschnittfläche.....	27
3.4.7. Kreissegmentfläche	28
3.5. Gelochte Flächen.....	29
3.5.1. Gelochte Fläche	31
3.5.2. Kreisringfläche	32
3.6. Flächenelemente für rotationssymmetrische Werkstücke.....	33
3.6.1. Zylindermantelfläche	34
3.6.2. Kreisfläche	35
3.6.3. Kegelmantelfläche	36
3.6.4. Torusabschnittfläche	37

3.5.2. Kreisringfläche	39
3.7. Beschreibung der technologischen Grundelemente.....	40
3.7.1. Kante (Edge)	40
3.7.2. Gewinde (Thread).....	43
3.7.3. Maßtoleranz (Size Standard).....	44
3.7.4. ISO-Passung nach DIN 7154 bzw. 7155 (ISO-Fit)	46
3.7.5. Allgmeintoleranz DIN 7168 (General Tolerance)	47
3.7.6. Formtoleranz DIN 7184 (Shape Tolerance).....	48
3.7.7. Lagetoleranzen (Positional Tolerance).....	50
3.7.8. Oberflächenangaben (Surface Finish).....	53
3.7.9. Härteangaben (Hardness).....	54
4. Topologieinformationen.....	55
4.1. Nachbarschaftsbeziehung (Neighbourhood Relation).....	57
5. Ausblick	59
6. Literatur	60
7. Beispiele einiger Repräsentationen	61

1. Einführung

Im Projektteil R von ARC-TEC ist unter anderem das Integrierte Wissensbasierte Produktmodell (IWP) zu erstellen. Mit Hilfe der unter diesem Begriff zusammengefaßten Formalismen soll es möglich sein, ein Werkstück so vollständig zu beschreiben, daß auf dieser Basis die im Rahmen des Gesamtprojekts auftretenden Aufgaben (Konfiguration, Arbeitsplanung, Qualitätssicherung) gelöst werden können.

Das IWP umfaßt neben Elementen zur Beschreibung des Fertigungsumfeldes (Werkzeugmaschinen, Werkzeuge, Spannmittel usw.) und der speziellen Elemente der Aufgabe (etwa Arbeitspläne ...) eine vollständige Repräsentation des Werkstücks. Hierzu werden Geometrie, Funktionselemente, technologische Information, Bemaßung, arbeitstechnische Bereiche ("Features"), aber auch der intendierte Einsatz und die Historie des Produkts usw. abgebildet. (vgl. /3/, /4/, /6/, /8/, /11/)

Der hier vorgestellte Entwurf befaßt sich ausschließlich mit der Repräsentation des Werkstücks. Dabei spielt die Darstellung der Geometrie eine zentrale Rolle; die meisten zusätzlich zu verwaltenden Informationen können als Ergänzungen zur Geometrie aufgefaßt und somit auch in Anlehnung an eine Geometriedarstellung repräsentiert werden. Der hier vorgestellte Ansatz trägt diesen Überlegungen Rechnung, indem ein Formalismus zur Repräsentation der Geometrie eines Werkstücks entworfen und um Elemente zur Darstellung von Funktionselementen, technologischen Informationen usw. erweitert wird.

1.1. Anforderungen an die Repräsentation

Im Hinblick auf die zugrundeliegende Aufgabenstellung sind folgende Anforderungen an den Repräsentationsformalismus zu stellen:

- Universalität: Es müssen zumindest alle ins Auge gefaßten Produkte dargestellt werden können; die Repräsentation soll jedoch auch so gestaltet sein, daß sie jederzeit erweitert werden kann.
- Spezialisierung: Das Projekt ARC-TEC bearbeitet zunächst eingeschränkte und damit einfache Beispiele, ohne jedoch die für eine Verallgemeinerung nötigen Erweiterungen außer acht zu lassen. Der Formalismus soll für die eingeschränkten Beispiele möglichst einfache Konstrukte zur Verfügung stellen, ohne damit prinzipielle Einschränkungen einzuführen.

- Verwaltung von Attributen: Neben der Geometrie eines Werkstücks sind Funktionselemente, Bemaßung und technologische Informationen darzustellen. Diese Informationen stehen alle in enger Beziehung zur Geometrie des Werkstücks. Es ist daher sinnvoll, die Geometriedarstellung des Werkstücks durch Attribute zu ergänzen, die diese Informationen repräsentieren.
- Symbolische Repräsentation: Der Einsatz von KI-Techniken im Rahmen von ARCTEC bedeutet, daß Wissen in symbolischer Form repräsentiert wird und daß auf der Basis symbolischer Repräsentationen Deduktionen durchgeführt werden. Dementsprechend muß der Repräsentationsformalismus für eine symbolische Verarbeitung geeignete Begriffe einführen und die Bildung höherer Begriffe erlauben (vgl. /6/). In diesem Zusammenhang muß die hier vorgestellte Repräsentation die Erkennung von Features des Werkstückes, zumindest der Application Features (arbeitstechnische Bereiche oder manufacturing form features) stark unterstützen.

Redundanzfreiheit: Im Hinblick auf die spätere symbolische Verarbeitung ist es wünschenswert, die Repräsentation so redundanzfrei wie möglich zu halten. Für ein gegebenes Werkstück soll in der Regel genau eine kürzeste Repräsentation existieren. Die Redundanzfreiheit sollte nicht implizieren, daß Elemente der Repräsentation eine Menge von Elementen nicht subsumiert. Elemente, die oft benötigt werden, wie etwa eine Kreisringfläche, müssen auch durch ein eigenes Element repräsentiert werden. Die Auswahl solcher Elemente bezieht sich nur auf elementare Flächen (kontext -unabhängige Flächen), die meist durch komplexere Operationen, wie etwa schneiden, erzeugt werden; also bezieht sich dies insbesondere nicht auf Features (Flächenaggregationen), die immer in Abhängigkeit von mehreren Flächen definiert sind. Alle Elemente sollen eigenständig Elemente sein: Eventuell bestehende Hierarchien zwischen Elementen, wie etwa, daß der Zylinder ein Spezialfall des Kegel ist, sollen aus dem Blickwinkel der Repräsentation nicht sichtbar sein.

- Realitätsbezug: Der Übergang von vorliegenden konkreten Daten zu einer symbolischen Repräsentation ist ein oft vernachlässigtes Problem der KI. Dies gilt insbesondere auch für die Übertragung von Daten eines CAD-Systems (solid modelling: B-REP oder CSG Darstellung (/9/)) in die abstrakteren Daten einer Feature-Darstellung. Der hier vorgestellte Repräsentationsformalismus zur Darstellung von Werkstücken soll diese Brückenfunktion übernehmen, indem er einerseits den Anschluß an die symbolische Welt (Feature-Repräsentation) gewährleistet und indem er andererseits den Anschluß an die Repräsentation eines Werkstücks in den CAD-Daten (etwa aus einem STEP¹-Datenfile) ermöglicht.

¹ STEP (STandard for the Exchange of Product model data) ist der projektierte ISO-Standard für den Datenaustausch in der CIM-Welt. Die Verwendung dieses Standards sollte u. a. die Anbindung existierender CAD-Systeme ermöglichen. Für eine Beschreibung von STEP vgl. /3/.

2. Die gewählte Repräsentation

2.1. Prinzipien

Die in diesem Papier vorgeschlagene Repräsentation stellt das Werkstück durch die es begrenzenden Oberflächen dar und reichert diese Darstellung mit weiteren Attributen an. Der vorgeschlagene Repräsentationsformalismus realisiert also eine erweiterte *Boundary Representation* (B-REP, vgl. /9/) des Werkstücks.

Die atomaren Beschreibungselemente sind Flächenprimitive, diese bilden die Basis einer symbolischen Werkstückrepräsentation. Die Oberfläche des zu repräsentierenden Werkstücks wird aus den Teilflächen zusammengesetzt, wobei die Position der einzelnen Teilflächen im Raum in einem globalen Koordinatensystem exakt angegeben wird. Alle Teilflächen sind Grenzflächen des Materials. Für jede Fläche wird angegeben, in welcher Richtung (also: auf welcher Seite der Fläche) sich das Material des Werkstücks befindet bzw. welche Seite der Fläche bearbeitet werden kann. Damit beschreiben die Teilflächen die elementaren Bearbeitungszonen des Werkstücks: jede Fläche entsteht durch Bearbeitung des Rohteils, jede Bearbeitung des Rohteils wirkt auf dessen Oberfläche ein.

Eine Fläche kann Bereiche enthalten, die bei der Bearbeitung der Fläche ausgespart werden müssen, also nicht bearbeitet werden dürfen. Diese *Löcher* im arbeitstechnischen Sinne, die durchaus nicht geometrische Löcher (d. h. Stellen, an denen in der Fläche Material fehlt) sein müssen, entstehen u. U. durch den Schnitt verschiedener Oberflächen. Durch Angabe dieser Oberflächen kann ein Loch explizit benannt werden.

Technologische Informationen beziehen sich immer auf Flächen. Sie werden durch Attribute zu den Flächen dargestellt. Vorgesehen sind Angaben über Formelemente (Gewinde), Toleranzen, Passungen, Oberflächen- und Härteangaben.

Wo Flächen zusammenstoßen, entstehen Kanten. Diese stellen die *Übergangszone* zwischen Flächen dar und können durch Angabe dieser Flächen und Bedingungen an die Art des *Überganges* repräsentiert werden.

Die Repräsentation eines Körpers geschieht in einem globalen kartesischen Koordinatensystem. Die Orientierung dieses Systems entspricht den in der NC-Technik üblichen Vereinbarungen: Die Symmetrieachse eines Körpers ist die z-Achse des Koordinatensystems und weist nach rechts, die x-Achse weist nach oben.

Die absoluten Beträge der Koordinaten geben die konkreten Maße des Werkstücks an. Die Maßeinheit des Koordinatensystems wird für ein Werkstück angegeben.

2.2. Betrachtete Alternativen

Im Laufe der Entwicklung der vorliegenden Repräsentation wurden verschiedene Alternativen diskutiert². Intensiv betrachtet wurden insbesondere die Konturlinien- und die Scheibchendarstellung.

Die Konturliniendarstellung beschreibt die Oberfläche des Werkstücks durch Angabe einer Kurve in der x - z -Ebene des Koordinatensystems. Die Oberfläche des Werkstücks entsteht, wenn die Kurve um die z -Achse rotiert. Diese Darstellung ist einfach, aber ausschließlich für rotationssymmetrische Körper brauchbar. Dies ist für ARC-TEC eine zu strenge Einschränkung, so daß die Darstellung unbrauchbar ist.

Die Idee der Scheibchendarstellung liegt ebenfalls nahe: Ein Werkstück wird als Folge von primitiven Elementen (Scheiben) dargestellt; den Scheiben können weitere Informationen zugeordnet werden. Probleme treten mit dieser Repräsentation dann auf, wenn Scheiben mit Löchern (etwa Bohrungen) oder mit Innenbearbeitung (vgl. die sogenannten Futterteile) dargestellt werden sollen. Außerdem sind Oberflächeninformationen nicht den ganzen Scheiben, sondern nur einzelnen ihrer Oberflächen zuzuordnen. Welche Scheibe welche Oberflächen beschreibt, hängt aber vom Kontext der sie umgebenden Scheiben ab. Aus diesen Gründen erscheint die Scheibchendarstellung ebenfalls ungeeignet.

²in diesem Abschnitt stützen die Autoren sich auf /1/, /3/, /8/ und /9/

2.3. Einschränkungen

Ziel der vorgestellten Repräsentation ist die Darstellung aller $2\frac{1}{2}$ D Körper und der $2\frac{1}{2}$ D Bearbeitungselemente. Das heißt, daß beliebig gekrümmte Körper vorerst nicht dargestellt werden können. Auch die Abbildung beliebiger Kegelschnitte ist zunächst nicht möglich. Prominentes Beispiel für einen einfachen Körper, der vorerst nicht repräsentiert werden kann, ist die *Leberwurst* .

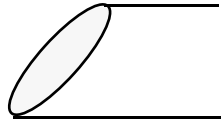


Abb.: "Die Leberwurst"

Diese Einschränkungen sind aber nicht prinzipieller Natur. Durch Hinzunahme von Elementen wie etwa Kugelflächen, Ellipsoidenflächen oder B-Spline-Flächen können auch beliebig geformte Körper dargestellt werden. Die in STEP definierten Primitive können ggf. als Vorbild dienen.

Weiterhin ist daran gedacht, u. a. im Raum gekrümmte Zylinder oder Toroiden durch die Angabe von Mittellinienfunktionen und entsprechende Radienfunktionen zu beschreiben, was eine weitere Verallgemeinerung der bisher eingeführten Grundelemente bedeutet.

3. Beschreibung der Primitive

3.1. Syntax

Die einzelnen Elemente werden durch Angabe ihres Namenskürzels und der dazugehörigen Attribute beschrieben. Jedes Beschreibungselement erhält außerdem eine eindeutige Nummer, über die es referenziert werden kann. Die Reihenfolge der Nummern selbst ist ohne Bedeutung, jedoch müssen alle Beschreibungselemente für ein Werkstück in einem durch ein Minimum und ein Maximum gegebenen Nummern-Intervall liegen, und in diesem Intervall darf kein Beschreibungselement eines anderen Werkstücks auftreten.

In Anlehnung an die STEP-Konventionen wird in diesem Papier folgende Syntax verwendet:

@Nummer := Namenskürzel (Attributname: Attributwert, ...)

Dabei treten als Attributwertebereiche folgende Bezeichnungen auf:

<string>	Zeichenreihe
<char>	Buchstabe
<num>	eine (reelle) Zahl
<vector>	Vektor: (x,y,z)
<list>	Liste

Wird in den Attributen eine Nummer referenziert, so wird dies durch ein vorangestelltes Kreuz # kenntlich gemacht. Vgl. hierzu die Beispiele im Anhang.

Für die maschinelle Verarbeitung im Rahmen von ARC-TEC ist eine LISP-ähnliche Syntax geeigneter. Für die Abbildung auf Dateien wird daher folgende äquivalente Syntax definiert:

(Namenskürzel Nummer (Attributname Attributwert))

Die Referenzen auf andere Nummern werden dabei nicht mehr syntaktisch gekennzeichnet.

3.2. Allgemeine Werkstückinformationen

Zur Repräsentation eines gegebenen Werkstücks sind neben den konkreten Geometrie- und Technologiedaten auch einige globale Informationen nötig. Dazu gehören Angaben zum Werkstück als Ganzes (wie Bezeichnung oder Gesamtlänge) und Verwaltungsinformationen des Repräsentationsformalismus (wie Maßeinheit und Numerierung).

3.2.1. Globale Werkstückinformationen

Dieses Beschreibungselement faßt die zu einem Werkstück gehörenden Informationen zusammen und spezifiziert die für das gesamte Werkstück gültigen Angaben wie Maßeinheit, Bezeichnung, Gewicht und allgemeine Informationen. Hier kann das gesamte Schriftfeld einer technischen Zeichnung abgebildet werden. Das genaue Format dieser zusätzlichen Informationen wird noch festgelegt.

Werkstück (**WorkPiece**):

```
WP( name: <string>,  
    min: <num>,  
    max: <num>,  
    unit: <string>,  
    info: <string>)
```

Attribute:

name	Bezeichnung des Werkstücks.
min	Kleinste Nummer eines Beschreibungselements für dieses Werkstück
max	Größte Nummer eines Beschreibungselements für dieses Werkstück
unit	Die Einheit des Koordinatensystems (m, cm ,mm ...)
info	beliebige weitere Informationen für das gesamte Werkstück. Dieses Attribut wird noch näher spezifiziert.

Konsistenzbedingungen:

Kein Beschreibungselement eines anderen Werkstücks darf eine Nummer zwischen Min und Max haben.

3.2.2. Geometrie- und Technologiemarkros

In diesem Beschreibungselement sollen einige wenige, für die Grobbeschreibung des Werkstücks wichtige Daten repräsentiert werden können. Dies sind z.B. die Gesamtlänge des Werkstücks, der mittlere Durchmesser, die schärfste Toleranz, die beste Oberflächenangabe usw.

Geometrie- und Technologiemarkro (Geometry and Technology Macros)

```
GTM( Lmax: <num>,  
      Dmax: <num>,  
      Dmin: <num>,  
      Daverage: <num>,  
      Hmax: <num>,  
      Fmin: <num>,  
      ITmin: <num>)
```

Attribute:

Lmax	die maximale Länge des Werkstücks
Dmax	der maximale Durchmesser des Werkstücks
Dmin	der minimale Durchmesser des Werkstücks
Daverage	der mittlere Durchmesser des Werkstücks
Hmax	die maximale Härte des Werkstücks
Fmin	die minimale (beste) Oberflächengüte des Werkstücks
ITmin	die minimale (schärfste) Toleranz des Werkstücks

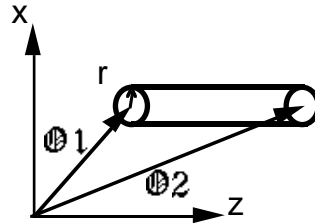
Konsistenzbedingungen:

keine

3.3. Rotationsflächen oder Teile von Rotationsflächen

3.3.1. Zylindermantelfläche

Zylindermantelfläche (Cylinder Jacket)



CJ(reference_point1: <vector>,
reference_point2: <vector>,
radius: <num>,
direction_of_material: [+,-])

Attribute:

reference_point1 Referenzpunkt P_1

reference_point2 Referenzpunkt P_2

radius Radius

direction_of_material Angabe der Materialrichtung

+ \Leftrightarrow das Material befindet sich in einer e-Umgebung der
Mantelfläche in positiver, radialer.

- \Leftrightarrow das Material befindet sich in einer e-Umgebung der
Mantelfläche in negativer, radialer Richtung.

Implizit: $L = |P_2 - P_1|$ Länge des Zylinders

Definition: $D = P_2 - P_1$ Richtung des Zylinders

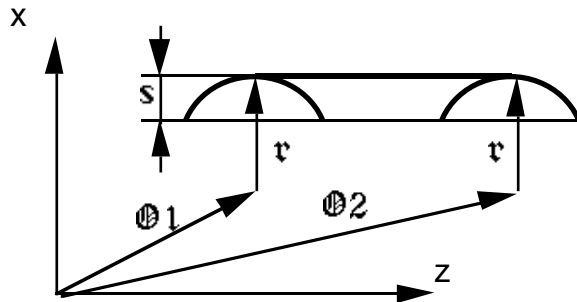
Konsistenzbedingungen:

$P_1 \neq P_2$

$r > 0$

3.3.2. Zylindermantelabschnittfläche

Zylindermantelabschnittfläche (Cylinder Jacket Section)



CJS(reference_point1: <vector>,
reference_point2: <vector>,
reference_point3: <vector>,
height: <num>
direction_of_material: [+,-])

Attribute:

reference_point1	Referenzpunkt Ø1
reference_point2	Referenzpunkt Ø2
reference_point3	Referenzpunkt \mathfrak{r} , der Radius des Zylindermantels und die Symmetrieachse
height	Höhe des Zylindermantelabschnitts
direction_of_material	Angabe der Materialrichtung
	+ \Leftrightarrow das Material befindet sich in einer e-Umgebung der Mantelfläche in positiver, radialer Richtung.
	- \Leftrightarrow das Material befindet sich in einer e-Umgebung der Mantelfläche in negativer, radialer Richtung.

Implizit:	$L = \text{Ø2} - \text{Ø1} $	Länge des Zylinders
Definition:	$\mathfrak{D} = \text{Ø2} - \text{Ø1}$	Richtung des Zylinders

Konsistenzbedingungen:

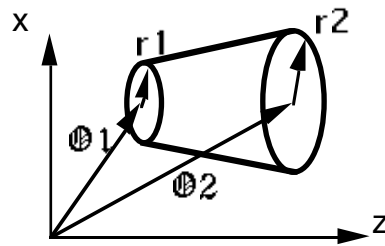
$$\text{Ø1} \neq \text{Ø2}$$

$$s > 0$$

$$|\mathfrak{r}| > 0$$

3.3.3. Kegelmantelfläche

Kegelmantelfläche (Envelope of Cone)



EC(reference_point1: <vector>,
reference_point2: <vector>,
radius1: <num>,
radius2: <num>,
direction_of_material: [+,-])

Attribute:

reference_point1	Referenzpunkt P_1
reference_point2	Referenzpunkt P_2
radius1	Radius r_1
radius2	Radius r_2
direction_of_material	Angabe der Materialrichtung
+ \Leftrightarrow	das Material befindet sich in einer e-Umgebung der Mantelfläche in positiver, radialer Richtung.
- \Leftrightarrow	das Material befindet sich in einer e-Umgebung der Mantelfläche in negativer, radialer Richtung.

Die Radien sind Beträge von Radiusvektoren, die in Ebenen liegen, die senkrecht auf dem Richtungsvektor $\mathbf{d} = \mathbf{P}_2 - \mathbf{P}_1$ liegen.

Implizit: $L = |\mathbf{P}_2 - \mathbf{P}_1|$ Länge des Kegels
Definition: $\mathbf{d} = \mathbf{P}_2 - \mathbf{P}_1$ Richtung des Kegels

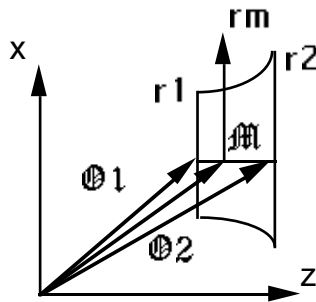
Konsistenzbedingungen:

- $P_1 \neq P_2$
- $r_1 \geq 0, r_2 \geq 0$
- $r_1 \neq r_2$

3.3.4. Torusabschnittfläche

Eine Torusfläche ist diejenige Fläche, die eine Kreislinie (die Erzeugende) beschreibt, wenn deren Mittelpunkt längs einer anderen Kreislinie (der Direktionslinie) bewegt wird. Die Torusabschnittfläche ist ein Ausschnitt einer Torusfläche.

Torusabschnittfläche (Toroidal Shell)



TS(reference_point1: <vector>,
reference_point2: <vector>,
reference_point3: <vector>,
radius1: <num>,
radius2: <num>,
radius3: <num>
direction_of_arch: [v,x]
direction_of_material: [+,-])

Attribute:

reference_point1	Referenzpunkt O1
reference_point2	Referenzpunkt O2
reference_point3	Referenzpunkt M: Mittelpunkt der Direktionslinie des Torus
radius1	Radius r1
radius2	Radius r2
radius3	Radius der Direktionslinie des Torus rm
direction_of_arch	Beschreibung der Wölbungsrichtung
x ⇔	die Wölbung ist konvex.
v ⇔	die Wölbung ist konkav
direction_of_material	Angabe der Materialrichtung
+ ⇔	das Material befindet sich in einer e-Umgebung der Torusfläche in positiver, radialer Richtung.

- \Leftrightarrow das Material befindet sich in einer e-Umgebung der Torusfläche in negativer, radialer Richtung.

Die Radien sind die Beträge von Radiusvektoren \mathbf{r} . Diese Radiusvektoren rotieren in Ebenen, die senkrecht auf dem Achsenvektor $\mathbf{a} = \mathbf{r}_2 - \mathbf{r}_1$ stehen.

Implizit: $\mathbf{a} = \mathbf{r}_2 - \mathbf{r}_1$ Achsenvektor des Torus
 $r_{\text{Erz}} = \sqrt{(r_1 - r_m)^2 + |\mathbf{r}_1 - \mathbf{m}|^2}$ Radius der Erzeugenden

Konsistenzbedingungen:

$$\mathbf{r}_2 \neq \mathbf{r}_1 \neq \mathbf{m}$$

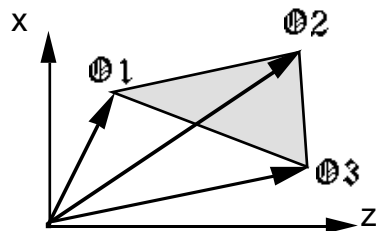
$r_1 \geq 0, r_2 \geq 0, r_m \geq 0$, wobei entweder $r_1 > 0$ oder $r_2 > 0$

$$(r_2 - r_m)^2 + |\mathbf{r}_2 - \mathbf{m}|^2 = (r_1 - r_m)^2 + |\mathbf{r}_1 - \mathbf{m}|^2$$

3.4. Ebene Flächen

3.4.1. Dreiecksfläche

Dreiecksfläche (Triangle)



T(**reference_point1:** <vector>,
 reference_point2: <vector>,
 reference_point3: <vector>,
 direction_of_material: [+,-])

Die Dreiecksfläche wird durch drei Punkte im Raum bestimmt.

Attribute:

reference_point1	Referenzpunkt P_1
reference_point2	Referenzpunkt P_2
reference_point3	Referenzpunkt P_3 : Mittelpunkt der Richtungsline des Torus
direction_of_material	Angabe der Materialrichtung
+ \leftrightarrow	das Material befindet sich in einer e-Umgebung der Dreiecksfläche in Richtung der Flächennormalen.
- \leftrightarrow	das Material befindet sich in einer e-Umgebung der Dreiecksfläche in entgegengesetzter Richtung der Normalen.

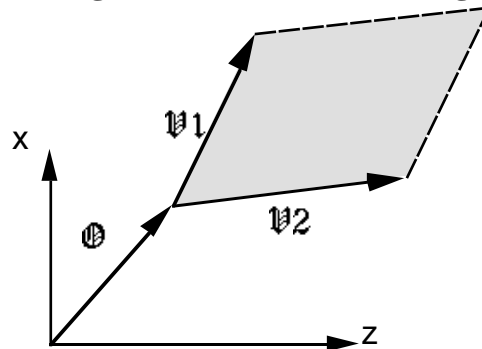
Definition: $\mathbf{n} := (\mathbf{P}_2 - \mathbf{P}_1) \times (\mathbf{P}_3 - \mathbf{P}_1)$ Normalenvektor der Fläche

Konsistenzbedingungen:

$$P_1 \neq P_2 \neq P_3$$

3.4.2. Parallelogrammfläche

Parallelogrammfläche (Parallelogram)



P(**reference_point:** <vector>,
 direction_vector1: <vector>,
 direction_vector2: <vector>,
 direction_of_material: [+,-])

Attribute:

reference_point1	Referenzpunkt P
direction_vector1	Richtungsvektor v_1
direction_vector2	Richtungsvektor v_2
direction_of_material	Angabe der Materialrichtung
+ \Leftrightarrow	das Material befindet sich in einer e-Umgebung der Dreiecksfläche in Richtung der Flächennormalen.
- \Leftrightarrow	das Material befindet sich in einer e-Umgebung der Dreiecksfläche in entgegengesetzter Richtung der Normalen.

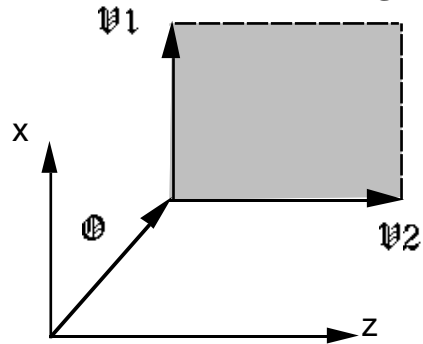
Definition: $\mathbf{N} := (v_1) \times (v_2)$ Normalenvektor der Fläche

Konsistenzbedingungen:

$$|(v_1) \times (v_2)| \neq 0$$

3.4.3. Rechteckfläche

Rechteckfläche (RectAngle)



RA(reference_point: <vector>,
 direction_vector1: <vector>,
 direction_vector2: <vector>,
 direction_of_material: [+,-])

Attribute:

reference_point1	Referenzpunkt P
direction_vector1	Richtungsvektor $v1$
direction_vector2	Richtungsvektor $v2$
direction_of_material	Angabe der Materialrichtung
+ \Leftrightarrow	das Material befindet sich in einer e-Umgebung der Dreiecksfläche in Richtung der Flächennormalen.
- \Leftrightarrow	das Material befindet sich in einer e-Umgebung der Dreiecksfläche in entgegengesetzter Richtung der Normalen.

Definition: $N := (v1) \times (v2)$ Normalenvektor der Fläche

Konsistenzbedingungen:

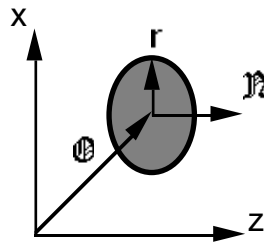
$$\langle (v1), (v2) \rangle = 0$$

$$v1 \neq 0$$

$$v2 \neq 0$$

3.4.4. Kreisfläche

Kreisfläche (Circular surface)



C(**reference_point:** <vector>,
 direction_vector: <vector>,
 radius: <num>,
 direction_of_material: [+,-])

Attribute:

reference_point	Referenzpunkt \mathbb{P} : Lage des Mittelpunktes
direction_vector	Richtungsvektor \mathbb{N} : Normalenvektor der Kreisfläche
radius	Radius r
direction_of_material	Angabe der Materialrichtung
+ \Leftrightarrow	das Material befindet sich in einer e-Umgebung der Fläche in Richtung des Normalenvektors.
- \Leftrightarrow	das Material befindet sich in einer e-Umgebung der Fläche in entgegengesetzter Richtung des Normalenvektors

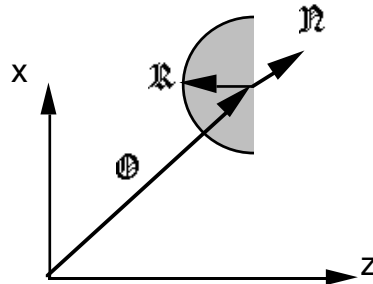
Konsistenzbedingungen:

$r > 0$

$\mathbb{N} \neq 0$

3.4.5. Halbkreisfläche

Halbkreisfläche (Area of the Semi-Circle)



ASC(reference_point1: <vector>,
reference_point2: <vector>,
direction_vector: <vector>,
direction_of_material: [+,-])

Attribute:

reference_point1 Referenzpunkt \odot : Lage des Mittelpunktes
reference_point2 Referenzpunkt \mathbb{R} , der Radius des Kreises und die
 Symmetrieachse
direction_vector Richtungsvektor \mathbb{N} : Normalenvektor der Halbkreisfläche
direction_of_material Angabe der Materialrichtung
+ \Leftrightarrow das Material befindet sich in einer e-Umgebung der Fläche in
 Richtung des Normalenvektors.
- \Leftrightarrow das Material befindet sich in einer e-Umgebung der Fläche in
 entgegengesetzter Richtung des Normalenvektors

Konsistenzbedingungen:

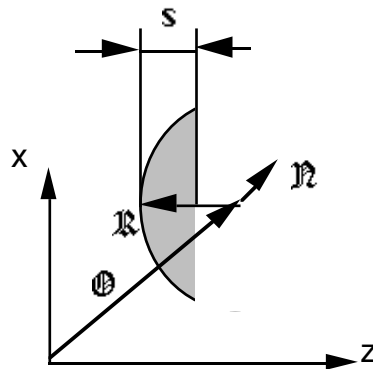
$$\mathbb{R} \neq 0$$

$$\mathbb{N} \neq 0$$

$$\langle \mathbb{N}, \mathbb{R} \rangle = 0$$

3.4.6. Kreisabschnittfläche

Kreisabschnittfläche (Circular surface Segment)



CS(reference_point1: <vector>,
reference_point2: <vector>,
direction_vector: <vector>,
height: <num>
direction_of_material: [+,-])

Attribute:

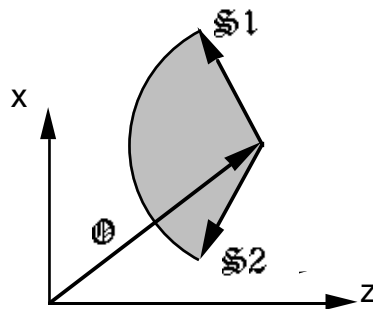
reference_point1	Referenzpunkt \textcircled{C} : Lage des Mittelpunktes
reference_point2	Referenzpunkt \mathcal{R} , der Radius des Kreises und die Symmetrieachse
direction_vector	Richtungsvektor \mathcal{N} : Normalenvektor der Halbkreisfläche
height	Höhe des Kreisabschnitts
direction_of_material	Angabe der Materialrichtung
+ \Leftrightarrow	das Material befindet sich in einer e-Umgebung der Fläche in Richtung des Normalenvektors.
- \Leftrightarrow	das Material befindet sich in einer e-Umgebung der Fläche in entgegengesetzter Richtung des Normalenvektors

Konsistenzbedingungen:

$\mathcal{R} \neq 0$
 $\mathcal{N} \neq 0$
 $\langle \mathcal{N}, \mathcal{R} \rangle = 0$
 $s > 0$

3.4.7. Kreissegmentfläche

Kreissegmentfläche (Circular surface sector)



CT(reference_point: <vector>,
direction_vector1: <vector>,
direction_vector2: <vector>,
angle: [g,l]
direction_of_material: [+,-])

Attribute:

reference_point1	Referenzpunkt \odot
direction_vector1	Richtungsvektor $\mathcal{S}1$
direction_vector2	Richtungsvektor $\mathcal{S}2$
angle	Angabe über den Öffnungswinkel des Segmentes
g \Leftrightarrow	Winkel größer 180 Grad
l \Leftrightarrow	Winkel kleiner 180 Grad
direction_of_material	Angabe der Materialrichtung
+ \Leftrightarrow	das Material befindet sich in einer e-Umgebung der Dreiecksfläche in Richtung der Flächennormalen.
- \Leftrightarrow	das Material befindet sich in einer e-Umgebung der Dreiecksfläche in entgegengesetzter Richtung der Normalen.

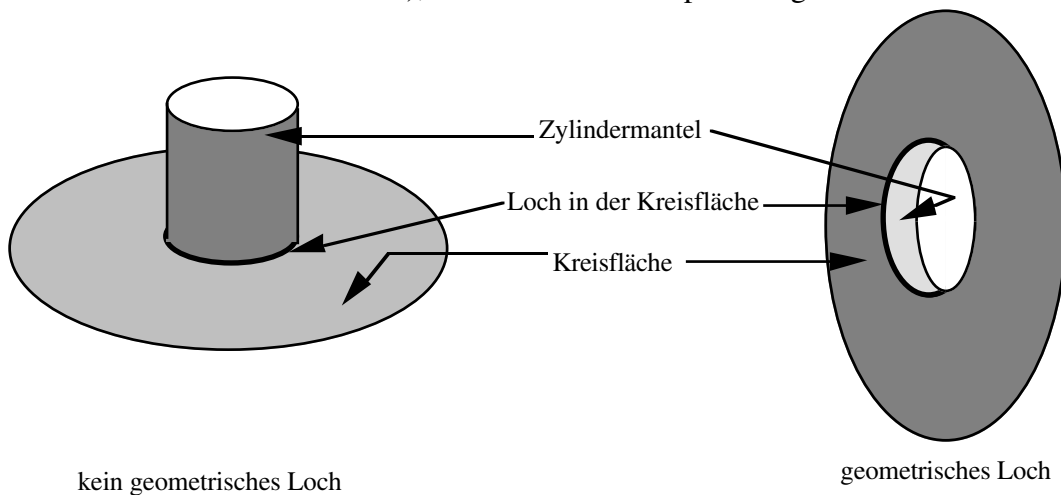
Definition: $\mathcal{N} := (\mathcal{V}1) \times (\mathcal{V}2)$ Normalenvektor der Fläche

Konsistenzbedingungen:

$|(\mathcal{V}1) \times (\mathcal{V}2)| \neq 0$

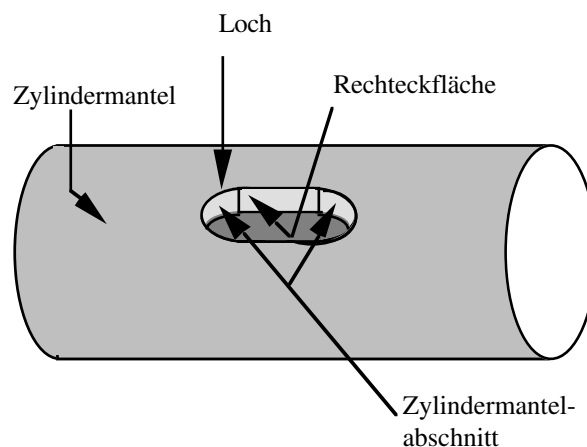
3.5. Gelochte Flächen

Eine Fläche kann Bereiche enthalten, die bei der Bearbeitung der Fläche ausgespart werden müssen, also nicht bearbeitet werden dürfen. Solche Bereiche sind im arbeitstechnischen Sinne Löcher. Sie können durch den Schnitt verschiedener Oberflächen entstehen und sind nicht notwendig geometrische Löcher (d. h. Stellen, an denen in der Fläche Material fehlt), wie die beiden Beispiele zeigen:

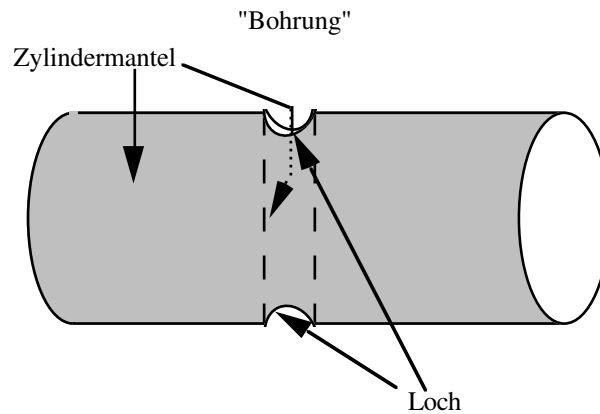


Um ein Loch zu beschreiben, wird die Fläche angegeben, die das Loch enthält ("Basisfläche"), und die Fläche, die sich mit der Basisfläche schneidet und dadurch das Loch erzeugt. (schneidende Fläche) Da zu jeder Fläche die Bearbeitungsrichtung (bzw. der Ort des Materials) angegeben ist, ist das Loch durch die Angabe der Flächen in der Regel eindeutig beschrieben.

In der Praxis treten einige Löcher auf, deren Repräsentation einen höheren Beschreibungsaufwand erfordert. So gibt es Löcher, die durch den Schnitt mehrerer Teilflächen mit einer Basisfläche entstehen. Ein wichtiges Beispiel hierfür ist eine Paßfedernut in einer Welle:



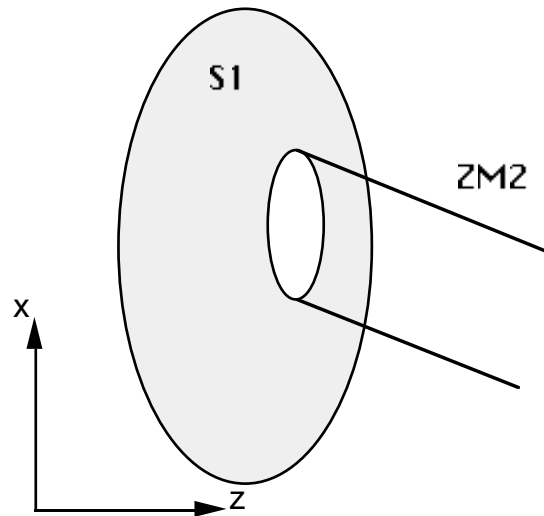
Schließlich kann es geschehen, daß eine einzige Fläche beim Schnitt mit einer anderen Fläche zwei Löcher erzeugt. Prominentes Beispiel ist die zylindrische Bohrung durch ein zylindrisches Werkstück:



Dieser Fall führt zu zwei bis auf die Referenznummer identischen Lochbeschreibungen, da ja beide Löcher von denselben Flächen gebildet werden. Zur eindeutigen Identifizierung eines einzelnen Loches muß daher ggf. auf die Kantenrepräsentation zurückgegriffen werden(siehe

3.5.1. Gelochte Fläche

Gelochte Fläche (Perforated Surface)



**PS(reference_number: <num>,
set_of_intersecting_surfaces: <list>)**

Attribute:

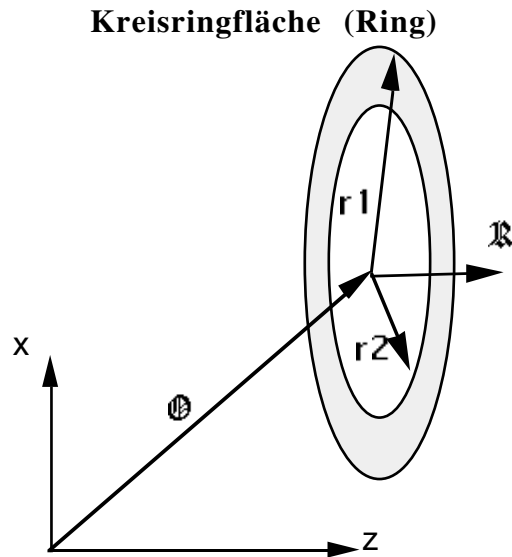
reference_number Referenz-Nummer der Basisfläche
set_of_intersecting_surfaces Referenz-Nummern der schneidenden Flächen

Konsistenzbedingungen:

Die Basisfläche muß (topologisch) zusammenhängend bleiben.

3.5.2. Kreisringfläche

Eine in der Praxis häufig vorkommende gelochte Fläche ist die Kreisringfläche, deshalb wird sie extra repräsentiert, um den Beschreibungsaufwand zu minimieren.



R(**reference_point:** <vector>,
 direction_vector: <vector>,
 radius1: <num>
 radius2: <num>
 direction_of_material: [+,-])

Attribute:

reference_point	Referenzpunkt O
direction_vector	Normalenvektor \mathbb{R}
radius1	Radius r1
radius2	Radius r2
direction_of_material	Angabe der Materialrichtung
+ \Leftrightarrow	das Material befindet sich in einer e-Umgebung der Fläche in Richtung des Normalenvektors \mathbb{R} .
- \Leftrightarrow	das Material befindet sich in einer e-Umgebung der Fläche in entgegengesetzter Richtung des Normalenvektors \mathbb{R}

Konsistenzbedingungen:

$\mathbb{R} \neq 0$
 $r1 > r2 > 0$

3.6. Flächenelemente für rotationssymmetrische Werkstücke

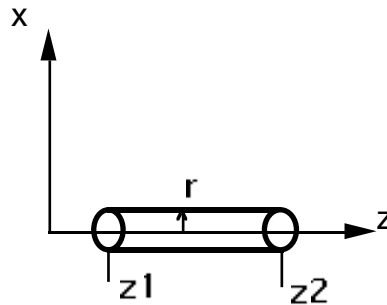
Die Beschreibung eines rotationssymmetrischen Teils ist immer eine Spezialisierung der allgemeinen Darstellung dieses Teils; bei der spezialisierten Darstellung für rotationssymmetrische Teile werden die ansonsten vorkommenden Ortsvektoren ($\mathbb{O}1$, $\mathbb{O}2$) durch Zahlen ($z1$, $z2$) ersetzt. Das heißt insbesondere, daß rotationssymmetrische Teile immer (rotationssymmetrisch) um die Z-Achse angeordnet sind. Eine weiterführende Änderung der Repräsentation ist nicht vorgesehen. Bei den Namens Kürzel wird zur Unterscheidung zu den nicht rotationssymmetrischen Flächen ein **RS** (Rotational Symmetrical) vorne angestellt.

Um rotationssymmetrische Körper geometrisch zu beschreiben, reichen fünf Beschreibungselemente aus. Dies sind Zylindermantelfläche, Kreisfläche, Kegelmantelfläche, Torusfläche und die Kreisringfläche.

Die jeweiligen Flächenbeschreibungen werden in den folgenden Kapiteln vorgestellt.

3.6.1. Zylindermantelfläche

Zylindermantel (Cylinder Jacket)



RSCJ(**reference_point1:** <num>,
 reference_point2: <num>,
 radius: <num>,
 direction_of_material: [+,-])

Attribute:

reference_point1	Referenzpunkt z1
reference_point2	Referenzpunkt z2
radius	Radius
direction_of_material	Angabe der Materialrichtung
+ ⇔	das Material befindet sich in einer e-Umgebung der Mantelfläche in positiver, radialer.
- ⇔	das Material befindet sich in einer e-Umgebung der Mantelfläche in negativer, radialer Richtung.

Implizit: $L = |z2 - z1|$ Länge des Zylinders

Definition: Richtung des Zylinders ist die Richtung der z-Achse

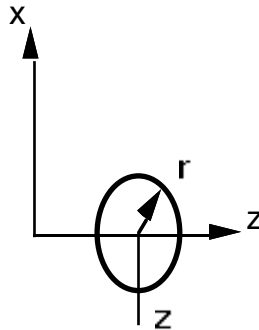
Konsistenzbedingungen:

$z1 \neq z2$

$r > 0$

3.6.2. Kreisfläche

Kreisfläche (Circular surface)



**RSC(reference_point: <num>,
radius: <num>,
direction_of_material: [+,-])**

Attribute:

reference_point	Referenzpunkt z: Lage des Mittelpunktes
radius	Radius r
direction_of_material	Angabe der Materialrichtung
+ ⇔	das Material befindet sich in einer e-Umgebung der Fläche in Richtung des Normalenvektors.
- ⇔	das Material befindet sich in einer e-Umgebung der Fläche in entgegengesetzter Richtung des Normalenvektors

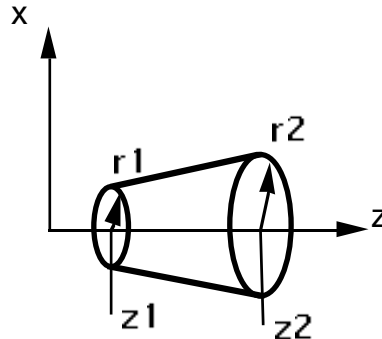
Definition: Richtung der Flächennormalen ist die Richtung der z-Achse

Konsistenzbedingungen:

$r > 0$

3.6.3. Kegelmantelfläche

Kegelmantelfläche (Envelope of Cone)



RSEC(**reference_point1:** <num>,
 reference_point2: <num>,
 radius1: <num>,
 radius2: <num>,
 direction_of_material: [+,-])

Attribute:

reference_point1	Referenzpunkt z1
reference_point2	Referenzpunkt z2
radius1	Radius r1
radius2	Radius r2
direction_of_material	Angabe der Materialrichtung
+ ⇔	das Material befindet sich in einer e-Umgebung der Mantelfläche in positiver, radialer Richtung.
- ⇔	das Material befindet sich in einer e-Umgebung der Mantelfläche in negativer, radialer Richtung.

Implizit: $L = |z1 - z2|$ Länge des Kegels

Definition: Richtung der Kegelmantelfläche ist die Richtung der z-Achse

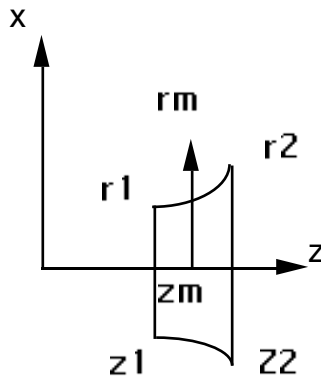
Konsistenzbedingungen:

$z1 \neq z2$
 $r1 \geq 0, r2 \geq 0$
 $r1 \neq r2$

3.6.4. Torusabschnittfläche

Eine Torusfläche ist diejenige Fläche, die eine Kreislinie (die Erzeugende) beschreibt, wenn deren Mittelpunkt längs einer anderen Kreislinie (der Direktionslinie) bewegt wird. Die Torusabschnittfläche ist ein Ausschnitt einer Torusfläche.

Torusabschnittfläche (Toroidal Shell)



RSTS(**reference_point1:** <num>,
 reference_point2: <num>,
 reference_point3: <num>,
 radius1: <num>,
 radius2: <num>,
 radius3: <num>
 direction_of_arch: [v,x]
 direction_of_material: [+,-])

Attribute:

reference_point1	Referenzpunkt z1
reference_point2	Referenzpunkt z2
reference_point2	Referenzpunkt zm: Mittelpunkt der Direktionslinie des Torus
radius1	Radius r1
radius2	Radius r2
radius3	Radius der Direktionslinie des Torus rm
direction_of_arch	Beschreibung der Wölbungsrichtung
x ⇔	die Wölbung ist konvex.
v ⇔	die Wölbung ist konkav
direction_of_material	Angabe der Materialrichtung
+ ⇔	das Material befindet sich in einer e-Umgebung der Torusfläche in positiver, radialer Richtung.

- \Leftrightarrow das Material befindet sich in einer e-Umgebung der Torusfläche in negativer, radialer Richtung.

Definition: Die von r_1 und r_2 aufgespannten Kreisflächen stehen senkrecht auf der z-Achse

$$r_{Erz} = \sqrt{(r_1 - r_m)^2 + |z_1 - z_m|^2} \quad \text{Radius der Erzeugenden}$$

Konsistenzbedingungen:

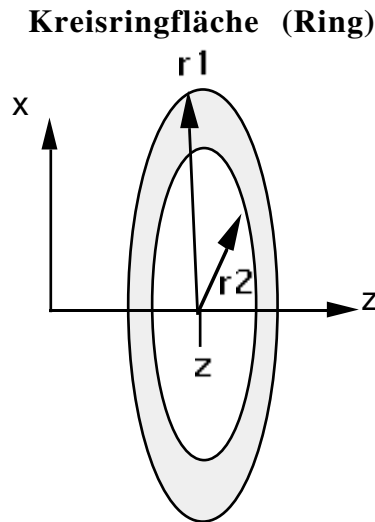
$$z_1 \neq z_2 \neq z_m$$

$r_1 \geq 0, r_2 \geq 0, r_m \geq 0$, wobei entweder $r_1 > 0$ oder $r_2 > 0$

$$(r_2 - r_m)^2 + |z_2 - z_m|^2 = (r_1 - r_m)^2 + |z_1 - z_m|^2$$

3.5.2. Kreisringfläche

Eine in der Praxis häufig vorkommende gelochte Fläche ist die Kreisringfläche, deshalb wird sie extra repräsentiert, um den Beschreibungsaufwand zu minimieren.



**RSR(reference_point: <vector>,
radius1: <num>
radius2: <num>
direction_of_material: [+,-])**

Attribute:

reference_point	Referenzpunkt z
radius1	Radius r1
radius2	Radius r2
direction_of_material	Angabe der Materialrichtung
+ ⇔	das Material befindet sich in einer e-Umgebung der Fläche in Richtung des Normalenvektors \mathbb{R} .
- ⇔	das Material befindet sich in einer e-Umgebung der Fläche in entgegengesetzter Richtung des Normalenvektors \mathbb{R} .

Definition: Die Flächennormale wird durch die Richtung der z-Achse definiert

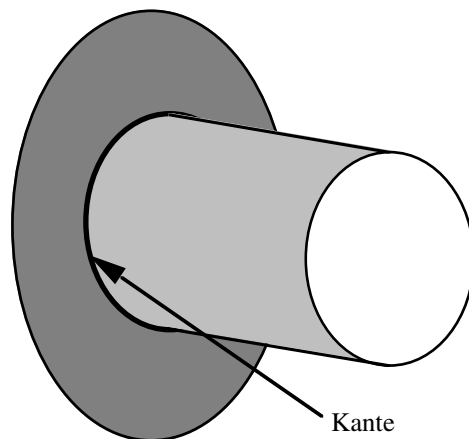
Konsistenzbedingungen:

$$r1 > r2 > 0$$

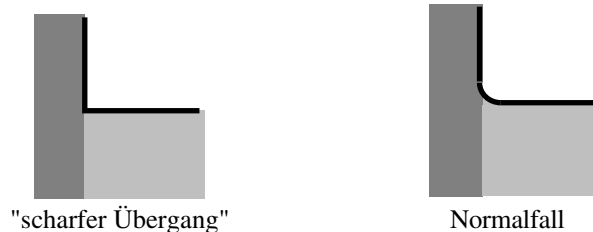
3.7. Beschreibung der technologischen Grundelemente

3.7.1. Kante (Edge)

Wo immer zwei Flächen zusammenstoßen, entsteht eine Kante. Diese beschreibt den Übergang zwischen den Flächen und kann mit technologischen Informationen angereichert werden.



Vorgesehen ist die Beschreibung der Art der Kante: Scharfer Übergang oder Normal. Eine "normale" Kante ist ein Übergang mit kleinem Radius. Der Radius entsteht durch die Verwendung eines "normalen", nicht speziell für "scharfe" Übergänge vorgesehenen Werkzeugs. Soll dieser Radius nicht auftreten, müssen spezielle Vorkehrungen getroffen werden, um einen "scharfen Übergang" zu erzeugen.



Soll eine Kante mit einem explizit spezifizierten Übergangsradius hergestellt werden, so muß dieser auch explizit repräsentiert werden, etwa (bei Wellen) als Torusfläche.

Defaultmäßig gelten alle Kanten als "normal", was bedeutet, daß die Übergänge nicht genauer spezifiziert sind und somit mit einem "normalen" Werkzeug mit Eckenradius bearbeitet werden können. (Was ein "normales" Werkzeug ist, hängt natürlich von Randbedingungen etwa des Fertigungsbetriebes ab). Dies gilt auch für Kanten, die in der Repräsentation nicht explizit aufgeführt sind, sich aber aus den Flächen ergeben.

**E(surface_number1: <num>,
 surface_number2: <num>,
 transition_condition: [blunt, normal, sharp])**

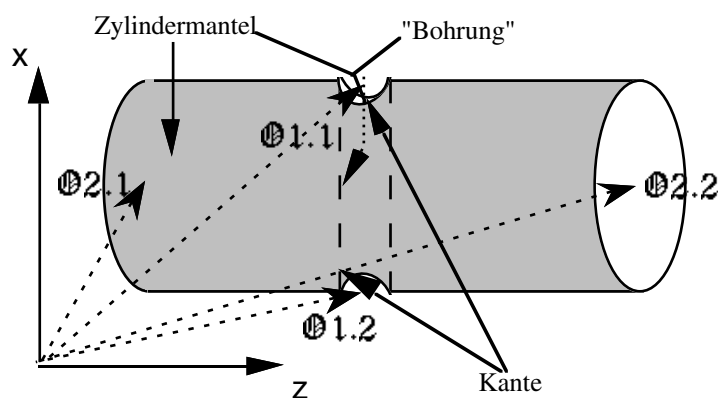
Atribute:

surface_number1 Elementnummer einer der beiden Flächen
 surface_number2 Elementnummer einer der beiden Flächen
 transition_condition Übergangsbedingung: stumpf, normal, scharf

Konsistenzbedingungen:

surface_number1 \neq surface_number2
 surface_number1 und surface_number2 müssen direkt benachbarte Flächen sein.

Wie bereits bei der Beschreibung der Löcher bemerkt, kann der Fall auftreten, daß zwei Flächen mehr als eine Kante miteinander bilden (siehe Abbildung)



Soll einer dieser Kanten eine technologische Information zugeordnet werden, so müssen die Kanten eindeutig identifiziert werden. Hierzu dient die erweiterte Beschreibung

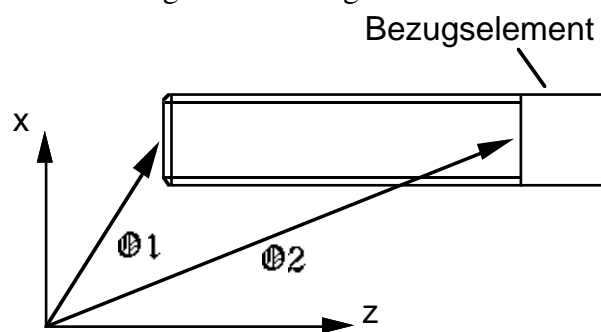
**ED(surface_number1: <num>,
 surface_number2: <num>,
 transition_condition: [blunt, normal, sharp]
 reference_point: <vector>)**

Atribute:

surface_number1 Elementnummer einer der beiden Flächen

3.7.2. Gewinde (Thread)

Die Beschreibung von Gewinden kann i. a. durch die Angabe von Parametern und Bezug zu einer Norm geschehen, wobei zusätzlich noch Informationen über die räumliche Lage des Gewindes gegeben werden müssen. Ein Gewinde ist immer auf einem geometrischen Grundelement, dem Bezugselement, angeordnet. Als Bezugselemente kommen insbesondere Zylinder- und Kegelmantel in Frage. Anfang und Ende des Gewindes auf dem Bezugselement werden durch zwei Ortsvektoren gegeben, die gleichzeitig die Symmetrielinie des Gewindes festlegen. Da diese mit der Symmetrielinie des Bezugselements übereinstimmen muß, müssen beide Ortsvektoren auf der Symmetrielinie des Bezugselements liegen.



**TH(surface_number: <num>,
reference_point1: <vector>
reference_point2: <vector>
thread_name: <string>)**

Attribute:

surface_number1	Elementnummer der Bezugsfläche
reference_point1	Referenzpunkt P1
reference_point2	Referenzpunkt P2
thread_name	Bezeichnung des Gewindes

Konsistenzbedingungen:

$P1 \neq P2$

$P1 - P2$ muss der Richtung oder entgegengesetzten Richtung der Bezugsfläche entsprechen.

$P1$ und $P2$ müssen innerhalb des Bereiches der Bezugsfläche liegen.

3.7.3. Maßtoleranz (Size Standard)

Maßtoleranzen beziehen sich entweder auf die Maße eines Geometrieelements oder sie beziehen sich auf Maße zwischen zwei Geometrieelementen. Aus diesem Grund werden die Beschreibungselemente SS und SSD benutzt. Mit SS lassen sich vor allem Durchmesser-toleranzen beschreiben, während mit SSD z.B. die Längentoleranz eines Zylinders über den Abstand der ihn begrenzenden Kreisflächen beschrieben werden kann.

**SS(surface_number: <num>,
nominal_size: <num>,
min_size: <num>,
max_size: <num>)**

Attribute:

surface_number	Elementnummer der Bezugsfläche
nominal_size	Nennmaß
min_size	Kleinstmaß
max_size	Größtmaß

Konsistenzbedingungen:

$\text{min_size} \leq \text{nominal_size} \leq \text{max_size}$

**SSD(surface_number1: <num>,
surface_number2: <num>,
nominal_size: <num>,
min_size: <num>,
max_size: <num>)**

Attribute:

surface_number1	Elementnummer der Bezugsfläche
surface_number2	Elementnummer der Bezugsfläche
nominal_size	Nennmaß
min_size	Kleinstmaß
max_size	Größtmaß

Konsistenzbedingungen:

$\text{min_size} \leq \text{nominal_size} \leq \text{max_size}$

$\text{surface_number1} \neq \text{surface_number2}$

3.7.4. ISO-Passung nach DIN 7154 bzw. 7155 (ISO-Fit)

Mittels dieses Beschreibungselements lassen sich Toleranzen nach dem System der Einheitsbohrung oder Einheitswelle beschreiben. Bei dieser Beschreibungsart werden i.

a. nur Durchmessertoleranzen beschrieben, so daß die Angabe von nur einem Bezugselement ausreichend ist. Eine Toleranzangabe setzt sich immer aus der Toleranzfeldlage (im Beispiel h)und der Toleranzfeldgröße (IT = 6) zusammen.

Beispiel: $\varnothing 57$ h6

ISO(**surface_number:** <num>,
tolerance_extent: <char>,
clearance_space: <num>)

Attribute:

surface_number	Elementnummer der Bezugsfläche
tolerance_extent	Toleranzfeldlage
clearance_space	IT-Qualität

Konsistenzbedingungen:

keine

3.7.5. Allgemeintoleranz DIN 7168 (General Tolerance)

Allgemeintoleranzen bestehen generell zwischen allen bemaßten Elementen eines Produkts und müssen prinzipiell immer in entsprechende Maßtoleranzen umgerechnet werden, um sie überprüfen zu können.

**GT(surface_number: <num>,
nominal_size: <num>,
min_size: <num>,
max_size: <num>)**

Attribute:

surface_number	Elementnummer der Bezugsfläche
nominal_size	Nennmaß
min_size	Kleinstmaß
max_size	Größtmaß

Konsistenzbedingungen:

$\text{min_size} \leq \text{nominal_size} \leq \text{max_size}$

**GTD(surface_number1: <num>,
surface_number2: <num>,
nominal_size: <num>,
min_size: <num>,
max_size: <num>)**

Attribute:

surface_number1	Elementnummer der Bezugsfläche
surface_number2	Elementnummer der Bezugsfläche
nominal_size	Nennmaß
min_size	Kleinstmaß
max_size	Größtmaß

Konsistenzbedingungen:

$\text{min_size} \leq \text{nominal_size} \leq \text{max_size}$

$\text{surface_number1} \neq \text{surface_number2}$

3.7.6. Formtoleranz DIN 7184 (Shape Tolerance)

Mit diesen Beschreibungselementen soll die geometrische Ausprägung von bestimmten Bereichen eines Werkstücks genauer beschrieben werden können.

**ST(surface_number: <num>,
tolerance_type: [SN, EN, RN, C, LD, PD],
tolerance_size: <num>)**


Attribute:


surface_number	Elementnummer der Bezugsfläche
tolerance_type	Toleranztyp
tolerance_size	Toleranzmaß

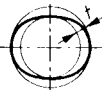
Konsistenzbedingungen:

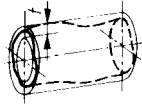
tolerance_type muß mit der Bezugsfläche surface_number kompatibel sein.

Im folgenden sind die möglichen Toleranztypen aufgezählt. Die Toleranzangaben sind immer in mm zu lesen!

SN  **Geradheit (Straightness)**
Die tolerierte Achse des äußeren Zylinders muß innerhalb eines Zylinders vom Durchmesser t liegen.

EN  **Ebenheit (Evenness)**
Die tolerierte Fläche muß zwischen zwei parallelen Ebenen vom Abstand t liegen.

RN  **Rundheit (Roundness)**
In jeder achsenkrechten Schnittebene muß die tolerierte Umfangslinie zwischen zwei konzentrischen Kreisen vom Abstand t liegen.



C

Zylinder (Cylinder)

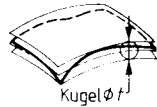
Die tolerierte Zylindermantelfläche muß zwischen zwei coaxialen Zylindern liegen, die einen Abstand von t haben.



LD

Linienform (Line Divergency)

In jeder Schnittebene, parallel zur Zeichenebene, muß das tolerierte Profil zwischen Hülllinien an Kreisen vom Durchmesser t liegen, deren Mittelpunkte auf der geometrisch idealen Linie liegen.



PD

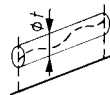
Flächenform (Plane Divergency)

Die tolerierte Fläche muß zwischen zwei Hüllflächen an Kugeln vom Durchmesser t liegen, deren Mittelpunkte auf der geometrisch idealen Fläche liegen.

3.7.7. Lagetoleranzen (Positional Tolerance)

Durch diese Beschreibungselemente sollen Bezüge zwischen einzelnen Bereichen eines Werkstücks dargestellt werden. In der vorgestellten Repräsentation werden Bezugsachsen nicht explizit repräsentiert, sondern werden durch die Bestimmung von Richtungsvektoren, die nach angegebenen Vorschriften aus den Attributen der Elementbeschreibungen berechnet werden, implizit dargestellt. Zur Angabe der Bezugsachse genügt somit die Angabe des Elements, auf dessen Bezugsachse (Richtungsvektor) referenziert werden soll.

Bei rotationssymmetrischen Elementen ist der Richtungsvektor immer die z-Achse und muß somit nicht mehr berechnet werden.



Parallelität (Parallelism)

Die tolerierte Achse eines Elements muß zwischen zwei zur Bezugsachse des Elements parallelen Ebenen vom Abstand t liegen.

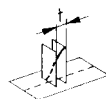
**PM(surface_number: <num>,
reference_axis: <num>,
tolerance_size: <num>)**

Attribute:

surface_number	Flächennummer
reference_axis	Die Nummer der Fläche, auf deren Achse Bezug genommen wird
tolerance_size	Toleranzmaß

Konsistenzbedingungen:

Die Fläche `surface_number` und die Achse `reference_axis` müssen mit dem Toleranztyp kompatibel sein.



Rechtwinkligkeit (Orthogonality)

Eine tolerierte Planfläche muß zwischen zwei parallelen und zur Bezugsachse senkrechten Ebenen vom Abstand t liegen.

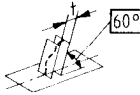
**O(surface_number: <num>,
reference_axis: <num>,
tolerance_size: <num>)**

Attribute:

surface_number	Flächennummer
reference_axis	Die Nummer der Fläche, auf deren Achse Bezug genommen wird
tolerance_size	Toleranzmaß

Konsistenzbedingungen:

Die Fläche surface_number und die Achse reference_axis müssen mit dem Toleranztyp kompatibel sein.



Neigung (Gradient)

Die tolerierte Achse eines Elements muß zwischen zwei parallelen und im Winkel von a Grad zur Bezugsfläche geneigten Ebenen liegen, die einen Abstand von t mm haben.

**G(surface_number1: <num>,
surface_number2: <num>,
tolerance_size: <num>
angle: <num>)**

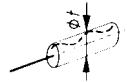
Attribute:

surface_number1	Flächennummer 1
surface_number2	Flächennummer 2
tolerance_size	Toleranzmaß
angle	Winkel zwischen den beiden Flächen

Konsistenzbedingungen:

surface_number1 \neq surface_number2

Die beiden Flächen müssen mit dem Toleranztyp kompatibel sein



Konzentrität (ConCentricity)

Die tolerierte Achse eines Elements muß innerhalb eines zur Bezugsachse koaxialen Zylinders vom Durchmesser t liegen.

**CC(surface_number: <num>,
reference_axis: <num>,
tolerance_size: <num>)**

Attribute:

surface_number	Flächennummer
reference_axis	Die Nummer der Fläche, auf deren Achse Bezug genommen wird
tolerance_size	Tolernzmaß

Konsistenzbedingungen:

Die Fläche surface_number und die Achse reference_axis müssen mit dem Toleranztyp kompatibel sein.

3.7.8. Oberflächenangaben (Surface Finish)

Oberflächenangaben können nach verschiedenen Ermittlungsverfahren angegeben werden, wobei eine Umrechnung zwischen den einzelnen Verfahren nicht korrekt durchgeführt werden kann. Deshalb muß zusätzlich zum angegebenen Wert noch die Bezeichnung des Meßverfahrens angegeben werden.

```
SF( surface_number: <num>,  
    surface_finish: [Rt, Rz, Rp, Ra,...],  
    value: <num>)
```

Attribute:

surface_number	Flächennummer
surface_finish	Oberflächenangabe, wie etwa Rt (Rauhtiefe) oder Rp (Glättungstiefe)
value	Maß der Oberflächengüte

Konsistenzbedingungen:

Der Wert value muß im Intervall der Oberflächengüte surface_finish liegen

3.7.9. Härteangaben (Hardness)

Für Härteangaben gilt das gleiche wie für die Angabe von Oberflächenangaben.

**H(surface_number: <num>,
hardness: [HRC, HRB, HV, HB, ...],
value: <num>)**

Attribute:

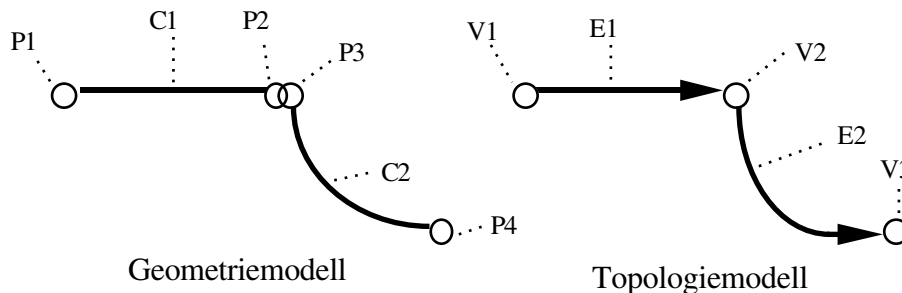
surface_number	Flächennummer
hardness	Härteangabe, wie etwa HRC (Härte in Rockwell Cone) oder HV (Vickers Härte)
value	Maß der Härteangabe

Konsistenzbedingungen:

Der Wert value muß im Intervall der Härteangabe hardness liegen

4. Topologieinformationen

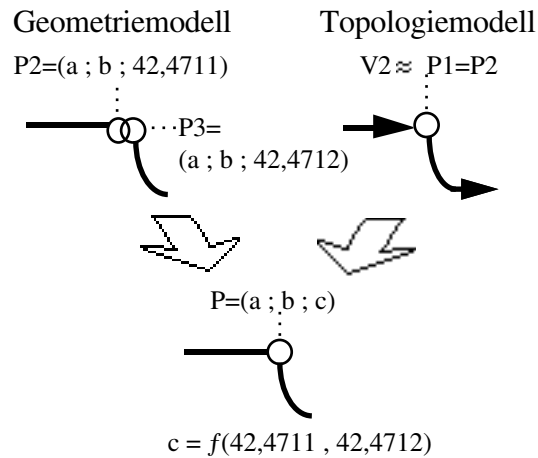
In einem Topologiemodell werden Beziehungen zwischen den geometrischen Objekten repräsentiert. Dadurch wird ermöglicht, daß zum Beispiel geometrische Flächen als Grenzflächen eines Körpers definiert werden können oder daß Nachbarschaftsbeziehungen von geometrischen Flächen hergestellt werden können. Mit anderen Worten: Die Elemente eines Topologiemodells repräsentieren die *logischen* Beziehungen zwischen den geometrischen Elementen. Typischerweise³ werden als Elemente Knoten, Kante und Oberfläche definiert, die mit den geometrischen Objekten Punkt, Linie und Fläche assoziiert sind. Darüber hinaus werden Elemente für eine Gruppierung der atomaren Elemente zur Verfügung gestellt, wie etwa Körperteile oder Bereiche. Diese Elemente sind meist nur durch Angabe von Mengen repräsentiert. Mit Hilfe der Topologie kann beispielsweise der aus der NC-Technik bekannte Fehler vermieden werden, daß durch Rechenungenauigkeiten des Prozessors ein geschlossenes NC-Programmiersystem einen übertragenen Konturzug als nicht geschlossen erkennt.



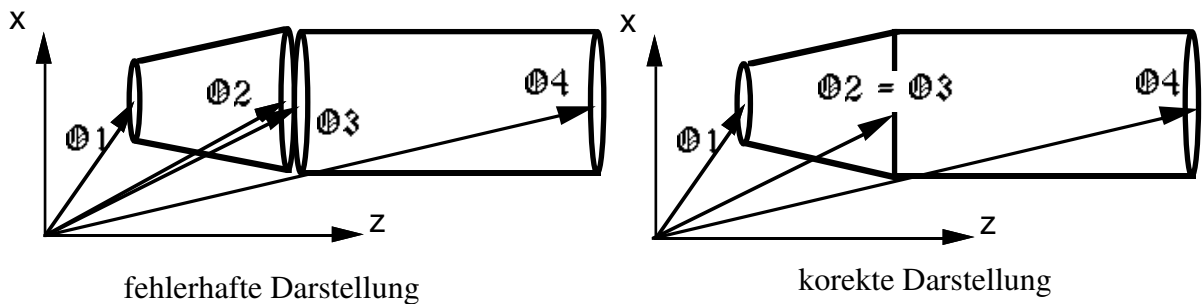
Innerhalb der hier vorgestellten Repräsentation sind die atomaren topologischen Elemente (Knoten, Kante und Oberfläche) in die Darstellung der einzelnen Flächen integriert: Die Flächen sind alle abgeschlossen, daß heißt die Begrenzungen (Kanten und Knoten) der Fläche gehören zum Bereich der Fläche. Topologisch eindeutige Objekte sind auch in der Geometrie eindeutig. Dies rührt vor allem daher, daß die hier vorgestellte Repräsentation immer auf einer allgemeineren Repräsentation, wie etwa in (LITSTEP) beschrieben, aufsetzt, in der Topologie und Geometrie in verschiedenen Modellen dargestellt werden. Bei der Transformation der Daten werden dann zur Erzeugung der in dieser Repräsentation vorgestellten Flächenelemente die geometrischen *und* topologischen Daten verwendet. Dies bedeutet, daß aus den Daten dieser Repräsentation durchaus die topologischen Daten der unterliegenden

³siehe u. a. in "STEP: Überblick über eine zukünftige Schnittstelle zum Produktdatenaustausch", DFKI-Dokument D-90-04, Kap. 2.2.

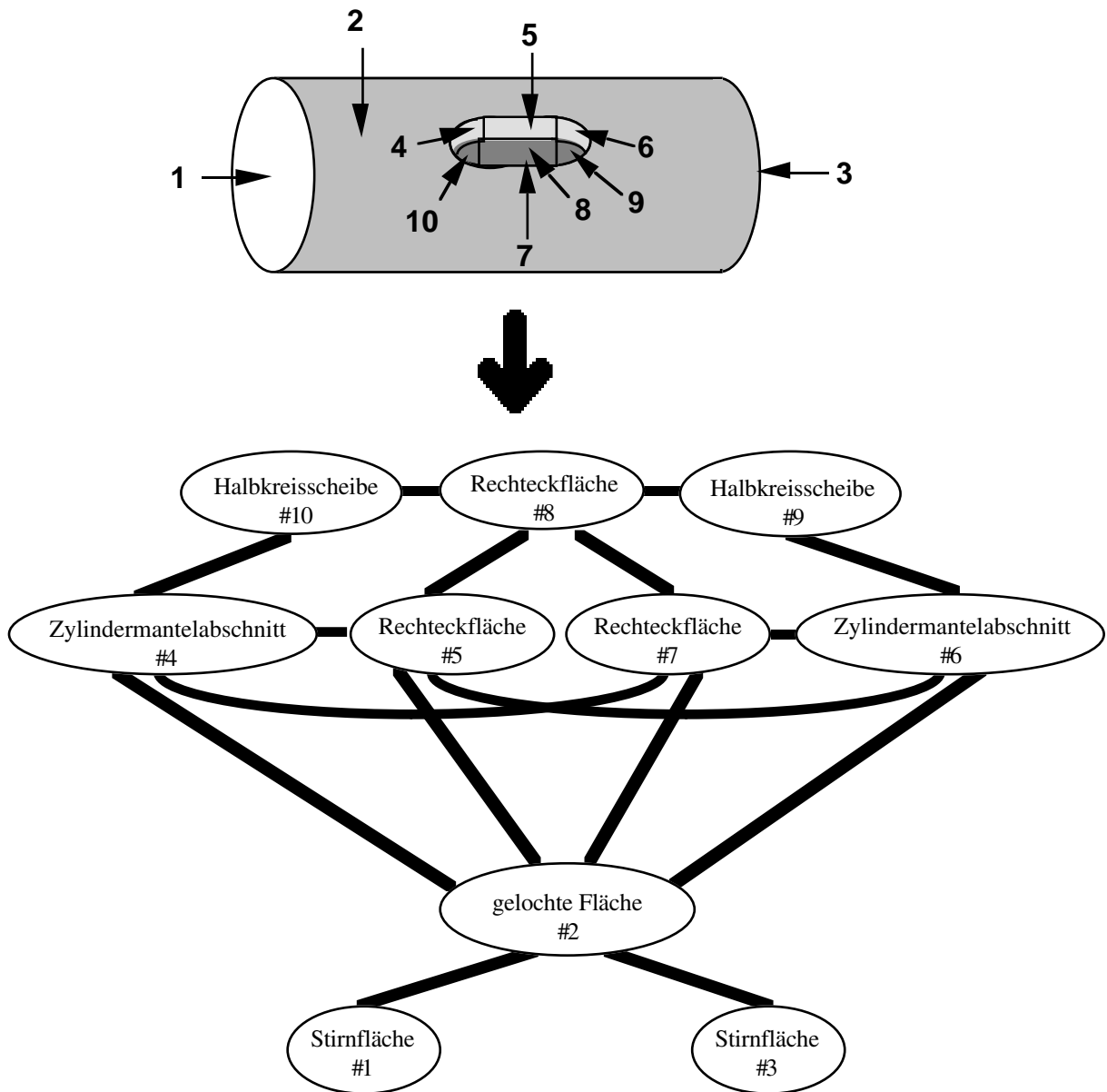
Repräsentation wieder gewonnen werden können, nicht aber die mehrdeutigen geometrischen Daten.



Der nachfolgend abgebildete Fehler kann somit in dieser Repräsentation nicht auftreten.



Die logischen Beziehungen zwischen den Flächen, sprich die *Nachbarschaftsbeziehungen*, werden auch in dieser Repräsentationssprache explizit dargestellt, da diese Informationen zur Erkennung von Features sehr wesentlich sind. Es wird hier aber im Gegensatz etwa zu STEP keine Mengendarstellung gewählt: Aus ihr würden sich direkte Nachbarschaftsbeziehungen bzw. auch indirekte Nachbarschaftsbeziehungen nur nach aufwendiger Betrachtung der geometrischen Daten erkennen lassen; Mengen sagen lediglich aus, welche Flächen einen Bereich bilden, aber nicht wie sie ihn bilden (Anordnung). Daher wird hier eine Darstellung der Nachbarschaftsbeziehungen in einem ungerichteten Graphen (Nachbarschaftsgraphen) vorgenommen: Jeder Knoten repräsentiert eine Fläche und jede Kante repräsentiert eine Nachbarschaftsbeziehung. Es sei an dieser Stelle angemerkt, daß durch die Namensgebung der Flächen, wie etwa *Kreisringfläche*, den einzelnen Knoten des Nachbarschaftsgraphen Sorten zugeordnet werden. Im folgenden ist als Beispiel ein Zylinder mit einer Paßfedernut in der Graphennotation dargestellt.



4.1. Nachbarschaftsbeziehung (Neighbourhood Relation)

Der Nachbarschaftsgraph wird durch eine Menge von Tupeln, die die einzelnen Kanten des Graphen repräsentieren, dargestellt.

**NR(surface_number1: <num>,
surface_number2: <num>)**

Attribute:

surface_number1 Flächennummer
surface_number2 Flächennummer

Konsistenzbedingungen:

Die angegebenen Flächen müssen direkt benachbart sein, sprich sie müssen eine gemeinsame (topologische) Kante haben; ein gemeinsamer Punkt alleine genügt nicht.

5. Ausblick

Ein wesentlicher Punkt der hier vorgestellten Repräsentationssprache ist die Konzentration auf die aus fertigungstechnischer oder auch konstruktionstechnischer Sicht wesentlichen Elemente der Werkstückbeschreibung: Die Oberflächen. Das heißt, im Gegensatz etwa zu STEP, daß keine Ecken, Kanten, lokale Achsensysteme, unendlich große Flächen, etc. explizit repräsentiert werden. Das heißt aber nicht, daß die konkreten geometrischen Daten nicht repräsentiert werden: Diese sind sehr wohl für die Fertigung wie auch für die Konstruktion sehr wesentlich.

Ein weiterer wesentlicher Punkt ist die Einführung von Namen für die Flächen. Damit kann man mit einem Namen, wie etwa *Kreisringfläche*, ein Konstrukt beschreiben, daß in anderen Formalismen mit wesentlich mehr Teilen beschrieben werden muß: In STEP braucht man zur Beschreibung der Kreisringfläche mindestens 6 Elemente!

Diese beiden Punkte sind für die auf dieser Repräsentationssprache aufsetzende *Feature*⁴-Erkennung sehr wichtig. Das Problem der Feature-Erkennung läßt sich in die Graphentheorie übertragen (vgl. /6/, /7/): Das Werkstück wie auch das Feature lassen sich in einem Graphen beschreiben. Die Feature-Erkennung besteht dann aus der Aufgabe, einen isomorphen Subgraphen (das Feature) in dem Graphen des Werkstücks zu finden. Dies ist ein bekanntes NP-vollständiges⁵ Problem (/2/). Dadurch, daß man sich bei der Suche auf die wesentlichen Teile (die Oberflächen) konzentriert und daß man Sorten (die Flächennamen) einführt, besteht die begründete Hoffnung einen Algorithmus zu finden, der sich in polynomialer Zeit ausführen läßt. Die Ergebnisse der in diesem Zusammenhang stattfindenden Untersuchungen werden in dem Papier zur Beschreibung der Repräsentationssprache für Features aufgeführt werden.

Die vorgestellte Repräsentation ist so gewählt, daß eine automatische Übersetzung eines STEP-Datenfiles in diese Repräsentation möglich ist, sofern das Werkstück in den Geometrie-, Topologie- und Toleranzen-Modell repräsentiert ist..

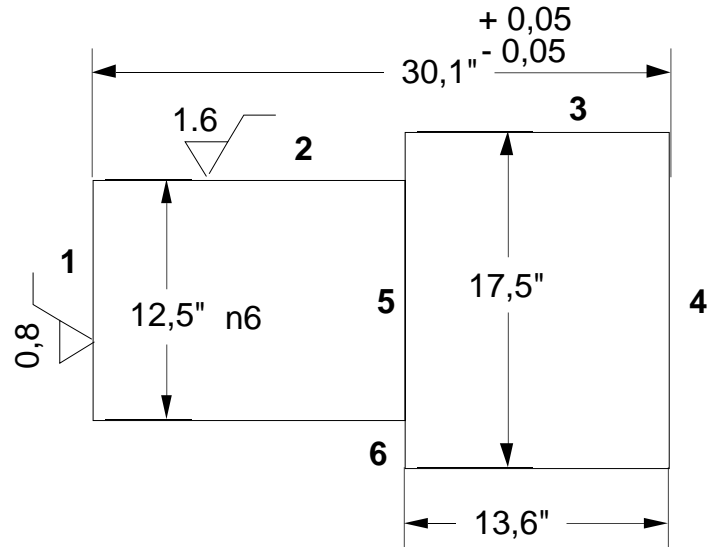
⁴Unter *Feature* kann man allgemein eine Aggregation von Flächen verstehen, die eine gewisse Eigenschaft besitzen.

⁵Ein Problem P heißt *praktisch lösbar*, falls es einen Polynomzeit-Algorithmus für P auf einem deterministischen Rechner (deterministische Turing Maschine) gibt. Ein Problem P heißt *NP-vollständig*, wenn es einen Polynomzeit-Algorithmus für P auf einem nicht deterministischen Rechner (nicht deterministische Turing Maschine) gibt, sprich wenn es nur einen Exponentialzeit-Algorithmus für P auf einem deterministischen Rechner (deterministische Turing Maschine) gibt.

6. Literatur

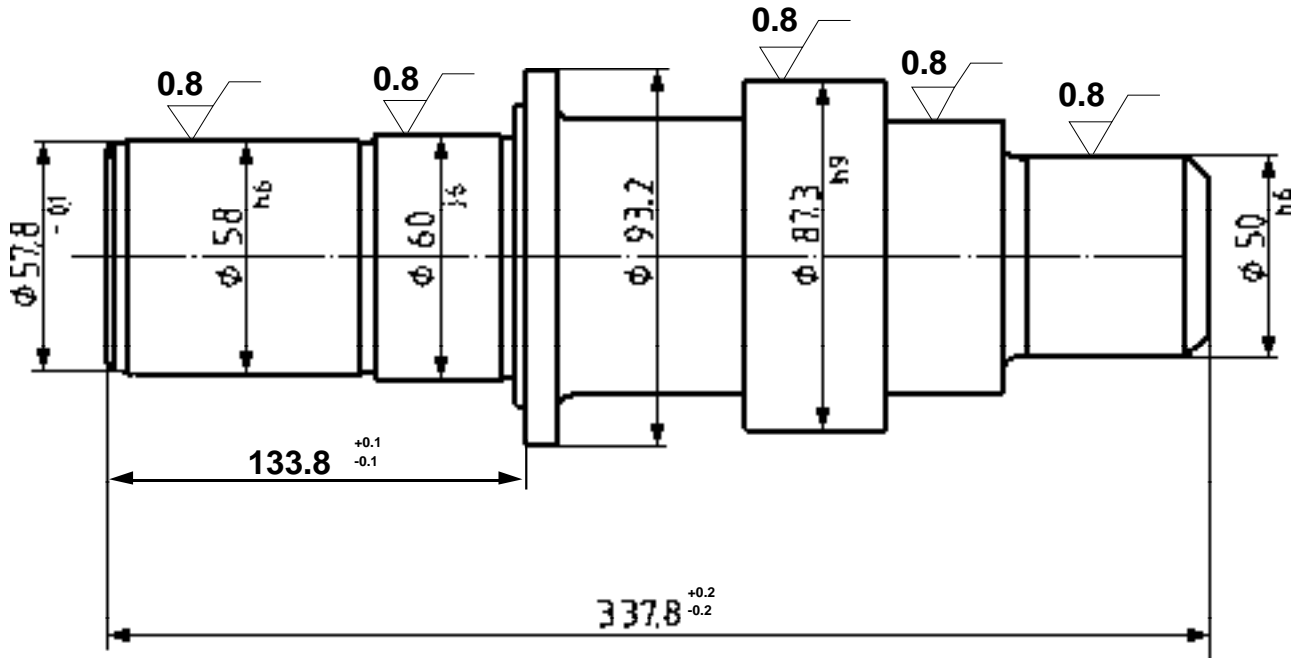
- /1/ O. Abeln: *Die CA.-Techniken in der industriellen Praxis*
München 1990
- /2/ A. V. Aho, J. E. Hopcroft, J. D. Ullman: *The Design and Analysis of Computer Algorithms* Menlo Park 1974
- /3/ A. Bernardi, Ch. Klauck, R. Legleitner: *STEP: Überblick über eine zukünftige Schnittstelle zum Produktdatenaustausch*
Dokument des Deutschen Forschungszentrum für Künstliche Intelligenz GmbH, D-90-04 Kaiserslautern 1990
- /4/ A. Bernardi, Ch. Klauck, R. Legleitner: *Abschlußbericht des Arbeitspaketes PROD*
Dokument des Deutschen Forschungszentrum für Künstliche Intelligenz GmbH, D-90-03 Kaiserslautern 1990
- /5/ S. Finger, J. R. Dixon: *A review of research in mechanical engineering design; part 1* In: *Research in Engineering Design*, Vol. 1 No. 1, pp. 51-67, 1989
- /6/ S. Finger, J. R. Dixon: *A review of research in mechanical engineering design; part 2* In: *Research in Engineering Design*, Vol. 1 No. 1, pp. 121-137, 1989
- /7/ S. Finger, S. A. Safier: *Parsing Features in Solid Geometric Models*
In *Proceedings of ECAI90*, pp. 566-572, 1990
- /8/ H. Grabowski, R. Anderl: *Produktdatenaustausch und CAD-Normteile*
Ehningen 1990
- /9/ J.F. Grätz: *Handbuch der 3D-CAD-Technik*
Berlin 1989
- /10/ H. Hoischen: *Technisches Zeichnen: Grundlagen, Normen, Beispiele, Darstellende Geometrie* Essen 1982
- /11/ G. Pritschow, G. Spur, M. Weck: *Künstliche Intelligenz in der Fertigungstechnik*
München 1989

7. Beispiele einiger Repräsentationen



- @0 := WP(**
 name: Abgesetzte Welle,
 min: 0,
 max: 9,
 unit: mm,
 info: ?)
- @1 := RSC(**
 reference_point: 0.0,
 radius: 6.75,
 direction_of_material: +)
- @2 := RSCJ(**
 reference_point1: 0.0,
 reference_point2: 16.5,
 radius: 6.75,
 direction_of_material: -)
- @3 := RSR(**
 reference_point: 16.5,
 radius1: 8.75,
 radius2: 6.75,
 direction_of_material: +)
- @4 := RSCJ(**
 reference_point1: 16.5,
 reference_point2: 30.1,
 radius: 8.75,
 direction_of_material: -)
- @5 := RSC(**
 reference_point: 30.1,
 radius: 8.75,
 direction_of_material: -)
- @6 :=SSD(**
 surface_number1: 1,
 surface_number2: 4,
 nominal_size: 30.1,
 min_size: 30.05,
 max_size: 30.15)
- @7 := ISOF(**
 surface_number: 2,
 tolerance_extent: n,
 clearance_space: 6)
- @8 := SF(**
 surface_number: 1,
 surface_finish: Ra,
 value: 0.8)

@9 := SF(
 surface_number: 2,
 surface_finish: Ra,
 value: 1.6)



@0 := WP(
 name: Antriebswelle,
 min: 0,
 max: 41,
 unit: mm,
 info: ?)

@1 := RSC(
 reference_point: 0.0,
 radius: 26.4,
 direction_of_material: +)

@2 := RSEC(
 reference_point1: 0.0,
 reference_point2: 2.5,
 radius1: 26.4,
 radius2: 28.9,
 direction_of_material: -)

@3 := RSCJ(
 reference_point1: 2.5,

reference_point2: 7.5,
 radius: 28.9,
 direction_of_material: -)

@4 := RSR(
 reference_point: 7.5,
 radius1: 29.0,
 radius2: 28.9,
 direction_of_material: +)

@5 := RSCJ(
 reference_point1: 7.5,
 reference_point2: 80.5,
 radius: 29.0,
 direction_of_material: -)

@6 := RSR(
 reference_point: 80.5,
 radius1: 29.0,
 radius2: 28.9,
 direction_of_material: -)

@7 := RSCJ(
 reference_point1: 80.5,

```

reference_point1: 80.5,
reference_point2: 85.5,
radius: 28.9,
direction_of_material: -)
@8 := RSR(
reference_point: 85.5,
radius1: 30.0,
radius2: 28.9,
direction_of_material: +)
@9 := RSCJ(
reference_point1: 85.5,
reference_point2: 124.5,
radius: 30.0,
direction_of_material: -)
@10 := RSR(
reference_point: 124.5,
radius1: 30.0,
radius2: 28.9,
direction_of_material: -)
@11 := RSCJ(
reference_point1: 124.5,
reference_point2: 129.5,
radius: 28.9,
direction_of_material: -)
@12 := RSR(
reference_point: 129.5,
radius1: 37.0,
radius2: 28.9,
direction_of_material: +)
@13 := RSCJ(
reference_point1: 129.5,
reference_point2: 133.8,
radius: 37.0,
direction_of_material: -)
@14 := RSR(
reference_point: 133.8,
radius1: 46.6,
radius2: 37.0,
direction_of_material: +)
@15 := RSCJ(
reference_point1: 133.8,
reference_point2: 143.8,
radius: 46.6,
direction_of_material: -)
@16 := RSR(
reference_point: 143.8,
radius1: 46.6,
radius2: 37.5,
direction_of_material: -)
@17 := RSTS(
reference_point1: 143.8,
reference_point2: 148.8,
reference_point3: 148.8,
radius1: 37.5,
radius2: 32.5,
radius3: 32.5,
direction_of_arch: v,
direction_of_material: -)
@18 := RSCJ(
reference_point1: 148.8,
reference_point2: 198.8,
radius: 32.5,
direction_of_material: -)
@19 := RSR(
reference_point: 198.8,
radius1: 43.65,
radius2: 32.5,
direction_of_material: +)
@20 := RSCJ(
reference_point1: 198.8,
reference_point2: 239.8,
radius: 43.65,
direction_of_material: -)

```

```

@21 := RSR(
    reference_point: 239.8,
    radius1: 43.65,
    radius2: 32.5,
    direction_of_material: -)
@22 := RSCJ(
    reference_point1: 239.8,
    reference_point2: 275.8,
    radius: 32.5,
    direction_of_material: -)
@23 := RSR(
    reference_point: 275.8,
    radius1: 32.5,
    radius2: 32.0,
    direction_of_material: -)
@24 := RSTS(
    reference_point1: 275.8,
    reference_point2: 282.8,
    reference_point3: 282.8,
    radius1: 32.0,
    radius2: 25.0,
    radius3: 25.0,
    direction_of_arch: v,
    direction_of_material: -)
@25 := RSCJ(
    reference_point1: 282.8,
    reference_point2: 330.8,
    radius: 25.0,
    direction_of_material: -)
@26 := RSEC(
    reference_point1: 330.8,
    reference_point2: 337.8,
    radius1: 25.0,
    radius2: 18.0,
    direction_of_material: -)
@27 := RSC(
    reference_point: 337.8,
    radius: 18.0,
    direction_of_material: -)
@28 := SS (
    surface_number: 3,
    nominal_size: 28.9,
    min_size: 28.85,
    max_size: 28.9)
@29 := SS (
    surface_number: 7,
    nominal_size: 28.9,
    min_size: 28.85,
    max_size: 28.9)
@30 := SS (
    surface_number: 11,
    nominal_size: 28.9,
    min_size: 28.85,
    max_size: 28.9)
@31 := SSD(
    surface_number1: 1,
    surface_number2:14,
    nominal_size: 133.8,
    min_size: 133.7,
    max_size:
    133.9)
@32 := SSD(
    surface_number1: 1,
    surface_number2:27,
    nominal_size: 337.8,
    min_size: 337.6,
    max_size: 338.0)
@33 := ISOF (
    surface_number: 5,
    tolerance_extent: h,
    clearance_space: 6)
@34 := ISOF (
    surface_number: 9,
    tolerance_extent: j,
    clearance_space: 6)

```


@35 := ISOF (
 surface_number: 20,
 tolerance_extent: h,
 clearance_space: 9)

@36 := ISOF (
 surface_number: 25,
 tolerance_extent: h,
 clearance_space: 6)

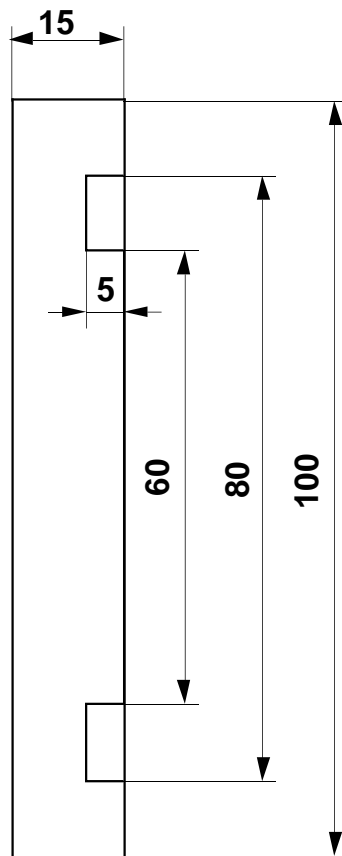
@37 := SF(
 surface_number: 5,
 surface_finish: Ra,
 value: 0.8)

@38 := SF(
 surface_number: 9,
 surface_finish: Ra,
 value: 0.8)

@39 := SF(
 surface_number: 20,
 surface_finish: Ra,
 value: 0.8)

@40 := SF(
 surface_number: 22,
 surface_finish: Ra,
 value: 0.8)

@41 := SF(
 surface_number: 25,
 surface_finish: Ra,
 value: 0.8)



```

@0 := WP(
    name: Flansch,
    min: 0,
    max: 7,
    unit: mm,
    info: ?)
@1 := RSC(
    reference_point: 0.0,
    radius: 50.0,
    direction_of_material: +)
@2 := RSCJ(
    reference_point1: 0.0,
    reference_point2: 15.0,
    radius: 50.0,
    direction_of_material: -)
@3 := RSR(
    reference_point: 15.0,
    radius1: 50.0
    radius2: 40.0
    direction_of_material: -)
@4 := RSCJ(
    reference_point1: 10.0,
    reference_point2: 15.0,
    radius: 40.0,
    direction_of_material: +)
@5 := RSR(
    reference_point: 10.0,
    radius1: 40.0
    radius2: 30.0
    direction_of_material: -)
@6 := RSCJ(
    reference_point1: 10.0,
    reference_point2: 15.0,
    radius: 30.0,
    direction_of_material: -)
@7 := RSC(
    reference_point: 15.0,
    radius: 30.0,
    direction_of_material: -)

```

DFKI Publikationen

Die folgenden DFKI Veröffentlichungen sowie die aktuelle Liste von allen bisher erschienenen Publikationen können von der oben angegebenen Adresse oder per anonymem ftp von ftp.dfki.uni-kl.de (131.246.241.100) unter pub/Publications bezogen werden.

Die Berichte werden, wenn nicht anders gekennzeichnet, kostenlos abgegeben.

DFKI Research Reports

RR-93-04

Christoph Klauck, Johannes Schwagereit:
GGD: Graph Grammar Developer for features in CAD/CAM
13 pages

RR-93-05

Franz Baader, Klaus Schulz: Combination Techniques and Decision Problems for Disunification
29 pages

RR-93-06

Hans-Jürgen Bürckert, Bernhard Hollunder, Armin Laux: On Skolemization in Constrained Logics
40 pages

RR-93-07

Hans-Jürgen Bürckert, Bernhard Hollunder, Armin Laux: Concept Logics with Function Symbols
36 pages

RR-93-08

Harold Boley, Philipp Hanschke, Knut Hinkelmann, Manfred Meyer: COLAB: A Hybrid Knowledge Representation and Compilation Laboratory
64 pages

RR-93-09

Philipp Hanschke, Jörg Würtz:
Satisfiability of the Smallest Binary Program
8 pages

RR-93-10

Martin Buchheit, Francesco M. Donini, Andrea Schaefer: Decidable Reasoning in Terminological Knowledge Representation Systems
35 pages

DFKI Publications

The following DFKI publications or the list of all published papers so far are obtainable from the above address or via anonymous ftp from ftp.dfki.uni-kl.de (131.246.241.100) under pub/Publications.

The reports are distributed free of charge except if otherwise indicated.

RR-93-11

Bernhard Nebel, Hans-Juergen Buerckert:
Reasoning about Temporal Relations:
A Maximal Tractable Subclass of Allen's
Interval Algebra
28 pages

RR-93-12

Pierre Sablayrolles: A Two-Level Semantics for French Expressions of Motion
51 pages

RR-93-13

Franz Baader, Karl Schlechta:
A Semantics for Open Normal Defaults via a Modified Preferential Approach
25 pages

RR-93-14

Joachim Niehren, Andreas Podelski, Ralf Treinen: Equational and Membership Constraints for Infinite Trees
33 pages

RR-93-15

Frank Berger, Thomas Fehrle, Kristof Klöckner, Volker Schölles, Markus A. Thies, Wolfgang Wahlster: PLUS - Plan-based User Support
Final Project Report
33 pages

RR-93-16

Gert Smolka, Martin Henz, Jörg Würtz: Object-Oriented Concurrent Constraint Programming in Oz
17 pages

RR-93-17

Rolf Backofen:
Regular Path Expressions in Feature Logic
37 pages

RR-93-18

Klaus Schild: Terminological Cycles and the Propositional m-Calculus
32 pages

RR-93-20

Franz Baader, Bernhard Hollunder:
Embedding Defaults into Terminological
Knowledge Representation Formalisms
34 pages

RR-93-22

Manfred Meyer, Jörg Müller:
Weak Looking-Ahead and its Application in
Computer-Aided Process Planning
17 pages

RR-93-23

Andreas Dengel, Ottmar Lutz:
Comparative Study of Connectionist Simulators
20 pages

RR-93-24

Rainer Hoch, Andreas Dengel:
Document Highlighting —
Message Classification in Printed Business
Letters
17 pages

RR-93-25

Klaus Fischer, Norbert Kuhn: A DAI Approach
to Modeling the Transportation Domain
93 pages

RR-93-26

Jörg P. Müller, Markus Pischel: The Agent
Architecture InteRRaP: Concept and
Application
99 pages

RR-93-27

Hans-Ulrich Krieger:
Derivation Without Lexical Rules
33 pages

RR-93-28

*Hans-Ulrich Krieger, John Nerbonne,
Hannes Pirker:* Feature-Based Allomorphy
8 pages

RR-93-29

Armin Laux: Representing Belief in Multi-
Agent Worlds via Terminological Logics
35 pages

RR-93-30

Stephen P. Spackman, Elizabeth A. Hinkelman:
Corporate Agents
14 pages

RR-93-31

Elizabeth A. Hinkelman, Stephen P. Spackman:
Abductive Speech Act Recognition, Corporate
Agents and the COSMA System
34 pages

RR-93-32

David R. Traum, Elizabeth A. Hinkelman:
Conversation Acts in Task-Oriented Spoken
Dialogue
28 pages

RR-93-33

Bernhard Nebel, Jana Koehler:
Plan Reuse versus Plan Generation: A
Theoretical and Empirical Analysis
33 pages

RR-93-34

Wolfgang Wahlster:
Verbmobil Translation of Face-To-Face Dialogs
10 pages

RR-93-35

*Harold Boley, François Bry, Ulrich Geske
(Eds.):* Neuere Entwicklungen der deklarativen
KI-Programmierung — *Proceedings*
150 Seiten
Note: This document is available only for a
nominal charge of 25 DM (or 15 US-\$).

RR-93-36

*Michael M. Richter, Bernd Bachmann, Ansgar
Bernardi, Christoph Klauck, Ralf Legleitner,
Gabriele Schmidt:* Von IDA bis IMCOD:
Expertensysteme im CIM-Umfeld
13 Seiten

RR-93-38

Stephan Baumann: Document Recognition of
Printed Scores and Transformation into MIDI
24 pages

RR-93-40

*Francesco M. Donini, Maurizio Lenzerini,
Daniele Nardi, Werner Nutt, Andrea Schaefer:*
Queries, Rules and Definitions as Epistemic
Statements in Concept Languages
23 pages

RR-93-41

Winfried H. Graf: LAYLAB: A Constraint-
Based Layout Manager for Multimedia
Presentations
9 pages

RR-93-42

Hubert Comon, Ralf Treinen:
The First-Order Theory of Lexicographic Path
Orderings is Undecidable
9 pages

RR-93-43

M. Bauer, G. Paul: Logic-based Plan
Recognition for Intelligent Help Systems
15 pages

RR-93-44

*Martin Buchheit, Manfred A. Jeusfeld, Werner
Nutt, Martin Staudt:* Subsumption between
Queries to Object-Oriented Databases
36 pages

RR-93-45

Rainer Hoch: On Virtual Partitioning of Large
Dictionaries for Contextual Post-Processing to
Improve Character Recognition
21 pages

RR-93-46

Philipp Hanschke: A Declarative Integration of
Terminological, Constraint-based, Data-driven,
and Goal-directed Reasoning
81 pages

RR-93-48

*Franz Baader, Martin Buchheit, Bernhard
Hollunder:* Cardinality Restrictions on Concepts
20 pages

RR-94-01

Elisabeth André, Thomas Rist:
Multimedia Presentations:
The Support of Passive and Active Viewing
15 pages

RR-94-02

Elisabeth André, Thomas Rist:
Von Textgeneratoren zu Intellimedia-
Präsentationssystemen
22 pages

RR-94-03

Gert Smolka:
A Calculus for Higher-Order Concurrent
Constraint Programming with Deep Guards
34 pages

RR-94-05

*Franz Schmalhofer,
J. Stuart Aitken, Lyle E. Bourne jr.:*
Beyond the Knowledge Level: Descriptions of
Rational Behavior for Sharing and Reuse
81 pages

RR-94-07

Harold Boley: Finite Domains and Exclusions as
First-Class Citizens
25 pages

RR-94-08

Otto Kühn, Björn Höfling: Conserving
Corporate Knowledge for Crankshaft Design
17 pages

RR-94-10

Knut Hinkelmann, Helge Hintze:
Computing Cost Estimates for Proof Strategies
22 pages

RR-94-11

Knut Hinkelmann: A Consequence Finding
Approach for Feature Recognition in CAPP
18 pages

RR-94-12

Hubert Comon, Ralf Treinen:
Ordering Constraints on Trees
34 pages

DFKI Technical Memos**TM-92-02**

Achim Schupeta: Organizing Communication
and Introspection in a Multi-Agent Blocksworld
32 pages

TM-92-03

Mona Singh:
A Cognitive Analysis of Event Structure
21 pages

TM-92-04

*Jürgen Müller, Jörg Müller, Markus Pischel,
Ralf Scheidhauer:*
On the Representation of Temporal Knowledge
61 pages

TM-92-05

*Franz Schmalhofer, Christoph Globig, Jörg
Thoben:*
The refitting of plans by a human expert
10 pages

TM-92-06

Otto Kühn, Franz Schmalhofer: Hierarchical
skeletal plan refinement: Task- and inference
structures
14 pages

TM-92-08

Anne Kilger: Realization of Tree Adjoining
Grammars with Unification
27 pages

TM-93-01

Otto Kühn, Andreas Birk: Reconstructive
Integrated Explanation of Lathe Production
Plans
20 pages

TM-93-02

Pierre Sablayrolles, Achim Schupeta:
Conflict Resolving Negotiation for COoperative
Schedule Management
21 pages

TM-93-03

*Harold Boley, Ulrich Buhrmann, Christof
Kremer:*
Konzeption einer deklarativen Wissensbasis
über recyclingrelevante Materialien
11 pages

TM-93-04

Hans-Günther Hein: Propagation Techniques in
WAM-based Architectures — The FIDO-III
Approach
105 pages

TM-93-05

Michael Sintek: Indexing PROLOG Procedures
into DAGs by Heuristic Classification
64 pages

DFKI Documents

D-93-03

Stephan Busemann, Karin Harbusch(Eds.):
DFKI Workshop on Natural Language Systems:
Reusability and Modularity - Proceedings
74 pages

D-93-04

DFKI Wissenschaftlich-Technischer
Jahresbericht 1992
194 Seiten

D-93-05

*Elisabeth André, Winfried Graf, Jochen
Heinsohn, Bernhard Nebel, Hans-Jürgen
Profilich, Thomas Rist, Wolfgang Wahlster:*
PPP: Personalized Plan-Based Presenter
70 pages

D-93-06

Jürgen Müller (Hrsg.):
Beiträge zum Gründungsworkshop der
Fachgruppe Verteilte Künstliche Intelligenz,
Saarbrücken, 29. - 30. April 1993
235 Seiten

Note: This document is available only for a
nominal charge of 25 DM (or 15 US-\$).

D-93-07

Klaus-Peter Gores, Rainer Bleisinger:
Ein erwartungsgesteuerter Koordinator zur
partiellen Textanalyse
53 Seiten

D-93-08

Thomas Kieninger, Rainer Hoch:
Ein Generator mit Anfragesystem für
strukturierte Wörterbücher zur Unterstützung
von Texterkennung und Textanalyse
125 Seiten

D-93-09

Hans-Ulrich Krieger, Ulrich Schäfer:
TDL ExtraLight User's Guide
35 pages

D-93-10

*Elizabeth Hinkelman, Markus
Vonderden, Christoph Jung:* Natural Language
Software Registry
(Second Edition)
174 pages

D-93-11

Knut Hinkelmann, Armin Laux (Eds.):
DFKI Workshop on Knowledge Representation
Techniques — Proceedings
88 pages

D-93-12

*Harold Boley, Klaus Elsbernd,
Michael Herfert, Michael Sintek, Werner Stein:*
RELFUN Guide: Programming with Relations
and Functions Made Easy
86 pages

D-93-14

Manfred Meyer (Ed.): Constraint Processing –
Proceedings of the International Workshop at
CSAM'93, July 20-21, 1993
264 pages

Note: This document is available only for a
nominal charge of 25 DM (or 15 US-\$).

D-93-15

Robert Laux: Untersuchung maschineller
Lernverfahren und heuristischer Methoden im
Hinblick auf deren Kombination zur
Unterstützung eines Chart-Parsers
86 Seiten

D-93-16

*Bernd Bachmann, Ansgar Bernardi, Christoph
Klauck, Gabriele Schmidt:* Design & KI
74 Seiten

D-93-20

Bernhard Herbig:
Eine homogene Implementierungsebene für
einen hybriden
Wissensrepräsentationsformalismus
97 Seiten

D-93-21

Dennis Drollinger:
Intelligentes Backtracking in Inferenzsystemen
am Beispiel Terminologischer Logiken
53 Seiten

D-93-22

Andreas Abecker: Implementierung graphischer
Benutzungsoberflächen mit Tcl/Tk und
Common Lisp
44 Seiten

D-93-24

Brigitte Krenn, Martin Volk:
DiTo-Datenbank: Datendokumentation zu
Funktionsverbgefügen und Relativsätzen
66 Seiten

D-93-25

Hans-Jürgen Bürckert, Werner Nutt (Eds.):
Modeling Epistemic Propositions
118 pages

Note: This document is available only for a
nominal charge of 25 DM (or 15 US-\$).

D-93-26

Frank Peters:
Unterstützung des Experten bei der
Formalisierung von Textwissen
INFOCOM - Eine interaktive
Formalisierungskomponente
58 Seiten

D-94-01

Josua Boon (Ed.):
DFKI-Publications: The First Four Years
1990 - 1993
75 pages