



**Deutsches  
Forschungszentrum  
für Künstliche  
Intelligenz GmbH**

**Document**  
D-91-18

**Definition von Problemklassen  
im Maschinenbau als eine  
Begriffsbildungsaufgabe**

**Thomas Reinartz**

**Dezember 1991**

**Deutsches Forschungszentrum für Künstliche Intelligenz  
GmbH**

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

Stuhlsatzenhausweg 3  
D-6600 Saarbrücken 11  
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. The shareholder companies are Daimler Benz, Fraunhofer Gesellschaft, GMD, IBM, Insiders, Krupp-Atlas, Mannesmann-Kienzle, Philips, Sema Group Systems, Siemens and Siemens-Nixdorf. 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

# **Definition von Problemklassen im Maschinenbau als eine Begriffsbildungsaufgabe**

**Thomas Reinartz**

DFKI-D-91-18



Universität Kaiserslautern als Projektarbeit im ARC-TEC-Projekt am Deutschen Forschungszentrum für Künstliche Intelligenz unter der Betreuung von Prof. M.M. Richter und Dr. Franz Schmalhofer erstellt.

Diese Arbeit wurde finanziell unterstützt durch das Bundesministerium für Forschung und Technologie (FKZ ITW-8902 C4).



**Definition von Problemklassen  
im Maschinenbau als eine  
Begriffsbildungsaufgabe**

Thomas Reinartz

Deutsches Forschungszentrum für Künstliche Intelligenz  
Erwin-Schrödinger-Str.  
D-6750 Kaiserslautern

e-mail: [reinartz@dfki.uni-kl.de](mailto:reinartz@dfki.uni-kl.de)

*It is better to travel hopefully than to arrive*

*japanisches Sprichwort*



# INHALTSVERZEICHNIS

<b>I</b>	<b>Zusammenfassung</b>	<b>5</b>
<b>II</b>	<b>Kontext und Domäne</b>	<b>6</b>
	II.1 ARC-TEC-Projekt	6
	II.2 Fertigung rotationssymmetrischer Drehteile	7
<b>III</b>	<b>Begriffsbildung</b>	<b>9</b>
	III.1 Begriffsbildung in der Kognition	9
	III.1.1 Phänomene	9
	III.1.2 Erklärungsansätze	13
	III.2 Begriffsbildung im Maschinenbau	16
	III.2.1 Experte und Neuling	16
	III.2.2 Phänomene	17
	III.2.3 Erklärungsansätze	19
<b>IV</b>	<b>Case Experience Combination System (CECoS)</b>	<b>21</b>
	IV.1 Überblick	21
	IV.2 Episoden	21
	IV.2.1 Erklärungsepisode	21
	IV.2.2 Wissensvergleichsepisode	24
	IV.2.3 Kompetenzabschätzungsepisode	25
	IV.2.4 Formalisierungsepisode	25
<b>V</b>	<b>Die Anwendung: 60 Fälle</b>	<b>29</b>

	V.1.1 Geometrien	29
	V.1.2 Werkstoffe	29
	V.1.3 Maschinen	30
	V.1.4 Matrix	30
	V.2 Episoden	31
	V.2.1 Erklärungsepisode	31
	V.2.2 Wissensvergleichsepisode	37
	V.2.3 Kompetenzabschätzungsepisode	37
	V.2.4 Formalisierungsepisode	37
<b>VI</b>	<b>Implementierung</b>	<b>42</b>
	VI.1 TAXON	42
	VI.2 Anforderungen	42
	VI.3 Probleme	42
	VI.3.1 Einschränkungen der konkreten Domäne	42
	VI.3.2 Disjunktionen	43
	VI.3.3 Listen undefinierter Länge	43
	VI.3.4 Rollen und Allquantifizierung	43
	VI.3.5 Attributpfade in der ABox	44
	VI.4 Ursachen	45
	VI.4.1 Vollständigkeit und Korrektheit	45
	VI.4.2 Bottom-Up versus Top-Down	45
	VI.4.3 Ineffiziente Algorithmen	45
	VI.4.4 Komplexe Anwendung	45
	VI.4.5 Hohe Anforderungen	45
	VI.5 Alternativen	46

VI.5.3	Andere Repräsentation	46
VI.5.4	Kombination mit einer Regelsprache	46
<b>VII</b>	<b>Diskussion</b>	<b>47</b>
VII.1	Zweck der Problemklassen	47
VII.2	Zuordnung neuer Fertigungsprobleme	47
VII.3	Einschätzung der Anwendbarkeit	47
<b>VIII</b>	<b>Danksagung</b>	<b>49</b>
<b>IX</b>	<b>Literatur</b>	<b>50</b>
<b>X</b>	<b>Anhang</b>	<b>55</b>
X.1	Erläuterungen	55
X.2	TBox	58
X.3	ABox	72
X.4	Beispiele	105
X.4.1	Mögliche Inferenzanfragen	105
X.4.2	Konkrete Beispiele möglicher Inferenzanfragen	105
X.4.3	Beurteilung	106

## I ZUSAMMENFASSUNG

Problemklassen stellen ein Schema zur Einteilung von Problemstellungen dar [Bauer, Stein, Weiner, 1991]. Ein Beispiel möglicher Problemstellungen stammt aus dem Bereich des Maschinenbaus: die Fertigung rotations-symmetrischer Drehteile. Die zu den Fertigungsproblemstellungen gehörenden Problemklassen können als Begriff verstanden werden, die Experten im Bereich ihrer Expertise bilden. Die Entwicklung der Problemklassen stellt somit eine Begriffsbildungsaufgabe dar. Gemeinsam mit den Begriffen der Problemklassen werden allgemeine Lösungsstrategien, die auf theoretischem Wissen der Domäne basieren, entwickelt und mit dem zugehörigen Begriff assoziiert. Die Begriffe eines Experten der Domäne sind anderen überlegen, weil sie einen großen Anteil qualitativen Wissens beinhalten. Die Lösung neuer Fertigungsprobleme wird durch die Repräsentation der Begriffe geleitet [Chi, Feltovich, Glaser, 1981].

Bei der Entwicklung eines Expertensystems muß eine Wissensbasis konstruiert werden. Diese Wissensbasis enthält unter anderem die Problemklassen, die formal repräsentiert werden müssen.

Eine Aufgabe der Wissensakquisition besteht in der Erhebung der Begriffe, die zu den Problemklassen gebildet sind. Dazu betrachtet man Ergebnisse der allgemeinen menschlichen Begriffsbildung aus der Kognition und versucht diese auf den Bereich des Maschinenbaus zu übertragen. Ausgehend von diesen Ergebnissen kann eine Wissensakquisitionsmethode entwickelt werden, in der die Begriffe des Experten erhoben werden. Die Erhebung dient nicht nur der Aufzeichnung der Begriffe, sondern einer Strukturierung und einer aufbereiteten Repräsentation. Die aufbereitete Repräsentation kann durch eine schrittweise Formalisierung in eine formale Wissensbasis übersetzt werden, die ein Expertensystem automatisch verwenden kann.

Im Wissensakquisitionsteil des ARC-TEC-Projektes am Deutschen Forschungszentrum für Künstliche Intelligenz ist eine integrierte Wissensakquisitionsmethode entwickelt worden, die neben einer Erhebung der Problemklassen eine frühe Wissensverifikation, in der die Relevanz, Suffizienz, Redundanz, und Konsistenz des Wissens bereits auf einer informalen Ebene beurteilt werden kann, ermöglicht. Dabei dient das Wissensakquisitionswerkzeug CECoS der Erhebung der Begriffe des Experten [Schmalhofer, Kühn, Schmidt, 1991].

Das in CECoS realisierte Verfahren wird im Zusammenhang mit der Begriffsbildungsaufgabe der Problemklassen diskutiert und erweitert. Eine Beispielanwendung ausgehend von 60 typischen Fertigungsproblemen bis hin zu Implementierungsversuchen der formalen Repräsentation der Problemklassen in einem taxonomischen Wissensrepräsentationssystem soll die Methoden der integrierten Wissensakquisition überprüfen und beurteilen.



## II KONTEXT UND DOMÄNE

### II.1 ARC-TEC-Projekt

Im ARC-TEC-Projekt (Akquisition, Repräsentation und Compilation von TECHNISCHEM WISSEN) am Deutschen Forschungszentrum für Künstliche Intelligenz wird eine Expertensystemshell entwickelt, die in der Domäne des Maschinenbaus angewendet werden soll. Das ARC-TEC-Projekt versucht dabei die Idee des 'Computer-Integrated-Manufacturing' (CIM) mit Ansätzen aus der Künstlichen Intelligenz zu verbinden. Neben konzeptuellen Lösungen sollen Software-Werkzeuge zur Akquisition, Repräsentation und Compilation von technischem Wissen erstellt werden. Die Ergebnisse sind in [1] dargestellt.

Mit dem System CECoS wird basierend auf Ähnlichkeitsurteilen, die paarweise zu Operatoren und Problemen von dem Experten abgegeben werden, eine hierarchische Klassifizierung durchgeführt. Die daraus resultierenden informellen Klassen von Operatoren und Problemen werden in einem weiteren Schritt zu formalen Definitionen der Operator- und Problemklassen erweitert.

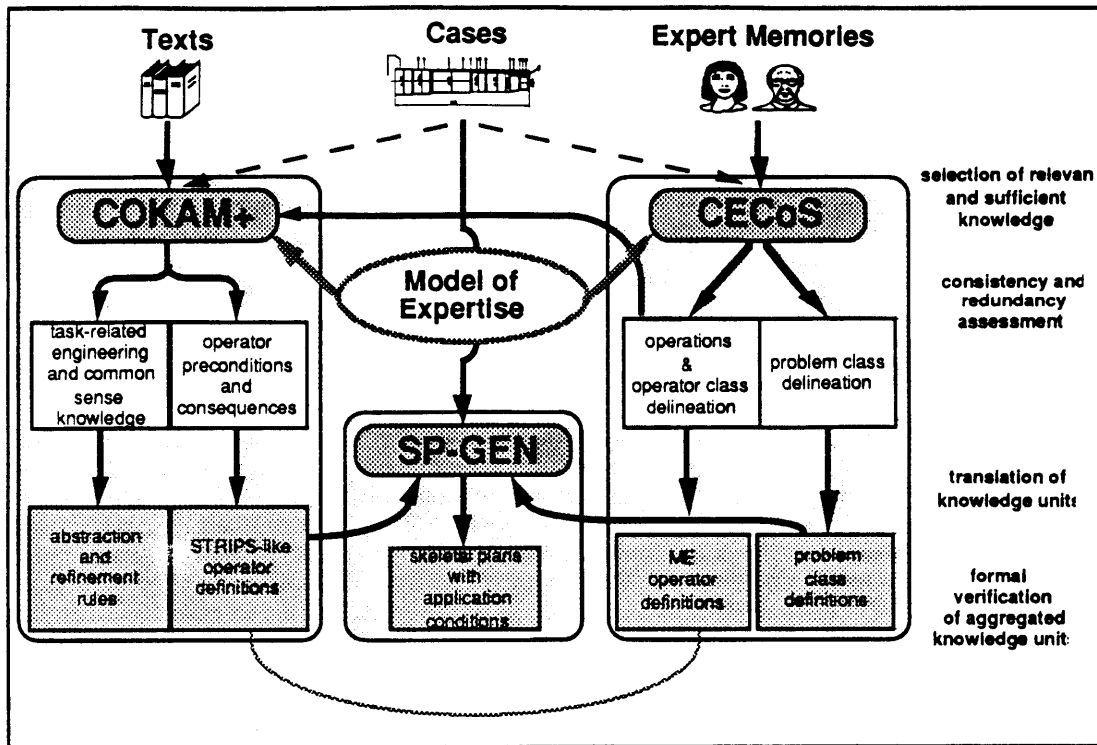


Abbildung 1: Integrierte Wissensakquisitionsmethode und Koordination der einzelnen Werkzeuge (aus: [Schmalhofer, Reinartz, 1991])

SP-GEN erzeugt nun mit Hilfe der durch COKAM+ und CECoS definierten formalen Beschreibungen, die einerseits als Domänentheorie und andererseits als Operationalitätskriterien verwendet werden, ausgehend von bereits früher bearbeiteten Problemen durch eine EBL-artige [Mitchell, Keller, Kedar-Cabelli, 1986] Prozedur einen mehr oder weniger generellen Skelettplan, der einer der mit CECoS ermittelten Problemklassen zugeordnet werden kann. Nacheinander simuliert und erklärt SP-GEN zunächst einen konkreten Produktionsplan, abstrahiert unter Berücksichtigung der Domänentheorie und der Operationalitätskriterien die einzelnen Schritte des Planes, analysiert die Abhängigkeiten zwischen den Operationen, und normalisiert den sich daraus ergebenden Skelettplan.

Da die Problemklassen in CECoS aufgrund von Urteilen eines Experten entstanden sind, kann angenommen werden, daß sich die Klassen auf einem nützlichen Grad der Generalität befinden, d.h. daß sie einerseits generell genug sind, so daß hinreichend viele neue Probleme in die Klassen fallen, und andererseits speziell genug, daß sich mit SP-GEN ein sinnvoller Skelettplan generieren läßt. In Zukunft können neue Probleme den einzelnen Klassen zugeordnet, und der mit dieser Klasse assoziierte Skelettplan zu einem konkreten Produktionsplan verfeinert werden.

## II.2 Fertigung rotationssymmetrischer Drehteile

Im Maschinenbau unterscheidet man sechs verschiedene Hauptgruppen in der Fertigung [DIN 8580]. Eine der Hauptgruppen ist neben Urformen, Umformen, Fügen, Beschichten und Stoffeigenschaft ändern, das Trennen. Die Hauptgruppe des Trennens ist ihrerseits in die Gruppen Zerteilen, Spanen mit geometrisch bestimmten Schneiden, Spanen mit geometrisch unbestimmten Schneiden, Abtragen, Zerlegen, Reinigen und Evakuieren untergliedert [DIN 8589]. Im Gegensatz zum Spanen mit geometrisch unbestimmten Schneiden, bei dem ein Werkzeug verwendet wird, dessen Schneidanzahl, Geometrie der Schneidkeile, und Lage der Schneiden zum Werkstück unbestimmt sind, werden diese Größen beim Spanen mit geometrisch bestimmten Schneiden festgelegt. Untergruppen dieses Spanens mit geometrisch bestimmten Schneiden

sind Drehen, Bohren / Senken / Reiben, Fräsen, Hobeln / Stoßen, Räumen, Sägen, Feilen / Raspeln, Bürstspanen und Schaben / Meißeln [DIN 8589]. Das Fertigungsverfahren des Drehens wird in der DIN-Norm 8589 als "Spanen mit geschlossener, meist kreisförmiger Schnittbewegung und beliebiger, quer zur Schnittichtung liegender Vorschubbewegung" definiert. "Die Drehachse der Schnittbewegung behält ihre Lage zum Werkstück unabhängig von der Vorschubbewegung bei (Drehachse werkstückgebunden)." Dieses Fertigungsverfahren kann wiederum je nach Lage der Bearbeitungsstelle am Werkstück, nach der Vorschubrichtung und der erzeugten Bearbeitungsfläche in feinere Arten des Drehens, wie zum Beispiel Außen- und Innendrehen, Längs- und Querdrehen, sowie Plan-, Rund-, Schraub-, Wälz-, Profil- und Formdrehen unterteilt werden [Falk, Gockel, Lernet, Schlossorsch, 1990; Fischer, 1990]. Die häufigste Anwendung des Drehens ist die Fertigung von rotationssymmetrischen Werkstück-Außen- bzw. Innenkonturen mit vorwiegend einschneidigen, ständig im Eingriff stehenden Werkzeugen, die hauptsächlich mit einer kreisförmigen Schnittbewegung und einer geradlinigen stetigen Vorschubbewegung erzeugt werden. Dabei wird meist die Schnittbewegung vom Werkstück und die Vorschubbewegung vom Werkzeug ausgeführt. Die Bearbeitung der Werkstücke durch Drehen ist im allgemeinen mit einer großen Zahl von Verfahrensvarianten technisch möglich. Die Auswahl der Variante muß unter wesentlicher Berücksichtigung des Werkstoffs des Werkstückes, der Form und den Abmessungen des Rohlings, den Anforderungen an Genauigkeit und Oberflächengüte des Werkstückes und den zur Verfügung stehenden Fertigungsmitteln erfolgen [Müller, 1989].

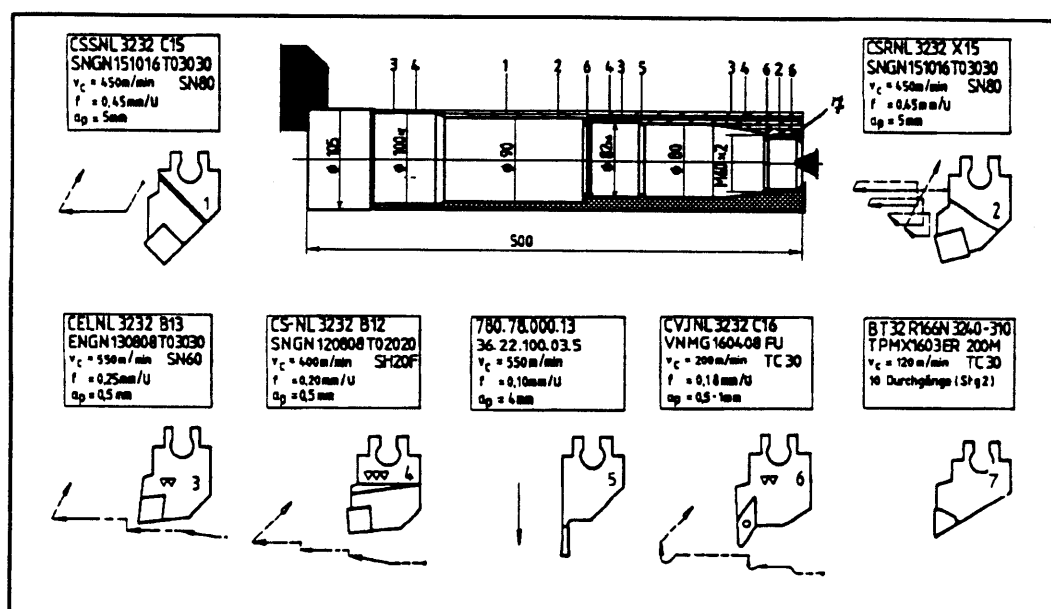


Abbildung 2: typischer Produktionsplan für die Fertigung eines rotationssymmetrischen Drehteiles (aus: [SPK-Feldmühle, 1984], Nachdruck mit Genehmigung)

In Abbildung 2 sind ein typisches rotationssymmetrisches Werkstück, sowie die Spezifikation der einzelnen zur Fertigung verwendeten Schnitte des Produktionsplanes dargestellt [SPK-Feldmühle, 1984]. Der Rohling ist in der Abbildung durch die schattierte Fläche gekennzeichnet. Die zu fertigende Kontur liegt innerhalb der schattierten Fläche, und ist mit Bemaßungen für die einzelnen Konturabschnitte versehen. Ferner zeigt die Abbildung die Art der Aufspannung, die durch die schwarze Fläche oben links und das Dreieck rechts kenntlich gemacht ist, und in diesem Falle aus einem Dreibackenfutter mit Spitze besteht. Die einzelnen Schnitte, Nummer eins bis sieben, sind durch Art und Form des Werkzeugs - graphische Darstellung, sowie konkrete Bezeichnung, z.B. CSSNL 3232 C15 bei Schnitt eins - Art des Schneidstoffes, z.B. SNGN 151016 T0 3030 SN80, den Verfahrenweg des Schnittes, und die Angabe der Schnittgeschwindigkeit, z.B.  $v_c = 450 \text{ m/min}$ , des Vorschubs, z.B.  $f = 0,45 \text{ mm/U}$ , und der Schnitttiefe, z.B.  $a_p = 5 \text{ mm}$ , beschrieben. Zu der vollständigen Beschreibung eines Produktionsplanes gehören zusätzlich noch Angaben über Toleranzen und Oberflächengüten des Werkstückes, sowie eine genaue Spezifikation der verwendeten Maschine.

### III BEGRIFFSBILDUNG

#### III.1 Begriffsbildung in der Kognition

Das Ziel menschlicher Begriffsbildung ist im allgemeinen der Versuch, die Vielfalt vorkommender Objekte und Ereignisse in verschiedene Klassen einzuteilen, um in angemessener Weise auf die Anforderungen unserer Umwelt reagieren zu können [Hoffmann, 1988; Wrobel 1991a].

Unter einem Begriff versteht man die kognitive Repräsentation einer Menge von zwei oder mehr Objekten oder Ereignissen, die aufgrund gemeinsamer Eigenschaften oder äquivalenten Funktionen in einem bestimmten Zusammenhang zusammengefasst werden können. Dagegen stellt eine Klasse die Menge der tatsächlichen Objekte oder Ereignisse dar, die mit dem zugehörigen Begriff assoziiert werden. In einem Begriff ist demnach die Intension, die Bedeutung, repräsentiert, während die Klasse die Extension, die expliziten Instanzen eines Begriffes, widerspiegelt [Anderson, 1990; Mervis, Rosch, 1981].

Zwei Prinzipien, die der Begriffsbildung zugrunde liegen, sind erstens das Bestreben, möglichst viele Informationen aus der Umwelt zu speichern, und dabei möglichst viele Wissensquellen zu erhalten und zweitens die Einsicht, daß die Welt nicht eine unstrukturierte Menge von gleichwahrscheinlich auftretenden Kombinationen von Eigenschaften darstellt. Durch die Bildung von Begriffen reduziert man sowohl die zu verarbeitenden Reize als auch die Unsicherheit im Austausch von Informationen [Hoffmann, Ziebler, 1982; Rosch, 1978].

Ein entscheidender Nachteil der vorherrschenden Begriffsbildungsforschung besteht darin, daß Experimente oft mit künstlich erzeugten Objekten durchgeführt werden, die sich von den realen Objekten qualitativ unterscheiden. Zudem werden Begriffsbildungsexperimente meist als Erinnerungsaufgabe simuliert, was nicht den eigentlichen Vorgängen im Alltag entspricht [Murphy, Wright, 1984].

##### III.1.1 Phänomene

###### Aufgaben der Begriffsbildung

Zunächst kann man im Rahmen der menschlichen Begriffsbildung verschiedene Aufgaben als Anforderungen an einen Begriffsbildungsprozess charakterisieren. Eine erste Einteilung erfolgt nach Identifikationsleistungen und Speicherung im Gedächtnis. Miller und Johnson-Laird [Armstrong, Gleitman, Gleitman, 1983; Miller, Johnson-Laird, 1976] schlagen die grundsätzliche Unterscheidung zwischen 'identification procedure' und 'core' vor. Die 'identification procedure' enthält Eigenschaften, die zur Klassifizierung eines Objektes verwendet werden. Der 'core' dagegen umfasst Eigenschaften, die Attribute enthalten und Relationen zu anderen Begriffen offenbaren. Unter Identifikation versteht man allgemein die Zuordnung von sensorisch gegebenen Objekten zu im Gedächtnis gespeicherten Begriffen. Die Identifikationsaufgaben unterscheiden sich wiederum in Benennung, Zuordnung und Entdeckung. Bei der Benennung ist das Objekt gegeben und es wird ein Begriff gesucht, zu dem das Objekt gehört. Bei der Zuordnung sind sowohl Objekt als auch Begriff gegeben, und es ist zu entscheiden, ob das Objekt zu diesem gegebenen Begriff gehört. Bei der Entdeckung schließlich ist nur der Begriff gegeben und es ist in einem vorgegebenen Rahmen zu entscheiden, ob ein Objekt existiert, das mit diesem Begriff assoziiert werden kann [Hoffmann, Ziebler, Grosser, Kämpf, 1985]. Die Speicherung im Gedächtnis ist im wesentlichen ein Repräsentationsproblem und eine Ordnungs- oder Verwaltungsaufgabe. Man geht sogar davon aus, daß der Mensch über hochentwickelte automatisierte Verfahren verfügt, die bei der Identifikation eines zu klassifizierenden Objektes benutzt werden [Murphy, Medin, 1985; Smith, Osherson, 1988].

###### Begriffshierarchie

Verschiedene Begriffe treten nicht unabhängig voneinander auf einer gleichen Ebene der Abstraktion auf, sondern werden in einer Hierarchie wechselnder Abstraktionsebenen wahrgenommen. Zum Beispiel ist eine Teerose auf unterer Ebene eine spezielle Art Rose, auf höheren Ebenen eine Rose, eine Blume und schließlich eine Pflanze. Die speziellste Ebene wird für jedes Objekt eine eigene Klasse enthalten, die allgemeinste fasst alle Objekte in einer Klasse zusammen. Eine solche Begriffshierarchie wird auch als Taxonomie bezeichnet [Anderson, 1990; Rosch, 1978]. Man spricht von einer vertikalen Dimension, die Aspekte der Begriffsinclusionen zwischen verschiedenen Ebenen der Hierarchie betrachtet, und einer horizontalen Dimension, die Aussagen über die Einteilung in Klassen auf einer Abstraktionsebene machen.



## Basic Level

In verschiedenen Studien wurde nachgewiesen, daß einer bestimmten Abstraktionsebene eine besondere Bedeutung zukommt. Rosch und ihre Mitarbeiter nannten diese Ebene den 'basic level' [Rosch, 1978; Mervis, Rosch, 1981]. Dieser 'basic level' zeichnet sich durch eine Reihe nachgewiesener spezieller Eigenschaften aus. Der 'basic level' ist die meist abstrakte Ebene, auf der gemeinsame Attribute, gemeinsame Bewegungssequenzen in der Interaktion mit dem Menschen, eine objektive Ähnlichkeit in der Form, und eine Identifizierbarkeit einer durchschnittlichen Gestalt auftreten. Darüberhinaus ist der 'basic level' die meist abstrakte Ebene, auf der Instanzen eines Begriffes ungefähr die selben Teile besitzen. Auf dieser Ebene ist die Ähnlichkeit zwischen einzelnen Instanzen einer Klasse maximal, während die Ähnlichkeit zwischen Instanzen verschiedener Klassen minimal ist [Hemenway, Tversky, 1984; Hoffmann, 1991b; Medin, Smith, 1984; Mervis, Rosch, 1981; Rosch, 1978]. Versuchspersonen erlernen Begriffe auf der Ebene des 'basic level' am schnellsten und besten, und ordnen Objekte spontan Begriffen dieser Ebene zu [Medin, Smith, 1984; Mervis, Rosch, 1981]. Der 'basic level' zeichnet sich ferner durch die meisten Attributnennungen aus, wenn eine Versuchsperson aufgefordert wird, charakteristische Merkmale eines Begriffes zu nennen. In den Begriffsketten Küchenstuhl - Stuhl - Möbel, oder weiße Birke - Birke - Baum, ist zum Beispiel Stuhl bzw. Birke der Begriff, bei dem man in Experimenten die Eigenschaften des 'basic level' beobachten kann [Rosch, 1978].

Hoffmann und Ziebler [Hoffmann, Ziebler, 1982] bezeichnen die dominierenden Begriffen mittlerer Allgemeinheit als Primärbegriffe, und wiesen nach, daß Primärbegriffe die Begriffe innerhalb einer Hierarchie sind, die die relativ abstrakteste Ebene darstellen, die noch mit sensorischen, also wahrnehmbaren, Eigenschaften beschrieben wird. Allgemeinere Klassen werden mit sogenannten kategorialen Begriffen assoziiert, die Objekte aufgrund gemeinsamer funktionaler Beziehungen in den verschiedensten Zusammenhängen zusammenfassen. Diese Trennung zwischen sensorischen und kategorialen Begriffen ist künstlich, und ist in der Realität als kontinuierlicher Übergang zu beobachten. Sie zeigten weiterhin, daß sensorische, in dem Falle visuelle, Wirkungen von Objekten unmittelbar die Ebene der Primärbegriffe aktivieren, daß Primärbegriffe bei der Benennung eines Begriffes für ein Objekt zuerst genannt werden, und daß die Zuordnung eines Objektes zu den Primärbegriffen bei verschiedenen vorgegebenen Begriffen den geringsten Zeitaufwand erfordert [Hoffmann, 1988; Hoffmann, Ziebler, Grosser, Kämpf, 1985].

Im Bereich der natürlichen Sprache kann beobachtet werden, daß Worte, die dem 'basic level' entsprechen, am häufigsten in Alltagskonversationen verwendet werden.

Welcher Abstraktionsgrad als 'basic level' angesehen wird, hängt sowohl vom kulturellen Umfeld, als auch unter Umständen vom Kenntnisstand der Person in einem bestimmten Wissensgebiet ab [Mervis, Rosch, 1981].

## Begriffsarten

In der Begriffsbildung unterscheidet man verschiedene Arten von Begriffen. Ein Faktor, der die Art eines Begriffes bestimmt, ist der Typ der zusammengefassten Erscheinungen. Es kann grundsätzlich zwischen ereignisbezogenen und merkmalsbezogenen Begriffen unterschieden werden. Ereignisbezogene Begriffe resultieren aus der Klassifizierung von Vorkommnissen, Geschehen, eben Ereignissen aufgrund der sie gestaltenden Personen und Dinge sowie deren regelhaften Beziehungen zueinander. Merkmalsbestimmte Begriffe dagegen entstehen aus der Klassifizierung von Eigenschaften im Gedächtnis. Ein ereignisbezogener Begriff wie Lehren hat keine Bedeutung für sich allein. Sein Verständnis erfordert eine zugehörige Gruppe anderer Begriffe, zum Beispiel Schüler, Lehrer, und Fachgebiet. Ein merkmalsbezogener Begriff wie Birke dagegen hat einen eigenständigen Inhalt [Klix, 1984]. An anderer Stelle finden sich Repräsentationen stereotyper Ereignisse, die 'scripts' genannt werden [Schank, Abelson, 1977], wie zum Beispiel 'Essen in einem Restaurant'. Es gibt auch Begriffe, die nie explizit erlernt werden, sondern als Ergebnis einer impliziten Begriffsbildung entstehen. Experten zeigen in den Gebieten ihrer Expertise beispielsweise deutlich andere Begriffsbildungen als Laien, die sie zudem oftmals nur schwer verbalisieren können. Die besonderen Klassen bilden sich offenbar allein im Resultat des intensiven Umgangs mit dem jeweiligen Fachgebiet heraus. Auch bei Kindern werden noch vor dem Spracherwerb klassifikatorische Unterscheidungen beobachtet. Welche genauen Mechanismen bei der Bildung impliziter Begriffe angewendet werden, ist allerdings noch ungeklärt [Hoffmann, 1991b].

## komplexe Begriffe

Als eine besondere Art Begriff kann man komplexe Begriffe betrachten. In den meisten Fällen beschäftigt sich die Begriffsbildungsforschung nur mit einfachen Begriffen, wie Stuhl, Fahrzeug, oder weiblich. In

komplexen Anwendungsbereichen muß man aber auch zusammengesetzte Begriffe betrachten, wie etwa rote Blume, schwerer Gegenstand, oder Pferderennen. In der natürlichen Sprache wird man in Form ganzer Sätze täglich mit Problemen komplexer Begriffe konfrontiert.

Komplexe Begriffe, die allgemein aus der Kombination einfacher Begriffe entstehen, können nicht als

Schnittmengenbildung definierender Features gekennzeichnet werden, oder in denen komplexe Begriffe durch Projektionen in verschiedene einfachere Begriffe geteilt werden, können nicht erklären, wie komplexe Begriffe in den verschiedenen Aufgaben der Begriffsbildung als Ganzes verarbeitet werden. Der Begriff Pferderennen zum Beispiel hat sicher nicht die Schnittmenge der Merkmale von Pferd und Rennen als charakteristische Eigenschaften. Es ist anzunehmen, daß bei der Bildung komplexer Begriffe eine Hintergrundtheorie über allgemeines Wissen in der Welt berücksichtigt werden muß [Mervis, Rosch, 1981; Smith, Osherson, 1988; Murphy, Medin, 1985].

Smith und Osherson untersuchten relativ simple komplexe Begriffe, die aus einem Adjektiv und einem Nomen konjunktiv zusammengesetzt sind. Sie betrachteten zunächst den Ansatz der 'Fuzzy Theorie' [Zadeh, 1965], die grundsätzlich von der Existenz einer charakteristischen Funktion ausgeht. Eine charakteristische Funktion bezüglich eines bestimmten Begriffes bildet ein Objekt der Domäne auf einen Wert des Intervalles  $[0,1]$  ab. Je höher der Wert der charakteristischen Funktion für ein Objekt ist, desto sicherer gehört es zu dem Begriff. Die charakteristische Funktion eines kombinierten Begriffes ist dann einfach die Bildung des Minimums der Werte der charakteristischen Funktionen der einzelnen Komponenten. In Experimenten wurde gezeigt, daß diese 'Fuzzy Theorie' nicht angemessen ist, da die Ergebnisse der Versuche zu der Bewertung der Typikalität eines komplexen Begriffes in den meisten Fällen über dem Minimum der Typikalität der Komponenten lagen. Smith und Osherson folgerten, daß es keine einfache Funktion zur Beschreibung der Typikalität von komplexen Begriffen geben kann. Weitere Experimente sollten den Einfluß des Adjektives auf das Nomen aufzeigen. Ein Adjektiv in einem solchen konjunktiv zusammengesetzten Begriff wirkt modifizierend, indem es erstens die Art des relevanten Attributes bestimmt (z.B. Farbe), zweitens die vorherrschende Ausprägung dieses Attributes festlegt (z.B. rot), und drittens den Wert in einer gewichteten Summe der Features für dieses Attribut erhöht. In den Untersuchungen unterschied man aufgrund dieser Merkmale zwischen positiv diagnostischen, negativ diagnostischen, und nicht diagnostischen Adjektiven, je nach dem, ob eine Begriffszugehörigkeit durch das Adjektiv positiv, negativ, oder gar nicht beeinflusst wird. Zum Beispiel ist in roter Apfel das Adjektiv positiv diagnostisch, in brauner Apfel negativ diagnostisch, und in ungeschnittener Apfel nicht diagnostisch.

Insgesamt findet man keine ausreichenden Untersuchungen komplexer Begriffe vor.

#### Begriffsrelationen

Neben einzelnen Begriffen gibt es auch Relationen zwischen Begriffen, die man wieder als eigene Begriffe ansehen kann. Mit den Begriffen ist in der Regel auch Wissen über die zwischen ihnen bestehenden Relationen bekannt, wie zum Beispiel Ober- und Unterbegriffsrelationen, nebengeordnete Relationen, Teil-von-Relationen, und Instrument-Relationen [Anderson, 1990; Hoffmann, 1991b; Medin, Smith, 1984].

Klix [Klix, 1984] unterscheidet grundsätzlich zwischen innerbegrifflichen und zwischenbegrifflichen Relationen. Analog zu der Unterscheidung zwischen ereignis- und merkmalsbezogenen Begriffen differenzieren sich innerbegriffliche und zwischenbegriffliche Relationen nach ihren Ursachen. Innerbegriffliche Relationen werden aus Merkmalseigenschaften hergeleitet, zwischenbegriffliche Relationen dagegen aus der Rolle oder Funktion, die zu klassifizierende Objekte in dem beschriebenen Kontext haben. Ober- und Unterbegriff, sowie nebengeordnete Begriffe sind Beispiele innerbegrifflicher Relationen, Teil-von- und Instrument-Relationen Beispiele zwischenbegrifflicher Relationen. Zwei weitere merkmalsbestimmte Relationen sind die Kontrast- und die Komparativrelation. Kontrastrelationen werden

mehr typische Instanzen einer Klasse. Typischere Objekte einer Klasse zeichnen sich zum Beispiel durch eine höhere Merkmalsübereinstimmung mit der Repräsentation des Begriffes aus, als weniger typische. Für typischere Beispiele einer Klasse werden Zugehörigkeitsentscheidungen schneller getroffen als für untypischere. Zum Beispiel wird schneller bestätigt 'Ein Rabe ist ein Vogel.' als 'Ein Strauss ist ein Vogel.', weil ein Rabe ein typischerer Vogel ist als ein Strauss. [Armstrong, Gleitman, Gleitman, 1983; Hoffmann, 1991b; Hoffmann, Zießler, 1982; Mervis, Rosch, 1981]. Armstrong et. al. untersuchten die Unterschiede zwischen wohl-definierten und nicht wohl-definierten Begriffen unter diesem Aspekt. Unter wohl-definierten Begriffen verstehen sie Begriffe, die eindeutig durch eine Menge von Features definiert werden können. Gerade Zahl ist zum Beispiel ein wohl-definierter Begriff und könnte etwa mit 'ohne Rest durch zwei teilbar' charakterisiert werden. Dagegen ist Fahrzeug kein wohl-definierter Begriff, da es keine eindeutigen Merkmale eines Fahrzeuges gibt. Interessanterweise gaben Versuchspersonen auch verschiedene Typikalität an, wenn es sich um wohl-definierte Begriffe handelte, für die sie in einem ersten Versuchsteil sagten, daß die Zugehörigkeit eines Objektes zu einem dieser Begriffe nach dem 'alles-oder-nichts'-Prinzip bestimmt werden könne, d.h. insbesondere, daß ein Objekt entweder zu diesem Begriff gehört oder nicht, es aber keine Abstufungen dieser Zugehörigkeit geben darf. Eine 10 wurde als typischere gerade Zahl klassifiziert als eine 34. Insgesamt ist allerdings eine Tendenz zu beobachten, daß es für wohl-definierte Begriffe keine so untypischen Objekte gibt wie für nicht wohl-definierte Begriffe.

### Kontextabhängigkeit

Sowohl die Begriffsbildung als auch die Bestimmung der Zugehörigkeit von Objekten zu bestimmten Begriffen ist nicht unabhängig vom Kontext. Eine Identifikationsentscheidung für ein Objekt kann durch den Kontext erleichtert, erschwert, oder sogar verändert werden. Es ist zum Beispiel leichter, einen Toaster zu identifizieren, wenn er in einer Küche steht, und schwieriger, wenn er in einem Badezimmer zu entdecken ist. Ein Objekt 13 wird im Kontext 12, 13, 14 als Ziffer mit dem Wert 13 erkannt, in einem Kontext A, 13, C aber als der Buchstabe B identifiziert. Hoffmann unterscheidet drei verschiedene Arten von Kontexten: gleichzeitig wirkende Reizbedingungen, vorausgegangene Reizbedingungen, und Handlungsabsichten. Begriffe enthalten vermutlich solche kontextabhängigen Merkmale, aber auch bestimmte Eigenschaften, die unabhängig vom Kontext wahrgenommen werden, aber sind nie durch vollständig voneinander unabhängige Features bestimmt. Ein Tennisball wird zum Beispiel immer die Eigenschaft rund haben, aber unter Umständen im Kontext Fußball spielender Kinder auf einem Schulhof als Fußball klassifiziert werden, auf einem Tennisplatz dagegen als Tennisball [Anderson, 1990; Hoffmann, 1991b; Mervis, Rosch, 1981; Smith, Medin, 1984].

Medin und Schaffer [Medin, Schaffer, 1978] entwickelten eine Kontexttheorie zur Erklärung der Begriffsbildung, um insbesondere den Einfluß des Kontextes in Experimenten nachzuweisen. Dabei wurde angenommen, daß Begriffsbildung auf einem Retrieval von gespeicherter Beispielinformation beruht, und ein Begriff immer durch eine Menge von verschiedenen Dimensionen, wie Gestalt, Farbe, etc., charakterisiert werden kann. Jede Dimension hat verschiedene mögliche Werte. Im einfachsten Fall ist jede Dimension zweiwertig, so daß ein Objekt immer durch eine Binärzahl kodiert werden kann, in der jede Ziffer den Wert einer Dimension bestimmt. Die einzelnen Komponenten dieser Objekte werden nicht unabhängig voneinander, sondern immer als zusammenhängende Kette gespeichert. Um die Ähnlichkeit zwischen einem neuen Objekt und Beispieloobjekten einer Klasse zu bestimmen, wird eine Formel verwendet, die die einzelnen Dimensionen nicht additiv, wie in unabhängigen Modellen, sondern interaktiv, multiplikativ berechnet. Dadurch werden immer die Kette als Ganzes und nicht die Komponenten unabhängig voneinander repräsentiert und bei der Ähnlichkeitsentscheidung berücksichtigt. Es wurden verschiedene Experimente durchgeführt, die typische Phänomene der Begriffsbildung nachweisen konnten, und zusätzlich zeigten, daß das Kontextmodell bessere Ergebnisse bei der Vorhersage der Versuchsausgänge leistet als unabhängige Modelle.

Der Begriffsbildungsprozess und die damit verbundenen Probleme müssen also immer in dem gegebenen Kontext untersucht werden.

### Begriffe und Verhalten

Ein relevanter Kontext, in dem die Aufgaben der Begriffsbildung betrachtet werden müssen, ist der Handlungsrahmen, in dem sie auftreten.

Die Ziele einer Handlung bestimmen die Begriffsbildung maßgeblich. Unter Umständen werden auch spezielle Begriffe herausgebildet. Zum Beispiel wird bei dem Ziel, Gewicht zu verlieren, ein spezieller Begriff 'Nahrungsmittel, die bei einer Diät nicht gegessen werden dürfen' gebildet. Auch bei speziellen Begriffen wird zum Beispiel Typikalität auftreten. Schokolade ist für obigen Begriff sicher typischer als Brot [Murphy, Medin, 1985; Medin, Smith, 1984; Hoffmann, 1991b].

Hoffmann betont, daß das entscheidende Kriterium für die Zusammenfassung individuell unterschiedlicher

Zum Beispiel wird auf dem Abstraktionsgrad des Begriffes Vogel das Feature 'zwei Beine' genannt, aber auf einem darunter liegenden Niveau wird 'zwei Beine' für den Begriff Rabe nicht mehr erwähnt. Die Nennung der relevanten Features variiert außerdem von Individuum zu Individuum [Rosch, 1978]. Weiterhin gibt es auch Features von geringerer Bedeutung, die genannt werden, aber nicht als definierend gelten können [Rosch, 1978]. Andere Features werden nicht explizit angegeben oder unterdrückt, obwohl sie zur Definition eines Begriffes implizit verwendet werden [Armstrong, Gleitman, Gleitman, 1983; Medin, Smith, 1984]. Hoffmann [Hoffmann, 1991b] weist darauf hin, daß grundsätzlich eine Unterscheidung zwischen relevanten und irrelevanten Features nur schwer möglich ist.

Neben der Problematik der Relevanz von Features und der Featurebestimmung ergeben sich noch andere Probleme für den klassischen Erklärungsansatz. Der klassische Erklärungsansatz suggeriert, daß die Grenzen zwischen einzelnen Begriffen streng zu ziehen sind, da Begriffe eindeutig durch ihre definierenden Features voneinander abgegrenzt sind, und daß die Zugehörigkeit eines Objektes zu einem Begriff immer gleich ist, da entweder die definierenden Features erfüllt sind oder nicht. Es muß aber davon ausgegangen werden, daß die Klassen, die mit Begriffen assoziiert sind, nicht streng voneinander abtrennbar sind und die Zugehörigkeit in Stufen von Typikalität vorliegt [Mervis, Rosch, 1981; Medin, Smith, 1984]. Eine Zugehörigkeitsentscheidung für ein Objekt zu einem Begriff ist auch nicht immer eindeutig entscheidbar (z.B. Ist ein Teppich ein Möbelstück ?) [Medin, Smith, 1984]. Die Zuordnung im klassischen Ansatz kann durch ein reines 'Feature-matching' durchgeführt werden, die aber dem menschlichen Prozess nicht angemessen entspricht, da auch Kenntnisse über Operationen, Transformationen, und Relationen zwischen Features in den Prozess mit eingehen [Murphy, Medin, 1985].<sup>1</sup> Auch die Repräsentation eines Begriffes, die nach dem klassischen Erklärungsansatz durch die Repräsentation der relevanten Features eindeutig bestimmt ist, ist nicht streng in diesem Sinne [Mervis, Rosch, 1981; Rosch, 1975]. Da die Begriffshierarchie durch Mengeninklusion der definierenden Featuremengen gekennzeichnet ist, und definierende Features einer Abstraktionsebene auf die darunterliegenden Ebenen vererbt werden, folgt unmittelbar daraus, daß ein bestimmter Begriff dem nächst höher liegenden Begriff in der Begriffshierarchie immer am meisten ähnlich ist. Diese Tatsache gilt aber nicht allgemein (z.B. Huhn - Vogel - Tier) [Medin, Smith, 1984]. In [Armstrong, Gleitman, Gleitman, 1983] wurde ein Experiment durchgeführt, das den klassischen Erklärungsansatz versuchte auf wohl-definierte Begriffe, wie zum Beispiel gerade Zahl oder weiblich, anzuwenden, da dort eine eindeutige Definition durch notwendige und hinreichende Features angenommen werden kann. Aber selbst bei solchen wohl-definierten Begriffen treten Effekte der Typikalität auf. Damit konnte man endgültig belegen, daß Begriffe nicht einfach als Summe definierender

---

Features angesehen werden können.

Dennoch scheint der klassische Ansatz nicht grundsätzlich völlig falsch, sondern nur unzureichend, um die tatsächlichen Vorgänge der menschlichen Begriffsbildung zu erklären [Murphy, Medin, 1985; Rosch, 1975].

#### Der probabilistische Erklärungsansatz

Der probabilistische Ansatz geht weiterhin davon aus, daß ein Begriff durch eine Menge von Features bestimmt ist. Diese Features sind aber nicht mehr definierend, sondern haben eine gewichtete Wahrscheinlichkeit, d.h. sie sind nur noch charakteristisch für einen Begriff. Ein Objekt gehört danach zu einem bestimmten Begriff, wenn eine bestimmte Schranke für die Summe von gewichteten Wahrscheinlichkeiten der Features überschritten wird [Anderson, 1990; Medin, Smith, 1984; Murphy, Medin, 1985].

Im Gegensatz zu dem klassischen Erklärungsansatz können durch den probabilistischen Ansatz einige Phänomene beschrieben werden. Die Grenzen eines Begriffes sind nicht mehr unbedingt scharf, da einerseits für die zu erreichende Schranke ein kleines Intervall angegeben werden kann, und andererseits Objekte nahe der Schranke eines oder auch mehrerer Begriffe liegen können, und damit zweifelhafte Zuordnungen ergeben. Das Phänomen der Typikalität wird dadurch erklärt, daß es Objekte gibt, die eine höhere bzw. geringere Summe von gewichteten Wahrscheinlichkeiten besitzen, und sich darin im Grad ihrer Typikalität unterscheiden. Da eine Begriffshierarchie nicht mehr eindeutig definiert ist wie bei dem klassischen Erklärungsansatz, kann auch eine größere Ähnlichkeit zu einem Begriff einer entfernteren Abstraktionsebene als der nächst höheren vorliegen. Ein Huhn beispielsweise kann mehr Ähnlichkeit zu einem Tier aufweisen als zu einem Vogel, weil Features wie 'geht', 'lebt auf einer Farm' charakteristisch für ein Tier, aber nicht unbedingt für einen Vogel sind.

Ein grundsätzliches Problem dieses Erklärungsansatzes besteht darin, daß er nicht angemessen das Wissen des Menschen, zum Beispiel über Wertebereiche einer Eigenschaft, oder Beziehungen zwischen einzelnen

---

<sup>1</sup> Collins und Loftus [Collins, Loftus, 1975] schlagen in ihrem 'spreading-activation-model' vor, die Relationen zwischen Features durch 'labeled links' zu repräsentieren.

Features, widerspiegelt. Insgesamt betrachtet ist der Ansatz zu uneingeschränkt, was die Zulassung der Features und möglicher Begriffe betrifft [Medin, Smith, 1984]. Ein weiterer Nachteil liegt auch hier in der Restriktion der Features und zusätzlich der Gewichte und Schranken für bestimmte Begriffe.

#### Der Beispiel Erklärungsansatz

Dem Beispiel Ansatz liegt nicht mehr die Idee zugrunde, daß ein Begriff durch eine Menge definierender oder charakteristischer Features bestimmt ist, sondern durch die Beschreibung eines oder mehrerer Beispiele des Begriffes charakterisiert wird. Bei diesem Ansatz ist die Begriffszugehörigkeit durch die größere Ähnlichkeit eines Objektes zu Beispielen eines Begriffes gegenüber denen eines anderen Begriffes determiniert [Hoffmann, 1991b; Hoffmann, Ziebler, 1982; Medin, Smith, 1984; Murphy, Medin, 1985; Wrobel, 1991b].

In diesem Ansatz können Begriffe, die keine strikten Grenzen haben, erklärt werden, indem ein Objekt Ähnlichkeit zu den Beispielen mehrerer Begriffe aufweist. Durch einen verschieden hohen Grad an Ähnlichkeit erhält man die Typikalität von Begriffszugehörigkeiten. Ähnlich wie bei dem probabilistischen Ansatz ist auch hier die Begriffshierarchie nicht eindeutig festgelegt, so daß die beobachteten Erscheinungen bei dem Beispiel Huhn - Vogel - Tier erklärt werden können [Medin, Smith, 1984].

Die Idee des Beispiel Ansatzes wird durch einige Versuchsergebnisse bestätigt, bei denen nachgewiesen werden konnte, daß Versuchspersonen eine Zuordnung sicherer und schneller treffen, je mehr ähnliche Beispiele sie in der Lernphase des Begriffes gezeigt bekamen. Bei der Frage, ob ein neues Objekt alt oder neu sei, antworteten die Versuchspersonen häufiger mit alt, wenn mehr ähnliche Beispiele bekannt waren [Medin, Schaffer, 1978; Medin, Smith, 1984]. Bewertet man die Güte neuer Beispiele anhand ihrer Ähnlichkeit zu bisher vorgegebenen Beispielen, so konnte festgestellt werden, daß gute Beispiele insgesamt schneller erlernt werden als schlechte Beispiele [Medin, Smith, 1984].

Der Beispiel Ansatz hat gegenüber dem probabilistischen Ansatz den Vorteil, daß er Informationen über den Wertebereich einer Eigenschaft und über Relationen zwischen Eigenschaften berücksichtigen kann, aber es werden keine Aussagen zu Einschränkungen an Eigenschaften eines Begriffes gemacht, und nichts darüber ausgesagt, was einen Begriff überhaupt konstituiert [Smith, Medin, 1984].

#### Der Prototyp Erklärungsansatz

Ein dem Beispiel Ansatz ähnlicher Versuch, die Begriffsbildung beim Menschen zu erklären, ist der Prototyp Ansatz. Beim Prototyp Ansatz wird angenommen, daß ausgehend von gegebenen Beispielen eine zentrale Tendenz des Begriffes abstrahiert wird, die alle oder die meist privilegierten Eigenschaften des Begriffes enthält. Diese zentrale Tendenz kann als Prototyp des Begriffes verstanden werden, der im

beider Begriffe, so ist die Ähnlichkeit durch eine 'Matching Funktion'  $S(A \cap B, A-B, B-A) = af(A \cap B) - bf(A-B) - cf(B-A)$  gegeben, mit Gewichtungsfaktoren  $a, b, c \geq 0$ . Die 'Matching Funktion' ist eine ansteigende Funktion der gemeinsamen Features, und eine fallende Funktion der unterschiedlichen Features. Tversky entwickelte mit dieser Funktion zum einen das Kontrastmodell und zum anderen das Ratiomodell, um die Eigenschaften von Ähnlichkeiten zu untersuchen. Er betont, daß die Ähnlichkeit weder eine transitive, noch eine symmetrische Funktion ist. Wenn  $x$  ähnlich zu  $y$  ist, und  $y$  ähnlich zu  $z$ , muß nicht unbedingt auch  $x$  ähnlich zu  $z$  sein, da die Ähnlichkeit zwischen  $x$  und  $y$  und  $y$  und  $z$  aufgrund verschiedener Eigenschaften auftreten kann. In einem Experiment zeigte er, daß zum Beispiel auf die Frage 'Mexico ähnlich zu USA?' mit ja, aber auf 'USA ähnlich zu Mexico?' mit nein geantwortet wird. Es ist für die Beurteilung der Ähnlichkeit entscheidend, an welcher Stelle das vorherrschende Objekt, hier die USA, steht. Der Kontext und die Art der Aufgabe bestimmen das Ähnlichkeitsurteil ebenfalls maßgeblich. Grundsätzlich ist keine allgemeingültige Aussage über die Ähnlichkeit zweier Objekte möglich, sondern immer nur bezogen auf bestimmte Begebenheiten. Außerdem zeigte er, daß Ähnlichkeit und Unterschiedlichkeit zwar entgegengesetzte Eigenschaften sind, aber das eine nicht das genaue Gegenteil des anderen sein muß. Diese Merkmale der Ähnlichkeit können je nach Wahl der Gewichte  $a, b, c$  berücksichtigt werden (zum Beispiel liegt bei  $b = c$  eine symmetrische Ähnlichkeitsfunktion vor), aber die Funktion ist immer davon abhängig, ob alle Features explizit angegeben werden können, und die Menge der definierenden Features endlich ist [Tversky, 1977]. Damit ergeben sich die Probleme des klassischen Erklärungsansatzes. Ist man bei der Wahl der Features absolut frei, so kann man triviale Features wie 'leichter als 10000 kg' (und 'leichter als 10001 kg', ...) oder 'existierte 10000000 Jahre vorher noch nicht' ('existierte 10000001 Jahre vorher noch nicht', ...) hinzunehmen, und die Ähnlichkeit zwischen zwei Objekten wird völlig willkürlich [Murphy, Medin, 1985]. Es ist außerdem unklar, ob Ähnlichkeitsurteile komponentenweise bezüglich einzelner Eigenschaften oder für einen Begriff insgesamt abgegeben werden [Medin, Schaffer, 1978]. Die Bewertung der Ähnlichkeit zweier Objekte als Grundlage eines Erklärungsansatzes zu wählen, führt zu einer Vielzahl neuer Probleme.

#### Der theoretische Erklärungsansatz

Die bisherigen Ansätze gehen immer mehr oder weniger davon aus, daß Begriffsbildung aus einem Vergleich von Features, und der Beurteilung von Ähnlichkeiten besteht. Begriffsbildung basiert aber auch auf kausalem Wissen, Regeln, theoretischer Konsistenz, und anderem theoretisch gearteten Wissen. Den Haupteinfluß haben solche theoretischen Kenntnisse bei der Klassifizierung neuer, bisher unbekannter Objekte, bei zweifelhaften Objekten, und bei der Rechtfertigung und Erklärung von bestimmten Begriffen [Anderson, 1990; Murphy, Medin, 1985].

Murphy und Medin versuchen die Ähnlichkeitsbasierten und die theoretischen Ansätze zu vergleichen und betonen, daß bei allen Aspekten der Begriffsbildungsforschung in den Ähnlichkeitsbasierten Ansätzen Features und gewichtete Ähnlichkeitmessungen vorherrschen, während theoretische Ansätze immer die zugrunde liegenden Prinzipien und Inferenz- und Erklärungsprozesse in Betracht ziehen.

Insgesamt werden theoretische Ansätze aber sehr viel weniger verfolgt und diskutiert als die Ähnlichkeitsbasierten Ansätze, obwohl es empirische Erkenntnisse für die Verwendung theoretischen Wissens bei der Begriffsbildung gibt [Murphy, Medin, 1985].

## III.2 Begriffsbildung im Maschinenbau

Der Bereich der Fertigung rotationssymmetrischer Drehteile im Maschinenbau ist eine technische Anwendungsdomäne, die für die Industrie von großer Bedeutung ist. In technischen Domänen ist die Unterscheidung zwischen Experten und Neulingen in besonderem Maße interessant. Insbesondere kann man das unterschiedliche Verhalten von Experten und Neulingen bei der Begriffsbildung untersuchen.

### III.2.1 Experte und Neuling

Experten zeichnen sich vor allem dadurch aus, daß sie über mehr Wissen verfügen als Neulinge. Dieses größere Wissen betrifft sowohl theoretische Kenntnisse als auch Erfahrungen in praktischer Anwendung dieser Kenntnisse. Das Wissen eines Experten ist aber nicht nur in seiner Menge dem Neuling überlegen, sondern hat auch eine andere Struktur. Der Experte weiß genauer über Relationen zwischen Objekten seiner Expertise bescheid.

Intuitiv wird man erwarten, daß ein Experte über eine größere Anzahl von Begriffen in seiner Domäne verfügt, die insbesondere auch spezieller ausgeprägt sein werden als die des Neulings. Die Begriffe des Experten sind differenzierter und weisen damit eine andere Struktur auf als die Begriffe des Neulings [Murphy, Wright, 1984].

Murphy und Wright führten einige Experimente im Bereich der klinischen Diagnose psychopathologischer Kinder durch, die diese intuitiven Erwartungen überprüfen sollten. In dieser Untersuchung wurden den Versuchspersonen bestimmte Arten psychischer Krankheiten als Begriffe vorgegeben und sie sollten Eigenschaften nennen, die typische Merkmale für diese Krankheiten sind. Die Versuchspersonen waren



Fertigungsprobleme eines Experten werden sowohl durch Eigenschaften und Merkmale der zu den Klassen gehörenden Probleme als auch durch typische Beispiele dieser Klassen repräsentiert werden. Bei der Bearbeitung eines neuen Fertigungsproblems kann ein zu den Klassen entworfener allgemeiner Produktionsplan nach einer Identifikation des neuen Problems angewendet werden.

#### Begriffshierarchie

Die Probleme der Fertigung werden auf verschiedenen Abstraktionsebenen klassifiziert. Auf einer oberen Ebene kann man die Kontur eines Problems beispielsweise zunächst als einseitig oder beidseitig monoton klassifizieren und erhält eine Konsequenz für die Auswahl einer Aufspannung oder die Anforderung an eine Maschine. Bei einseitig monotoner Kontur wird eine einmalige Aufspannung ausreichend sein, dagegen muß bei einer beidseitig monotonen Kontur eine Umspannung vorgenommen werden oder die Existenz von zwei Revolvern für eine beidseitige Bearbeitung als eine Anforderung an die Maschine gestellt werden. Auf einer unteren Ebene werden erst spezielle Konturverläufe interessant, wenn die genaue Schnittreihenfolge festzulegen ist.

Die Begriffshierarchie der Problemklassen kann unmittelbar mit dem hierarchischen Planen [Sacerdoti, 1974] in Verbindung gebracht werden. Ein Fertigungsproblem wird zunächst auf einer abstrakten Ebene bearbeitet, auf der beispielsweise grundsätzliche Planeigenschaften wie die Aufspannungsart festgelegt werden. Erst in weiteren Schritten wird der Experte feinere Planschritte entwickeln, die auf unterster Ebene bis zur genauen Festlegung einzelner Schnittkoordinaten reichen.

#### Basic Level

Es wird sich eine Ebene der Abstraktion ergeben, die man als 'basic level' bezeichnen kann, die sich für eine grundsätzliche Lösung eines Problems anbietet. Diese Ebene ist einerseits nicht zu allgemein, daß jedes Problem bearbeitet werden kann, und andererseits aber noch nicht so speziell, daß nur noch ein Problem bearbeitet werden kann. Auf dieser Ebene zeichnen sich die Fertigungsprobleme einer Klasse durch gemeinsame Eigenschaften (z.B. Werkstoffeigenschaften), eine hohe Ähnlichkeit in der Form (z.B. Monotonie der Werkstückkontur), und die Existenz gleicher Teile (z.B. Gewinde) aus. Instanzen der Klassen auf dieser Ebene besitzen eine maximale Ähnlichkeit zu Instanzen der selben Klasse und eine minimale Ähnlichkeit zu Instanzen anderer Klassen. Ein Experte wird vermutlich ein neues Fertigungsproblem spontan einer Klasse dieser Ebene zuweisen. Aufgrund dieser Eigenschaften liegt die Idee nahe, daß sinnvolle Skelettpläne gerade auf der Abstraktionsebene des 'basic level' konstruiert werden können.

#### Begriffsarten

Bei Problemklassen der Fertigung handelt es sich sowohl um ereignisbezogene als auch um merkmalsbezogene Begriffe. Einerseits ist für die Herausbildung einer Problemklasse ein allen Instanzen dieser Klasse gemeinsamer Skelettplan, also das Ereignis der Fertigung, von Bedeutung. Andererseits spielen gemeinsame Eigenschaften und Merkmale der Instanzen eine wichtige Rolle.

Eine Schwierigkeit liegt darin, daß Experten viele Eigenschaften implizit berücksichtigen, aber diese nicht explizit nennen werden. Ein erfahrener Dreher kennt zum Beispiel seine zur Verfügung stehenden Maschinen, und wird keine Anforderungen mehr nennen, die er in seinem Fertigungsumfeld nicht erfüllen kann, obwohl er grundsätzlich dieses Merkmal abgewägt hat, ohne es aber explizit zu erwähnen.

#### komplexe Begriffe

Bei den Begriffen, die mit den Problemklassen assoziiert werden, handelt es sich offenbar um komplexe Begriffe. Ein Problem wird niemals durch ein Merkmal allein beschrieben sein, so daß die Beschreibung einer Klasse durch mehrere Features als komplexer Begriff gedeutet werden kann.

#### Begriffsrelationen

Zwischen den einzelnen Begriffen der Problemklassen bestehen Relationen. Zum Beispiel kann eine Problemklasse eine andere Klasse vollständig umfassen, wenn alle Instanzen der zweiten Klasse auch Instanzen der ersten sind. Die erste Klasse ist dann ein Oberbegriff der zweiten, bzw. die zweite ein Unterbegriff der ersten. Durch die verschiedenen Abstraktionsebenen auf denen Fertigungsprobleme

## Typikalität

Für jede Klasse von Problemen gibt es typische Fälle und weniger typische Fälle. Ein Werkstück kann nahezu alle angegebenen Merkmale erfüllen, aber auch nur wenige, oder spezielle Grenzfälle eines Problems erreichen. Alle Probleme einer Klasse sollten aber so viel Ähnlichkeit zueinander aufweisen, daß sie mit einer gemeinsamen Lösungsstrategie bearbeitet werden können.

## Kontextabhängigkeit

Der Kontext spielt im Bereich der Fertigung eine erhebliche Rolle. Beispiele für Kontextaspekte sind im Bereich des Maschinenbaus die Beschreibung der Werkstatt, etwa die zur Verfügung stehenden Maschinen und Werkzeuge, und die Losgröße, d.h. die Anzahl der in einer Serie zu fertigenden Teile. Merkmale können in verschiedenen Kontexten andere Bedeutungen annehmen. Eine Maschine kann als 'stark' klassifiziert werden, weil sie eine Leistung von 50 kW hat und nur kleinere Wellen gefertigt werden müssen, aber die gleiche Maschine kann auch als 'sehr schwach' eingestuft werden, wenn ausschließlich große Schiffswellen gefertigt werden sollen. Das Merkmal 'Stärke' der selben Maschine hat in

Die Klassifizierung von Werkstückproblemen geschieht immer mit der Handlungsabsicht, die Fertigung eines Werkstückes durchzuführen. Es werden keine Klassen gebildet, die nicht Gemeinsamkeiten bei der Erreichung dieses Zieles aufweisen. Probleme einer Klasse müssen gemeinsame Ausgangsbedingungen erfüllen und bei Anwendung bestimmter Plansequenzen auf die Probleme einer Klasse müssen sich die gleichen Konsequenzen ergeben.

### III.2.3 Erklärungsansätze

#### Der klassische Erklärungsansatz

Nach dem klassischen Erklärungsansatz können die Begriffe der Problemklassen durch eine Menge von Features definiert werden. Solche Features sind im Bereich des Maschinenbaus sehr allgemein zu verstehen und können jeden beliebigen Ausdruck enthalten, den Experten in ihrer Domäne verwenden. Die definierenden Features einer Problemklasse beziehen sich immer auf die Probleme der Fertigung, die zu dieser Klasse gehören. Bei den Features kann es sich sowohl um Merkmale handeln, die sich auf Teile des Problems, zum Beispiel die Aufspannungsart, oder auf physikalische Eigenschaften, zum Beispiel Werkstoffeigenschaften, beziehen, als auch Relationen, zum Beispiel die Monotonie einer Kontur als Relation zwischen einzelnen Absätzen, oder Funktionen, zum Beispiel die ausreichende Leistung einer Maschine, repräsentieren. Die Bestimmung der Relevanz von Features stellt auch im Bereich des Maschinenbaus ein schwieriges Problem dar. Die Existenz strenger Grenzen zwischen einzelnen Problemklassen trifft bei der Fertigung rotationssymmetrischer Drehteile bedingt zu. Einerseits kann ein Fertigungsproblem immer auf verschiedene Weise gelöst werden, so daß auch die Zugehörigkeit eines Problems zu mehreren Problemklassen denkbar ist. Andererseits wird jedes Fertigungsproblem schließlich eindeutig durch einen Produktionsplan bearbeitet, so daß eine eindeutige Zuordnung zu einer Problemklasse immer möglich ist. Zum Beispiel kann jedes Fertigungsproblem, daß durch eine

Gewindes immer vollständige Plansequenzen existieren, die je nach Kennzeichnung des Gewindes eingesetzt werden können. Lediglich die Existenz eines Gewindes ist somit für die Zugehörigkeit eines Fertigungsproblems zu einer Problemklasse relevant, nicht aber die genaue Kennzeichnung des Gewindes. Ein Experte wird jedoch niemals absolut irrelevante Features nennen. Es ist daher sinnvoll, gewichtete Wahrscheinlichkeiten einzuführen, die über die Relevanz eines Features aussagen.

#### Der Beispiel Erklärungsansatz

Im Bereich des Maschinenbaus kann nicht davon ausgegangen werden, daß Problemklassen ausschließlich durch eine Menge von Beispielen repräsentiert sind. Dagegen ist anzunehmen, daß Experten ihre Problemklassen mit zugehörigen Lösungsstrategien aufgrund ihrer Erfahrungen in der Fertigung konkreter Werkstücke entwickeln. Die Erhebung der Begriffe eines Experten muß diesen Stellenwert der Beispiele berücksichtigen, und sollte ausgehend von Beispielen durchgeführt werden.

#### Der Prototyp Erklärungsansatz

Ähnlich dem Beispiel Erklärungsansatz kann auch der Prototyp Ansatz nicht alleine die Begriffsbildung im Bereich des Maschinenbaus erklären. Es ist jedoch wahrscheinlich, daß Experten mit jeder Problemklasse eine Art Prototyp assoziieren, der wie ein Skelett in jedem Fertigungsproblem dieser Klasse enthalten ist. Die Zuordnung eines neuen Fertigungsproblems beruht nicht nur auf einem Vergleich mit der Menge definierender Features, sondern wird auch durch einen Vergleich mit einem implizit existierenden Prototyp vorgenommen.

#### Der theoretische Erklärungsansatz

Wie die oben dargestellten Untersuchungen zur Begriffsbildung eines Experten in verschiedenen Domänen zeigen, entstehen die Begriffe eines Experten immer unter Berücksichtigung des theoretischen Hintergrundwissens und allgemeiner Lösungsstrategien.

Zusammenfassend läßt sich keine eindeutige Aussage machen, welcher Ansatz aus der allgemeinen Begriffsbildungsforschung angemessen die Bildung von Problemklassen im Bereich des Maschinenbaus erklären kann. Es liegt die Vermutung nahe, daß nicht ein einziger Ansatz alleine die Phänomene der Problemklassen erklärt, sondern ein hybrider Ansatz verwendet werden muß.

Die Begriffe der Problemklassen werden ausgehend von einer Menge von Beispielen gebildet. Anschließend können die Problemklassen mit Hilfe definierender Features unterschiedlicher Relevanz erklärt werden. Mit jeder Problemklasse wird ein Prototyp verbunden und eine Lösungsstrategie für die Fertigung der Probleme einer Klasse basierend auf theoretischem Hintergrundwissen assoziiert.

## IV CASE EXPERIENCE COMBINATION SYSTEM (CECoS)

### IV.1 Überblick

Das Wissensakquisitionswerkzeug CECoS ist kein automatisches Begriffsbildungssystem wie beispielsweise der GCC-Algorithmus [Maarek, 1990], UNIMEM [Lebowitz, 1986; Lebowitz, 1987], COBWEB [Fisher, 1987], RESEARCHER [Lebowitz, 1986], CARL [Burstein, 1983], und CLUSTER/2 [Michalski, Stepp, 1983a; Michalski, Stepp, 1983b]. Diese Systeme führen alle ein inkrementelles hierarchisches konzeptuelles Clustern ausgehend von einer Anzahl von Beispielen durch oder ermitteln neue Konzepte durch Generalisierung aus Beispielen.

Das Verfahren von CECoS geht davon aus, daß ein Experte in seiner Domäne Begriffe herausgebildet hat, die er mit Klassen von Fertigungsproblemen assoziiert. Diese Begriffe des Experten sollen erhoben werden. Dabei werden die Begriffe aufbereitet und gegebenenfalls leicht modifiziert.

Ausgehend von einer Menge von beispielhaften Fällen wird aufgrund von Ähnlichkeitsurteilen eines menschlichen Experten eine Begriffshierarchie durch ein hierarchisches Clusterverfahren erzeugt. Anschließend wird diese extensionale Definition durch die Nennung von Features intensional erweitert. Durch eine Reihe von Redundanz- und Konsistenztests wird die Hierarchie überprüft und schließlich schrittweise formalisiert. Auf diese Weise wird eine formale Wissensbasis erzeugt, die von einem Expertensystem weiterverwendet werden kann.

Die im Akquisitionsteil des ARC-TEC-Projektes entwickelte integrierte Wissensakquisitionsmethode gliedert sich in vier Episoden. In der Erklärungs-episode sollen die Begriffe der Problemklassen erklärt

ausgewählten Beispielfälle nicht zu groß ist, und sich der Vorteil der Voreinteilung nicht wesentlich auswirken wird.

Nach der Voreinteilung der Beispielfälle in Klassen wird der Experte aufgefordert, zu jedem Paar zweier Klassen ein Ähnlichkeitsurteil abzugeben. Das Ähnlichkeitsurteil wird auf einer numerischen diskreten Skala zwischen eins (sehr unähnlich) und sieben (sehr ähnlich) durchgeführt. Die grundsätzliche Struktur der Beispiele ist gleich, so daß bei einem Paarvergleich keine Klasse dominierend ist. Damit kann die Ähnlichkeit in diesem Falle als symmetrisch angenommen werden. Es ergibt sich nach diesem Schritt eine obere (bzw. untere) Dreiecksmatrix mit Ähnlichkeitsmaßen zwischen eins und sieben.

Welchen Wert der Experte für die Ähnlichkeit zweier Klassen angibt, liegt in seinem Ermessen. Es wird keine Berechnungsvorschrift für die Bestimmung der Ähnlichkeit vorgegeben. Da einerseits davon ausgegangen werden kann, daß die Vorstellung eines Beispiels bei einem Experten äquivalente Merkmale auslöst wie die konkrete Fertigung [Hoffmann, 1988], und andererseits die Lösung des Problems bei einem Experten immer mit der Problemklasse repräsentiert ist [Chi, Feltovich, Glaser, 1981], ergibt sich eine sinnvolle Ähnlichkeitsbeurteilung im Sinne ähnlicher Fertigungspläne zur Lösung der Beispielfälle.

#### Clusteranalyse

Die Ähnlichkeitsmatrix wird durch ein hierarchisches Clusterverfahren ausgewertet [Backhaus, Erichson, Plinke, Schuchhard-Fischer, Weiber, 1987; Baind, Noma, 1978; Borg, Staufenbiehl, 1989]. Im Gegensatz zu anderen Clusterverfahren wie graphentheoretische, partitionierende oder Optimierungsverfahren haben hierarchische Clusterverfahren den Vorteil, daß sie nicht alle Kombinationen möglicher Klasseneinteilungen betrachten müssen. Dadurch wird die Komplexität erheblich verringert. Bei den hierarchischen Clusterverfahren unterscheidet man agglomerative und divisive Verfahren. Agglomerative Verfahren gehen von  $N$  Klassen aus, die jede genau ein Element enthalten <sup>4</sup>, und führen diese sukzessive zu größeren Klassen zusammen, während die divisiven Verfahren von einer Klasse, in der alle Objekte enthalten sind, ausgehen, und diese schrittweise in kleinere Klassen zerteilen. Da sich bei divisiven Verfahren auf jeder Stufe der Höhe  $k$   $2(N/k)-1$  Teilungsmöglichkeiten ergeben, zum Beispiel bei  $N=60$  und  $k=1$  bereits

Produktionsplan für die Fertigung der Probleme einer Klasse relevant ist, oder der Erkennung planrelevanter Features dient.<sup>5</sup> Darüberhinaus kann der Experte jede beliebige natürlichsprachliche Formulierung wählen, die ihm angemessen erscheint. Es ist bei der Angabe der Features zu beachten, daß innerhalb der Hierarchie das Subsumptionsprinzip gilt. Das Subsumptionsprinzip sagt aus, daß jedes genannte Feature automatisch an alle Nachfolgerknoten weitervererbt wird. In diesem Sinne muß der Experte berücksichtigen, daß ein genanntes Feature sich auf alle darunter liegenden Fälle bezieht. Es wird weder gefordert, daß alle einzelnen Features notwendig, noch daß die Features zusammen hinreichend sind. Die Features werden unterschiedlich stark relevant sein, so daß eine Gewichtung nach dem probabilistischen Ansatz denkbar ist. Die genannten Features skizzieren eine Art Prototyp der Problemklasse. Auf diese Weise kombiniert man prinzipiell die vier typischen Erklärungsansätze der menschlichen Begriffsbildung.

Im Maschinenbau kann man erwarten, daß Problemklassen im Bereich der rotationssymmetrischen Drehteile ähnlich genau definiert sind wie zum Beispiel geometrische Formen in der Mathematik, so daß eine Definition der Begriffe durch Merkmale sinnvoll ist.

Die Reihenfolge der ersten Schritte ist dadurch bestimmt, daß die Charakterisierung der Problemklassen durch Features erst nach der Bildung einer Begriffshierarchie vorgenommen werden sollte [Rosch, 1978].

#### Zuordnung zu den Sichten

Das Wissen der Anwendungsdomäne ist durch das 'Modell der Expertise' strukturiert [Schmalhofer, Kühn, Schmidt, 1991]. Ein weiteres Ziel der Erklärungsepisode besteht in einer Assimilation der Wissenseinheiten, in diesem Falle der Features, in dieses 'Modell der Expertise'. Zu diesem Zweck definiert man verschiedene Sichten auf die Sprache des Maschinenbaus. Jede Sicht entspricht einer Menge von Termen, die in einem Teilbereich der Domäne von Experten verwendet werden. Diese Teilbereiche ergeben sich unmittelbar aus dem 'Modell der Expertise'. Man unterscheidet Werkstück-, Werkstatt-, und Planwissen. Die Teilbereiche unterscheiden sich weiterhin in ihrem Abstraktionsgrad. Es gibt konkretes und abstraktes Wissen. Mit konkretem Wissen beschreibt man zum Beispiel genaue Eigenschaften eines Werkstückes, oder einer Werkstatt. Durch Kombination eines Abstraktionsgrades und eines Teilbereiches der Domäne ergeben sich insgesamt sechs verschiedene einfache Sichten: 'konkrete Werkstück-Sicht', 'abstrakte Werkstück-Sicht', 'konkrete Werkstatt-Sicht', 'abstrakte Werkstatt-Sicht', 'konkrete Plan-Sicht' und 'abstrakte Plan-Sicht'. Jede dieser einfachen Sichten entspricht genau einem Rechteck des 'Modells der Expertise' (vgl. auch Abbildung 8). Nicht jeder Begriff, der im Maschinenbau verwendet wird, bezieht sich genau auf einen Teilbereich. Die einzelnen Teilbereiche interagieren und hängen wechselseitig voneinander ab. Daher führt man zusätzlich zu den einfachen Sichten kombinierte Sichten ein, die jeweils aus der Kombination von einfachen Sichten bestehen. Beispielsweise sind 'konkrete Werkstück-, abstrakte Werkstatt-Sicht' und 'konkrete Werkstück-, konkrete Werkstatt-, konkrete Plan-Sicht' kombinierte Sichten. Insgesamt ergeben sich also  $\sum_{k=1}^6 \binom{6}{k} = 63$  mögliche Sichten. Nicht alle möglichen Sichten sind sinnvoll. Zum Beispiel ergibt die Kombination aller sechs einfachen Sichten keine sinnvolle kombinierte Sicht.

Bei der Zuordnung zu den Sichten soll der Experte nun für jedes seiner genannten Features ankreuzen, zu welcher Sicht es gehört. Grundsätzlich gilt dabei, daß ein Feature genau dann konkret ist, wenn es sich auf die Beschreibung einer Instanz einer Klasse bezieht, und genau dann abstrakt ist, wenn es sich auf eine ganze Klasse beliebigen Abstraktionsgrades, zu der mehr als ein Objekt gehört, bezieht. Das Feature 'Werkstücklänge gleich 500 mm' ist zum Beispiel konkret, weil die Länge festgelegt ist, und keine weitere Instanziierung mehr ermöglicht. Das Feature 'Werkstücklänge größergleich 500 mm' ist dagegen abstrakt, da eine Instanz der Länge nicht vorgegeben wird. Der Experte darf auch mehrere Kreuze machen und ordnet das Feature damit der entsprechenden kombinierten Sicht zu.

Durch die Zuordnung der einzelnen Wissenseinheiten zu den Sichten wird das verwendete Wissen partitioniert. Das Wissen wird durch diese Partitionierung mit einer Struktur versehen, die den Zugang zu einzelnen Wissenseinheiten leichter und die weitere Verwendung des gesamten Wissens übersichtlicher gestaltet. Jede Sicht sollte für sich gesehen eine sinnvolle und abgeschlossene Menge von Termen bilden. Das Wissen jeder Sicht bildet eine unabhängige Teilwissensbasis. Die weitere Formalisierung der gesamten Wissensbasis ist damit modularisiert, und in Teilaufgaben getrennt. Diese Modularisierung erleichtert die weiteren Episoden, indem der Aufwand eingeschränkt wird. Jede Teilaufgabe ist insgesamt einfacher zu bearbeiten als die Gesamtaufgabe.

---

<sup>5</sup> Die Bezeichnung Feature wird hier in einem allgemeineren Sinne verwendet als beispielsweise in [Klauck, Bernardi, Legleitner, 1991].

## IV.2.2 Wissensvergleichsepisode

Die Zuordnung der Wissensseinheiten zu den verschiedenen Sichten dient zunächst der Strukturierung der Wissensvergleichsepisode. In der Wissensvergleichsepisode werden die genannten Features mit Hilfe von Redundanz- und Konsistenztests verglichen.

### Redundanz- und Konsistenztests

Durch die Durchführung von Redundanz- und Konsistenztests soll vermieden werden, daß zum Beispiel Doppelnennungen, die wörtlich, aber auch nur sinngemäß, übereinstimmen, auftreten, oder Inkonsistenzen enthalten sind, etwa widersprüchliche Definitionen verschiedener Merkmale, die sich gegenseitig ausschließen, aber wegen der Subsumptionseigenschaft zusammen gehören.

Die Tests werden für Pfade des Hierarchiebaumes, einzelne Klassen und gemeinsame Nachfolgerklassen durchgeführt. Eine typische Redundanz ist beispielsweise, wenn ein identisches Feature für alle Nachfolgerklassen einer Klasse genannt wird. Diese Nennungen können gestrichen und auf die abstraktere Klasse übertragen werden. Eine typische Inkonsistenz tritt auf, wenn ein Feature einer Klasse in einer Nachfolgerklasse negiert auftritt. Da sich nach dem Subsumptionsprinzip alle Features auf die Nachfolgerklassen vererben, entsteht ein klassischer Widerspruch. Eine genauere Definition der einzelnen Tests findet sich in [Tschaischian, 1991].

Die einzelnen Tests werden zunächst für jede einfache Sicht getrennt durchgeführt. Die Idee dieser Trennung liegt darin, daß nur Wissensseinheiten aus einem gemeinsamen Teilbereich der Wissensbasis zueinander redundant oder inkonsistent sein können. Beispielsweise kann ein Feature, das sich nur auf die Beschreibung des Werkstückes bezieht, nicht zu einem Feature im Widerspruch stehen, das sich ausschließlich auf die Werkstatt bezieht. Um den Vorteil dieser Einteilung nach Sichten zu verdeutlichen, soll ein einfaches Beispiel konstruiert werden. Angenommen es gäbe genau einen Redundanz- und einen Konsistenztest. Weiter seien bei der Nennung der Features 30 Wissensseinheiten gebildet worden. Führt man nun die beiden Tests ohne eine Strukturierung der Wissensseinheiten durch, so muß in beiden Tests theoretisch jedes Paar von Wissensseinheiten untersucht werden. Dann ergeben sich 435 Vergleiche für jeden Test, bei zwei Tests also 870 Vergleiche. Sind die Wissensseinheiten dagegen durch die Zuordnung zu den Sichten partitioniert, so senkt sich der Aufwand erheblich. Idealisierend sei hier einmal angenommen, daß die Features nur einfachen Sichten zugeordnet wurden; zu drei einfachen Sichten seien jeweils 10 Features genannt. Die beiden Tests können nun für jede einfache Sicht getrennt durchgeführt werden. Für jede Sicht müssen dann nur noch 45 Untersuchungen für jeden Test gemacht werden. Bei drei Sichten und zwei Tests ergeben sich 270 Vergleiche. Durch die Strukturierung der Wissensseinheiten nach Sichten konnte der Vergleichsaufwand damit auf ein Drittel reduziert werden gegenüber der Durchführung der Tests ohne Modularisierung des Problems. Da in CECoS nicht nur zwei Tests durchgeführt werden müssen, wirkt sich der Vorteil der Strukturierung nach Sichten weiter verstärkt aus.

Nachdem die Tests für alle einfachen Sichten durchgeführt worden sind, kann in gleicher Weise mit den kombinierten Sichten vorgegangen werden.

Zum Schluß dieser Episode müssen noch die Redundanzen und Inkonsistenzen zwischen einfachen Sichten und kombinierten Sichten, in denen eine der entsprechenden einfachen Sichten enthalten ist, betrachtet werden. An dieser Stelle sind ebenfalls Redundanzen und Inkonsistenzen möglich, obwohl es sich um verschiedene Sichten handelt. Zum Beispiel kann ein Feature, das sich auf die absolute Länge eines Werkstückes bezieht und der einfachen Werkstück-Sicht zugeordnet wurde, durchaus zu einem Feature, das die Länge relativ zu einer Maschine beschreibt und der kombinierten Werkstück-, Werkstatt-Sicht zugeordnet wurde, im Widerspruch stehen. Beide Features beziehen sich auf die Länge des Werkstücks und haben damit einen gemeinsamen Inhalt. Dieses einfache Beispiel erschöpft allerdings noch nicht die möglichen Redundanzen und Inkonsistenzen, die zwischen einfachen und kombinierten Sichten auftreten können. Man kann versuchen die Ursachen solcher Redundanzen und Inkonsistenzen, wie zum Beispiel Bezug zu einer gemeinsamen Größe, zu ermitteln, um die Tests zwischen einfachen und kombinierten Sichten auf eine Teilmenge aller Möglichkeiten einzuschränken.<sup>6</sup>

Man spricht bei der Eliminierung von Redundanzen und Inkonsistenzen auch von einer frühen Wissensverifikation, da durch die Tests das Wissen bereits auf einer informalen Ebene verifiziert werden kann.

---

<sup>6</sup> An dieser Stelle soll auf dieses Problem nicht vertieft eingegangen werden, sondern lediglich auf dieses Problem der möglichen Redundanzen und Inkonsistenzen zwischen einfachen und kombinierten Sichten hingewiesen werden.

### IV.2.3 Kompetenzabschätzungsepisode

In der Kompetenzabschätzungsepisode wird die Kompetenz des Expertensystems, das die Problemklassen als Teil seiner Wissensbasis verwenden wird, bereits auf einer informalen Ebene beurteilt. Die Kompetenz des Systems ist unmittelbar durch die Erklärungen der einzelnen Problemklassen gegeben. Zunächst ist das Expertensystem für die Menge der ausgewählten Beispiele kompetent. Jedes Beispiel kann wieder einer der Problemklassen zugeordnet werden. Da die Problemklassen aus diesen Beispielfällen ermittelt wurden, wird ein Beispielfall einer Problemklasse zugeordnet werden können, die auf unterster Ebene der Problemklassenhierarchie liegt. Der mit dieser Klasse assoziierte Plan ist ein konkreter Produktionsplan und kann ohne zusätzlichen Aufwand zur Lösung des Fertigungsproblems angewendet werden. Für diese Beispielfälle ist das Expertensystem in hohem Maße kompetent. Darüberhinaus ist das Expertensystem in der Lage, Lösungen für ähnliche Probleme zu generieren. Es ist zu erwarten, daß je ähnlicher ein neues Problem zu einem der ausgewählten Beispielfälle ist, desto größer wird die Kompetenz des Systems für dieses neue Problem sein. Für ein neues Problem wird zunächst versucht, eine Zuordnung zu einer der Problemklassen vorzunehmen. Gehört das neue Problem bereits nicht zu der mit der Wurzel assoziierten Problemklasse, so ist das Expertensystem für dieses neue Problem nicht kompetent. Erfüllt das neue Problem dagegen die Erklärung dieser Problemklasse, so ist das System für das neue Problem kompetent. Das Expertensystem ist also für alle neuen Probleme kompetent, die der Problemklasse der Wurzel der Begriffshierarchie zugeordnet werden können. Das Maß der Kompetenz für ein neues Problem entscheidet sich dann aufgrund der Tiefe der Problemklasse in der Hierarchie, der das neue Problem zugeordnet wird. Je tiefer die Problemklasse in der Hierarchie liegt, desto genauer ist das Problem klassifiziert. Je genauer das Problem klassifiziert ist, desto konkreter ist der zugehörige Skelettplan. Der Aufwand der Lösung eines Problems wird mit konkreter werdenden Plänen geringer. Das System ist also kompetenter, wenn das neue Problem einer Problemklasse zugeordnet werden kann, die weiter unten in der Hierarchie auftritt, und weniger kompetent bei Problemen, die zu einer Klasse einer oberen Ebene gehören. Man spricht in diesem Falle der Kompetenzabschätzung von gradueller Kompetenz.

Der Bereich der Kompetenz des Expertensystems kann auf diese Weise eingeschätzt werden und gegebenenfalls bei nicht ausreichender Kompetenz zu einer erneuten Durchführung des Verfahrens führen, um die Kompetenz des Systems zu erhöhen.

### IV.2.4 Formalisierungsepisode

In der Formalisierungsepisode wird jede Wissenseinheit in eine formale Repräsentation übersetzt und in der formalen Wissensbasis gespeichert. Durch die Formalisierung will man eine präzise und genaue Definition der einzelnen Problemklassen erreichen, um diese als Wissensbasis in einem Expertensystem weiterverwenden zu können. Für die automatische Weiterverarbeitung einer Wissensbasis ist es notwendig, eine Formalisierung vorzunehmen, die eine Repräsentationssprache einsetzt, die von einer Maschine bearbeitet werden kann. Die Formalisierung geschieht in zwei Schritten. Der erste Schritt soll die eigentliche Formalisierung vorbereiten, indem bereits die spätere Struktur der Repräsentation angestrebt wird.

Die formale Definition von Problemklassen sollte insgesamt berücksichtigen, wie der menschliche Experte solche Problemklassen versteht. Dabei will man eine Formalisierung erreichen, die dem



Sprachen", oder "Konzeptbeschreibungssprachen" bezeichnet.

Das Basiskonstrukt einer jeden Termbeschreibungssprache ist die Einführung und gegebenenfalls Definition von Begrifflichkeiten durch Angabe der notwendigen und gegebenenfalls hinreichenden Bedingungen, die diesen Begriff unter Rückgriff auf schon eingeführte Begriffe konstituieren. Jede Termbeschreibungssprache besitzt eine modelltheoretische Semantik, die es - analog zur klassischen Logik - erlaubt, die Bedeutung so eingeführter Begriffe kompositional zu errechnen.

In terminologischen Wissensrepräsentationssprachen unterscheidet man zwei grundsätzliche Komponenten, die der 'Terminological Box' (TBox) und die der 'Assertional Box' (ABox) [Hanschke, Abecker, 1991].

In der TBox werden die Begriffe beschrieben, die zur Definition der Domäne verwendet werden sollen. Diese Begriffsbeschreibung geschieht zunächst einerseits durch die Definition von atomaren Konzepten und andererseits durch die Definition von Beziehungen, die zwischen verschiedenen Konzepten bestehen können. Diese Relationen zwischen Konzepten werden traditionell Rollen genannt [Brachman, Schmolze, 1985]. Eine spezielle Art Rolle stellen die einwertigen Prädikate dar, die man als Attribute bezeichnet. Der wesentliche Unterschied zwischen Rollen und Attributen besteht darin, daß eine Rolle mehrere 'Füller' haben kann, ein Attribut dagegen nur einen. Ein solches atomares Konzept könnte zum Beispiel in einer Anwendung zur Simulation von Familienhierarchien das Konzept MENSCH oder MÄNNLICH sein. Ein Beispiel für eine Rolle ist KIND, eine Relation, die zwischen einem Elternteil und einem weiteren Menschen besteht. Ein Attribut könnte zum Beispiel das ALTER eines Menschen sein. Der Unterschied zwischen Rolle und Attribut läßt sich an diesen Beispielen illustrieren. KIND ist als Rolle deklariert, da ein Mensch mehrere Kinder haben kann, das ALTER dagegen wird als Attribut definiert, da ein Mensch nur ein Alter hat. Mit Hilfe dieser atomaren Beschreibungselemente lassen sich nun komplexere Terme definieren. Zur Formulierung solcher komplexeren Begriffe stehen Verknüpfungen zur Verfügung, wie

unentscheidbaren Subsumptionsproblemen [Baader, Hanschke, 1991b].

Da man allerdings erwartet, daß terminologische Systeme in einem größeren System eingebettet verwendet werden, schlagen Baader und Hanschke [Baader, Hanschke, 1991b] vor, solche Anfragen in einem anderen Teil des Gesamtsystems zu bearbeiten. Ein Versuch dieser Realisierung findet sich in dem System COLAB, in dem man bemüht ist, verschiedene Teilsysteme unter einer Anwendung im Maschinenbau zu vereinen [Boley, Hanschke, Hinkelmann, Meyer, 1991].

Es gibt verschiedene Gründe für die Wahl einer terminologischen Wissensrepräsentationssprache als Formalisierungssprache.

Zunächst kann eine direkte Abbildung der Begriffe auf Konzepte erfolgen. Ferner entsprechen die möglichen Inferenzen einer terminologischen Wissensrepräsentationssprache genau den Identifikationsaufgaben der Begriffsbildung. Die Klassifizierung einer Subsumptionshierarchie bildet die Begriffshierarchie. Subsumptionsanfragen der Form 'subsumiert Konzept<sub>x</sub> Konzept<sub>y</sub>?' entsprechen der Analyse von Ober- und Unterbegriffsrelationen. Die Benennung erfolgt durch Realisierungsanfragen eines bestimmten Objektes der ABox. Die Zuordnung kann mit Hilfe von Kontrollen, ob eine bestimmte Instanz der ABox ein bestimmtes Konzept der TBox erfüllt, gelöst werden. Und schließlich wird die Entdeckung durch Erfüllbarkeitsanfragen realisiert.

Die für eine weitere Anwendung der formalen Wissensbasis innerhalb eines Expertensystems notwendigen

Aufgaben werden durch eine terminologische Wissensrepräsentationssprache gut erfüllt.

#### Semiformalisierung

Der die Formalisierung vorbereitende Schritt wird Semiformalisierung genannt. Man gibt eine Äquivalenz vor, die der Äquivalenzbeziehung bei Konzeptdefinitionen zwischen beiden Seiten des Gleichheitszeichens entspricht. Auf der linken Seite steht das genannte Feature und auf der rechten Seite erfolgt die genauere Definition des Features. Die Zuordnung zu den Sichten spiegelt die Strukturierung der Fachsprache der Expertise wider. Man behält diese Strukturierung bei und kann so eine Einteilung der Sprache in verschiedenen Mengen von Termen vornehmen. Somit leitet die Einteilung der Wissensseinheiten nach den verschiedenen Sichten die Episode der Formalisierung. Ähnlich wie bereits bei der Wissensvergleichsepisode modularisiert die Zuordnung zu den Sichten die Aufgabe der Formalisierung. Die Formalisierung der Features kann für jede Sicht unabhängig von anderen Sichten durchgeführt werden. Dadurch entstehen Teilaufgaben der Formalisierung, die wesentlich übersichtlicher zu bearbeiten sind als die Formalisierungsaufgabe insgesamt. Neben der Modularisierung der Aufgabe der Formalisierung kann durch die Zuordnung der Features zu den Sichten eine Struktur vorgegeben werden, die eine Formalisierung erleichtert. Die Sichten entscheiden, zu welchem Teil der Wissensbasis die formalisierte Wissensseinheit gehört, und geben ferner an, welche Teile des Domänenwissens bei der formalen Definition einer Einheit verwendet werden dürfen. Aus diesen Überlegungen ergibt sich eine Schablone, die dem Experten bei der Semiformalisierung vorgegeben wird.

Eine Schablone hat die Form 'Feature(Sicht<sub>1</sub>, ..., Sicht<sub>n</sub>) = "\_\_\_"'. Für jedes genannte Feature wird eine solche Schablone erstellt. Nannte der Experte beispielsweise das Feature 'langes Werkstück' und ordnete dieses der 'abstrakten Werkstück-Sicht' zu, so gibt man die Schablone 'langes\_Werkstück(abstrakte Werkstück-Sicht) = "\_\_\_" vor. Der Experte wird nun aufgefordert, die Lücke der Schablone zu füllen. Die Lücke soll mit einer feineren Beschreibung des genannten Features gefüllt werden. In der genaueren Beschreibung dürfen nur Terme der angekreuzten Sichten verwendet werden. Dabei darf keine angekreuzte Sicht unberücksichtigt bleiben. Die Anzahl der zu verwendenden Terme jeder vorgegebenen Sicht ist nicht festgelegt. Verwendet der Experte einen Term aus einer nicht angekreuzten Sicht, oder hat der Experte eine

kleiner als 20 mm ist." heißen. Das Feature 'langes Werkstück' gewinnt damit eine relative Bedeutung bezogen auf das Werkstück und die Maschine. Bei dem Feature 'Aufspannung mit Stirnseitenmitnehmer' erwartet man zum Beispiel keine weitere Beschreibung, da dies ein feststehender Begriff im Bereich des Maschinenbaus ist, der die Aufspannungsart festlegt. Dieses Feature wird dann als atomarer Term in der formalen Definition der Problemklasse verwendet.

Jeder Term, der bei der genaueren Beschreibung eines Features verwendet wird, kann als eigenes Feature einer niedrigeren Abstraktionsebene aufgefasst werden. Daher wiederholen sich die Schritte der Zuordnung zu den Sichten, der Redundanz- und Konsistenztests, und der Semiformalisierung, bis keine neuen Terme hinzugenommen werden, d.h. nur noch atomare Terme auftreten.

#### Formalisierung

Im letzten Schritt werden nun schließlich die Problemklassen und die verwendeten Features formalisiert. Jede Problemklasse stellt ein Konzept dar, dessen Definition eine Konjunktion der genannten Features ist. Dabei werden die Features in ihrer Relevanz zunächst nicht gewichtet. Es ist denkbar an dieser Stelle Gewichte für jedes Feature einzuführen, die widerspiegeln, wie relevant ein Feature für die Definition einer Problemklasse ist. Dies entspricht dem probabilistischen Ansatz. Jede Schablone wird in eine Konzeptdefinition umgesetzt. Das genannte Feature entspricht dem Konzeptnamen. Die Zuordnung zu den Sichten bestimmt die Teilsprache der Expertise, der dieses Konzept zugeordnet wird. Die genauere Beschreibung auf der rechten Seite der Schablone ergibt die Definition des Konzeptes. Die Schablone 'langes\_Werkstück(abstrakte Werkstück-Sicht) = "Ein Werkstück ist lang, wenn es eine Länge von mindestens 500 mm hat."' wird beispielsweise umgesetzt in `langes_Werkstück = groessergleich_500 (ws länge)`. Dabei ist 'langes\_Werkstück' der Konzeptname, 'groessergleich\_500' ein Prädikat der konkreten Domäne der reellen Zahlen, und '(ws länge)' eine Attributkette der Attribute 'ws' und 'länge'. Die Probleme der Fertigung können nun in eine geeignete Repräsentation übersetzt und als Individuen in der ABox zugesichert werden. Die Aufgabe des Inferenzsystems besteht dann in der Zuordnung eines Problems in die Subsumptionshierarchie, die der Begriffshierarchie der Problemklassen entspricht.

## V DIE ANWENDUNG: 60 FÄLLE

Im Rahmen dieser Arbeit ist das Wissensakquisitionsverfahren CECoS auf eine Menge von 60 Problemfällen angewendet worden. Jeder einzelne Problemfall ist grundsätzlich durch drei Komponenten beschrieben: die Geometrie, den Werkstoff, und die Maschine. Diese drei Komponenten entscheiden maßgeblich und im wesentlichen die Auswahl des verwendeten Bearbeitungsverfahrens bei der Fertigung eines Werkstückes [Steinmüller, 1987, S. 117]. Bei der Auswahl der Problemfälle strebte man eine möglichst repräsentative Erfassung typischer Geometrien, Werkstoffe und Maschinen an. Die Erweiterung von bisher im Projekt betrachteten fünf Fällen auf 60 Fälle ist einerseits durch eine breitere Abdeckung des Fertigungsspektrums und andererseits durch eine andere Untersuchung zur Erhebung von Skelettplänen [Thoben, Schmalhofer, Reinartz, 1991], die auf der selben Menge von Fällen basierte, motiviert worden.

### V.1 Auswahl der Fälle

#### V.1.1 Geometrien

Unter der Geometrie eines Werkstückes versteht man die Merkmale der Kontur, sowie Eigenschaften spezieller Konturelemente. In den meisten Fällen wird eine Geometrie durch eine CAD (Computer Aided Design) - Zeichnung repräsentiert, die zusätzlich Angaben über die genauen Bemassungen und gegebenenfalls über Toleranzen (z.B. h6) enthält. Die ausgewählten Geometrien umfassen zwei Antriebswellen (g1 und g2), eine Ritzelwelle (g3), und zwei Achswellen (g4 und g5), und sind in Abbildung 3 dargestellt. Die typischen Konturelemente Nut, Einstich, Mittelabsatz, Kragen, Zapfen, und Gewinde treten jeweils mindestens einmal auf, so daß die Probleme der Fertigung rotationssymmetrischer Drehteile angemessen berücksichtigt sind.

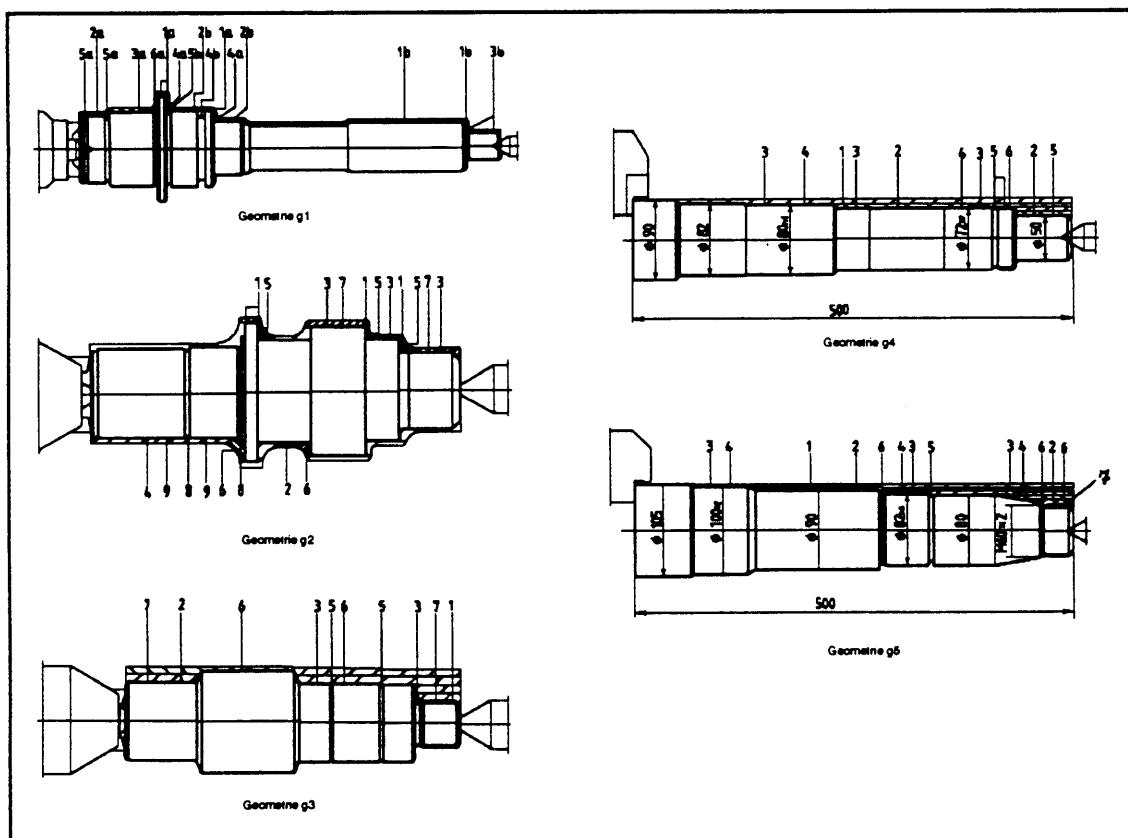


Abbildung 3: Die Geometrien g1, g2, g3, g4 und g5

#### V.1.2 Werkstoffe

Man unterscheidet im Maschinenbau drei Werkstoffgruppen: die Metalle, die Nichtmetalle, und die

Verbundwerkstoffe [Fischer, 1990; Domke, 1982]. Die Metalle sind ihrerseits in Eisenwerkstoffe, und Nichteisenmetalle untergliedert, und die Nichtmetalle in Naturwerkstoffe und Künstliche Werkstoffe. Des weiteren teilt man Eisenwerkstoffe in Stähle und Eisengußwerkstoffe ein, und Nichteisenmetalle in Schwermetalle und Leichtmetalle.

Stähle sind Eisenwerkstoffe mit großer Festigkeit, die vorwiegend zur Herstellung von Maschinenteilen verwendet werden, die Kräfte aufnehmen und übertragen müssen. Zwei Beispiele aus der Gruppe der Stähle sind Vergütungs- und Edelstahl. In der Untersuchung sind der unlegierte Vergütungsstahl C45 (w1) und der hochlegierte Edelstahl X5CrNi 18 9 (w4) verwendet worden.

Eisengußwerkstoffe sind gut vergießbare Werkstoffe, die meist zur Fertigung von Maschinenteilen verwendet werden, deren schwierige Form am besten durch Gießen herstellbar ist, die aber auch beim Drehen wegen ihrer guten Eigenschaften bezüglich des Spanbruchs eingesetzt werden. Ein typischer Eisengußwerkstoff ist Gußeisen. Ein spezieller Gußeisen GG25 (w2) ist in der Untersuchung vertreten.

Schwermetalle zeichnen sich durch eine Dichte größer als  $5 \text{ kg/dm}^3$  aus und werden meist wegen besonderer, werkstofftypischer Eigenschaften verwendet. Demgegenüber besitzen Leichtmetalle eine Dichte kleiner als  $5 \text{ kg/dm}^3$  und besitzen dennoch eine große Festigkeit. AlMgSi (w3) ist ein spezielles Aluminium und gehört zu der Gruppe der Leichtmetalle.

Natur-Werkstoffe, Künstliche Werkstoffe und Verbundwerkstoffe finden bei der Fertigung rotations-symmetrischer Drehteile in diesem Bereich des Maschinenbaus, der hier betrachtet wird, nicht so häufig eine Anwendung, so daß sie in der Untersuchung außer acht gelassen werden konnten.

### V.1.3 Maschinen

Die bei der Fertigung eingesetzten Maschinen spielen als Kontext des Prozesses eine entscheidende Rolle sowohl für die Planung der Fertigung an sich als auch für die Klassifizierung eines Problems im allgemeinen. Drehmaschinen werden vorwiegend in horizontaler Bauart in unterschiedlichen Baugrößen hergestellt. Bekannte Bauarten sind die Leit- und Zugspindel-, Revolver-, Plan- und Karusselldrehmaschinen und Drehautomaten. Haupt- und Vorschubgetriebe sind meistens gestuft, häufig aus Kupplungsgetrieben aufgebaut. Die Werkzeugaufnahme erfolgt in Mehrfachstahlhaltern oder Stern- bzw. Trommelrevolverköpfen. Die Werkstückaufnahme erfolgt auf Planscheiben, in Backenfuttern (Drehbackenfutter) in Spindelhalterköpfen, in Gabel- oder in Zangenhalterköpfen.

Skelettplan erzeugt worden ist. Die mit m1, m2, m3, m4 und m5 gekennzeichneten Zellen entsprechen den Fertigungsproblemen, die ursprünglich im Projekt betrachtet worden sind. Im folgenden wird ein spezieller Fall durch eine Kennzeichnung der Art giwjdk bezeichnet mit der Bedeutung, daß es sich bei diesem Fall um die Problemstellung mit der Geometrie gi, dem Werkstoff wj, und der Maschine dk handelt.

Geometrie		Werkstück																						
		Antriebswelle g1				Antriebswelle g2				Ritzelwelle g3				Achswelle g4				Achswelle g5						
		w1	w2	w3	w4	w1	w2	w3	w4	w1	w2	w3	w4	w1	w2	w3	w4	w1	w2	w3	w4			
d1	0	0	0	0	3	3	1	0	3	3	1	0	4	4	2	0	4	4	2	0	*	*	*	*
d2	5	5	7	9	5	5	7	9	5	5	7	9	5	5	8	10	5	5	8	10	*	*	*	*
d3	5	5	7	9	5	5	7	9	5	5	7	9	5	5	8	10	5	5	8	10	*	*	*	*

Abbildung 5: Die 60 Beispielprobleme

## V.2 Episoden

Bei der Anwendung von CECoS auf die 60 Problemfälle der Matrix sollte nachgewiesen werden, daß das

Nachteile ergeben.

Die sich aus dieser Einteilung ergebenden Klassen sind in Abbildung 5 durch Numerierung der Zellen der Matrix dargestellt. Alle Problemfälle, die in der zugehörigen Zelle mit der selben Ziffer gekennzeichnet sind, gehören zu einer gemeinsamen Klasse.

Die Klasse 0 umfasst die Problemfälle, die der Experte als 'ungewöhnlich' klassifiziert hat. Die Problemfälle der Geometrie g1 mit einem beliebigen Werkstoff werden grundsätzlich nicht auf der Maschine d1 gefertigt, da für die Fertigung der Geometrie mindestens sechs Werkzeuge erforderlich sind, die Maschine d1 aber nur vier Werkzeuge zur Verfügung stellt. Ferner werden Problemfälle mit dem Werkstoff w4 bei beliebiger Geometrie nicht auf der Maschine d1 gefertigt, da die Anforderungen von Edelstahl an die Leistung und Stabilität einer Maschine von d1 nicht erfüllt werden. Die Einschränkungen begründen sich also auf den mangelnden Fähigkeiten der Maschine d1. Da d1 eine Ausbildungsmaschine ist, die meist verwendet wird, um Lehrlinge im Bereich des Drehens anzulernen, sind die Qualitäten dieser Maschine begrenzt.

Die restlichen 52 Problemfälle wurden zunächst aufgrund der Monotonie der Werkstückkontur weiter kategorisiert. Die Geometrien g1, g2 und g3 haben eine beidseitig ansteigende monotone Kontur, während die Geometrien g4 und g5 eine einseitig ansteigende monotone Kontur aufweisen.

Die einzelnen Werkstoffe können wegen ihrer zu verschiedenen Merkmale nicht mehr zusammengefasst werden. Der Experte entschied sich dennoch, die Werkstoffe w1 und w2 gemeinsam zu betrachten, da ähnliche Eigenschaften der beiden Werkstoffe bei der Fertigung dieser Konturen auf diesen Maschinen erwartet werden können.

Ferner können die Maschinen d2 und d3 grundsätzlich zusammengefasst werden, da beide Maschinen über die gleichen relevanten Fähigkeiten verfügen. Der Unterschied der beiden Maschinen besteht in der Anzahl der Revolver, aus dem im Falle der Maschine d3 die Möglichkeit resultiert, eine parallele Bearbeitung vornehmen zu können, wohingegen bei Maschine d2 nur eine sequentielle Bearbeitung möglich ist. Da sich die Bearbeitungsart nicht auf die Schnittreihenfolge auswirkt, sondern lediglich auf die spätere genaue Produktion, können Problemfälle, die sich nur in der Maschine zwischen d2 und d3 unterscheiden, als gleich oder zumindestens sehr ähnlich angenommen werden. Insbesondere bezeichnete der Experte die höhere Geschwindigkeit, die höhere Leistung und die größere Anzahl Werkzeuge, die die Maschine d3 zur Verfügung stellt, als für die Bearbeitung dieser Fälle nicht notwendig.

Insgesamt entstanden auf diese Weise elf Klassen, so daß 55 Paarvergleiche durchgeführt werden mußten.<sup>7</sup> Im weiteren Verlauf der Erhebungen wurden die Klassen genauso bezeichnet wie die einzelnen Fälle. Dabei werden zusammengefasste Merkmale als kombinierte Indizes gekennzeichnet. Die Klasse 5 wurde zum Beispiel mit g123w12d23 bezeichnet, mit der Bedeutung, daß in dieser Klasse die Geometrien g1, g2 und g3, die Werkstoffe w1 und w2, und die Maschinen d2 und d3 zusammengefasst sind. Die Anzahl der Beispiele einer Klasse ergibt sich aus der Multiplikation der Anzahl der einzelnen Indizes. Die Klasse 5 enthält demnach  $3 \cdot 2 \cdot 2 = 12$  Beispiele und ist damit die Klasse, die die meisten Problemfälle der Matrix enthält. Der Experte konnte dadurch bei der Abgabe seiner Ähnlichkeitsurteile immer auf einen implizit existierenden Prototyp zurückgreifen. Es wurde kein Prototyp explizit festgelegt oder definiert, um dem Experten an dieser Stelle einen größeren Freiraum zu gewähren.

Der Experte gab nun zu jeweils paarweise dargebotenen Klassen ein Ähnlichkeitsurteil auf der Skala eins (sehr unähnlich) bis sieben (sehr ähnlich) ab. Zum Beispiel nannte er für das Paar der Klassen 1 und 2 das Ähnlichkeitsurteil sechs und für die Klassen 5 und 6 eins.

### Clusteranalyse

Nach Anwendung des hierarchischen Clusterverfahrens ergab sich der in Abbildung 6 dargestellte Hierarchiebaum. Die in der Abbildung 6 gewählte Darstellung spiegelt den Grad der Ähnlichkeiten wider. Die Länge der Äste ist ein Maß für die Distanz zwischen zwei Klassen. Zum Beispiel sind sich die Klassen 1 und 2 ähnlicher als die Klassen 5 und 6, da die Länge der Äste bis zur Zusammenfassung in eine Klasse bei den Klassen 1 und 2 kürzer ist als bei den Klassen 5 und 6. Dies entspricht den Ähnlichkeitsurteilen, die der Experte abgegeben hat. Die Klassen C und D unterscheiden sich nach den abgegebenen Ähnlichkeitsmaßen nicht. Eine Rückfrage an den Experten ergab aber, daß es trotzdem sinnvoll ist, die Trennung in zwei Klassen beizubehalten, da sich in der Klasse C ausschließlich Problemfälle mit den Werkstoffen w1 und w2 befinden, und in der Klasse D nur Problemfälle mit den Werkstoffen w3 und w4 enthalten sind. Eine sinnvolle Ausnahme bilden die Problemklassen 1 und 2, da sie aufgrund der eingeschränkten Möglichkeiten der Maschine d1 den Problemfällen mit den Werkstoffen w1 und w2 auf den Maschinen d2 und d3 in Klasse C ähnlicher sind als den Problemfällen mit den

<sup>7</sup> Ohne die Voreinteilung hätte man  $60 \cdot (60-1)/2 = 1770$  Paarvergleiche durchführen müssen.

Werkstoffen w3 und w4 auf den Maschinen d2 und d3 in Klasse D. Die Schnittbedingungen für den Werkstoff w3 können von der Maschine d1 nicht erfüllt werden. Die maximal von der Maschine d1 geleisteten Bedingungen sind daher den Schnittbedingungen der Fälle der Klasse C ähnlicher als denen der Klasse D. Der Experte sagte in diesem Zusammenhang aus, daß bei dieser speziellen Auswahl von Problemfällen der Werkstoff das dominierende Unterscheidungsmerkmal ist, da sich die Geometrien nicht wesentlich unterscheiden, und die Maschinen d2 und d3 gleich zu setzen sind, sofern man die Maschine d1 wegen ihrer Eigenschaft als Ausbildungsmaschine vernachlässigt. Allgemein läßt sich eine solche Dominanz eines speziellen Aspektes nicht angeben. Der längste Ast zwischen der Wurzel und Klasse 0 zeigt die zu erwartende Eigenschaft, daß 'ungewöhnliche' Problemfälle sehr unähnlich zu allen anderen Problemfällen sind. Der Experte editierte den Baum nicht. Er war mit dem Ergebnis zufrieden.

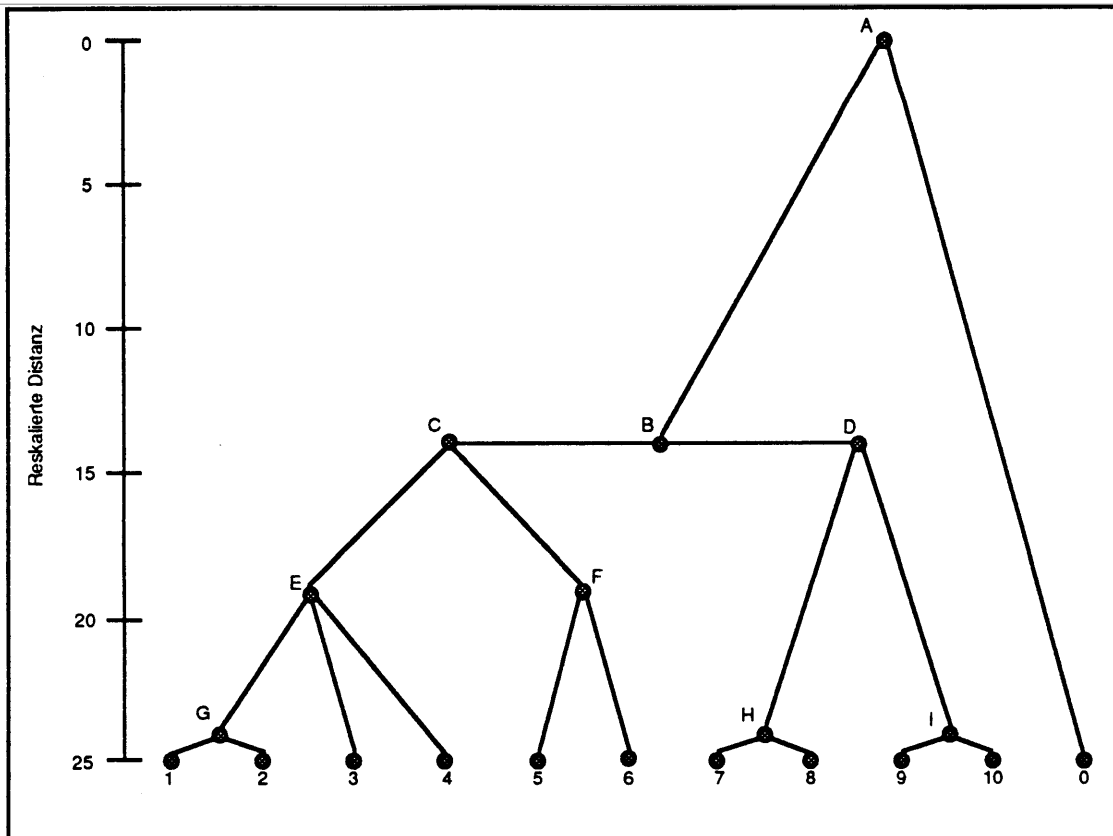


Abbildung 6: Begriffshierarchie der Problemklassen

Das selbe Verfahren wurde exemplarisch für die einzelnen Problemfälle von zwei Problemklassen durchgeführt. Die Auswahl der zwei entsprechenden Problemklassen geschah unter zwei Gesichtspunkten. Erstens berücksichtigte man die Klasse 6, da diese den speziellen Fall m5 (=g5w1d3) enthält, der im gesamten Projekt ARC-TEC einheitlich als Anwendungsbeispiel verwendet wird. Zweitens entschied man sich für die Klasse 5, da für diese bereits eine Reihe Skelettpläne erhoben wurden [Thoben, Schmalhofer, Reinartz, 1991], und dies großen Nutzen für parallele Forschungsabsichten haben kann. In Abbildung 5 sind die beiden Klassen durch Schattierungen kenntlich gemacht.

Die aus den Ähnlichkeitsurteilen mit dem Clusterverfahren erstellten Hierarchien der Klassen 5 und 6 sind in Abbildung 7 dargestellt. Diesmal wurde auf die aussagekräftigere Darstellungsart verzichtet. In beiden

Klassen wird zunächst nach den Werkstoffen w1 und w2 unterschieden. In Klasse 5 entscheidet danach die Art der Geometrie die Einteilung, in Klasse 6 dagegen die Art der Maschine, da sich die Geometrien g1, g2, und g3 einander nicht so ähnlich sind wie die Geometrien g4 und g5. Der Experte war auch hier mit den Ergebnissen zufrieden und editierte die beiden Bäume nicht mehr.



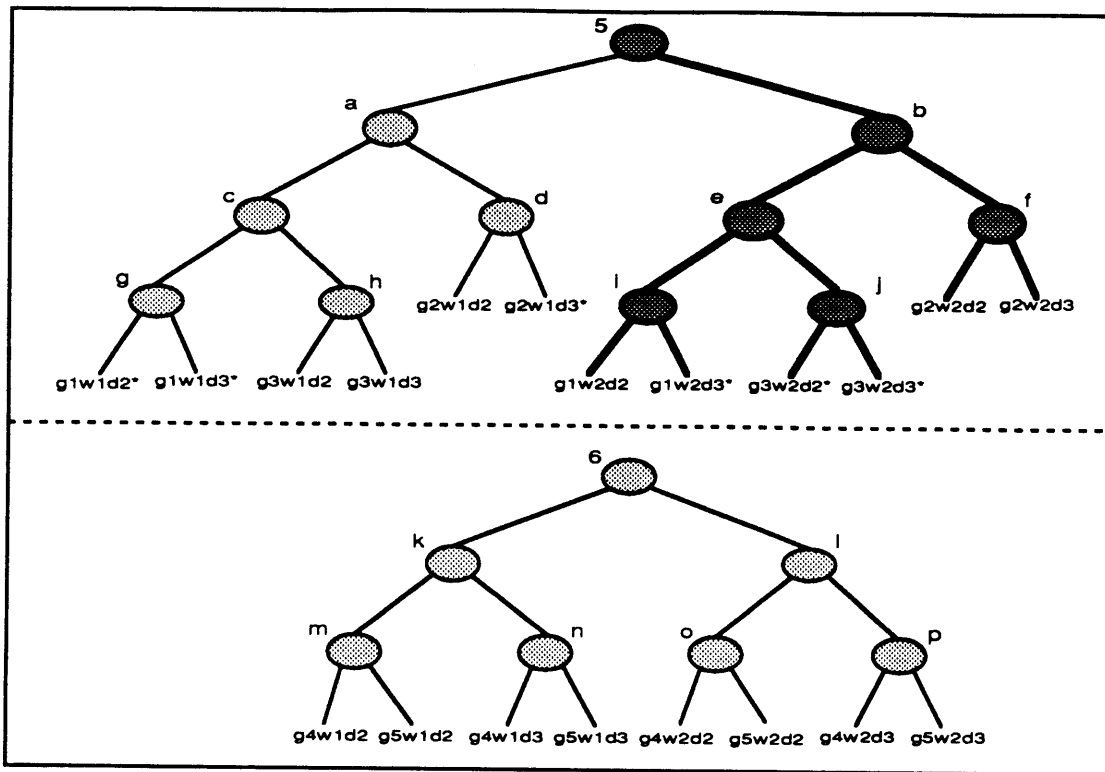


Abbildung 7: Begriffshierarchie der Fertigungsprobleme der Problemklassen 5 und 6

#### Nennung der Features

In den folgenden Teilen der Erhebung beschränkte man sich auf den rechten Unterbaum der Klasse 5, der in Abbildung 7 besonders hervorgehoben ist.

Der Experte erklärte die Problemklassen mit folgenden Features:

#### Klasse 5:

- beidseitig ansteigende Welle,
- Leistungsstarke Maschine,
- stabile Maschine,
- mindestens sechs Werkzeuge,
- beliebige Anzahl Zentrierbohrungen erlaubt,
- Aufspannung mit Stirnseitenmitnehmer,
- mindestens ein Einstich.

#### Klasse b:

- Werkstoff Gußeisen,
- Werkstoff spröde,
- keine Probleme mit dem Spanbruch.

#### Klasse e:

- keine Nut im Mittelabsatz,
- Gewinde,
- Gewinde am rechten äußeren Zapfen.

#### Klasse f:

- Nut im Mittelabsatz,
- Kragen,
- Ausgangsteil nicht zylindrisch.

Klasse i:

- Kragen,
- Mittelabsatz ist ein Kragen,
- Nut in rechten Absätzen,
- Einstiche direkt am Kragen,
- Ausgangsteil nicht zylindrisch.

Klasse j:

- ansteigende Kontur in den rechten Absätzen,
- Ausgangsteil zylindrisch.

Klasse g1w2d2:

- Bearbeitung sequentiell.

Klasse g1w2d3:

- Bearbeitung parallel.

Klasse g3w2d2:

- Bearbeitung sequentiell.

Klasse g3w2d3:

- Bearbeitung parallel.

Klasse g2w2d2:

- Bearbeitung sequentiell.

Klasse g2w2d3:

- Bearbeitung parallel.

Innerhalb der Problemklassen gilt das Subsumptionsprinzip. Der Experte nannte daher an dieser Stelle keine Features, die zu abstrakteren Klassen gehören. Beispielsweise gilt das Feature 'Länge des Werkstücks kleiner oder gleich 500 mm' für alle 60 Problemfälle der Matrix und ist somit ein Feature der Klasse A, und darf bei weniger abstrakten Problemklassen nicht genannt werden.

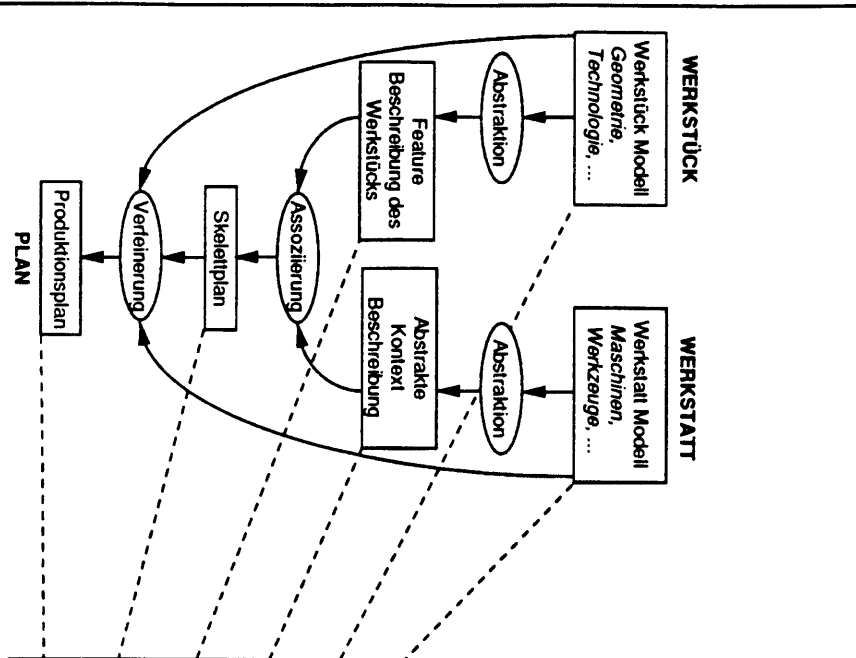
Dies stellt einen der wesentlichen Gründe dar, die für die geringe Anzahl genannter Features sprechen. Bei der gesamten Untersuchung sind Toleranzangaben und Oberflächenanforderungen nicht betrachtet worden. Diese Aspekte hätten weitere Features ergeben. Ein dritter Grund liegt darin, daß die Beschreibung der Problemklassen bewußt nicht zu speziell werden sollte, sondern eine gewisse Abstraktion enthält. Ohne die Einhaltung einer Abstraktion hinreichender Größe fallen zu wenig neue Probleme der Fertigung in die einzelnen Klassen und eine Anwendung zahlt sich nicht aus. Es sollten nur Features genannt werden, die den Plan beeinflussen oder der Erkennung solcher Features dienen.

Zuordnung zu den Sichten

Die von dem Experten vorgenommenen Zuordnungen der Features zu den Sichten ist in Abbildung 8 dargestellt.

Bei den Zuordnungen der Features zu den Sichten fällt auf, daß kein einziges Feature der 'konkreten Werkstück-Sicht' zugeordnet wurde. Dies bekräftigt die Annahme, daß Problemklassen sich auf einem abstrakten Niveau ergeben, und keine direkten Anforderungen an fest instanziierte Werte gestellt werden müssen. Die Beschreibungen der Problemklassen lassen immer Spielraum für konkrete Instanzen, so daß hinreichend viele neue Probleme in die einzelnen Problemklassen fallen werden. Die meisten Zuordnungen erhielt die 'abstrakte Werkstück-Sicht'. Dies ist ebenfalls eine erwartete Tendenz, da bei der Fertigung rotationssymmetrischer Drehteile die Werkstückbeschreibung vorrangig ist. Es gibt sowohl Features, die eindeutig einer Sicht zugeordnet wurden, so daß eine Einteilung der relevanten Aspekte eines Fertigungsproblems in einzelne Teilbereiche der Domäne sinnvoll ist, als auch Features, die einer Kombination mehrerer einfacher Sichten zugeordnet wurden. Die Annahme, daß verschiedene Aspekte der Domäne interagieren und Abhängigkeiten aufweisen, konnte dadurch ebenfalls bestätigt werden.

**Modell der Expertise**



**Zuordnung**

FEATURE		SICHT					
		konkrete Werkstatt	konkrete Werkstück	abstrakte Werkstatt	abstrakte Werkstück	abstrakte Plan	konkrete Plan
	beidseitig ansteigende Welle						
	Leistungsstarke Maschine			X	X		
	stabile Maschine			X	X		
	mindestens sechs Werkzeuge			X	X		
	beliebige Anzahl Zentrierbohrungen erlaubt			X	X	X	
	Aufspannung mit Stirnseitenmitnehmer			X			X
	mindestens ein Einstich			X			
	Werkstoff Gusseisen			X			
	Werkstoff spröde			X			
	keine Probleme mit dem Spanbruch			X			
	keine Nut im Mittelabsatz			X	X		
	Gewinde			X	X		
	Gewinde am rechten äußeren Zapfen			X	X		
	Nut im Mittelabsatz			X	X		
	Kragen			X	X		
	Ausgangsflans nicht zylindrisch			X	X	X	

## V.2.2 Wissensvergleichsepisode

In dieser Episode wurden nun die genannten Features miteinander verglichen.

### Redundanz- und Konsistenztests

Die nach Sichten getrennte Wissensvergleichsepisode ergab noch keine Redundanzen oder Inkonsistenzen. Die Erklärungsstruktur des Experten wurde durch die Tests verifiziert. Es waren weder überflüssige noch widersprüchliche Features genannt worden. Erst in einem späteren Iterationsschritt mussten Redundanzen korrigiert werden, die an dieser Stelle noch nicht aufgetreten sind (siehe auch Seite 39).

## V.2.3 Kompetenzabschätzungsepisode

Die Kompetenz der resultierenden Wissensbasis erstreckt sich mindestens über die 60 Beispielfälle. Alle Fälle können einer der Problemklassen zugeordnet werden, und der entsprechende Produktionsplan zur Lösung des Falles verwendet werden. Weitere ähnliche neue Probleme werden ebenfalls einer der Problemklassen zugeordnet werden können, so daß die assoziierten Skelettpläne zu einem konkreten Produktionsplan verfeinert werden können. Vernachlässigt man an dieser Stelle, daß die Erhebung sich nur auf einen Teilbaum der gesamten Problemklassenhierarchie bezieht, und nimmt an, daß es sich bei dem Knoten der Problemklasse 5 um die Wurzel der Hierarchie handelt, so ist das Expertensystem für alle neuen Fertigungsprobleme kompetent, in denen eine beidseitig ansteigende Welle mit mindestens einem Einstich und der Erlaubnis beliebiger Zentrierbohrungen gefertigt werden muß, und deren Fertigungsumfeld über eine leistungsstarke und stabile Maschine mit mindestens sechs Werkzeugen

eine Begründung für die Zuordnung zu den Sichten.

- `Aufspannung_mit_Stirnseitenmitnehmer`(konkrete Plan-Sicht) = "Die gewählte Aufspannungsart im Plan ist Stirnseitenmitnehmer mit Spitze.",
- `mindestens_ein_Einstich`(abstrakte Werkstück-Sicht) = "Das Werkstück hat mindestens einen Einstich in seiner Geometrie."

Das Feature 'mindestens ein Einstich' kann als atomarer Term bei der formalen Definition der Problemklasse verwendet werden. 'Einstich' ist ein Begriff, der in der Fachwelt des Maschinenbaus verwendet wird. An dieser Stelle sei allerdings bereits darauf hingewiesen, daß bei einer konkreten Implementierung solche geometrischen Features in Anforderungen an die Koordinaten der Werkstückkontur übersetzt werden müssen, damit ein gegebenes Problem in der üblichen Form der Konturkoordinaten repräsentiert und ausgehend von dieser Repräsentation klassifiziert werden kann. Eine Alternative zu dieser Vorgehensweise ist eine Übersetzung der CAD-Zeichnung oder der Konturkoordinaten in abstraktere Features der Maschinenbausprache. Es ist aber nicht erwünscht, die originale Repräsentation eines Werkstückes zu verändern, um sie verarbeiten zu können.

Klasse b:

- `Werkstoff_Gußeisen`(abstrakte Werkstück-Sicht) = "Ein Werkstoff ist ein Gußeisen, wenn er einen Kohlenstoffgehalt von 2.06 % hat.",
- `Werkstoff_spröde`(abstrakte Werkstück-Sicht) = "Der Werkstoff ist spröde. Man spricht bei Werkstoffen immer von spröde oder duktil."

An dieser Stelle konnte der Experte auch genauere chemische Eigenschaften nennen, die einen spröden bzw. duktilen Werkstoff kennzeichnen, und die Verfahren spezifizieren, mit denen man diese Eigenschaften ermittelt. Für die formale Definition einer Problemklasse ist eine so feine Definition nicht sinnvoll, und entspricht auch nicht dem verwendeten Sprachgebrauch der Maschinenbauexperten.

- `keine_Probleme_mit_dem_Spanbruch`(abstrakte Plan-Sicht) = "Fallen bei der Fertigung des Werkstückes kurze Späne an, so entstehen keine Probleme mit dem Spanbruch, und es sind keine Maßnahmen im Plan notwendig, die den Spanbruch verbessern."

Klasse e:

- `keine_Nut_im_Mittelabsatz`(abstrakte Werkstück-Sicht) = "Das Werkstück hat keine Nut im Mittelabsatz."

'Nut' und 'Mittelabsatz' sind wie 'Einstich' feststehende Maschinenbaubezeichnungen, die abstraktere Kontureigenschaften benennen.

- `Gewinde`(abstrakte Werkstück-Sicht) = "Das Werkstück hat ein Gewinde.",
- `Gewinde_am_rechten_äußeren_Zapfen`(abstrakte Werkstück-Sicht) = "Das Gewinde des Werkstückes liegt am rechten äußeren Zapfen der Geometrie."

Die genaue Art des Gewindes spielt keine Rolle, da für ein bestimmtes Gewinde feste Sequenzen zur Fertigung existieren, die jeweils in den Plan einfach eingefügt werden können.

Klasse f:

- `Nut_im_Mittelabsatz`(abstrakte Werkstück-Sicht) = "Das Werkstück hat eine Nut im Mittelabsatz.",
- `Kragen`(abstrakte Werkstück-Sicht) = "Ein Kragen ist ein schmaler Mittelabsatz."

Interessanterweise gibt der Experte auch bei Hinterfragen keine genaue Definition von 'schmal' an. Es liegt der Schluß nahe, daß es im Maschinenbau Bezeichnungen gibt, die intuitiv von Experten übereinstimmend verwendet werden, ohne daß genaue Maße bestimmt werden. Höper [Höper, 1991] führt zur Zeit eine Untersuchung durch, in der unter anderem solche Werte definiert werden sollen. Er geht dabei tatsächlich von Expertenmeinungen aus, und berechnet einen Mittelwert.

- `Ausgangsteil_nicht_zyklindrisch`(abstrakte Werkstück-Sicht, abstrakte Plan-Sicht) = "Das Werkstück wird ausgehend von einem nicht zylindrischen Ausgangsteil gefertigt. Dies muß bei der Schnittfolge des Produktionsplanes berücksichtigt werden."

Klasse i:

- `Kragen`(abstrakte Werkstück-Sicht) = "Ein Kragen ist ein schmaler Mittelabsatz.",
- `Mittelabsatz_ist_ein_Kragen`(abstrakte Werkstück-Sicht) = "Der Mittelabsatz ist in diesem Falle ein Kragen.",
- `Nut_in_rechten_Absätzen`(abstrakte Werkstück-Sicht) = "Das Werkstück hat eine Nut in einem Absatz, der rechts vom Mittelabsatz liegt.",
- `Einstiche_direkt_am_Kragen`(abstrakte Werkstück-Sicht) = "Das Werkstück hat einen Einstich direkt links am Kragen und einen Einstich direkt rechts am Kragen.",
- `Ausgangsteil_nicht_zyklindrisch`(abstrakte Werkstück-Sicht, abstrakte Plan-Sicht) = "Das Werkstück wird

ausgehend von einem nicht zylindrischen Ausgangsteil gefertigt. Dies muß bei der Schnittfolge des Produktionsplanes berücksichtigt werden."

Klasse j:

- ansteigende\_Kontur\_in\_den\_rechten\_Absätzen(abstrakte Werkstück-Sicht) = "Die Kontur des Werkstückes ist in den rechten Absätzen ansteigend."

Die Monotonie einer Kontur wird unter Vernachlässigung beliebiger Nuten und Einstiche beurteilt.

- Ausgangsteil\_zylindrisch(abstrakte Werkstück-Sicht, abstrakte Plan-Sicht) = "Das Werkstück wird ausgehend von einem zylindrischen Ausgangsteil gefertigt. Dies muß bei der Schnittfolge des Produktionsplanes berücksichtigt werden."

Klasse g1w2d2:

- Bearbeitung\_sequentiell(konkrete Werkstatt-Sicht, konkrete Plan-Sicht) = "Die Maschine hat nur einen Revolver, daher kann das Werkstück nur sequentiell gefertigt werden."

Klasse g1w2d3:

- Bearbeitung\_parallel(konkrete Werkstatt-Sicht, konkrete Plan-Sicht) = "Die Maschine hat zwei Revolver, daher kann das Werkstück parallel von beiden Seiten bearbeitet werden".

Klasse g3w2d2:

- Bearbeitung\_sequentiell(konkrete Werkstatt-Sicht, konkrete Plan-Sicht) = "Die Maschine hat nur einen Revolver, daher kann das Werkstück nur sequentiell gefertigt werden."

Klasse g3w2d3:

- Bearbeitung\_parallel(konkrete Werkstatt-Sicht, konkrete Plan-Sicht) = "Die Maschine hat zwei Revolver, daher kann das Werkstück parallel von beiden Seiten bearbeitet werden".

Klasse g2w2d2:

- Bearbeitung\_sequentiell(konkrete Werkstatt-Sicht, konkrete Plan-Sicht) = "Die Maschine hat nur einen Revolver, daher kann das Werkstück nur sequentiell gefertigt werden."

Klasse g2w2d3:

- Bearbeitung\_parallel(konkrete Werkstatt-Sicht, konkrete Plan-Sicht) = "Die Maschine hat zwei Revolver, daher kann das Werkstück parallel von beiden Seiten bearbeitet werden".

Die genaueren Beschreibungen, die der Experte angab, sind nicht immer feinere Beschreibungen des Features. Einige Angaben dienen mehr der Rechtfertigung, warum das Feature zu der entsprechenden Sicht

stabile\_Maschine  $\sqcap$   
 mindestens\_sechs\_Werkzeuge  $\sqcap$   
 beliebige\_Anzahl\_Zentrierbohrungen  $\sqcap$   
 Aufspannung\_mit\_Stirnseitenmitnehmer  $\sqcap$   
 mindestens\_ein\_Einstich

mit den folgenden Featuredefinitionen:

beidseitig\_ansteigende\_Welle =  
 Welle  $\sqcap$   
 ansteigende\_Kontur\_rechte\_Absätze  $\sqcap$   
 ansteigende\_Kontur\_linke\_Absätze  $\sqcap$   
 Mittelabsatz  $\sqcap$   
 zwei\_Zapfen,

Leistungsstarke\_Maschine =  
 $\exists d$ .Maschine  $\sqcap$   
 größergleich-500 (d U/MIN)  $\sqcap$   
 größergleich-50 (d KW),

stabile\_Maschine =  
 $\exists d$ .Maschine  $\sqcap$   
 $\exists (d \text{ ST}).\text{stabil}$ ,

mindestens\_sechs\_Werkzeuge =  
 $\exists d$ .Maschine  $\sqcap$   
 größergleich-6 (d WZ),

beliebige\_Anzahl\_Zentrierbohrungen =  
 null\_Zentrierbohrung  $\sqcap$   
 eine\_Zentrierbohrung  $\sqcap$   
 zwei\_Zentrierbohrungen,

Aufspannung\_mit\_Stirnseitenmitnehmer =  
 beliebige\_Anzahl\_Zentrierbohrungen  $\sqcap$   
 $\exists d$ .Maschine  $\sqcap$   
 $\exists (d \text{ AM}).\text{Stirnseitenmitnehmer\_mit\_Spitze}$ , und

mindestens\_ein\_Einstich =  
 $\exists K$ .Einstich.

Features, die in den iterativen Schritten der Zuordnung zu den Sichten, der Redundanz- und Konsistenztests, und der Semiformalisierung weiter feiner zerlegt worden sind, wie zum Beispiel das Feature 'Mittelabsatz', sind wiederum entsprechend formalisiert worden.

Da innerhalb der Problemklassenhierarchie das Subsumptionsprinzip gilt, werden Nachfolgerklassen immer zusätzlich mit den Vorgängerklassen formalisiert, um die Vererbung explizit auszudrücken. Zum Beispiel wird die Klasse j durch folgendes Konzept repräsentiert:

Klasse\_j =  
 Klasse\_5  $\sqcap$   
 Klasse\_b  $\sqcap$   
 Klasse\_e  $\sqcap$   
 Ausgangsteil\_zyindrisch,

da die Klassen 5, b und e Vorgängerklassen der Klasse j sind. Eine vollständige Formalisierung der Problemklassen und der Features befindet sich in den Implementierungsversuchen im Anhang.

Insgesamt stellte sich bei der praktischen Anwendung des Verfahrens heraus, daß der Experte mit einigen Schritten Schwierigkeiten hatte. Die Zuordnung zu den Sichten beispielsweise mußte mehrfach wiederholt und korrigiert werden, da sich bei der Semiformalisierung Widersprüche ergaben. Nachdem der Experte einmal alle Episoden durchlaufen hatte, erkannte er die Zusammenhänge besser und konnte leichter die geforderten Beschreibungen machen.



## VI IMPLEMENTIERUNG

### VI.1 TAXON

TAXON ist ein terminologisches Wissensrepräsentationssystem mit bestimmten Erweiterungen, das angemessen ist, den Formalismus einer konkreten Domäne aufzunehmen. Es ist eine Instanz des in [Baader, Hanschke, 1991a] beschriebenen Systems. Konzepte können in Termen anderer Konzepte, Rollen, und Attributen definiert werden. Zusätzlich können sich Konzeptdefinitionen auf eine konkrete Domäne beziehen, die reelle oder ganzzahlige Arithmetik, einige Bewegungen in der Zeit, Zugang zu einem Datenbanksystem, etc. unterstützt. Das bedeutet, daß Konzepte definiert werden können, in denen Prädikate einer konkreten Domäne auftreten. Die Prädikate können auf Listen von Attributpfaden angewendet werden [Abecker, Drollinger, Hanschke, 1991; Abecker, Hanschke, 1991; Becker, 1991].

Um die Implementierungsversuche der formalen Definitionen der Problemklassen nachvollziehen zu können, ist es notwendig einige Bemerkungen zu der Syntax von TAXON zu machen. Es werden jedoch nur einige grundsätzliche Schlüsselwörter erläutert. Eine genaue Spezifikation der Syntax findet sich in [Abecker, Hanschke, 1991].

Für die Definition von Konzepten in der TBox werden folgende Schlüsselwörter verwendet: PRED (Definition eines Prädikates der konkreten Domäne), PRIM (Definition eines primitiven Konzeptes), CONC (Definition eines Konzeptes), ATTR (Definition eines Attributes), ROLE (Definition einer Rolle), NOT (Negation), AND (Konjunktion), OR (Disjunktion), SOME (Existenzquantifizierung), und ALL (Allquantifizierung).

Für die Definition von Individuen und Zusicherungen in der ABox gibt es die beiden Schlüsselwörter INDI (Definition eines Individuums) und ASSE (Zusicherung an ein Individuum).

### VI.2 Anforderungen

Die Implementierung stellt den Versuch dar, die formale Definition der Problemklassen in dem terminologischen Wissensrepräsentationssystem TAXON durchzuführen.

Das Ziel der Implementierung besteht darin, eine Begriffshierarchie der Problemklassen zu erhalten, in die Repräsentationen der verwendeten Beispielprobleme und neuer Fertigungsprobleme eingeordnet werden können. Zu diesem Zweck muß sich neben der eigentlichen Definition der Problemklassen auch eine geeignete Repräsentation der Fertigungsprobleme ergeben. Die angestrebte Repräsentation soll einerseits allgemein genug sein, um hinreichend viele Fertigungsprobleme zu erfassen, und andererseits speziell genug, daß sie den Anforderungen genügt, aber im System realisierbar bleibt. Eine weitere Anforderung an eine Repräsentation ist die Möglichkeit, Teile der Beschreibung der Fertigungsprobleme in üblicher Form übernehmen zu können. Beispielsweise wird eine Geometrie eines Fertigungsproblems typischerweise durch eine CAD-Zeichnung dargestellt, aus der die Koordinaten der Kontur abgelesen werden können. Ein Einstich sollte dann aus diesen Koordinaten abgeleitet werden können.

Da die Problemklassen durch eine Konjunktion der Features definiert sind, gilt es zunächst die Features zu repräsentieren. Die Features sind nach ihrer Nennung den verschiedenen Sichten der Maschinenbausprache zugeordnet und damit strukturiert worden. Diese Strukturierung soll beibehalten werden, indem für jede Sicht eine eigene TBox gebildet wird. Die Summe der einzelnen TBoxes ergibt schließlich die Definition der gesamten Features.

An dieser Stelle sollen nicht die einzelnen Implementierungsversuche beschrieben, sondern die aufgetretenen Probleme anhand ausgewählter Beispiele diskutiert werden. Eine genauere Darstellung der Implementierungsversuche befindet sich im Anhang.

### VI.3 Probleme

#### VI.3.1 Einschränkungen der konkreten Domäne

In der konkreten Domäne der reellen Zahlen, die bei der Darstellung der Werkstückgeometrien eingesetzt werden soll, können keine arithmetischen Operationen, wie beispielsweise die Subtraktion, verwendet werden, sondern lediglich Vergleiche und logische Verknüpfungen solcher Vergleiche. Dadurch entsteht zusätzlicher Repräsentationsaufwand. Die Länge einer Zylindermantelfläche, die implizit durch die Subtraktion der z-Koordinaten der Referenzpunkte berechnet werden sollte, muß zum Beispiel aus diesem Grunde explizit angegeben werden.

Eine weitere Einschränkung der konkreten Domäne besteht darin, daß bei Prädikatsaufrufen bei der Definition komplexer Konzepte keine Zahlen verwendet werden dürfen. Um einen Vergleich zwischen

einem definierten Attributwert und einer reellen Zahl durchzuführen, muß ein eigenes Prädikat in der konkreten Domäne definiert werden.

### VI.3.2 Disjunktionen

Die Verwendung von Disjunktionen bei der Definition komplexer Konzepte bereitet Schwierigkeiten in der Laufzeit. Die Konzepte der einfachen Werkstück-Sicht werden beispielsweise ohne die Definition

(conc Absatz  
    (or       steigender-Absatz  
             fallender-Absatz))

und ohne die Definition der Zapfen in etwa 380 Sekunden klassifiziert. Nimmt man hingegen nur die disjunktive Definition des Absatzes hinzu, so steigt der Klassifizierungsaufwand bis zu nicht brauchbaren Zeiten. Ein Klassifizierungsversuch ist nach sechseinhalb Stunden abgebrochen worden.

### VI.3.3 Listen undefinierter Länge

In TAXON ist keine Definition von Listen undefinierter Länge möglich. Eine Kontur ist beispielsweise eine Folge von Absätzen. Die Länge dieser Folge steht nicht allgemein fest, da die Anzahl der Absätze einer Kontur unterschiedlich sein kann. Eine entsprechende Erweiterung einer terminologischen Wissensrepräsentationssprache durch Einführung einer transitiven Hülle führt in Kombination mit konkreten Domänen zu unentscheidbaren Problemen [Baader, Hanschke, 1991b].

### VI.3.4 Rollen und Allquantifizierung

Versucht man diesem Problem Abhilfe zu schaffen, indem man eine Kontur als Rolle definiert, der dann unbestimmt viele 'Füller' zugewiesen werden können, beispielsweise in der Art

(role K)

(conc Kontur  
    (all K Absatz)),

so ergeben sich zwei neue Probleme. Erstens kann man nicht mehr auf spezielle Absätze zugreifen, da eine Rolle mehrere 'Füller' haben kann, so daß bei einem Zugriff nicht eindeutig festgelegt ist, um welchen Rollenfüller es sich genau handelt. Bei der Definition abstrakter Features kann dies erforderlich sein. Zum Beispiel die Definition des Features 'Nut-in-rechten-Absätzen' erfordert die Existenz einer Nut und eines rechten Absatzes, auf deren Koordinaten dann zugegriffen werden muß, um die Lage der Nut relativ zu dem Absatz zu bestimmen. Zweitens wird das Konzept 'Kontur' in dieser Form nie erfüllt sein. Da in TAXON nicht die 'Closed-World-Assumption' zugrunde gelegt wird, kann es immer noch potentielle Rollenfüller von 'K' geben, die kein Absatz sind. Eine explizite Zusicherung in der ABox kann dies ausschließen, schränkt aber die Allgemeinheit der Featureerkennung erheblich ein. Das Feature 'Nut-in-rechten-Absätzen' kann bei einer Einschränkung der Allgemeinheit durch

(conc Nut-in-rechten-Absätzen  
    (and     (some N Nut)  
            (some A rechter-Absatz)  
            (kleiner (A erste-Koordinate)  
                    (N erste-Koordinate))  
            (kleiner (N letzte-Koordinate)  
                    (A letzte-Koordinate))))

definiert werden. Dabei müssen Absätze aber ausdrücklich dem Attribut A und Nuten dem Attribut N zugesichert werden. A und N müssen weiterhin Attribute und keine Rollen sein, damit ein Zugriff möglich wird. Insbesondere darf es dann nur einen solchen Absatz geben und nur eine Nut. Ohne eine Verwendung eines eigenen Attributes oder einer eigenen Rolle ist ein Zugriff generell nicht kontrolliert möglich. Zum Beispiel eine Definition

(conc Nut-in-rechten-Absätzen

(and rechter-Absatz  
Nut  
(liegt-in Nut-Koordinaten  
Absatz-Koordinaten)))

mit einem Prädikat 'liegt-in' der konkreten Domäne der reellen Zahlen ist nicht möglich, da bei den

(asse (punkt1 x 0)  
(punkt1 y 1)  
(punkt2 x 0)  
(punkt2 y 3))

(indi g1)

(asse (g1 p1 punkt1)  
(g1 p2 punkt2)).

Durch diese Umwege entsteht erheblicher Mehraufwand an Zusicherungen in der ABox.

## VI.4 Ursachen

### VI.4.1 Vollständigkeit und Korrektheit

Eine erste Ursache liegt grundsätzlich in der Forderung nach Vollständigkeit und Korrektheit der Algorithmen einer terminologischen Wissensrepräsentationssprache. Die Kombination einer transitiven Hülle, die es ermöglicht Listen undefinierter Länge einzuführen, und einer konkreten Domäne kann daher zum Beispiel nicht realisiert werden.

### VI.4.2 Bottom-Up versus Top-Down

Um Disjunktionen zu vermeiden, sollte man bei der Definition einer Taxonomie immer einen Top-Down-ansatz eines Bottom-Up-Ansatzes wählen. Bei einem Bottom-Up-Ansatz definiert man zunächst einfachere Konzepte und bildet komplexere Konzepte häufig aus Disjunktionen dieser einfacheren Konzepte. Dagegen

## VI.5 Alternativen

### VI.5.1 Einschränkung der Allgemeinheit

Eine erste Alternative besteht in einer Einschränkung der Forderung nach hinreichend allgemeiner Darstellung. Man kann eine Repräsentation wählen, in der maximal die Beispielprobleme und einige weitere neue Fertigungsprobleme repräsentiert werden können. Dadurch ergeben sich einfachere Konzeptdefinitionen und weniger Laufzeitprobleme.

### VI.5.2 Abstrakte Repräsentation

Eine weitere Alternative könnte eine abstraktere Darstellung der Probleme sein. Man repräsentiert beispielsweise eine Werkstückgeometrie nicht mehr durch eine Menge von Koordinaten, sondern durch abstraktere Features, die zum Beispiel direkt die Existenz eines Einstiches zusichern.

### VI.5.3 Andere Repräsentation

Eine dritte Möglichkeit besteht in einer kombinierten Darstellung in verschiedenen Repräsentationssystemen. Features werden in einer anderen Repräsentation dargestellt und lediglich die Begriffshierarchie der Problemklassen wird in TAXON formalisiert. Es ist allerdings unbekannt, welche alternative Repräsentation für die Features gewählt werden kann.

Ein solcher Versuch wurde simuliert, indem alle Features als primitive Konzepte definiert wurden und lediglich die Problemklassen noch als Konjunktionen dieser Features dargestellt waren. Dabei ergab sich zwangsläufig für eine Klassifizierung eine sehr kurze Zeit von 7 Sekunden.

### VI.5.4 Kombination mit einer Regelsprache

Die vielversprechendste Alternative scheint eine Einbettung des taxonomischen Wissensrepräsentationssystems in ein Gesamtsystem zu sein, das weitere Systemteile zur Verfügung stellt, die beispielsweise die Featureerkennung erleichtern können. Ein solcher Ansatz ist im ARC-TEC-Projekt bereits verfolgt worden [Boley, Hanschke, Harm, Hinkelmann, et. al., 1991]. Die Erkennung komplexerer Features, wie beispielsweise 'ansteigende-Kontur', wird in einer Regelsprache FORWARD durchgeführt.

## VII DISKUSSION

### VII.1 Zweck der Problemklassen

Das Ergebnis der Anwendung des Wissensakquisitionswerkzeuges CECoS auf eine Menge von ausgewählten Beispielen ist eine formale Definition einer Menge von Problemklassen. Zu jeder Problemklasse eines nicht zu allgemeinen Abstraktionsgrades existiert ein Skelettplan, der mit SP-GEN erzeugt wird. Ein neues Problem der Fertigung kann nun einer der Problemklassen zugeordnet werden. Die Lösung des neuen Problems ergibt sich durch eine Verfeinerung des zu der entsprechenden Problemklasse gehörenden Skelettplans.

Um ein neues Problem den Problemklassen zuzuordnen, prüft man an jedem Knoten des Hierarchiebaumes, ob das neue Problem die Features der formalen Definition der zu dem Knoten gehörenden Problemklasse erfüllt. Sind die Features erfüllt, so gehört das neue Problem zu dieser Problemklasse, und alle Nachfolgerklassen dieser Klasse werden überprüft, bis ein Abstraktionsgrad erreicht ist, auf dem das neue Problem zu keiner Nachfolgerklasse mehr zugeordnet werden kann. Je feiner der Abstraktionsgrad der sich ergebenden Problemklasse ist, desto genauer ist der resultierende Skelettplan und desto weniger Aufwand wird für die Verfeinerung zu einem konkreten Produktionsplan benötigt. Gehört das neue Problem nur zu der zur Wurzel der Hierarchie gehörenden Problemklasse, so ist der Skelettplan sehr abstrakt. Die Verfeinerung entspricht nahezu einer Neuplanung. Gehört das neue Problem zu einer zu einem Blatt der Hierarchie gehörenden Problemklasse, so ist der Skelettplan sehr genau oder sogar bereits ein konkreter Produktionsplan. Die Verfeinerung ist eine Varianten- oder Modifikationsplanung.

Bei diesem Verfahren der Zuordnung wird erwartet, daß immer eine eindeutige Aussage über die Zugehörigkeit eines neuen Problems zu einer Problemklasse gemacht werden kann.

### VII.2 Zuordnung neuer Probleme

Bei der Zuordnung eines neuen Problems kann unter Umständen keine eindeutige Entscheidung gefällt werden, wenn die formale Definition der Problemklassen nicht geeignet gewählt ist.

Es ergeben sich bei der Zuordnung eines neuen Problems theoretisch folgende Möglichkeiten:

1) Das neue Problem gehört zu keiner Problemklasse der Hierarchie. In diesem Fall gehört das neue Problem bereits nicht zu der zur Wurzel gehörenden Problemklasse, und kann mit den vorhandenen Skelettplänen nicht bearbeitet werden.

2) Das neue Problem gehört eindeutig zu einer Problemklasse eines bestimmten Abstraktionsgrades.

Features definierten Problemklassen ergeben. Sei ein neues Problem gegeben, das einerseits alle Features einer Problemklasse und das einzige Feature einer ersten Nachfolgerklasse erfüllt, andererseits ein Feature einer zweiten Nachfolgerklasse nicht erfüllt. Darüber hinaus erfülle das neue Problem eine Menge in der gesamten Hierarchie nicht verwendeter Features. Das neue Problem gehört eindeutig zu der ersten Nachfolgerklasse, obwohl vielleicht ein Feature der nicht verwendeten Features dazu im Widerspruch steht. Je weniger Features eine Problemklasse definieren, desto eher kann ein solcher Fall auftreten.

Diese in diesem allgemeinen Beispiel geschilderte Schwierigkeit tritt nicht auf, wenn man die 'Closed-World-Assumption' voraussetzt, denn dann ist jedes nicht genannte Feature automatisch nicht gültig, d.h. implizit als Negierung bei der Definition der Problemklasse angegeben. Im Bereich des Maschinenbaus kann man aber nicht von einer 'Closed-World-Assumption' ausgehen, da nicht jedes nicht genannte Feature negiert angenommen werden darf. Die 'Closed-World-Assumption' führte dazu, daß man jedes triviale Feature explizit angeben muß, sofern man nicht das Gegenteil voraussetzen will. Dadurch müssen viele Features angegeben werden, die nicht wesentlich relevant für die Definition einer Problemklasse sind. Dadurch erhöht sich die Komplexität der Aufgabe erheblich, so daß man die 'Closed-World-Assumption' im Bereich des Maschinenbaus nicht annehmen darf. Die explizite Angabe aller negierten Features, die nicht erwünscht sind, stellt allerdings eine ähnliche Komplexitätserhöhung der Aufgabe dar. Bei typischen Problemen der Fertigung einfacher rotationssymmetrischer Drehteile und den daraus resultierenden Problemklassen wird diese Schwierigkeit in kleinerem Ausmaß zu erwarten sein als bei ausgefalleneren oder komplexeren Problemen.

### VII.3 Einschätzung der Anwendbarkeit

Die Anwendbarkeit der Hierarchie von Problemklassen ist abhängig von der Existenz der zugehörigen Skelettpläne. Eine allgemeine Frage, die sich dabei stellt, ist der maximale Abstraktionsgrad sinnvoller Skelettpläne. Der Experte schätzte die Möglichkeit, für die Abstraktionsebene der Klassen 0 bis 10, einen Skelettplan zu konstruieren, als realistisch ein. In dieser Untersuchung sind allerdings keine Versuche unternommen worden, konkrete Skelettpläne für einzelne Problemklassen zu erheben. Erst bei einer solchen Konstruktion von Skelettplänen ließe sich die Nutzbarkeit der formalen Definitionen der Problemklassen beurteilen. Die Anzahl der genannten Features ist sehr gering. Es ist anzunehmen, daß aufgrund dieser wenigen Features kein sinnvoller Skelettplan bestimmt werden kann. Um die Nutzbarkeit der genannten Features zu überprüfen, sollte die Konstruktion eines Skelettplanes durchgeführt und der Versuch diesen Skelettplan mit einer Problemklasse zu assoziieren unternommen werden. Es besteht die Möglichkeit, daß Skelettpläne besonders auf der Abstraktionsebene geeignet sind, auf der in der entsprechenden NC-Programmierung Macro-Operatoren oder Unterprogramme zur Verfügung stehen. Beispielsweise lassen sich ganze Konturabschnitte mit einem NC-Operator fertigen [FANUC, 1985; Keller, 1983]. Dabei muß allerdings die Kontur bekannt sein. Inwieweit eine Kontur abstrahiert werden kann, und trotzdem ein solcher NC-Operator anwendbar bleibt, oder man eine mögliche Instanziierung eines abstrakten NC-Operators erwarten kann, wurde an dieser Stelle nicht untersucht. Neben der Möglichkeit, Skelettpläne generell zu konstruieren, muß man auch den Aufwand einer Skelettplanverfeinerung abschätzen. Ein sehr abstrakter Skelettplan wird einen so großen Aufwand der Verfeinerung erfordern, daß er praktisch keinen Wert in der Anwendung haben wird. Es ist zu erwarten, daß Verfeinerung auf einer niedrigen Ebene ohne Schwierigkeiten möglich sein wird, dagegen auf einer hohen Ebene nahezu unmöglich ist.

Die grundsätzlichen Erfahrungen aus der Praxis, in denen Dreher tatsächlich häufig eine Fertigung durch Abänderung eines bereits bestehenden ähnlichen Produktionsplanes, die als Abstraktion und anschließende Verfeinerung interpretiert werden kann, durchführen, motivieren insgesamt dazu, den gewählten Ansatz weiter zu verfolgen und zu verbessern.

## **VIII DANKSAGUNG**

Mein Dank gilt den Experten des Maschinenbaus, Ralf Legleitner und Hans Werner Höper, die mir in unermüdlicher Geduld mit Rat und Tat bei Erhebungen und ergänzenden Fragen zur Verfügung standen. Weiterer Dank gilt Franz Schmalhofer, Gabriele Schmidt und Otto Kühn, die mich in inhaltlichen Fragen berieten, sowie Philipp Hanschke, Andreas Abecker und Klaus Becker, die mich bei der Implementierung durch ihre eigenen Erfahrungen unterstützten.



## IX LITERATUR

Abecker, A., und Hanschke, P. (1991). *TAXON: instructions for use*. German Research Center for Artificial Intelligence. Draft.

Abecker, A., Drollinger, D., und Hanschke, P., (1991). *TAXON: a concept language with concrete domains*. in: Richter, M.M., und Boley, H. (Eds.) *Preprints of the Proceedings of the International Workshop on Processing Declarative Knowledge (PDK '91)*. German Research Center for Artificial Intelligence.

Achilles, E., Hollunder, B., Laux, A., und Mohren, J.-P. (1991). *KRIS: Knowledge Representation and Inference System - Benutzerhandbuch* -. German Research Center for Artificial Intelligence. Document, No. 91-14.

Anderson, J.R. (1990). *The adaptive character of thought*. Lawrence Erlbaum, Hillsdale, New Jersey.

Armstrong, S.L., Gleitman, L.R., und Gleitman, H. (1983). What some concepts might not be. *Cognition*, 13, pp. 263-308.

Baader, F. (1990). *Augmenting concept languages by transitive closure of roles: an alternative to terminological cycles*. German Research Center for Artificial Intelligence, Research Report, No. 90-13.

Baader, F., und Hanschke, P. (1991a) A scheme for integrating concrete domains into concept languages. in: *Proceedings of the 12th Conference on Artificial Intelligence*, Sydney, Australia, pp. 452-457.

Baader, F., und Hanschke, P. (1991b) A scheme for integrating concrete domains into concept languages. German Research Center for Artificial Intelligence, Research Report, No. 91-10.

Baader, F., und Hollunder, B. (1990). *KRIS: Knowledge Representation and Inference System - System Description* - German Research Center for Artificial Intelligence, Technical Memo, No. 90-02

Backhaus, K., Erichson, B., Plinke, W., Schuchhard-Fischer, Chr., und Weiber, R. (1987). *Multivariate Analysemethoden*. Springer-Verlag. Berlin, Heidelberg, New York, London, Paris, Tokyo.

Baind, J.C., und Noma, E. (1978). *Fundamentals of Scaling and Psychophysics*. Wiley Series in Behavior.

Bauer, M., Stein, B., und Weiner, J. (1991) Problemklassen in Expertensystemen. *Künstliche Intelligenz*, 3, pp. 13-18.

Becker, K. (1990). *Wissensrepräsentation mit TAXON*. German Research Center for Artificial Intelligence. unpublished.

Bergmann, R. (1990). *Generierung von Skelettplänen als Problem der Wissensakquisition*. Universität Kaiserslautern. Diplomarbeit im Fachbereich Informatik.

Bergmann, R., und Schmalhofer, F. (1991). CECoS: a case-experience combination system for knowledge acquisition for expert systems. *Behavior Research Methods, Instruments, and Computers*, 23(2), pp. 142-148.

Bernardi, A., Boley, H., Hanschke, P., Klauk, C., Kühn, O., Legleitner, R., Meyer, M., Richter, M.M., Schmalhofer, F., Schmidt, G., und Sommer, W. (1991). *ARC-TEC: acquisition, representation and compilation of technical knowledge*. German Research Center for Artificial Intelligence, Technical Memo, No. 91-03

*CAD-like geometries into abstract NC programs.* German Research Center for Artificial Intelligence, Document, No. 91-15.

Boley, H., Hanschke, P., Hinkelmann, K., und Meyer, M. (1991). COLAB: A hybrid knowledge compilation laboratory. *3rd International Workshop on Data, Expert Knowledge and Decisions: Using Knowledge to Transform Data into Information for Decision Support.*

Borg, I., und Staufenbiel, T. (1989). Theorien und Methoden der Skalierung. in: Pawlik, K. (Hrsg.) *Methoden der Psychologie, Band 11*, Verlag Hans Huber, Germany.

Brachman, R.J. (1977). What's in a concept: structural foundations for semantic networks. *International Journal of Man-Machine Studies*, 9(2), pp. 127-152.

Brachman, R.J., und Schmolze, J.G. (1985). An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2), pp. 171-216.

Breuker, J., und Wielinga, B. (1989). Models of expertise in knowledge acquisition. in: Guida, G., und ~~T. ...~~ *Topics in expert system design: methodologies and tools*. Amsterdam: North Holland, pp. 265

295.

Burstein, M.H. (1983). Concept formation by incremental analogical reasoning and debugging. in: Michalski, R.S., Carbonell, J.G., und Mitchell, T.M. (Eds.) *Machine Learning, Vol. II*, pp. 351-370.

Collins, A.M., und Loftus, E.F. (1975). A spreading-activation theory of semantic processing. *Psychological Review*, Vol. 82, No. 6, pp. 407-428.

Chi, M.T.H., Feltovich, P.J., und Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science*, 5, pp. 121-152.

Domke, W. (1982). *Werkstoffkunde und Werkstoffprüfung*. Verlag W. Girardet, Essen, Germany.

Erickson, J.R., und Jones, M.R. (1978). Thinking. *Annual Review Psychology*, 29, pp. 61-90.

Falk, D., Gockel, H.-K., Lernet, F., und Schlossorsch, B. (1990). *Metalltechnik Grundstufe*. Westermann-Verlag GmbH, 3300 Braunschweig, Germany.

FANUC, (1985). FANUC System 10T-Modell F, *Betriebshandbuch*.

Fikes, R., und Nilsson, N. (1971). STRIPS: a new approach to theorem proving in problem solving. *Artificial Intelligence*, 2, pp. 189-208.

Fischer, U. (Lekt. u. Leitung) (1990). *Fachkunde Metall*. Verlag Europa-Lehrmittel, Nourney, Vollmer GmbH & Co. 5657 Haan-Gruiten, Germany.

Fischer, U., und Schmalhofer, F. (1990). *Historische, Soziologische, und Kognitive Aspekte der Fertigungstechnik*. German Research Center for Artificial Intelligence, interner Bericht des ARC-TEC-Projektes.

Fisher, D.H. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2.

Hemenway, K., und Tversky, B. (1984). Objects, Parts, and Categories. *Journal of Experimental Psychology: General*, 113, pp. 169-193.

Höper, H.W. (1991). *Systematik zur Beschreibung von Werkstücken in der Terminologie der Featuresprache*. Universität Kaiserslautern. Studienarbeit im Fachbereich Maschinenbau.

Hoffmann, J. (1988). Wird Wissen in Begriffen repräsentiert ?. *Sprache und Kognition*, 7, pp. 193-204.

Hoffmann, J. (1991a). *Begriffsbildung, ihre funktionalen Ursachen, THESEN*. Workshop: Concept Formation in Man and Machine, Schloss Reichartshausen, 1991.

Hoffmann, J. (1991b). Probleme der Begriffsbildungsforschung: Von S-R Verbindungen zu A-V-K Einheiten. eingereicht bei: *Sprache und Kognition*.

Hoffmann, J., und Zießler, M. (1982). Begriffe und ihre Merkmale. *Zeitschrift für Psychologie*, 190, pp. 46-77.

Hoffmann, J., Zießler, M., Grosser, U., und Kämpf, U. (1985). Struktur- und Prozeßkomponenten in begrifflichen Identifikationsleistungen. *Zeitschrift für Psychologie*, 193, pp. 51-70.

Keller, S. (1983). *Grundlagen der Programmierung von CNC-Werkzeugmaschinen*. Verlag R. Keller, 5600 Wuppertal, Germany.

Klauck, C., Bernardi, A., und Legleitner, R. (1991). *FEAT-REP: representing features in CAD/CAM*. German Research Center for Artificial Intelligence, Research Report, No. 91-20.

Klix, F. (1984). Über Erkennungsprozesse im menschlichen Gedächtnis. *Zeitschrift für Psychologie*, 192, pp. 18-46.

Lebowitz, M. (1986). Concept learning in a rich input domain: generalization-based memory. in: Michalski, R.S., Carbonell, J.G., und Mitchell, T.M. (Eds.) *Machine Learning, Vol. II*, pp. 193-214.

Lebowitz, M. (1987). Experiments with incremental concept formation: UNIMEM. *Machine Learning*, 2, pp. 103-138.

Luck, K. von (1991). Hybride logikbasierte Systeme. *Künstliche Intelligenz*, 2, pp. 27-31.

Maarek, Y.S. (1990). An incremental conceptual clustering algorithm that reduces input-ordering bias. in: Golumbic, M.C. (ed.) *Advances in Artificial Intelligence*, Springer-Verlag, pp. 129-144.

Medin, D.L., und Schaffer, M.M. (1978). Context theory of classification learning. *Psychological Review*, Vol. 85, No. 3, pp. 207-238.

Medin, D.L., und Smith, E.E. (1984). Concepts and concept formation. *Annual Review Psychology*, 35, pp. 113-138.

unifying view. *Machine Learning*, 1, pp. 47-80.

Müller, G. (Hrsg.) (1989). *Lexikon Technologie, Metallverarbeitende Industrie*. Verlag Europa-Lehrmittel, Nourney, Vollmer GmbH & Co., 5657 Haan-Gruiten, Germany.

Murphy, G.L., und Medin, D.L. (1985). The role of theories in conceptual coherence. *Psychological Review*, Vol. 92, No. 3, pp. 289-316.

Murphy, G.L., und Wright, J.C. (1984). Changes in conceptual structure with expertise: differences between real-world experts and novices. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, Vol. 10, No. 1, pp. 144-155.

Neimark, E.D., und Santa, J.L. (1975). Thinking and Concept attainment. *Annual Review Psychology*, 26, pp. 173-205.

Rosch, E. (1975). Cognitive representation of semantic categories. *Journal of Experimental Psychology: General*, Vol. 104, No. 3, pp. 192-233.

Rosch, E. (1978). Principles in categorization. in: Rosch, E., und Lloyd, B. (Eds.), *Cognition and categorization*. Hillsdale, Erlbaum, New Jersey.

Sacerdoti, E.D. (1974). Planning in a hierarchy of abstraction spaces. *Artificial Intelligence*, 5, pp. 115-135.

Sammut, C., und Banerji, R.B. (1986). Learning concepts by asking questions. in: Michalski, R.S., Carbonell, J.G., und Mitchell, T.M. (Eds.) *Machine Learning, Vol. II*, pp. 167-192.

Schank, R.C., und Abelson, R.P. (1977). *Scripts, Plans, Goals, and Understanding*. Hillsdale, New Jersey: Erlbaum.

Schmalhofer, F., Bergmann, R., Kühn, O., und Schmidt, G. (1991). Using an integrated knowledge acquisition method to prepare sophisticated expert plans for the reuse in novel situations. in: Christaller, T. (Ed.), *Proceedings of the 15th German Workshop on Artificial Intelligence*, Springer-Verlag, pp. 62-71.

Schmalhofer, F., Kühn, O., und Schmidt, G. (1991). Integrated knowledge acquisition from text, previously solved cases, and expert memories. *Applied Artificial Intelligence*, 5, pp. 311-337.

Schmalhofer, F., und Reinartz, T. (1991). *Intelligent documentation as a catalyst for developing cooperative knowledge-based systems*. Manuskript.

Schmidt, G., und Schmalhofer, F. (1990). Case-oriented knowledge acquisition from texts. in: Wielinga, B., Boose, J., Gaines, B., Schreiber, G., und van Someren, M. (eds.), *Current trends in knowledge acquisition (EKAW '90)*, Amsterdam: IOS Press, pp. 302-312.

Smith, E.E., und Osherson, D.N. (1988). Conceptual combination with prototype concepts. in: Collins, A., und Smith, E.E. *Readings in Cognitive Science*. Morgan Kaufmann, Publ., Inc., San Mateo, California. pp. 323-335.

SPK-Feldmühle (1984). *Werkzeuge, Examples for application (turning and milling)*, Feldmühle AG, Produktbereich SPK-Werkzeuge, D-7333 Ebersbach a.d. Fils, Germany.

Steinmüller, A. (Ed. u. Leitung) (1987). *Metalltechnik. Grundstufe Fertigungstechnik und Metallbau*.

---

Verlag Europa-Lehrmittel, Nourney, Vollmer GmbH & Co., 5657 Haan-Gruiten, Germany.

Thoben, J., Schmalhofer, F., und Reinartz, T. (1991). *Wiederholungs-, Varianten- und Neuplanung bei der Fertigung rotationssymmetrischer Drehteile*. German Research Center for Artificial Intelligence, Document, No. 91-16.

Tschaitshian, B. (1991). *Eine integrative Wissenserhebung mit CECoS: Konzepte und prototypische*

*Implementierung*. Universität Kaiserslautern. Projektarbeit im Fachbereich Informatik.

Tversky, A. (1977). Features of Similarity. *Psychological Review*, 84, pp. 327-352.

Wrobel, S. (1991a). Die Umweltverankerung von Begriffsbildungsprozessen. *Künstliche Intelligenz*, 1, pp. 22-26.

Wrobel, S. (1991b). *Concepts and concept formation: fundamental issues*. Arbeitspapiere der GMD Subreihe KI, No. 560.

Zadeh, L.A. (1965). Fuzzy Sets. *Information and Control*, 8, pp. 338-353.

## X ANHANG

### X.1 Erläuterungen

Um die Implementierungsversuche leichter verständlich zu machen, wird zunächst die Idee der Repräsentation erläutert. Der Programmtext ist zusätzlich mit einigen Kommentaren versehen, die jeweils mit einem Semikolon beginnen.

Das PD ist in zwei Teile gegliedert. In einem ersten Teil sind die Prädikate der konkreten Domäne der

ist man gezwungen, Informationen explizit anzugeben, die in TEC-REP als implizit definiert gelten, weil sie berechnet werden können.

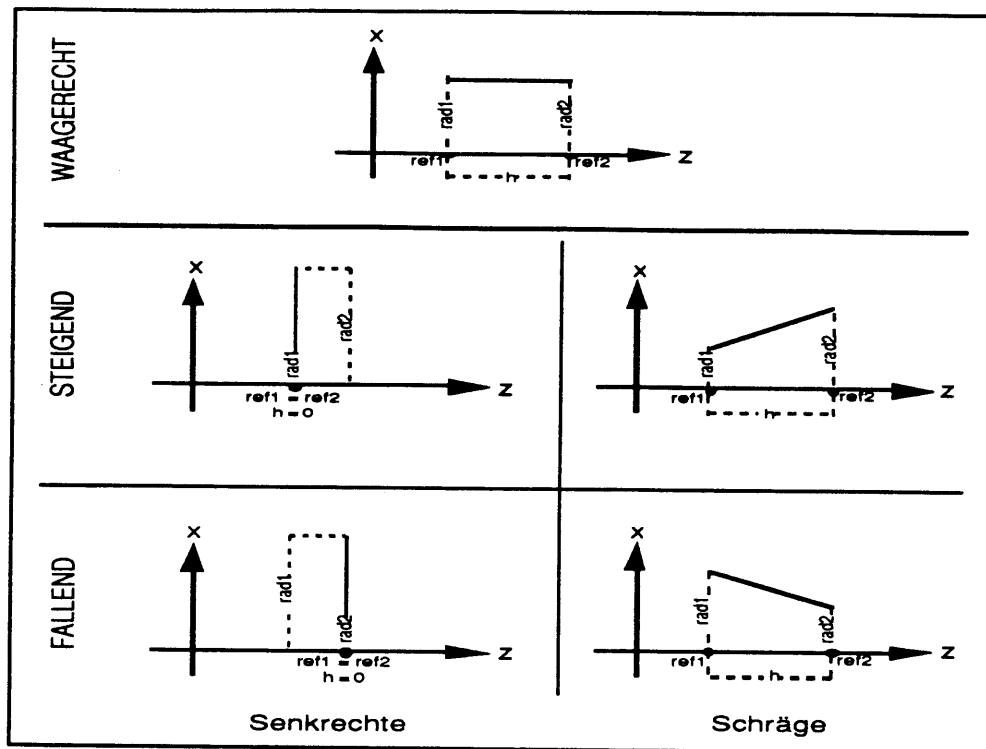


Abbildung 10: Zwischenstücke der geometrischen Repräsentation

Zum Beispiel kann man bei einer Zylindermantelfläche durch Subtraktion der z-Koordinaten die Höhe des Zylinders berechnen. In TAXON ist eine solche Subtraktion nicht möglich, so daß die Höhe einer Zylindermantelfläche explizit angegeben werden muß. Um eine einheitliche Verarbeitung der Flächenprimitive in TAXON zu ermöglichen, ohne daß bekannt ist, um welches Flächenprimitiv es sich genau handelt, sollten alle Primitive die gleiche Repräsentationsform haben. In TEC-REP ist die Anzahl notwendiger Angaben für die Repräsentation eines Primitives nicht immer einheitlich, so daß an dieser Stelle eine Angleichung notwendig war. Diese Angleichung erforderte an einigen Stellen eine Einschränkung der Allgemeinheit. Es wurde dennoch darauf geachtet, daß erstens alle Beispielpunkte repräsentiert werden können, und darüberhinaus eine Vielzahl weiterer neuer Fertigungsprobleme auszudrücken sind.

In Abbildung 9 und 10 sind die Flächenprimitive dargestellt. Es handelt sich im Prinzip um die TEC

Absatz, Zapfen, Mittelabsatz, Kragen, Nut, und Einstich zusammengesetzt werden.

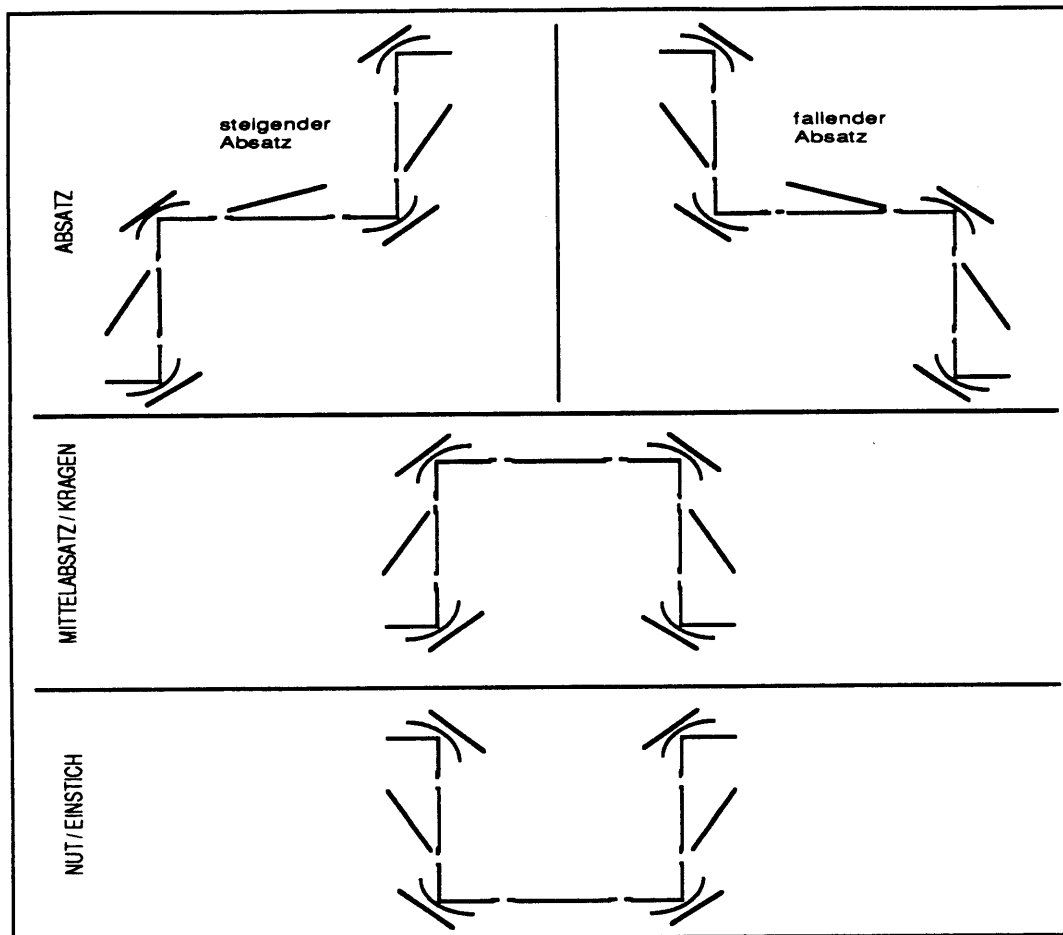


Abbildung 11: Kontur-Elemente der geometrischen Repräsentation

In Abbildung 11 sind die einzelnen Kontur-Elemente gezeigt. Nebeneinander angeordnete geometrische Primitive können alternativ eingesetzt werden. Bei dieser Art der Repräsentation können nicht alle denkbaren Ausnahmefälle der Kontur-Elemente erfasst werden. Ein Experte des Maschinenbaus, der diese Alternativen definiert hat [Höper, 1991], bestätigte aber, daß auf diese Weise bis zu 90 % aller möglichen Kontur-Elemente dargestellt werden können. Alle Kontur-Elemente haben eine einheitliche Grundform. Sie sind aus sieben geometrischen Primitiven zusammengesetzt, wobei sich mit einem Verbindungsstück beginnend jeweils Verbindungsstück und Zwischenstück abwechseln. Durch diese einheitliche Grundform kann unabhängig von der Art des Kontur-Elementes auf wichtige Größen zugegriffen werden. Beispielsweise ist der Anfang eines Kontur-Elementes immer der erste Referenzpunkt des ersten geometrischen Primitives, und das Ende der zweite Referenzpunkt des siebten geometrischen Primitives. Die Höhe eines Kontur-Elementes ist immer hauptsächlich durch die Höhe des vierten geometrischen Primitives bestimmt. Durch diese einheitliche Grundform ist es allerdings notwendig einfache Ausprägungen der Kontur-Elemente komplizierter auszudrücken. Die einfachste Form einer Nut, die aus drei Seiten eines Rechteckes besteht, kann beispielsweise nicht durch drei Strecken definiert werden, sondern muß ebenfalls in sieben Teile zerlegt werden, um die allgemeine Darstellung zu erfüllen. Die gesamte Geometrie setzt sich nun aus einer Folge von Kontur-Elementen zusammen. Zusätzlich gibt man den maximalen Radius und die maximale Höhe der Werkstückgeometrie an.

Ausgehend von diesen Erläuterungen und den zusätzlich im Programmtext enthaltenen Kommentaren sollten die Implementierungsversuche nachvollzogen werden können.



## X.2 TBox

```
; *****
; * PRAEDIKATE DER KONKRETEN DOMAENE *
; *****

; Definition der konkreten Domaene der reellen Zahlen unter dem Namen
; "reell":

(doma reell edom-real-ord)

; Praedikate der konkreten Domaene

(pred gleich
  (reell (x y)
    (= x y)))

(pred ungleich
  (reell (x y)
    (!= x y)))

(pred kleiner
  (reell (x y)
    (< x y)))

(pred kleingleich
  (reell (x y)
    (<= x y)))

(pred groesser
  (reell (x y)
    (> x y)))

(pred groessergleich
  (reell (x y)
    (>= x y)))

(pred gleich-0
  (reell (x)
    (= x 0)))

(pred gleich-1
  (reell (x)
    (= x 1)))

(pred gleich-2
  (reell (x)
    (= x 2)))

(pred gleich-2.06
  (reell (x)
    (= x 2.06)))

(pred kleiner-5
  (reell (x)
```

```

                (< x 5)))

(pred groesser-0
  (reell (x)
    (> x 0)))

(pred groessergleich-0
  (reell (x)
    (>= x 0)))

(pred groessergleich-6
  (reell (x)
    (>= x 6)))

(pred groessergleich-50
  (reell (x)
    (>= x 50)))

(pred groessergleich-5000
  (reell (x)
    (>= x 5000)))

(pred gleich-Vektor
  (reell (x1 z1 x2 z2)
    (and (= x1 x2)
          (= z1 z2))))

(pred ungleich-Vektor
  (reell (x1 z1 x2 z2)
    (or  (= x1 x2)
          (= z1 z2))))

(pred Null-Vektor
  (reell (x z)
    (and (= x 0)
          (= z 0))))

(pred Max-Vektor
  (reell (x z)
    (and (= x 0)
          (= z 210))))

; Dieses Maximum gilt nur fuer die Geometrie 1 !

; *****
; * KONKRETE WERKSTUECK-SICHT *
; *****

; In der konkreten Werkstueck-Sicht sind keine Definitionen erfolgt.

; *****
; * ABSTRAKTE WERKSTUECK-SICHT *
; *****

; Um die Definition der Features vorzubereiten, muss zunaechst
; eine allgemeine Repraesentation der Geometrien entwickelt werden.

```

```

; ein Vektor hat eine x- und eine z- Koordinate

(attr x z)

(conc Vektor
  (and (groessergleich-0 x)
        (groessergleich-0 z)))

; Geometrisches-Primitiv

(attr ref1 ref2 rad1 rad2 h)

(conc Geometrisches-Primitiv
  (and (some ref1 Vektor)
        (some ref2 Vektor)
        (groessergleich-0 rad1)
        (groessergleich-0 rad2)
        (groessergleich-0 h)))

(conc waagerechtes-Geometrisches-Primitiv
  (and Geometrisches-Primitiv
        (ungleich-Vektor (ref1 x) (ref1 z) (ref2 x) (ref2 z))
        (gleich rad1 rad2)
        (groesser-0 h)))

(conc steigendes-Geometrisches-Primitiv
  (and Geometrisches-Primitiv
        (kleiner_rad1 rad2)))



---



(conc fallendes-Geometrisches-Primitiv
  (and Geometrisches-Primitiv
        (groesser rad1 rad2)))

; Verbindungsstuecke

(conc Verbindungsstueck
  (and (ungleich-Vektor (ref1 x) (ref1 z) (ref2 x) (ref2 z))
        (kleiner-5 h)
        (groesser-0 h)))

(conc steigendes-Verbindungsstueck
  (and steigendes-Geometrisches-Primitiv
        Verbindungsstueck))

(conc fallendes-Verbindungsstueck
  (and fallendes-Geometrisches-Primitiv
        Verbindungsstueck))

(attr rad3)

; ein besonderes Verbindungsstueck: Bogen

(conc Bogen
  (and Verbindungsstueck

```

```

(groesser-0 rad3)))

; Zwischenstuecke

; Ein Zwischenstueck ist ein Geometrisches-Primitiv.

; besondere Zwischenstuecke:

; Senkrechte

(conc Senkrechte
  (and (gleich-Vektor (ref1 x) (ref1 z) (ref2 x) (ref2 z))
        (gleich-0 h)))

(conc steigende-Senkrechte
  (and steigendes-Geometrisches-Primitiv
        Senkrechte))

(conc fallende-Senkrechte
  (and fallendes-Geometrisches-Primitiv
        Senkrechte))

; Schraege

(conc Schraege
  (and (ungleich-Vektor (ref1 x) (ref1 z) (ref2 x) (ref2 z))
        (groesser-0 h)))

(conc steigende-Schraege
  (and steigendes-Geometrisches-Primitiv
        Schraege))

(conc fallende-Schraege
  (and fallendes-Geometrisches-Primitiv

```

Schraege))

```

; Kontur-Elemente

; Ein Kontur-Element setzt sich zunaechst aus sieben Geometrischen-
; Primitiven zusammen. Die schwachste Forderung an ein Kontur-
; Element ist der Zusammenhang der einzelnen Geometrischen-Primitive.

```

```

(prim konkav)
(prim konvex)
(prim gerade)

```

```

(attr ge1 ge2 ge3 ge4 ge5 ge6 ge7)

```

```

(conc Kontur-Element
  (and (gleich-Vektor (ge1 ref2 x) (ge1 ref2 z)
                    (ge2 ref1 x) (ge2 ref1 z))
        (gleich-Vektor (ge2 ref2 x) (ge2 ref2 z)
                    (ge3 ref1 x) (ge3 ref1 z))
        (gleich-Vektor (ge3 ref2 x) (ge3 ref2 z)
                    (ge4 ref1 x) (ge4 ref1 z))

```

```

(gleich-Vektor (ge4 ref2 x) (ge4 ref2 z)
               (ge5 ref1 x) (ge5 ref1 z))
(gleich-Vektor (ge5 ref2 x) (ge5 ref2 z)
               (ge6 ref1 x) (ge6 ref1 z))
(gleich-Vektor (ge6 ref2 x) (ge6 ref2 z)
               (ge7 ref1 x) (ge7 ref1 z)))

; verschiedene Kontur-Elemente:

; Nut

(conc Nut
  (and Kontur-Element
        (some ge1 (and fallendes-Verbindungsstueck
                       (or konkav gerade)))
        (some ge2 fallendes-Geometrisches-Primitiv)
        (some ge3 (and fallendes-Verbindungsstueck
                       (or konvex gerade)))
        (some ge4 waagerechtes-Geometrisches-Primitiv)
        (some ge5 (and steigendes-Verbindungsstueck
                       (or konvex gerade)))
        (some ge6 steigendes-Geometrisches-Primitiv)
        (some ge7 (and steigendes-Verbindungsstueck
                       (or konkav gerade))))))

; Einstich
; Ein Einstich ist eine spezielle Nut, die sich durch eine geringe
; Hoehe auszeichnet. Die maximale Hoehe wurde hier mit 5 festgelegt.
; Dieser Wert kann kleiner sein.

(conc Einstich
  (and Nut
        (kleiner-5 (ge4 h))))

; steigender-Absatz

(conc steigender-Absatz
  (and Kontur-Element
        (some ge1 (and steigendes-Verbindungsstueck
                       (or konvex gerade)))
        (some ge2 steigendes-Geometrisches-Primitiv)
        (some ge3 (and steigendes-Verbindungsstueck
                       (or konkav gerade)))
        (some ge4 (or waagerechtes-Geometrisches-Primitiv
                      steigende-Schraege))
        (some ge5 (and steigendes-Verbindungsstueck
                       (or konvex gerade)))
        (some ge6 steigendes-Geometrisches-Primitiv)
        (some ge7 (and steigendes-Verbindungsstueck
                       (or konkav gerade))))))

; fallender-Absatz

(conc fallender-Absatz
  (and Kontur-Element

```

```

      (some ge1 (and fallendes-Verbindungsstueck
                    (or konkav gerade)))
      (some ge2 fallendes-Geometrisches-Primitiv)
      (some ge3 (and fallendes-Verbindungsstueck
                    (or konvex gerade)))
      (some ge4 (or waagerechtes-Geometrisches-Primitiv
                    fallende-Schraege))

```

```

      (some ge5 (and fallendes-Verbindungsstueck
                    (or konkav gerade)))
      (some ge6 fallendes-Geometrisches-Primitiv)
      (some ge7 (and fallendes-Verbindungsstueck
                    (or konvex gerade))))

```

; Absatz

```

(conc Absatz
  (or steigender-Absatz
      fallender-Absatz))

```

; Zapfen

; Ein Zapfen ist ein spezieller Absatz. Ein linker-Zapfen ist ein  
 ; erster Absatz bzw. ein rechter-Zapfen ist ein letzter Absatz,  
 ; der eine gewisse Hoehe nicht ueberschreitet. Auch hier wurde  
 ; das Maximum auf 5 gesetzt.

```

(conc linker-Zapfen
  (and Absatz
        (Null-Vektor (ge1 refl x) (ge1 refl z))
        (kleiner-5 (ge4 h))))

```

```

(conc rechter-Zapfen
  (and Absatz
        (Max-Vektor (ge7 refl x) (ge7 refl z))
        (kleiner-5 (ge4 h))))

```

```

(conc Zapfen
  (or linker-Zapfen
      rechter-Zapfen))

```

; Mittelabsatz

```

(conc Mittelabsatz
  (and Kontur-Element
        (some ge1 (and steigendes-Verbindungsstueck
                      (or konvex gerade)))
        (some ge2 steigendes-Geometrisches-Primitiv)
        (some ge3 (and steigendes-Verbindungsstueck
                      (or konkav gerade)))
        (some ge4 waagerechtes-Geometrisches-Primitiv)
        (some ge5 (and fallendes-Verbindungsstueck
                      (or konkav gerade)))
        (some ge6 fallendes-Geometrisches-Primitiv)
        (some ge7 (and fallendes-Verbindungsstueck
                      (or konvex gerade))))))

```

```

; Kragen
; Ein Kragen ist ein spezieller Mittelabsatz. Ein Kragen zeichnet
; sich wiederum durch eine geringe Hoehe aus, die mit 5 festgesetzt
; wurde.

(conc Kragen
  (and Mittelabsatz
        (kleiner-5 (ge4 h))))

; Gewinde
; Ein Gewinde ist ein besonderes Kontur-Element, da fuer ein Gewinde
; nicht alle Konturkoordinaten bekannt sein muessen. Bei einem Gewinde
; genuegt es, die Anfangs- und Endkoordinaten, sowie die Bezeichnung
; anzugeben. Bei der Fertigung gibt es fuer bestimmte Gewinde feste
; Sequenzen im Plan.

; einige typische Gewindebezeichnungen

(prim M40x2)
(prim M60x2)
(prim M80x2)

(conc Gewinde-Bezeichnung
  (or M40x2
       M60x2
       M80x2))

(attr GA GE GB)

(conc Gewinde
  (and (some GA Vektor)
        (some GE Vektor)
        (some GB Gewinde-Bezeichnung)))

; Eine Geometrie besteht aus Kontur-Elementen, einem maximalen Radius
; und einer maximalen Hoehe. Der Zusammenhang der Kontur-Elemente kann
; nicht ueberprueft werden, da ein Zugriff auf Rollenfueller nicht
; moeglich ist. Da eine Allquantifizierung nicht erfuehrt werden kann,
; wird lediglich die Existenz eines Kontur-Elementes gefordert.

(attr Maxrad Maxh)

(role K)

(attr g)

(conc Geometrie
  (and (groesser-0 Maxrad)
        (groesser-0 Maxh)
        (some K Kontur-Element)))

; Die Werkstoffe sind zunaechst lediglich nach 'duktil' und 'sproede'
; unterschieden. Der Werkstoff Gusseisen bildet eine Ausnahme, da fuer
; diesen Werkstoff der Kohlenstoffgehalt angegeben wurde. Analog zu
; dieser Angabe kann man fuer die anderen Werkstoffe aehnliche Angaben

```

```
; machen (siehe zum Beispiel Domke, 1982). Fuer diesen Zweck ist eine
; grundsaeztliche Unterscheidung nach 'duktil' und 'sproede'
; ausreichend. Weiterhin sind aus dem Spektrum der meoglichen
; Werkstoffe vier ausgewaehlt worden, die fuer die Anwendung relevant
; sind.
```

```
(prim sproede)
```

```
(conc duktil
      (not sproede))
```

```
(conc Edelstahl
      duktil)
```

```
(attr Anteil-Kohlenstoff)
```

```
(conc Gusseisen
      (and sproede
           (gleich-2.06 Anteil-Kohlenstoff)))
```

```
(conc Aluminium
      duktil)
```

```
(conc Verguetungsstahl
      duktil)
```

```
(attr w)
```

```
(conc Werkstoff
      (or Edelstahl
          Gusseisen
          Aluminium
          Verguetungsstahl))
```

```
; weitere Features der abstrakten Werkstueck-Sicht:
```

```
; Welle
```

```
(prim Welle)
```

```
; Features, die im Programmtext auskommentiert sind, wie
; beispielsweise 'ansteigende-Kontur-linke-Absaetze' und
; 'ansteigende-Kontur-rechte-Absaetze', koennen in der
; gewaehlten Repraesentation nicht ohne weiteres dargestellt
; werden.
```

```
; (conc ansteigende-Kontur-linke-Absaetze ...)
```

```
; (conc ansteigende-Kontur-rechte-Absaetze ...)
```

```
(conc zwei-Zapfen
      (and linker-Zapfen
           rechter-Zapfen))
```

```
(conc beidseitig-ansteigende-Welle
```



```

        (and Welle
;           ansteigende-Kontur-linke-Absaetze
;           ansteigende-Kontur-rechte-Absaetze
           Mittelabsatz
           zwei-Zapfen))

(conc mindestens-ein-Einstich
  (some K Einstich))

; (conc Nut-im-Mittelabsatz ...)

; (conc keine-Nut-im-Mittelabsatz
;   (not Nut-im-Mittelabsatz))

; (conc Nut-in-rechten-Absaetzen ...)

; (conc Einstich-direkt-links-am-Kragen ...)

; (conc Einstich-direkt-rechts-am-Kragen ...)

; (conc Einstiche-direkt-am-Kragen
;   (and Einstich-direkt-links-am-Kragen
;         Einstich-direkt-rechts-am-Kragen))

; (conc Gewinde-am-rechten-aeusseren-Zapfen ...)

(prim zylindrisch)

(conc Ausgangsteil-zylindrisch
  zylindrisch)

(conc Ausgangsteil-nicht-zylindrisch
  (not zylindrisch))

(conc Werkstoff-Gusseisen
  (some w Gusseisen))

(conc Werkstoff-sproede
  (some w sproede))

; *****
; * KONKRETE WERKSTATT-SICHT *
; *****

; In der konkreten Werkstatt-Sicht sind keine Definitionen erfolgt.

; *****
; * ABSTRAKTE WERKSTATT-SICHT *
; *****

; Eine Maschine wird durch die Angabe der typischen Kenngrößen
; repräsentiert.

(prim Stirnseitenmitnehmer-mit-Spitze)
(prim Futter-mit-Spitze)

```

```

; Hier sind weitere Aufspannungsmittel denkbar.

(conc Aufspannungsmittel
  (or Stirnseitenmitnehmer-mit-Spitze
    Futter-mit-Spitze))

(prim stabil)

(conc instabil
  (not stabil))

(conc Stabilitaet
  (or stabil
    instabil))

(attr U/MIN KW WZ AM RA ST)

(attr d)

(conc Maschine
  (and (groessergleich-0 U/MIN)           ; Geschwindigkeit
        (groessergleich-0 KW)           ; Leistung
        (groessergleich-0 WZ)           ; Werkzeuge
        (some AM Aufspannungsmittel)
        (groessergleich-0 RA)           ; Revolver
        (some ST Stabilitaet)))

; Bei den Werkzeugen und den Revolvern wird nur die Anzahl festgelegt.

; weitere Features der abstrakten Werkstatt-Sicht:

(conc mindestens-6-Werkzeuge
  (and (some d Maschine)
    (groessergleich-6 (d WZ))))

; *****
; * KONKRETE PLAN-SICHT *
; *****

(prim Null-Zentrierbohrung)
(prim Eine-Zentrierbohrung)
(prim Zwei-Zentrierbohrungen)

(conc beliebige-Anzahl-Zentrierbohrungen
  (and Null-Zentrierbohrung
    Eine-Zentrierbohrung
    Zwei-Zentrierbohrungen))

(conc Aufspannung-mit-Stirnseitenmitnehmer
  (and beliebige-Anzahl-Zentrierbohrungen
    (some d Maschine)
    (some (d AM) Stirnseitenmitnehmer-mit-Spitze)))

; ACHTUNG: Das Feature 'Aufspannung-mit-Stirnseitenmitnehmer' gehoert

```

```
; zu der kombinierten abstrakte Werkstueck, abstrakte Werkstatt,  
; abstrakte Plan - Sicht. Um hier verwendet werden zu koennen, muss es  
; aus technischen Gruenden aber bereits hier definiert werden.
```

```
; *****  
; * ABSTRAKTE PLAN-SICHT *  
; *****
```

```
; Im Grunde ist die Art des Spanbruchs nicht nur abhaengig vom  
; verwendeten Werkstoff, sondern auch von den verwendeten Werkzeugen,  
; der Maschinengeschwindigkeit, etc. In diesem Falle ist aber  
; angenommen worden, dass fuer dieses Fertigungsspektrum Probleme mit  
; dem Spanbruch direkt von der Laenge der Spaene abhaengen, und diese  
; wiederum direkt von der Beschaffenheit des Werkstoffes.
```

```
(conc kurze-Spaene  
    sproede)
```

```
(conc lange-Spaene  
    duktil)
```

```
(conc Spanart  
    (or kurze-Spaene  
        lange-Spaene))
```

```
(conc kein-Problem-Spanbruch  
    kurze-Spaene)
```

```
(conc Problem-Spanbruch  
    lange-Spaene)
```

```
; *****  
; * KOMBINIERTE SICHTEN *  
; *****
```

```
; konkrete Werkstatt, konkrete Plan - Sicht
```

```
(conc Bearbeitung-sequentiell  
    (and (some d Maschine)  
        (gleich-1 (d RA))))
```

```
(conc Bearbeitung-parallel  
    (and (some d Maschine)  
        (gleich-2 (d RA))))
```

```
; abstrakte Werkstueck, abstrakte Werkstatt - Sicht
```

```
(conc Leistungsstarke-Maschine  
    (and (some d Maschine)  
        (groessergleich-5000 (d U/MIN))  
        (groessergleich-50 (d KW))))
```

```
(conc stabile-Maschine  
    (and (some d Maschine)
```

```

; *****
; * DEFINITION EINES FERTIGUNGSPROBLEMS *
; *****

; Ein Fertigungsproblem besteht allgemein aus einer Angabe zu der
; Anzahl erlaubter Zentrierbohrungen, der Form des Ausgangsstueckes
; (hier vereinfachend nur zwischen zylindrisch und nicht zylindrisch
; unterschieden; der Werkstoff wird als korrekt gewaehlt
; vorausgesetzt), einer Geometrie der Zielkontur, einem Werkstoff, und
; einer zur Verfuegung stehenden Maschine.

(conc Zentrierbohrung
  (or Null-Zentrierbohrung
    Eine-Zentrierbohrung
    Zwei-Zentrierbohrungen))

(conc Problem
  (and (or zylindrisch (not zylindrisch))
    Zentrierbohrung
    (some g Geometrie)
    (some w Werkstoff)
    (some d Maschine)))

; *****
; * DEFINITION DER PROBLEMKLASSEN *
; *****

; Features, die in der hier gewaehlten Repraesentation nicht zur
; Verfuegung stehen, sind auskommentiert.

(conc Klasse-5
  (and
;      beidseitig-ansteigende-Welle
;      Leistungsstarke-Maschine
;      stabile-Maschine
;      mindestens-6-Werkzeuge
;      beliebige-Anzahl-Zentrierbohrungen
;      Aufspannung-mit-Stirnseitenmitnehmer
;      mindestens-ein-Einstich
  ))

(conc Klasse-b
  (and
;      Klasse-5
;      Werkstoff-Gusseisen
;      Werkstoff-sproede
;      kein-Problem-Spanbruch
  ))

(conc Klasse-e
  (and
;      Klasse-5
;      Klasse-b
;      keine-Nut-im-Mittelabsatz
  ))

```

```

; Gewinde
Gewinde-am-rechten-aeusseren-Zapfen
))

(conc Klasse-f
  (and
    Klasse-5
    Klasse-b
; Nut-im-Mittelabsatz
Kragen
Ausgangsteil-nicht-zyllindrisch
))

(conc Klasse-i
  (and
    Klasse-5
    Klasse-b
    Klasse-e
    Kragen
; Nut-in-rechten-Absaetzen
; Einstiche-direkt-am-Kragen
Ausgangsteil-nicht-zyllindrisch
))

(conc Klasse-j
  (and
    Klasse-5
    Klasse-b
    Klasse-e
    Ausgangsteil-zyllindrisch
))

(conc Klasse-g1w2d2
  (and
    Klasse-5
    Klasse-b
    Klasse-e
    Klasse-i
    Bearbeitung-sequentiell
))

(conc Klasse-g3w2d2
  (and
    Klasse-5
    Klasse-b
    Klasse-e
    Klasse-j
    Bearbeitung-sequentiell
))

(conc Klasse-g3w2d3
  (and
    Klasse-5
    Klasse-b
    Klasse-e

```

```
        Klasse-j
        Bearbeitung-parallel
    ))

(conc Klasse-g2w2d2
    (and
        Klasse-5
        Klasse-b
        Klasse-f
        Bearbeitung-sequentiell
    ))

(conc Klasse-g2w2d3
    (and
        Klasse-5
        Klasse-b
        Klasse-f
        Bearbeitung-parallel
    ))
```

### X.3 ABox

```
; *****
; * A-BOX *
; *****

; einige Hilfsindividuen

(indi stirnseiten futter stab instab)

(asse (Stirnseitenmitnehmer-mit-Spitze      stirnseiten)
      (Futter-mit-Spitze                    futter)
      (stabil                                stab)
      (instabil                              instab))

; Definition der Geometrie g1

; Geometrie 1

; Geometrie 1, Absatz 1

; Geometrie 1, Absatz 1, ge 1

(indi glabsatzlge1ref1 glabsatzlge1ref2)

(asse (glabsatzlge1ref1      x      0)
      (glabsatzlge1ref1      z      0)
      (glabsatzlge1ref2      x      0)
      (glabsatzlge1ref2      z      0.01))

(indi glabsatzlge1)

(asse (glabsatzlge1  ref1  glabsatzlge1ref1)
      (glabsatzlge1  ref2  glabsatzlge1ref2)
      (glabsatzlge1  rad1  0)
      (glabsatzlge1  rad2  0.01)
      (glabsatzlge1  h     0.01)
      (konvex        glabsatzlge1))

; Geometrie 1, Absatz 1, ge 2

(indi glabsatzlge2ref1 glabsatzlge2ref2)

(asse (glabsatzlge2ref1      x      0)
      (glabsatzlge2ref1      z      0.01)
      (glabsatzlge2ref2      x      0)
      (glabsatzlge2ref2      z      0.01))

(indi glabsatzlge2)

(asse (glabsatzlge2  ref1  glabsatzlge2ref1)
      (glabsatzlge2  ref2  glabsatzlge2ref2)
      (glabsatzlge2  rad1  0)
      (glabsatzlge2  rad2  15)
      (glabsatzlge2  h     0))
```

```

; Geometrie 1, Absatz 1, ge 3

(indi glabsatzlge3ref1 glabsatzlge3ref2)

(asse (glabsatzlge3ref1 x 0)
      (glabsatzlge3ref1 z 0.01)
      (glabsatzlge3ref2 x 0)
      (glabsatzlge3ref2 z 2))

(indi glabsatzlge3)

(asse (glabsatzlge3 ref1 glabsatzlge3ref1)
      (glabsatzlge3 ref2 glabsatzlge3ref2)
      (glabsatzlge3 rad1 15)
      (glabsatzlge3 rad2 17)
      (glabsatzlge3 h 2)
      (gerade glabsatzlge3))

; Geometrie 1, Absatz 1, ge 4

(indi glabsatzlge4ref1 glabsatzlge4ref2)

(asse (glabsatzlge4ref1 x 0)
      (glabsatzlge4ref1 z 2)
      (glabsatzlge4ref2 x 0)
      (glabsatzlge4ref2 z 14.8))

(indi glabsatzlge4)

(asse (glabsatzlge4 ref1 glabsatzlge4ref1)
      (glabsatzlge4 ref2 glabsatzlge4ref2)
      (glabsatzlge4 rad1 17)
      (glabsatzlge4 rad2 17)
      (glabsatzlge4 h 12.8))

; Geometrie 1, Absatz 1, ge 5

(indi glabsatzlge5ref1 glabsatzlge5ref2)

(asse (glabsatzlge5ref1 x 0)
      (glabsatzlge5ref1 z 14.8)
      (glabsatzlge5ref2 x 0)
      (glabsatzlge5ref2 z 15))

(indi glabsatzlge5)

(asse (glabsatzlge5 ref1 glabsatzlge5ref1)
      (glabsatzlge5 ref2 glabsatzlge5ref2)
      (glabsatzlge5 rad1 17)
      (glabsatzlge5 rad2 17.2)
      (glabsatzlge5 h 0.2)
      (konvex glabsatzlge5))

; Geometrie 1, Absatz 1, ge 6

```



(indi glabsatzlge6ref1 glabsatzlge6ref2)

(asse (glabsatzlge6ref1 x 0)  
(glabsatzlge6ref1 z 15)  
(glabsatzlge6ref2 x 0)  
(glabsatzlge6ref2 z 15))

(indi glabsatzlge6)

(asse (glabsatzlge6 ref1 glabsatzlge6ref1)  
(glabsatzlge6 ref2 glabsatzlge6ref2)  
(glabsatzlge6 rad1 17.2)  
(glabsatzlge6 rad2 18.8)  
(glabsatzlge6 h 0))

; Geometrie 1, Absatz 1, ge 7

(indi glabsatzlge7ref1 glabsatzlge7ref2)

(asse (glabsatzlge7ref1 x 0)  
(glabsatzlge7ref1 z 15)  
(glabsatzlge7ref2 x 0)  
(glabsatzlge7ref2 z 15.2))

(indi glabsatzlge7)

(asse (glabsatzlge7 ref1 glabsatzlge7ref1)  
(glabsatzlge7 ref2 glabsatzlge7ref2)  
(glabsatzlge7 rad1 18.8)  
(glabsatzlge7 rad2 19)  
(glabsatzlge7 h 0.2)  
(konkav glabsatzlge7))

(indi glabsatzl)

(asse (glabsatzl ge1 glabsatzlge1)  
(glabsatzl ge2 glabsatzlge2)  
(glabsatzl ge3 glabsatzlge3)  
(glabsatzl ge4 glabsatzlge4)  
(glabsatzl ge5 glabsatzlge5)  
(glabsatzl ge6 glabsatzlge6)  
(glabsatzl ge7 glabsatzlge7))

; Geometrie 1

; Geometrie 1, Absatz 2

; Geometrie 1, Absatz 2, ge 1

(indi glabsatz2gelref1 glabsatz2gelref2)

(asse (glabsatz2gelref1 x 0)  
(glabsatz2gelref1 z 14.8)

```

      (glabsatz2gelref2      x      0)
      (glabsatz2gelref2      z      15))

(indi glabsatz2gel)

(asse  (glabsatz2gel  ref1  glabsatz2gelref1)
      (glabsatz2gel  ref2  glabsatz2gelref2)
      (glabsatz2gel  rad1  17)
      (glabsatz2gel  rad2  17.2)
      (glabsatz2gel  h     0.2)
      (konvex        glabsatz2gel))

```

; Geometrie 1, Absatz 2, ge 2

```

(indi glabsatz2ge2ref1 glabsatz2ge2ref2)

(asse  (glabsatz2ge2ref1      x      0)
      (glabsatz2ge2ref1      z      15)
      (glabsatz2ge2ref2      x      0)
      (glabsatz2ge2ref2      z      15))

```

```

(indi glabsatz2ge2)

(asse  (glabsatz2ge2  ref1  glabsatz2ge2ref1)
      (glabsatz2ge2  ref2  glabsatz2ge2ref2)
      (glabsatz2ge2  rad1  17.2)
      (glabsatz2ge2  rad2  18.8)
      (glabsatz2ge2  h     0))

```

; Geometrie 1, Absatz 2, ge 3

```

(indi glabsatz2ge3ref1 glabsatz2ge3ref2)

(asse  (glabsatz2ge3ref1      x      0)
      (glabsatz2ge3ref1      z      15)
      (glabsatz2ge3ref2      x      0)
      (glabsatz2ge3ref2      z      15.2))

```

```

(indi glabsatz2ge3)

(asse  (glabsatz2ge3  ref1  glabsatz2ge3ref1)
      (glabsatz2ge3  ref2  glabsatz2ge3ref2)
      (glabsatz2ge3  rad1  18.8)
      (glabsatz2ge3  rad2  19)
      (glabsatz2ge3  h     0.2)
      (konkav        glabsatz2ge3))

```

; Geometrie 1, Absatz 2, ge 4

```

(indi glabsatz2ge4ref1 glabsatz2ge4ref2)

(asse  (glabsatz2ge4ref1      x      0)
      (glabsatz2ge4ref1      z      15.2)
      (glabsatz2ge4ref2      x      0))

```

(indi glabsatz2ge4)

(asse (glabsatz2ge4 ref1 glabsatz2ge4ref1)  
(glabsatz2ge4 ref2 glabsatz2ge4ref2)  
(glabsatz2ge4 rad1 19)  
(glabsatz2ge4 rad2 19)  
(glabsatz2ge4 h 22.6))

; Geometrie 1, Absatz 2, ge 5

(indi glabsatz2ge5ref1 glabsatz2ge5ref2)

(asse (glabsatz2ge5ref1 x 0)  
(glabsatz2ge5ref1 z 37.8)  
(glabsatz2ge5ref2 x 0)  
(glabsatz2ge5ref2 z 38))

(indi glabsatz2ge5)

(asse (glabsatz2ge5 ref1 glabsatz2ge5ref1)  
(glabsatz2ge5 ref2 glabsatz2ge5ref2)  
(glabsatz2ge5 rad1 19)  
(glabsatz2ge5 rad2 19.2)  
(glabsatz2ge5 h 0.2)  
(konvex glabsatz2ge5))

; Geometrie 1, Absatz 2, ge 6

(indi glabsatz2ge6ref1 glabsatz2ge6ref2)

(asse (glabsatz2ge6ref1 x 0)  
(glabsatz2ge6ref1 z 38)  
(glabsatz2ge6ref2 x 0)  
(glabsatz2ge6ref2 z 38))

(indi glabsatz2ge6)

(asse (glabsatz2ge6 ref1 glabsatz2ge6ref1)  
(glabsatz2ge6 ref2 glabsatz2ge6ref2)  
(glabsatz2ge6 rad1 19.2)  
(glabsatz2ge6 rad2 26)  
(glabsatz2ge6 h 0))

; Geometrie 1, Absatz 2, ge 7

(indi glabsatz2ge7ref1 glabsatz2ge7ref2)

(asse (glabsatz2ge7ref1 x 0)

(glabsatz2ge7ref2 z 40))

(indi glabsatz2ge7)

```
(asse (glabsatz2ge7 ref1 glabsatz2ge7ref1)
      (glabsatz2ge7 ref2 glabsatz2ge7ref2)
      (glabsatz2ge7 rad1 26)
      (glabsatz2ge7 rad2 28)
      (glabsatz2ge7 rad3 0.1)
      (glabsatz2ge7 h 2)
      (konkav glabsatz2ge7))
```

```
(indi glabsatz2)
```

```
(asse (glabsatz2 ge1 glabsatz2ge1)
      (glabsatz2 ge2 glabsatz2ge2)
      (glabsatz2 ge3 glabsatz2ge3)
      (glabsatz2 ge4 glabsatz2ge4)
      (glabsatz2 ge5 glabsatz2ge5)
      (glabsatz2 ge6 glabsatz2ge6)
      (glabsatz2 ge7 glabsatz2ge7))
```

```
; Geometrie 1
```

```
; Geometrie 1, Absatz 3
```

```
; Geometrie 1, Absatz 3, ge 1
```

```
(indi glabsatz3gelref1 glabsatz3gelref2)
```

```
(asse (glabsatz3gelref1 x 0)
      (glabsatz3gelref1 z 37.8)
      (glabsatz3gelref2 x 0)
      (glabsatz3gelref2 z 38))
```

```
(indi glabsatz3gel)
```

```
(asse (glabsatz3gel ref1 glabsatz3gelref1)
      (glabsatz3gel ref2 glabsatz3gelref2)
      (glabsatz3gel rad1 19)
      (glabsatz3gel rad2 19.2)
      (glabsatz3gel h 0.2)
      (konvex glabsatz3gel))
```

```
; Geometrie 1, Absatz 3, ge 2
```

```
(indi glabsatz3ge2ref1 glabsatz3ge2ref2)
```

```
(asse (glabsatz3ge2ref1 x 0)
      (glabsatz3ge2ref1 z 38)
      (glabsatz3ge2ref2 x 0)
      (glabsatz3ge2ref2 z 38))
```

```

        (glabsatz3ge2   rad2   26)
        (glabsatz3ge2   h     0))

; Geometrie 1, Absatz 3, ge 3

(indi glabsatz3ge3ref1 glabsatz3ge3ref2)

(asse   (glabsatz3ge3ref1   x     0)
        (glabsatz3ge3ref1   z    38)
        (glabsatz3ge3ref2   x     0)
        (glabsatz3ge3ref2   z    40))

(indi glabsatz3ge3)

(asse   (glabsatz3ge3   ref1   glabsatz3ge3ref1)
        (glabsatz3ge3   ref2   glabsatz3ge3ref2)
        (glabsatz3ge3   rad1   26)
        (glabsatz3ge3   rad2   28)
        (glabsatz3ge3   rad3   0.1)
        (glabsatz3ge3   h     2)
        (konkav          glabsatz3ge3))

; Geometrie 1, Absatz 3, ge 4

(indi glabsatz3ge4ref1 glabsatz3ge4ref2)

(asse   (glabsatz3ge4ref1   x     0)
        (glabsatz3ge4ref1   z    40)
        (glabsatz3ge4ref2   x     0)
        (glabsatz3ge4ref2   z    45))

(indi glabsatz3ge4)

(asse   (glabsatz3ge4   ref1   glabsatz3ge4ref1)
        (glabsatz3ge4   ref2   glabsatz3ge4ref2)
        (glabsatz3ge4   rad1   28)
        (glabsatz3ge4   rad2   28)
        (glabsatz3ge4   h     5))

; Geometrie 1, Absatz 3, ge 5

(indi glabsatz3ge5ref1 glabsatz3ge5ref2)

(asse   (glabsatz3ge5ref1   x     0)
        (glabsatz3ge5ref1   z    45)
        (glabsatz3ge5ref2   x     0)
        (glabsatz3ge5ref2   z    47))

(indi glabsatz3ge5)

(asse   (glabsatz3ge5   ref1   glabsatz3ge5ref1)
        (glabsatz3ge5   ref2   glabsatz3ge5ref2)
        (glabsatz3ge5   rad1   28)
        (glabsatz3ge5   rad2   26)
        (glabsatz3ge5   rad3   0.1))

```

```

      (glabsatz3ge5   h      2)
      (konkav        glabsatz3ge5))

; Geometrie 1, Absatz 3, ge 6

(indi glabsatz3ge6ref1 glabsatz3ge6ref2)

(asse  (glabsatz3ge6ref1   x      0)
       (glabsatz3ge6ref1   z      47)
       (glabsatz3ge6ref2   x      0)
       (glabsatz3ge6ref2   z      47))

(indi glabsatz3ge6)

(asse  (glabsatz3ge6   ref1   glabsatz3ge6ref1)
       (glabsatz3ge6   ref2   glabsatz3ge6ref2)
       (glabsatz3ge6   rad1   26)
       (glabsatz3ge6   rad2   19.2)
       (glabsatz3ge6   h      0))

; Geometrie 1, Absatz 3, ge 7

(indi glabsatz3ge7ref1 glabsatz3ge7ref2)

(asse  (glabsatz3ge7ref1   x      0)
       (glabsatz3ge7ref1   z      47)
       (glabsatz3ge7ref2   x      0)
       (glabsatz3ge7ref2   z      47.2))

(indi glabsatz3ge7)

(asse  (glabsatz3ge7   ref1   glabsatz3ge7ref1)
       (glabsatz3ge7   ref2   glabsatz3ge7ref2)
       (glabsatz3ge7   rad1   19.2)
       (glabsatz3ge7   rad2   19)
       (glabsatz3ge7   h      0.2)
       (konvex        glabsatz3ge7))

(indi glabsatz3)

(asse  (glabsatz3   ge1   glabsatz3ge1)
       (glabsatz3   ge2   glabsatz3ge2)
       (glabsatz3   ge3   glabsatz3ge3)
       (glabsatz3   ge4   glabsatz3ge4)
       (glabsatz3   ge5   glabsatz3ge5)
       (glabsatz3   ge6   glabsatz3ge6)
       (glabsatz3   ge7   glabsatz3ge7))

; Geometrie 1

; Geometrie 1, Absatz 4

; Geometrie 1, Absatz 4, ge 1

```

```

(indi glabsatz4gelref1 glabsatz4gelref2)

(asse (glabsatz4gelref1 x 0)
      (glabsatz4gelref1 z 45)
      (glabsatz4gelref2 x 0)
      (glabsatz4gelref2 z 47))

(indi glabsatz4gel)

(asse (glabsatz4gel ref1 glabsatz4gelref1)
      (glabsatz4gel ref2 glabsatz4gelref2)
      (glabsatz4gel rad1 28)
      (glabsatz4gel rad2 26)
      (glabsatz4gel rad3 0.1)
      (glabsatz4gel h 2)
      (konkav glabsatz4gel))

; Geometrie 1, Absatz 4, ge 2

(indi glabsatz4ge2ref1 glabsatz4ge2ref2)

(asse (glabsatz4ge2ref1 x 0)
      (glabsatz4ge2ref1 z 47)
      (glabsatz4ge2ref2 x 0)
      (glabsatz4ge2ref2 z 47))

(indi glabsatz4ge2)

(asse (glabsatz4ge2 ref1 glabsatz4ge2ref1)
      (glabsatz4ge2 ref2 glabsatz4ge2ref2)
      (glabsatz4ge2 rad1 26)
      (glabsatz4ge2 rad2 19.2)
      (glabsatz4ge2 h 0))

; Geometrie 1, Absatz 4, ge 3

(indi glabsatz4ge3ref1 glabsatz4ge3ref2)

(asse (glabsatz4ge3ref1 x 0)
      (glabsatz4ge3ref1 z 47)
      (glabsatz4ge3ref2 x 0)
      (glabsatz4ge3ref2 z 47.2))

(indi glabsatz4ge3)

(asse (glabsatz4ge3 ref1 glabsatz4ge3ref1)
      (glabsatz4ge3 ref2 glabsatz4ge3ref2)
      (glabsatz4ge3 rad1 19.2)
      (glabsatz4ge3 rad2 19)
      (glabsatz4ge3 h 0.2)
      (konvex glabsatz4ge3))

; Geometrie 1, Absatz 4, ge 4

```

```

(indi glabsatz4ge4ref1 glabsatz4ge4ref2)

(asse (glabsatz4ge4ref1 x 0)
      (glabsatz4ge4ref1 z 47.2)
      (glabsatz4ge4ref2 x 0)
      (glabsatz4ge4ref2 z 65))

(indi glabsatz4ge4)

(asse (glabsatz4ge4 ref1 glabsatz4ge4ref1)
      (glabsatz4ge4 ref2 glabsatz4ge4ref2)
      (glabsatz4ge4 rad1 19)
      (glabsatz4ge4 rad2 19)
      (glabsatz4ge4 h 17.8))

; Geometrie 1, Absatz 4, ge 5

(indi glabsatz4ge5ref1 glabsatz4ge5ref2)

(asse (glabsatz4ge5ref1 x 0)
      (glabsatz4ge5ref1 z 65)
      (glabsatz4ge5ref2 x 0)
      (glabsatz4ge5ref2 z 67))

(indi glabsatz4ge5)

(asse (glabsatz4ge5 ref1 glabsatz4ge5ref1)
      (glabsatz4ge5 ref2 glabsatz4ge5ref2)
      (glabsatz4ge5 rad1 19)
      (glabsatz4ge5 rad2 17)
      (glabsatz4ge5 rad3 0.1)
      (glabsatz4ge5 h 2)
      (konkav glabsatz4ge5))

; Geometrie 1, Absatz 4, ge 6

(indi glabsatz4ge6ref1 glabsatz4ge6ref2)

(asse (glabsatz4ge6ref1 x 0)
      (glabsatz4ge6ref1 z 67)
      (glabsatz4ge6ref2 x 0)
      (glabsatz4ge6ref2 z 67))

(indi glabsatz4ge6)

(asse (glabsatz4ge6 ref1 glabsatz4ge6ref1)
      (glabsatz4ge6 ref2 glabsatz4ge6ref2)
      (glabsatz4ge6 rad1 17)
      (glabsatz4ge6 rad2 13.2)
      (glabsatz4ge6 h 0))

; Geometrie 1, Absatz 4, ge 7

(indi glabsatz4ge7ref1 glabsatz4ge7ref2)

```



```

(asse (glabsatz4ge7ref1 x 0)
      (glabsatz4ge7ref1 z 67)
      (glabsatz4ge7ref2 x 0)
      (glabsatz4ge7ref2 z 67.2))

(indi glabsatz4ge7)

(asse (glabsatz4ge7 ref1 glabsatz4ge7ref1)
      (glabsatz4ge7 ref2 glabsatz4ge7ref2)
      (glabsatz4ge7 rad1 13.2)
      (glabsatz4ge7 rad2 13)
      (glabsatz4ge7 h 0.2)
      (konvex glabsatz4ge7))

(indi glabsatz4)

(asse (glabsatz4 ge1 glabsatz4ge1)
      (glabsatz4 ge2 glabsatz4ge2)
      (glabsatz4 ge3 glabsatz4ge3)
      (glabsatz4 ge4 glabsatz4ge4)
      (glabsatz4 ge5 glabsatz4ge5)
      (glabsatz4 ge6 glabsatz4ge6)
      (glabsatz4 ge7 glabsatz4ge7))

; Geometrie 1

; Geometrie 1, Absatz 5

; Geometrie 1, Absatz 5, ge 1

(indi glabsatz5gelref1 glabsatz5gelref2)

(asse (glabsatz5gelref1 x 0)
      (glabsatz5gelref1 z 65)
      (glabsatz5gelref2 x 0)
      (glabsatz5gelref2 z 67))

(indi glabsatz5gel)

(asse (glabsatz5gel ref1 glabsatz5gelref1)
      (glabsatz5gel ref2 glabsatz5gelref2)
      (glabsatz5gel rad1 19)
      (glabsatz5gel rad2 17)
      (glabsatz5gel rad3 0.1)
      (glabsatz5gel h 2)
      (konkav glabsatz5gel))

; Geometrie 1, Absatz 5, ge 2

(indi glabsatz5ge2ref1 glabsatz5ge2ref2)

(asse (glabsatz5ge2ref1 x 0)
      (glabsatz5ge2ref1 z 67)
      (glabsatz5ge2ref2 x 0)
      (glabsatz5ge2ref2 z 67))

```

```

(indi glabsatz5ge2)

(asse (glabsatz5ge2 ref1 glabsatz5ge2ref1)
      (glabsatz5ge2 ref2 glabsatz5ge2ref2)
      (glabsatz5ge2 rad1 17)
      (glabsatz5ge2 rad2 13.2)
      (glabsatz5ge2 h 0))

; Geometrie 1, Absatz 5, ge 3

(indi glabsatz5ge3ref1 glabsatz5ge3ref2)

(asse (glabsatz5ge3ref1 x 0)
      (glabsatz5ge3ref1 z 67)
      (glabsatz5ge3ref2 x 0)
      (glabsatz5ge3ref2 z 67.2))

(indi glabsatz5ge3)

(asse (glabsatz5ge3 ref1 glabsatz5ge3ref1)
      (glabsatz5ge3 ref2 glabsatz5ge3ref2)
      (glabsatz5ge3 rad1 13.2)
      (glabsatz5ge3 rad2 13)
      (glabsatz5ge3 h 0.2)
      (konvex glabsatz5ge3))

; Geometrie 1, Absatz 5, ge 4

(indi glabsatz5ge4ref1 glabsatz5ge4ref2)

(asse (glabsatz5ge4ref1 x 0)
      (glabsatz5ge4ref1 z 67.2)
      (glabsatz5ge4ref2 x 0)
      (glabsatz5ge4ref2 z 193))

(indi glabsatz5ge4)

(asse (glabsatz5ge4 ref1 glabsatz5ge4ref1)
      (glabsatz5ge4 ref2 glabsatz5ge4ref2)
      (glabsatz5ge4 rad1 13)
      (glabsatz5ge4 rad2 13)
      (glabsatz5ge4 h 125.8))

; Geometrie 1, Absatz 5, ge 5

(indi glabsatz5ge5ref1 glabsatz5ge5ref2)

(asse (glabsatz5ge5ref1 x 0)
      (glabsatz5ge5ref1 z 193)
      (glabsatz5ge5ref2 x 0)
      (glabsatz5ge5ref2 z 195))

(indi glabsatz5ge5)

```

```
(asse (glabsatz5ge5 ref1 glabsatz5ge5ref1)
      (glabsatz5ge5 ref2 glabsatz5ge5ref2)
      (glabsatz5ge5 rad1 13)
      (glabsatz5ge5 rad2 10)
      (glabsatz5ge5 h 2)
      (gerade glabsatz5ge5))
```

; Geometrie 1, Absatz 5, ge 6

```
(indi glabsatz5ge6ref1 glabsatz5ge6ref2)
```

```
(asse (glabsatz5ge6ref1 x 0)
      (glabsatz5ge6ref1 z 195)
      (glabsatz5ge6ref2 x 0)
      (glabsatz5ge6ref2 z 195))
```

```
(indi glabsatz5ge6)
```

```
(asse (glabsatz5ge6 ref1 glabsatz5ge6ref1)
      (glabsatz5ge6 ref2 glabsatz5ge6ref2)
      (glabsatz5ge6 rad1 10)
      (glabsatz5ge6 rad2 7.2)
      (glabsatz5ge6 h 0))
```

; Geometrie 1, Absatz 5, ge 7

```
(indi glabsatz5ge7ref1 glabsatz5ge7ref2)
```

```
(asse (glabsatz5ge7ref1 x 0)
      (glabsatz5ge7ref1 z 195)
      (glabsatz5ge7ref2 x 0)
      (glabsatz5ge7ref2 z 195.2))
```

```
(indi glabsatz5ge7)
```

```
(asse (glabsatz5ge7 ref1 glabsatz5ge7ref1)
      (glabsatz5ge7 ref2 glabsatz5ge7ref2)
      (glabsatz5ge7 rad1 7.2)
      (glabsatz5ge7 rad2 7)
      (glabsatz5ge7 h 0.2)
      (konvex glabsatz5ge7))
```

```
(indi glabsatz5)
```

```
(asse (glabsatz5 ge1 glabsatz5ge1)
      (glabsatz5 ge2 glabsatz5ge2)
      (glabsatz5 ge3 glabsatz5ge3))
```

```
(glabsatz5 ge5 glabsatz5ge5)
(glabsatz5 ge6 glabsatz5ge6)
(glabsatz5 ge7 glabsatz5ge7))
```

; Geometrie 1

; Geometrie 1, Absatz 6

; Geometrie 1, Absatz 6, ge 1

(indi glabsatz6gelref1 glabsatz6gelref2)

(asse (glabsatz6gelref1 x 0)  
(glabsatz6gelref1 z 193)  
(glabsatz6gelref2 x 0)  
(glabsatz6gelref2 z 195))

(indi glabsatz6gel)

(asse (glabsatz6gel ref1 glabsatz6gelref1)  
(glabsatz6gel ref2 glabsatz6gelref2)  
(glabsatz6gel rad1 13)  
(glabsatz6gel rad2 10)  
(glabsatz6gel h 2)  
(gerade glabsatz6gel))

; Geometrie 1, Absatz 6, ge 2

(indi glabsatz6ge2ref1 glabsatz6ge2ref2 glabsatz6ge2ref3)

(asse (glabsatz6ge2ref1 x 0)  
(glabsatz6ge2ref1 z 195)  
(glabsatz6ge2ref2 x 0)  
(glabsatz6ge2ref2 z 195))

(indi glabsatz6ge2)

(asse (glabsatz6ge2 ref1 glabsatz6ge2ref1)  
(glabsatz6ge2 ref2 glabsatz6ge2ref2)  
(glabsatz6ge2 rad1 10)  
(glabsatz6ge2 rad2 7.2)  
(glabsatz6ge2 h 0))

; Geometrie 1, Absatz 6, ge 3

(indi glabsatz6ge3ref1 glabsatz6ge3ref2)

(asse (glabsatz6ge3ref1 x 0)  
(glabsatz6ge3ref1 z 195)  
(glabsatz6ge3ref2 x 0)  
(glabsatz6ge3ref2 z 195.2))

(indi glabsatz6ge3)

(glabsatz6ge3 ref2 glabsatz6ge3ref2)  
(glabsatz6ge3 rad1 7.2)  
(glabsatz6ge3 rad2 7)  
(glabsatz6ge3 h 0.2)  
(konvex glabsatz6ge3))

```

(indi glabsatz6ge4ref1 glabsatz6ge4ref2)

(asse (glabsatz6ge4ref1 x 0)
      (glabsatz6ge4ref1 z 195.2)
      (glabsatz6ge4ref2 x 0)
      (glabsatz6ge4ref2 z 209.8))

(indi glabsatz6ge4)

(asse (glabsatz6ge4 ref1 glabsatz6ge4ref1)
      (glabsatz6ge4 ref2 glabsatz6ge4ref2)
      (glabsatz6ge4 rad1 7)
      (glabsatz6ge4 rad2 7)
      (glabsatz6ge4 h 14.6))

; Geometrie 1, Absatz 6, ge 5

(indi glabsatz6ge5ref1 glabsatz6ge5ref2)

(asse (glabsatz6ge5ref1 x 0)
      (glabsatz6ge5ref1 z 209.8)
      (glabsatz6ge5ref2 x 0)
      (glabsatz6ge5ref2 z 210))

(indi glabsatz6ge5)

(asse (glabsatz6ge5 ref1 glabsatz6ge5ref1)
      (glabsatz6ge5 ref2 glabsatz6ge5ref2)
      (glabsatz6ge5 rad1 7)
      (glabsatz6ge5 rad2 6.8)
      (glabsatz6ge5 h 0.2)
      (konkav glabsatz6ge5))

; Geometrie 1, Absatz 6, ge 6

(indi glabsatz6ge6ref1 glabsatz6ge6ref2)

(asse (glabsatz6ge6ref1 x 0)
      (glabsatz6ge6ref1 z 210)
      (glabsatz6ge6ref2 x 0)
      (glabsatz6ge6ref2 z 210))

(indi glabsatz6ge6)

(asse (glabsatz6ge6 ref1 glabsatz6ge6ref1)
      (glabsatz6ge6 ref2 glabsatz6ge6ref2)
      (glabsatz6ge6 rad1 6.8)
      (glabsatz6ge6 rad2 0)
      (glabsatz6ge6 h 0))

; Geometrie 1, Absatz 6, ge 7

(indi glabsatz6ge7ref1 glabsatz6ge7ref2)

```

```
(asse (glabsatz6ge7ref1 x 0)
      (glabsatz6ge7ref1 z 210)
      (glabsatz6ge7ref2 x 0)
      (glabsatz6ge7ref2 z 210.001))
```

```
(indi glabsatz6ge7)
```

```
(asse (glabsatz6ge7 ref1 glabsatz6ge7ref1)
      (glabsatz6ge7 ref2 glabsatz6ge7ref2)
      (glabsatz6ge7 rad1 0.001)
      (glabsatz6ge7 rad2 0)
      (glabsatz6ge7 h 0.001)
      (konvex glabsatz6ge7))
```

```
(indi glabsatz6)
```

```
(asse (glabsatz6 ge1 glabsatz6ge1)
      (glabsatz6 ge2 glabsatz6ge2)
      (glabsatz6 ge3 glabsatz6ge3)
      (glabsatz6 ge4 glabsatz6ge4)
      (glabsatz6 ge5 glabsatz6ge5)
      (glabsatz6 ge6 glabsatz6ge6)
      (glabsatz6 ge7 glabsatz6ge7))
```

```
; Geometrie 1
```

```
; Geometrie 1, Einstich 1
```

```
; Geometrie 1, Einstich 1, ge 1
```

```
(indi gleinstichlgelref1 gleinstichlgelref2)
```

```
(asse (gleinstichlgelref1 x 0)
      (gleinstichlgelref1 z 1.8)
      (gleinstichlgelref2 x 0)
      (gleinstichlgelref2 z 2))
```

```
(indi gleinstichlgel)
```

```
(asse (gleinstichlgel ref1 gleinstichlgelref1)
      (gleinstichlgel ref2 gleinstichlgelref2)
      (gleinstichlgel rad1 17)
      (gleinstichlgel rad2 16.8))
```

```

(indi gleinstichlge2)

(asse (gleinstichlge2 ref1 gleinstichlge2ref1)
      (gleinstichlge2 ref2 gleinstichlge2ref2)
      (gleinstichlge2 rad1 16.8)
      (gleinstichlge2 rad2 15.2)
      (gleinstichlge2 h 0))

; Geometrie 1, Einstich 1, ge 3

(indi gleinstichlge3ref1 gleinstichlge3ref2)

(asse (gleinstichlge3ref1 x 0)
      (gleinstichlge3ref1 z 2)
      (gleinstichlge3ref2 x 0)
      (gleinstichlge3ref2 z 2.2))

(indi gleinstichlge3)

(asse (gleinstichlge3 ref1 gleinstichlge3ref1)
      (gleinstichlge3 ref2 gleinstichlge3ref2)
      (gleinstichlge3 rad1 15.2)
      (gleinstichlge3 rad2 15)
      (gleinstichlge3 h 0.2)
      (konvex gleinstichlge3))

; Geometrie 1, Einstich 1, ge 4

(indi gleinstichlge4ref1 gleinstichlge4ref2)

(asse (gleinstichlge4ref1 x 0)
      (gleinstichlge4ref1 z 2.2)
      (gleinstichlge4ref2 x 0)
      (gleinstichlge4ref2 z 3.8))

(indi gleinstichlge4)

(asse (gleinstichlge4 ref1 gleinstichlge4ref1)
      (gleinstichlge4 ref2 gleinstichlge4ref2)
      (gleinstichlge4 rad1 15)
      (gleinstichlge4 rad2 15)
      (gleinstichlge4 h 1.6))

; Geometrie 1, Einstich 1, ge 5

(indi gleinstichlge5ref1 gleinstichlge5ref2)

(asse (gleinstichlge5ref1 x 0)
      (gleinstichlge5ref1 z 3.8)
      (gleinstichlge5ref2 x 0)
      (gleinstichlge5ref2 z 4))

(indi gleinstichlge5)

(asse (gleinstichlge5 ref1 gleinstichlge5ref1)

```

```

        (gleinstichlge5 ref2      gleinstichlge5ref2)
        (gleinstichlge5 rad1      15)
        (gleinstichlge5 rad2      15.2)
        (gleinstichlge5 h          0.2)
        (konvex                    gleinstichlge5))

; Geometrie 1, Einstich 1, ge 6

(indi gleinstichlge6ref1 gleinstichlge6ref2)

(asse  (gleinstichlge6ref1      x          0)
        (gleinstichlge6ref1      z          4)
        (gleinstichlge6ref2      x          0)
        (gleinstichlge6ref2      z          4))

(indi gleinstichlge6)

(asse  (gleinstichlge6 ref1      gleinstichlge6ref1)
        (gleinstichlge6 ref2      gleinstichlge6ref2)
        (gleinstichlge6 rad1      15.2)
        (gleinstichlge6 rad2      16.8)
        (gleinstichlge6 h          0))

; Geometrie 1, Einstich 1, ge 7

(indi gleinstichlge7ref1 gleinstichlge7ref2)

(asse  (gleinstichlge7ref1      x          0)
        (gleinstichlge7ref1      z          4)
        (gleinstichlge7ref2      x          0)
        (gleinstichlge7ref2      z          4.2))

(indi gleinstichlge7)

(asse  (gleinstichlge7 ref1      gleinstichlge7ref1)
        (gleinstichlge7 ref2      gleinstichlge7ref2)
        (gleinstichlge7 rad1      16.8)
        (gleinstichlge7 rad2      17)
        (gleinstichlge7 h          0.2)
        (konkav                    gleinstichlge7))

(indi gleinstichl)

(asse  (gleinstichl      ge1      gleinstichlge1)
        (gleinstichl      ge2      gleinstichlge2)
        (gleinstichl      ge3      gleinstichlge3)
        (gleinstichl      ge4      gleinstichlge4)
        (gleinstichl      ge5      gleinstichlge5)
        (gleinstichl      ge6      gleinstichlge6)
        (gleinstichl      ge7      gleinstichlge7))

; Geometrie 1

; Geometrie 1, Einstich 2

```



```

; Geometrie 1, Einstich 2, ge 1

(indi gleinstich2gelref1 gleinstich2gelref2)

(asse (gleinstich2gelref1 x 0)
      (gleinstich2gelref1 z 12.8)
      (gleinstich2gelref2 x 0)
      (gleinstich2gelref2 z 13))

(indi gleinstich2gel)

(asse (gleinstich2gel ref1 gleinstich2gelref1)
      (gleinstich2gel ref2 gleinstich2gelref2)
      (gleinstich2gel rad1 17)
      (gleinstich2gel rad2 16.8)
      (gleinstich2gel h 0.2)
      (konkav gleinstich2gel))

; Geometrie 1, Einstich 2, ge 2

(indi gleinstich2ge2ref1 gleinstich2ge2ref2)

(asse (gleinstich2ge2ref1 x 0)
      (gleinstich2ge2ref1 z 13)
      (gleinstich2ge2ref2 x 0)
      (gleinstich2ge2ref2 z 13))

(indi gleinstich2ge2)

(asse (gleinstich2ge2 ref1 gleinstich2ge2ref1)
      (gleinstich2ge2 ref2 gleinstich2ge2ref2)
      (gleinstich2ge2 rad1 16.8)
      (gleinstich2ge2 rad2 15.2)
      (gleinstich2ge2 h 0))

; Geometrie 1, Einstich 2, ge 3

(indi gleinstich2ge3ref1 gleinstich2ge3ref2)

(asse (gleinstich2ge3ref1 x 0)
      (gleinstich2ge3ref1 z 13)
      (gleinstich2ge3ref2 x 0)
      (gleinstich2ge3ref2 z 13.2))

(indi gleinstich2ge3)

(asse (gleinstich2ge3 ref1 gleinstich2ge3ref1)
      (gleinstich2ge3 ref2 gleinstich2ge3ref2)
      (gleinstich2ge3 rad1 15.2)
      (gleinstich2ge3 rad2 15)
      (gleinstich2ge3 h 0.2)
      (konvex gleinstich2ge3))

; Geometrie 1, Einstich 2, ge 4

```

(indi gleinstich2ge4ref1 gleinstich2ge4ref2)

(asse (gleinstich2ge4ref1 x 0)  
(gleinstich2ge4ref1 z 13.2)  
(gleinstich2ge4ref2 x 0)  
(gleinstich2ge4ref2 z 14.8))

(indi gleinstich2ge4)

(asse (gleinstich2ge4 ref1 gleinstich2ge4ref1)  
(gleinstich2ge4 ref2 gleinstich2ge4ref2)  
(gleinstich2ge4 rad1 15)  
(gleinstich2ge4 rad2 15)  
(gleinstich2ge4 h 1.6))

; Geometrie 1, Einstich 2, ge 5

(indi gleinstich2ge5ref1 gleinstich2ge5ref2)

(asse (gleinstich2ge5ref1 x 0)  
(gleinstich2ge5ref1 z 14.8)  
(gleinstich2ge5ref2 x 0)  
(gleinstich2ge5ref2 z 15))

(indi gleinstich2ge5)

(asse (gleinstich2ge5 ref1 gleinstich2ge5ref1)  
(gleinstich2ge5 ref2 gleinstich2ge5ref2)  
(gleinstich2ge5 h 1.6))

```

(gleinstich2ge7ref1 z 15)
(gleinstich2ge7ref2 x 0)
(gleinstich2ge7ref2 z 15.2))

```

```
(indi gleinstich2ge7)
```

```
(asse (gleinstich2ge7 ref1 gleinstich2ge7ref1)
(gleinstich2ge7 ref2 gleinstich2ge7ref2)
(gleinstich2ge7 rad1 18.8)
(gleinstich2ge7 rad2 19)
(gleinstich2ge7 h 0.2)
(konkav gleinstich2ge7))

```

```
(indi gleinstich2)
```

```
(asse (gleinstich2 ge1 gleinstich2ge1)
(gleinstich2 ge2 gleinstich2ge2)
(gleinstich2 ge3 gleinstich2ge3)
(gleinstich2 ge4 gleinstich2ge4)
(gleinstich2 ge5 gleinstich2ge5)
(gleinstich2 ge6 gleinstich2ge6)
(gleinstich2 ge7 gleinstich2ge7))

```

```
; Geometrie 1
```

```
; Geometrie 1, Einstich 3
```

```
; Geometrie 1, Einstich 3, ge 1
```

```
(indi gleinstich3gelref1 gleinstich3gelref2)
```

```
(asse (gleinstich3gelref1 x 0)
(gleinstich3gelref1 z 36.8)
(gleinstich3gelref2 x 0)
(gleinstich3gelref2 z 37))

```

```
(indi gleinstich3ge1)
```

```
(asse (gleinstich3ge1 ref1 gleinstich3gelref1)
(gleinstich3ge1 ref2 gleinstich3gelref2)
(gleinstich3ge1 rad1 19)
(gleinstich3ge1 rad2 18.8)
(gleinstich3ge1 h 0.2)
(konkav gleinstich3ge1))

```

```
; Geometrie 1, Einstich 3, ge 2
```

```
(indi gleinstich3ge2ref1 gleinstich3ge2ref2)
```

```
(asse (gleinstich3ge2ref1 x 0)
(gleinstich3ge2ref1 z 37)
(gleinstich3ge2ref2 x 0)
(gleinstich3ge2ref2 z 37))

```

```
(indi gleinstich3ge2)
```

```
(asse (gleinstich3ge2 ref1 gleinstich3ge2ref1)
      (gleinstich3ge2 ref2 gleinstich3ge2ref2)
      (gleinstich3ge2 rad1 18.8)
      (gleinstich3ge2 rad2 18.2)
      (gleinstich3ge2 h 0))
```

; Geometrie 1, Einstich 3, ge 3

```
(indi gleinstich3ge3ref1 gleinstich3ge3ref2)
```

```
(asse (gleinstich3ge3ref1 x 0)
      (gleinstich3ge3ref1 z 37)
      (gleinstich3ge3ref2 x 0)
      (gleinstich3ge3ref2 z 37.2))
```

```
(indi gleinstich3ge3)
```

```
(asse (gleinstich3ge3 ref1 gleinstich3ge3ref1)
      (gleinstich3ge3 ref2 gleinstich3ge3ref2))
```

```
(gleinstich3ge3 rad2 18)
(gleinstich3ge3 h 0.2)
(konvex gleinstich3ge3))
```

; Geometrie 1, Einstich 3, ge 4

```
(indi gleinstich3ge4ref1 gleinstich3ge4ref2)
```

```
(asse (gleinstich3ge4ref1 x 0)
      (gleinstich3ge4ref1 z 37.2)
      (gleinstich3ge4ref2 x 0)
      (gleinstich3ge4ref2 z 37.8))
```

```
(indi gleinstich3ge4)
```

```
(asse (gleinstich3ge4 ref1 gleinstich3ge4ref1)
      (gleinstich3ge4 ref2 gleinstich3ge4ref2)
      (gleinstich3ge4 rad1 18)
      (gleinstich3ge4 rad2 18)
      (gleinstich3ge4 h 0.6))
```

; Geometrie 1, Einstich 3, ge 5

```
(indi gleinstich3ge5ref1 gleinstich3ge5ref2)
```

```
(asse (gleinstich3ge5ref1 x 0)
      (gleinstich3ge5ref1 z 37.8)
      (gleinstich3ge5ref2 x 0)
      (gleinstich3ge5ref2 z 38))
```

```
(indi gleinstich3ge5)
```

```

        (gleinstich3ge5 rad1      18)
        (gleinstich3ge5 rad2     18.2)
        (gleinstich3ge5 h        0.2)
        (konvex                  gleinstich3ge5))

; Geometrie 1, Einstich 3, ge 6

(indi gleinstich3ge6ref1 gleinstich3ge6ref2)

(asse  (gleinstich3ge6ref1  x      0)
        (gleinstich3ge6ref1  z     38)
        (gleinstich3ge6ref2  x      0)
        (gleinstich3ge6ref2  z     38))

(indi gleinstich3ge6)

(asse  (gleinstich3ge6 ref1    gleinstich3ge6ref1)
        (gleinstich3ge6 ref2    gleinstich3ge6ref2)
        (gleinstich3ge6 rad1    18.2)
        (gleinstich3ge6 rad2    26)
        (gleinstich3ge6 h      0))

; Geometrie 1, Einstich 3, ge 7

(indi gleinstich3ge7ref1 gleinstich3ge7ref2)

(asse  (gleinstich3ge7ref1  x      0)
        (gleinstich3ge7ref1  z     38)
        (gleinstich3ge7ref2  x      0)
        (gleinstich3ge7ref2  z     40))

(indi gleinstich3ge7)

(asse  (gleinstich3ge7 ref1    gleinstich3ge7ref1)
        (gleinstich3ge7 ref2    gleinstich3ge7ref2)
        (gleinstich3ge7 rad1    26)
        (gleinstich3ge7 rad2    28)
        (gleinstich3ge7 rad3    0) ; ?
        (gleinstich3ge7 h      2)
        (konkav                  gleinstich3ge7))

(indi gleinstich3)

(asse  (gleinstich3  ge1      gleinstich3ge1)
        (gleinstich3  ge2      gleinstich3ge2)
        (gleinstich3  ge3      gleinstich3ge3)
        (gleinstich3  ge4      gleinstich3ge4)
        (gleinstich3  ge5      gleinstich3ge5)
        (gleinstich3  ge6      gleinstich3ge6)
        (gleinstich3  ge7      gleinstich3ge7))

; Geometrie 1

; Geometrie 1, Einstich 4

```

; Geometrie 1, Einstich 4, ge 1

(indi gleinstich4gelref1 gleinstich4gelref2)

(asse (gleinstich4gelref1 x 0)  
(gleinstich4gelref1 z 45)  
(gleinstich4gelref2 x 0)  
(gleinstich4gelref2 z 47))

(indi gleinstich4ge1)

(gleinstich4ge1 rad1 28)  
(gleinstich4ge1 rad2 26)  
(gleinstich4ge1 rad3 0.1)  
(gleinstich4ge1 h 2)  
(konkav gleinstich4ge1))

; Geometrie 1, Einstich 4, ge 2

(indi gleinstich4ge2ref1 gleinstich4ge2ref2)

(asse (gleinstich4ge2ref1 x 0)  
(gleinstich4ge2ref1 z 47)  
(gleinstich4ge2ref2 x 0)  
(gleinstich4ge2ref2 z 47))

(indi gleinstich4ge2)

(asse (gleinstich4ge2 ref1 gleinstich4ge2ref1)  
(gleinstich4ge2 ref2 gleinstich4ge2ref2)  
(gleinstich4ge2 rad1 26)  
(gleinstich4ge2 rad2 18.2)  
(gleinstich4ge2 h 0))

; Geometrie 1, Einstich 4, ge 3

(indi gleinstich4ge3ref1 gleinstich4ge3ref2)

(asse (gleinstich4ge3ref1 x 0)  
(gleinstich4ge3ref1 z 47)  
(gleinstich4ge3ref2 x 0)  
(gleinstich4ge3ref2 z 47))

```
(indi gleinstich4ge4ref1 gleinstich4ge4ref2)

(asse  (gleinstich4ge4ref1  x      0)
        (gleinstich4ge4ref1  z      47.2)
        (gleinstich4ge4ref2  x      0)
        (gleinstich4ge4ref2  z      47.8))
```

```
(indi gleinstich4ge4)
```

```
(asse  (gleinstich4ge4 ref1  gleinstich4ge4ref1)
        (gleinstich4ge4 ref2  gleinstich4ge4ref2)
        (gleinstich4ge4 rad1  18)
        (gleinstich4ge4 rad2  18)
        (gleinstich4ge4 h     0.6))
```

```
; Geometrie 1, Einstich 4, ge 5
```

```
(indi gleinstich4ge5ref1 gleinstich4ge5ref2)
```

```
(asse  (gleinstich4ge5ref1  x      0)
        (gleinstich4ge5ref1  z      47.8)
        (gleinstich4ge5ref2  x      0)
        (gleinstich4ge5ref2  z      48))
```

```
(indi gleinstich4ge5)
```

```
(asse  (gleinstich4ge5 ref1  gleinstich4ge5ref1)
        (gleinstich4ge5 ref2  gleinstich4ge5ref2)
        (gleinstich4ge5 rad1  18)
        (gleinstich4ge5 rad2  18.2)
        (gleinstich4ge5 h     0.2)
        (konvex               gleinstich4ge5))
```

```
; Geometrie 1, Einstich 4, ge 6
```

```
(indi gleinstich4ge6ref1 gleinstich4ge6ref2)
```

```
(asse  (gleinstich4ge6ref1  x      0)
        (gleinstich4ge6ref1  z      48)
        (gleinstich4ge6ref2  x      0)
        (gleinstich4ge6ref2  z      48))
```

```
(indi gleinstich4ge6)
```

```
(asse  (gleinstich4ge6 ref1  gleinstich4ge6ref1)
        (gleinstich4ge6 ref2  gleinstich4ge6ref2)
        (gleinstich4ge6 rad1  18.2)
        (gleinstich4ge6 rad2  18.8)
        (gleinstich4ge6 h     0))
```

```
; Geometrie 1, Einstich 4, ge 7
```

```
(indi gleinstich4ge7ref1 gleinstich4ge7ref2)
```

```

(asse (gleinstich4ge7ref1 x 0)
      (gleinstich4ge7ref1 z 48)
      (gleinstich4ge7ref2 x 0)
      (gleinstich4ge7ref2 z 48.2))

(indi gleinstich4ge7)

(asse (gleinstich4ge7 ref1 gleinstich4ge7ref1)
      (gleinstich4ge7 ref2 gleinstich4ge7ref2)
      (gleinstich4ge7 rad1 18.8)
      (gleinstich4ge7 rad2 19)
      (gleinstich4ge7 h 0.2)
      (konkav gleinstich4ge7))

(indi gleinstich4)

(asse (gleinstich4 ge1 gleinstich4ge1)
      (gleinstich4 ge2 gleinstich4ge2)
      (gleinstich4 ge3 gleinstich4ge3)
      (gleinstich4 ge4 gleinstich4ge4)
      (gleinstich4 ge5 gleinstich4ge5)
      (gleinstich4 ge6 gleinstich4ge6)
      (gleinstich4 ge7 gleinstich4ge7))

; Geometrie 1

; Geometrie 1, Einstich 5

; Geometrie 1, Einstich 5, ge 1

(indi gleinstich5gelref1 gleinstich5gelref2)

(asse (gleinstich5gelref1 x 0)
      (gleinstich5gelref1 z 59.8)
      (gleinstich5gelref2 x 0)
      (gleinstich5gelref2 z 60))

(indi gleinstich5gel)

(asse (gleinstich5gel ref1 gleinstich5gelref1)
      (gleinstich5gel ref2 gleinstich5gelref2)
      (gleinstich5gel rad1 19)
      (gleinstich5gel rad2 18.8)
      (gleinstich5gel h 0.2)
      (konkav gleinstich5gel))

; Geometrie 1, Einstich 5, ge 2

(indi gleinstich5ge2ref1 gleinstich5ge2ref2)

(asse (gleinstich5ge2ref1 x 0)
      (gleinstich5ge2ref1 z 60)
      (gleinstich5ge2ref2 x 0)
      (gleinstich5ge2ref2 z 60))

```



```
(indi gleinstich5ge2)

(asse (gleinstich5ge2 ref1 gleinstich5ge2ref1)
      (gleinstich5ge2 ref2 gleinstich5ge2ref2)
      (gleinstich5ge2 rad1 18.8)
      (gleinstich5ge2 rad2 16.2)
      (gleinstich5ge2 h 0))
```

; Geometrie 1, Einstich 5, ge 3

```
(indi gleinstich5ge3ref1 gleinstich5ge3ref2)
```

```
(asse (gleinstich5ge3ref1 x 0)
      (gleinstich5ge3ref1 z 60)
      (gleinstich5ge3ref2 x 0)
      (gleinstich5ge3ref2 z 60.2))
```

```
(indi gleinstich5ge3)
```

```
(asse (gleinstich5ge3 ref1 gleinstich5ge3ref1)
      (gleinstich5ge3 ref2 gleinstich5ge3ref2)
      (gleinstich5ge3 rad1 16.2)
      (gleinstich5ge3 rad2 16)
      (gleinstich5ge3 h 0.2)
      (konvex gleinstich5ge3))
```

; Geometrie 1, Einstich 5, ge 4

```
(indi gleinstich5ge4ref1 gleinstich5ge4ref2)
```

```
(asse (gleinstich5ge4ref1 x 0)
      (gleinstich5ge4ref1 z 60.2)
      (gleinstich5ge4ref2 x 0)
      (gleinstich5ge4ref2 z 62.8))
```

```
(indi gleinstich5ge4)
```

```
(asse (gleinstich5ge4 ref1 gleinstich5ge4ref1)
      (gleinstich5ge4 ref2 gleinstich5ge4ref2)
      (gleinstich5ge4 rad1 16)
      (gleinstich5ge4 rad2 16)
      (gleinstich5ge4 h 2.6))
```

```
(indi gleinstich4ge6)
```

```
(asse (gleinstich4ge6 ref1 gleinstich4ge6ref1)
      (gleinstich4ge6 ref2 gleinstich4ge6ref2)
      (gleinstich4ge6 rad1 18.2)
      (gleinstich4ge6 rad2 18.8)
      (gleinstich4ge6 h 0))
```

; Geometrie 1, Einstich 4, ge 7

```
(indi gleinstich4ge7ref1 gleinstich4ge7ref2)
```

```

(asse (gleinstich4ge7ref1 x 0)
      (gleinstich4ge7ref1 z 48)
      (gleinstich4ge7ref2 x 0)
      (gleinstich4ge7ref2 z 48.2))

(indi gleinstich4ge7)

(asse (gleinstich4ge7 ref1 gleinstich4ge7ref1)
      (gleinstich4ge7 ref2 gleinstich4ge7ref2)
      (gleinstich4ge7 rad1 18.8)
      (gleinstich4ge7 rad2 19)
      (gleinstich4ge7 h 0.2)
      (konkav gleinstich4ge7))

(indi gleinstich4)

(asse (gleinstich4 ge1 gleinstich4ge1)
      (gleinstich4 ge2 gleinstich4ge2)
      (gleinstich4 ge3 gleinstich4ge3)
      (gleinstich4 ge4 gleinstich4ge4)
      (gleinstich4 ge5 gleinstich4ge5)
      (gleinstich4 ge6 gleinstich4ge6)
      (gleinstich4 ge7 gleinstich4ge7))

; Geometrie 1

; Geometrie 1, Einstich 5

; Geometrie 1, Einstich 5, ge 1

(indi gleinstich5gelref1 gleinstich5gelref2)

(asse (gleinstich5gelref1 x 0)
      (gleinstich5gelref1 z 59.8)
      (gleinstich5gelref2 x 0)
      (gleinstich5gelref2 z 60))

(indi gleinstich5ge1)

(asse (gleinstich5ge1 ref1 gleinstich5ge1ref1)
      (gleinstich5ge1 ref2 gleinstich5ge1ref2)
      (gleinstich5ge1 rad1 19)
      (gleinstich5ge1 rad2 18.8)
      (gleinstich5ge1 h 0.2)
      (konkav gleinstich5ge1))

; Geometrie 1, Einstich 5, ge 2

(indi gleinstich5ge2ref1 gleinstich5ge2ref2)

(asse (gleinstich5ge2ref1 x 0)
      (gleinstich5ge2ref1 z 60)
      (gleinstich5ge2ref2 x 0)
      (gleinstich5ge2ref2 z 60))

```

(indi gleinstich5ge2)

(asse (gleinstich5ge2 ref1 gleinstich5ge2ref1)  
(gleinstich5ge2 ref2 gleinstich5ge2ref2)  
(gleinstich5ge2 rad1 18.8)  
(gleinstich5ge2 rad2 16.2)  
(gleinstich5ge2 h 0))

; Geometrie 1, Einstich 5, ge 3

(indi gleinstich5ge3ref1 gleinstich5ge3ref2)

(asse (gleinstich5ge3ref1 x 0)  
(gleinstich5ge3ref1 z 60)  
(gleinstich5ge3ref2 x 0)  
(gleinstich5ge3ref2 z 60.2))

(indi gleinstich5ge3)

(asse (gleinstich5ge3 ref1 gleinstich5ge3ref1)  
(gleinstich5ge3 ref2 gleinstich5ge3ref2)  
(gleinstich5ge3 rad1 16.2)  
(gleinstich5ge3 rad2 16)  
(gleinstich5ge3 h 0.2)  
(konvex gleinstich5ge3))

; Geometrie 1, Einstich 5, ge 4

(indi gleinstich5ge4ref1 gleinstich5ge4ref2)

(asse (gleinstich5ge4ref1 x 0)  
(gleinstich5ge4ref1 z 60.2)  
(gleinstich5ge4ref2 x 0)  
(gleinstich5ge4ref2 z 62.8))

(indi gleinstich5ge4)

(asse (gleinstich5ge4 ref1 gleinstich5ge4ref1)  
(gleinstich5ge4 ref2 gleinstich5ge4ref2)  
(gleinstich5ge4 rad1 16))

(gleinstich5ge4 h 2.6))

(indi glnut1ge2ref1 glnut1ge2ref2)

(asse (glnut1ge2ref1 x 0)  
(glnut1ge2ref1 z 82.2)  
(glnut1ge2ref2 x 0)  
(glnut1ge2ref2 z 83.8))

(indi glnut1ge2)

(asse (glnut1ge2 ref1 glnut1ge2ref1)  
(glnut1ge2 ref2 glnut1ge2ref2)  
(glnut1ge2 rad1 12.8))

```

                (glnutlge2      rad2      11.2)
                (glnutlge2      h         1.6))

; Geometrie 1, Nut 1, ge 3

(indi glnutlge3ref1 glnutlge3ref2)

(asse  (glnutlge3ref1 x         0)
       (glnutlge3ref1 z        83.8)
       (glnutlge3ref2 x         0)
       (glnutlge3ref2 z        84))

(indi glnutlge3)

(asse  (glnutlge3      ref1      glnutlge3ref1)
       (glnutlge3      ref2      glnutlge3ref2)
       (glnutlge3      rad1      11.2)
       (glnutlge3      rad2      11)
       (glnutlge3      h         0.2)
       (gerade         glnutlge3))

; Geometrie 1, Nut 1, ge 4

(indi glnutlge4ref1 glnutlge4ref2)

(asse  (glnutlge4ref1 x         0)
       (glnutlge4ref1 z        84)
       (glnutlge4ref2 x         0)
       (glnutlge4ref2 z       130))

(indi glnutlge4)

(asse  (glnutlge4      ref1      glnutlge4ref1)
       (glnutlge4      ref2      glnutlge4ref2)
       (glnutlge4      rad1      11)
       (glnutlge4      rad2      11)
       (glnutlge4      h         46))

; Geometrie 1, Nut 1, ge 5

(indi glnutlge5ref1 glnutlge5ref2)

(asse  (glnutlge5ref1 x         0)
       (glnutlge5ref1 z       130)
       (glnutlge5ref2 x         0)
       (glnutlge5ref2 z       130.2))

(indi glnutlge5)

(asse  (glnutlge5      ref1      glnutlge5ref1)
       (glnutlge5      ref2      glnutlge5ref2)
       (glnutlge5      rad1      11)
       (glnutlge5      rad2      11.2)
       (glnutlge5      h         0.2)
       (gerade         glnutlge5))

```

```

; Geometrie 1, Nut 1, ge 6

(indi glnutlge6ref1 glnutlge6ref2)

(asse (glnutlge6ref1 x 0)
      (glnutlge6ref1 z 130.2)
      (glnutlge6ref2 x 0)
      (glnutlge6ref2 z 131.8))

(indi glnutlge6)

(asse (glnutlge6 ref1 glnutlge6ref1)
      (glnutlge6 ref2 glnutlge6ref2)
      (glnutlge6 rad1 11.2)
      (glnutlge6 rad2 12.8)
      (glnutlge6 h 1.6))

; Geometrie 1, Nut 1, ge 7

(indi glnutlge7ref1 glnutlge7ref2)

(asse (glnutlge7ref1 x 0)
      (glnutlge7ref1 z 131.8)
      (glnutlge7ref2 x 0)
      (glnutlge7ref2 z 132))

(indi glnutlge7)

(asse (glnutlge7 ref1 glnutlge7ref1)
      (glnutlge7 ref2 glnutlge7ref2)
      (glnutlge7 rad1 12.8)
      (glnutlge7 rad2 13)
      (glnutlge7 h 0.2)
      (gerade glnutlge7))

(indi glnutl)

(asse (glnutl ge1 glnutlge1)
      (glnutl ge2 glnutlge2)
      (glnutl ge3 glnutlge3)
      (glnutl ge4 glnutlge4)
      (glnutl ge5 glnutlge5)
      (glnutl ge6 glnutlge6)
      (glnutl ge7 glnutlge7))

; Geometrie 1, Gewinde 1

(indi glgewanf1 glgewend1 glgewbez1)

(asse (M60x2 glgewbez1))

(asse (glgewanf1 x 0)
      (glgewanf1 z 195)
      (glgewend1 x 0))

```

```

                (glgewend1      z      210))

; Geometrie 1

(indi g1)

(asse  (g1      GA      glgewanf1)
       (g1      GE      glgewend1)
       (g1      GB      glgewbez1))

(asse  (g1      Maxrad  28)
       (g1      Maxh   210)
       (g1      K      glabsatz1)
       (g1      K      glabsatz2)
       (g1      K      glabsatz3)
       (g1      K      glabsatz4)
       (g1      K      glabsatz5)
       (g1      K      glabsatz6)
       (g1      K      gleinstich1)
       (g1      K      gleinstich2)
       (g1      K      gleinstich3)
       (g1      K      gleinstich4)
       (g1      K      gleinstich5)
       (g1      K      glnut1))

(asse  (Welle  g1))

; Die Geometrien g2 und g3 koennen analog repraesentiert werden.

; Definition der Werkstoffe

(indi w1 w2 w3 w4)

; w1

(asse  (Verguetungsstahl      w1))

; w2

(asse  (Gusseisen      w2))

; w3

(asse  (Aluminium      w3))

; w4

(asse  (Edelstahl      w4))

; Definition der Maschinen

(indi d1 d2 d3)

; d1

```

```
(asse (d1 U/MIN 2500)
      (d1 KW 5.5)
      (d1 WZ 6)
      (d1 AM stirnseiten)
      (d1 RA 1)
      (d1 ST instab))
```

```
; d2
```

```
(asse (d2 U/MIN 5000)
      (d2 KW 70)
      (d2 WZ 6)
      (d2 AM stirnseiten)
      (d2 RA 1)
      (d2 ST stab))
```

```
; d3
```

```
(asse (d3 U/MIN 7000)
      (d3 KW 90)
      (d3 WZ 6)
      (d3 AM stirnseiten)
      (d3 RA 2)
      (d3 ST stab))
```

```
; Definition der Beispielprobleme mit Geometrie g1
```

```
; glw2d2
```

```
(indi glw2d2)
```

```
(asse (zylindrisch glw2d2)
      (beliebige-Anzahl-Zentrierbohrungen glw2d2)
      (glw2d2 g g1)
      (glw2d2 w w2)
      (glw2d2 d d2))
```

```
; glw2d3
```

```
(indi glw2d3)
```

```
(asse (zylindrisch glw2d2)
      (beliebige-Anzahl-Zentrierbohrungen glw2d2)
      (glw2d2 g g1)
      (glw2d2 w w2)
      (glw2d2 d d3))
```

```
; Die anderen Beispielprobleme koennen analog repraesentiert werden.
```

## X.4 Beispiele

Einige Beispiele sollen zeigen, welche Inferenzanfragen in den dargestellten Implementierungsversuchen möglich sind, und welche Zeit dafür benötigt wird. Da die Implementierungsversuche aufgrund der oben beschriebenen Probleme keine vollständige formale Definition der Problemklassen bieten, wie sie bei dem Verfahren insgesamt erwünscht ist, können keine komplexen Anfragen, wie beispielsweise die Zuordnung eines neuen Fertigungsproblems zu einer der Problemklassen, gestellt werden. Es ist selbst dann nicht möglich, solche Anfragen zu stellen, wenn die Problemklassen vollständig im Sinne des Verfahrens formal definiert wären. Ein Beispiel für eine komplexere Anfrage, die nur erkennen sollte, ob es sich bei einem formal repräsentierten Beispiel um ein Fertigungsproblem handelt, ist 'instance? g1w2d2 Problem'. Der Versuch dieser Anfrage in TAXON ist nach 44.5 Stunden abgebrochen worden. Daher beziehen sich die hier dargestellten Beispiele lediglich auf primitivere Konzepte.

### X.4.1 Mögliche Inferenzanfragen

Im TAXON-System sind verschiedene Inferenzanfragen möglich. Für die Bearbeitung der in der Begriffsbildung interessanten Aufgaben sind insbesondere die Anfragetypen von Bedeutung, die den Identifikationsaufgaben der Begriffsbildung entsprechen.

Neben den für die Identifikationsaufgaben interessanten Inferenzanfragen berechnet TAXON mit 'classify-all' die Subsumptionshierarchie der in der TBox definierten Konzepte. Dies entspricht der Begriffshierarchie.

Bei der Benennung ist das Objekt gegeben und es wird ein Begriff gesucht, zu dem das Objekt gehört. Dies entspricht der Inferenzanfrage 'realizei individual' in TAXON. Dabei ist 'individual' ein konkretes Objekt, das in der ABox definiert sein muß, und gesucht wird ein Konzept aus der TBox, das durch dieses 'individual' erfüllt wird. Bei der Zuordnung sind sowohl das Objekt als auch der Begriff gegeben und es muß entschieden werden, ob das gegebene Objekt zu dem gegebenen Begriff gehört. In TAXON entspricht dies der Anfrage 'instance? individual concept'. Dabei ist 'individual' wiederum ein konkretes Objekt, das in der ABox definiert sein muß, und 'concept' ist ein in der TBox definiertes Konzept. Schließlich ist bei der Entdeckung ein Begriff gegeben und es muß entschieden werden, ob für den Begriff in dem gegebenen Kontext ein Objekt existiert, das zu diesem Begriff gehört. In TAXON gibt es eine Anfrage der Form 'sati? concept', die bei einem gegebenen Konzept 'concept' aus der TBox überprüft, ob das Konzept erfüllbar ist. Dieser Erfüllbarkeitstest prüft allerdings nur allgemein, ob es grundsätzlich denkbar ist, daß ein Objekt existiert, welches dieses Konzept erfüllt. 'sati?' sucht also nicht in der gegebenen ABox nach einem solchen Objekt, wie es die Entdeckung aus der Begriffsbildung fordert.

Zusätzlich zu der Berechnung der gesamten Begriffshierarchie und den Identifikationsaufgaben können zum Beispiel auch Unter- und Oberbegriffsrelationen überprüft werden. Die in TAXON realisierte Anfrage 'subs? concept<sub>x</sub> concepty' kontrolliert, ob das Konzept 'concept<sub>x</sub>' ein Oberbegriff zu dem Konzept 'concepty' ist, bzw. umgekehrt, ob das Konzept 'concepty' ein Unterbegriff zu dem Konzept 'concept<sub>x</sub>' ist.

Nachfolgend werden für jeden Inferenzanfragetyp einige Beispiele dargestellt.

### X.4.2 Konkrete Beispiele der möglichen Inferenzanfragen

Berechnung der Begriffshierarchie: 'classify-all'

Die Berechnung der Subsumptionshierarchie dauerte bei Vernachlässigung der Konzepte Absatz und Zapfen 380 Sekunden. Eine Berechnung der Subsumptionshierarchie mit den Konzepten Absatz und Zapfen ist in angemessener Zeit nicht möglich. Ein Versuch wurde nach sechseinhalb Stunden abgebrochen.

Benennung: 'realizei individual'

1. 'realizei glabsatz1'

Der erste Absatz der Geometrie 1 wurde nach 2 Sekunden richtig als steigender Absatz benannt.

2. 'realizei gleinstich1'

Der Einstich der Geometrie 1 wurde nach 2 Sekunden richtig als Einstich benannt.



3. 'realizei g1absatz3'

Der dritte Absatz der Geometrie 1 wurde nach 2 Sekunden richtig als Mittelabsatz benannt.

Die drei Individuen g1absatz1, g1einstich1, und g1absatz3 dieser drei Anfragen sind in der ABox lediglich durch die Menge der zu diesen Absätzen bzw. zu diesem Einstich gehörenden Konturkoordinaten definiert. Es ist keine explizite Zusicherung erfolgt, daß es sich bei diesem Individuum tatsächlich um einen steigenden Absatz, einen Einstich, bzw. einen Mittelabsatz handelt. Die benannten Konzepte sind mit allgemeinen Regeln und Forderungen in der TBox definiert.

Zuordnung: 'instance? individual concept'

4. 'instance? g1absatz1 steigender-Absatz'

Diese Zuordnung zwischen dem Individuum g1absatz1 und dem Konzept steigender-Absatz wurde nach 1 Sekunde als korrekt erkannt.

5. 'instance? g1einstich1 Einstich'

Diese Zuordnung zwischen dem Individuum g1einstich1 und dem Konzept Einstich wurde ebenfalls nach 1 Sekunde korrekt erkannt.

6. 'instance? g1absatz3 Mittelabsatz'

Diese Zuordnung zwischen dem Individuum g1absatz3 und dem Konzept Mittelabsatz wurde ebenfalls nach 1 Sekunde korrekt erkannt.

7. 'instance? g1 Gewinde'

Nach 26 Sekunden erkannte das System, daß die Geometrie g1 ein Gewinde besitzt.

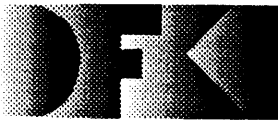
8. 'instance? w1 Werkstoff'

Nach 1 Sekunde bestätigte TAXON, daß es sich bei dem Individuum w1 um einen Werkstoff handelt.

9. 'instance? d1 Maschine'

nicht eindeutig festzulegen sind. Stellt man eine Anfrage, die nach mehreren Stunden abgebrochen werden muß, kann nicht darüber geurteilt werden, ob die Ursache für diese unzureichende Leistung im System oder in der Implementierung liegt.

Betrachtet man allerdings die Komplexität der Beispiele und berücksichtigt, daß keine komplexeren Anfragen an das System gestellt werden konnten, ohne eine Laufzeit von mehreren Stunden in Kauf zu nehmen, so kann auf jeden Fall die Aussage gemacht werden, daß ein taxonomisches Wissensrepräsentationssystem in dieser Form ungeeignet ist, um die Aufgaben der menschlichen Begriffsbildung eines Experten im Bereich des Maschinenbaus angemessen zu bearbeiten.



**Deutsches  
Forschungszentrum  
für Künstliche  
Intelligenz GmbH**

DFKI  
-Bibliothek-  
PF 2080  
D-6750 Kaiserslautern  
FRG

## DFKI Publikationen

Die folgenden DFKI Veröffentlichungen sowie die aktuelle Liste von allen bisher erschienenen Publikationen können von der oben angegebenen Adresse bezogen werden.

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

## DFKI Publications

The following DFKI publications or the list of all published papers so far can be ordered from the above address.

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

### DFKI Research Reports

#### RR-90-07

*Elisabeth André, Thomas Rist:*

Wissensbasierte Informationspräsentation:

Zwei Beiträge zum Fachgespräch Graphik und KI

1. Ein planbasierter Ansatz zur Synthese illustrierter Dokumente
2. Wissensbasierte Perspektivenwahl für die automatische Erzeugung von 3D-Objektdarstellungen

24 Seiten

#### RR-90-08

*Andreas Dengel:* A Step Towards Understanding Paper Documents

25 pages

#### RR-90-09

*Susanne Biundo:* Plan Generation Using a Method of Deductive Program Synthesis

17 pages

#### RR-90-10

*Franz Baader, Hans-Jürgen Bürckert, Bernhard Hollunder, Werner Nutt, Jörg H. Siekmann:*

Concept Logics

26 pages

#### RR-90-11

*Elisabeth André, Thomas Rist:* Towards a Plan-Based Synthesis of Illustrated Documents

14 pages

#### RR-90-12

*Harold Boley:* Declarative Operations on Nets

43 pages

#### RR-90-13

*Franz Baader:* Augmenting Concept Languages by Transitive Closure of Roles: An Alternative to Terminological Cycles

40 pages

#### RR-90-14

*Franz Schmalhofer, Otto Kühn, Gabriele Schmidt:*

Integrated Knowledge Acquisition from Text, Previously Solved Cases, and Expert Memories

20 pages

#### RR-90-15

*Harald Trost:* The Application of Two-level

Morphology to Non-concatenative German

Morphology

13 pages

#### RR-90-16

*Franz Baader, Werner Nutt:* Adding Homomorphisms to Commutative/Monoidal Theories, or: How Algebra Can Help in Equational Unification

25 pages

#### RR-90-17

*Stephan Busemann:*

Generalisierte Phasenstrukturgrammatiken und ihre Verwendung zur maschinellen Sprachverarbeitung

114 Seiten

#### RR-91-01

*Franz Baader, Hans-Jürgen Bürckert, Bernhard*

*Nebel, Werner Nutt, Gert Smolka:* On the

Expressivity of Feature Logics with Negation, Functional Uncertainty, and Sort Equations

20 pages

#### RR-91-02

*Francesco Donini, Bernhard Hollunder, Maurizio*

*Lenzerini, Alberto Marchetti Spaccamela, Daniele*

*Nardi, Werner Nutt:* The Complexity of Existential Quantification in Concept Languages

22 pages

#### RR-91-03

*B.Hollunder, Franz Baader:* Qualifying Number Restrictions in Concept Languages

34 pages

**RR-91-04**

*Harald Trost: X2MORE: A Morphological Component Based on Augmented Two-Level Morphology*  
19 pages

**RR-91-05**

*Wolfgang Wahlster, Elisabeth André, Winfried Graf, Thomas Rist: Designing Illustrated Texts: How Language Production is Influenced by Graphics Generation.*  
17 pages

**RR-91-06**

*Elisabeth André, Thomas Rist: Synthesizing Illustrated Documents: A Plan-Based Approach*  
11 pages

**RR-91-07**

*Günter Neumann, Wolfgang Finkler: A Head-Driven Approach to Incremental and Parallel Generation of Syntactic Structures*  
13 pages

**RR-91-08**

*Wolfgang Wahlster, Elisabeth André, Som Bandyopadhyay, Winfried Graf, Thomas Rist: WIP: The Coordinated Generation of Multimodal Presentations from a Common Representation*  
23 pages

**RR-91-09**

*Hans-Jürgen Bürckert, Jürgen Müller, Achim Schupeta: RATMAN and its Relation to Other Multi-Agent Testbeds*  
31 pages

**RR-91-10**

*Franz Baader, Philipp Hanschke: A Scheme for Integrating Concrete Domains into Concept Languages*  
31 pages

**RR-91-11**

*Bernhard Nebel: Belief Revision and Default Reasoning: Syntax-Based Approaches*  
37 pages

**RR-91-12**

*J.Mark Gawron, John Nerbonne, Stanley Peters: The Absorption Principle and E-Type Anaphora*  
33 pages

**RR-91-13**

*Gert Smolka: Residuation and Guarded Rules for Constraint Logic Programming*  
17 pages

**RR-91-14**

*Peter Breuer, Jürgen Müller: A Two Level Representation for Spatial Relations, Part I*  
27 pages

**RR-91-15**

*Bernhard Nebel, Gert Smolka: Attributive Description Formalisms ... and the Rest of the World*  
20 pages

**RR-91-16**

*Stephan Busemann: Using Pattern-Action Rules for the Generation of GPSG Structures from Separate Semantic Representations*  
18 pages

**RR-91-17**

*Andreas Dengel, Nelson M. Mattos: The Use of Abstraction Concepts for Representing and Structuring Documents*  
17 pages

**RR-91-18**

*John Nerbonne, Klaus Netter, Abdel Kader Diagne, Ludwig Dickmann, Judith Klein: A Diagnostic Tool for German Syntax*  
20 pages

**RR-91-19**

*Munindar P. Singh: On the Commitments and Precommitments of Limited Agents*  
15 pages

**RR-91-20**

*Christoph Klauck, Ansgar Bernardi, Ralf Legleitner: FEAT-Rep: Representing Features in CAD/CAM*  
48 pages

**RR-91-21**

*Klaus Netter: Clause Union and Verb Raising Phenomena in German*  
38 pages

**RR-91-22**

*Andreas Dengel: Self-Adapting Structuring and Representation of Space*  
27 pages

**RR-91-23**

*Michael Richter, Ansgar Bernardi, Christoph Klauck, Ralf Legleitner: Akquisition und Repräsentation von technischem Wissen für Planungsaufgaben im Bereich der Fertigungstechnik*  
24 Seiten

**RR-91-24**

*Jochen Heinsohn: A Hybrid Approach for Modeling Uncertainty in Terminological Logics*  
22 pages

**RR-91-25**

*Karin Harbusch, Wolfgang Finkler, Anne Schauder: Incremental Syntax Generation with Tree Adjoining Grammars*  
16 pages

**RR-91-26**

*M. Bauer, S. Biundo, D. Dengler, M. Hecking,  
J. Koehler, G. Merziger:*  
Integrated Plan Generation and Recognition  
- A Logic-Based Approach -  
17 pages

**RR-91-27**

*A. Bernardi, H. Boley, Ph. Hanschke,  
K. Hinkelmann, Ch. Klauck, O. Kühn,  
P. Leitner, M. Meyer, M. M. Dickson*

**DFKI Technical Memos****TM-91-01**

*Jana Köhler:* Approaches to the Reuse of Plan  
Schemata in Planning Formalisms  
52 pages

**TM-91-02**

*Knut Hinkelmann:* Bidirectional Reasoning of Horn  
Clause Programs: Transformation and Compilation

**ARC-TEC:** Acquisition, Representation and  
Compilation of Technical Knowledge  
18 pages

**RR-91-28**

*Rolf Backofen, Harald Trost, Hans Uszkoreit:*  
Linking Typed Feature Formalisms and  
Terminological Knowledge Representation  
Languages in Natural Language Front-Ends  
11 pages

**RR-91-29**

*Hans Uszkoreit:* Strategies for Adding Control  
Information to Declarative Grammars  
17 pages

**RR-91-30**

*Dan Flickinger, John Nerbonne:*  
Inheritance and Complementation: A Case Study of  
Easy Adjectives and Related Nouns  
39 pages

**RR-91-31**

*H.-U. Krieger, J. Nerbonne:*  
Feature-Based Inheritance Networks for  
Computational Lexicons  
11 pages

**RR-91-32**

*Rolf Backofen, Lutz Euler, Günther Görz:*  
Towards the Integration of Functions, Relations and  
Types in an AI Programming Language  
14 pages

**RR-91-33**

*Franz Baader, Klaus Schulz:*  
Unification in the Union of Disjoint Equational  
Theories: Combining Decision Procedures

**TM-91-03**

*Otto Kühn, Marc Linster, Gabriele Schmidt:*  
Clamping, COKAM, KADS, and OMOS:  
The Construction and Operationalization  
of a KADS Conceptual Model  
20 pages

**TM-91-04**

*Harold Boley (Ed.):*  
A sampler of Relational/Functional Definitions  
12 pages

**TM-91-05**

*Jay C. Weber, Andreas Dengel, Rainer Bleisinger:*  
Theoretical Consideration of Goal Recognition  
Aspects for Understanding Information in Business  
Letters  
10 pages

**TM-91-06**

*Johannes Stein:* Aspects of Cooperating Agents  
22 pages

**TM-91-08**

*Munindar P. Singh:* Social and Psychological  
Commitments in Multiagent Systems  
11 pages

**TM-91-09**

*Munindar P. Singh:* On the Semantics of Protocols  
Among Distributed Intelligent Agents  
18 pages

**TM-91-10**

*Béla Buschauer, Peter Poller, Anne Schauder, Karin  
Harbusch:* Tree Adjoining Grammars mit  
Unifikation

**TM-91-13***Knut Hinkelmann:*

Forward Logic Evaluation: Developing a Compiler  
from a Partially Evaluated Meta Interpreter  
16 pages

**TM-91-14**

*Rainer Bleisinger, Rainer Hoch, Andreas Dengel:*  
ODA-based modeling for document analysis  
14 pages

---

**DFKI Documents****D-91-03**

*Harold Boley, Klaus Elsbernd, Hans-Günther Hein,  
Thomas Krause:* RFM Manual: Compiling  
RELFUN into the Relational/Functional Machine  
43 pages

**D-91-04**

DFKI Wissenschaftlich-Technischer Jahresbericht  
1990  
93 Seiten

**D-91-06**

*Gerd Kamp:* Entwurf, vergleichende Beschreibung  
und Integration eines Arbeitsplanerstellungssystems  
für Drehteile  
130 Seiten

**D-91-07**

*Ansgar Bernardi, Christoph Klauck, Ralf Legleitner*  
TEC-REP: Repräsentation von Geometrie- und  
Technologieinformationen  
70 Seiten

**D-91-08**

*Thomas Krause:* Globale Datenflußanalyse und  
horizontale Compilation der relational-funktionalen  
Sprache RELFUN  
137 Seiten

**D-91-09**

*David Powers, Lary Reeker (Eds.):*  
Proceedings MLNLO'91 - Machine Learning of  
Natural Language and Ontology  
211 pages  
**Note:** This document is available only for a  
nominal charge of 25 DM (or 15 US-\$).

**D-91-10**

*Donald R. Steiner, Jürgen Müller (Eds.):*  
MAAMAW'91: Pre-Proceedings of the 3rd  
European Workshop on „Modeling Autonomous  
Agents and Multi-Agent Worlds“  
246 pages  
**Note:** This document is available only for a  
nominal charge of 25 DM (or 15 US-\$).

**D-91-11**

*Thilo C. Horstmann:* Distributed Truth Maintenance  
61 pages

**D-91-12***Bernd Bachmann:*

HieraCon - a Knowledge Representation System  
with Typed Hierarchies and Constraints  
75 pages

**D-91-13**

International Workshop on Terminological Logics  
*Organizers: Bernhard Nebel, Christof Peltason,  
Kai von Luck*  
131 pages

**D-91-14**

*Erich Achilles, Bernhard Hollunder, Armin Laux,  
Jörg-Peter Mohren:* KRIS: Knowledge  
Representation and Inference System  
- Benutzerhandbuch -  
28 Seiten

**D-91-15**

*Harold Boley, Philipp Hanschke, Martin Harm,  
Knut Hinkelmann, Thomas Labisch, Manfred  
Meyer, Jörg Müller, Thomas Oltzen, Michael  
Sintek, Werner Stein, Frank Steinle:*  
µCAD2NC: A Declarative Lathe-Worplanning  
Model Transforming CAD-like Geometries into  
Abstract NC Programs  
100 pages

**D-91-16**

*Jörg Thoben, Franz Schmalhofer, Thomas Reinartz:*  
Wiederholungs-, Varianten- und Neuplanung bei der  
Fertigung rotationssymmetrischer Drehteile  
134 Seiten

**D-91-17**

*Andreas Becker:*  
Analyse der Planungsverfahren der KI im Hinblick  
auf ihre Eignung für die Arbeitsplanung  
86 Seiten

**D-91-18**

*Thomas Reinartz:* Definition von Problemklassen  
im Maschinenbau als eine Begriffsbildungsaufgabe  
107 Seiten

**D-91-19**

*Peter Wazinski:* Objektlokalisierung in graphischen  
Darstellungen  
110 Seiten



# **Definition von Problemklassen im Maschinenbau als eine Begriffsbildungsaufgabe**

**Thomas Rehnartz**

**D-91-18**

Document